

PRAKTIKUM KIMIA KUANTUM

Bagian 3: Membuat Laporan & Penutup

$$\hat{H}\psi = E\psi$$

$$E_n = -\frac{R_H}{n^2}$$



KASMUI

PRAKTIKUM KIMIA KUANTUM

BAGIAN III: MEMBUAT LAPORAN DAN PENUTUP



KATA PENGANTAR

BAGIAN III: MEMBUAT LAPORAN DAN PENUTUP

Puji syukur ke hadirat Allah Swt. atas limpahan rahmat, ilmu, kesempatan, dan kemudahan sehingga **Bagian III: Membuat Laporan dan Penutup** dalam buku *Praktikum Kimia Kuantum* ini dapat disusun sebagai bagian akhir dari rangkaian pembelajaran praktikum kimia kuantum dan kimia komputasi. Bagian ini mencakup **Bab 11**, **Bab 12**, dan **Lampiran 1**, yang secara khusus diarahkan untuk membimbing mahasiswa dalam membaca output, memvalidasi data, mengolah hasil komputasi, membuat grafik, menyusun laporan, melaksanakan proyek mini, mengikuti evaluasi akhir, memahami istilah penting, serta menjaga keteraturan file dan reproduibilitas kerja ilmiah.

Bagian III ini disusun dengan kesadaran bahwa praktikum kimia kuantum tidak selesai hanya ketika perhitungan berhasil dijalankan. Dalam praktik kimia komputasi, keberhasilan memperoleh file output belum otomatis berarti mahasiswa telah menghasilkan data ilmiah yang sah. Output harus dibaca secara teliti, diperiksa statusnya, divalidasi konvergensinya, ditafsirkan satuannya, dibandingkan dengan teori atau literatur, diolah menjadi tabel dan grafik, lalu ditulis dalam laporan yang sistematis. Dengan demikian, bagian ini menempatkan mahasiswa pada tahap akhir yang sangat penting: mengubah hasil perhitungan menjadi pengetahuan ilmiah yang dapat dipertanggungjawabkan.

Bab 11 membahas **Analisis Output, Validasi Data, dan Otomatisasi Laporan dengan Python**. Bab ini dirancang untuk melatih mahasiswa membaca file output komputasi secara cermat. Mahasiswa diarahkan untuk mengenali informasi penting seperti metode, basis set, charge, multiplicity, energi total, geometri akhir, momen dipol, frekuensi, HOMO-LUMO, dan status konvergensi. Keterampilan ini sangat penting karena kesalahan membaca output dapat menyebabkan kesalahan analisis dan kesimpulan.

Selain membaca output, Bab 11 juga menekankan pentingnya validasi data. Mahasiswa harus memeriksa apakah job selesai normal, apakah SCF konvergen, apakah struktur hasil optimasi telah mencapai kriteria konvergensi, apakah terdapat frekuensi imajiner, apakah charge dan multiplicity benar, serta apakah geometri akhir masuk akal secara kimia. Validasi ini menjadi pembeda antara hasil komputasi mentah dan data ilmiah yang layak digunakan. Melalui pembahasan kesalahan umum, mahasiswa juga diajak mengenali masalah seperti struktur awal buruk, charge salah, multiplicity salah, basis set tidak sesuai, SCF tidak konvergen, salah membaca satuan energi, dan salah menafsirkan frekuensi imajiner.

Bab 11 kemudian memperkenalkan penggunaan Python untuk membaca output, mengekstrak data penting, membuat ringkasan CSV, dan menghasilkan grafik laporan. Penggunaan Python dalam bagian ini bukan dimaksudkan untuk menjadikan mahasiswa sebagai programmer profesional, melainkan untuk membekali mereka dengan keterampilan dasar pengolahan data ilmiah. Python membantu mahasiswa bekerja lebih tertib, mengurangi kesalahan manual, membuat grafik yang

konsisten, serta mendukung reproduibilitas laporan. Namun, bagian ini juga menegaskan bahwa Python tidak menggantikan pemahaman kimia. Kode dapat membantu mengolah data, tetapi validasi dan interpretasi tetap harus dilakukan oleh mahasiswa berdasarkan teori dan penalaran ilmiah.

Bab 12 membahas **Proyek Mini, Evaluasi Akhir, Glosarium, dan Lampiran Kode Python**. Bab ini merupakan bagian integratif yang mengarahkan mahasiswa untuk menerapkan seluruh keterampilan yang telah dipelajari ke dalam proyek mini. Mahasiswa diberi pilihan topik seperti analisis HOMO-LUMO senyawa aromatik tersubstitusi, perbandingan IR komputasi dan IR literatur, optimasi geometri molekul obat sederhana, studi efek pelarut terhadap kestabilan ion, analisis konformer etanol atau butana, simetri dan polaritas molekul anorganik, energi ikatan hidrogen dimer air, serta energi reaksi sederhana fase gas dan pelarut.

Melalui proyek mini, mahasiswa belajar bahwa kimia komputasi bukan sekadar menjalankan software, tetapi merupakan proses ilmiah yang dimulai dari perumusan masalah. Mahasiswa harus memilih molekul target, menentukan metode komputasi, memilih software, merancang kode Python pendukung, menentukan data yang dikumpulkan, menyusun proposal, menjalankan perhitungan, memvalidasi output, membuat grafik, menyusun laporan, dan mempresentasikan hasil. Dengan demikian, proyek mini menjadi latihan kemandirian ilmiah dalam skala terbatas tetapi bermakna.

Bab 12 juga menyajikan format proposal proyek mini dan format laporan proyek mini. Proposal membantu mahasiswa merancang pekerjaan sebelum perhitungan dijalankan, sedangkan laporan menjadi dokumen pertanggungjawaban ilmiah setelah proyek selesai. Format laporan yang dibahas mencakup abstrak, pendahuluan, dasar teori, metode komputasi, kode Python, hasil dan pembahasan, kesimpulan, daftar pustaka, serta lampiran input-output. Dengan format ini, mahasiswa diharapkan mampu menulis laporan yang tidak hanya lengkap secara administratif, tetapi juga valid secara metodologis dan kuat secara argumentatif.

Evaluasi akhir praktikum dalam Bab 12 dirancang untuk menilai kompetensi mahasiswa secara menyeluruh. Evaluasi tidak hanya berupa ujian teori singkat, tetapi juga mencakup ujian membaca output, ujian menjalankan kode Python, ujian interpretasi grafik, presentasi proyek mini, dan laporan akhir. Bentuk evaluasi ini dipilih karena praktikum kimia kuantum memiliki karakter multidimensional: konseptual, teknis, numerik, visual, dan komunikatif. Mahasiswa dinilai bukan hanya dari hasil akhir, tetapi juga dari proses berpikir, ketelitian validasi, kemampuan diagnostik, dan kemampuan menyampaikan hasil secara ilmiah.

Glosarium yang disajikan dalam Bab 12 membantu mahasiswa memahami istilah utama kimia kuantum dan kimia komputasi, seperti *ab initio*, basis set, ChemCompute, DFT, energi total, frekuensi imajiner, fungsi gelombang, Hartree-Fock, HOMO, LUMO, momen dipol, optimasi geometri, orbital molekul, SCF, simetri molekul, dan *zero-point energy*. Glosarium ini penting karena ketepatan istilah merupakan bagian dari ketepatan berpikir ilmiah. Kesalahan membedakan software, metode, basis set, job, output, dan data dapat menyebabkan kekeliruan dalam laporan.

Bagian akhir Bab 12 memuat lampiran kode Python yang mendukung seluruh praktikum, mulai dari partikel dalam kotak, optimasi geometri sederhana, Hartree-Fock dan basis set, DFT dan

HOMO-LUMO, simetri dan momen dipol, spektrum IR, energi reaksi dan termodinamika, parser output komputasi, hingga template laporan otomatis. Kode-kode ini dapat digunakan ulang, dimodifikasi, dan dikembangkan sesuai kebutuhan pembelajaran. Dengan adanya lampiran kode, mahasiswa memiliki arsip kerja yang dapat mendukung pembelajaran mandiri dan pelaporan yang lebih rapi.

Lampiran 1 dalam bagian ini disusun untuk melengkapi hal-hal teknis yang belum secara khusus dibahas dalam bab utama. Lampiran tersebut memuat standar organisasi folder, standar penamaan file, checklist sebelum menjalankan perhitungan, checklist setelah perhitungan selesai, standar satuan penting, format metadata perhitungan, troubleshooting singkat, standar kelengkapan file, prinsip kejujuran akademik, prinsip reproduktibilitas, template README praktikum, dan checklist akhir sebelum pengumpulan. Lampiran ini diharapkan membantu mahasiswa menjaga keteraturan kerja dan menghindari kesalahan teknis yang sering terjadi dalam praktikum.

Bagian III ini juga menekankan pentingnya **kejujuran akademik**. Mahasiswa tidak boleh mengarang data, mengubah angka agar sesuai harapan, menyembunyikan output yang gagal, menggunakan data contoh sebagai data praktikum, atau membuat grafik dari file yang tidak sesuai. Dalam kimia komputasi, kejujuran akademik dapat diperiksa melalui keterlacakan file input, file output, CSV, kode Python, grafik, dan laporan. Oleh karena itu, setiap data yang ditulis dalam laporan harus dapat ditelusuri sumbernya.

Selain itu, bagian ini menegaskan pentingnya **reproduktibilitas**. Hasil komputasi yang baik harus dapat diperiksa ulang. Informasi seperti software, metode, basis set, charge, multiplicity, jenis job, model pelarut, file input, file output, data CSV, kode Python, grafik, dan catatan validasi harus tersedia secara jelas. Tanpa reproduktibilitas, laporan praktikum menjadi lemah karena pembaca tidak dapat memastikan bagaimana hasil diperoleh.

Bagi mahasiswa, Bagian III ini diharapkan menjadi panduan akhir untuk menyelesaikan praktikum secara bertanggung jawab. Bagi dosen atau instruktur, bagian ini dapat digunakan sebagai acuan penilaian, panduan evaluasi, dan standar pelaporan. Bagi calon guru kimia, bagian ini dapat menjadi contoh bagaimana pembelajaran kimia komputasi dapat dirancang secara sistematis, mulai dari konsep, praktik, data, validasi, laporan, hingga evaluasi.

Penulis menyadari bahwa penyusunan bagian ini masih dapat dikembangkan lebih lanjut sesuai perkembangan software, metode komputasi, pustaka Python, dan kebutuhan pembelajaran kimia modern. Oleh karena itu, bagian ini terbuka untuk penyempurnaan, baik dari sisi kedalaman teori, variasi proyek, format evaluasi, maupun pengembangan kode. Namun, prinsip dasarnya tetap sama: praktikum kimia kuantum harus melatih mahasiswa berpikir ilmiah, bekerja teliti, memvalidasi data, menulis secara sistematis, dan menyampaikan kesimpulan berdasarkan bukti.

Akhirnya, semoga **Bagian III: Membuat Laporan dan Penutup** ini dapat membantu mahasiswa menyelesaikan Praktikum Kimia Kuantum dengan lebih utuh. Output yang panjang dapat diubah menjadi data yang rapi. Data dapat diolah menjadi tabel dan grafik. Grafik dapat dijelaskan dalam analisis. Analisis dapat dirangkum dalam kesimpulan. Kesimpulan dapat dipertanggungjawabkan melalui lampiran input-output dan kode Python. Dengan demikian, praktikum kimia kuantum tidak

hanya menjadi kegiatan teknis, tetapi menjadi latihan ilmiah yang membentuk ketelitian, kejujuran, kemandirian, dan tanggung jawab akademik.

Semoga buku ini memberi manfaat bagi mahasiswa, dosen, calon guru kimia, dan siapa pun yang ingin mempelajari kimia kuantum dan kimia komputasi secara praktis, sistematis, dan bertanggung jawab.

Semarang, Juni 2026

Penulis,

A handwritten signature in black ink, appearing to be 'Kasmui', written over a horizontal line.

Kasmui

DAFTAR ISI

KATA PENGANTAR.....	4
DAFTAR ISI.....	8
BAGIAN 3.....	22
MEMBUAT LAPORAN & PENUTUP	22
BAB 11	23
ANALISIS OUTPUT, VALIDASI DATA, DAN OTOMATISASI LAPORAN DENGAN PYTHON	23
Pengantar Bab 11	23
11.1 Membaca File Output Komputasi	24
1. Pengertian File Output Komputasi	24
2. Latar Historis Pembacaan Output dalam Kimia Komputasi	25
3. Struktur Umum File Output	26
4. Membaca Metode Perhitungan	26
5. Membaca Basis Set	27
6. Membaca Charge	28
7. Membaca Multiplicity	29
8. Membaca Energi Total	30
9. Energi Total, Energi Elektronik, dan Energi Termokimia	30
10. Membaca Geometri Akhir	31
11. Membaca Momen Dipol	32
12. Membaca Frekuensi	33
13. Frekuensi Imajiner dan Maknanya	33
14. Membaca HOMO-LUMO	34
15. Membaca Status Konvergensi SCF	35
16. Membaca Status Konvergensi Optimasi Geometri	36
17. Membaca Pesan Akhir Output	36
18. Strategi Praktis Mencari Informasi dalam Output	37
19. Contoh Ringkasan Informasi Output	37
20. Membaca Output sebagai Proses Validasi Awal	38
21. Analisis Kritis: Output Bukan Kebenaran Otomatis	39

22. Contoh Kasus: Membaca Output Air	39
23. Contoh Kasus: Membaca Output CO ₂	40
24. Contoh Kasus: Membaca Output NH ₄ ⁺	40
25. Hubungan Membaca Output dengan Praktikum Sebelumnya	41
26. Implikasi Metodologis Membaca Output	41
27. Implikasi Pedagogis Membaca Output	42
28. Format Catatan Manual Sebelum Otomatisasi.....	42
29. Kesalahan Umum Saat Membaca Output	43
30. Sintesis Subbab 11.1.....	43
31. Transisi ke Subbab 11.2.....	44
11.2 Validasi Hasil Praktikum	44
1. Pengertian Validasi dalam Kimia Komputasi.....	45
2. Latar Historis Pentingnya Validasi	46
3. Hubungan Validasi dengan Reprodusibilitas.....	46
4. Kriteria 1: Job Selesai Normal	47
5. Mengapa Job Selesai Normal Belum Cukup	48
6. Kriteria 2: Struktur Konvergen	49
7. Makna Konvergensi Struktur secara Kimia	49
8. Kriteria 3: Tidak Ada Frekuensi Imajiner untuk Struktur Minimum.....	50
9. Frekuensi Imajiner dan Keadaan Transisi.....	50
10. Contoh Validasi Frekuensi pada Dimer Air.....	51
11. Kriteria 4: Geometri Masuk Akal	52
12. Contoh Geometri Masuk Akal.....	52
13. Geometri dan Visualisasi.....	53
14. Kriteria 5: Charge dan Multiplicity Benar.....	53
15. Contoh Kesalahan Charge	54
16. Contoh Kesalahan Multiplicity.....	55
17. Kriteria 6: Hasil Dapat Dibandingkan dengan Literatur	55
18. Perbandingan Literatur sebagai Validasi Tren	56
19. Batasan Perbandingan Literatur.....	56
20. Format Tabel Validasi Hasil Praktikum.....	57
21. Validasi pada Praktikum Optimasi Geometri.....	58
22. Validasi pada Praktikum Orbital HOMO-LUMO	58

23. Validasi pada Praktikum Momen Dipol	59
24. Validasi pada Praktikum Frekuensi dan Spektrum IR.....	59
25. Validasi pada Praktikum Energi Reaksi dan Termodinamika	60
26. Validasi Konsistensi Metode	60
27. Validasi Model Pelarut	61
28. Validasi Satuan Energi	61
29. Validasi Data Sebelum Otomatisasi Python	62
30. Contoh Lembar Validasi Praktikum.....	63
31. Analisis Kritis terhadap Validasi.....	63
32. Kesalahan Umum dalam Validasi	64
33. Implikasi Keilmuan Validasi.....	64
34. Implikasi Pedagogis Validasi	65
35. Sintesis Subbab 11.2.....	65
36. Transisi ke Subbab 11.3.....	66
11.3 Kesalahan Umum.....	66
1. Makna Kesalahan dalam Kimia Komputasi	67
2. Latar Historis Kesalahan Komputasi	68
3. Kesalahan 1: Struktur Awal Buruk	68
4. Dampak Struktur Awal Buruk terhadap Optimasi.....	69
5. Cara Menghindari Struktur Awal Buruk	70
6. Kesalahan 2: Charge Salah	70
7. Contoh Kesalahan Charge pada Protonasi Amonia	71
8. Cara Menghindari Kesalahan Charge	72
9. Kesalahan 3: Multiplicity Salah.....	72
10. Contoh Kesalahan Multiplicity pada O ₂	73
11. Cara Menghindari Kesalahan Multiplicity	73
12. Kesalahan 4: Basis Set Tidak Sesuai	74
13. Dampak Basis Set Tidak Sesuai.....	75
14. Cara Memilih Basis Set secara Wajar	75
15. Kesalahan 5: SCF Tidak Konvergen.....	76
16. Dampak SCF Tidak Konvergen	77
17. Cara Mengatasi SCF Tidak Konvergen.....	77
18. Kesalahan 6: Salah Membaca Satuan Energi	78

19. Cara Membaca dan Mengonversi Energi dengan Benar	79
20. Kesalahan 7: Salah Menafsirkan Frekuensi Imajiner	79
21. Contoh Salah Menafsirkan Frekuensi Imajiner	80
22. Cara Memeriksa Frekuensi Imajiner	80
23. Kesalahan Tambahan: Mengambil Energi dari Bagian Output yang Salah	81
24. Kesalahan Tambahan: Mencampur Metode dalam Perbandingan Energi	81
25. Kesalahan Tambahan: Mengabaikan Model Pelarut	82
26. Kesalahan Tambahan: Menganggap Output Grafik Selalu Benar	82
27. Diagnosis Kesalahan Berdasarkan Gejala Output	83
28. Strategi Umum Memperbaiki Kesalahan	84
29. Kesalahan sebagai Bagian dari Pembelajaran	84
30. Tabel Ringkas Kesalahan, Dampak, dan Solusi	85
31. Implikasi Keilmuan Kesalahan Umum	85
32. Implikasi Pedagogis Kesalahan Umum	86
33. Format Catatan Kesalahan dalam Laporan Praktikum	86
34. Sintesis Subbab 11.3	87
35. Transisi ke Subbab 11.4	87
11.4 Python untuk Parsing Output	87
1. Pengertian Parsing Output	88
2. Latar Historis Parsing dalam Kimia Komputasi	89
3. Prinsip Dasar Parser Output	90
4. Informasi yang Akan Diekstrak	91
5. Struktur Folder Parsing Output	91
6. Nama File Kode yang Dianjurkan	92
7. Kode Python Utama untuk Parsing Output	92
8. Penjelasan Bagian Pembacaan File Log	101
9. Penjelasan Pencarian Status Job	102
10. Penjelasan Pencarian Metode dan Basis Set	102
11. Penjelasan Pencarian Charge dan Multiplicity	103
12. Penjelasan Pencarian Energi Total	103
13. Penjelasan Pencarian Momen Dipol	104
14. Penjelasan Ekstraksi Frekuensi	104
15. Penjelasan Pencarian HOMO-LUMO	105

16. Output CSV yang Dihasilkan.....	106
17. Contoh Hasil CSV	106
18. Validasi Hasil Parsing.....	107
19. Analisis Kritis terhadap Parsing Otomatis.....	107
20. Membaca File Log Banyak Sistem	108
21. Menghubungkan Parsing dengan Validasi Praktikum	109
22. Kode Tambahan untuk Validasi Awal Otomatis	109
23. Parsing Output Gaussian, ORCA, dan GAMESS	110
24. Menambahkan Pola Regex Baru	111
25. Prinsip Regular Expression dalam Parsing.....	111
26. Keterbatasan Regex Sederhana	112
27. Aplikasi Parsing untuk Laporan Praktikum.....	112
28. Contoh Analisis Hasil Parsing	113
29. Parsing Tidak Menggantikan Pemeriksaan Geometri	113
30. Integrasi Parsing dengan Subbab Sebelumnya.....	113
31. Implikasi Keilmuan Parsing Output	114
32. Implikasi Pedagogis Parsing Output.....	115
33. Kesalahan Umum dalam Parsing Output.....	115
34. Format Laporan Hasil Parsing	116
35. Sintesis Subbab 11.4.....	116
36. Transisi ke Subbab 11.5.....	117
11.5 Python untuk Grafik Laporan.....	117
1. Pengertian Grafik Laporan dalam Kimia Komputasi	118
2. Latar Historis Visualisasi Data Komputasi	119
3. Prinsip Ilmiah dalam Membuat Grafik	119
4. Pustaka Python yang Digunakan.....	120
5. Struktur Folder untuk Grafik Laporan.....	120
6. Format Data untuk Grafik Energi	121
7. Kode Python untuk Grafik Energi.....	121
8. Analisis Grafik Energi	122
9. Format Data untuk Diagram Orbital.....	122
10. Kode Python untuk Diagram Orbital	123
11. Analisis Diagram Orbital	124

12. Format Data untuk Spektrum IR.....	124
13. Kode Python untuk Spektrum IR Garis Sederhana.....	125
14. Kode Python untuk Spektrum IR dengan Pelebaran Gaussian	125
15. Analisis Spektrum IR.....	126
16. Format Data untuk Diagram Energi Reaksi	126
17. Kode Python untuk Diagram Energi Reaksi.....	127
18. Analisis Diagram Energi Reaksi.....	127
19. Format Data untuk Grafik Perbandingan Metode.....	128
20. Kode Python untuk Grafik Perbandingan Metode.....	128
21. Analisis Grafik Perbandingan Metode.....	129
22. Kode Terpadu untuk Semua Grafik Laporan.....	129
23. Validasi Grafik Sebelum Masuk Laporan.....	132
24. Kesalahan Umum dalam Membuat Grafik Laporan	133
25. Grafik dan Narasi Laporan	133
26. Standar Nama File Grafik.....	134
27. Resolusi dan Format Gambar.....	134
28. Hubungan Grafik dengan Reprodusibilitas	135
29. Analisis Kritis terhadap Visualisasi	135
30. Aplikasi Grafik untuk Tujuh Mata Praktikum	136
31. Contoh Narasi Umum untuk Bagian Grafik dalam Laporan.....	136
32. Implikasi Keilmuan Grafik Laporan	137
33. Implikasi Pedagogis Grafik Laporan.....	137
34. Sintesis Subbab 11.5.....	137
35. Transisi ke Subbab 11.6.....	138
11.6 Template Laporan Praktikum.....	138
1. Pengertian Template Laporan Praktikum	139
2. Latar Historis Penulisan Laporan Ilmiah.....	140
3. Prinsip Umum Template Laporan	140
4. Struktur Umum Laporan Praktikum	141
5. Halaman Judul.....	142
6. Abstrak Singkat	143
7. Tujuan Praktikum	143
8. Dasar Teori.....	144

9. Software, Metode, dan Parameter	144
10. Struktur Input dan Data Awal	145
11. Prosedur Komputasi	146
12. Hasil Output	146
13. Validasi Data	147
14. Tabel dan Grafik.....	148
15. Analisis dan Pembahasan	148
16. Kesimpulan	149
17. Pertanyaan Diskusi	150
18. Lampiran	150
19. Template Lengkap Laporan Praktikum	151
20. Template Khusus Praktikum Optimasi Geometri	153
21. Template Khusus Praktikum Metode dan Basis Set	153
22. Template Khusus Praktikum HOMO-LUMO	154
23. Template Khusus Praktikum Momen Dipol.....	154
24. Template Khusus Praktikum Frekuensi Vibrasi dan IR	155
25. Template Khusus Praktikum Energi Reaksi dan Termodinamika.....	155
26. Template Khusus Praktikum Analisis Output dan Otomatisasi Python	156
27. Rubrik Penilaian Laporan Praktikum.....	157
28. Kesalahan Umum dalam Menulis Laporan Praktikum	157
29. Integrasi Template dengan Python	158
30. Contoh Narasi Metodologi dalam Laporan	158
31. Implikasi Keilmuan Template Laporan	159
32. Implikasi Pedagogis Template Laporan	159
33. Analisis Kritis terhadap Template	160
34. Sintesis Subbab 11.6.....	160
35. Transisi ke Penutup Bab 11	161
Penutup Bab 11.....	161
Daftar Pustaka Bab 11.....	164
BAB 12	167
PROYEK MINI, EVALUASI AKHIR, GLOSARIUM, DAN LAMPIRAN KODE PYTHON	167
Pengantar Bab 12.....	167
12.1 Proyek Mini Praktikum Kimia Kuantum	168

1. Pengertian Proyek Mini dalam Praktikum Kimia Kuantum	169
2. Tujuan Proyek Mini.....	169
3. Latar Historis Proyek Berbasis Komputasi dalam Pendidikan Kimia	170
4. Prinsip Pemilihan Proyek Mini	171
5. Pilihan Proyek 1: Analisis HOMO-LUMO Senyawa Aromatik Tersubstitusi.....	171
6. Pilihan Proyek 2: Perbandingan IR Komputasi dan IR Literatur.....	172
7. Pilihan Proyek 3: Optimasi Geometri Molekul Obat Sederhana	174
8. Pilihan Proyek 4: Studi Efek Pelarut terhadap Kestabilan Ion.....	175
9. Pilihan Proyek 5: Analisis Konformer Etanol atau Butana	176
10. Pilihan Proyek 6: Simetri dan Polaritas Molekul Anorganik	177
11. Pilihan Proyek 7: Energi Ikatan Hidrogen Dimer Air	178
12. Pilihan Proyek 8: Energi Reaksi Sederhana Fase Gas dan Pelarut	179
13. Kriteria Kelayakan Proyek Mini.....	180
14. Alur Kerja Proyek Mini	180
15. Peran Python dalam Proyek Mini	181
16. Contoh Rancangan Proyek Mini: HOMO-LUMO Nitrobenzena dan Anilina	181
17. Contoh Rancangan Proyek Mini: Spektrum IR Formaldehida	182
18. Contoh Rancangan Proyek Mini: Efek Pelarut pada NH_4^+.....	183
19. Contoh Rancangan Proyek Mini: Dimer Air	184
20. Standar Minimal Output Proyek Mini	184
21. Standar Minimal Analisis Proyek Mini	185
22. Kesalahan Umum dalam Proyek Mini.....	186
23. Penilaian Proyek Mini	186
24. Implikasi Keilmuan Proyek Mini	187
25. Implikasi Pedagogis Proyek Mini.....	187
26. Sintesis Subbab 12.1.....	188
27. Transisi ke Subbab 12.2.....	188
12.2 Format Proposal Proyek Mini	189
1. Pengertian Proposal Proyek Mini	190
2. Latar Historis Proposal dalam Kerja Ilmiah	190
3. Fungsi Proposal Proyek Mini.....	191
4. Prinsip Umum Proposal yang Baik	191
5. Komponen 1: Judul.....	192

6. Cara Menyusun Judul yang Tepat.....	193
7. Komponen 2: Latar Belakang.....	193
8. Unsur Teori dalam Latar Belakang	194
9. Komponen 3: Rumusan Masalah	195
10. Ciri Rumusan Masalah yang Baik	195
11. Komponen 4: Molekul Target.....	196
12. Prinsip Pemilihan Molekul Target	197
13. Komponen 5: Metode Komputasi.....	198
14. Basis Set dalam Proposal.....	198
15. Jenis Job Komputasi.....	199
16. Model Pelarut dalam Proposal	199
17. Komponen 6: Software	200
18. Software Bukan Metode	201
19. Komponen 7: Kode Python Pendukung.....	201
20. Contoh Kode Python yang Direncanakan dalam Proposal.....	202
21. Komponen 8: Data yang Dikumpulkan	203
22. Validasi Data dalam Proposal.....	203
23. Komponen 9: Jadwal Kerja	204
24. Contoh Format Proposal Proyek Mini.....	204
25. Contoh Proposal Singkat Proyek HOMO-LUMO	205
26. Contoh Proposal Singkat Proyek IR	207
27. Contoh Proposal Singkat Proyek Dimer Air	208
28. Analisis Kritis terhadap Proposal Proyek Mini	209
29. Kesalahan Umum dalam Menyusun Proposal	209
30. Rubrik Penilaian Proposal Proyek Mini.....	210
31. Implikasi Keilmuan Proposal Proyek Mini	210
32. Implikasi Pedagogis Proposal Proyek Mini.....	211
33. Hubungan Proposal dengan Laporan Akhir	211
34. Template Ringkas Proposal Proyek Mini Siap Pakai.....	212
35. Sintesis Subbab 12.2.....	213
36. Transisi ke Subbab 12.3.....	214
12.3 Format Laporan Proyek Mini	214
1. Pengertian Laporan Proyek Mini.....	215

2. Latar Historis Penulisan Laporan Komputasi	216
3. Fungsi Laporan Proyek Mini.....	216
4. Prinsip Umum Laporan Proyek Mini	217
5. Struktur Umum Laporan Proyek Mini.....	218
6. Bagian Judul dan Identitas	218
7. Bagian Abstrak.....	219
8. Kesalahan Umum dalam Menulis Abstrak.....	220
9. Bagian Pendahuluan	220
10. Rumusan Masalah dan Tujuan dalam Pendahuluan	221
11. Bagian Dasar Teori.....	221
12. Cara Menulis Dasar Teori yang Baik.....	223
13. Bagian Metode Komputasi.....	224
14. Contoh Penulisan Metode Komputasi.....	225
15. Validasi dalam Bagian Metode	225
16. Bagian Kode Python	226
17. Prinsip Penulisan Kode Python dalam Laporan.....	226
18. Bagian Hasil dan Pembahasan.....	227
19. Tabel Hasil Utama	227
20. Tabel Validasi Output	228
21. Grafik dalam Hasil dan Pembahasan	229
22. Cara Menulis Pembahasan yang Argumentatif	229
23. Perbandingan dengan Literatur	230
24. Keterbatasan Hasil.....	231
25. Bagian Kesimpulan	231
26. Bagian Daftar Pustaka dalam Laporan Proyek.....	232
27. Cara Menggunakan Referensi dalam Laporan.....	233
28. Bagian Lampiran Input-Output.....	233
29. Cara Memilih Potongan Output untuk Lampiran	234
30. Format Lengkap Laporan Proyek Mini Siap Pakai	234
31. Contoh Alur Laporan Proyek HOMO-LUMO	236
32. Contoh Alur Laporan Proyek IR	237
33. Contoh Alur Laporan Proyek Dimer Air	238
34. Kesalahan Umum dalam Menulis Laporan Proyek Mini	238

35. Rubrik Penilaian Laporan Proyek Mini.....	239
36. Implikasi Keilmuan Laporan Proyek Mini	239
37. Implikasi Pedagogis Laporan Proyek Mini	240
38. Analisis Kritis terhadap Format Laporan	240
39. Sintesis Subbab 12.3.....	241
40. Transisi ke Subbab 12.4.....	241
12.4 Evaluasi Akhir Praktikum	241
1. Pengertian Evaluasi Akhir Praktikum	242
2. Latar Historis Evaluasi dalam Praktikum Komputasi.....	243
3. Prinsip Evaluasi Akhir yang Baik	244
4. Komponen 1: Ujian Teori Singkat	244
5. Contoh Bentuk Soal Ujian Teori	245
6. Rubrik Ujian Teori Singkat	246
7. Komponen 2: Ujian Membaca Output.....	246
8. Contoh Soal Membaca Output	247
9. Rubrik Ujian Membaca Output	248
10. Komponen 3: Ujian Menjalankan Kode Python.....	248
11. Contoh Tugas Python: Energi Relatif.....	249
12. Contoh Tugas Python: Spektrum IR.....	249
13. Rubrik Ujian Python	250
14. Komponen 4: Ujian Interpretasi Grafik.....	250
15. Contoh Soal Interpretasi Grafik.....	251
16. Rubrik Interpretasi Grafik	252
17. Komponen 5: Presentasi Proyek Mini.....	252
18. Kriteria Presentasi yang Baik.....	252
19. Rubrik Presentasi Proyek Mini	253
20. Komponen 6: Laporan Akhir	254
21. Rubrik Laporan Akhir	254
22. Rancangan Bobot Evaluasi Akhir Praktikum.....	255
23. Evaluasi Formatif dan Sumatif.....	255
24. Analisis Kritis terhadap Evaluasi Akhir.....	256
25. Evaluasi Kejujuran Akademik	257
26. Evaluasi Reprodusibilitas.....	257

27. Evaluasi Keterampilan Diagnostik.....	258
28. Evaluasi Kemampuan Menarik Kesimpulan	259
29. Contoh Paket Evaluasi Akhir	259
30. Contoh Instruksi Evaluasi untuk Mahasiswa.....	260
31. Implikasi Keilmuan Evaluasi Akhir.....	260
32. Implikasi Pedagogis Evaluasi Akhir	260
33. Hubungan Evaluasi Akhir dengan Proyek Mini.....	261
34. Sintesis Subbab 12.4.....	261
35. Transisi ke Subbab 12.5.....	262
12.5 Glosarium Kimia Kuantum dan Komputasi	262
1. Fungsi Glosarium dalam Praktikum Kimia Kuantum.....	263
2. Latar Historis Istilah Kimia Kuantum dan Komputasi	264
3. Prinsip Membaca Glosarium	264
Glosarium Utama.....	265
1. Ab initio.....	265
2. Basis Set	266
3. ChemCompute.....	267
4. DFT.....	268
5. Energi Total.....	269
6. Frekuensi Imajiner	270
7. Fungsi Gelombang	271
8. Hartree-Fock	272
9. HOMO	273
10. LUMO	274
11. Momen Dipol	274
12. Optimasi Geometri.....	275
13. Orbital Molekul.....	276
14. SCF	277
15. Simetri Molekul.....	278
16. Zero-Point Energy.....	279
Ringkasan Hubungan Antaristilah.....	279
Analisis Kritis terhadap Penggunaan Istilah.....	280
Aplikasi Glosarium dalam Laporan Praktikum	281

Implikasi Keilmuan Glosarium	281
Implikasi Pedagogis Glosarium	282
Sintesis Subbab 12.5.....	282
Transisi ke Subbab 12.6	283
12.6 Lampiran Kode Python	283
1. Pengertian Lampiran Kode Python	284
2. Latar Historis Penggunaan Python dalam Kimia Komputasi	284
3. Prinsip Umum Penulisan Kode Praktikum	285
4. Struktur Folder yang Dianjurkan	286
5. Kebutuhan Pustaka Python	287
Lampiran Kode 1	287
praktikum_01_partikel_dalam_kotak.py	287
Lampiran Kode 2	289
praktikum_02_optimasi_geometri.py	289
Lampiran Kode 3	291
praktikum_03_hartree_fock_basis_set.py	291
Lampiran Kode 4	293
praktikum_04_dft_homo_lumo.py	293
Lampiran Kode 5	295
praktikum_05_simetri_momen_dipol.py	295
Lampiran Kode 6	297
praktikum_06_spektrum_ir.py	297
Lampiran Kode 7	299
praktikum_07_energi_reaksi_termodinamika.py	299
Lampiran Kode 8	302
parser_output_komputasi.py	302
Lampiran Kode 9	306
template_laporan_otomatis.py	306
6. Cara Menjalankan Kode	310
7. Cara Menggunakan Kode dalam Laporan	310
8. Analisis Kritis terhadap Lampiran Kode	311
9. Kesalahan Umum dalam Penggunaan Kode Python	312
10. Implikasi Keilmuan Lampiran Kode	312

11. Implikasi Pedagogis Lampiran Kode	313
12. Sintesis Subbab 12.6	313
13. Transisi ke Penutup Bab 12	314
Penutup Bab 12	314
Daftar Pustaka Bab 12	317
LAMPIRAN 1	321
CATATAN PENTING, STANDAR TEKNIS, DAN CHECKLIST PRAKTIKUM KIMIA KUANTUM	321
A. Tujuan Lampiran	321
B. Standar Organisasi Folder Praktikum	321
C. Standar Penamaan File	322
D. Checklist Sebelum Menjalankan Perhitungan	323
E. Checklist Setelah Perhitungan Selesai	323
F. Standar Satuan Penting	324
G. Standar Minimal Validasi Struktur Minimum	325
H. Format Metadata Perhitungan	325
I. Panduan Troubleshooting Singkat	326
1. SCF Tidak Konvergen	326
2. Optimasi Geometri Gagal	327
3. Frekuensi Imajiner Muncul	327
4. Grafik Tidak Sesuai Data	327
J. Standar Kelengkapan File yang Dikumpulkan	328
K. Prinsip Kejujuran Akademik	328
L. Prinsip Reprodusibilitas	329
M. Template README Praktikum	329
N. Checklist Akhir Sebelum Pengumpulan	331
O. Ringkasan Lampiran 1	331
LAMPIRAN 2 PELENGKAP	333
Tahap 1 — Praktikum Software	334
Tahap 2 — Praktikum Python	335

BAGIAN 3

MEMBUAT LAPORAN & PENUTUP

BAB 11

ANALISIS OUTPUT, VALIDASI DATA, DAN OTOMATISASI LAPORAN DENGAN PYTHON

Pengantar Bab 11

Bab 11 membahas tahap yang sangat penting dalam kimia komputasi, yaitu **membaca output, memvalidasi hasil, menemukan kesalahan umum, dan mengotomatisasi laporan dengan Python**. Setelah mahasiswa menjalankan berbagai praktikum pada bab-bab sebelumnya, mulai dari optimasi geometri, analisis orbital, momen dipol, frekuensi vibrasi, spektrum IR, hingga energi reaksi dan efek pelarut, tahap berikutnya adalah memastikan bahwa hasil yang diperoleh benar, dapat dipercaya, dan dapat ditulis menjadi laporan ilmiah yang rapi.

Dalam kimia komputasi, menjalankan software hanyalah satu bagian dari pekerjaan. Bagian yang tidak kalah penting adalah membaca output dengan teliti. Output komputasi dapat memuat ratusan hingga ribuan baris informasi: metode, basis set, muatan, multiplicity, energi total, geometri akhir, momen dipol, frekuensi, orbital, status SCF, status optimasi, hingga pesan kesalahan. Jika mahasiswa tidak mampu membaca output, maka ia hanya menjadi pengguna software pasif. Sebaliknya, jika mahasiswa mampu membaca output secara kritis, ia dapat menilai apakah perhitungan berhasil, apakah data valid, apakah struktur masuk akal, dan apakah hasil dapat digunakan dalam laporan.

Bab ini juga menekankan pentingnya **validasi data**. Tidak semua output yang selesai berjalan layak digunakan. Perhitungan dapat berhenti tidak normal. Optimasi dapat gagal konvergen. Struktur dapat memiliki frekuensi imajiner. Muatan dapat salah. Multiplicity dapat tidak sesuai. Basis set dapat tidak memadai. Energi dapat dibaca dari bagian output yang salah. Kesalahan-kesalahan tersebut dapat menghasilkan laporan yang tampak ilmiah tetapi sesungguhnya tidak sah.

Karena itu, Bab 11 disusun untuk melatih mahasiswa melakukan tiga hal utama:

```
membaca output  
→ memvalidasi hasil  
→ mengotomatisasi laporan
```

Python digunakan dalam bab ini sebagai alat bantu untuk membaca file log, mencari pola teks penting, mengekstrak energi total, momen dipol, frekuensi, HOMO-LUMO, dan menyusun ringkasan otomatis ke dalam tabel CSV. Dengan demikian, mahasiswa tidak hanya membaca output secara manual, tetapi mulai belajar membangun alur kerja yang lebih efisien, konsisten, dan reproduibel.

11.1 Membaca File Output Komputasi

Membaca file output komputasi adalah keterampilan dasar yang wajib dimiliki oleh mahasiswa kimia komputasi. File output adalah dokumen hasil perhitungan yang dihasilkan oleh software seperti ChemCompute, GAMESS, ORCA, Gaussian, Psi4, PySCF, atau paket komputasi lain. Di dalam file output terdapat informasi penting yang menentukan apakah suatu perhitungan dapat digunakan untuk analisis ilmiah.

Dalam konteks praktikum, output tidak boleh dibaca secara sepintas. Mahasiswa harus mampu mencari dan mencatat informasi berikut:

1. **Metode.**
2. **Basis set.**
3. **Charge.**
4. **Multiplicity.**
5. **Energi total.**
6. **Geometri akhir.**
7. **Momen dipol.**
8. **Frekuensi.**
9. **HOMO-LUMO.**
10. **Status konvergensi.**

Sepuluh informasi tersebut membentuk identitas dan validitas hasil komputasi. Tanpa informasi metode dan basis set, hasil tidak dapat direproduksi. Tanpa charge dan multiplicity, jumlah elektron dan keadaan spin tidak dapat diverifikasi. Tanpa energi total, tidak ada dasar untuk perbandingan energi. Tanpa geometri akhir, struktur hasil optimasi tidak dapat dinilai. Tanpa momen dipol, polaritas molekul tidak dapat dianalisis. Tanpa frekuensi, struktur minimum tidak dapat divalidasi dan spektrum IR tidak dapat dibahas. Tanpa HOMO-LUMO, sifat elektronik dan reaktivitas sulit ditafsirkan. Tanpa status konvergensi, mahasiswa tidak tahu apakah perhitungan benar-benar selesai secara layak.

Dengan demikian, membaca output adalah pekerjaan ilmiah, bukan pekerjaan administratif.

1. Pengertian File Output Komputasi

File output komputasi adalah file teks yang berisi rekaman lengkap atau ringkas dari proses perhitungan kimia komputasi. File ini biasanya dihasilkan setelah software menjalankan perintah seperti optimasi geometri, single point energy, perhitungan frekuensi, analisis orbital, atau perhitungan termokimia.

File output dapat memiliki ekstensi berbeda, misalnya:

```
.log
.out
```

```
.dat
.txt
```

Dalam beberapa software, file output utama disebut `log file`, sementara pada software lain disebut `output file`. Walaupun nama ekstensi berbeda, fungsinya sama: menyimpan informasi tentang input, proses perhitungan, hasil akhir, dan pesan diagnostik.

Secara umum, file output memuat:

```
identitas perhitungan
→ metode dan basis set
→ struktur awal
→ proses iterasi
→ energi
→ status konvergensi
→ struktur akhir
→ sifat molekul
→ pesan akhir
```

Mahasiswa harus memahami bahwa file output bukan hanya tempat mencari angka energi. File output adalah dokumen ilmiah yang menjelaskan bagaimana hasil diperoleh.

2. Latar Historis Pembacaan Output dalam Kimia Komputasi

Pada masa awal kimia komputasi, sebagian besar pekerjaan dilakukan melalui terminal dan file teks. Pengguna menulis input secara manual, menjalankan program, kemudian membaca output baris demi baris. Tidak ada antarmuka visual yang memudahkan semua informasi. Oleh karena itu, kemampuan membaca output menjadi keterampilan inti bagi ahli kimia komputasi.

Seiring berkembangnya software seperti Gaussian, GAMESS, ORCA, Psi4, PySCF, serta antarmuka visual seperti GaussView, Avogadro, dan platform web seperti ChemCompute, proses komputasi menjadi lebih mudah diakses. Namun, kemudahan antarmuka tidak menghilangkan kebutuhan membaca output. Bahkan dalam antarmuka modern, informasi paling lengkap tetap berada dalam file output.

Literatur kimia komputasi seperti Jensen, Cramer, Leach, Lewars, Foresman-Frisch, serta dokumentasi resmi ORCA, Gaussian, GAMESS, Psi4, dan PySCF selalu menekankan pentingnya memahami output. Alasan utamanya sederhana: software dapat menghasilkan angka, tetapi pengguna harus menentukan apakah angka tersebut sah.

Dengan kata lain:

```
software menghitung
→ output mencatat
→ pengguna menilai
```


Bab ini menempatkan mahasiswa pada posisi pengguna ilmiah yang aktif menilai hasil, bukan sekadar menerima hasil.

3. Struktur Umum File Output

Walaupun format output setiap software berbeda, struktur umumnya memiliki kesamaan. Mahasiswa dapat membaca output dengan mencari bagian-bagian berikut:

Bagian Output	Isi Utama
Header	Nama software, versi, tanggal, informasi sistem
Input summary	Metode, basis set, charge, multiplicity, koordinat awal
Iterasi SCF	Proses pencarian energi elektronik
Optimasi	Perubahan geometri dan kriteria konvergensi
Energi akhir	Energi total atau energi elektronik akhir
Geometri akhir	Koordinat atom setelah optimasi
Sifat molekul	Momen dipol, orbital, muatan parsial jika tersedia
Frekuensi	Frekuensi vibrasi, intensitas IR, ZPE, termokimia
Pesan status	Normal termination, convergence achieved, error message

Struktur ini membantu mahasiswa mencari informasi tanpa harus membaca seluruh file dari awal sampai akhir secara acak. Pembacaan output harus dilakukan dengan strategi.

Strategi umum:

```
cari identitas metode
→ cari charge dan multiplicity
→ cari status SCF
→ cari status optimasi
→ cari energi akhir
→ cari geometri akhir
→ cari sifat molekul
→ cari frekuensi
→ cari pesan akhir
```

4. Membaca Metode Perhitungan

Metode perhitungan menunjukkan pendekatan teori yang digunakan untuk menghitung energi dan sifat molekul. Contoh metode yang sering digunakan dalam praktikum adalah HF, DFT, B3LYP, semiempiris, MP2, atau metode lain sesuai software.

Dalam output, metode dapat muncul di bagian awal sebagai bagian dari input atau route section. Contoh konseptual:

```
# B3LYP/6-31G(d) Opt Freq
```

atau:

```
! B3LYP 6-31G(d) Opt Freq
```

atau dalam bentuk ringkasan:

```
Method: B3LYP
```

Metode sangat penting karena hasil energi, geometri, frekuensi, dan orbital bergantung pada metode. Perhitungan dengan HF dapat berbeda dari DFT. Perhitungan dengan B3LYP dapat berbeda dari PBE, M06, atau metode lain. Oleh karena itu, laporan praktikum harus mencantumkan metode secara eksplisit.

Pernyataan laporan yang baik:

Perhitungan dilakukan menggunakan metode B3LYP dengan basis set 6-31G(d). Struktur dioptimasi terlebih dahulu sebelum dilakukan analisis energi dan frekuensi.

Pernyataan yang kurang tepat:

Perhitungan dilakukan menggunakan software Gaussian.

Pernyataan tersebut belum cukup karena Gaussian adalah software, bukan metode. Software hanya alat; metode adalah pendekatan teorinya.

5. Membaca Basis Set

Basis set adalah kumpulan fungsi matematika yang digunakan untuk merepresentasikan orbital atom dalam perhitungan kimia kuantum. Basis set menentukan keluwesan deskripsi elektron. Basis set kecil lebih cepat tetapi kurang akurat. Basis set besar lebih akurat tetapi membutuhkan biaya komputasi lebih tinggi.

Contoh basis set:

```
STO-3G
3-21G
6-31G
6-31G(d)
6-31G*
6-31G(d,p)
def2-SVP
def2-TZVP
cc-pVDZ
```

Dalam output, basis set biasanya muncul di bagian awal, bersama metode. Misalnya:

B3LYP/6-31G(d)

Artinya metode yang digunakan adalah B3LYP dan basis set yang digunakan adalah 6-31G(d).

Basis set harus dicatat karena hasil tidak dapat direproduksi tanpa informasi ini. Dua perhitungan dengan metode sama tetapi basis set berbeda dapat menghasilkan energi dan frekuensi yang berbeda.

Contoh analisis:

Basis set 6-31G(d) digunakan karena memiliki fungsi polarisasi yang lebih baik dibanding basis set minimal. Fungsi polarisasi penting untuk menggambarkan perubahan bentuk distribusi elektron pada ikatan dan interaksi antarmolekul.

Dalam laporan praktikum, metode dan basis set biasanya ditulis sebagai satu level teori:

B3LYP/6-31G(d)

6. Membaca Charge

Charge atau muatan total sistem menunjukkan jumlah muatan listrik keseluruhan molekul atau ion. Charge sangat penting karena menentukan jumlah elektron dalam perhitungan. Jika charge salah, maka sistem yang dihitung bukan sistem kimia yang dimaksud.

Contoh:

Spesies	Charge yang Benar
H ₂ O	0
NH ₃	0
NH ₄ ⁺	+1
Cl ⁻	-1
CH ₃ COOH	0

Dalam input dan output, charge biasanya muncul bersama multiplicity. Contoh format umum:

0 1

Angka pertama adalah charge, angka kedua adalah multiplicity. Jadi 0 1 berarti sistem netral singlet.

Contoh lain:

1 1

Artinya sistem bermuatan +1 dan singlet. Ini dapat digunakan untuk NH₄⁺.

Kesalahan charge merupakan salah satu kesalahan paling serius. Misalnya, jika NH_4^+ dihitung dengan charge 0, maka jumlah elektronnya salah. Energi, geometri, dan sifat molekul yang dihasilkan tidak mewakili NH_4^+ .

Pernyataan laporan yang baik:

NH_4^+ dihitung dengan charge +1 dan multiplicity 1, sedangkan NH_3 dihitung dengan charge 0 dan multiplicity 1.

7. Membaca Multiplicity

Multiplicity menunjukkan keadaan spin total sistem. Dalam kimia komputasi, multiplicity berhubungan dengan jumlah elektron tidak berpasangan. Rumus umum multiplicity adalah:

$$\text{Multiplicity} = 2S + 1$$

dengan s adalah total spin.

Untuk sebagian besar molekul organik netral berpasangan elektron, multiplicity adalah 1 atau singlet. Jika terdapat satu elektron tidak berpasangan, multiplicity biasanya 2 atau doublet. Jika terdapat dua elektron tidak berpasangan dengan spin sejajar, multiplicity dapat 3 atau triplet.

Contoh:

Sistem	Multiplicity Umum
H_2O	1
CO_2	1
NH_3	1
CH_4	1
Radikal metil $\text{CH}_3\cdot$	2
O_2 keadaan dasar	3

Dalam praktikum buku ini, sebagian besar sistem adalah singlet. Namun, mahasiswa tetap harus mencatat multiplicity karena kesalahan multiplicity dapat mengubah energi dan struktur.

Contoh kesalahan:

O_2 dihitung sebagai singlet padahal keadaan dasar O_2 adalah triplet.

Kesalahan seperti ini menghasilkan perhitungan yang tidak sesuai keadaan dasar molekul.

8. Membaca Energi Total

Energi total adalah salah satu data paling penting dalam output. Energi ini digunakan untuk membandingkan kestabilan molekul, menghitung energi reaksi, menghitung energi isomerisasi, atau membandingkan konformer.

Dalam output, energi total dapat muncul dengan berbagai nama tergantung software:

```
SCF Done
Total Energy
Final Energy
Electronic Energy
E(RB3LYP)
FINAL SINGLE POINT ENERGY
Total SCF energy
```

Contoh konseptual:

```
FINAL SINGLE POINT ENERGY      -76.3456789012
```

atau:

```
SCF Done: E(RB3LYP) = -76.3456789012 A.U.
```

Mahasiswa harus memastikan energi yang diambil adalah energi akhir, bukan energi dari iterasi awal. Pada output panjang, energi dapat muncul berkali-kali. Energi yang digunakan biasanya energi setelah SCF konvergen dan setelah struktur akhir diperoleh.

Dalam optimasi geometri, energi muncul pada setiap langkah optimasi. Energi yang digunakan untuk analisis adalah energi pada geometri akhir yang konvergen.

Pernyataan laporan:

Energi total diambil dari bagian akhir output setelah optimasi geometri dinyatakan konvergen.

9. Energi Total, Energi Elektronik, dan Energi Termokimia

Mahasiswa harus membedakan energi elektronik dari energi termokimia. Energi elektronik biasanya berasal dari perhitungan SCF atau DFT. Energi termokimia seperti entalpi dan Gibbs biasanya diperoleh dari perhitungan frekuensi.

Perbedaan:

Jenis Energi	Makna
Energi elektronik	Energi struktur elektronik pada geometri tertentu
Energi + ZPE	Energi elektronik ditambah zero-point energy

Entalpi	Energi elektronik dengan koreksi termal entalpi
Gibbs	Entalpi dikurangi kontribusi entropi

Kesalahan umum adalah mengambil energi elektronik lalu menyebutnya sebagai Gibbs. Ini harus dihindari. Jika laporan membahas ΔG , maka nilai yang digunakan harus berasal dari bagian output yang memuat energi bebas Gibbs atau koreksi Gibbs.

Contoh catatan:

Energi elektronik digunakan untuk menghitung ΔE , sedangkan nilai Gibbs dari output frekuensi digunakan untuk menghitung ΔG .

10. Membaca Geometri Akhir

Geometri akhir adalah susunan koordinat atom setelah optimasi selesai. Geometri akhir digunakan untuk melihat apakah struktur yang dihitung masuk akal secara kimia. Dalam output, geometri akhir dapat muncul dalam bentuk koordinat Cartesian, koordinat internal, atau tabel atom dengan posisi x, y, z.

Contoh format konseptual:

C	0.000000	0.000000	0.000000
O	1.220000	0.000000	0.000000
H	-0.600000	0.930000	0.000000
H	-0.600000	-0.930000	0.000000

Geometri akhir harus diperiksa karena optimasi dapat menghasilkan struktur yang tidak diharapkan. Misalnya, dimer dapat terpisah terlalu jauh, konformer dapat berubah menjadi konformer lain, atau ikatan dapat memanjang secara tidak realistis.

Hal yang perlu diperiksa:

1. Apakah ikatan utama masih terbentuk?
2. Apakah panjang ikatan masuk akal?
3. Apakah sudut ikatan wajar?
4. Apakah struktur akhir sesuai sistem yang dimaksud?
5. Apakah dimer masih membentuk interaksi?
6. Apakah konformer akhir masih berbeda dari konformer lain?
7. Apakah tidak terjadi fragmentasi tidak diinginkan?

Contoh analisis:

Geometri akhir dimer air menunjukkan adanya jarak $O \cdots H$ yang sesuai dengan interaksi ikatan hidrogen. Hal ini menunjukkan bahwa struktur dimer tetap terbentuk setelah optimasi.

Geometri akhir sangat penting sebelum energi digunakan. Energi dari struktur yang salah tidak dapat digunakan untuk kesimpulan kimia yang benar.

11. Membaca Momen Dipol

Momen dipol menunjukkan polaritas molekul. Molekul dengan distribusi muatan tidak simetris memiliki momen dipol lebih besar. Dalam output, momen dipol biasanya ditampilkan dalam satuan Debye.

Contoh informasi output:

```
Dipole moment (Debye):
X = ...
Y = ...
Z = ...
Tot = ...
```

Nilai total momen dipol adalah nilai yang biasanya dicatat dalam laporan.

Momen dipol penting dalam beberapa konteks:

1. Menjelaskan polaritas molekul.
2. Menghubungkan struktur dengan interaksi pelarut.
3. Menjelaskan intensitas IR.
4. Membandingkan isomer atau konformer.
5. Menilai sensitivitas terhadap pelarut.

Contoh:

CO₂ memiliki momen dipol total mendekati nol karena strukturnya linear dan simetris.

H₂O memiliki momen dipol tidak nol karena bentuknya bengkok.

Dalam praktikum, mahasiswa dapat menggunakan momen dipol untuk menjelaskan mengapa molekul tertentu lebih dipengaruhi pelarut polar dibanding molekul lain.

Contoh analisis:

Konformer etanol dengan momen dipol lebih besar dapat mengalami stabilisasi lebih kuat dalam pelarut polar dibanding konformer yang momen dipolnya lebih kecil.

12. Membaca Frekuensi

Frekuensi vibrasi diperoleh dari perhitungan frekuensi. Frekuensi digunakan untuk beberapa tujuan:

1. Memvalidasi struktur minimum.
2. Menyimulasikan spektrum IR.
3. Mengidentifikasi gugus fungsi.
4. Menghitung ZPE.
5. Menghitung koreksi entalpi dan Gibbs.
6. Mendeteksi frekuensi imajiner.

Dalam output, frekuensi biasanya muncul dalam satuan cm^{-1} .

Contoh format konseptual:

```
Frequencies -- 3650.2 1595.4 1200.8
IR Inten    -- 45.1 12.8 8.3
```

Untuk struktur minimum, semua frekuensi harus bernilai positif. Jika terdapat frekuensi imajiner, software biasanya menampilkannya sebagai angka negatif atau dengan tanda khusus. Dalam banyak output praktikum, frekuensi negatif diperlakukan sebagai indikator frekuensi imajiner.

Contoh:

```
Frequencies -- -250.4 120.5 340.1
```

Frekuensi negatif pertama menunjukkan adanya mode imajiner. Untuk struktur minimum, ini menandakan struktur belum valid dan perlu diperiksa ulang.

13. Frekuensi Imajiner dan Maknanya

Frekuensi imajiner memiliki makna penting. Untuk struktur minimum, tidak boleh ada frekuensi imajiner. Jika ada, struktur mungkin belum mencapai minimum atau berada pada titik pelana di permukaan energi potensial.

Interpretasi umum:

```
tidak ada frekuensi imajiner
→ struktur dapat dianggap minimum lokal
```

```
satu frekuensi imajiner
→ dapat menunjukkan keadaan transisi jika mode sesuai koordinat reaksi
```

```
lebih dari satu frekuensi imajiner
→ struktur belum valid untuk minimum atau keadaan transisi sederhana
```

Dalam praktikum bab-bab sebelumnya, terutama optimasi dan frekuensi IR, mahasiswa biasanya bekerja dengan struktur minimum. Karena itu, frekuensi imajiner harus dianggap sebagai tanda perlunya optimasi ulang, kecuali jika praktikum memang membahas keadaan transisi.

Contoh laporan:

Perhitungan frekuensi tidak menunjukkan frekuensi imajiner, sehingga struktur hasil optimasi dapat dianggap sebagai minimum lokal pada tingkat teori yang digunakan.

Jika ada frekuensi imajiner:

Output menunjukkan satu frekuensi imajiner. Karena sistem yang dianalisis dimaksudkan sebagai struktur minimum, hasil ini belum digunakan untuk analisis energi dan perlu dioptimasi ulang.

14. Membaca HOMO-LUMO

HOMO dan LUMO merupakan orbital molekul penting dalam analisis sifat elektronik dan reaktivitas.

HOMO adalah:

Highest Occupied Molecular Orbital

atau orbital molekul terisi tertinggi.

LUMO adalah:

Lowest Unoccupied Molecular Orbital

atau orbital molekul kosong terendah.

Selisih energi HOMO-LUMO sering disebut celah energi atau energy gap:

$$\Delta E_{\text{gap}} = E_{\text{LUMO}} - E_{\text{HOMO}}$$

Celah HOMO-LUMO digunakan secara kualitatif untuk menilai kecenderungan reaktivitas elektronik. Molekul dengan celah kecil sering dianggap lebih mudah mengalami eksitasi elektronik atau lebih reaktif secara elektronik, meskipun interpretasi ini harus dilakukan hati-hati.

Dalam output, orbital dapat muncul dalam bagian eigenvalues atau molecular orbital energies. Contoh konseptual:

Occupied orbital energies:

...

Virtual orbital energies:

...

Mahasiswa harus mencari orbital terisi terakhir sebagai HOMO dan orbital kosong pertama sebagai LUMO.

Contoh analisis:

Nilai celah HOMO-LUMO yang lebih kecil menunjukkan bahwa transisi elektronik dari orbital terisi ke orbital kosong membutuhkan energi lebih rendah. Namun, interpretasi reaktivitas tidak boleh hanya didasarkan pada celah HOMO-LUMO tanpa mempertimbangkan struktur, muatan, dan konteks kimia.

15. Membaca Status Konvergensi SCF

SCF atau Self-Consistent Field adalah prosedur iteratif untuk memperoleh energi elektronik yang konsisten. Jika SCF tidak konvergen, energi elektronik tidak dapat dipercaya.

Dalam output, status SCF dapat muncul sebagai:

```
SCF converged
SCF Done
SCF convergence achieved
SCF NOT CONVERGED
```

Jika SCF berhasil, energi akhir dapat digunakan. Jika SCF gagal, perhitungan harus diperbaiki. Perbaikan dapat berupa menggunakan tebakan awal yang lebih baik, mengubah parameter SCF, memperbaiki geometri awal, menggunakan metode yang lebih stabil, atau menyesuaikan basis set.

Pernyataan laporan:

SCF dinyatakan konvergen, sehingga energi elektronik akhir digunakan dalam analisis.

Jika gagal:

SCF tidak konvergen, sehingga energi elektronik tidak digunakan sebagai dasar analisis.

Mahasiswa harus memahami bahwa output yang muncul bukan berarti hasil valid. Status SCF harus diperiksa.

16. Membaca Status Konvergensi Optimasi Geometri

Selain SCF, optimasi geometri juga harus konvergen. Optimasi geometri mencari struktur dengan energi minimum. Jika optimasi belum konvergen, struktur akhir belum layak digunakan sebagai geometri minimum.

Indikator konvergensi optimasi dapat berupa:

```
Optimization completed
Stationary point found
Geometry optimization converged
Convergence criteria satisfied
```

Jika output menunjukkan optimasi gagal atau jumlah langkah maksimum tercapai, struktur belum valid.

Contoh pesan bermasalah:

```
Maximum number of optimization cycles exceeded
Optimization did not converge
```

Mahasiswa harus membedakan antara SCF konvergen dan optimasi konvergen. SCF dapat konvergen pada setiap langkah, tetapi optimasi geometri tetap belum selesai. Untuk analisis struktur, keduanya harus diperhatikan.

Pernyataan laporan yang baik:

```
Optimasi geometri dinyatakan konvergen dan tidak ditemukan frekuensi imajiner,
sehingga struktur akhir digunakan sebagai dasar analisis energi dan sifat
molekul.
```

17. Membaca Pesan Akhir Output

Banyak software menampilkan pesan akhir yang menunjukkan apakah job selesai normal atau tidak. Pesan ini harus diperiksa sebelum mengambil data.

Contoh pesan selesai normal:

```
Normal termination
ORCA TERMINATED NORMALLY
EXECUTION OF GAMESS TERMINATED NORMALLY
```

Contoh pesan bermasalah:

```
Error termination
SCF failed to converge
Optimization failed
File not found
```

Memory exceeded

Pesan akhir sangat penting karena file output yang tidak selesai normal dapat tetap memuat sebagian data. Jika mahasiswa mengambil angka dari output yang gagal, laporan dapat menjadi salah.

Prinsipnya:

job selesai normal
→ lanjut validasi data

job tidak selesai normal
→ jangan langsung gunakan hasil

Pesan akhir tidak cukup menjadi satu-satunya validasi, tetapi merupakan langkah awal yang wajib diperiksa.

18. Strategi Praktis Mencari Informasi dalam Output

Karena output dapat sangat panjang, mahasiswa perlu strategi mencari informasi penting. Jika menggunakan editor teks, fitur pencarian dapat digunakan dengan kata kunci tertentu.

Contoh kata kunci pencarian:

Informasi	Kata Kunci yang Dicari
Metode	B3LYP, HF, Method, Functional
Basis set	Basis, 6-31G, def2, STO-3G
Charge	Charge, Total Charge, 0 1, 1 1
Multiplicity	Multiplicity, Spin, 0 1, 1 1
Energi total	SCF Done, Total Energy, FINAL SINGLE POINT ENERGY
Geometri akhir	Final geometry, Standard orientation, CARTESIAN COORDINATES
Momen dipol	Dipole moment, Total Dipole Moment
Frekuensi	Frequencies, VIBRATIONAL FREQUENCIES
HOMO-LUMO	Orbital energies, HOMO, LUMO, MO energies
Konvergensi	converged, Normal termination, terminated normally

Kata kunci berbeda antarsoftware. Oleh karena itu, mahasiswa perlu membiasakan diri dengan format output software yang digunakan.

19. Contoh Ringkasan Informasi Output

Setelah membaca output, mahasiswa sebaiknya membuat ringkasan seperti berikut:

Informasi	Hasil
Nama file output	h2o_b3lyp_631g_optfreq.log
Software	Gaussian / ORCA / GAMESS / lainnya
Metode	B3LYP
Basis set	6-31G(d)
Charge	0
Multiplicity	1
Energi total	... Hartree
Geometri akhir	Tersedia dan masuk akal
Momen dipol	... Debye
Frekuensi imajiner	Tidak ada
HOMO	... Hartree/eV
LUMO	... Hartree/eV
Status SCF	Konvergen
Status optimasi	Konvergen
Status job	Selesai normal

Tabel ringkasan seperti ini sangat membantu dalam laporan praktikum. Tabel juga dapat menjadi dasar otomatisasi pada subbab 11.4.

20. Membaca Output sebagai Proses Validasi Awal

Membaca output bukan sekadar mengambil data, tetapi juga validasi awal. Sebelum mahasiswa menyatakan hasil sebagai benar, ia harus memeriksa apakah output memenuhi syarat minimal.

Syarat minimal:

```
metode jelas
basis set jelas
charge benar
multiplicity benar
SCF konvergen
optimasi konvergen
geometri akhir masuk akal
tidak ada frekuensi imajiner untuk minimum
energi akhir tersedia
job selesai normal
```

Jika salah satu syarat penting tidak terpenuhi, hasil harus diberi catatan atau perhitungan harus diulang.

Contoh:

Jika optimasi belum konvergen, energi akhir tidak boleh langsung digunakan untuk membandingkan kestabilan molekul.

atau:

Jika struktur minimum masih memiliki frekuensi imajiner, struktur perlu dioptimasi ulang sebelum data frekuensi dan termokimia digunakan.

21. Analisis Kritis: Output Bukan Kebenaran Otomatis

Mahasiswa harus memahami bahwa output software tidak otomatis benar. Software menjalankan perintah yang diberikan pengguna. Jika input salah, output juga dapat salah. Jika charge salah, software tetap menghitung sistem yang salah. Jika struktur awal buruk, optimasi dapat menuju struktur yang tidak diinginkan. Jika basis set tidak sesuai, hasil dapat kurang baik. Jika SCF gagal, energi tidak dapat dipercaya.

Prinsip penting:

output numerik
≠ otomatis valid

output numerik
+ validasi
+ interpretasi
= data ilmiah yang layak digunakan

Sikap kritis ini menjadi inti Bab 11. Mahasiswa harus belajar bahwa angka yang rapi dalam output tidak selalu berarti hasil benar. Angka harus diperiksa konteksnya.

Contoh analisis kritis:

Energi total yang diperoleh tidak digunakan secara langsung sebelum dipastikan bahwa SCF dan optimasi geometri telah konvergen. Selain itu, charge dan multiplicity diperiksa agar sistem yang dihitung sesuai dengan spesies kimia yang dimaksud.

22. Contoh Kasus: Membaca Output Air

Untuk molekul air, mahasiswa dapat membaca output dengan urutan berikut:

1. Pastikan metode dan basis set, misalnya B3LYP/6-31G(d).
2. Pastikan charge dan multiplicity adalah 0 1.
3. Cari energi total akhir.
4. Periksa geometri akhir.
5. Pastikan bentuk molekul bengkok, bukan linear.
6. Cari momen dipol.
7. Cari frekuensi vibrasi.
8. Pastikan tidak ada frekuensi imajiner.
9. Cari status konvergensi.

10. Pastikan job selesai normal.

Analisis singkat:

Output air menunjukkan struktur akhir berbentuk bengkok dan tidak memiliki frekuensi imajiner. Hal ini sesuai dengan struktur minimum H_2O . Momen dipol tidak nol karena distribusi muatan molekul air tidak simetris.

Contoh ini menunjukkan bagaimana data output dihubungkan dengan konsep kimia.

23. Contoh Kasus: Membaca Output CO_2

Untuk CO_2 , output dibaca dengan urutan yang sama. Namun, analisis kimianya berbeda.

Hal yang diperiksa:

1. Charge 0.
2. Multiplicity 1.
3. Struktur akhir linear.
4. Momen dipol mendekati nol.
5. Frekuensi vibrasi positif untuk struktur minimum.
6. Mode vibrasi sesuai dengan regangan dan tekukan CO_2 .

Analisis:

Output CO_2 menunjukkan struktur linear dengan momen dipol total mendekati nol. Hal ini sesuai dengan simetri molekul CO_2 yang menyebabkan dua dipol ikatan $\text{C}=\text{O}$ saling meniadakan.

Dari contoh ini mahasiswa belajar bahwa membaca output harus selalu dihubungkan dengan struktur dan teori.

24. Contoh Kasus: Membaca Output NH_4^+

Untuk NH_4^+ , mahasiswa harus sangat memperhatikan charge. Ion amonium harus dihitung dengan charge +1 dan multiplicity 1.

Hal yang diperiksa:

Charge = +1
 Multiplicity = 1
 Geometri tetrahedral
 SCF konvergen
 Optimasi konvergen

Tidak ada frekuensi imajiner

Jika NH_4^+ salah dihitung dengan charge 0, maka output tidak mewakili ion amonium. Ini merupakan kesalahan kimia yang serius.

Analisis:

Output NH_4^+ hanya valid jika sistem dihitung dengan charge +1 dan multiplicity 1. Jika charge salah, jumlah elektron tidak sesuai sehingga energi dan sifat molekul tidak dapat digunakan untuk analisis protonasi amonia.

25. Hubungan Membaca Output dengan Praktikum Sebelumnya

Subbab ini berhubungan langsung dengan semua praktikum sebelumnya.

Pada praktikum optimasi geometri, mahasiswa perlu membaca geometri akhir dan status konvergensi.

Pada praktikum orbital, mahasiswa perlu membaca HOMO, LUMO, dan celah energi.

Pada praktikum momen dipol, mahasiswa perlu membaca nilai dipol total dan komponennya.

Pada praktikum frekuensi IR, mahasiswa perlu membaca frekuensi vibrasi, intensitas IR, dan frekuensi imajiner.

Pada praktikum energi reaksi, mahasiswa perlu membaca energi total, entalpi, dan Gibbs.

Dengan demikian, membaca output adalah keterampilan lintas bab. Tanpa keterampilan ini, semua praktikum sebelumnya menjadi lemah karena mahasiswa tidak tahu apakah data yang digunakan benar.

26. Implikasi Metodologis Membaca Output

Membaca output memiliki implikasi metodologis yang besar.

Pertama, membaca output memastikan **reprodusibilitas**. Metode, basis set, charge, multiplicity, dan kondisi perhitungan harus tercatat agar hasil dapat diulang.

Kedua, membaca output memastikan **validitas**. SCF dan optimasi harus konvergen, frekuensi harus sesuai, dan struktur harus masuk akal.

Ketiga, membaca output memastikan **ketelitian interpretasi**. Energi elektronik tidak boleh disamakan dengan Gibbs. Frekuensi imajiner tidak boleh diabaikan. HOMO-LUMO tidak boleh ditafsirkan berlebihan.

Keempat, membaca output memastikan **integritas laporan**. Laporan yang baik harus dapat menunjukkan dari mana data diambil.

Dengan demikian, output adalah sumber bukti, bukan sekadar lampiran.

27. Implikasi Pedagogis Membaca Output

Secara pedagogis, membaca output melatih mahasiswa untuk berpikir kritis dan teliti. Mahasiswa belajar bahwa kimia komputasi bukan sekadar menekan tombol. Mereka harus memahami hubungan antara input, proses perhitungan, output, dan kesimpulan.

Keterampilan yang dilatih:

1. Mengenali informasi penting dalam file teks.
2. Membedakan output valid dan tidak valid.
3. Menghubungkan angka dengan konsep kimia.
4. Menyusun ringkasan data.
5. Menyiapkan data untuk parsing Python.
6. Menulis laporan berbasis bukti.
7. Menghindari kesalahan interpretasi.

Keterampilan ini penting bagi calon guru dan calon peneliti. Guru yang memahami output komputasi dapat menjelaskan hasil simulasi kepada siswa secara lebih bertanggung jawab. Peneliti yang memahami output dapat menilai keandalan data sebelum menarik kesimpulan.

28. Format Catatan Manual Sebelum Otomatisasi

Sebelum menggunakan Python untuk parsing otomatis, mahasiswa sebaiknya mampu membuat catatan manual dari output. Format catatan manual dapat berupa:

```
Nama molekul:
Nama file output:
Software:
Metode:
Basis set:
Charge:
Multiplicity:
Energi total:
Geometri akhir:
Momen dipol:
Frekuensi terendah:
Frekuensi imajiner:
HOMO:
LUMO:
Celah HOMO-LUMO:
```

Status SCF:
 Status optimasi:
 Status job:
 Catatan:

Catatan ini menjadi jembatan menuju subbab 11.4, yaitu Python untuk parsing output. Jika mahasiswa tidak tahu informasi apa yang harus dicari secara manual, maka ia juga tidak akan mampu membuat kode parsing yang benar.

29. Kesalahan Umum Saat Membaca Output

Beberapa kesalahan umum saat membaca output adalah:

Kesalahan	Dampak
Mengambil energi dari iterasi awal	Energi tidak mewakili hasil akhir
Tidak memeriksa konvergensi SCF	Energi elektronik tidak valid
Tidak memeriksa optimasi	Geometri belum minimum
Mengabaikan frekuensi imajiner	Struktur salah dianggap valid
Salah membaca charge	Sistem kimia berbeda dari yang dimaksud
Salah membaca multiplicity	Keadaan spin tidak sesuai
Menyamakan energi elektronik dan Gibbs	Analisis termodinamika keliru
Tidak mencatat basis set	Hasil tidak reproduibel
Mengambil HOMO-LUMO dari bagian yang salah	Analisis orbital keliru
Mengandalkan tampilan ringkas tanpa output lengkap	Validasi tidak memadai

Kesalahan ini akan dibahas lebih lanjut pada subbab 11.3, tetapi sejak subbab ini mahasiswa sudah harus menyadarinya.

30. Sintesis Subbab 11.1

Membaca file output komputasi adalah keterampilan inti dalam kimia komputasi. Output memuat informasi yang menentukan apakah hasil perhitungan dapat digunakan secara ilmiah. Mahasiswa harus mencari metode, basis set, charge, multiplicity, energi total, geometri akhir, momen dipol, frekuensi, HOMO-LUMO, dan status konvergensi.

Alur sintesisnya:

```
file output
→ identitas perhitungan
→ validasi proses
→ data hasil
```

→ interpretasi kimia
→ laporan praktikum

Dengan menguasai pembacaan output, mahasiswa dapat menghindari kesalahan serius dalam analisis. Mereka juga lebih siap menggunakan Python untuk parsing otomatis pada subbab berikutnya.

31. Transisi ke Subbab 11.2

Subbab 11.1 telah membahas cara membaca file output komputasi dan mencari informasi penting seperti metode, basis set, charge, multiplicity, energi total, geometri akhir, momen dipol, frekuensi, HOMO-LUMO, dan status konvergensi. Pembacaan output menjadi langkah awal untuk menentukan apakah hasil perhitungan layak digunakan dalam laporan praktikum.

Subbab berikutnya akan membahas validasi hasil praktikum. Pada subbab 11.2, mahasiswa akan mempelajari kriteria validasi yang lebih sistematis, yaitu job harus selesai normal, struktur harus konvergen, tidak boleh ada frekuensi imajiner untuk struktur minimum, geometri harus masuk akal, charge dan multiplicity harus benar, serta hasil sebaiknya dapat dibandingkan dengan literatur. Dengan demikian, pembacaan output pada subbab 11.1 akan dilanjutkan menjadi proses validasi data yang lebih ketat pada subbab 11.2.

11.2 Validasi Hasil Praktikum

Validasi hasil praktikum adalah proses memastikan bahwa data yang diperoleh dari perhitungan kimia komputasi **layak digunakan, sesuai dengan sistem kimia yang dimaksud, dapat dipertanggungjawabkan, dan tidak mengandung kesalahan metodologis yang mendasar**. Dalam kimia komputasi, angka yang muncul dalam file output tidak otomatis berarti benar. Software dapat menghasilkan energi, geometri, frekuensi, momen dipol, atau orbital, tetapi pengguna tetap harus memeriksa apakah perhitungan tersebut selesai dengan benar, apakah struktur telah mencapai minimum, apakah input kimianya tepat, dan apakah hasilnya masuk akal bila dibandingkan dengan teori atau literatur.

Validasi merupakan tahap lanjutan setelah membaca file output. Pada subbab 11.1, mahasiswa telah mempelajari cara mencari informasi penting dari output, seperti metode, basis set, charge, multiplicity, energi total, geometri akhir, momen dipol, frekuensi, HOMO-LUMO, dan status konvergensi. Pada subbab ini, informasi tersebut digunakan untuk menjawab pertanyaan yang lebih mendasar:

Apakah hasil praktikum ini valid?

Apakah data ini boleh dipakai dalam laporan?

Apakah kesimpulan kimia yang diambil dari data ini dapat dipercaya?

Dalam Praktikum Kimia Kuantum, validasi hasil dilakukan berdasarkan enam kriteria utama:

1. **Job selesai normal.**
2. **Struktur konvergen.**
3. **Tidak ada frekuensi imajiner untuk struktur minimum.**
4. **Geometri masuk akal.**
5. **Charge dan multiplicity benar.**
6. **Hasil dapat dibandingkan dengan literatur.**

Keenam kriteria tersebut tidak boleh dipahami sebagai formalitas. Masing-masing memiliki fungsi ilmiah. Job yang tidak selesai normal dapat menghasilkan output yang belum lengkap. Struktur yang tidak konvergen belum dapat dianggap sebagai minimum. Frekuensi imajiner pada struktur minimum menunjukkan adanya masalah geometri. Geometri yang tidak masuk akal dapat menunjukkan optimasi menuju struktur yang salah. Charge dan multiplicity yang salah membuat sistem yang dihitung berbeda dari sistem kimia yang dimaksud. Perbandingan dengan literatur membantu menilai apakah hasil komputasi berada dalam kisaran yang wajar.

Dengan demikian, validasi hasil praktikum adalah bentuk **kontrol mutu ilmiah**. Tanpa validasi, laporan komputasi dapat tampak rapi tetapi sebenarnya tidak sah.

1. Pengertian Validasi dalam Kimia Komputasi

Validasi dalam kimia komputasi adalah proses pemeriksaan sistematis terhadap input, proses, dan output perhitungan untuk memastikan bahwa hasil yang diperoleh sesuai dengan tujuan perhitungan. Validasi tidak hanya memeriksa apakah software berhasil dijalankan, tetapi juga memeriksa apakah data yang dihasilkan memiliki makna kimia.

Secara sederhana:

```
input benar
→ proses perhitungan berjalan benar
→ output lengkap
→ hasil masuk akal
→ data layak dianalisis
```

Jika salah satu tahap bermasalah, hasil harus diperiksa ulang.

Contoh:

```
input salah charge
→ jumlah elektron salah
→ energi salah
→ kesimpulan tidak valid
```

atau:

optimasi tidak konvergen
→ geometri belum minimum
→ energi tidak layak dibandingkan

Validasi adalah pembeda antara penggunaan software secara mekanis dan penggunaan software secara ilmiah. Mahasiswa yang hanya mengambil angka dari output tanpa validasi belum dapat disebut melakukan analisis kimia komputasi secara benar.

2. Latar Historis Pentingnya Validasi

Sejak awal perkembangan kimia komputasi, validasi menjadi bagian penting dari praktik ilmiah. Pada masa ketika perhitungan dilakukan dengan input teks dan output panjang, peneliti harus membaca output secara manual untuk memastikan bahwa perhitungan selesai normal, SCF konvergen, geometri mencapai titik stasioner, dan hasil sesuai dengan teori kimia.

Ketika software berkembang menjadi lebih ramah pengguna melalui antarmuka grafis dan platform web, proses menjalankan perhitungan menjadi lebih mudah. Namun, kemudahan ini tidak menghilangkan kebutuhan validasi. Justru, pengguna pemula sering terjebak karena mengira bahwa tampilan visual yang menarik atau file output yang berhasil terbentuk selalu berarti hasil benar. Dalam praktik profesional, asumsi seperti itu berbahaya.

Literatur kimia komputasi seperti Jensen, Cramer, Leach, Lewars, Foresman-Frisch, serta panduan penggunaan Gaussian, ORCA, GAMESS, Psi4, dan PySCF menekankan bahwa hasil komputasi harus diperiksa dari sisi metode, konvergensi, geometri, frekuensi, basis set, dan kesesuaian kimia. Hal ini menunjukkan bahwa validasi bukan tambahan, tetapi bagian inti dari metodologi komputasi.

Secara historis, validasi tumbuh dari kebutuhan untuk memastikan bahwa perhitungan numerik yang kompleks benar-benar merepresentasikan sistem kimia yang dimaksud. Dalam konteks pendidikan, validasi melatih mahasiswa untuk tidak mudah percaya pada angka, tetapi menilai angka secara kritis.

3. Hubungan Validasi dengan Reprodusibilitas

Validasi berkaitan erat dengan reprodusibilitas. Reprodusibilitas berarti hasil dapat diperoleh kembali jika perhitungan diulang dengan input, metode, basis set, dan kondisi yang sama. Hasil yang tidak tervalidasi sulit direproduksi karena mungkin berasal dari output yang gagal, input yang salah, atau interpretasi yang keliru.

Agar hasil praktikum reprodusibel, mahasiswa harus mencatat:

1. Nama software.
2. Versi software jika tersedia.

3. Metode.
4. Basis set.
5. Charge.
6. Multiplicity.
7. Jenis job.
8. Model pelarut jika digunakan.
9. Status SCF.
10. Status optimasi.
11. Frekuensi imajiner.
12. Energi akhir.
13. Geometri akhir.
14. Nama file input dan output.

Tanpa catatan tersebut, hasil sulit diperiksa ulang. Misalnya, laporan yang hanya menulis “energi molekul air adalah sekian Hartree” tidak cukup. Laporan harus menjelaskan energi tersebut diperoleh dengan metode apa, basis set apa, charge berapa, multiplicity berapa, dan apakah optimasi sudah konvergen.

Dengan demikian:

```
validasi
→ memastikan hasil layak

reprodusibilitas
→ memastikan hasil dapat diulang
```

Keduanya merupakan prinsip penting dalam sains komputasi.

4. Kriteria 1: Job Selesai Normal

Kriteria pertama validasi adalah memastikan bahwa **job selesai normal**. Job adalah tugas perhitungan yang dijalankan oleh software, misalnya optimasi geometri, single point energy, frekuensi, atau optimasi-frekuensi. Job selesai normal berarti software menyelesaikan perhitungan tanpa berhenti karena error.

Dalam output, pesan selesai normal dapat muncul dengan berbagai bentuk, misalnya:

```
Normal termination
ORCA TERMINATED NORMALLY
EXECUTION OF GAMESS TERMINATED NORMALLY
Job finished successfully
```

Jika output menunjukkan pesan seperti itu, langkah validasi pertama terpenuhi. Namun, job selesai normal belum berarti semua data otomatis valid. Itu hanya berarti program tidak berhenti karena error besar. Setelah itu, mahasiswa tetap harus memeriksa konvergensi, geometri, frekuensi, charge, multiplicity, dan data hasil.

Sebaliknya, jika output menunjukkan pesan error, hasil tidak boleh langsung digunakan. Contoh pesan bermasalah:

```
Error termination
SCF failed to converge
Optimization failed
Maximum number of cycles exceeded
Insufficient memory
File not found
```

Jika job tidak selesai normal, mahasiswa harus mencari penyebabnya. Penyebab dapat berupa struktur awal buruk, charge salah, basis set tidak tersedia, memori tidak cukup, SCF gagal, atau input tidak sesuai format.

5. Mengapa Job Selesai Normal Belum Cukup

Job selesai normal adalah syarat awal, bukan validasi akhir. Banyak mahasiswa pemula mengira bahwa jika output berakhir normal, semua hasil otomatis benar. Anggapan ini keliru.

Contoh:

```
job selesai normal
tetapi charge salah
→ hasil tidak valid
```

atau:

```
job selesai normal
tetapi struktur memiliki frekuensi imajiner
→ struktur belum valid sebagai minimum
```

atau:

```
job selesai normal
tetapi geometri akhir berubah menjadi struktur lain
→ hasil tidak sesuai tujuan
```

Dengan demikian, job selesai normal hanya menunjukkan bahwa program menyelesaikan instruksi. Validasi ilmiah tetap memerlukan pemeriksaan kimia.

Pernyataan laporan yang baik:

Perhitungan selesai normal. Selanjutnya, validasi dilakukan dengan memeriksa status optimasi, frekuensi imajiner, geometri akhir, charge, multiplicity, dan kesesuaian hasil dengan literatur.

Pernyataan tersebut menunjukkan bahwa mahasiswa memahami validasi sebagai proses bertahap.

6. Kriteria 2: Struktur Konvergen

Kriteria kedua adalah memastikan bahwa **struktur konvergen**. Struktur konvergen berarti optimasi geometri telah mencapai kriteria yang ditetapkan oleh software. Dalam optimasi geometri, software mengubah posisi atom secara bertahap untuk mencari struktur dengan energi minimum atau titik stasioner. Optimasi dianggap selesai jika gaya, perubahan energi, dan perpindahan geometri memenuhi batas toleransi.

Pesan konvergensi dapat berupa:

```
Optimization completed
Geometry optimization converged
Stationary point found
Convergence criteria satisfied
```

Jika optimasi belum konvergen, struktur akhir belum dapat dianggap sebagai hasil final. Energi dari struktur tersebut tidak layak digunakan untuk membandingkan kestabilan atau menghitung energi reaksi.

Contoh pesan bermasalah:

```
Optimization did not converge
Maximum optimization cycles exceeded
Geometry optimization failed
```

Dalam kasus seperti itu, mahasiswa harus memperbaiki struktur awal, menambah jumlah langkah optimasi, mengubah metode, memperbaiki koordinat, atau menjalankan optimasi ulang.

7. Makna Konvergensi Struktur secara Kimia

Konvergensi bukan hanya istilah numerik. Secara kimia, struktur konvergen menunjukkan bahwa molekul telah mencapai susunan atom yang stabil menurut model komputasi yang digunakan. Struktur ini dapat berupa minimum lokal atau titik stasioner tertentu. Untuk praktikum yang membahas struktur stabil, yang diharapkan adalah minimum lokal.

Optimasi yang belum konvergen dapat menimbulkan masalah:

1. Panjang ikatan belum stabil.
2. Sudut ikatan masih berubah signifikan.
3. Energi belum mencapai nilai minimum.
4. Frekuensi dapat tidak bermakna.
5. Energi total tidak layak dibandingkan.
6. Momen dipol dapat berubah jika optimasi dilanjutkan.

7. Kesimpulan struktur dapat salah.

Contoh:

Jika optimasi etanol belum konvergen, maka perbandingan energi konformer etanol belum dapat dipercaya karena struktur akhir belum benar-benar mewakili minimum masing-masing konformer.

Validasi konvergensi struktur sangat penting pada semua praktikum, terutama optimasi geometri, analisis orbital, frekuensi IR, dan energi reaksi.

8. Kriteria 3: Tidak Ada Frekuensi Imajiner untuk Struktur Minimum

Kriteria ketiga adalah memastikan bahwa **tidak ada frekuensi imajiner untuk struktur minimum**. Setelah struktur dioptimasi, perhitungan frekuensi digunakan untuk memeriksa apakah struktur tersebut benar-benar minimum lokal pada permukaan energi potensial.

Untuk struktur minimum, semua frekuensi vibrasi harus positif. Jika ada frekuensi imajiner, output sering menampilkannya sebagai frekuensi negatif.

Contoh:

```
Frequencies -- 3650.2 1600.5 1200.7
```

Semua frekuensi positif. Struktur dapat dianggap minimum lokal.

Contoh bermasalah:

```
Frequencies -- -250.4 125.8 340.2
```

Frekuensi negatif menunjukkan frekuensi imajiner. Untuk struktur minimum, ini berarti struktur belum valid.

Frekuensi imajiner memiliki makna geometris. Mode imajiner menunjukkan arah di mana struktur masih dapat bergerak menuju energi lebih rendah. Oleh karena itu, struktur dengan frekuensi imajiner belum benar-benar berada pada minimum.

9. Frekuensi Imajiner dan Keadaan Transisi

Mahasiswa harus membedakan struktur minimum dan keadaan transisi. Pada struktur minimum, frekuensi imajiner tidak boleh ada. Namun, pada keadaan transisi, satu frekuensi imajiner justru diharapkan karena keadaan transisi adalah titik pelana orde pertama pada permukaan energi potensial.

Dalam praktikum Bab 4 sampai Bab 10, sebagian besar sistem yang dihitung adalah struktur minimum: molekul stabil, konformer, isomer, monomer, dimer, reaktan, dan produk. Karena itu, kriteria validasinya adalah:

struktur minimum
→ tidak boleh ada frekuensi imajiner

Jika kelak praktikum membahas keadaan transisi, kriterianya berbeda:

keadaan transisi
→ satu frekuensi imajiner
→ mode imajiner harus sesuai koordinat reaksi

Namun, untuk subbab ini, fokus validasi adalah struktur minimum. Maka, frekuensi imajiner harus dianggap sebagai tanda bahwa struktur perlu diperbaiki atau dioptimasi ulang.

10. Contoh Validasi Frekuensi pada Dimer Air

Dimer air digunakan untuk memahami ikatan hidrogen. Setelah optimasi, mahasiswa perlu memeriksa frekuensi.

Jika tidak ada frekuensi imajiner, struktur dimer dapat dianggap sebagai minimum lokal. Selanjutnya, energi dimer dapat digunakan untuk menghitung energi dimerisasi.

Namun, jika ada frekuensi imajiner, struktur perlu diperiksa. Frekuensi imajiner mungkin menunjukkan bahwa orientasi dua molekul air belum stabil, jarak antarmolekul tidak tepat, atau optimasi berhenti pada titik yang belum benar.

Pernyataan laporan:

Perhitungan frekuensi dimer air tidak menunjukkan frekuensi imajiner. Oleh karena itu, struktur dimer hasil optimasi dapat dianggap sebagai minimum lokal dan digunakan untuk analisis energi dimerisasi.

Jika ada masalah:

Ditemukan satu frekuensi imajiner pada struktur dimer air. Karena sistem dimaksudkan sebagai struktur minimum, hasil ini belum digunakan untuk analisis energi dan perlu dilakukan optimasi ulang.

Pernyataan seperti ini menunjukkan sikap ilmiah yang benar.

11. Kriteria 4: Geometri Masuk Akal

Kriteria keempat adalah memastikan bahwa **geometri akhir masuk akal secara kimia**. Geometri masuk akal berarti susunan atom hasil optimasi sesuai dengan struktur molekul yang dimaksud dan tidak mengandung ikatan, jarak, sudut, atau orientasi yang tidak realistis.

Geometri harus diperiksa karena software dapat menyelesaikan perhitungan secara normal tetapi menghasilkan struktur yang tidak sesuai tujuan. Misalnya, struktur awal dimer dapat terpisah menjadi dua monomer jauh. Konformer awal dapat berubah menjadi konformer lain. Molekul dapat mengalami fragmentasi. Gugus yang seharusnya berinteraksi tidak lagi berinteraksi.

Hal yang perlu diperiksa:

1. Panjang ikatan utama.
2. Sudut ikatan.
3. Bentuk molekul.
4. Orientasi gugus penting.
5. Jarak ikatan hidrogen.
6. Apakah dimer masih dimer.
7. Apakah konformer akhir masih sesuai konformer awal.
8. Apakah tidak terjadi fragmentasi.
9. Apakah geometri sesuai teori kimia dasar.
10. Apakah hasil dapat divisualisasikan dengan benar.

Geometri masuk akal tidak harus identik dengan data eksperimen, tetapi harus berada dalam kisaran kimia yang wajar.

12. Contoh Geometri Masuk Akal

Contoh validasi geometri:

Air, H₂O

Struktur akhir harus bengkok.
Momen dipol tidak nol.
Frekuensi tidak boleh imajiner.

Jika hasil optimasi membuat H₂O linear, maka geometri tidak sesuai keadaan dasar air.

Karbon dioksida, CO₂

Struktur akhir harus linear.
Momen dipol total mendekati nol.

Jika CO₂ menjadi sangat bengkok tanpa alasan, perlu diperiksa input atau keadaan elektroniknya.

Amonium, NH_4^+

Struktur akhir harus mendekati tetrahedral.
Charge harus +1.
Multiplicity harus 1.

Jika NH_4^+ dihitung netral, geometri dan energi tidak valid untuk ion amonium.

Dimer asam asetat

Dimer akhir seharusnya mempertahankan dua ikatan hidrogen jika model yang dimaksud adalah dimer siklik.

Jika setelah optimasi dua monomer menjauh, maka struktur akhir bukan dimer yang dimaksud.

13. Geometri dan Visualisasi

Validasi geometri sebaiknya tidak hanya dilakukan melalui angka koordinat, tetapi juga melalui visualisasi. Software seperti Avogadro, GaussView, Molden, Chemcraft, atau visualizer lain dapat digunakan untuk membuka struktur akhir.

Visualisasi membantu mahasiswa melihat:

1. Apakah atom terhubung secara benar.
2. Apakah ada ikatan terlalu pendek.
3. Apakah ada atom terpisah terlalu jauh.
4. Apakah dimer masih terbentuk.
5. Apakah bentuk molekul sesuai teori.
6. Apakah orientasi konformer sesuai tujuan.

Namun, visualisasi juga harus dikombinasikan dengan angka. Tampilan visual dapat menyesatkan jika skala atau deteksi ikatan otomatis tidak tepat. Oleh karena itu, panjang ikatan dan jarak penting tetap perlu diperiksa.

Pernyataan laporan:

Geometri akhir diperiksa melalui visualisasi dan nilai panjang ikatan utama. Struktur yang diperoleh sesuai dengan bentuk molekul yang diharapkan, sehingga digunakan untuk analisis lanjutan.

14. Kriteria 5: Charge dan Multiplicity Benar

Kriteria kelima adalah memastikan bahwa **charge dan multiplicity benar**. Ini adalah salah satu aspek paling mendasar dalam validasi. Charge menentukan jumlah elektron total, sedangkan

multiplicity menentukan keadaan spin. Jika salah satu salah, sistem yang dihitung tidak sesuai dengan sistem kimia yang dimaksud.

Contoh:

Spesies	Charge	Multiplicity
H ₂ O	0	1
CO ₂	0	1
NH ₃	0	1
NH ₄ ⁺	+1	1
CH ₃ ·	0	2
O ₂ keadaan dasar	0	3

Dalam praktikum Bab 4 sampai Bab 10, sebagian besar sistem adalah singlet netral. Namun, ada pengecualian penting seperti NH₄⁺ dalam protonasi amonia. Jika NH₄⁺ dihitung dengan charge 0, hasilnya tidak valid.

Kesalahan charge dan multiplicity dapat menyebabkan:

1. Jumlah elektron salah.
2. Energi salah.
3. Orbital salah.
4. Momen dipol salah.
5. Geometri tidak sesuai.
6. Frekuensi tidak bermakna.
7. Kesimpulan reaksi salah.

Karena itu, charge dan multiplicity harus diperiksa pada input dan output.

15. Contoh Kesalahan Charge

Contoh kesalahan serius:

NH₄⁺ dihitung dengan charge 0 dan multiplicity 1.

Masalahnya, NH₄⁺ adalah ion bermuatan +1. Jika dihitung netral, software menghitung sistem dengan jumlah elektron berbeda. Hasil tersebut bukan energi NH₄⁺.

Pernyataan validasi:

NH₄⁺ divalidasi sebagai spesies bermuatan +1 dengan multiplicity 1. Dengan demikian, jumlah elektron sesuai dengan ion amonium yang dimaksud.

Jika salah:

Output menunjukkan charge 0 untuk struktur yang dimaksudkan sebagai NH_4^+ . Oleh karena itu, perhitungan tidak valid dan harus diulang dengan charge +1.

Contoh ini menunjukkan bahwa validasi input sama pentingnya dengan validasi output.

16. Contoh Kesalahan Multiplicity

Multiplicity juga dapat menjadi sumber kesalahan. Contoh yang sering dibahas adalah O_2 . Keadaan dasar molekul O_2 adalah triplet, bukan singlet. Jika mahasiswa menghitung O_2 sebagai singlet tanpa alasan khusus, hasilnya tidak mewakili keadaan dasar O_2 .

Pernyataan validasi:

Untuk molekul O_2 keadaan dasar, multiplicity harus dipilih sebagai triplet. Jika O_2 dihitung sebagai singlet, hasilnya harus ditafsirkan sebagai keadaan elektronik berbeda, bukan keadaan dasar.

Walaupun O_2 mungkin bukan sistem utama dalam bab ini, contoh tersebut penting untuk menunjukkan bahwa multiplicity bukan formalitas. Dalam sistem radikal, ion transisi, atau molekul dengan elektron tidak berpasangan, multiplicity sangat menentukan hasil.

17. Kriteria 6: Hasil Dapat Dibandingkan dengan Literatur

Kriteria keenam adalah memastikan bahwa hasil dapat dibandingkan dengan literatur atau nilai referensi yang relevan. Perbandingan literatur tidak selalu berarti hasil harus sama persis. Perhitungan komputasi bergantung pada metode, basis set, model pelarut, dan asumsi. Namun, hasil harus berada dalam kisaran yang masuk akal.

Contoh data yang dapat dibandingkan:

1. Panjang ikatan.
2. Sudut ikatan.
3. Momen dipol.
4. Frekuensi vibrasi utama.
5. Urutan kestabilan isomer.
6. Energi relatif konformer.
7. Kecenderungan dimerisasi.
8. Pola spektrum IR.
9. Nilai HOMO-LUMO secara kualitatif.
10. Tren efek pelarut.

Literatur yang dapat digunakan meliputi buku kimia fisik, buku kimia komputasi, artikel penelitian, dokumentasi software, database spektrum, data eksperimen, atau sumber akademik bereputasi.

Dalam laporan praktikum, perbandingan literatur dapat ditulis secara singkat:

Nilai momen dipol air hasil perhitungan menunjukkan bahwa H_2O bersifat polar, sesuai dengan literatur yang menyatakan bahwa air memiliki bentuk bengkok dan momen dipol tidak nol.

atau:

Urutan kestabilan trans-2-butena yang lebih rendah energinya dibanding cis-2-butena sesuai dengan penjelasan sterik dalam kimia organik.

18. Perbandingan Literatur sebagai Validasi Tren

Pada praktikum pendidikan, perbandingan dengan literatur sering lebih penting sebagai validasi tren daripada pencocokan angka absolut. Misalnya, jika perhitungan menggunakan basis set kecil, nilai frekuensi vibrasi mungkin tidak sama persis dengan eksperimen. Namun, posisi relatif puncak dan identifikasi gugus fungsi dapat tetap benar.

Contoh:

frekuensi $\text{C}=\text{O}$ komputasi
→ mungkin berbeda dari eksperimen

tetapi
→ tetap berada di wilayah khas regangan karbonil

Demikian pula, energi relatif isomer dapat dibandingkan secara tren:

trans-2-butena lebih stabil daripada cis-2-butena
→ tren sesuai teori sterik

Perbandingan tren membantu mahasiswa memahami bahwa validasi tidak selalu berarti angka identik. Perbedaan dapat berasal dari metode, basis set, fase gas, pelarut, suhu, atau koreksi skala.

19. Batasan Perbandingan Literatur

Perbandingan literatur juga harus dilakukan hati-hati. Tidak semua data literatur dapat dibandingkan langsung dengan hasil praktikum. Misalnya, hasil eksperimen larutan tidak selalu dapat dibandingkan langsung dengan perhitungan fase gas. Frekuensi eksperimen dipengaruhi lingkungan, anharmonitas, dan kondisi pengukuran. Energi bebas eksperimen bergantung pada suhu, pelarut, konsentrasi, dan keadaan standar.

Karena itu, mahasiswa harus bertanya:

1. Apakah data literatur fase gas atau larutan?
2. Apakah suhu sama?
3. Apakah molekul dalam keadaan yang sama?
4. Apakah metode komputasi sebanding?
5. Apakah nilai literatur eksperimen atau teori?
6. Apakah diperlukan faktor skala untuk frekuensi?
7. Apakah perbandingan dilakukan secara angka atau tren?

Pernyataan kritis:

Perbandingan dengan literatur dilakukan secara kualitatif karena hasil praktikum menggunakan model fase gas dan basis set terbatas, sedangkan data eksperimen dapat dipengaruhi oleh kondisi lingkungan dan anharmonitas.

Pernyataan seperti ini menunjukkan kematangan ilmiah.

20. Format Tabel Validasi Hasil Praktikum

Agar validasi rapi, mahasiswa sebaiknya membuat tabel validasi. Format yang dianjurkan:

Kriteria Validasi	Status	Keterangan
Job selesai normal	Ya/Tidak	Pesan akhir output
Struktur konvergen	Ya/Tidak	Optimasi memenuhi kriteria
Frekuensi imajiner	Ada/Tidak	Untuk minimum harus tidak ada
Geometri masuk akal	Ya/Tidak	Dicek visual dan parameter ikatan
Charge benar	Ya/Tidak	Sesuai spesies
Multiplicity benar	Ya/Tidak	Sesuai keadaan spin
Energi tersedia	Ya/Tidak	Energi akhir terbaca
Literatur/tren sesuai	Ya/Tidak/Sebagian	Dibandingkan secara wajar

Contoh tabel validasi untuk H₂O:

Kriteria Validasi	Status	Keterangan
Job selesai normal	Ya	Output berakhir normal
Struktur konvergen	Ya	Optimasi selesai
Frekuensi imajiner	Tidak ada	Semua frekuensi positif
Geometri masuk akal	Ya	Struktur bengkok
Charge benar	Ya	0
Multiplicity benar	Ya	1
Energi tersedia	Ya	Energi akhir terbaca
Literatur/tren sesuai	Ya	Air polar dan bengkok

Tabel seperti ini membuat validasi mudah diperiksa oleh dosen dan mahasiswa.

21. Validasi pada Praktikum Optimasi Geometri

Pada praktikum optimasi geometri, validasi berfokus pada:

1. Job selesai normal.
2. Optimasi konvergen.
3. Geometri akhir masuk akal.
4. Panjang ikatan dan sudut ikatan wajar.
5. Charge dan multiplicity benar.

Jika perhitungan frekuensi belum dilakukan, mahasiswa belum dapat memastikan bahwa struktur adalah minimum melalui frekuensi. Namun, jika praktikum mencakup frekuensi, maka tidak adanya frekuensi imajiner menjadi validasi tambahan.

Contoh analisis:

Optimasi geometri formaldehida selesai konvergen. Struktur akhir menunjukkan gugus karbonil C=O yang tetap terbentuk dan geometri planar yang masuk akal. Dengan demikian, struktur dapat digunakan untuk analisis energi dan sifat molekul.

22. Validasi pada Praktikum Orbital HOMO-LUMO

Pada praktikum HOMO-LUMO, validasi berfokus pada:

1. Energi elektronik konvergen.
2. Struktur yang digunakan valid.
3. Orbital terbaca dari output akhir.
4. HOMO dan LUMO diambil dari bagian orbital yang benar.
5. Charge dan multiplicity benar.
6. Interpretasi tidak berlebihan.

Kesalahan umum adalah mengambil orbital dari struktur yang belum dioptimasi atau dari output SCF yang tidak konvergen.

Pernyataan validasi:

Analisis HOMO-LUMO dilakukan pada struktur yang telah dioptimasi dan SCF dinyatakan konvergen. Oleh karena itu, energi orbital yang digunakan berasal dari keadaan elektronik yang konsisten.

23. Validasi pada Praktikum Momen Dipol

Pada praktikum momen dipol, validasi mencakup:

1. Struktur akhir benar.
2. Orientasi molekul tidak mengubah nilai total dipol.
3. Momen dipol total dicatat dalam Debye.
4. Charge dan multiplicity benar.
5. Interpretasi sesuai geometri dan simetri molekul.

Contoh:

CO₂ memiliki momen dipol total mendekati nol karena struktur linear dan simetris. Jika output menunjukkan momen dipol besar untuk CO₂, geometri atau input perlu diperiksa ulang.

Validasi momen dipol harus dikaitkan dengan geometri dan simetri.

24. Validasi pada Praktikum Frekuensi dan Spektrum IR

Pada praktikum frekuensi dan spektrum IR, validasi sangat bergantung pada frekuensi.

Kriteria:

1. Optimasi konvergen.
2. Tidak ada frekuensi imajiner untuk struktur minimum.
3. Mode vibrasi sesuai struktur.
4. Intensitas IR terbaca.
5. Frekuensi berada dalam kisaran wajar.
6. Jika perlu, faktor skala digunakan secara tepat.
7. Gugus fungsi sesuai puncak utama.

Contoh analisis:

Spektrum IR formaldehida divalidasi dengan memastikan tidak ada frekuensi imajiner dan puncak regangan C=O muncul pada wilayah khas karbonil. Perbedaan angka terhadap eksperimen dapat disebabkan oleh pendekatan harmonik dan level teori yang digunakan.

Validasi IR tidak cukup hanya melihat grafik; frekuensi imajiner dan kesesuaian mode harus diperiksa.

25. Validasi pada Praktikum Energi Reaksi dan Termodinamika

Pada praktikum energi reaksi, validasi harus dilakukan untuk semua spesies reaktan dan produk. Satu spesies yang tidak valid dapat merusak seluruh perhitungan ΔE , ΔH , dan ΔG .

Kriteria:

1. Semua job selesai normal.
2. Semua struktur konvergen.
3. Tidak ada frekuensi imajiner untuk semua minimum.
4. Semua spesies menggunakan metode dan basis set yang sama.
5. Charge dan multiplicity semua spesies benar.
6. Koefisien stoikiometri benar.
7. Energi diambil dari bagian output yang tepat.
8. Model pelarut konsisten jika digunakan.
9. Hasil dapat dibandingkan dengan tren literatur.

Contoh:

Pada dimerisasi air, energi monomer dan dimer hanya dibandingkan setelah keduanya dioptimasi, tidak memiliki frekuensi imajiner, dan dihitung dengan metode serta basis set yang sama.

Jika monomer valid tetapi dimer tidak valid, energi dimerisasi tidak boleh digunakan.

26. Validasi Konsistensi Metode

Konsistensi metode adalah bagian penting dari validasi. Reaktan dan produk tidak boleh dihitung dengan metode berbeda jika energinya akan dibandingkan secara langsung.

Contoh salah:

reaktan dihitung dengan HF/STO-3G
produk dihitung dengan B3LYP/6-31G(d)

Hasil ΔE dari perbandingan tersebut tidak valid karena selisih energi mencampur perbedaan kimia dan perbedaan metode.

Contoh benar:

reaktan dihitung dengan B3LYP/6-31G(d)
produk dihitung dengan B3LYP/6-31G(d)

Pernyataan validasi:

Seluruh spesies dalam sistem reaksi dihitung dengan level teori yang sama, sehingga perbandingan energi dapat dilakukan secara konsisten.

Konsistensi metode sangat penting untuk semua perbandingan energi: isomer, konformer, reaktan-produk, monomer-dimer, dan fase gas-pelarut.

27. Validasi Model Pelarut

Jika model pelarut digunakan, validasi harus memastikan bahwa model diterapkan secara konsisten. Jangan menghitung reaktan dalam fase gas dan produk dalam pelarut lalu membandingkannya langsung sebagai energi reaksi pelarut.

Validasi model pelarut mencakup:

1. Model pelarut dicatat.
2. Nama pelarut dicatat.
3. Semua spesies dalam satu kondisi dihitung dengan model yang sama.
4. Perbandingan gas-pelarut dilakukan dengan jelas.
5. Interpretasi efek pelarut tidak berlebihan.
6. Keterbatasan model kontinu disebutkan.

Contoh:

Untuk membandingkan ΔG fase gas dan pelarut pada protonasi amonia, NH_3 , H^+ , dan NH_4^+ harus dihitung dalam fase gas dan kemudian dihitung kembali dengan model pelarut yang sama.

Pernyataan validasi:

Model pelarut diterapkan secara konsisten pada semua spesies dalam reaksi, sehingga perubahan ΔG dapat ditafsirkan sebagai efek pelarut terhadap stabilisasi relatif reaktan dan produk.

28. Validasi Satuan Energi

Satuan energi harus diperiksa. Energi output umumnya dalam Hartree, sedangkan laporan kimia sering menggunakan kJ/mol atau kcal/mol. Kesalahan satuan dapat menghasilkan interpretasi yang salah.

Konversi yang sering digunakan:

$$1 \text{ Hartree} \approx 2625,5 \text{ kJ/mol}$$

Validasi satuan mencakup:

1. Mengetahui satuan energi output.
2. Menghitung selisih energi sebelum konversi.
3. Mengonversi dengan faktor yang benar.
4. Mencantumkan satuan dalam tabel.
5. Tidak mencampur Hartree dan kJ/mol.

Contoh kesalahan:

$\Delta E = 0,01$ Hartree ditulis sebagai $0,01$ kJ/mol

Kesalahan ini sangat besar karena:

$0,01$ Hartree $\approx 26,255$ kJ/mol

Pernyataan validasi:

Energi dari output dibaca dalam Hartree. Selisih energi dihitung terlebih dahulu dalam Hartree, kemudian dikonversi ke kJ/mol menggunakan faktor 2625,5.

29. Validasi Data Sebelum Otomatisasi Python

Sebelum data diproses dengan Python, mahasiswa harus memastikan bahwa data manual valid. Python tidak dapat membedakan apakah data berasal dari output yang benar atau salah. Python hanya mengolah angka yang diberikan.

Prinsip penting:

data salah
→ Python menghitung salah dengan rapi

Karena itu, sebelum parsing atau pengolahan otomatis, mahasiswa harus memvalidasi:

1. File output lengkap.
2. Job selesai normal.
3. Energi akhir benar.
4. Struktur valid.
5. Frekuensi valid.
6. Charge dan multiplicity benar.
7. Metode dan basis set konsisten.
8. Satuan jelas.
9. Data yang dimasukkan ke CSV sesuai output.

Otomatisasi Python mempercepat pekerjaan, tetapi tidak menggantikan penilaian ilmiah. Kode parsing dapat mencari angka, tetapi pengguna harus tahu apakah angka itu layak digunakan.

30. Contoh Lembar Validasi Praktikum

Mahasiswa dapat menggunakan format lembar validasi berikut:

Nama praktikum:
 Nama sistem:
 Nama file input:
 Nama file output:
 Software:
 Metode:
 Basis set:
 Charge:
 Multiplicity:
 Jenis job:
 Model pelarut:

VALIDASI:

☐ Job selesai normal
☐ SCF konvergen
☐ Optimasi geometri konvergen
☐ Geometri akhir masuk akal
☐ Tidak ada frekuensi imajiner untuk minimum
☐ Energi akhir terbaca
☐ Momen dipol terbaca jika diperlukan
☐ HOMO-LUMO terbaca jika diperlukan
☐ Satuan energi jelas
☐ Hasil dibandingkan dengan literatur/tren teori
☐ Data layak digunakan dalam laporan

Catatan:

Lembar validasi ini dapat dilampirkan dalam laporan atau digunakan sebagai catatan kerja.

31. Analisis Kritis terhadap Validasi

Validasi tidak selalu menghasilkan jawaban hitam-putih. Ada situasi ketika hasil sebagian valid tetapi memiliki keterbatasan. Misalnya, optimasi konvergen dan tidak ada frekuensi imajiner, tetapi basis set terlalu kecil untuk akurasi tinggi. Dalam kasus ini, hasil dapat digunakan untuk pembelajaran, tetapi kesimpulan kuantitatif harus dibatasi.

Contoh:

Hasil valid untuk menunjukkan tren umum, tetapi belum cukup untuk prediksi kuantitatif presisi tinggi karena menggunakan basis set minimal.

Atau:

Frekuensi vibrasi menunjukkan pola gugus fungsi yang sesuai, tetapi nilai absolut dapat berbeda dari eksperimen karena pendekatan harmonik dan belum menggunakan faktor skala.

Analisis kritis seperti ini menunjukkan bahwa mahasiswa memahami tingkat validitas hasil.

32. Kesalahan Umum dalam Validasi

Beberapa kesalahan umum dalam validasi hasil praktikum adalah:

Kesalahan	Dampak
Menganggap output selesai normal berarti valid	Kesalahan input tetap tidak terdeteksi
Tidak memeriksa charge	Sistem kimia salah
Tidak memeriksa multiplicity	Keadaan spin salah
Mengabaikan optimasi tidak konvergen	Geometri belum layak
Mengabaikan frekuensi imajiner	Struktur minimum salah
Tidak memeriksa geometri akhir	Struktur tidak sesuai tujuan
Tidak mencatat basis set	Hasil tidak reproduibel
Tidak membandingkan literatur	Tidak tahu apakah hasil wajar
Salah satuan energi	Kesimpulan numerik salah
Menggunakan Python sebelum validasi manual	Kesalahan data menjadi otomatis

Kesalahan ini menjadi penghubung menuju subbab 11.3, yang secara khusus membahas kesalahan umum.

33. Implikasi Keilmuan Validasi

Validasi memiliki implikasi keilmuan yang kuat. Dalam penelitian kimia komputasi, hasil yang tidak tervalidasi tidak dapat dijadikan dasar kesimpulan. Validasi menentukan apakah suatu angka dapat diperlakukan sebagai data ilmiah atau hanya sebagai keluaran program yang belum diperiksa.

Validasi juga membantu membedakan antara:

hasil numerik

→ angka dari software

data ilmiah

→ hasil numerik yang telah diperiksa dan ditafsirkan

Dengan demikian, validasi adalah tahap transformasi output menjadi data ilmiah.

34. Implikasi Pedagogis Validasi

Dalam pembelajaran, validasi melatih mahasiswa menjadi pengguna software yang kritis. Mahasiswa belajar bahwa:

1. Software tidak menggantikan pemahaman kimia.
2. Output harus diperiksa.
3. Geometri harus masuk akal.
4. Energi harus berasal dari struktur valid.
5. Frekuensi imajiner memiliki makna.
6. Charge dan multiplicity menentukan sistem.
7. Literatur membantu menilai kewajaran hasil.
8. Otomatisasi harus didahului validasi.

Keterampilan validasi sangat penting bagi calon guru kimia. Guru yang memahami validasi dapat menjelaskan kepada siswa bahwa simulasi komputer bukan sekadar gambar menarik, tetapi proses ilmiah yang memerlukan pemeriksaan dan penalaran.

35. Sintesis Subbab 11.2

Validasi hasil praktikum adalah proses memastikan bahwa hasil komputasi layak digunakan. Enam kriteria utama validasi adalah job selesai normal, struktur konvergen, tidak ada frekuensi imajiner untuk struktur minimum, geometri masuk akal, charge dan multiplicity benar, serta hasil dapat dibandingkan dengan literatur.

Alur validasi dapat diringkas sebagai berikut:

```

baca output
→ cek job selesai normal
→ cek SCF dan optimasi
→ cek geometri akhir
→ cek frekuensi imajiner
→ cek charge dan multiplicity
→ cek satuan dan energi
→ bandingkan dengan literatur
→ nyatakan data valid atau perlu diulang
  
```

Dengan validasi yang baik, laporan praktikum menjadi lebih kuat, lebih jujur, dan lebih dapat dipertanggungjawabkan. Validasi juga menjadi dasar sebelum mahasiswa menggunakan Python untuk parsing output dan otomatisasi laporan.

36. Transisi ke Subbab 11.3

Subbab 11.2 telah membahas validasi hasil praktikum dengan kriteria job selesai normal, struktur konvergen, tidak ada frekuensi imajiner untuk struktur minimum, geometri masuk akal, charge dan multiplicity benar, serta hasil dapat dibandingkan dengan literatur. Validasi ini memastikan bahwa data komputasi tidak langsung diterima begitu saja, tetapi diperiksa kelayakannya sebelum digunakan dalam analisis dan laporan.

Subbab berikutnya akan membahas kesalahan umum dalam praktikum kimia komputasi. Pada subbab 11.3, mahasiswa akan mempelajari berbagai sumber kesalahan seperti struktur awal buruk, charge salah, multiplicity salah, basis set tidak sesuai, SCF tidak konvergen, salah membaca satuan energi, dan salah menafsirkan frekuensi imajiner. Dengan memahami kesalahan umum tersebut, mahasiswa dapat mencegah kegagalan perhitungan dan meningkatkan kualitas validasi data.

11.3 Kesalahan Umum

Kesalahan umum dalam praktikum kimia komputasi adalah berbagai kekeliruan yang sering terjadi pada tahap **menyiapkan struktur, menentukan parameter input, menjalankan perhitungan, membaca output, mengolah data, dan menafsirkan hasil**. Kesalahan-kesalahan ini dapat terlihat sederhana, tetapi dampaknya sangat besar. Satu kesalahan kecil pada charge, multiplicity, basis set, satuan energi, atau interpretasi frekuensi imajiner dapat membuat seluruh laporan praktikum menjadi tidak valid.

Dalam kimia komputasi, kesalahan tidak selalu tampak jelas. Software dapat tetap menghasilkan file output, energi, grafik, atau struktur molekul meskipun input yang diberikan salah. Oleh karena itu, mahasiswa harus memahami bahwa hasil komputasi yang “keluar” dari software belum tentu benar. Hasil tersebut harus diperiksa melalui validasi sebagaimana dijelaskan pada subbab 11.2.

Subbab ini membahas tujuh kesalahan umum yang sering terjadi dalam praktikum kimia komputasi:

1. **Struktur awal buruk.**
2. **Charge salah.**
3. **Multiplicity salah.**
4. **Basis set tidak sesuai.**
5. **SCF tidak konvergen.**
6. **Salah membaca satuan energi.**
7. **Salah menafsirkan frekuensi imajiner.**

Ketujuh kesalahan tersebut dipilih karena sangat sering muncul dalam praktikum pemula maupun menengah. Kesalahan ini juga berkaitan langsung dengan bab-bab sebelumnya: optimasi geometri, orbital molekul, momen dipol, frekuensi IR, energi reaksi, termodinamika, efek pelarut, dan analisis output.

Tujuan subbab ini bukan hanya membuat daftar kesalahan, tetapi melatih mahasiswa untuk memahami **mengapa kesalahan itu terjadi, apa dampaknya terhadap hasil, bagaimana**

mengenalinya dalam output, dan bagaimana memperbaikinya. Dengan demikian, mahasiswa tidak hanya tahu bahwa suatu kesalahan berbahaya, tetapi juga mampu mendiagnosis dan memperbaiki perhitungan secara mandiri.

1. Makna Kesalahan dalam Kimia Komputasi

Kesalahan dalam kimia komputasi dapat dibedakan menjadi beberapa jenis.

Pertama, **kesalahan input**, yaitu kesalahan yang terjadi sebelum perhitungan dijalankan. Contohnya adalah struktur awal yang buruk, charge salah, multiplicity salah, basis set tidak sesuai, atau keyword tidak tepat.

Kedua, **kesalahan proses**, yaitu kesalahan yang terjadi selama perhitungan berlangsung. Contohnya adalah SCF tidak konvergen, optimasi gagal, memori tidak cukup, atau jumlah langkah optimasi habis.

Ketiga, **kesalahan output**, yaitu kesalahan dalam membaca hasil. Contohnya adalah mengambil energi dari iterasi awal, salah membaca satuan energi, salah mengambil HOMO-LUMO, atau tidak membedakan energi elektronik dan energi bebas Gibbs.

Keempat, **kesalahan interpretasi**, yaitu kesalahan dalam menjelaskan makna data. Contohnya adalah menyatakan bahwa ΔG negatif berarti reaksi pasti cepat, menganggap semua frekuensi negatif sebagai kesalahan fatal tanpa konteks, atau menganggap pelarut selalu menstabilkan produk.

Kelima, **kesalahan metodologis**, yaitu kesalahan dalam desain perhitungan. Contohnya adalah membandingkan energi reaktan dan produk yang dihitung dengan metode berbeda, membandingkan fase gas dengan fase pelarut secara tidak konsisten, atau menggunakan basis set terlalu kecil untuk kesimpulan kuantitatif yang terlalu jauh.

Secara ringkas:

kesalahan input

→ sistem yang dihitung salah

kesalahan proses

→ perhitungan tidak selesai dengan benar

kesalahan output

→ data yang diambil salah

kesalahan interpretasi

→ kesimpulan kimia salah

kesalahan metodologis

→ desain praktikum tidak sah

Mahasiswa perlu memahami bahwa kesalahan dalam kimia komputasi sering bersifat berantai. Kesalahan input dapat menghasilkan kesalahan output. Kesalahan output dapat menghasilkan kesalahan analisis. Kesalahan analisis dapat menghasilkan kesimpulan laporan yang keliru.

2. Latar Historis Kesalahan Komputasi

Dalam sejarah perkembangan kimia komputasi, kesalahan input dan interpretasi selalu menjadi persoalan penting. Pada masa awal, pengguna menulis file input secara manual dan membaca file output teks yang panjang. Kesalahan kecil seperti salah menulis charge, salah mengetik koordinat, atau salah memilih basis set dapat menyebabkan perhitungan gagal atau menghasilkan sistem yang tidak sesuai.

Ketika antarmuka grafis dan platform web berkembang, proses menjalankan perhitungan menjadi lebih mudah. Namun, jenis kesalahan tidak hilang. Kesalahan hanya berubah bentuk. Pengguna mungkin tidak lagi salah mengetik banyak koordinat, tetapi dapat salah memilih menu, salah menentukan muatan, salah menganggap struktur telah teroptimasi, atau salah membaca output ringkas.

Literatur kimia komputasi seperti Jensen, Cramer, Leach, Lewars, Foresman-Frisch, serta dokumentasi Gaussian, ORCA, GAMESS, Psi4, dan PySCF secara umum menekankan bahwa keberhasilan komputasi tidak hanya bergantung pada kemampuan software, tetapi pada keputusan pengguna. Software dapat menjalankan instruksi, tetapi pengguna harus memastikan bahwa instruksi tersebut benar secara kimia.

Prinsip pentingnya adalah:

software tidak memahami tujuan kimia pengguna secara otomatis
software hanya menjalankan input yang diberikan

Jika pengguna memberi input yang salah, software dapat tetap menghitung. Karena itu, kesalahan umum harus dipelajari sebagai bagian dari literasi komputasi.

3. Kesalahan 1: Struktur Awal Buruk

Struktur awal buruk adalah kesalahan ketika geometri molekul yang diberikan sebelum optimasi tidak masuk akal secara kimia. Struktur awal dapat memiliki panjang ikatan terlalu pendek, atom bertumpuk, gugus fungsi salah orientasi, ikatan hidrogen tidak terbentuk, dimer terlalu jauh, atau konformer tidak sesuai tujuan.

Struktur awal buruk dapat menyebabkan beberapa masalah:

1. Optimasi gagal konvergen.

2. Struktur akhir menjadi tidak sesuai tujuan.
3. Molekul mengalami fragmentasi.
4. Dimer terpisah menjadi monomer.
5. Konformer berubah menjadi konformer lain.
6. SCF sulit konvergen.
7. Energi awal sangat tinggi.
8. Waktu komputasi menjadi lebih lama.
9. Frekuensi imajiner muncul setelah optimasi.
10. Hasil tidak dapat digunakan untuk laporan.

Dalam praktikum optimasi geometri, struktur awal adalah titik mula pencarian minimum. Jika titik mula terlalu buruk, software dapat mengalami kesulitan menemukan minimum yang benar. Walaupun algoritma optimasi dapat memperbaiki geometri, algoritma tersebut bukan jaminan bahwa struktur akhir akan sesuai dengan sistem yang dimaksud.

Contoh struktur awal buruk:

dua atom H pada H_2O ditempatkan terlalu dekat dengan atom O
 dimer air diletakkan terlalu jauh sehingga tidak membentuk ikatan hidrogen
 asam asetat dimer tidak disusun dengan orientasi donor-akseptor yang benar
 etanol konformer awal berubah menjadi konformer lain setelah optimasi

Struktur awal harus diperiksa sebelum perhitungan dijalankan.

4. Dampak Struktur Awal Buruk terhadap Optimasi

Optimasi geometri bekerja dengan mencari arah perubahan koordinat yang menurunkan energi. Jika struktur awal sangat buruk, permukaan energi yang dijelajahi dapat menjadi sulit. Gaya antaratom dapat sangat besar. SCF dapat gagal karena distribusi elektron sulit distabilkan. Optimasi dapat berjalan ke minimum yang tidak diinginkan.

Contoh pada dimer air:

Jika dua molekul H_2O terlalu jauh,
 optimasi mungkin tidak membentuk dimer.

Jika dua molekul H_2O terlalu dekat,
 struktur dapat mengalami tolakan kuat dan optimasi menjadi tidak stabil.

Contoh pada dimer asam asetat:

Jika dua monomer asam asetat tidak diarahkan membentuk dua ikatan hidrogen,
 optimasi dapat menghasilkan struktur dimer yang berbeda atau monomer yang terpisah.

Contoh pada konformer etanol:

Jika struktur awal konformer B terlalu dekat dengan konformer A, setelah optimasi keduanya dapat berakhir pada struktur yang sama.

Jika ini terjadi, perbandingan energi konformer menjadi tidak bermakna karena sebenarnya struktur akhir yang dibandingkan bukan dua konformer berbeda.

5. Cara Menghindari Struktur Awal Buruk

Untuk menghindari struktur awal buruk, mahasiswa dapat melakukan beberapa langkah.

Pertama, gunakan editor molekul seperti Avogadro, GaussView, Molden, atau editor struktur lain untuk membangun geometri awal yang wajar.

Kedua, lakukan optimasi awal dengan medan gaya seperti UFF atau MMFF94 sebelum menjalankan metode kuantum. Optimasi awal ini membantu menghilangkan benturan atom yang terlalu dekat.

Ketiga, periksa panjang ikatan dan sudut ikatan utama. Jangan hanya mengandalkan tampilan visual.

Keempat, untuk dimer atau kompleks, susun orientasi donor dan akseptor ikatan hidrogen secara masuk akal.

Kelima, untuk konformer, pastikan sudut dihedral awal benar-benar berbeda.

Keenam, setelah optimasi selesai, bandingkan struktur akhir dengan struktur awal untuk memastikan tidak berubah menjadi sistem lain.

Pernyataan laporan yang baik:

Struktur awal dibuat dengan editor molekul dan diperiksa secara visual sebelum optimasi. Untuk sistem dimer, orientasi awal disusun agar memungkinkan terbentuknya ikatan hidrogen yang wajar.

6. Kesalahan 2: Charge Salah

Charge atau muatan total sistem adalah salah satu parameter paling penting dalam perhitungan kimia komputasi. Charge menentukan jumlah elektron dalam sistem. Jika charge salah, maka sistem yang dihitung bukan sistem kimia yang dimaksud.

Contoh:

Species	Charge Benar
H ₂ O	0

NH ₃	0
NH ₄ ⁺	+1
Cl ⁻	-1
CH ₃ COOH	0
H ₃ O ⁺	+1

Kesalahan charge dapat terjadi karena mahasiswa lupa mengubah nilai default. Banyak software secara otomatis memberi charge 0. Jika mahasiswa menghitung ion seperti NH₄⁺ tetapi tidak mengubah charge menjadi +1, software akan menghitung spesies netral yang berbeda dari ion amonium.

Contoh kesalahan:

NH₄⁺ dihitung dengan charge 0

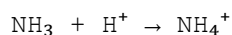
Dampaknya:

jumlah elektron salah
 → energi salah
 → orbital salah
 → momen dipol salah
 → frekuensi salah
 → ΔE, ΔH, dan ΔG salah

Kesalahan charge sangat serius karena memengaruhi semua hasil.

7. Contoh Kesalahan Charge pada Protonasi Amonia

Dalam Praktikum 7, salah satu sistem model adalah protonasi amonia:



Untuk sistem ini:

NH₃ → charge 0
 H⁺ → charge +1
 NH₄⁺ → charge +1

Jika NH₄⁺ dihitung dengan charge 0, maka perhitungan tidak lagi menggambarkan ion amonium. Akibatnya, energi protonasi tidak valid.

Pernyataan yang benar:

NH₄⁺ dihitung dengan charge +1 dan multiplicity 1 karena ion amonium memiliki muatan total positif satu dan seluruh elektronnya berpasangan.

Pernyataan validasi jika ditemukan kesalahan:

Output menunjukkan charge 0 untuk spesies yang dimaksudkan sebagai NH_4^+ . Oleh karena itu, perhitungan tidak valid dan harus diulang dengan charge +1.

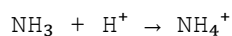
Mahasiswa harus membiasakan diri memeriksa charge pada input dan output.

8. Cara Menghindari Kesalahan Charge

Untuk menghindari kesalahan charge, mahasiswa dapat menggunakan langkah berikut:

1. Tulis rumus kimia dan muatan spesies sebelum membuat input.
2. Hitung jumlah elektron secara kasar jika perlu.
3. Periksa apakah spesies netral, kation, atau anion.
4. Jangan hanya mengikuti nilai default software.
5. Catat charge dalam tabel validasi.
6. Periksa kembali charge di output.
7. Untuk reaksi, pastikan total charge reaktan dan produk konsisten.

Contoh validasi charge pada reaksi:



total charge reaktan = $0 + 1 = +1$

total charge produk = +1

Total charge reaktan dan produk konsisten. Jika tidak konsisten, reaksi yang ditulis perlu diperiksa.

9. Kesalahan 3: Multiplicity Salah

Multiplicity adalah parameter yang menunjukkan keadaan spin sistem. Multiplicity dihitung dengan rumus:

$$\text{Multiplicity} = 2S + 1$$

dengan s adalah total spin. Untuk molekul dengan semua elektron berpasangan, multiplicity biasanya 1 atau singlet. Untuk spesies dengan satu elektron tidak berpasangan, multiplicity biasanya 2 atau doublet. Untuk dua elektron tidak berpasangan dengan spin sejajar, multiplicity dapat 3 atau triplet.

Contoh:

Sistem	Multiplicity Umum
H ₂ O	1
CH ₄	1
NH ₃	1
NH ₄ ⁺	1
Radikal CH ₃ ·	2
O ₂ keadaan dasar	3

Kesalahan multiplicity terjadi ketika mahasiswa memilih keadaan spin yang salah. Kesalahan ini dapat menyebabkan energi, geometri, orbital, dan sifat molekul menjadi tidak sesuai dengan keadaan kimia yang dimaksud.

10. Contoh Kesalahan Multiplicity pada O₂

Contoh klasik adalah molekul oksigen, O₂. Keadaan dasar O₂ adalah triplet. Jika O₂ dihitung sebagai singlet tanpa alasan khusus, maka hasilnya menggambarkan keadaan elektronik berbeda, bukan keadaan dasar O₂.

Contoh kesalahan:

O₂ dihitung dengan charge 0 dan multiplicity 1

Padahal untuk keadaan dasar:

O₂ → charge 0, multiplicity 3

Dampaknya:

energi keadaan elektronik salah
 → orbital salah
 → reaktivitas salah
 → interpretasi tidak sesuai keadaan dasar

Walaupun O₂ tidak selalu menjadi sistem utama dalam praktikum, contoh ini penting untuk menunjukkan bahwa multiplicity bukan parameter teknis kecil. Multiplicity adalah informasi kimia tentang keadaan elektron.

11. Cara Menghindari Kesalahan Multiplicity

Untuk menghindari kesalahan multiplicity, mahasiswa perlu memahami keadaan elektronik spesies.

Langkah praktis:

1. Tentukan apakah spesies memiliki elektron tidak berpasangan.
2. Untuk molekul organik netral biasa, gunakan singlet jika semua elektron berpasangan.
3. Untuk radikal, gunakan doublet jika ada satu elektron tidak berpasangan.
4. Untuk O_2 keadaan dasar, gunakan triplet.
5. Untuk ion logam transisi, periksa kemungkinan spin tinggi dan spin rendah.
6. Catat multiplicity dalam tabel validasi.
7. Jika ragu, konsultasikan literatur atau instruktur.

Pernyataan laporan:

Multiplicity dipilih berdasarkan keadaan spin sistem. Molekul netral dengan semua elektron berpasangan dihitung sebagai singlet, sedangkan spesies radikal memerlukan multiplicity yang sesuai dengan jumlah elektron tidak berpasangan.

12. Kesalahan 4: Basis Set Tidak Sesuai

Basis set adalah kumpulan fungsi matematika yang digunakan untuk merepresentasikan orbital atom. Kesalahan basis set terjadi ketika basis set yang dipilih terlalu kecil, tidak tersedia untuk unsur tertentu, tidak konsisten antarspesies, atau tidak sesuai dengan tujuan perhitungan.

Contoh basis set:

```
STO-3G
3-21G
6-31G
6-31G (d)
6-31G (d, p)
def2-SVP
def2-TZVP
cc-pVDZ
```

Basis set minimal seperti STO-3G dapat digunakan untuk pembelajaran dasar, tetapi biasanya kurang memadai untuk kesimpulan kuantitatif yang presisi. Basis set dengan fungsi polarisasi seperti 6-31G(d) lebih baik untuk menggambarkan perubahan distribusi elektron. Untuk anion atau sistem dengan elektron tersebar luas, fungsi difus dapat diperlukan.

Kesalahan basis set dapat berbentuk:

1. Menggunakan basis set terlalu kecil untuk hasil kuantitatif.
2. Menggunakan basis set berbeda untuk reaktan dan produk.
3. Menggunakan basis set yang tidak mendukung unsur tertentu.
4. Tidak menambahkan fungsi polarisasi pada sistem yang memerlukannya.
5. Tidak menggunakan fungsi difus untuk anion atau interaksi lemah.
6. Tidak mencatat basis set dalam laporan.

13. Dampak Basis Set Tidak Sesuai

Basis set yang tidak sesuai dapat memengaruhi:

1. Energi total.
2. Panjang ikatan.
3. Sudut ikatan.
4. Momen dipol.
5. Frekuensi vibrasi.
6. Energi HOMO-LUMO.
7. Energi interaksi.
8. Energi reaksi.
9. Efek pelarut.
10. Kesimpulan kestabilan.

Contoh:

Dimer air dihitung dengan basis set minimal tanpa fungsi polarisasi.

Dampaknya, energi ikatan hidrogen dapat kurang tergambarkan dengan baik. Interaksi antarmolekul sering memerlukan deskripsi distribusi elektron yang lebih fleksibel.

Contoh lain:

Reaktan dihitung dengan 6-31G(d), produk dihitung dengan STO-3G.

Perbandingan energi tidak valid karena perbedaan energi mencampur efek kimia dan efek basis set.

Pernyataan validasi:

Seluruh spesies dalam satu sistem dihitung dengan basis set yang sama agar perbandingan energi valid secara metodologis.

14. Cara Memilih Basis Set secara Wajar

Pemilihan basis set harus disesuaikan dengan tujuan praktikum.

Untuk pembelajaran sangat dasar, basis set kecil dapat digunakan untuk mempercepat perhitungan. Namun, kesimpulan harus bersifat kualitatif.

Untuk optimasi dan frekuensi molekul organik kecil, basis set seperti 6-31G(d) atau def2-SVP sering menjadi pilihan yang lebih wajar.

Untuk hasil lebih baik, basis set lebih besar seperti def2-TZVP dapat digunakan jika sumber daya komputasi mencukupi.

Untuk anion atau sistem dengan ikatan hidrogen dan interaksi lemah, fungsi difus dan koreksi dispersi dapat menjadi penting pada kajian lanjutan.

Prinsip umum:

basis set kecil
→ cepat, tetapi terbatas

basis set lebih besar
→ lebih baik, tetapi lebih mahal

basis set harus konsisten
→ agar energi dapat dibandingkan

Pernyataan laporan yang baik:

Basis set 6-31G(d) digunakan karena menyediakan fungsi polarisasi yang diperlukan untuk menggambarkan perubahan distribusi elektron pada ikatan kimia.

15. Kesalahan 5: SCF Tidak Konvergen

SCF atau Self-Consistent Field adalah prosedur iteratif untuk memperoleh distribusi elektron dan energi elektronik yang konsisten. SCF tidak konvergen berarti proses iteratif gagal mencapai kriteria konvergensi.

Pesan output dapat berupa:

```
SCF failed to converge
SCF NOT CONVERGED
Maximum SCF cycles exceeded
No convergence in SCF
```

Jika SCF tidak konvergen, energi elektronik tidak dapat dipercaya. Walaupun output mungkin tetap memuat angka energi, angka tersebut belum stabil secara numerik.

Penyebab SCF tidak konvergen dapat meliputi:

1. Struktur awal buruk.
2. Sistem bermuatan sulit.
3. Multiplicity salah.
4. Basis set terlalu besar atau tidak sesuai.
5. Geometri terlalu ekstrem.
6. Sistem memiliki keadaan elektronik dekat degenerat.
7. Tebakan awal orbital buruk.
8. Parameter SCF terlalu ketat atau tidak sesuai.
9. Molekul terlalu besar untuk sumber daya komputasi.
10. Model pelarut memperumit konvergensi.

SCF tidak konvergen harus dianggap sebagai masalah serius.

16. Dampak SCF Tidak Konvergen

Jika SCF tidak konvergen, dampaknya sangat luas:

energi elektronik tidak stabil
 → gradien optimasi tidak dapat dipercaya
 → geometri dapat salah
 → frekuensi dapat salah
 → orbital tidak valid
 → momen dipol tidak valid
 → energi reaksi tidak valid

Dalam optimasi geometri, setiap langkah memerlukan energi dan gradien. Jika SCF bermasalah, optimasi juga dapat bermasalah. Dalam analisis HOMO-LUMO, orbital dari SCF yang tidak konvergen tidak boleh digunakan. Dalam energi reaksi, satu spesies dengan SCF tidak konvergen dapat merusak seluruh perhitungan.

Pernyataan laporan jika SCF gagal:

SCF tidak mencapai konvergensi, sehingga energi elektronik tidak digunakan untuk analisis. Perhitungan perlu diulang dengan perbaikan struktur awal atau pengaturan SCF.

17. Cara Mengatasi SCF Tidak Konvergen

Beberapa cara umum untuk mengatasi SCF tidak konvergen adalah:

1. Memperbaiki struktur awal.
2. Melakukan optimasi awal dengan mekanika molekuler.
3. Menggunakan tebakan awal orbital yang lebih baik.
4. Menambah jumlah siklus SCF.
5. Menggunakan algoritma SCF yang lebih stabil.
6. Menggunakan damping atau level shifting jika tersedia.
7. Mengubah metode sementara untuk mendapatkan struktur awal.
8. Memeriksa charge dan multiplicity.
9. Menggunakan basis set yang lebih sederhana terlebih dahulu.
10. Memecah workflow menjadi tahap bertahap.

Alur perbaikan sederhana:

struktur awal buruk
 → optimasi awal MMFF/UFF
 → optimasi kuantum basis kecil
 → optimasi level teori target

→ frekuensi

Pendekatan bertahap sering lebih stabil daripada langsung menjalankan perhitungan besar dari struktur awal yang kasar.

18. Kesalahan 6: Salah Membaca Satuan Energi

Kesalahan satuan energi adalah kesalahan yang sangat umum. Output kimia kuantum biasanya memberikan energi dalam Hartree atau atomic unit. Namun, dalam kimia termodinamika, energi sering ditafsirkan dalam kJ/mol atau kcal/mol. Jika mahasiswa tidak mengonversi satuan dengan benar, kesimpulan numerik dapat salah besar.

Konversi penting:

$$1 \text{ Hartree} \approx 2625,5 \text{ kJ/mol}$$

atau:

$$1 \text{ Hartree} \approx 627,5 \text{ kcal/mol}$$

Kesalahan contoh:

$$\Delta E = 0,010 \text{ Hartree ditulis sebagai } 0,010 \text{ kJ/mol}$$

Padahal:

$$0,010 \text{ Hartree} \approx 26,255 \text{ kJ/mol}$$

Perbedaan ini sangat besar dalam konteks kimia.

Kesalahan satuan dapat terjadi karena mahasiswa:

1. Tidak mengetahui satuan output.
2. Mengira semua energi dalam kJ/mol.
3. Mengonversi energi absolut setelah pembulatan kasar.
4. Mencampur Hartree, eV, kcal/mol, dan kJ/mol.
5. Tidak menulis satuan dalam tabel.
6. Menggunakan faktor konversi salah.
7. Mengambil energi orbital dalam eV lalu membandingkannya dengan energi total dalam Hartree.

19. Cara Membaca dan Mengonversi Energi dengan Benar

Prinsip yang benar adalah menghitung selisih energi dalam satuan asli dengan digit yang cukup, lalu mengonversinya ke satuan kimia yang lebih mudah dibaca.

Contoh:

```
E_produk = -152,005000 Hartree
E_reaktan = -152,000000 Hartree
```

```
ΔE = -0,005000 Hartree
```

Konversi:

```
ΔE = -0,005000 × 2625,5
ΔE = -13,13 kJ/mol
```

Jangan membulatkan energi absolut terlalu awal. Jika energi absolut dibulatkan menjadi terlalu sedikit digit, selisih energi dapat berubah.

Pernyataan laporan:

```
Energi total diambil dalam Hartree. Selisih energi dihitung terlebih dahulu
dalam Hartree, kemudian dikonversi ke kJ/mol menggunakan faktor 1 Hartree =
2625,5 kJ/mol.
```

Satuan harus selalu ditulis pada tabel dan grafik.

20. Kesalahan 7: Salah Menafsirkan Frekuensi Imajiner

Frekuensi imajiner adalah salah satu bagian output yang sering disalahpahami. Dalam banyak output, frekuensi imajiner ditampilkan sebagai frekuensi negatif. Mahasiswa pemula kadang mengira frekuensi negatif adalah “angka kecil yang boleh diabaikan”, atau sebaliknya menganggap semua frekuensi imajiner selalu berarti perhitungan gagal total. Keduanya tidak selalu tepat.

Makna frekuensi imajiner bergantung pada tujuan perhitungan.

Untuk struktur minimum:

```
tidak boleh ada frekuensi imajiner
```

Jika ada frekuensi imajiner, struktur belum valid sebagai minimum.

Untuk keadaan transisi:

satu frekuensi imajiner diperlukan

Namun, mode imajiner harus sesuai dengan koordinat reaksi yang diharapkan.

Dalam praktikum Bab 4 sampai Bab 10, mayoritas struktur yang dianalisis adalah minimum. Maka, frekuensi imajiner harus dianggap sebagai tanda bahwa struktur perlu diperiksa ulang.

21. Contoh Salah Menafsirkan Frekuensi Imajiner

Contoh salah:

Struktur formaldehida memiliki satu frekuensi negatif, tetapi tetap digunakan sebagai struktur minimum karena hanya satu.

Pernyataan tersebut salah jika formaldehida dimaksudkan sebagai struktur minimum. Satu frekuensi imajiner pada struktur minimum tetap menunjukkan bahwa struktur belum valid.

Contoh benar:

Struktur formaldehida menunjukkan satu frekuensi imajiner. Karena struktur ini dimaksudkan sebagai minimum, hasil belum digunakan untuk analisis energi dan perlu dioptimasi ulang.

Contoh lain:

Keadaan transisi memiliki satu frekuensi imajiner, dan mode vibrasinya sesuai dengan pembentukan atau pemutusan ikatan pada koordinat reaksi.

Pernyataan kedua benar jika memang praktikum membahas keadaan transisi.

22. Cara Memeriksa Frekuensi Imajiner

Untuk memeriksa frekuensi imajiner, mahasiswa harus:

1. Membuka bagian frekuensi dalam output.
2. Melihat apakah ada nilai negatif.
3. Memeriksa jumlah frekuensi negatif.
4. Jika ada, visualisasikan mode vibrasi jika memungkinkan.
5. Tentukan apakah sistem dimaksudkan sebagai minimum atau keadaan transisi.
6. Jika minimum memiliki frekuensi imajiner, optimasi ulang.
7. Jika keadaan transisi memiliki satu frekuensi imajiner, pastikan mode sesuai reaksi.

Untuk struktur minimum:

0 frekuensi imajiner
→ valid

≥ 1 frekuensi imajiner
→ tidak valid sebagai minimum

Untuk keadaan transisi:

1 frekuensi imajiner
→ mungkin valid jika mode benar

> 1 frekuensi imajiner
→ biasanya bukan keadaan transisi sederhana

Dalam laporan Praktikum Kimia Kuantum Bagian 2, sebagian besar struktur adalah minimum, sehingga kriteria yang digunakan adalah tidak ada frekuensi imajiner.

23. Kesalahan Tambahan: Mengambil Energi dari Bagian Output yang Salah

Selain tujuh kesalahan utama, ada kesalahan lain yang sering terjadi: mengambil energi dari bagian output yang salah. Dalam output optimasi, energi dapat muncul berkali-kali pada setiap langkah optimasi. Mahasiswa harus mengambil energi akhir setelah optimasi konvergen.

Kesalahan:

mengambil energi pada langkah optimasi pertama

Padahal energi yang dibutuhkan:

energi pada geometri akhir yang konvergen

Dalam perhitungan frekuensi, mahasiswa juga harus membedakan energi elektronik, energi + ZPE, entalpi, dan Gibbs. Jika ingin menghitung ΔG , nilai yang diambil harus nilai Gibbs, bukan energi elektronik.

Pernyataan laporan:

Energi yang digunakan diambil dari bagian akhir output setelah optimasi konvergen, bukan dari energi iterasi awal.

24. Kesalahan Tambahan: Mencampur Metode dalam Perbandingan Energi

Kesalahan lain yang sangat penting adalah mencampur metode. Misalnya, reaktan dihitung dengan HF/STO-3G, produk dihitung dengan B3LYP/6-31G(d), lalu energinya dibandingkan langsung. Ini tidak valid.

Contoh salah:

```
E_reaktan dari HF/STO-3G
E_produk dari B3LYP/6-31G(d)
ΔE = E_produk - E_reaktan
```

Kesalahan tersebut mencampur perbedaan kimia dan perbedaan metode.

Contoh benar:

```
reaktan dan produk semua dihitung dengan B3LYP/6-31G(d)
```

Pernyataan validasi:

Seluruh spesies yang dibandingkan dihitung dengan level teori yang sama agar perbedaan energi mencerminkan perbedaan kimia, bukan perbedaan metode.

25. Kesalahan Tambahan: Mengabaikan Model Pelarut

Pada praktikum efek pelarut, mahasiswa kadang membandingkan energi fase gas dan pelarut secara tidak konsisten. Misalnya, produk dihitung dalam pelarut, tetapi reaktan dihitung dalam fase gas, lalu hasilnya digunakan sebagai $\Delta G_{\text{pelarut}}$. Ini tidak valid.

Untuk menghitung $\Delta G_{\text{pelarut}}$, semua spesies dalam reaksi harus dihitung dengan model pelarut yang sama.

Contoh protonasi amonia:

```
NH3 pelarut
H+ pelarut
NH4+ pelarut
```

Baru kemudian:

```
ΔG_pelarut = G_NH4+_pelarut - (G_NH3_pelarut + G_H+_pelarut)
```

Jika data gas dan pelarut ingin dibandingkan, keduanya harus dihitung sebagai dua set lengkap.

26. Kesalahan Tambahan: Menganggap Output Grafik Selalu Benar

Dalam praktikum Python, grafik dibuat dari data input. Jika data input salah, grafik tetap dapat terlihat rapi. Hal ini berbahaya karena visualisasi yang menarik dapat memberikan kesan seolah-olah hasil sudah benar.

Prinsip penting:

grafik rapi
 $\neq$ data valid

Sebelum grafik dimasukkan ke laporan, mahasiswa harus memeriksa:

1. Apakah data CSV benar.
2. Apakah satuan benar.
3. Apakah label sumbu benar.
4. Apakah reaktan dan produk tidak tertukar.
5. Apakah nilai energi sesuai tabel.
6. Apakah grafik tidak menyesatkan.

Pernyataan laporan:

Grafik dibuat berdasarkan data CSV yang telah divalidasi dan satuan energi telah dikonversi ke kJ/mol.

27. Diagnosis Kesalahan Berdasarkan Gejala Output

Mahasiswa dapat mendiagnosis kesalahan berdasarkan gejala yang muncul dalam output.

Gejala Output	Kemungkinan Penyebab
SCF tidak konvergen	Struktur buruk, charge salah, multiplicity salah, sistem sulit
Optimasi gagal	Struktur awal buruk, langkah optimasi terbatas, permukaan energi sulit
Frekuensi negatif pada minimum	Struktur belum minimum
Energi sangat tidak wajar	Charge salah, struktur buruk, basis set salah
Momen dipol tidak sesuai simetri	Geometri salah atau struktur tidak sesuai
Dimer terpisah setelah optimasi	Struktur awal dimer kurang baik atau interaksi lemah
HOMO-LUMO tidak masuk akal	SCF bermasalah atau charge/multiplicity salah
Grafik energi terbalik	Peran reaktan-produk tertukar
Nilai kJ/mol terlalu kecil	Hartree belum dikonversi
ΔE dimerisasi sangat salah	Koefisien monomer lupa dikalikan dua

Tabel diagnosis ini membantu mahasiswa memperbaiki masalah secara sistematis.

28. Strategi Umum Memperbaiki Kesalahan

Jika terjadi kesalahan, mahasiswa sebaiknya tidak langsung mengganti banyak hal sekaligus. Perbaikan harus dilakukan bertahap agar penyebab masalah dapat diketahui.

Strategi umum:

```

periksa struktur awal
→ periksa charge
→ periksa multiplicity
→ gunakan metode/basis sederhana
→ jalankan optimasi
→ cek SCF
→ cek geometri
→ jalankan frekuensi
→ cek frekuensi imajiner
→ naikan level teori jika perlu

```

Jika SCF gagal, jangan langsung menyimpulkan software rusak. Periksa input terlebih dahulu. Jika optimasi gagal, periksa struktur awal. Jika energi aneh, periksa satuan dan charge. Jika grafik salah, periksa CSV dan rumus Python.

Prinsipnya:

```

perbaiki dari sumber paling dasar
bukan dari hasil akhir

```

29. Kesalahan sebagai Bagian dari Pembelajaran

Kesalahan dalam praktikum tidak harus selalu dipandang negatif. Dalam pembelajaran kimia komputasi, kesalahan dapat menjadi sarana memahami hubungan antara input, proses, output, dan interpretasi. Mahasiswa yang pernah salah menentukan charge akan lebih memahami pentingnya jumlah elektron. Mahasiswa yang pernah melihat frekuensi imajiner akan lebih memahami permukaan energi potensial. Mahasiswa yang pernah salah mengonversi Hartree akan lebih memahami skala energi kimia.

Namun, kesalahan hanya menjadi pembelajaran jika dianalisis dan diperbaiki. Jika kesalahan diabaikan, maka kesalahan menjadi sumber kesimpulan palsu.

Pernyataan pedagogis:

```

Kesalahan komputasi harus dilacak, dijelaskan, dan diperbaiki. Proses ini
membentuk keterampilan diagnostik yang penting dalam kimia komputasi.

```

30. Tabel Ringkas Kesalahan, Dampak, dan Solusi

Kesalahan	Dampak	Solusi
Struktur awal buruk	Optimasi gagal atau struktur akhir salah	Perbaiki geometri awal, gunakan UFF/MMFF
Charge salah	Jumlah elektron salah	Tentukan muatan spesies sebelum input
Multiplicity salah	Keadaan spin salah	Periksa elektron tidak berpasangan
Basis set tidak sesuai	Energi dan sifat kurang valid	Pilih basis set sesuai tujuan dan konsisten
SCF tidak konvergen	Energi elektronik tidak valid	Perbaiki struktur, parameter SCF, atau metode
Salah satuan energi	Interpretasi numerik salah	Konversi Hartree ke kJ/mol dengan benar
Salah frekuensi imajiner	Struktur salah dianggap valid	Periksa jumlah dan makna frekuensi negatif
Energi diambil dari bagian salah	$\Delta E/\Delta G$ salah	Ambil energi akhir setelah konvergensi
Metode tidak konsisten	Perbandingan energi tidak valid	Gunakan level teori sama
Grafik dari data salah	Visualisasi menyesatkan	Validasi CSV sebelum plotting

Tabel ini dapat digunakan sebagai daftar pemeriksaan sebelum laporan dikumpulkan.

31. Implikasi Keilmuan Kesalahan Umum

Kesalahan umum dalam praktikum kimia komputasi memiliki implikasi keilmuan yang luas. Kesalahan input menunjukkan bahwa model kimia harus didefinisikan dengan tepat. Kesalahan charge dan multiplicity menunjukkan bahwa perhitungan struktur elektronik sangat bergantung pada jumlah elektron dan spin. Kesalahan basis set menunjukkan bahwa akurasi hasil bergantung pada representasi matematika orbital. Kesalahan SCF menunjukkan bahwa solusi numerik tidak selalu mudah diperoleh. Kesalahan satuan menunjukkan bahwa interpretasi kimia memerlukan pemahaman skala energi. Kesalahan frekuensi imajiner menunjukkan bahwa struktur molekul harus divalidasi pada permukaan energi potensial.

Dengan memahami kesalahan-kesalahan ini, mahasiswa belajar bahwa kimia komputasi bukan sekadar teknik menjalankan software, tetapi ilmu yang memerlukan keputusan metodologis dan penalaran kimia.

32. Implikasi Pedagogis Kesalahan Umum

Secara pedagogis, subbab ini melatih mahasiswa untuk menjadi pengguna software yang mandiri. Mahasiswa tidak hanya mengikuti instruksi praktikum, tetapi mulai mampu mengenali masalah, mencari penyebab, dan memperbaiki perhitungan.

Kemampuan yang dilatih meliputi:

1. Mendiagnosis kegagalan SCF.
2. Mengenali struktur awal yang buruk.
3. Memeriksa charge dan multiplicity.
4. Memilih basis set secara rasional.
5. Membaca satuan energi.
6. Menafsirkan frekuensi imajiner.
7. Memvalidasi grafik dan CSV.
8. Menulis catatan kesalahan dalam laporan.
9. Mengulang perhitungan secara bertanggung jawab.
10. Tidak mengambil kesimpulan dari data yang belum valid.

Keterampilan ini penting bagi calon guru kimia. Guru yang memahami kesalahan umum dapat membimbing siswa atau mahasiswa agar tidak hanya tertarik pada visualisasi molekul, tetapi juga memahami proses ilmiah di balik simulasi.

33. Format Catatan Kesalahan dalam Laporan Praktikum

Jika mahasiswa mengalami kesalahan dan memperbaikinya, catatan tersebut dapat ditulis dalam laporan secara ringkas. Contoh format:

Masalah:

SCF tidak konvergen pada perhitungan awal.

Penyebab kemungkinan:

Struktur awal terlalu buruk atau jarak antaratom terlalu dekat.

Tindakan perbaikan:

Struktur awal dioptimasi ulang dengan medan gaya, kemudian perhitungan kuantum dijalankan kembali.

Hasil:

SCF berhasil konvergen dan optimasi geometri selesai normal.

Format seperti ini menunjukkan bahwa mahasiswa memahami proses diagnostik. Kesalahan yang diperbaiki secara ilmiah justru dapat memperkuat laporan karena menunjukkan proses kerja yang nyata.

34. Sintesis Subbab 11.3

Kesalahan umum dalam praktikum kimia komputasi dapat terjadi pada input, proses, output, dan interpretasi. Struktur awal buruk dapat menyebabkan optimasi gagal. Charge salah membuat jumlah elektron salah. Multiplicity salah membuat keadaan spin tidak sesuai. Basis set tidak sesuai dapat menurunkan validitas hasil. SCF tidak konvergen membuat energi elektronik tidak dapat dipercaya. Salah membaca satuan energi dapat menghasilkan kesimpulan numerik keliru. Salah menafsirkan frekuensi imajiner dapat membuat struktur yang belum valid dianggap sebagai minimum.

Alur pencegahan kesalahan dapat diringkas sebagai berikut:

```

periksa struktur awal
→ tetapkan charge dan multiplicity
→ pilih metode dan basis set konsisten
→ jalankan perhitungan
→ cek SCF dan optimasi
→ cek frekuensi imajiner
→ baca satuan dengan benar
→ validasi output
→ baru analisis
  
```

Dengan memahami kesalahan umum, mahasiswa lebih siap melakukan validasi hasil dan membangun workflow komputasi yang rapi. Kesalahan bukan hanya hambatan, tetapi bagian dari proses belajar apabila dilacak, dijelaskan, dan diperbaiki secara ilmiah.

35. Transisi ke Subbab 11.4

Subbab 11.3 telah membahas kesalahan umum dalam praktikum kimia komputasi, meliputi struktur awal buruk, charge salah, multiplicity salah, basis set tidak sesuai, SCF tidak konvergen, salah membaca satuan energi, dan salah menafsirkan frekuensi imajiner. Pembahasan ini menunjukkan bahwa hasil komputasi harus selalu diperiksa dari sisi input, proses, output, dan interpretasi.

Subbab berikutnya akan membahas Python untuk parsing output. Pada subbab 11.4, mahasiswa akan belajar menggunakan Python untuk membaca file log, mencari energi total, mencari momen dipol, mengekstrak frekuensi, membuat tabel otomatis, dan menyimpan ringkasan output ke CSV. Dengan demikian, pemahaman tentang kesalahan umum pada subbab ini menjadi dasar penting agar otomatisasi Python tidak sekadar mengambil angka, tetapi mengambil data yang benar dan tervalidasi.

11.4 Python untuk Parsing Output

Python untuk parsing output adalah penggunaan bahasa pemrograman Python untuk membaca file output komputasi, mencari informasi penting, mengekstrak data numerik, menyusun tabel

ringkasan, dan menyimpan hasil dalam format yang mudah dianalisis, misalnya CSV. Pada subbab 11.1 mahasiswa telah mempelajari cara membaca file output secara manual. Pada subbab 11.2 mahasiswa telah mempelajari validasi hasil praktikum. Pada subbab 11.3 mahasiswa telah mempelajari kesalahan umum yang harus dihindari. Subbab 11.4 melanjutkan ketiga keterampilan tersebut ke tahap otomatisasi.

Parsing output tidak berarti menggantikan pemahaman kimia. Python hanya membantu membaca dan mengumpulkan informasi secara lebih cepat. Keputusan apakah data tersebut valid tetap berada pada pengguna. Dengan kata lain, Python dapat membantu menemukan energi total, momen dipol, frekuensi, HOMO-LUMO, dan status konvergensi, tetapi mahasiswa tetap harus memeriksa apakah struktur masuk akal, charge benar, multiplicity benar, metode konsisten, dan frekuensi imajiner ditafsirkan sesuai konteks.

Dalam subbab ini, Python digunakan untuk:

1. **Membaca file log.**
2. **Mencari energi total.**
3. **Mencari momen dipol.**
4. **Mengekstrak frekuensi.**
5. **Membuat tabel otomatis.**
6. **Menyimpan ringkasan output ke CSV.**

Subbab ini sangat penting karena file output kimia komputasi sering panjang. Satu file output dapat berisi ratusan hingga ribuan baris. Jika mahasiswa harus membaca puluhan output secara manual, risiko salah salin, salah mengambil angka, atau salah memilih bagian output menjadi besar. Python membantu mengurangi risiko tersebut dengan cara membaca pola teks tertentu secara otomatis.

Namun, otomatisasi harus digunakan secara kritis. Parser sederhana yang dibuat dalam praktikum ini tidak boleh dianggap sebagai pengganti parser profesional. Format output Gaussian, ORCA, GAMESS, Psi4, dan PySCF berbeda-beda. Bahkan versi software yang berbeda dapat mengubah tampilan output. Oleh karena itu, kode Python yang disusun dalam subbab ini bersifat pedagogis: cukup kuat untuk membaca pola umum, tetapi tetap perlu dicek ulang secara manual pada beberapa file contoh.

1. Pengertian Parsing Output

Parsing output adalah proses membaca file teks hasil komputasi dan mengambil informasi tertentu berdasarkan pola teks. Dalam kimia komputasi, informasi yang sering diambil dari output adalah energi total, momen dipol, frekuensi vibrasi, intensitas IR, HOMO, LUMO, geometri akhir, status SCF, status optimasi, dan pesan akhir job.

Secara sederhana:

```
file output
→ dibaca sebagai teks
```

- dicari pola penting
- angka diekstrak
- disusun ke tabel
- disimpan sebagai CSV

Contoh pola yang dicari:

```
SCF Done
FINAL SINGLE POINT ENERGY
Total Energy
Dipole moment
Frequencies
HOMO
LUMO
Normal termination
ORCA TERMINATED NORMALLY
```

Jika pola ditemukan, Python dapat mengambil angka yang mengikuti pola tersebut. Jika pola tidak ditemukan, Python dapat memberikan nilai kosong atau catatan bahwa data tidak ditemukan.

Parsing output sangat berguna ketika mahasiswa memiliki banyak file hasil praktikum. Misalnya, dalam Praktikum 7 mahasiswa dapat memiliki output untuk cis-2-butena, trans-2-butena, NH_3 , NH_4^+ , H_2O monomer, dimer air, etanol konformer, dan dimer asam asetat. Jika setiap file dibaca manual, pekerjaan menjadi lambat dan rawan kesalahan. Dengan parsing, ringkasan semua file dapat dibuat secara otomatis.

2. Latar Historis Parsing dalam Kimia Komputasi

Dalam sejarah kimia komputasi, file output teks menjadi pusat dokumentasi hasil perhitungan. Program seperti Gaussian, GAMESS, ORCA, dan paket lain menuliskan hasil dalam format teks panjang. Para peneliti kemudian membaca output tersebut untuk mengambil energi, geometri, frekuensi, orbital, dan data lain.

Pada awalnya, banyak pengambilan data dilakukan secara manual. Namun, ketika jumlah perhitungan meningkat, terutama dalam studi konformer, screening molekul, optimasi banyak struktur, atau analisis energi reaksi, pembacaan manual menjadi tidak efisien. Dari sinilah kebutuhan parsing dan otomatisasi meningkat.

Dalam komputasi modern, parsing output menjadi bagian penting dari workflow. Banyak peneliti menggunakan Python, shell script, atau perangkat khusus untuk mengekstrak data dari output. Beberapa pustaka seperti cclib juga dikembangkan untuk membaca berbagai format output kimia komputasi. Namun, dalam konteks praktikum, mahasiswa perlu memahami prinsip dasarnya terlebih dahulu: bagaimana file dibaca, bagaimana pola ditemukan, bagaimana angka diekstrak, dan bagaimana hasil disimpan.

Dengan memahami parsing sederhana, mahasiswa tidak hanya menjadi pengguna software, tetapi mulai memahami alur kerja komputasi ilmiah yang lebih luas:

```

input
→ perhitungan
→ output
→ parsing
→ tabel data
→ visualisasi
→ laporan

```

Alur ini menjadi dasar otomatisasi laporan praktikum.

3. Prinsip Dasar Parser Output

Parser output bekerja berdasarkan prinsip pencarian pola. File output dibaca sebagai teks, kemudian Python mencari baris tertentu yang mengandung kata kunci. Setelah kata kunci ditemukan, angka yang relevan diambil dengan bantuan pemrosesan string atau ekspresi reguler.

Contoh sederhana:

```
SCF Done: E(RB3LYP) = -76.3456789012 A.U.
```

Dari baris tersebut, Python perlu mengambil angka:

```
-76.3456789012
```

Untuk melakukan itu, Python dapat menggunakan `re`, yaitu modul regular expression.

Contoh pola angka:

```
r"[-+]?[d+\\.\\d+]"
```

Pola tersebut dapat menangkap angka desimal positif atau negatif.

Namun, output software tidak selalu sama. ORCA mungkin menuliskan energi sebagai:

```
FINAL SINGLE POINT ENERGY      -76.3456789012
```

Gaussian mungkin menuliskan:

```
SCF Done:  E(RB3LYP) =  -76.3456789012      A.U.
```

GAMESS dapat menuliskan energi dalam format lain. Karena itu, parser praktikum harus menggunakan beberapa pola pencarian.

4. Informasi yang Akan Diekstrak

Dalam subbab ini, parser dirancang untuk mengambil informasi berikut:

Data	Fungsi
Nama file	Identitas output
Status job	Menentukan apakah job selesai normal
Metode	Mengetahui level teori
Basis set	Mengetahui basis yang digunakan
Charge	Memastikan muatan sistem
Multiplicity	Memastikan keadaan spin
Energi total	Dasar analisis energi
Momen dipol	Analisis polaritas
Frekuensi	Analisis vibrasi dan validasi minimum
Jumlah frekuensi imajiner	Validasi struktur
Frekuensi minimum	Deteksi mode bermasalah
HOMO	Analisis orbital terisi tertinggi
LUMO	Analisis orbital kosong terendah
Gap HOMO-LUMO	Analisis sifat elektronik
Catatan	Informasi tambahan jika data tidak ditemukan

Tidak semua output akan memuat semua data. Misalnya, file single point energy mungkin tidak memuat frekuensi. File optimasi biasa mungkin tidak memuat termokimia. File dari satu software mungkin tidak menuliskan HOMO-LUMO dengan label yang sama. Oleh karena itu, parser harus memberi nilai kosong jika data tidak ditemukan.

5. Struktur Folder Parsing Output

Struktur folder yang dianjurkan:

```
praktikum_analisis_output/
├── output_log/
│   ├── h2o.log
│   ├── co2.log
│   ├── nh3.log
│   ├── nh4_plus.log
│   └── dimer_air.log
├── kode/
│   └── parsing_output_komputasi.py
└── ringkasan/
    └── ringkasan_output_komputasi.csv
```

Folder `output_log` berisi file output dari software komputasi. Folder `kode` berisi skrip Python. Folder `ringkasan` berisi hasil parsing dalam bentuk CSV.

Pemisahan folder seperti ini mendukung reproduibilitas. Mahasiswa dapat menunjukkan bahwa file CSV berasal dari output tertentu dan dibuat oleh kode tertentu.

6. Nama File Kode yang Dianjurkan

Nama file kode untuk subbab ini dapat ditulis:

```
parsing_output_komputasi.py
```

Nama ini jelas karena menunjukkan bahwa file digunakan untuk parsing output komputasi. Jika ingin mengikuti struktur praktikum Bab 11, nama yang lebih spesifik dapat digunakan:

```
praktikum_08_parsing_output_komputasi.py
```

Namun, karena Bab 11 bersifat lintas praktikum, nama `parsing_output_komputasi.py` sudah cukup baik.

Nama file sebaiknya menggunakan huruf kecil dan garis bawah agar mudah dijalankan di terminal.

7. Kode Python Utama untuk Parsing Output

Kode berikut dirancang untuk membaca beberapa file output dalam folder `output_log`, mencari energi total, momen dipol, frekuensi, HOMO-LUMO, status normal, charge, multiplicity, metode, basis set, lalu menyimpan ringkasan ke CSV.

Kode ini bersifat pedagogis dan menggunakan pola umum dari beberapa software. Jika format output berbeda, pola regex dapat ditambahkan.

```
from pathlib import Path
import re
import pandas as pd

# =====
# BAB 11 - ANALISIS OUTPUT, VALIDASI DATA,
# DAN OTOMATISASI LAPORAN DENGAN PYTHON
# -----
# Nama file:
# parsing_output_komputasi.py
#
# Fungsi:
# 1. Membaca file log/output komputasi
```

```

# 2. Mencari energi total
# 3. Mencari momen dipol
# 4. Mengekstrak frekuensi
# 5. Mencari HOMO-LUMO secara sederhana
# 6. Membuat tabel otomatis
# 7. Menyimpan ringkasan output ke CSV
# =====

# -----
# 1. Folder kerja
# -----

FOLDER_LOG = Path("output_log")
FOLDER_RINGKASAN = Path("ringkasan")
FOLDER_RINGKASAN.mkdir(exist_ok=True)

FILE_CSV = FOLDER_RINGKASAN / "ringkasan_output_komputasi.csv"

# -----
# 2. Fungsi pembantu umum
# -----

def baca_teks(path_file):
    """
    Membaca file output sebagai teks.
    errors='ignore' digunakan agar file tetap dapat dibaca
    meskipun ada karakter yang tidak dikenali.
    """
    return path_file.read_text(encoding="utf-8", errors="ignore")

def ambil_angka_pertama(teks):
    """
    Mengambil angka desimal pertama dari sebuah teks.
    Mengembalikan None jika tidak ditemukan angka.
    """
    pola = r"[-+]?[d+\\.d+](?:[Ee] [-+]?[d+])?"
    cocok = re.search(pola, teks)
    if cocok:
        return float(cocok.group(0))
    return None

def cari_terakhir(pola, teks, flags=re.IGNORECASE):
    """
    Mencari semua kecocokan regex dan mengambil kecocokan terakhir.
    Berguna karena energi atau data lain dapat muncul berkali-kali.
    """
    hasil = re.findall(pola, teks, flags)
    if hasil:
        return hasil[-1]
    return None

# -----

```



```

# 3. Mencari status job selesai normal
# -----

def cari_status_normal(teks):
    """
    Mencari tanda bahwa job selesai normal.
    Pola dibuat umum untuk beberapa software.
    """
    pola_normal = [
        "Normal termination",
        "ORCA TERMINATED NORMALLY",
        "EXECUTION OF GAMESS TERMINATED NORMALLY",
        "Job finished successfully",
        "Psi4 exiting successfully",
    ]

    for pola in pola_normal:
        if pola.lower() in teks.lower():
            return "Selesai normal"

    pola_error = [
        "Error termination",
        "SCF failed",
        "SCF NOT CONVERGED",
        "Optimization failed",
        "Maximum number of cycles exceeded",
        "abnormally",
    ]

    for pola in pola_error:
        if pola.lower() in teks.lower():
            return "Ada indikasi masalah"

    return "Tidak terdeteksi"

# -----
# 4. Mencari metode dan basis set
# -----

def cari_metode_basis(teks):
    """
    Mencari metode dan basis set secara sederhana.
    Parser ini membaca pola umum seperti:
    B3LYP/6-31G(d)
    HF/STO-3G
    MP2/6-31G(d)
    atau baris ORCA: ! B3LYP 6-31G(d) Opt Freq
    """

    metode = None
    basis = None

    # Pola Gaussian umum: B3LYP/6-31G(d)
    pola_gaussian = r"\b(HF|B3LYP|PBE0|PBE|M06-2X|M06|MP2|CAM-  
B3LYP)\s*/\s*([A-Za-z0-9\-\+\*\(\)\, ]+)"
    cocok = re.search(pola_gaussian, teks, re.IGNORECASE)

```

```

if cocok:
    metode = cocok.group(1)
    basis = cocok.group(2)
    return metode, basis

# Pola ORCA sederhana: ! B3LYP 6-31G(d) Opt Freq
pola_orca = r"!\\s*(HF|B3LYP|PBE0|PBE|M06-2X|M06|MP2|CAM-B3LYP)\\s+([A-Za-z0-9\\-\\+\\*\\(\\),\\+])"
cocok = re.search(pola_orca, teks, re.IGNORECASE)
if cocok:
    metode = cocok.group(1)
    basis = cocok.group(2)
    return metode, basis

# Pola fallback
if re.search(r"\\bB3LYP\\b", teks, re.IGNORECASE):
    metode = "B3LYP"
elif re.search(r"\\bHF\\b", teks, re.IGNORECASE):
    metode = "HF"
elif re.search(r"\\bMP2\\b", teks, re.IGNORECASE):
    metode = "MP2"

basis_list = [
    "STO-3G", "3-21G", "6-31G", "6-31G(d)", "6-31G*", "6-31G(d,p)",
    "def2-SVP", "def2-TZVP", "cc-pVDZ", "cc-pVTZ"
]

for b in basis_list:
    if b.lower() in teks.lower():
        basis = b
        break

return metode, basis

# -----
# 5. Mencari charge dan multiplicity
# -----

def cari_charge_multiplicity(teks):
    """
    Mencari charge dan multiplicity secara sederhana.
    Format umum:
    Charge = 0 Multiplicity = 1
    atau
    Total Charge          Charge .... 0
    """

    charge = None
    multiplicity = None

    pola1 = r"Charge\\s*=\\s*([\\-+]?\\d+)\\s+Multiplicity\\s*=\\s*(\\d+)"
    cocok = re.search(pola1, teks, re.IGNORECASE)
    if cocok:
        charge = int(cocok.group(1))
        multiplicity = int(cocok.group(2))
        return charge, multiplicity

```

```

pola2 = r"Total\s+Charge\s*[:=]?\s*([-+]?\d+)"
cocok = re.search(pola2, teks, re.IGNORECASE)
if cocok:
    charge = int(cocok.group(1))

pola3 = r"Multiplicity\s*[:=]?\s*(\d+)"
cocok = re.search(pola3, teks, re.IGNORECASE)
if cocok:
    multiplicity = int(cocok.group(1))

return charge, multiplicity

# -----
# 6. Mencari energi total
# -----

def cari_energi_total(teks):
    """
    Mencari energi total akhir dari beberapa format umum.
    Energi dikembalikan dalam Hartree jika output menggunakan Hartree.
    """

    pola_list = [
        r"SCF Done:\s+E\([^\)]+\)\s*=\s*([-+]?\d+\.\d+(:[Ee]([-+]?\d+)?)",
        r"FINAL SINGLE POINT ENERGY\s+([-+]?\d+\.\d+(:[Ee]([-+]?\d+)?)",
        r"Total Energy\s*[:=]\s*([-+]?\d+\.\d+(:[Ee]([-+]?\d+)?)",
        r"Total SCF energy\s*=\s*([-+]?\d+\.\d+(:[Ee]([-+]?\d+)?)",
        r"E\ (TOTAL\)\s*=\s*([-+]?\d+\.\d+(:[Ee]([-+]?\d+)?)",
    ]

    for pola in pola_list:
        hasil = re.findall(pola, teks, re.IGNORECASE)
        if hasil:
            return float(hasil[-1])

    return None

# -----
# 7. Mencari momen dipol
# -----

def cari_momen_dipol(teks):
    """
    Mencari momen dipol total dalam Debye.
    Output software dapat berbeda.
    Fungsi ini mencari beberapa pola umum.
    """

    # Pola Gaussian: Tot= 1.2345
    pola_gaussian = r"Tot=\s*([-+]?\d+\.\d+(:[Ee]([-+]?\d+)?) "
    hasil = re.findall(pola_gaussian, teks, re.IGNORECASE)
    if hasil:
        return float(hasil[-1])

```

```

# Pola umum: Total Dipole Moment : 1.2345
pola_total = r"Total\s+Dipole\s+Moment\s*[:=]?s*([-+]?\d+\.\d+(?:[Ee] [-+]? \d+)?)"
hasil = re.findall(pola_total, teks, re.IGNORECASE)
if hasil:
    return float(hasil[-1])

# Pola ORCA tertentu dapat memuat "Magnitude (Debye)"
pola_magnitude = r"Magnitude\s* \(Debye\) \s*[:=]?s*([-+]?\d+\.\d+(?:[Ee] [-+]? \d+)?)"
hasil = re.findall(pola_magnitude, teks, re.IGNORECASE)
if hasil:
    return float(hasil[-1])

return None

# -----
# 8. Mengekstrak frekuensi vibrasi
# -----

def cari_frekuensi(teks):
    """
    Mengekstrak daftar frekuensi vibrasi.
    Frekuensi biasanya dalam cm-1.
    Fungsi ini mencari pola Gaussian dan pola umum.
    """

    frekuensi = []

    # Gaussian: Frequencies -- 100.0 200.0 300.0
    pola_gaussian = r"Frequencies\s+--\s+(.*)"
    baris_frekuensi = re.findall(pola_gaussian, teks, re.IGNORECASE)

    for baris in baris_frekuensi:
        angka = re.findall(r"[-+]?\d+\.\d+(?:[Ee] [-+]? \d+)?", baris)
        frekuensi.extend([float(a) for a in angka])

    # ORCA/GAMESS-like fallback: baris mengandung cm-1 atau cm-1
    if not frekuensi:
        for baris in teks.splitlines():
            if "cm-1" in baris.lower() or "cm-1" in baris.lower():
                angka = re.findall(r"[-+]?\d+\.\d+(?:[Ee] [-+]? \d+)?", baris)
                frekuensi.extend([float(a) for a in angka])

    return frekuensi

def ringkas_frekuensi(frekuensi):
    """
    Membuat ringkasan frekuensi:
    jumlah frekuensi,
    jumlah frekuensi imajiner,
    frekuensi minimum,
    frekuensi maksimum.
    Dalam banyak output, frekuensi imajiner tampil sebagai angka negatif.
    """

```

```

if not frekuensi:
    return {
        "jumlah_frekuensi": 0,
        "jumlah_frekuensi_imajiner": None,
        "frekuensi_min_cm-1": None,
        "frekuensi_max_cm-1": None,
    }

jumlah_imajiner = sum(1 for f in frekuensi if f < 0)

return {
    "jumlah_frekuensi": len(frekuensi),
    "jumlah_frekuensi_imajiner": jumlah_imajiner,
    "frekuensi_min_cm-1": min(frekuensi),
    "frekuensi_max_cm-1": max(frekuensi),
}

# -----
# 9. Mencari HOMO-LUMO secara sederhana
# -----

def cari_homo_lumo(teks):
    """
    Mencari HOMO dan LUMO secara sederhana.
    Karena format output berbeda-beda, fungsi ini memakai pendekatan umum.

    Catatan:
    Untuk analisis profesional, bagian orbital perlu diperiksa manual.
    Fungsi ini hanya contoh pedagogis.
    """

    homo = None
    lumo = None

    # Pola ORCA umum:
    # NO   OCC           E(Eh)           E(eV)
    # atau baris orbital dengan okupansi.
    # Parser sederhana mencari baris yang mengandung angka okupansi 2.0000 dan
    0.0000.
    orbital_occupied = []
    orbital_virtual = []

    for baris in teks.splitlines():
        bagian = baris.split()

        if len(bagian) >= 4:
            try:
                # Coba baca kolom okupansi dan energi Hartree/eV
                angka = [float(x) for x in bagian if re.match(r"[-+]?[0-9]+\.[0-9]+(?:[Ee][-+]?[0-9]+)?$", x)]
                if len(angka) >= 2:
                    # Heuristik: angka pertama bisa okupansi, angka berikutnya
                    energi

                    occ = angka[0]
                    energi = angka[1]

```

```

        if occ > 0.5:
            orbital_occupied.append(energi)
        elif abs(occ) < 0.1:
            orbital_virtual.append(energi)
    except ValueError:
        pass

    if orbital_occupied:
        homo = orbital_occupied[-1]
    if orbital_virtual:
        lumo = orbital_virtual[0]

    # Pola fallback: jika ada label HOMO dan LUMO eksplisit
    pola_homo = r"HOMO\s*[:=]?s*([-+]?d+\.\d+([Ee]([-+]?d+)?)"
    hasil = re.findall(pola_homo, teks, re.IGNORECASE)
    if hasil:
        homo = float(hasil[-1])

    pola_lumo = r"LUMO\s*[:=]?s*([-+]?d+\.\d+([Ee]([-+]?d+)?)"
    hasil = re.findall(pola_lumo, teks, re.IGNORECASE)
    if hasil:
        lumo = float(hasil[-1])

    gap = None
    if homo is not None and lumo is not None:
        gap = lumo - homo

    return homo, lumo, gap

# -----
# 10. Parsing satu file output
# -----

def parsing_satu_file(path_file):
    """
    Membaca satu file output dan mengembalikan ringkasan data.
    """

    teks = baca_teks(path_file)

    status = cari_status_normal(teks)
    metode, basis = cari_metode_basis(teks)
    charge, multiplicity = cari_charge_multiplicity(teks)
    energi = cari_energi_total(teks)
    dipol = cari_momen_dipol(teks)
    frekuensi = cari_frekuensi(teks)
    ringkas_frekuensi = ringkas_frekuensi(frekuensi)
    homo, lumo, gap = cari_homo_lumo(teks)

    catatan = []

    if energi is None:
        catatan.append("Energi tidak ditemukan")
    if dipol is None:
        catatan.append("Momen dipol tidak ditemukan")

```

```

if not frekuensi:
    catatan.append("Frekuensi tidak ditemukan")
if status != "Selesai normal":
    catatan.append("Status normal tidak terkonfirmasi")
if charge is None:
    catatan.append("Charge tidak ditemukan")
if multiplicity is None:
    catatan.append("Multiplicity tidak ditemukan")

return {
    "nama_file": path_file.name,
    "status_job": status,
    "metode": metode,
    "basis_set": basis,
    "charge": charge,
    "multiplicity": multiplicity,
    "energi_total_Hartree": energi,
    "momen_dipol_Debye": dipol,
    "HOMO": homo,
    "LUMO": lumo,
    "gap_HOMO_LUMO": gap,
    **ringkas_frekuensi,
    "catatan": "; ".join(catatan) if catatan else "OK",
}

# -----
# 11. Parsing semua file output dalam folder
# -----

def parsing_semua_file():
    """
    Membaca semua file .log, .out, dan .txt dalam folder output_log.
    """

    if not FOLDER_LOG.exists():
        raise FileNotFoundError(
            f"Folder {FOLDER_LOG} tidak ditemukan. "
            "Buat folder output_log dan masukkan file output komputasi."
        )

    daftar_file = []
    daftar_file.extend(FOLDER_LOG.glob("*.log"))
    daftar_file.extend(FOLDER_LOG.glob("*.out"))
    daftar_file.extend(FOLDER_LOG.glob("*.txt"))

    if not daftar_file:
        raise FileNotFoundError(
            "Tidak ada file .log, .out, atau .txt dalam folder output_log."
        )

    hasil = []

    for path_file in sorted(daftar_file):
        ringkasan = parsing_satu_file(path_file)
        hasil.append(ringkasan)

```

```

df = pd.DataFrame(hasil)
df.to_csv(FILE_CSV, index=False)

print("Ringkasan output berhasil dibuat:")
print(FILE_CSV)
print()
print(df.to_string(index=False))

# -----
# 12. Menjalankan program
# -----

if __name__ == "__main__":
    parsing_semua_file()

```

Kode tersebut akan membaca file `.log`, `.out`, dan `.txt` dalam folder `output_log`, lalu menghasilkan file:

`ringkasan/ringkasan_output_komputasi.csv`

File CSV tersebut dapat dibuka dengan spreadsheet, dibaca kembali dengan Python, atau dimasukkan ke laporan praktikum.

8. Penjelasan Bagian Pembacaan File Log

Bagian pertama kode menggunakan `Path` dari pustaka `pathlib`:

```

FOLDER_LOG = Path("output_log")
FOLDER_RINGKASAN = Path("ringkasan")

```

`Path` digunakan agar pengelolaan folder lebih rapi dan mudah dipindahkan antar komputer. Kode ini mengharapkan file output berada dalam folder `output_log`.

Fungsi:

```

def baca_teks(path_file):

```

digunakan untuk membaca file sebagai teks. Parameter `errors="ignore"` membantu agar file tetap dapat dibaca walaupun ada karakter tertentu yang tidak dikenali.

Dalam praktikum, mahasiswa perlu memastikan bahwa file output benar-benar berada dalam folder yang tepat. Jika file tidak ditemukan, Python akan menampilkan pesan error. Pesan ini bukan kegagalan konsep, tetapi tanda bahwa struktur folder belum sesuai.

9. Penjelasan Pencarian Status Job

Fungsi:

```
def cari_status_normal(teks):
```

mencari apakah output menunjukkan job selesai normal. Fungsi ini memeriksa beberapa pola seperti:

```
Normal termination
ORCA TERMINATED NORMALLY
EXECUTION OF GAMESS TERMINATED NORMALLY
```

Jika pola ditemukan, status ditulis sebagai:

```
Selesai normal
```

Jika ditemukan indikasi error seperti:

```
Error termination
SCF failed
Optimization failed
```

status ditulis sebagai:

```
Ada indikasi masalah
```

Jika tidak ditemukan pola, status menjadi:

```
Tidak terdeteksi
```

Mahasiswa harus memahami bahwa status ini adalah validasi awal. Jika status tidak normal, data tidak boleh langsung digunakan tanpa pemeriksaan manual.

10. Penjelasan Pencarian Metode dan Basis Set

Fungsi:

```
def cari_metode_basis(teks):
```

mencoba mencari metode dan basis set dari output. Contoh yang dapat dikenali:

```
B3LYP/6-31G(d)
HF/STO-3G
MP2/6-31G(d)
! B3LYP 6-31G(d) Opt Freq
```

Jika pola ditemukan, kode menyimpan metode dan basis set dalam kolom:

```
metode
basis_set
```

Metode dan basis set penting untuk reproduibilitas. Jika parser tidak menemukan metode atau basis set, mahasiswa harus memeriksanya secara manual.

Penting dipahami bahwa format output sangat bervariasi. Parser sederhana mungkin gagal membaca beberapa format. Karena itu, hasil parsing metode dan basis set harus dicek ulang pada beberapa file contoh.

11. Penjelasan Pencarian Charge dan Multiplicity

Fungsi:

```
def cari_charge_multiplicity(teks):
```

mencari informasi charge dan multiplicity. Format yang paling umum adalah:

```
Charge = 0 Multiplicity = 1
```

atau bentuk lain yang memuat Total Charge dan Multiplicity.

Charge dan multiplicity sangat penting karena menentukan sistem elektronik. Jika parser tidak menemukan charge dan multiplicity, hasil CSV akan memuat nilai kosong. Nilai kosong tersebut harus dianggap sebagai tanda bahwa mahasiswa perlu memeriksa output secara manual.

Parser tidak dapat menebak charge dan multiplicity dari rumus molekul secara otomatis. Pengguna harus tetap mencatatnya dan memvalidasinya.

12. Penjelasan Pencarian Energi Total

Fungsi:

```
def cari_energi_total(teks):
```

mencari energi total akhir dari output. Kode menggunakan beberapa pola umum, antara lain:

```
SCF Done
FINAL SINGLE POINT ENERGY
Total Energy
Total SCF energy
E(TOTAL)
```

Karena energi dapat muncul berkali-kali, kode mengambil hasil terakhir:

```
return float(hasil[-1])
```

Hal ini masuk akal karena dalam banyak output, energi akhir muncul di bagian akhir. Namun, mahasiswa tetap harus berhati-hati. Pada beberapa output, ada energi dari tahap berbeda, misalnya energi optimasi, energi frekuensi, atau energi single point tambahan. Karena itu, energi hasil parser perlu dicek pada beberapa file.

Pernyataan kritis:

Parser mengambil energi berdasarkan pola teks. Jika output memiliki beberapa jenis energi, mahasiswa harus memastikan bahwa energi yang diambil adalah energi yang sesuai dengan tujuan analisis.

13. Penjelasan Pencarian Momen Dipol

Fungsi:

```
def cari_momen_dipol(teks):
```

mencari momen dipol total dalam Debye. Beberapa output menuliskan momen dipol dengan label:

```
Tot=
Total Dipole Moment
Magnitude (Debye)
```

Nilai momen dipol penting untuk analisis polaritas molekul. Molekul seperti H₂O memiliki momen dipol tidak nol, sedangkan CO₂ linear memiliki momen dipol total mendekati nol.

Jika parser tidak menemukan momen dipol, ada beberapa kemungkinan:

1. Perhitungan tidak meminta analisis momen dipol.
2. Format output berbeda.
3. Pola regex belum sesuai.
4. File output tidak lengkap.

Mahasiswa harus memeriksa output manual sebelum menyimpulkan bahwa molekul tidak memiliki momen dipol.

14. Penjelasan Ekstraksi Frekuensi

Fungsi:

```
def cari_frekuensi(teks):
```

mencari frekuensi vibrasi. Pada output Gaussian, frekuensi sering muncul sebagai:

```
Frequencies -- 3650.2 1600.5 1200.7
```

Kode mengambil semua angka pada baris tersebut dan menyimpannya sebagai daftar frekuensi.

Fungsi:

```
def ringkas_frekuensi(frekuensi):
```

kemudian menghitung:

1. Jumlah frekuensi.
2. Jumlah frekuensi imajiner.
3. Frekuensi minimum.
4. Frekuensi maksimum.

Frekuensi imajiner dihitung sebagai frekuensi bernilai negatif:

```
jumlah_imajiner = sum(1 for f in frekuensi if f < 0)
```

Untuk struktur minimum, jumlah frekuensi imajiner harus nol. Jika jumlah frekuensi imajiner lebih dari nol, CSV akan menunjukkan bahwa struktur perlu diperiksa.

15. Penjelasan Pencarian HOMO-LUMO

Fungsi:

```
def cari_homo_lumo(teks):
```

mencari HOMO dan LUMO secara sederhana. Bagian ini paling sulit karena format orbital dalam output sangat bervariasi. Gaussian, ORCA, GAMESS, Psi4, dan PySCF dapat menampilkan orbital dengan gaya berbeda.

Kode mencoba membaca baris yang memiliki okupansi orbital dan energi. Jika okupansi lebih besar dari 0, orbital dianggap terisi. Jika okupansi mendekati 0, orbital dianggap virtual. Orbital terisi terakhir dianggap HOMO, sedangkan orbital virtual pertama dianggap LUMO.

Namun, bagian ini harus dipahami sebagai contoh pedagogis, bukan parser universal. Untuk analisis HOMO-LUMO yang serius, mahasiswa harus memeriksa bagian orbital secara manual atau menggunakan pustaka yang lebih khusus.

Pernyataan kritis:

HOMO-LUMO hasil parsing otomatis harus divalidasi manual karena format orbital berbeda antarsoftware dan dapat dipengaruhi oleh spin, okupansi, serta jenis perhitungan.

16. Output CSV yang Dihasilkan

Kode menghasilkan CSV dengan kolom seperti berikut:

Kolom	Makna
nama_file	Nama file output
status_job	Status job selesai normal atau bermasalah
metode	Metode perhitungan
basis_set	Basis set
charge	Muatan total
multiplicity	Keadaan spin
energi_total_Hartree	Energi total
momen_dipol_Debye	Momen dipol total
HOMO	Energi HOMO jika ditemukan
LUMO	Energi LUMO jika ditemukan
gap_HOMO_LUMO	Selisih LUMO-HOMO
jumlah_frekuensi	Jumlah frekuensi yang ditemukan
jumlah_frekuensi_imajiner	Jumlah frekuensi negatif
frekuensi_min_cm-1	Frekuensi minimum
frekuensi_max_cm-1	Frekuensi maksimum
catatan	Catatan hasil parsing

CSV ini dapat digunakan sebagai tabel ringkasan output praktikum.

17. Contoh Hasil CSV

Contoh hasil CSV:

nama_file	status_job	metode	basis_set	charge	multiplicity	energi_total_Hartree	momen_dipol_Debye	jumlah_frekuensi_imajiner	catatan
h2o.log	Selesai normal	B3LYP	6-31G(d)	0	1	-76.3456	1.85	0	OK
co2.log	Selesai normal	B3LYP	6-31G(d)	0	1	-188.5000	0.00	0	OK

nh4_plus.log	Selesai normal	B3 LYP	6-31G (d)	1	1	-56.3000	0.00	0	OK
dimer_air.log	Ada indikasi masalah	B3 LYP	6-31G (d)	0	1	-152.0100	2.50	1	Status normal tidak terkonfirmasi

Dari tabel tersebut, mahasiswa dapat segera melihat bahwa `dimer_air.log` perlu diperiksa ulang karena statusnya bermasalah dan memiliki satu frekuensi imajiner.

18. Validasi Hasil Parsing

Hasil parsing harus divalidasi. Jangan langsung memasukkan CSV ke laporan tanpa memeriksa beberapa file secara manual. Validasi hasil parsing dapat dilakukan dengan cara berikut:

1. Pilih beberapa file output contoh.
2. Buka file output secara manual.
3. Cari energi total.
4. Bandingkan dengan nilai di CSV.
5. Cari momen dipol.
6. Bandingkan dengan nilai di CSV.
7. Cari frekuensi.
8. Pastikan jumlah frekuensi imajiner sesuai.
9. Periksa metode dan basis set.
10. Periksa charge dan multiplicity.

Jika hasil parser cocok dengan pembacaan manual, parser dapat digunakan untuk file lain dengan format serupa. Jika tidak cocok, pola regex harus diperbaiki.

Prinsip penting:

parsing otomatis
→ harus diuji dengan pembacaan manual

19. Analisis Kritis terhadap Parsing Otomatis

Parsing otomatis memiliki kelebihan dan keterbatasan. Kelebihannya adalah cepat, konsisten, dan mengurangi kesalahan salin. Namun, keterbatasannya adalah bergantung pada format teks. Jika format output berubah, parser dapat gagal atau mengambil angka yang salah.

Risiko parsing otomatis:

1. Mengambil energi dari bagian yang salah.
2. Tidak menemukan metode karena format berbeda.
3. Salah membaca HOMO-LUMO.
4. Tidak mendeteksi status error tertentu.
5. Menganggap frekuensi tidak ada karena format berbeda.
6. Membaca angka yang bukan data utama.
7. Menghasilkan CSV rapi dari data yang belum valid.

Pernyataan kritis:

Parser otomatis membantu mempercepat pengambilan data, tetapi hasil parsing tetap harus divalidasi karena format output berbeda antarsoftware dan antarkasus perhitungan.

Mahasiswa harus memahami bahwa otomatisasi bukan pengganti penalaran.

20. Membaca File Log Banyak Sistem

Parsing sangat berguna untuk banyak sistem. Misalnya, mahasiswa memiliki output:

```
h2o.log
co2.log
nh3.log
formaldehida.log
etanol_konformer_A.log
etanol_konformer_B.log
dimer_air.log
asam_asetat_dimer.log
```

Dengan kode parsing, semua file dapat diringkas ke satu CSV. Setelah itu, mahasiswa dapat memilih data untuk laporan:

1. Energi total untuk perbandingan energi.
2. Momen dipol untuk analisis polaritas.
3. Frekuensi untuk validasi minimum.
4. HOMO-LUMO untuk analisis elektronik.
5. Status job untuk validasi.

Tanpa parsing, mahasiswa harus membuka setiap file satu per satu. Parsing membuat pekerjaan lebih sistematis.

21. Menghubungkan Parsing dengan Validasi Praktikum

Parsing output harus mendukung validasi praktikum. CSV hasil parsing dapat digunakan untuk membuat kolom validasi otomatis sederhana.

Contoh logika validasi:

```
status_job = Selesai normal
jumlah_frekuensi_imajiner = 0
energi_total_Hartree tidak kosong
charge tidak kosong
multiplicity tidak kosong
→ data sementara dianggap layak diperiksa lanjut
```

Namun, validasi geometri masuk akal tetap memerlukan pemeriksaan visual atau analisis struktur.

Mahasiswa dapat menambahkan kolom validasi sederhana dengan Python:

```
def status_validasi_awal(baris):
    if baris["status_job"] != "Selesai normal":
        return "Periksa: job tidak normal"
    if pd.isna(baris["energi_total_Hartree"]):
        return "Periksa: energi tidak ditemukan"
    if baris["jumlah_frekuensi_imajiner"] not in [0, None]:
        return "Periksa: ada frekuensi imajiner"
    if pd.isna(baris["charge"]) or pd.isna(baris["multiplicity"]):
        return "Periksa: charge/multiplicity tidak terbaca"
    return "Validasi awal OK"
```

Kolom ini membantu mahasiswa menyeleksi output mana yang perlu diperiksa ulang.

22. Kode Tambahan untuk Validasi Awal Otomatis

Kode berikut dapat ditambahkan setelah DataFrame terbentuk:

```
def status_validasi_awal(baris):
    """
    Memberi status validasi awal berdasarkan hasil parsing.
    Validasi ini bukan pengganti pemeriksaan manual.
    """

    if baris["status_job"] != "Selesai normal":
        return "Periksa job"

    if pd.isna(baris["energi_total_Hartree"]):
        return "Periksa energi"

    if pd.isna(baris["charge"]) or pd.isna(baris["multiplicity"]):
        return "Periksa charge/multiplicity"
```



```

jumlah_imajiner = baris["jumlah_frekuensi_imajiner"]

if pd.notna(jumlah_imajiner) and jumlah_imajiner > 0:
    return "Periksa frekuensi imajiner"

return "Validasi awal OK"

```

Lalu tambahkan sebelum menyimpan CSV:

```
df["validasi_awal"] = df.apply(status_validasi_awal, axis=1)
```

Dengan tambahan ini, CSV memiliki kolom `validasi_awal`.

Contoh:

nama_file	status_job	jumlah_frekuensi_imajiner	validasi_awal
h2o.log	Selesai normal	0	Validasi awal OK
dimer_air.log	Selesai normal	1	Periksa frekuensi imajiner
nh4_plus.log	Tidak terdeteksi	0	Periksa job

Kolom ini membantu mahasiswa memprioritaskan pengecekan.

23. Parsing Output Gaussian, ORCA, dan GAMESS

Format output berbeda antarsoftware. Karena itu, parser harus mengenali beberapa pola.

Contoh Gaussian:

```

SCF Done: E(RB3LYP) = -76.3456789012 A.U.
Frequencies -- 3650.2 1600.5 1200.7
Dipole moment (field-independent basis, Debye):
Tot= 1.8500
Normal termination of Gaussian

```

Contoh ORCA:

```

FINAL SINGLE POINT ENERGY      -76.3456789012
ORCA TERMINATED NORMALLY

```

Contoh GAMESS:

```

TOTAL ENERGY = -76.3456789012
EXECUTION OF GAMESS TERMINATED NORMALLY

```

Parser dalam praktikum ini memasukkan beberapa pola umum tersebut. Jika mahasiswa menggunakan software lain, pola baru dapat ditambahkan.

24. Menambahkan Pola Regex Baru

Jika parser tidak menemukan data tertentu, mahasiswa dapat menambahkan pola baru. Misalnya output software menulis energi sebagai:

```
ENERGY TOTAL = -76.3456789012
```

Maka pola baru dapat ditambahkan ke `pola_list` dalam fungsi `cari_energi_total`:

```
r"ENERGY TOTAL\s*=\s*([-+]?\d+\.\d+(?:[Ee]([-+]?\d+)?)")
```

Kemampuan menambahkan pola regex sangat penting. Mahasiswa belajar bahwa parser bukan benda mati, tetapi alat yang dapat dikembangkan sesuai format output.

25. Prinsip Regular Expression dalam Parsing

Regular expression atau regex adalah pola teks untuk mencari bagian tertentu dalam string. Dalam parsing output, regex digunakan untuk menemukan angka, label, atau struktur kalimat tertentu.

Contoh pola angka desimal:

```
r"([-+]?\d+\.\d+)"
```

Maknanya:

Bagian	Makna
<code>[-+]?</code>	Tanda minus atau plus, boleh ada atau tidak
<code>\d+</code>	Satu atau lebih digit
<code>\.</code>	Titik desimal
<code>\d+</code>	Satu atau lebih digit setelah titik

Untuk menangkap notasi ilmiah:

```
r"([-+]?\d+\.\d+(?:[Ee]([-+]?\d+)?)")
```

Regex ini dapat membaca angka seperti:

```
-76.345678
1.234E-05
-2.50e+02
```

Dalam kimia komputasi, regex sangat berguna karena output berisi banyak angka.

26. Keterbatasan Regex Sederhana

Regex sederhana memiliki keterbatasan. Ia dapat mencari pola teks, tetapi tidak memahami konteks kimia. Misalnya, jika output memiliki banyak angka setelah kata “energy”, regex mungkin mengambil angka yang bukan energi akhir. Jika bagian orbital memiliki banyak angka, parser HOMO-LUMO sederhana dapat salah mengambil kolom.

Oleh karena itu, parser harus diuji.

Prinsip:

regex menemukan pola
tetapi pengguna menentukan makna kimianya

Jika hasil parser aneh, jangan langsung menyalahkan software. Periksa pola regex, format output, dan konteks angka.

27. Aplikasi Parsing untuk Laporan Praktikum

Hasil parsing dapat digunakan untuk membuat tabel laporan otomatis. Misalnya, tabel ringkasan output:

Molekul	Energi Total	Dipol	Frekuensi Imajiner	Status
H ₂ O	...	...	0	Valid
CO ₂	...	...	0	Valid
NH ₃	...	...	0	Valid
NH ₄ ⁺	...	...	0	Valid

Tabel ini dapat dimasukkan ke laporan praktikum. Dengan cara ini, laporan menjadi lebih rapi dan konsisten.

Parsing juga dapat digunakan untuk memilih file yang bermasalah. Misalnya, semua output dengan frekuensi imajiner dapat difilter:

```
df_bermasalah = df[df["jumlah_frekuensi_imajiner"] > 0]
```

Atau semua output yang job-nya tidak normal:

```
df_error = df[df["status_job"] != "Selesai normal"]
```

Dengan demikian, Python membantu validasi massal.

28. Contoh Analisis Hasil Parsing

Contoh narasi laporan:

Parsing output dilakukan dengan Python untuk mengekstrak energi total, momen dipol, frekuensi, HOMO-LUMO, charge, multiplicity, dan status job. Hasil parsing disimpan dalam file ringkasan_output_komputasi.csv. Berdasarkan hasil parsing, seluruh file yang digunakan dalam analisis menunjukkan status job selesai normal dan tidak memiliki frekuensi imajiner, sehingga data layak digunakan untuk analisis lanjutan.

Jika ada file bermasalah:

Hasil parsing menunjukkan bahwa file dimer_air.log memiliki satu frekuensi imajiner. Karena dimer air dimaksudkan sebagai struktur minimum, file tersebut tidak langsung digunakan dalam analisis energi. Struktur perlu diperiksa dan dioptimasi ulang.

Narasi seperti ini menunjukkan bahwa mahasiswa menggunakan Python secara kritis.

29. Parsing Tidak Menggantikan Pemeriksaan Geometri

Salah satu keterbatasan parsing adalah tidak dapat sepenuhnya menilai apakah geometri masuk akal. Parser dapat membaca energi, frekuensi, dan status job, tetapi tidak dapat dengan mudah memastikan apakah dimer masih terbentuk, apakah konformer berubah, atau apakah orientasi ikatan hidrogen benar.

Oleh karena itu, pemeriksaan geometri tetap diperlukan.

Alur yang benar:

```
parsing output
→ cek status dan angka penting
→ buka struktur akhir
→ validasi geometri
→ baru gunakan data dalam laporan
```

Jika parser menunjukkan semua status baik tetapi geometri akhir ternyata tidak sesuai, data tetap tidak boleh digunakan untuk kesimpulan yang dimaksud.

30. Integrasi Parsing dengan Subbab Sebelumnya

Subbab 11.4 mengintegrasikan tiga subbab sebelumnya.

Dari subbab 11.1, mahasiswa tahu informasi apa yang harus dicari dalam output.

Dari subbab 11.2, mahasiswa tahu kriteria validasi.

Dari subbab 11.3, mahasiswa tahu kesalahan umum yang harus diwaspadai.

Pada subbab 11.4, Python digunakan untuk membantu proses pencarian dan ringkasan.

Alur integrasinya:

```
membaca output
→ mengetahui pola penting
→ validasi hasil
→ mengenali kesalahan
→ parsing otomatis dengan Python
→ tabel ringkasan
→ laporan
```

Dengan demikian, parsing bukan keterampilan terpisah, tetapi bagian dari workflow validasi dan pelaporan.

31. Implikasi Keilmuan Parsing Output

Parsing output memiliki implikasi keilmuan penting. Dalam penelitian kimia komputasi, jumlah file output bisa sangat banyak. Tanpa otomasi, pengolahan data menjadi lambat dan rawan kesalahan. Parsing memungkinkan peneliti membuat dataset dari banyak perhitungan.

Misalnya, dalam studi konformer, puluhan atau ratusan struktur dapat dihitung. Parser dapat mengambil energi setiap konformer dan menyusun urutan kestabilan. Dalam studi spektrum, parser dapat mengambil frekuensi dan intensitas. Dalam studi energi reaksi, parser dapat mengambil energi reaktan dan produk. Dalam studi metode, parser dapat membandingkan hasil berbagai level teori.

Dengan demikian, parsing merupakan dasar dari workflow komputasi modern:

```
banyak perhitungan
→ banyak output
→ parsing
→ dataset
→ analisis statistik
→ kesimpulan ilmiah
```

32. Implikasi Pedagogis Parsing Output

Secara pedagogis, parsing output melatih mahasiswa memahami hubungan antara teks output dan data ilmiah. Mahasiswa belajar bahwa file output bukan dokumen yang harus dibaca secara pasif, tetapi sumber data yang dapat diolah secara sistematis.

Kemampuan yang dilatih:

1. Membaca file teks dengan Python.
2. Menggunakan regular expression.
3. Mengekstrak data numerik.
4. Menyusun tabel otomatis.
5. Menyimpan CSV.
6. Memvalidasi hasil parsing.
7. Mengenali keterbatasan otomatisasi.
8. Menghubungkan parsing dengan laporan praktikum.

Keterampilan ini penting untuk calon guru dan calon peneliti. Guru dapat menggunakan parsing untuk membuat contoh analisis data yang lebih efisien. Peneliti dapat menggunakannya sebagai dasar workflow komputasi yang lebih besar.

33. Kesalahan Umum dalam Parsing Output

Beberapa kesalahan umum dalam parsing output adalah:

Kesalahan	Dampak
Tidak memeriksa hasil parsing manual	Data salah tidak terdeteksi
Menganggap semua output memiliki format sama	Parser gagal pada software berbeda
Mengambil energi pertama, bukan energi akhir	Energi tidak valid
Tidak membedakan energi elektronik dan Gibbs	Analisis termodinamika salah
Salah membaca HOMO-LUMO	Analisis orbital salah
Tidak menangani file error	Output gagal tetap masuk laporan
Tidak mencatat catatan parsing	Data kosong tidak disadari
Tidak menyimpan CSV	Hasil tidak reproduisibel
Mengabaikan geometri	Struktur salah tetap digunakan
Tidak memperbarui regex	Parser tidak sesuai format output

Kesalahan ini harus dihindari agar otomatisasi benar-benar membantu, bukan menambah masalah.

34. Format Laporan Hasil Parsing

Dalam laporan praktikum, hasil parsing dapat ditulis sebagai berikut:

File output komputasi dibaca menggunakan skrip Python `parsing_output_komputasi.py`. Skrip tersebut mengekstrak energi total, momen dipol, frekuensi, HOMO-LUMO, charge, multiplicity, metode, basis set, dan status job. Hasil parsing disimpan dalam file `ringkasan_output_komputasi.csv`. Data yang digunakan untuk analisis dipilih dari output yang selesai normal, memiliki charge dan multiplicity benar, serta tidak memiliki frekuensi imajiner untuk struktur minimum.

Jika ada keterbatasan:

Hasil parsing HOMO-LUMO diperiksa ulang secara manual karena format orbital berbeda antarsoftware. Oleh karena itu, nilai HOMO-LUMO dari parser digunakan sebagai ringkasan awal, bukan satu-satunya dasar interpretasi.

Pernyataan seperti ini menunjukkan kehati-hatian metodologis.

35. Sintesis Subbab 11.4

Python untuk parsing output memungkinkan mahasiswa membaca file log, mencari energi total, momen dipol, frekuensi, HOMO-LUMO, charge, multiplicity, metode, basis set, dan status job secara otomatis. Hasil parsing dapat disimpan dalam CSV sehingga memudahkan penyusunan tabel laporan dan validasi banyak file sekaligus.

Namun, parsing otomatis harus selalu digunakan bersama validasi manual. Python dapat membaca pola, tetapi tidak memahami konteks kimia secara penuh. Mahasiswa tetap harus memeriksa geometri, metode, basis set, charge, multiplicity, frekuensi imajiner, dan kesesuaian hasil dengan teori.

Alur sintesisnya:

```
file output
→ Python membaca teks
→ regex mencari pola
→ data diekstrak
→ ringkasan CSV dibuat
→ hasil divalidasi
→ laporan disusun
```

Dengan menguasai parsing output, mahasiswa mulai memasuki tahap kerja kimia komputasi yang lebih modern, efisien, dan reproduibel.

36. Transisi ke Subbab 11.5

Subbab 11.4 telah membahas penggunaan Python untuk parsing output. Mahasiswa telah mempelajari cara membaca file log, mencari energi total, mencari momen dipol, mengekstrak frekuensi, membaca HOMO-LUMO secara sederhana, membuat tabel otomatis, dan menyimpan ringkasan output ke CSV. Subbab ini juga menegaskan bahwa parsing otomatis harus tetap divalidasi secara manual agar data yang diambil benar-benar bermakna secara kimia.

Subbab berikutnya akan membahas Python untuk grafik laporan. Pada subbab 11.5, mahasiswa akan belajar membuat grafik energi, diagram orbital, spektrum IR, diagram energi reaksi, dan grafik perbandingan metode. Dengan demikian, data yang telah diekstrak melalui parsing output pada subbab 11.4 akan divisualisasikan secara lebih komunikatif pada subbab 11.5.

11.5 Python untuk Grafik Laporan

Python untuk grafik laporan adalah penggunaan Python sebagai alat visualisasi data hasil praktikum kimia komputasi. Setelah file output dibaca dan diringkas melalui parsing pada subbab 11.4, data yang telah diperoleh perlu disajikan dalam bentuk grafik agar lebih mudah dipahami, dibandingkan, dan dijelaskan dalam laporan. Grafik bukan sekadar hiasan laporan. Grafik adalah media ilmiah untuk menampilkan pola, tren, perbedaan, hubungan, dan kesimpulan yang mungkin sulit dilihat hanya dari tabel angka.

Dalam konteks Bab 11, grafik laporan berfungsi sebagai jembatan antara **data komputasi** dan **argumentasi ilmiah**. Data energi, orbital, frekuensi, intensitas IR, energi reaksi, dan perbandingan metode dapat ditampilkan secara visual sehingga mahasiswa lebih mudah menyusun analisis. Grafik yang baik membantu pembaca memahami hasil secara cepat, tetapi grafik yang buruk dapat menyesatkan. Oleh karena itu, subbab ini tidak hanya membahas cara membuat grafik dengan Python, tetapi juga membahas prinsip ilmiah di balik penyajian grafik.

Pada subbab ini, mahasiswa belajar membuat:

1. **Grafik energi.**
2. **Diagram orbital.**
3. **Spektrum IR.**
4. **Diagram energi reaksi.**
5. **Grafik perbandingan metode.**

Kelima jenis grafik tersebut mewakili kebutuhan visualisasi utama dalam praktikum kimia komputasi. Grafik energi digunakan untuk membandingkan energi molekul, isomer, konformer, atau spesies reaktan-produk. Diagram orbital digunakan untuk menampilkan HOMO, LUMO, dan celah energi. Spektrum IR digunakan untuk memvisualisasikan hasil frekuensi vibrasi dan intensitas serapan. Diagram energi reaksi digunakan untuk menampilkan perubahan energi dari reaktan ke produk. Grafik perbandingan metode digunakan untuk melihat pengaruh metode atau basis set terhadap hasil komputasi.

Secara umum, alur pembuatan grafik laporan dapat diringkas sebagai berikut:


```

output komputasi
→ parsing data
→ tabel CSV
→ Python membaca CSV
→ grafik dibuat
→ grafik disimpan
→ grafik dimasukkan ke laporan
→ grafik dianalisis secara kimia

```

Alur ini memperlihatkan bahwa grafik tidak dibuat dari data sembarangan. Grafik harus berasal dari data yang telah divalidasi. Jika data belum valid, grafik yang dihasilkan juga tidak valid, meskipun tampak rapi.

1. Pengertian Grafik Laporan dalam Kimia Komputasi

Grafik laporan adalah representasi visual dari data hasil perhitungan kimia komputasi yang digunakan untuk menjelaskan hasil praktikum. Grafik dapat berbentuk diagram batang, grafik garis, diagram energi, spektrum, atau visualisasi perbandingan.

Dalam kimia komputasi, grafik digunakan untuk menyajikan:

1. Energi total.
2. Energi relatif.
3. Energi HOMO dan LUMO.
4. Celah HOMO-LUMO.
5. Frekuensi vibrasi.
6. Intensitas IR.
7. ΔE , ΔH , dan ΔG .
8. Perbandingan metode.
9. Perbandingan basis set.
10. Perbandingan fase gas dan pelarut.

Grafik yang baik harus memenuhi beberapa syarat:

Syarat Grafik	Penjelasan
Jelas	Label, sumbu, dan satuan mudah dibaca
Relevan	Menampilkan data yang benar-benar dibutuhkan
Tidak menyesatkan	Skala, tanda, dan interpretasi tidak keliru
Reprodusibel	Dapat dibuat ulang dari data dan kode
Terhubung dengan analisis	Grafik harus dijelaskan dalam laporan
Berbasis data valid	Data berasal dari output yang telah diperiksa

Dengan demikian, grafik bukan sekadar pelengkap visual. Grafik adalah bagian dari argumen ilmiah.

2. Latar Historis Visualisasi Data Komputasi

Dalam perkembangan kimia komputasi, visualisasi selalu memiliki peran penting. Pada awalnya, hasil komputasi banyak disajikan dalam bentuk tabel energi dan koordinat atom. Seiring berkembangnya perangkat lunak grafis dan pemrograman ilmiah, visualisasi molekul, orbital, spektrum, dan energi menjadi bagian penting dari komunikasi ilmiah.

Visualisasi data membantu peneliti melihat pola yang sulit dikenali dari tabel. Dalam spektroskopi, grafik intensitas terhadap bilangan gelombang membantu mengenali gugus fungsi. Dalam analisis orbital, diagram HOMO-LUMO membantu memahami reaktivitas elektronik. Dalam termodinamika, diagram energi membantu menjelaskan kestabilan relatif. Dalam studi metode, grafik perbandingan membantu menilai sensitivitas hasil terhadap level teori.

Literatur kimia komputasi dan kimia fisik modern tidak hanya menampilkan persamaan dan tabel, tetapi juga grafik yang dirancang untuk memperjelas hubungan antara struktur, energi, dan sifat molekul. Oleh karena itu, mahasiswa kimia komputasi perlu menguasai visualisasi data sebagai bagian dari literasi ilmiah.

Python, khususnya pustaka seperti `matplotlib` dan `pandas`, menjadi alat yang sangat berguna karena memungkinkan mahasiswa membuat grafik dari data CSV secara terstruktur, mengulang grafik dengan data baru, dan menyimpan hasil dalam format gambar berkualitas tinggi.

3. Prinsip Ilmiah dalam Membuat Grafik

Sebelum menulis kode, mahasiswa perlu memahami prinsip ilmiah dalam membuat grafik.

Pertama, grafik harus memiliki **judul yang jelas**. Judul harus menunjukkan sistem dan jenis data yang ditampilkan.

Kedua, sumbu harus memiliki **label dan satuan**. Misalnya, energi harus ditulis dalam kJ/mol , frekuensi dalam cm^{-1} , dan momen dipol dalam Debye.

Ketiga, grafik harus menggunakan **data yang sudah divalidasi**. Jangan membuat grafik dari output yang tidak konvergen, memiliki charge salah, atau frekuensi imajiner yang belum ditangani.

Keempat, grafik harus **sesuai dengan jenis data**. Spektrum IR lebih tepat dibuat sebagai grafik garis. Perbandingan metode lebih tepat menggunakan diagram batang. Diagram energi reaksi dapat dibuat dengan garis atau level energi.

Kelima, grafik harus **tidak terlalu ramai**. Terlalu banyak label, garis, atau data dalam satu grafik dapat membuat pembaca sulit memahami pesan utama.

Keenam, grafik harus **dianalisis**, bukan hanya ditempel. Setiap grafik dalam laporan harus diikuti penjelasan singkat tentang makna kimianya.

Pernyataan penting:

grafik yang baik
= data valid + desain jelas + interpretasi ilmiah

4. Pustaka Python yang Digunakan

Subbab ini menggunakan dua pustaka utama:

```
import pandas as pd
import matplotlib.pyplot as plt
from pathlib import Path
```

pandas digunakan untuk membaca dan mengolah data tabel, terutama CSV. matplotlib.pyplot digunakan untuk membuat grafik. Path dari pathlib digunakan untuk mengatur folder input dan output.

Pustaka ini dipilih karena sederhana, umum digunakan, dan cukup untuk kebutuhan praktikum. Mahasiswa tidak perlu menggunakan pustaka visualisasi yang terlalu kompleks pada tahap ini. Tujuan utama adalah memahami alur data dan grafik secara ilmiah.

5. Struktur Folder untuk Grafik Laporan

Struktur folder yang dianjurkan:

```
praktikum_grafik_laporan/
├── data/
│   ├── data_energi.csv
│   ├── data_orbital.csv
│   ├── data_ir.csv
│   ├── data_reaksi.csv
│   └── data_perbandingan_metode.csv
├── kode/
│   └── grafik_laporan_komputasi.py
└── grafik/
    ├── grafik_energi.png
    ├── diagram_orbital.png
    ├── spektrum_ir.png
    ├── diagram_energi_reaksi.png
    └── grafik_perbandingan_metode.png
```

Folder `data` berisi CSV. Folder `kode` berisi skrip Python. Folder `grafik` berisi gambar hasil visualisasi. Struktur ini membuat proses lebih reproduisibel.

Jika laporan memuat gambar `spektrum_ir.png`, pembaca dapat menelusuri kembali bahwa gambar tersebut berasal dari `data_ir.csv` dan dibuat oleh `grafik_laporan_komputasi.py`.

6. Format Data untuk Grafik Energi

Grafik energi digunakan untuk membandingkan energi beberapa spesies. Data dapat berupa energi total atau energi relatif. Dalam laporan, energi relatif biasanya lebih mudah dipahami.

Contoh CSV:

```
species,energi_kJmol
cis_2butena,0.00
trans_2butena,-4.20
etanol_konformer_A,0.00
etanol_konformer_B,2.75
h2o_monomer,0.00
h2o_dimer,-13.10
```

Untuk perbandingan isomer atau konformer, energi terendah sering dijadikan nol. Energi spesies lain dihitung relatif terhadap energi terendah.

Contoh interpretasi:

Jika trans-2-butena memiliki energi relatif lebih rendah daripada cis-2-butena, maka trans-2-butena lebih stabil pada tingkat teori yang digunakan.

7. Kode Python untuk Grafik Energi

Kode berikut membuat grafik energi sederhana dari file `data_energi.csv`.

```
from pathlib import Path
import pandas as pd
import matplotlib.pyplot as plt

folder_data = Path("data")
folder_grafik = Path("grafik")
folder_grafik.mkdir(exist_ok=True)

file_data = folder_data / "data_energi.csv"

df = pd.read_csv(file_data)

plt.figure(figsize=(8, 5))
plt.bar(df["species"], df["energi_kJmol"])
```

```
plt.axhline(0, linewidth=1)
plt.xlabel("Spesies")
plt.ylabel("Energi relatif (kJ/mol)")
plt.title("Grafik Energi Relatif Spesies")
plt.xticks(rotation=45, ha="right")
plt.tight_layout()
plt.savefig(folder_grafik / "grafik_energi.png", dpi=300)
plt.close()
```

Kode tersebut menghasilkan grafik batang energi relatif. Garis horizontal pada nilai nol membantu pembaca melihat spesies mana yang berada di bawah atau di atas acuan energi.

8. Analisis Grafik Energi

Grafik energi harus dianalisis berdasarkan konteks kimia. Jika grafik menunjukkan energi relatif konformer, maka analisis berfokus pada kestabilan konformer. Jika grafik menunjukkan energi isomer, analisis berfokus pada perbedaan struktur. Jika grafik menunjukkan dimerisasi, analisis berfokus pada pembentukan interaksi antarmolekul.

Contoh analisis untuk konformer etanol:

Grafik energi relatif konformer etanol menunjukkan bahwa konformer A memiliki energi terendah dan dijadikan acuan 0 kJ/mol. Konformer B berada beberapa kJ/mol lebih tinggi, sehingga kurang stabil pada tingkat teori yang digunakan. Karena selisih energi antarkonformer kecil, interpretasi harus mempertimbangkan ketelitian metode dan basis set.

Contoh analisis untuk cis-trans 2-butena:

Grafik energi relatif menunjukkan bahwa trans-2-butena lebih rendah energinya dibanding cis-2-butena. Hal ini sesuai dengan penjelasan sterik bahwa gugus metil pada isomer trans lebih berjauhan sehingga tolakan sterik lebih kecil.

Grafik energi tidak boleh dibiarkan tanpa narasi.

9. Format Data untuk Diagram Orbital

Diagram orbital digunakan untuk menampilkan energi HOMO, LUMO, dan celah HOMO-LUMO. Data dapat berasal dari parsing output atau pembacaan manual.

Contoh CSV:

```
molekul,HOMO_eV,LUMO_eV
formaldehida,-6.80,-1.20
asetona,-6.50,-0.90
etanol,-7.80,0.50
```

```
amonia, -6.10, 1.00
```

Energi orbital dapat ditampilkan dalam eV atau Hartree. Dalam laporan, eV sering lebih mudah dipahami untuk orbital. Namun, satuan harus jelas.

Celah HOMO-LUMO dihitung:

```
gap = LUMO - HOMO
```

Semakin kecil celah, semakin mudah transisi elektronik secara kualitatif. Namun, interpretasi reaktivitas tidak boleh hanya didasarkan pada celah orbital tanpa mempertimbangkan konteks molekul.

10. Kode Python untuk Diagram Orbital

Kode berikut membuat diagram orbital HOMO-LUMO untuk beberapa molekul.

```
from pathlib import Path
import pandas as pd
import matplotlib.pyplot as plt

folder_data = Path("data")
folder_grafik = Path("grafik")
folder_grafik.mkdir(exist_ok=True)

df = pd.read_csv(folder_data / "data_orbital.csv")

plt.figure(figsize=(8, 6))

x_pos = range(len(df))

plt.scatter(x_pos, df["HOMO_eV"], marker="_", s=600, label="HOMO")
plt.scatter(x_pos, df["LUMO_eV"], marker="_", s=600, label="LUMO")

for i, baris in df.iterrows():
    plt.plot([i, i], [baris["HOMO_eV"], baris["LUMO_eV"]], linewidth=1)
    gap = baris["LUMO_eV"] - baris["HOMO_eV"]
    plt.text(i, (baris["HOMO_eV"] + baris["LUMO_eV"]) / 2, f"{gap:.2f} eV",
             ha="center", va="center")

plt.xticks(list(x_pos), df["molekul"], rotation=45, ha="right")
plt.ylabel("Energi orbital (eV)")
plt.title("Diagram HOMO-LUMO")
plt.legend()
plt.tight_layout()
plt.savefig(folder_grafik / "diagram_orbital.png", dpi=300)
plt.close()
```

Diagram ini menampilkan HOMO dan LUMO sebagai garis pendek. Jarak antara HOMO dan LUMO menunjukkan celah energi.

11. Analisis Diagram Orbital

Diagram orbital harus dianalisis secara hati-hati. HOMO sering dikaitkan dengan kemampuan molekul mendonorkan elektron, sedangkan LUMO dikaitkan dengan kemampuan menerima elektron. Celah HOMO-LUMO dapat digunakan sebagai indikator kualitatif reaktivitas elektronik.

Contoh analisis:

Diagram HOMO-LUMO menunjukkan bahwa formaldehida memiliki celah energi lebih kecil dibanding etanol. Secara kualitatif, hal ini dapat menunjukkan bahwa formaldehida lebih mudah mengalami interaksi elektronik tertentu, terutama karena adanya gugus karbonil yang memiliki orbital π dan π^* . Namun, interpretasi reaktivitas tidak boleh hanya didasarkan pada gap HOMO-LUMO; struktur, polaritas, muatan, dan konteks reaksi tetap harus dipertimbangkan.

Analisis seperti ini menunjukkan kehati-hatian ilmiah.

12. Format Data untuk Spektrum IR

Spektrum IR dibuat dari data frekuensi dan intensitas. Data ini dapat berasal dari output frekuensi yang telah diparsing atau disalin ke CSV.

Contoh CSV:

```
frekuensi_cm1,intensitas
500,5
750,12
1050,20
1600,35
1750,80
3000,25
3600,60
```

Frekuensi biasanya dinyatakan dalam cm^{-1} . Intensitas dapat berupa intensitas IR relatif atau intensitas langsung dari output.

Dalam spektrum IR eksperimen, puncak tidak berupa garis vertikal murni, tetapi memiliki bentuk pita. Dalam simulasi praktikum, puncak dapat dibuat sebagai garis vertikal sederhana atau diperhalus menggunakan fungsi pelebaran.

13. Kode Python untuk Spektrum IR Garis Sederhana

Kode berikut membuat spektrum IR sederhana menggunakan garis vertikal.

```
from pathlib import Path
import pandas as pd
import matplotlib.pyplot as plt

folder_data = Path("data")
folder_grafik = Path("grafik")
folder_grafik.mkdir(exist_ok=True)

df = pd.read_csv(folder_data / "data_ir.csv")

plt.figure(figsize=(9, 5))
plt.vlines(df["frekuensi_cm1"], 0, df["intensitas"], linewidth=1)
plt.xlabel("Bilangan gelombang ( $\text{cm}^{-1}$ )")
plt.ylabel("Intensitas IR")
plt.title("Spektrum IR Komputasi")
plt.gca().invert_xaxis()
plt.tight_layout()
plt.savefig(folder_grafik / "spektrum_ir_garis.png", dpi=300)
plt.close()
```

Perintah:

```
plt.gca().invert_xaxis()
```

digunakan karena spektrum IR biasanya ditampilkan dari bilangan gelombang tinggi ke rendah.

14. Kode Python untuk Spektrum IR dengan Pelebaran Gaussian

Agar spektrum tampak lebih mirip spektrum nyata, setiap puncak dapat dilebarkan dengan fungsi Gaussian sederhana.

```
from pathlib import Path
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

folder_data = Path("data")
folder_grafik = Path("grafik")
folder_grafik.mkdir(exist_ok=True)

df = pd.read_csv(folder_data / "data_ir.csv")

x = np.linspace(400, 4000, 4000)
y = np.zeros_like(x)

lebar = 15.0
```



```

for _, baris in df.iterrows():
    pusat = baris["frekuensi_cm1"]
    intensitas = baris["intensitas"]
    y += intensitas * np.exp(-0.5 * ((x - pusat) / lebar) ** 2)

plt.figure(figsize=(9, 5))
plt.plot(x, y)
plt.xlabel("Bilangan gelombang (cm-1)")
plt.ylabel("Intensitas relatif")
plt.title("Spektrum IR Komputasi dengan Pelebaran Gaussian")
plt.gca().invert_xaxis()
plt.tight_layout()
plt.savefig(folder_grafik / "spektrum_ir_gaussian.png", dpi=300)
plt.close()

```

Kode ini menghasilkan spektrum yang lebih halus. Nilai `lebar` dapat disesuaikan. Namun, mahasiswa harus menjelaskan bahwa pelebaran ini adalah simulasi visual, bukan hasil eksperimen langsung.

15. Analisis Spektrum IR

Spektrum IR harus dianalisis dengan menghubungkan puncak dengan mode vibrasi. Misalnya:

Wilayah Frekuensi	Interpretasi Umum
3200–3600 cm ⁻¹	Regangan O–H atau N–H
2800–3100 cm ⁻¹	Regangan C–H
sekitar 1700 cm ⁻¹	Regangan C=O
1000–1300 cm ⁻¹	Regangan C–O atau mode rangka tertentu
di bawah 1000 cm ⁻¹	Mode tekukan atau deformasi

Contoh analisis:

Spektrum IR formaldehida menunjukkan puncak kuat pada wilayah karbonil, yang dapat dikaitkan dengan regangan C=O. Puncak pada wilayah C–H menunjukkan mode regangan ikatan C–H. Karena perhitungan menggunakan pendekatan harmonik, nilai frekuensi dapat berbeda dari eksperimen dan dapat memerlukan faktor skala.

Spektrum IR harus dikaitkan dengan struktur molekul, bukan hanya daftar puncak.

16. Format Data untuk Diagram Energi Reaksi

Diagram energi reaksi digunakan untuk menampilkan energi relatif reaktan dan produk. Data dapat berupa ΔE , ΔH , atau ΔG . Dalam Praktikum 7, diagram energi Gibbs sering digunakan.

Contoh CSV:

```
tahap,energi_kJmol
Reaktan,0.00
Produk,-12.50
```

Untuk reaksi dengan lebih banyak tahap, misalnya reaktan, intermediat, keadaan transisi, dan produk, data dapat diperluas. Namun, dalam Bab 10, diagram energi reaksi difokuskan pada perbandingan reaktan dan produk.

17. Kode Python untuk Diagram Energi Reaksi

Kode berikut membuat diagram energi reaksi sederhana.

```
from pathlib import Path
import pandas as pd
import matplotlib.pyplot as plt

folder_data = Path("data")
folder_grafik = Path("grafik")
folder_grafik.mkdir(exist_ok=True)

df = pd.read_csv(folder_data / "data_reaksi.csv")

plt.figure(figsize=(7, 5))
plt.plot(df["tahap"], df["energi_kJmol"], marker="o")
plt.axhline(0, linewidth=1)
plt.xlabel("Tahap reaksi")
plt.ylabel("Energi relatif (kJ/mol)")
plt.title("Diagram Energi Reaksi")
plt.tight_layout()
plt.savefig(folder_grafik / "diagram_energi_reaksi.png", dpi=300)
plt.close()
```

Jika energi produk lebih rendah daripada reaktan, grafik menunjukkan penurunan energi. Jika produk lebih tinggi, grafik menunjukkan kenaikan energi.

18. Analisis Diagram Energi Reaksi

Diagram energi reaksi harus dianalisis sebagai diagram termodinamika jika hanya memuat reaktan dan produk. Diagram ini tidak menunjukkan energi aktivasi.

Contoh analisis:

Diagram energi reaksi menunjukkan bahwa produk berada lebih rendah daripada reaktan, sehingga ΔG bernilai negatif. Hal ini menunjukkan bahwa produk lebih menguntungkan secara termodinamika pada kondisi perhitungan. Namun, diagram ini

tidak memberikan informasi tentang laju reaksi atau energi aktivasi karena keadaan transisi tidak dihitung.

Jika diagram menunjukkan produk lebih tinggi:

Produk berada pada energi relatif lebih tinggi daripada reaktan, sehingga ΔG bernilai positif. Pada kondisi perhitungan, reaktan lebih menguntungkan secara termodinamika dibanding produk.

Mahasiswa harus membedakan diagram termodinamika dan diagram kinetika.

19. Format Data untuk Grafik Perbandingan Metode

Grafik perbandingan metode digunakan untuk melihat bagaimana hasil berubah jika metode atau basis set berbeda. Misalnya, energi relatif dihitung dengan HF, B3LYP, dan PBE0.

Contoh CSV:

```
metode,delta_G_kJmol
HF_STO-3G,25.40
B3LYP_6-31Gd,12.80
PBE0_6-31Gd,10.95
B3LYP_def2SVP,11.70
```

Data ini dapat digunakan untuk membahas sensitivitas hasil terhadap level teori.

20. Kode Python untuk Grafik Perbandingan Metode

Kode berikut membuat grafik perbandingan metode.

```
from pathlib import Path
import pandas as pd
import matplotlib.pyplot as plt

folder_data = Path("data")
folder_grafik = Path("grafik")
folder_grafik.mkdir(exist_ok=True)

df = pd.read_csv(folder_data / "data_perbandingan_metode.csv")

plt.figure(figsize=(8, 5))
plt.bar(df["metode"], df["delta_G_kJmol"])
plt.axhline(0, linewidth=1)
plt.xlabel("Metode / basis set")
plt.ylabel("ΔG (kJ/mol)")
plt.title("Perbandingan ΔG Berdasarkan Metode")
plt.xticks(rotation=45, ha="right")
```

```
plt.tight_layout()
plt.savefig(folder_grafik / "grafik_perbandingan_metode.png", dpi=300)
plt.close()
```

Grafik ini membantu mahasiswa melihat apakah kesimpulan berubah ketika metode berubah. Jika semua metode menunjukkan tanda ΔG yang sama, maka tren lebih kuat. Jika metode berbeda menghasilkan tanda berbeda, kesimpulan harus dibuat lebih hati-hati.

21. Analisis Grafik Perbandingan Metode

Perbandingan metode sangat penting dalam kimia komputasi. Hasil perhitungan tidak hanya bergantung pada sistem kimia, tetapi juga pada metode dan basis set yang digunakan. Oleh karena itu, grafik perbandingan metode membantu mahasiswa memahami sensitivitas hasil.

Contoh analisis:

Grafik perbandingan metode menunjukkan bahwa nilai ΔG berbeda antara HF/STO-3G dan B3LYP/6-31G(d). Perbedaan ini menunjukkan bahwa hasil termodinamika sensitif terhadap level teori. Jika semua metode menghasilkan tanda ΔG yang sama, maka kesimpulan kualitatif lebih kuat. Namun, jika tanda ΔG berubah antar metode, hasil harus ditafsirkan secara hati-hati.

Analisis ini penting karena kimia komputasi bukan sekadar satu angka dari satu metode. Pemilihan metode memengaruhi kualitas hasil.

22. Kode Terpadu untuk Semua Grafik Laporan

Kode berikut menyatukan pembuatan lima jenis grafik dalam satu file. Nama file dapat dibuat:

```
grafik_laporan_komputasi.py
```

Kode ini membaca beberapa CSV dan membuat grafik jika file data tersedia.

```
from pathlib import Path
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

# =====
# BAB 11
# PYTHON UNTUK GRAFIK LAPORAN
# -----
# Nama file:
# grafik_laporan_komputasi.py
#
```

```

# Fungsi:
# 1. Membuat grafik energi
# 2. Membuat diagram orbital HOMO-LUMO
# 3. Membuat spektrum IR
# 4. Membuat diagram energi reaksi
# 5. Membuat grafik perbandingan metode
# =====

folder_data = Path("data")
folder_grafik = Path("grafik")
folder_grafik.mkdir(exist_ok=True)

def simpan_grafik(nama_file):
    """
    Menyimpan grafik ke folder grafik dengan resolusi tinggi.
    """
    plt.tight_layout()
    plt.savefig(folder_grafik / nama_file, dpi=300)
    plt.close()

def buat_grafik_energi():
    file_data = folder_data / "data_energi.csv"

    if not file_data.exists():
        print("File data_energi.csv tidak ditemukan.")
        return

    df = pd.read_csv(file_data)

    plt.figure(figsize=(8, 5))
    plt.bar(df["spesies"], df["energi_kJmol"])
    plt.axhline(0, linewidth=1)
    plt.xlabel("Spesies")
    plt.ylabel("Energi relatif (kJ/mol)")
    plt.title("Grafik Energi Relatif Spesies")
    plt.xticks(rotation=45, ha="right")

    simpan_grafik("grafik_energi.png")
    print("Grafik energi dibuat.")

def buat_diagram_orbital():
    file_data = folder_data / "data_orbital.csv"

    if not file_data.exists():
        print("File data_orbital.csv tidak ditemukan.")
        return

    df = pd.read_csv(file_data)

    plt.figure(figsize=(8, 6))
    x_pos = range(len(df))

    plt.scatter(x_pos, df["HOMO_eV"], marker="_", s=600, label="HOMO")

```

```

plt.scatter(x_pos, df["LUMO_eV"], marker="_", s=600, label="LUMO")

for i, baris in df.iterrows():
    plt.plot([i, i], [baris["HOMO_eV"], baris["LUMO_eV"]], linewidth=1)
    gap = baris["LUMO_eV"] - baris["HOMO_eV"]
    plt.text(i, (baris["HOMO_eV"] + baris["LUMO_eV"]) / 2,
             f"{gap:.2f} eV", ha="center", va="center")

plt.xticks(list(x_pos), df["molekul"], rotation=45, ha="right")
plt.ylabel("Energi orbital (eV)")
plt.title("Diagram HOMO-LUMO")
plt.legend()

simpan_grafik("diagram_orbital.png")
print("Diagram orbital dibuat.")

def buat_spektrum_ir():
    file_data = folder_data / "data_ir.csv"

    if not file_data.exists():
        print("File data_ir.csv tidak ditemukan.")
        return

    df = pd.read_csv(file_data)

    x = np.linspace(400, 4000, 4000)
    y = np.zeros_like(x)

    lebar = 15.0

    for _, baris in df.iterrows():
        pusat = baris["frekuensi_cm1"]
        intensitas = baris["intensitas"]
        y += intensitas * np.exp(-0.5 * ((x - pusat) / lebar) ** 2)

    plt.figure(figsize=(9, 5))
    plt.plot(x, y)
    plt.xlabel("Bilangan gelombang (cm-1)")
    plt.ylabel("Intensitas relatif")
    plt.title("Spektrum IR Komputasi")
    plt.gca().invert_xaxis()

    simpan_grafik("spektrum_ir.png")
    print("Spektrum IR dibuat.")

def buat_diagram_energi_reaksi():
    file_data = folder_data / "data_reaksi.csv"

    if not file_data.exists():
        print("File data_reaksi.csv tidak ditemukan.")
        return

    df = pd.read_csv(file_data)

    plt.figure(figsize=(7, 5))

```

```

plt.plot(df["tahap"], df["energi_kJmol"], marker="o")
plt.axhline(0, linewidth=1)
plt.xlabel("Tahap reaksi")
plt.ylabel("Energi relatif (kJ/mol)")
plt.title("Diagram Energi Reaksi")

simpan_grafik("diagram_energi_reaksi.png")
print("Diagram energi reaksi dibuat.")

def buat_grafik_perbandingan_metode():
    file_data = folder_data / "data_perbandingan_metode.csv"

    if not file_data.exists():
        print("File data_perbandingan_metode.csv tidak ditemukan.")
        return

    df = pd.read_csv(file_data)

    plt.figure(figsize=(8, 5))
    plt.bar(df["metode"], df["delta_G_kJmol"])
    plt.axhline(0, linewidth=1)
    plt.xlabel("Metode / basis set")
    plt.ylabel("ΔG (kJ/mol)")
    plt.title("Perbandingan ΔG Berdasarkan Metode")
    plt.xticks(rotation=45, ha="right")

    simpan_grafik("grafik_perbandingan_metode.png")
    print("Grafik perbandingan metode dibuat.")

if __name__ == "__main__":
    buat_grafik_energi()
    buat_diagram_orbital()
    buat_spektrum_ir()
    buat_diagram_energi_reaksi()
    buat_grafik_perbandingan_metode()

    print("Selesai membuat grafik laporan.")

```

Kode terpadu ini akan membuat grafik jika file CSV tersedia. Jika salah satu file tidak ada, program tidak berhenti total, tetapi memberi pesan bahwa file tersebut tidak ditemukan.

23. Validasi Grafik Sebelum Masuk Laporan

Sebelum grafik dimasukkan ke laporan, mahasiswa harus memvalidasi grafik. Validasi grafik mencakup:

1. Apakah data sumber benar?
2. Apakah satuan sudah benar?
3. Apakah sumbu diberi label?

4. Apakah judul sesuai?
5. Apakah nilai pada grafik sesuai CSV?
6. Apakah grafik tidak menampilkan data yang belum valid?
7. Apakah grafik mudah dibaca?
8. Apakah grafik sesuai dengan analisis?
9. Apakah resolusi gambar cukup baik?
10. Apakah grafik disimpan dengan nama file yang jelas?

Daftar periksa:

- [] CSV sudah divalidasi.
- [] Grafik sesuai data CSV.
- [] Label sumbu lengkap.
- [] Satuan energi/frekuensi jelas.
- [] Judul tidak menyesatkan.
- [] Gambar tersimpan dengan resolusi memadai.
- [] Grafik dijelaskan dalam laporan.

Grafik yang tidak divalidasi dapat menyesatkan pembaca.

24. Kesalahan Umum dalam Membuat Grafik Laporan

Beberapa kesalahan umum dalam membuat grafik laporan adalah:

Kesalahan	Dampak
Tidak mencantumkan satuan	Pembaca tidak tahu skala data
Grafik berasal dari data belum valid	Kesimpulan salah
Label sumbu tidak jelas	Grafik sulit dipahami
Judul terlalu umum	Konteks data tidak jelas
Skala menyesatkan	Perbedaan tampak terlalu besar atau terlalu kecil
Data reaktan-produk tertukar	Diagram energi salah
Spektrum IR tidak dibalik	Tampilan tidak sesuai konvensi IR
HOMO-LUMO tanpa satuan	Interpretasi energi orbital kabur
Grafik terlalu ramai	Pesan utama hilang
Grafik tidak dianalisis	Hanya menjadi hiasan laporan

Kesalahan ini harus dihindari agar grafik menjadi alat komunikasi ilmiah yang efektif.

25. Grafik dan Narasi Laporan

Setiap grafik harus diikuti narasi. Narasi harus menjelaskan apa yang terlihat dalam grafik dan apa maknanya secara kimia.

Format narasi yang baik:

Gambar menunjukkan ...
 Nilai tertinggi/terendah terdapat pada ...
 Perbedaan tersebut menunjukkan ...
 Secara kimia, hal ini dapat dijelaskan oleh ...
 Namun, interpretasi ini memiliki keterbatasan karena ...

Contoh untuk spektrum IR:

Spektrum IR komputasi menunjukkan puncak kuat pada wilayah karbonil. Puncak ini dikaitkan dengan mode regangan C=O. Nilai frekuensi dapat berbeda dari eksperimen karena perhitungan menggunakan pendekatan harmonik dan belum menerapkan faktor skala.

Contoh untuk perbandingan metode:

Grafik perbandingan metode menunjukkan bahwa nilai ΔG berubah ketika metode dan basis set diganti. Hal ini menunjukkan bahwa hasil termodinamika sensitif terhadap level teori. Oleh karena itu, kesimpulan kuantitatif harus disampaikan dengan menyebut metode yang digunakan.

26. Standar Nama File Grafik

Nama file grafik harus jelas. Hindari nama seperti:

grafik1.png
 gambar_baru.png
 hasil.png
 plot_fix.png
 final.png

Nama yang lebih baik:

grafik_energi.png
 diagram_orbital_homo_lumo.png
 spektrum_ir_formaldehida.png
 diagram_energi_reaksi_hidrasi_formaldehida.png
 grafik_perbandingan_metode_delta_g.png

Nama file yang jelas membantu keterlacakan. Jika laporan memuat grafik, file sumbernya mudah ditemukan.

27. Resolusi dan Format Gambar

Untuk laporan, grafik sebaiknya disimpan dengan resolusi cukup tinggi. Dalam kode, digunakan:

```
plt.savefig(nama_file, dpi=300)
```

Resolusi 300 dpi umumnya cukup baik untuk dokumen akademik. Format gambar yang dapat digunakan:

Format	Kegunaan
PNG	Baik untuk laporan digital dan dokumen umum
PDF	Baik untuk grafik vektor dan publikasi
SVG	Baik untuk editing vektor
JPG	Kurang ideal untuk grafik karena kompresi

Untuk laporan praktikum, PNG biasanya cukup. Untuk naskah publikasi, PDF atau SVG dapat lebih baik.

28. Hubungan Grafik dengan Reprodusibilitas

Grafik yang baik harus dapat dibuat ulang dari data dan kode. Artinya, laporan tidak hanya menyimpan gambar, tetapi juga menyimpan CSV dan skrip Python.

Alur reprodusibilitas grafik:

```
data CSV
→ kode Python
→ grafik PNG
→ laporan
```

Jika grafik dibuat secara manual tanpa data dan kode, sulit untuk memeriksa ulang. Jika ada kesalahan, grafik harus dibuat ulang dari awal. Dengan Python, grafik dapat diperbarui hanya dengan mengganti data CSV dan menjalankan ulang kode.

Pernyataan metodologis:

Semua grafik laporan dibuat dari data CSV menggunakan Python sehingga dapat direproduksi dan diperiksa ulang.

Pernyataan ini menunjukkan praktik ilmiah yang baik.

29. Analisis Kritis terhadap Visualisasi

Visualisasi dapat memperjelas data, tetapi juga dapat menyesatkan jika tidak dirancang dengan baik. Misalnya, diagram batang dengan skala yang tidak tepat dapat membuat perbedaan kecil tampak besar. Spektrum IR tanpa label satuan dapat membingungkan. Diagram energi reaksi tanpa keterangan bahwa itu bukan diagram mekanisme dapat menyebabkan salah interpretasi.

Karena itu, mahasiswa harus selalu bertanya:

1. Apakah grafik menjawab pertanyaan praktikum?
2. Apakah grafik menampilkan data yang benar?
3. Apakah grafik terlalu ramai?
4. Apakah grafik memerlukan satuan tambahan?
5. Apakah grafik menunjukkan energi relatif atau energi absolut?
6. Apakah grafik termodinamika atau kinetika?
7. Apakah pembaca dapat memahami grafik tanpa membuka kode?
8. Apakah grafik sudah dijelaskan dalam teks?

Grafik bukan tujuan akhir. Grafik adalah alat untuk menyampaikan argumen ilmiah.

30. Aplikasi Grafik untuk Tujuh Mata Praktikum

Grafik dalam Bab 11 dapat digunakan untuk seluruh mata praktikum sebelumnya.

Praktikum	Grafik yang Relevan
Optimasi geometri	Grafik energi selama optimasi
Metode dan basis set	Grafik perbandingan energi/metode
Orbital molekul	Diagram HOMO-LUMO
Momen dipol	Grafik momen dipol beberapa molekul
Frekuensi vibrasi	Spektrum IR
Energi reaksi	Diagram energi reaksi
Efek pelarut	Grafik perbandingan gas dan pelarut

Dengan demikian, Python untuk grafik laporan bukan hanya keterampilan teknis tambahan, tetapi alat pemersatu seluruh hasil praktikum.

31. Contoh Narasi Umum untuk Bagian Grafik dalam Laporan

Mahasiswa dapat menulis bagian grafik laporan sebagai berikut:

Data hasil praktikum divisualisasikan menggunakan Python. Grafik yang dibuat meliputi grafik energi relatif, diagram HOMO-LUMO, spektrum IR komputasi, diagram energi reaksi, dan grafik perbandingan metode. Semua grafik dibuat dari data CSV yang telah divalidasi. Label sumbu, satuan, dan judul grafik disesuaikan dengan jenis data yang ditampilkan. Grafik digunakan untuk memperjelas hubungan antara hasil numerik dan interpretasi kimia.

Narasi tersebut dapat disesuaikan dengan praktikum tertentu.

32. Implikasi Keilmuan Grafik Laporan

Grafik laporan memiliki implikasi keilmuan penting. Dalam penelitian dan pembelajaran kimia komputasi, data sering kompleks dan multidimensi. Grafik membantu menyederhanakan data tanpa menghilangkan makna pentingnya. Dengan grafik, mahasiswa dapat melihat pola kestabilan, perbedaan metode, hubungan frekuensi-intensitas, serta perubahan energi reaksi.

Namun, grafik juga menuntut tanggung jawab ilmiah. Setiap grafik harus mewakili data secara benar. Skala, label, satuan, dan judul harus jelas. Grafik tidak boleh digunakan untuk memperkuat kesimpulan yang tidak didukung data.

Dengan demikian, grafik adalah bagian dari etika komunikasi ilmiah.

33. Implikasi Pedagogis Grafik Laporan

Secara pedagogis, subbab ini melatih mahasiswa menghubungkan keterampilan kimia, komputasi, dan komunikasi ilmiah. Mahasiswa belajar bahwa hasil praktikum tidak cukup dihitung, tetapi harus disampaikan dengan jelas. Grafik membantu mahasiswa memahami data secara visual, tetapi juga menuntut mereka menjelaskan makna grafik secara konseptual.

Kemampuan yang dilatih meliputi:

1. Membaca CSV dengan Python.
2. Membuat grafik dengan Matplotlib.
3. Menentukan jenis grafik yang sesuai.
4. Memberi label dan satuan dengan benar.
5. Menyimpan grafik resolusi tinggi.
6. Menafsirkan grafik secara kimia.
7. Menghindari visualisasi yang menyesatkan.
8. Menyusun laporan yang komunikatif dan reproduktibel.

Keterampilan ini sangat penting bagi calon guru kimia. Guru tidak hanya perlu memahami data, tetapi juga mampu menyajikan data secara jelas kepada peserta didik.

34. Sintesis Subbab 11.5

Python untuk grafik laporan membantu mahasiswa mengubah data hasil praktikum menjadi visualisasi yang informatif. Grafik energi menampilkan kestabilan relatif. Diagram orbital menampilkan HOMO, LUMO, dan celah energi. Spektrum IR menampilkan frekuensi dan intensitas vibrasi. Diagram energi reaksi menampilkan perubahan energi reaktan-produk. Grafik perbandingan metode menampilkan sensitivitas hasil terhadap level teori.

Alur sintesisnya:

```
data CSV tervalidasi
→ Python membaca data
→ grafik dibuat
→ grafik disimpan
→ grafik dianalisis
→ laporan menjadi lebih komunikatif
```

Namun, grafik hanya bermakna jika data valid, satuan benar, label jelas, dan interpretasi tepat. Oleh karena itu, mahasiswa harus melihat grafik sebagai bagian dari argumentasi ilmiah, bukan sekadar gambar tambahan.

35. Transisi ke Subbab 11.6

Subbab 11.5 telah membahas penggunaan Python untuk membuat grafik laporan, meliputi grafik energi, diagram orbital, spektrum IR, diagram energi reaksi, dan grafik perbandingan metode. Data yang telah diekstrak melalui parsing output pada subbab 11.4 dapat divisualisasikan dengan Matplotlib sehingga laporan praktikum menjadi lebih jelas, komunikatif, dan reproduibel.

Subbab berikutnya akan membahas template laporan praktikum. Pada subbab 11.6, mahasiswa akan mempelajari format laporan yang seragam untuk tujuh mata praktikum. Template tersebut akan mengintegrasikan identitas praktikum, tujuan, dasar teori singkat, metode, input, output, validasi data, tabel hasil, grafik, analisis, kesimpulan, dan lampiran. Dengan demikian, grafik yang dibuat pada subbab ini akan ditempatkan secara sistematis dalam laporan praktikum yang utuh.

11.6 Template Laporan Praktikum

Template laporan praktikum adalah format baku yang digunakan untuk menyusun laporan hasil praktikum secara sistematis, konsisten, dan mudah diperiksa. Dalam Bab 11, template laporan praktikum disusun untuk menyeragamkan pelaporan pada tujuh mata praktikum yang telah dibahas dalam bagian sebelumnya. Template ini tidak hanya berfungsi sebagai kerangka administratif, tetapi juga sebagai alat pedagogis untuk melatih mahasiswa menyusun laporan kimia komputasi secara ilmiah, terukur, reproduibel, dan argumentatif.

Dalam praktikum kimia kuantum dan kimia komputasi, laporan tidak cukup hanya berisi gambar molekul dan angka energi. Laporan harus menunjukkan hubungan antara **tujuan praktikum, dasar teori, metode komputasi, struktur input, output software, validasi data, hasil numerik, grafik, analisis, kesimpulan, dan lampiran pendukung**. Tanpa template yang seragam, laporan mahasiswa mudah menjadi tidak konsisten: ada yang terlalu banyak menampilkan gambar tetapi kurang analisis, ada yang mencantumkan energi tetapi tidak menyebut metode, ada yang membuat grafik tetapi tidak menjelaskan satuan, dan ada pula yang menyimpulkan hasil tanpa memvalidasi output.

Template laporan praktikum dalam subbab ini dirancang untuk tujuh mata praktikum utama:

1. Praktikum optimasi geometri dan struktur molekul.
2. Praktikum metode dan basis set.
3. Praktikum orbital molekul dan HOMO-LUMO.
4. Praktikum momen dipol dan polaritas molekul.
5. Praktikum frekuensi vibrasi dan spektrum IR.
6. Praktikum energi reaksi, termodinamika, dan efek pelarut.
7. Praktikum analisis output, validasi data, dan otomatisasi laporan dengan Python.

Meskipun setiap praktikum memiliki fokus berbeda, semua laporan harus mengikuti prinsip dasar yang sama:

tujuan jelas
 → teori relevan
 → metode tercatat
 → input terdokumentasi
 → output divalidasi
 → data disajikan
 → grafik dijelaskan
 → analisis argumentatif
 → kesimpulan berbasis bukti
 → lampiran lengkap

Dengan template yang seragam, mahasiswa akan terbiasa menyusun laporan bukan sekadar sebagai tugas, tetapi sebagai dokumen ilmiah.

1. Pengertian Template Laporan Praktikum

Template laporan praktikum adalah susunan bagian laporan yang telah ditetapkan agar setiap praktikum dilaporkan dengan pola yang sama. Template berisi bagian-bagian wajib, urutan penulisan, jenis data yang harus dimasukkan, serta standar minimal analisis.

Dalam konteks buku ini, template laporan memiliki tiga fungsi utama.

Pertama, template berfungsi sebagai **panduan penulisan**. Mahasiswa tidak perlu bingung menentukan bagian apa saja yang harus ada dalam laporan.

Kedua, template berfungsi sebagai **alat validasi**. Jika laporan mengikuti template, maka data penting seperti metode, basis set, charge, multiplicity, status konvergensi, dan frekuensi imajiner tidak mudah terlewat.

Ketiga, template berfungsi sebagai **alat penilaian**. Dosen atau instruktur dapat menilai laporan secara lebih objektif karena seluruh mahasiswa menggunakan struktur yang sama.

Dengan demikian, template laporan bukan pembatas kreativitas ilmiah. Template justru membantu mahasiswa berpikir lebih teratur.

2. Latar Historis Penulisan Laporan Ilmiah

Tradisi laporan ilmiah berkembang dari kebutuhan untuk mendokumentasikan percobaan agar dapat diperiksa, diulang, dan dikritik oleh orang lain. Dalam sains modern, hasil penelitian tidak dianggap cukup hanya karena peneliti menyatakan bahwa eksperimen atau perhitungan telah dilakukan. Hasil harus ditulis dalam bentuk yang memungkinkan pembaca mengetahui apa yang dilakukan, bagaimana caranya, apa hasilnya, apakah hasil itu valid, dan bagaimana hasil tersebut ditafsirkan.

Dalam kimia eksperimen, laporan praktikum biasanya memuat tujuan, teori, alat dan bahan, prosedur, data pengamatan, perhitungan, pembahasan, kesimpulan, dan daftar pustaka. Dalam kimia komputasi, struktur laporan memiliki karakter khusus karena “alat dan bahan” berubah menjadi software, metode, basis set, file input, file output, parameter perhitungan, dan data digital.

Karena itu, laporan kimia komputasi harus memuat hal-hal yang tidak selalu muncul dalam laporan kimia basah, misalnya:

1. Nama software dan versi jika tersedia.
2. Metode komputasi.
3. Basis set.
4. Charge dan multiplicity.
5. Jenis job.
6. Model pelarut.
7. Status SCF.
8. Status optimasi.
9. Frekuensi imajiner.
10. File input dan output.
11. Data CSV.
12. Kode Python.
13. Grafik hasil otomatisasi.

Literatur kimia komputasi dan panduan penggunaan software seperti Gaussian, ORCA, GAMESS, Psi4, PySCF, serta buku-buku seperti karya Jensen, Cramer, Leach, Lewars, dan Foresman-Frisch menekankan pentingnya dokumentasi metode dan validasi hasil. Dalam konteks pendidikan, prinsip ini diterjemahkan menjadi template laporan yang mudah digunakan mahasiswa.

3. Prinsip Umum Template Laporan

Template laporan praktikum dalam buku ini mengikuti beberapa prinsip umum.

Pertama, laporan harus **reproduksibel**. Artinya, orang lain dapat mengulang perhitungan berdasarkan informasi yang ditulis dalam laporan.

Kedua, laporan harus **terverifikasi**. Artinya, data yang disajikan berasal dari output yang telah diperiksa.

Ketiga, laporan harus **terstruktur**. Artinya, bagian-bagian laporan disusun dalam urutan yang jelas.

Keempat, laporan harus **berbasis bukti**. Artinya, kesimpulan harus didukung oleh tabel, grafik, output, dan analisis.

Kelima, laporan harus **kritis**. Artinya, mahasiswa harus menyebutkan keterbatasan metode, basis set, model pelarut, pendekatan harmonik, atau faktor lain yang memengaruhi hasil.

Keenam, laporan harus **komunikatif**. Artinya, grafik, tabel, dan narasi harus mudah dibaca.

Ketujuh, laporan harus **jujur secara akademik**. Artinya, mahasiswa tidak boleh mengubah data agar sesuai harapan, tidak boleh menyembunyikan kegagalan perhitungan, dan tidak boleh mengklaim hasil yang tidak didukung output.

Prinsip tersebut dapat diringkas sebagai berikut:

```
rapi
+ lengkap
+ valid
+ kritis
+ reproduibel
= laporan praktikum kimia komputasi yang baik
```

4. Struktur Umum Laporan Praktikum

Struktur umum laporan praktikum yang dianjurkan adalah:

```
HALAMAN JUDUL
ABSTRAK SINGKAT
I. TUJUAN PRAKTIKUM
II. DASAR TEORI
III. SOFTWARE, METODE, DAN PARAMETER
IV. STRUKTUR INPUT DAN DATA AWAL
V. PROSEDUR KOMPUTASI
VI. HASIL OUTPUT
VII. VALIDASI DATA
VIII. TABEL DAN GRAFIK
IX. ANALISIS DAN PEMBAHASAN
X. KESIMPULAN
XI. PERTANYAAN DISKUSI
XII. LAMPIRAN
```

Struktur ini dapat digunakan untuk semua praktikum dengan penyesuaian kecil sesuai topik. Misalnya, pada praktikum spektrum IR, bagian grafik memuat spektrum IR. Pada praktikum energi

reaksi, bagian grafik memuat diagram energi reaksi. Pada praktikum HOMO-LUMO, bagian grafik memuat diagram orbital.

5. Halaman Judul

Halaman judul memuat identitas praktikum dan identitas mahasiswa.

Format yang dianjurkan:

LAPORAN PRAKTIKUM KIMIA KUANTUM

Judul Praktikum:

[isi judul praktikum]

Nama:

[isi nama mahasiswa]

NIM:

[isi NIM]

Kelas:

[isi kelas]

Program Studi:

[isi program studi]

Dosen/Instruktur:

[isi nama dosen/instruktur]

Tanggal Praktikum:

[isi tanggal]

Tanggal Pengumpulan:

[isi tanggal]

Judul praktikum harus spesifik. Jangan hanya menulis “Laporan Praktikum 1”. Tulis topiknya, misalnya:

Optimasi Geometri Molekul Air dan Karbon Dioksida

atau:

Analisis Frekuensi Vibrasi dan Spektrum IR Formaldehida

Judul yang jelas membantu pembaca memahami fokus laporan sejak awal.

6. Abstrak Singkat

Abstrak singkat berisi ringkasan tujuan, metode, sistem yang dianalisis, hasil utama, dan kesimpulan. Panjang abstrak cukup satu paragraf pendek, sekitar 100–150 kata.

Format abstrak:

Praktikum ini bertujuan untuk ...
 Perhitungan dilakukan menggunakan ...
 Sistem yang dianalisis adalah ...
 Hasil utama menunjukkan bahwa ...
 Berdasarkan validasi output, ...
 Kesimpulan utama dari praktikum ini adalah ...

Contoh abstrak untuk praktikum energi reaksi:

Praktikum ini bertujuan untuk menghitung energi reaksi, entalpi, energi bebas Gibbs, dan efek pelarut pada beberapa sistem model. Data energi diperoleh dari output komputasi dan diolah menggunakan Python untuk menghitung ΔE , ΔH , dan ΔG . Hasil menunjukkan bahwa energi elektronik tidak selalu sama dengan energi bebas Gibbs karena adanya kontribusi entropi. Pada sistem dimerisasi, pembentukan ikatan hidrogen dapat menguntungkan secara entalpi, tetapi penalti entropi dapat membuat ΔG kurang menguntungkan. Dengan demikian, analisis termodinamika harus membedakan energi elektronik, entalpi, dan Gibbs.

Abstrak tidak perlu terlalu panjang, tetapi harus mencerminkan isi laporan.

7. Tujuan Praktikum

Bagian tujuan praktikum menjelaskan kemampuan yang ingin dicapai. Tujuan harus ditulis dalam bentuk operasional.

Contoh tujuan:

1. Menjalankan optimasi geometri molekul menggunakan metode komputasi tertentu.
2. Membaca energi total dari output komputasi.
3. Memvalidasi struktur berdasarkan status konvergensi dan frekuensi imajiner.
4. Mengolah data output menggunakan Python.
5. Menyajikan hasil dalam bentuk tabel dan grafik.
6. Menafsirkan hasil berdasarkan konsep kimia kuantum.

Tujuan tidak boleh terlalu umum. Tujuan seperti “memahami kimia komputasi” terlalu luas. Tujuan harus menunjukkan kegiatan nyata yang dilakukan mahasiswa.

8. Dasar Teori

Bagian dasar teori memuat konsep inti yang diperlukan untuk memahami praktikum. Dasar teori tidak perlu menjadi kajian pustaka panjang, tetapi harus cukup untuk menjelaskan data yang dianalisis.

Isi dasar teori dapat mencakup:

1. Konsep metode komputasi.
2. Energi elektronik.
3. Basis set.
4. Optimasi geometri.
5. HOMO-LUMO.
6. Momen dipol.
7. Frekuensi vibrasi.
8. Spektrum IR.
9. Entalpi, entropi, dan Gibbs.
10. Model pelarut.
11. Parsing output.
12. Visualisasi data.

Dasar teori harus relevan dengan praktikum. Jika praktikum membahas IR, teori harus memuat frekuensi vibrasi dan intensitas IR. Jika praktikum membahas energi reaksi, teori harus memuat ΔE , ΔH , ΔG , dan entropi.

Contoh kalimat dasar teori:

Energi bebas Gibbs digunakan untuk menilai kecenderungan termodinamika suatu reaksi pada suhu dan tekanan tetap. Nilai ΔG dihitung dari selisih energi bebas Gibbs produk dan reaktan. Jika ΔG bernilai negatif, produk lebih menguntungkan secara termodinamika pada kondisi perhitungan.

9. Software, Metode, dan Parameter

Bagian ini wajib ada dalam laporan kimia komputasi. Tanpa bagian ini, hasil tidak dapat direproduksi.

Format tabel:

Komponen	Keterangan
Software	ChemCompute / ORCA / Gaussian / GAMESS / Psi4 / PySCF
Versi software	Jika tersedia
Metode	HF / B3LYP / PBE0 / MP2 / lainnya
Basis set	STO-3G / 6-31G(d) / def2-SVP / lainnya
Jenis job	Opt / Freq / Opt Freq / Single Point

Charge	0 / +1 / -1
Multiplicity	1 / 2 / 3
Model pelarut	Gas / PCM / CPCM / SMD / lainnya
Suhu	298,15 K jika digunakan
Satuan energi	Hartree, kJ/mol
Pustaka Python	pandas, matplotlib, numpy jika digunakan

Contoh penulisan:

Perhitungan dilakukan menggunakan metode B3LYP dengan basis set 6-31G(d). Jenis job yang digunakan adalah Opt Freq. Seluruh spesies netral dihitung dengan charge 0 dan multiplicity 1. Data hasil output kemudian diolah menggunakan Python dengan pustaka pandas dan matplotlib.

Bagian ini sangat penting untuk reproduibilitas.

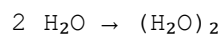
10. Struktur Input dan Data Awal

Bagian ini memuat struktur molekul atau spesies yang digunakan. Data dapat berupa gambar struktur, rumus molekul, nama file input, atau koordinat XYZ.

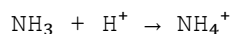
Format tabel:

Molekul/Spesies	Nama File Input	Charge	Multiplicity	Keterangan
H ₂ O	h2o.xyz	0	1	Molekul air
CO ₂	co2.xyz	0	1	Molekul linear
NH ₄ ⁺	nh4_plus.xyz	+1	1	Ion amonium
Etanol A	etanol_A.xyz	0	1	Konformer A

Jika laporan membahas reaksi, tuliskan persamaan reaksi:



atau:



Persamaan reaksi penting agar koefisien stoikiometri jelas.

11. Prosedur Komputasi

Bagian prosedur menjelaskan langkah kerja. Prosedur harus cukup detail agar dapat diulang, tetapi tidak perlu terlalu panjang.

Format umum:

1. Struktur molekul dibuat menggunakan editor molekul.
2. Struktur awal diperiksa secara visual.
3. Charge dan multiplicity ditentukan.
4. Metode dan basis set dipilih.
5. Job optimasi geometri dijalankan.
6. Jika diperlukan, job frekuensi dijalankan.
7. File output disimpan.
8. Energi, geometri, frekuensi, momen dipol, atau orbital dibaca dari output.
9. Data divalidasi.
10. Data diolah menggunakan Python.
11. Grafik dan tabel hasil dibuat.
12. Hasil dianalisis dalam laporan.

Untuk praktikum Python, prosedur dapat mencakup:

1. File output dimasukkan ke folder output_log.
2. Skrip parsing_output_komputasi.py dijalankan.
3. Ringkasan output disimpan dalam CSV.
4. CSV dibaca untuk membuat grafik laporan.
5. Grafik disimpan dalam folder grafik.

Prosedur harus mencerminkan pekerjaan yang benar-benar dilakukan.

12. Hasil Output

Bagian hasil output memuat data mentah atau ringkasan data dari output komputasi. Data yang dimasukkan harus relevan dengan praktikum.

Contoh tabel hasil output:

Molekul	Energi (Hartree)	Total	Momen (Debye)	Dipol	Frekuensi Imajiner	Status Job
H ₂ O	...		...		0	Selesai normal
CO ₂	...		...		0	Selesai normal
NH ₃	...		...		0	Selesai normal

Untuk praktikum orbital:

Molekul	HOMO (eV)	LUMO (eV)	Gap (eV)
Formaldehida	...	...	...
Etanol	...	...	...

Untuk praktikum energi reaksi:

Sistem	ΔE (kJ/mol)	ΔH (kJ/mol)	ΔG (kJ/mol)
cis-trans 2-butena	...	...	...
Dimer air	...	...	...

Data hasil output harus disajikan dalam tabel yang rapi dan memiliki satuan.

13. Validasi Data

Bagian validasi data wajib ada. Bagian ini menunjukkan apakah hasil praktikum layak dianalisis.

Tabel validasi:

Kriteria Validasi	Status	Keterangan
Job selesai normal	Ya/Tidak	Berdasarkan pesan output
SCF konvergen	Ya/Tidak	Energi elektronik stabil
Optimasi konvergen	Ya/Tidak	Struktur mencapai kriteria
Frekuensi imajiner	Ada/Tidak	Untuk minimum harus tidak ada
Geometri masuk akal	Ya/Tidak	Dicek visual dan parameter
Charge benar	Ya/Tidak	Sesuai spesies
Multiplicity benar	Ya/Tidak	Sesuai keadaan spin
Metode konsisten	Ya/Tidak	Untuk perbandingan energi
Satuan benar	Ya/Tidak	Hartree/kJ/mol/eV
Data layak dianalisis	Ya/Tidak	Kesimpulan validasi

Contoh narasi validasi:

Seluruh perhitungan selesai normal dan optimasi geometri dinyatakan konvergen. Perhitungan frekuensi tidak menunjukkan frekuensi imajiner, sehingga struktur dapat dianggap sebagai minimum lokal. Charge dan multiplicity telah sesuai dengan spesies yang dihitung. Oleh karena itu, data digunakan untuk analisis lanjutan.

Jika ada masalah, tuliskan secara jujur:

File dimer_air.log menunjukkan satu frekuensi imajiner. Karena struktur dimer air dimaksudkan sebagai minimum, data tersebut tidak digunakan untuk analisis energi dan perlu dilakukan optimasi ulang.

14. Tabel dan Grafik

Bagian tabel dan grafik memuat hasil yang telah diolah. Grafik harus diberi judul, label sumbu, dan satuan.

Jenis grafik sesuai praktikum:

Praktikum	Grafik yang Dianjurkan
Optimasi geometri	Grafik energi selama optimasi
Metode dan basis set	Grafik perbandingan energi
HOMO-LUMO	Diagram orbital
Momen dipol	Grafik momen dipol
Frekuensi IR	Spektrum IR
Energi reaksi	Diagram energi reaksi
Efek pelarut	Grafik perbandingan gas-pelarut
Parsing output	Tabel ringkasan CSV
Otomatisasi laporan	Grafik dari CSV

Setiap grafik harus diberi keterangan.

Contoh keterangan gambar:

Gambar 1. Diagram energi Gibbs untuk reaksi hidrasi formaldehida. Energi reaktan dijadikan acuan 0 kJ/mol.

atau:

Gambar 2. Spektrum IR komputasi formaldehida berdasarkan data frekuensi dan intensitas hasil perhitungan.

Grafik tanpa keterangan akan sulit dipahami.

15. Analisis dan Pembahasan

Bagian analisis adalah inti laporan. Di sinilah mahasiswa menjelaskan makna hasil. Analisis tidak boleh hanya mengulang angka dalam tabel.

Analisis harus mencakup:

1. Apa hasil utama?
2. Mengapa hasil tersebut muncul?
3. Apakah hasil sesuai teori?
4. Apakah hasil sesuai literatur atau tren umum?
5. Apakah ada perbedaan antara data komputasi dan teori?

6. Apakah ada keterbatasan metode?
7. Apa implikasi kimianya?

Format analisis:

Berdasarkan tabel ...
 terlihat bahwa ...
 Hal ini menunjukkan ...
 Secara kimia, hasil ini dapat dijelaskan oleh ...
 Namun, hasil ini perlu ditafsirkan hati-hati karena ...

Contoh analisis HOMO-LUMO:

Formaldehida memiliki celah HOMO-LUMO lebih kecil dibanding etanol. Hal ini dapat dikaitkan dengan keberadaan gugus karbonil yang memiliki sistem π dan π^* . Celah yang lebih kecil menunjukkan bahwa formaldehida lebih mudah mengalami interaksi elektronik tertentu. Namun, reaktivitas tidak boleh disimpulkan hanya dari gap HOMO-LUMO karena faktor struktur, muatan, dan lingkungan juga berperan.

Contoh analisis IR:

Spektrum IR formaldehida menunjukkan puncak kuat pada wilayah regangan C=O. Hal ini sesuai dengan keberadaan gugus karbonil. Perbedaan nilai frekuensi terhadap eksperimen dapat disebabkan oleh pendekatan osilator harmonik dan belum digunakannya faktor skala.

16. Kesimpulan

Kesimpulan harus menjawab tujuan praktikum. Kesimpulan tidak boleh berisi data baru. Kesimpulan harus singkat, jelas, dan berbasis hasil.

Format kesimpulan:

1. Perhitungan menunjukkan bahwa ...
2. Validasi output menunjukkan bahwa ...
3. Grafik menunjukkan bahwa ...
4. Berdasarkan analisis, ...

Contoh:

1. Struktur H₂O berhasil dioptimasi dan menunjukkan geometri bengkok.
2. Tidak ditemukan frekuensi imajiner, sehingga struktur dapat dianggap sebagai minimum lokal.
3. Momen dipol H₂O tidak nol, sesuai dengan bentuk molekul yang tidak simetris.
4. Hasil praktikum menunjukkan hubungan antara geometri molekul dan polaritas.

Kesimpulan harus konsisten dengan hasil dan analisis.

17. Pertanyaan Diskusi

Bagian pertanyaan diskusi berisi jawaban atas pertanyaan yang diberikan dalam setiap praktikum. Jawaban harus argumentatif dan berbasis data.

Format jawaban:

Pertanyaan:
[isi pertanyaan]

Jawaban:
[definisi singkat]
[penjelasan teori]
[kaitkan dengan hasil praktikum]
[contoh dari data]
[kesimpulan]

Contoh:

Pertanyaan:
Mengapa ΔE tidak selalu sama dengan ΔG ?

Jawaban:
 ΔE adalah perubahan energi elektronik, sedangkan ΔG adalah perubahan energi bebas Gibbs yang memasukkan kontribusi entalpi dan entropi. Karena $\Delta G = \Delta H - T\Delta S$, perubahan entropi dapat membuat ΔG berbeda dari ΔE . Pada dimerisasi air, pembentukan ikatan hidrogen dapat membuat ΔE negatif, tetapi penggabungan dua molekul menjadi satu dimer menurunkan entropi sehingga ΔG dapat menjadi kurang menguntungkan.

Jawaban diskusi tidak boleh terlalu singkat.

18. Lampiran

Lampiran memuat dokumen pendukung yang terlalu panjang jika dimasukkan ke bagian utama laporan.

Lampiran dapat berisi:

1. File input.
2. Potongan output penting.
3. Koordinat XYZ.
4. Data CSV.
5. Kode Python.
6. Gambar struktur.
7. Grafik tambahan.
8. Tabel validasi lengkap.

9. Pesan error dan perbaikannya.
10. Catatan parsing output.

Format lampiran:

Lampiran 1. File input
 Lampiran 2. Potongan output energi
 Lampiran 3. Tabel CSV hasil parsing
 Lampiran 4. Kode Python
 Lampiran 5. Grafik hasil analisis

Lampiran membantu menjaga laporan utama tetap rapi, tetapi data penting tetap terdokumentasi.

19. Template Lengkap Laporan Praktikum

Berikut template lengkap yang dapat digunakan mahasiswa.

LAPORAN PRAKTIKUM KIMIA KUANTUM

Judul Praktikum:
 Nama:
 NIM:
 Kelas:
 Program Studi:
 Dosen/Instruktur:
 Tanggal Praktikum:
 Tanggal Pengumpulan:

ABSTRAK

[Tulis ringkasan singkat tujuan, metode, sistem, hasil utama, dan kesimpulan.]

I. TUJUAN PRAKTIKUM

1. ...
2. ...
3. ...

II. DASAR TEORI

[Tulis konsep penting yang relevan dengan praktikum.]

III. SOFTWARE, METODE, DAN PARAMETER

Software:
 Versi:
 Metode:
 Basis set:
 Jenis job:
 Charge:
 Multiplicity:

Model pelarut:
Suhu:
Satuan energi:
Pustaka Python:

IV. STRUKTUR INPUT DAN DATA AWAL

[Tampilkan tabel molekul, file input, charge, multiplicity, dan keterangan.]

V. PROSEDUR KOMPUTASI

1. ...
2. ...
3. ...

VI. HASIL OUTPUT

[Tampilkan energi, geometri, momen dipol, frekuensi, HOMO-LUMO, atau data lain sesuai praktikum.]

VII. VALIDASI DATA

[Tampilkan tabel validasi job, konvergensi, frekuensi imajiner, geometri, charge, multiplicity, dan satuan.]

VIII. TABEL DAN GRAFIK

[Tampilkan tabel hasil olahan dan grafik Python.]

IX. ANALISIS DAN PEMBAHASAN

[Jelaskan makna hasil, hubungan dengan teori, validitas, keterbatasan, dan implikasi kimia.]

X. KESIMPULAN

1. ...
2. ...
3. ...

XI. PERTANYAAN DISKUSI

Pertanyaan 1:

Jawaban:

Pertanyaan 2:

Jawaban:

XII. LAMPIRAN

- Lampiran 1. File input
- Lampiran 2. Potongan output
- Lampiran 3. CSV hasil analisis
- Lampiran 4. Kode Python
- Lampiran 5. Grafik tambahan

Template ini dapat digunakan untuk semua praktikum dengan menyesuaikan isi setiap bagian.

20. Template Khusus Praktikum Optimasi Geometri

Untuk praktikum optimasi geometri, fokus laporan adalah struktur akhir.

Bagian hasil harus memuat:

1. Geometri awal.
2. Geometri akhir.
3. Energi sebelum dan sesudah optimasi.
4. Panjang ikatan.
5. Sudut ikatan.
6. Status konvergensi.

Pertanyaan analisis utama:

Apakah struktur akhir masuk akal?
 Apakah energi menurun setelah optimasi?
 Apakah panjang ikatan sesuai teori?
 Apakah struktur akhir dapat dianggap minimum?

Contoh kesimpulan:

Optimasi geometri berhasil menurunkan energi sistem dan menghasilkan struktur akhir yang masuk akal secara kimia. Struktur akhir digunakan sebagai dasar untuk analisis lanjutan.

21. Template Khusus Praktikum Metode dan Basis Set

Untuk praktikum metode dan basis set, fokus laporan adalah perbandingan hasil berdasarkan level teori.

Bagian hasil harus memuat:

1. Metode yang dibandingkan.
2. Basis set yang digunakan.
3. Energi total.
4. Geometri hasil.
5. Perbandingan panjang ikatan atau sudut.
6. Grafik perbandingan energi.

Pertanyaan analisis utama:

Bagaimana metode memengaruhi energi?
 Bagaimana basis set memengaruhi geometri?
 Apakah hasil berubah signifikan?
 Metode mana yang paling sesuai untuk tujuan praktikum?

Contoh kesimpulan:

Perbandingan metode menunjukkan bahwa energi dan geometri dipengaruhi oleh level teori. Basis set yang lebih fleksibel memberikan deskripsi struktur yang lebih baik, tetapi membutuhkan biaya komputasi lebih besar.

22. Template Khusus Praktikum HOMO-LUMO

Untuk praktikum HOMO-LUMO, fokus laporan adalah sifat elektronik molekul.

Bagian hasil harus memuat:

1. Energi HOMO.
2. Energi LUMO.
3. Gap HOMO-LUMO.
4. Diagram orbital.
5. Visualisasi orbital jika tersedia.
6. Interpretasi reaktivitas kualitatif.

Pertanyaan analisis utama:

Molekul mana yang memiliki gap paling kecil?
 Apa makna HOMO sebagai donor elektron?
 Apa makna LUMO sebagai akseptor elektron?
 Apakah interpretasi reaktivitas cukup hanya dari gap?

Contoh kesimpulan:

Analisis HOMO-LUMO menunjukkan perbedaan sifat elektronik antar molekul. Molekul dengan gap lebih kecil cenderung lebih mudah mengalami interaksi elektronik, tetapi interpretasi reaktivitas harus tetap mempertimbangkan struktur dan konteks reaksi.

23. Template Khusus Praktikum Momen Dipol

Untuk praktikum momen dipol, fokus laporan adalah polaritas molekul.

Bagian hasil harus memuat:

1. Geometri molekul.
2. Komponen dipol x, y, z jika tersedia.
3. Momen dipol total.

4. Hubungan momen dipol dengan simetri.
5. Grafik perbandingan momen dipol.

Pertanyaan analisis utama:

Mengapa H₂O memiliki momen dipol?
 Mengapa CO₂ memiliki momen dipol mendekati nol?
 Bagaimana bentuk molekul memengaruhi polaritas?
 Bagaimana momen dipol berkaitan dengan efek pelarut?

Contoh kesimpulan:

Momen dipol dipengaruhi oleh distribusi muatan dan geometri molekul. Molekul dengan bentuk tidak simetris seperti H₂O memiliki momen dipol tidak nol, sedangkan molekul simetris seperti CO₂ memiliki momen dipol total mendekati nol.

24. Template Khusus Praktikum Frekuensi Vibrasi dan IR

Untuk praktikum frekuensi vibrasi dan spektrum IR, fokus laporan adalah mode vibrasi dan spektrum.

Bagian hasil harus memuat:

1. Frekuensi vibrasi.
2. Intensitas IR.
3. Frekuensi imajiner.
4. Spektrum IR.
5. Identifikasi puncak utama.
6. Penjelasan gugus fungsi.

Pertanyaan analisis utama:

Apakah struktur memiliki frekuensi imajiner?
 Mode vibrasi apa yang paling kuat?
 Puncak mana yang berkaitan dengan O-H, N-H, C-H, C=O, atau C-O?
 Mengapa frekuensi komputasi dapat berbeda dari eksperimen?

Contoh kesimpulan:

Spektrum IR komputasi menunjukkan puncak vibrasi yang sesuai dengan gugus fungsi molekul. Tidak adanya frekuensi imajiner menunjukkan bahwa struktur hasil optimasi dapat dianggap sebagai minimum lokal.

25. Template Khusus Praktikum Energi Reaksi dan Termodinamika

Untuk praktikum energi reaksi, fokus laporan adalah ΔE , ΔH , ΔG , dan efek pelarut.

Bagian hasil harus memuat:

1. Energi reaktan.
2. Energi produk.
3. Koefisien stoikiometri.
4. ΔE .
5. ΔH .
6. ΔG .
7. Diagram energi reaksi.
8. Perbandingan gas dan pelarut jika ada.

Pertanyaan analisis utama:

Mengapa ΔE tidak selalu sama dengan ΔG ?
 Apa peran entropi?
 Apakah produk lebih stabil secara Gibbs?
 Bagaimana pelarut memengaruhi spesies bermuatan?
 Apa keterbatasan model pelarut kontinu?

Contoh kesimpulan:

Perhitungan energi reaksi menunjukkan bahwa energi elektronik, entalpi, dan energi bebas Gibbs memberikan informasi yang berbeda. ΔG menjadi dasar utama untuk menilai kecenderungan termodinamika, sedangkan efek pelarut dianalisis melalui perubahan kestabilan relatif reaktan dan produk.

26. Template Khusus Praktikum Analisis Output dan Otomatisasi Python

Untuk praktikum Bab 11, fokus laporan adalah pembacaan output, validasi, parsing, grafik, dan otomatisasi laporan.

Bagian hasil harus memuat:

1. Ringkasan file output.
2. Tabel hasil parsing.
3. Status validasi.
4. Grafik laporan.
5. CSV hasil parsing.
6. Kode Python yang digunakan.
7. Catatan kesalahan jika ada.

Pertanyaan analisis utama:

Apakah semua output selesai normal?
 Apakah parser berhasil membaca data penting?
 Data apa yang tidak ditemukan?
 Apakah grafik sesuai CSV?
 Apa keterbatasan parsing otomatis?

Bagaimana Python membantu reproduibilitas laporan?

Contoh kesimpulan:

Python dapat membantu membaca output, membuat ringkasan CSV, dan menghasilkan grafik laporan secara otomatis. Namun, hasil parsing tetap harus divalidasi secara manual karena format output berbeda antarsoftware dan tidak semua informasi kimia dapat dinilai otomatis.

27. Rubrik Penilaian Laporan Praktikum

Agar laporan dapat dinilai secara objektif, rubrik berikut dapat digunakan.

Aspek	Bobot Anjuran	Kriteria
Identitas dan format	10%	Judul, identitas, struktur laporan rapi
Tujuan dan teori	15%	Tujuan jelas, teori relevan
Metode dan parameter	15%	Software, metode, basis set, charge, multiplicity lengkap
Hasil output	15%	Data utama lengkap dan bersatuan
Validasi data	15%	Konvergensi, frekuensi, geometri, status job diperiksa
Grafik dan tabel	10%	Grafik jelas, label dan satuan benar
Analisis	15%	Argumentatif, berbasis data, kritis
Kesimpulan dan lampiran	5%	Kesimpulan menjawab tujuan, lampiran lengkap

Rubrik ini dapat disesuaikan oleh dosen, tetapi komponen validasi dan analisis sebaiknya tetap diberi bobot tinggi.

28. Kesalahan Umum dalam Menulis Laporan Praktikum

Beberapa kesalahan umum dalam laporan praktikum adalah:

Kesalahan	Dampak
Tidak mencantumkan metode	Hasil tidak reproduibel
Tidak mencantumkan basis set	Perhitungan tidak jelas
Tidak mencantumkan charge	Sistem tidak dapat diverifikasi
Tidak memeriksa konvergensi	Data belum tentu valid
Tidak memeriksa frekuensi imajiner	Struktur minimum belum pasti
Grafik tanpa satuan	Interpretasi kabur
Tabel tanpa keterangan	Data sulit dibaca
Analisis hanya mengulang angka	Tidak menunjukkan pemahaman

Kesimpulan tidak menjawab tujuan	Laporan tidak fokus
Lampiran tidak lengkap	Data sulit diverifikasi
Kode Python tidak disertakan	Otomatisasi tidak reproduibel
CSV tidak disimpan	Grafik tidak dapat diperiksa ulang

Kesalahan ini harus dihindari dengan mengikuti template.

29. Integrasi Template dengan Python

Template laporan dapat diintegrasikan dengan Python. Data hasil parsing dapat dimasukkan otomatis ke tabel. Grafik dapat dibuat otomatis dari CSV. Bahkan, pada tahap lanjut, Python dapat digunakan untuk membuat draf laporan dalam format Markdown, HTML, atau DOCX.

Alur integrasi:

```
output log
→ parsing Python
→ ringkasan CSV
→ grafik Python
→ tabel laporan
→ template laporan
→ dokumen akhir
```

Integrasi ini membantu mahasiswa mengurangi pekerjaan berulang. Namun, narasi analisis tetap harus ditulis dengan pemahaman kimia. Python dapat membantu menyusun data, tetapi tidak menggantikan penalaran ilmiah.

30. Contoh Narasi Metodologi dalam Laporan

Contoh narasi yang dapat digunakan:

Perhitungan dilakukan menggunakan metode B3LYP dengan basis set 6-31G(d). Struktur molekul dioptimasi terlebih dahulu, kemudian dilakukan perhitungan frekuensi untuk memvalidasi struktur minimum. Output komputasi dibaca secara manual dan diparsing menggunakan Python. Data energi, momen dipol, frekuensi, dan status job disimpan dalam CSV. Grafik dibuat menggunakan Python berdasarkan data yang telah divalidasi.

Contoh narasi validasi:

Validasi dilakukan dengan memeriksa status job, konvergensi SCF, konvergensi optimasi, frekuensi imajiner, geometri akhir, charge, dan multiplicity. Data hanya digunakan dalam analisis jika perhitungan selesai normal, struktur konvergen, dan tidak terdapat frekuensi imajiner untuk struktur minimum.

Contoh narasi keterbatasan:

Hasil praktikum ditafsirkan sebagai hasil model komputasi pada level teori yang digunakan. Perbedaan dengan data eksperimen dapat muncul karena keterbatasan basis set, pendekatan harmonik, model pelarut, atau kondisi perhitungan yang tidak identik dengan kondisi eksperimen.

31. Implikasi Keilmuan Template Laporan

Template laporan memiliki implikasi keilmuan penting. Laporan yang baik tidak hanya menyajikan hasil, tetapi juga menunjukkan bagaimana hasil itu diperoleh dan divalidasi. Dalam kimia komputasi, dokumentasi metode sangat penting karena hasil bergantung pada level teori, basis set, charge, multiplicity, model pelarut, dan jenis job.

Template laporan membantu memastikan bahwa informasi tersebut tidak hilang. Dengan demikian, laporan menjadi dokumen ilmiah yang dapat dibaca, diperiksa, dibandingkan, dan diulang.

Template juga membantu membedakan antara:

hasil software
→ keluaran numerik

laporan ilmiah
→ data tervalidasi + analisis + kesimpulan

Perbedaan ini penting dalam pendidikan kimia komputasi.

32. Implikasi Pedagogis Template Laporan

Secara pedagogis, template laporan melatih mahasiswa menulis secara sistematis. Banyak mahasiswa dapat menjalankan software, tetapi kesulitan menjelaskan hasilnya. Template membantu mereka membangun alur berpikir.

Keterampilan yang dilatih:

1. Menyusun tujuan praktikum.
2. Menulis teori relevan.
3. Mendokumentasikan metode.
4. Menyajikan data output.
5. Memvalidasi hasil.
6. Membuat tabel dan grafik.
7. Menulis analisis argumentatif.
8. Menarik kesimpulan berbasis data.
9. Melampirkan kode dan output.

10. Menjaga reproduibilitas.

Bagi calon guru kimia, keterampilan ini sangat penting. Guru tidak hanya harus memahami hasil komputasi, tetapi juga mampu membimbing peserta didik menulis laporan ilmiah yang benar.

33. Analisis Kritis terhadap Template

Template laporan sangat membantu, tetapi tidak boleh digunakan secara mekanis. Ada risiko mahasiswa hanya mengisi bagian-bagian template tanpa memahami isinya. Misalnya, mahasiswa dapat menulis “validasi berhasil” tanpa benar-benar memeriksa output. Mahasiswa juga dapat menempel grafik tanpa menjelaskan maknanya.

Karena itu, template harus dipahami sebagai kerangka berpikir, bukan formulir kosong.

Prinsipnya:

template memberi struktur
tetapi analisis tetap memerlukan pemahaman

Dosen atau instruktur perlu menekankan bahwa kualitas laporan bukan hanya ditentukan oleh kelengkapan bagian, tetapi oleh kedalaman analisis dan ketepatan validasi.

34. Sintesis Subbab 11.6

Template laporan praktikum dalam Bab 11 dirancang untuk menyeragamkan pelaporan tujuh mata praktikum kimia kuantum dan kimia komputasi. Template ini mencakup halaman judul, abstrak, tujuan, dasar teori, software dan metode, struktur input, prosedur, hasil output, validasi data, tabel dan grafik, analisis, kesimpulan, pertanyaan diskusi, dan lampiran.

Alur sintesis template laporan adalah:

identitas praktikum
→ tujuan
→ teori
→ metode
→ input
→ prosedur
→ output
→ validasi
→ grafik
→ analisis
→ kesimpulan
→ lampiran

Dengan template ini, mahasiswa dilatih menyusun laporan yang rapi, valid, kritis, komunikatif, dan reproduibel. Template ini juga mengintegrasikan seluruh keterampilan Bab 11: membaca output, memvalidasi data, mengenali kesalahan umum, melakukan parsing dengan Python, membuat grafik, dan menyusun laporan ilmiah.

35. Transisi ke Penutup Bab 11

Subbab 11.6 telah membahas template laporan praktikum yang seragam untuk tujuh mata praktikum. Template ini menjadi kerangka akhir yang mengintegrasikan semua keterampilan dalam Bab 11, mulai dari membaca output, memvalidasi hasil, mengenali kesalahan umum, melakukan parsing output dengan Python, membuat grafik laporan, hingga menyusun laporan praktikum yang sistematis.

Dengan selesainya subbab ini, seluruh rangkaian utama Bab 11 telah lengkap. Tahap berikutnya adalah menyusun penutup Bab 11 yang merangkum pentingnya analisis output, validasi data, otomatisasi Python, dan template laporan sebagai satu kesatuan dalam pembelajaran kimia kuantum dan kimia komputasi.

Penutup Bab 11

Bab 11 telah membahas tahap akhir yang sangat penting dalam Praktikum Kimia Kuantum, yaitu **analisis output, validasi data, pengenalan kesalahan umum, parsing output dengan Python, visualisasi grafik laporan, dan penyusunan template laporan praktikum**. Jika bab-bab sebelumnya menekankan bagaimana menjalankan perhitungan, menghitung energi, membaca sifat molekul, dan menafsirkan fenomena kimia, maka Bab 11 menempatkan mahasiswa pada tahap yang lebih matang: memastikan bahwa semua hasil tersebut benar-benar dapat dipercaya dan dapat disusun menjadi laporan ilmiah yang rapi.

Dalam kimia komputasi, hasil yang keluar dari software tidak boleh langsung diterima sebagai kebenaran. Output komputasi harus dibaca, diperiksa, divalidasi, dan ditafsirkan. Software dapat menghasilkan energi, geometri, frekuensi, momen dipol, HOMO-LUMO, atau data termodinamika, tetapi tanggung jawab ilmiah tetap berada pada pengguna. Mahasiswa harus mampu menilai apakah job selesai normal, apakah SCF konvergen, apakah optimasi geometri berhasil, apakah struktur minimum bebas dari frekuensi imajiner, apakah charge dan multiplicity benar, apakah metode dan basis set konsisten, serta apakah hasil berada dalam kisaran yang masuk akal secara kimia.

Bab ini dimulai dengan pembahasan tentang **membaca file output komputasi**. Mahasiswa diarahkan untuk mencari informasi penting seperti metode, basis set, charge, multiplicity, energi total, geometri akhir, momen dipol, frekuensi, HOMO-LUMO, dan status konvergensi. Keterampilan membaca output merupakan fondasi utama karena seluruh analisis praktikum bergantung pada kemampuan menemukan data yang benar dari file output yang sering kali panjang dan kompleks.

Setelah mampu membaca output, mahasiswa kemudian diperkenalkan pada **validasi hasil praktikum**. Validasi bukan sekadar mencentang bahwa perhitungan selesai, tetapi proses ilmiah untuk memastikan bahwa data layak digunakan. Enam kriteria utama yang ditekankan adalah job selesai normal, struktur konvergen, tidak ada frekuensi imajiner untuk struktur minimum, geometri masuk akal, charge dan multiplicity benar, serta hasil dapat dibandingkan dengan literatur atau tren teori yang relevan. Dengan validasi ini, mahasiswa belajar bahwa data komputasi harus diperlakukan secara hati-hati.

Bab ini juga membahas **kesalahan umum** yang sering terjadi dalam praktikum kimia komputasi. Kesalahan seperti struktur awal buruk, charge salah, multiplicity salah, basis set tidak sesuai, SCF tidak konvergen, salah membaca satuan energi, dan salah menafsirkan frekuensi imajiner dapat menyebabkan kesimpulan yang keliru. Pembahasan ini penting karena mahasiswa tidak hanya perlu tahu cara memperoleh hasil, tetapi juga harus tahu bagaimana mengenali hasil yang bermasalah.

Pada bagian berikutnya, Bab 11 memperkenalkan **Python untuk parsing output**. Python digunakan untuk membaca file log, mencari energi total, momen dipol, frekuensi, HOMO-LUMO, status job, charge, multiplicity, metode, dan basis set. Hasil parsing kemudian disimpan dalam CSV agar mudah diperiksa dan digunakan untuk laporan. Namun, bab ini juga menekankan bahwa parsing otomatis tidak menggantikan validasi manual. Python dapat membantu menemukan pola teks dan menyusun tabel, tetapi Python tidak dapat sepenuhnya memahami konteks kimia. Oleh karena itu, hasil parsing harus tetap diperiksa.

Selanjutnya, mahasiswa mempelajari **Python untuk grafik laporan**. Data hasil praktikum divisualisasikan dalam bentuk grafik energi, diagram orbital, spektrum IR, diagram energi reaksi, dan grafik perbandingan metode. Visualisasi ini membantu mahasiswa melihat pola yang sulit dipahami hanya dari tabel angka. Namun, grafik harus dibuat dari data yang valid, memiliki label dan satuan yang jelas, serta dijelaskan dalam narasi laporan. Grafik bukan hiasan, melainkan bagian dari argumentasi ilmiah.

Bab ini ditutup dengan pembahasan **template laporan praktikum**. Template laporan dirancang untuk menyeragamkan pelaporan tujuh mata praktikum. Template tersebut mencakup halaman judul, abstrak, tujuan, dasar teori, software dan parameter, struktur input, prosedur komputasi, hasil output, validasi data, tabel dan grafik, analisis, kesimpulan, pertanyaan diskusi, dan lampiran. Dengan template ini, mahasiswa diharapkan dapat menyusun laporan yang rapi, lengkap, kritis, dan reproduksibel.

Secara keseluruhan, Bab 11 menunjukkan bahwa kimia komputasi bukan hanya persoalan menjalankan software, tetapi juga persoalan **membaca, memeriksa, mengolah, memvisualisasikan, dan mempertanggungjawabkan data**. Mahasiswa yang baik tidak berhenti pada tahap memperoleh output. Ia harus mampu menjelaskan dari mana data diperoleh, bagaimana data divalidasi, bagaimana data diolah, bagaimana grafik dibuat, dan bagaimana kesimpulan ditarik.

Alur besar Bab 11 dapat diringkas sebagai berikut:

file output komputasi

- pembacaan informasi penting
- validasi hasil
- identifikasi kesalahan umum
- parsing dengan Python
- penyusunan CSV
- pembuatan grafik laporan
- penyusunan laporan praktikum
- kesimpulan berbasis data

Alur ini menjadi dasar kerja ilmiah dalam praktikum kimia kuantum modern. Dengan menguasai alur tersebut, mahasiswa tidak hanya menjadi pengguna software, tetapi juga menjadi analis data komputasi yang lebih teliti dan bertanggung jawab.

Bab 11 juga memperlihatkan pentingnya **reproduksibilitas**. Setiap hasil praktikum harus dapat ditelusuri kembali ke file input, file output, metode, basis set, charge, multiplicity, kode Python, CSV, dan grafik. Jika data tidak dapat ditelusuri, maka laporan menjadi lemah. Jika data dapat ditelusuri dan divalidasi, maka laporan menjadi lebih kuat secara akademik.

Dari sisi pedagogis, Bab 11 melatih mahasiswa untuk berpikir kritis. Mahasiswa belajar bahwa tidak semua output valid, tidak semua grafik benar, tidak semua angka dapat langsung ditafsirkan, dan tidak semua perhitungan yang selesai normal bebas dari masalah. Sikap kritis ini sangat penting dalam pendidikan kimia, terutama bagi calon guru dan calon peneliti yang kelak akan menggunakan simulasi molekul sebagai alat pembelajaran, penelitian, atau pengembangan bahan ajar.

Dengan demikian, capaian utama Bab 11 adalah membentuk kebiasaan ilmiah berikut:

1. Membaca output secara teliti.
2. Memvalidasi data sebelum menganalisis.
3. Mengenali kesalahan umum.
4. Menggunakan Python untuk membantu pengolahan data.
5. Membuat grafik dari data yang benar.
6. Menulis laporan berdasarkan bukti.
7. Menyimpan file input, output, CSV, grafik, dan kode secara rapi.
8. Menarik kesimpulan secara hati-hati dan bertanggung jawab.

Bab 11 menjadi penguat akhir bagi seluruh rangkaian praktikum dalam buku ini. Jika Bab 4 sampai Bab 10 membekali mahasiswa dengan pengalaman menjalankan berbagai jenis perhitungan kimia kuantum dan kimia komputasi, maka Bab 11 membekali mahasiswa dengan kemampuan untuk menilai dan melaporkan hasilnya. Inilah keterampilan yang menjadikan praktikum kimia komputasi tidak hanya bersifat teknis, tetapi juga ilmiah.

Akhirnya, melalui Bab 11 ini, mahasiswa diharapkan memiliki pemahaman bahwa kualitas praktikum tidak hanya ditentukan oleh keberhasilan menjalankan program, tetapi oleh kemampuan menyusun data menjadi pengetahuan. Output yang panjang harus diubah menjadi tabel yang jelas. Tabel harus diubah menjadi grafik yang komunikatif. Grafik harus dijelaskan dalam analisis yang kritis. Analisis harus ditutup dengan kesimpulan yang berbasis bukti. Dengan cara itulah praktikum kimia kuantum menjadi kegiatan akademik yang utuh.

Daftar Pustaka Bab 11

Atkins, P., & de Paula, J. (2010). *Atkins' Physical Chemistry* (9th ed.). Oxford University Press.

Beazley, D., & Jones, B. K. (2013). *Python Cookbook* (3rd ed.). O'Reilly Media.

Cramer, C. J. (2004). *Essentials of Computational Chemistry: Theories and Models* (2nd ed.). John Wiley & Sons.

Foresman, J. B., & Frisch, Æ. (2015). *Exploring Chemistry with Electronic Structure Methods* (3rd ed.). Gaussian, Inc.

Friedl, J. E. F. (2006). *Mastering Regular Expressions* (3rd ed.). O'Reilly Media.

Gordon, M. S., & Schmidt, M. W. (2005). Advances in electronic structure theory: GAMESS a decade later. In C. E. Dykstra, G. Frenking, K. S. Kim, & G. E. Scuseria (Eds.), *Theory and Applications of Computational Chemistry: The First Forty Years* (pp. 1167–1189). Elsevier.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., & Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585, 357–362.

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3), 90–95.

Jensen, F. (2017). *Introduction to Computational Chemistry* (3rd ed.). John Wiley & Sons.

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B. E., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J. B., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S., & Willing, C. (2016). Jupyter Notebooks—a publishing format for reproducible computational workflows. In F. Loizides & B. Schmidt (Eds.), *Positioning and Power in Academic Publishing: Players, Agents and Agendas* (pp. 87–90). IOS Press.

Leach, A. R. (2001). *Molecular Modelling: Principles and Applications* (2nd ed.). Pearson Education.

Levine, I. N. (2014). *Quantum Chemistry* (7th ed.). Pearson.

Lewars, E. G. (2016). *Computational Chemistry: Introduction to the Theory and Applications of Molecular and Quantum Mechanics* (3rd ed.). Springer.

McKinney, W. (2010). Data structures for statistical computing in Python. In S. van der Walt & J. Millman (Eds.), *Proceedings of the 9th Python in Science Conference* (pp. 56–61).

McQuarrie, D. A., & Simon, J. D. (1997). *Physical Chemistry: A Molecular Approach*. University Science Books.

Neese, F. (2012). The ORCA program system. *WIREs Computational Molecular Science*, 2(1), 73–78.

Neese, F. (2022). Software update: The ORCA program system—Version 5.0. *WIREs Computational Molecular Science*, 12(5), e1606.

O’Boyle, N. M., Tenderholt, A. L., & Langner, K. M. (2008). cclib: A library for package-independent computational chemistry algorithms. *Journal of Computational Chemistry*, 29(5), 839–845.

Parrish, R. M., Burns, L. A., Smith, D. G. A., Simmonett, A. C., DePrince, A. E., Bozkaya, U., Sokolov, A. Y., Di Remigio, R., Richard, R. M., Gonthier, J. F., James, A. M., McAlexander, H. R., Kumar, A., Saitow, M., Wang, X., Pritchard, B. P., Verma, P., Schaefer, H. F., Patkowski, K., King, R. A., Valeev, E. F., Evangelista, F. A., Turney, J. M., Crawford, T. D., & Sherrill, C. D. (2017). Psi4 1.1: An open-source electronic structure program emphasizing automation, advanced libraries, and interoperability. *Journal of Chemical Theory and Computation*, 13(7), 3185–3197.

Python Software Foundation. (n.d.). *The Python Standard Library: re—Regular expression operations*. Python Software Foundation.

Sandve, G. K., Nekrutenko, A., Taylor, J., & Hovig, E. (2013). Ten simple rules for reproducible computational research. *PLOS Computational Biology*, 9(10), e1003285.

Schmidt, M. W., Baldridge, K. K., Boatz, J. A., Elbert, S. T., Gordon, M. S., Jensen, J. H., Koseki, S., Matsunaga, N., Nguyen, K. A., Su, S., Windus, T. L., Dupuis, M., & Montgomery, J. A. (1993). General atomic and molecular electronic structure system. *Journal of Computational Chemistry*, 14(11), 1347–1363.

Sun, Q., Berkelbach, T. C., Blunt, N. S., Booth, G. H., Guo, S., Li, Z., Liu, J., McClain, J. D., Sayfutyarova, E. R., Sharma, S., Wouters, S., & Chan, G. K.-L. (2018). PySCF: The Python-based simulations of chemistry framework. *WIREs Computational Molecular Science*, 8(1), e1340.

Sun, Q., Zhang, X., Banerjee, S., Bao, P., Barbry, M., Blunt, N. S., Bogdanov, N. A., Booth, G. H., Chen, J., Cui, Z.-H., Eriksen, J. J., Gao, Y., Guo, S., Hermann, J., Hermes, M. R., Koh, K., Koval, P., Lehtola, S., Li, Z., Liu, J., Mardirossian, N., McClain, J. D., Motta, M., Mussard, B., Pham, H. Q., Pulkin, A., Purwanto, W., Robinson, P. J., Ronca, E., Sayfutyarova, E. R., Scheurer, M., Schurkus, H. F., Smith, J. E. T., Sun, C., Sun, S.-N., Upadhyay, S., Wagner, L. K., Wang, X., White, A., Whitfield, J. D., Williamson, M. J., Wouters, S., Yang, J., Yu, J. M., Zhu, T., Berkelbach, T. C., Sharma, S., Sokolov, A. Y., & Chan, G. K.-L. (2020). Recent developments in the PySCF program package. *The Journal of Chemical Physics*, 153(2), 024109.

Szabo, A., & Ostlund, N. S. (1996). *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory*. Dover Publications.

The pandas development team. (2020). *pandas-dev/pandas: Pandas*. Zenodo.

van Rossum, G., & Drake, F. L. (2009). *Python 3 Reference Manual*. CreateSpace.

VanderPlas, J. (2016). *Python Data Science Handbook: Essential Tools for Working with Data*. O'Reilly Media.

Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., da Silva Santos, L. B., Bourne, P. E., Bouwman, J., Brookes, A. J., Clark, T., Crosas, M., Dillo, I., Dumon, O., Edmunds, S., Evelo, C. T., Finkers, R., Gonzalez-Beltran, A., Gray, A. J. G., Groth, P., Goble, C., Grethe, J. S., Heringa, J., 't Hoen, P. A. C., Hooft, R., Kuhn, T., Kok, R., Kok, J., Lusher, S. J., Martone, M. E., Mons, A., Packer, A. L., Persson, B., Rocca-Serra, P., Roos, M., van Schaik, R., Sansone, S.-A., Schultes, E., Sengstag, T., Slater, T., Strawn, G., Swertz, M. A., Thompson, M., van der Lei, J., van Mulligen, E., Velterop, J., Waagmeester, A., Wittenburg, P., Wolstencroft, K., Zhao, J., & Mons, B. (2016). The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*, 3, 160018.

Wilson, G., Aruliah, D. A., Brown, C. T., Chue Hong, N. P., Davis, M., Guy, R. T., Haddock, S. H. D., Huff, K. D., Mitchell, I. M., Plumbley, M. D., Waugh, B., White, E. P., & Wilson, P. (2014). Best practices for scientific computing. *PLOS Biology*, 12(1), e1001745.

Young, D. C. (2001). *Computational Chemistry: A Practical Guide for Applying Techniques to Real-World Problems*. John Wiley & Sons.

BAB 12

PROYEK MINI, EVALUASI AKHIR, GLOSARIUM, DAN LAMPIRAN KODE PYTHON

Pengantar Bab 12

Bab 12 merupakan bagian penutup dari rangkaian Praktikum Kimia Kuantum Bagian 2. Setelah mahasiswa mempelajari konsep, menjalankan perhitungan, membaca output, memvalidasi data, menggunakan Python untuk parsing output, membuat grafik, dan menyusun laporan, bab ini mengarahkan mahasiswa untuk mengintegrasikan seluruh keterampilan tersebut ke dalam **proyek mini, evaluasi akhir, glosarium, dan lampiran kode Python**.

Bab ini memiliki posisi penting karena menjadi jembatan antara praktikum terbimbing dan kerja ilmiah mandiri. Pada bab-bab sebelumnya, mahasiswa mengikuti alur praktikum yang telah ditentukan: sistem molekul sudah disediakan, tujuan sudah jelas, prosedur telah diarahkan, dan jenis output sudah ditetapkan. Dalam Bab 12, mahasiswa mulai diberi ruang untuk memilih masalah, merancang pendekatan, mengumpulkan data, mengolah hasil, menyusun grafik, dan menulis laporan proyek secara lebih mandiri.

Proyek mini dalam bab ini bukan sekadar tugas tambahan. Proyek mini adalah bentuk latihan berpikir ilmiah dalam kimia komputasi. Mahasiswa tidak hanya menjalankan software, tetapi harus memilih molekul, menentukan alasan pemilihan, merumuskan masalah, memilih metode, menjalankan perhitungan, memvalidasi output, mengolah data dengan Python, membandingkan hasil dengan teori atau literatur, dan menyusun kesimpulan. Dengan demikian, proyek mini menjadi puncak integrasi seluruh kompetensi praktikum.

Bab 12 juga memuat evaluasi akhir praktikum. Evaluasi tidak hanya menguji hafalan teori, tetapi juga kemampuan membaca output, menjalankan kode Python, menafsirkan grafik, mempresentasikan proyek, dan menulis laporan akhir. Evaluasi semacam ini lebih sesuai dengan karakter kimia komputasi karena keterampilan yang dibutuhkan bersifat konseptual, teknis, analitis, dan komunikatif sekaligus.

Selain itu, bab ini menyajikan glosarium kimia kuantum dan komputasi agar mahasiswa memiliki pegangan istilah penting. Glosarium membantu menyatukan pemahaman atas istilah seperti *ab initio*, basis set, DFT, energi total, frekuensi imajiner, fungsi gelombang, Hartree-Fock, HOMO, LUMO, momen dipol, optimasi geometri, orbital molekul, SCF, simetri molekul, dan zero-point energy.

Bagian akhir bab ini memuat lampiran kode Python. Lampiran kode berfungsi sebagai arsip praktikum yang dapat digunakan ulang, dimodifikasi, dan dikembangkan. Kode-kode tersebut mencakup praktikum partikel dalam kotak, optimasi geometri, Hartree-Fock dan basis set, DFT dan HOMO-LUMO, simetri dan momen dipol, spektrum IR, energi reaksi dan termodinamika, parser output komputasi, serta template laporan otomatis.

Secara umum, alur Bab 12 dapat diringkas sebagai berikut:

```
kompetensi praktikum
→ pemilihan proyek mini
→ penyusunan proposal
→ pelaksanaan perhitungan
→ validasi data
→ pengolahan Python
→ laporan proyek
→ evaluasi akhir
→ glosarium
→ lampiran kode
```

Dengan alur tersebut, Bab 12 diharapkan membantu mahasiswa menyelesaikan rangkaian Praktikum Kimia Kuantum secara utuh: dari pemahaman konsep hingga produksi laporan ilmiah berbasis data komputasi.

12.1 Proyek Mini Praktikum Kimia Kuantum

Proyek mini Praktikum Kimia Kuantum adalah kegiatan mandiri terbimbing yang menugaskan mahasiswa untuk memilih satu topik kecil dalam kimia kuantum atau kimia komputasi, menjalankan perhitungan, memvalidasi output, mengolah data, menyajikan grafik, dan menyusun laporan ilmiah. Proyek mini dirancang sebagai bentuk sintesis dari seluruh materi praktikum yang telah dipelajari sebelumnya.

Dalam proyek mini, mahasiswa memilih satu dari delapan pilihan berikut:

1. **Analisis HOMO-LUMO senyawa aromatik tersubstitusi.**
2. **Perbandingan IR komputasi dan IR literatur.**
3. **Optimasi geometri molekul obat sederhana.**
4. **Studi efek pelarut terhadap kestabilan ion.**
5. **Analisis konformer etanol atau butana.**
6. **Simetri dan polaritas molekul anorganik.**
7. **Energi ikatan hidrogen dimer air.**
8. **Energi reaksi sederhana fase gas dan pelarut.**

Setiap pilihan proyek memiliki karakter keilmuan yang berbeda. Ada proyek yang menekankan orbital molekul, ada yang menekankan spektrum vibrasi, ada yang menekankan struktur molekul, ada yang menekankan solvasi, ada yang menekankan konformasi, ada yang menekankan simetri, dan ada pula yang menekankan termodinamika reaksi. Meskipun berbeda, seluruh proyek memiliki pola kerja yang sama:

```
pilih masalah
→ pilih molekul
→ tentukan metode
→ jalankan komputasi
→ baca output
```

- validasi hasil
- olah data
- buat grafik
- analisis
- susun laporan

Proyek mini bukan sekadar latihan teknis, tetapi latihan metodologis. Mahasiswa harus belajar bahwa perhitungan kimia komputasi yang baik tidak dimulai dari menekan tombol software, melainkan dari merumuskan pertanyaan kimia yang jelas.

1. Pengertian Proyek Mini dalam Praktikum Kimia Kuantum

Proyek mini adalah kegiatan penelitian kecil yang ruang lingkupnya terbatas, tetapi tetap mengikuti prinsip kerja ilmiah. Dalam konteks Praktikum Kimia Kuantum, proyek mini tidak dituntut menghasilkan temuan baru setingkat penelitian publikasi, tetapi harus menunjukkan kemampuan mahasiswa menerapkan konsep dan prosedur komputasi secara benar.

Proyek mini memiliki ciri-ciri berikut:

1. Topiknya spesifik.
2. Sistem molekulnya terbatas.
3. Metodenya jelas.
4. Outputnya dapat divalidasi.
5. Datanya dapat diolah.
6. Grafiknya dapat disajikan.
7. Analisisnya berbasis teori.
8. Kesimpulannya menjawab rumusan masalah.

Proyek mini berbeda dari praktikum biasa. Dalam praktikum biasa, mahasiswa mengikuti instruksi yang telah disediakan. Dalam proyek mini, mahasiswa mulai membuat keputusan sendiri: memilih molekul, memilih aspek yang dianalisis, menentukan data yang diperlukan, dan menyusun argumentasi hasil.

Dengan demikian, proyek mini melatih kemandirian ilmiah.

2. Tujuan Proyek Mini

Tujuan utama proyek mini adalah mengintegrasikan seluruh keterampilan yang telah dipelajari dalam Praktikum Kimia Kuantum.

Secara khusus, proyek mini bertujuan agar mahasiswa mampu:

1. Menentukan topik komputasi kimia yang layak dikaji secara terbatas.

2. Memilih molekul target yang sesuai dengan tujuan proyek.
3. Merumuskan pertanyaan penelitian sederhana.
4. Menentukan metode komputasi dan basis set yang tepat secara pedagogis.
5. Menjalankan perhitungan menggunakan software yang tersedia.
6. Membaca output komputasi secara kritis.
7. Memvalidasi hasil perhitungan.
8. Mengolah data menggunakan Python.
9. Membuat tabel dan grafik yang komunikatif.
10. Membandingkan hasil dengan teori atau literatur.
11. Menyusun laporan proyek mini secara ilmiah.
12. Mempresentasikan hasil secara ringkas dan bertanggung jawab.

Tujuan tersebut menunjukkan bahwa proyek mini bukan hanya menguji kemampuan teknis, tetapi juga kemampuan konseptual, metodologis, analitis, dan komunikatif.

3. Latar Historis Proyek Berbasis Komputasi dalam Pendidikan Kimia

Pembelajaran kimia komputasi berkembang seiring meningkatnya akses terhadap perangkat lunak molekuler, komputer pribadi, platform web, dan bahasa pemrograman ilmiah seperti Python. Pada masa awal, kimia komputasi lebih banyak digunakan oleh peneliti dengan akses ke komputer besar dan software khusus. Namun, perkembangan perangkat lunak seperti GAMESS, ORCA, Psi4, PySCF, Avogadro, ChemCompute, dan berbagai pustaka Python membuat kimia komputasi semakin dapat digunakan dalam pendidikan.

Dalam pendidikan kimia modern, proyek berbasis komputasi mulai banyak digunakan karena dapat menghubungkan teori abstrak dengan representasi visual dan numerik. Konsep seperti orbital molekul, energi elektronik, frekuensi vibrasi, momen dipol, dan termodinamika reaksi sering sulit dipahami hanya melalui ceramah. Dengan proyek komputasi, mahasiswa dapat melihat bagaimana konsep tersebut muncul dalam data nyata.

Buku-buku kimia komputasi seperti karya Jensen, Cramer, Leach, Lewars, Levine, Szabo-Ostlund, dan Foresman-Frisch menunjukkan bahwa kimia komputasi selalu berada di antara teori, model, algoritma, dan interpretasi kimia. Proyek mini dalam bab ini mengikuti semangat tersebut: mahasiswa tidak hanya mempelajari teori, tetapi mempraktikkan bagaimana teori diterjemahkan menjadi perhitungan, output, grafik, dan kesimpulan.

Dalam konteks pembelajaran, proyek mini juga selaras dengan pendekatan pembelajaran berbasis proyek. Mahasiswa diberi masalah, memilih strategi, mengelola data, dan menghasilkan produk akhir berupa laporan serta presentasi. Pendekatan ini cocok untuk kimia komputasi karena hasil belajar yang diharapkan bukan hanya jawaban tunggal, tetapi proses berpikir ilmiah.

4. Prinsip Pemilihan Proyek Mini

Pemilihan proyek harus memperhatikan kelayakan. Proyek yang terlalu besar akan sulit diselesaikan dalam waktu praktikum. Proyek yang terlalu sederhana tidak cukup menantang. Oleh karena itu, proyek mini harus memenuhi kriteria berikut:

1. **Molekul tidak terlalu besar.**
Molekul besar membutuhkan waktu komputasi lebih lama dan lebih sulit divalidasi.
2. **Pertanyaan kimia jelas.**
Proyek harus menjawab pertanyaan tertentu, bukan hanya “menghitung molekul”.
3. **Data dapat diperoleh dari output.**
Misalnya energi, frekuensi, momen dipol, HOMO-LUMO, atau geometri.
4. **Analisis dapat dilakukan dengan Python.**
Minimal data dapat disusun dalam CSV dan dibuat grafik.
5. **Hasil dapat dibandingkan dengan teori atau literatur.**
Perbandingan tidak harus angka identik, tetapi minimal tren dapat dijelaskan.
6. **Metode sesuai sumber daya.**
Pilihan metode dan basis set harus realistis untuk komputer atau platform yang digunakan.
7. **Validasi dapat dilakukan.**
Output harus dapat diperiksa status job, konvergensi, charge, multiplicity, dan frekuensi jika diperlukan.

Prinsip ini penting agar proyek mini tetap berada dalam ruang lingkup pendidikan, bukan menjadi pekerjaan komputasi yang terlalu berat.

5. Pilihan Proyek 1: Analisis HOMO-LUMO Senyawa Aromatik Tersubstitusi

Proyek pertama adalah analisis HOMO-LUMO senyawa aromatik tersubstitusi. Proyek ini cocok untuk mahasiswa yang tertarik pada hubungan antara struktur aromatik, substituen, dan sifat elektronik.

Contoh molekul yang dapat dipilih:

1. Benzena.
2. Toluena.
3. Fenol.
4. Anilina.
5. Nitrobenzena.
6. Klorobenzena.
7. Benzaldehida.
8. Asam benzoat.

Pertanyaan proyek dapat berupa:

Bagaimana substituen donor atau penarik elektron memengaruhi energi HOMO, LUMO, dan gap HOMO-LUMO pada senyawa aromatik?

Data yang dikumpulkan:

1. Energi HOMO.
2. Energi LUMO.
3. Gap HOMO-LUMO.
4. Struktur teroptimasi.
5. Momen dipol jika relevan.
6. Visualisasi orbital jika tersedia.

Analisis yang diharapkan:

Mahasiswa menjelaskan bagaimana substituen dapat memengaruhi distribusi elektron pada cincin aromatik. Substituen donor elektron seperti gugus amino atau hidroksi dapat menaikkan energi HOMO karena meningkatkan ketersediaan elektron. Substituen penarik elektron seperti nitro atau karbonil dapat menurunkan energi LUMO karena menarik kerapatan elektron dan membuat orbital kosong lebih mudah menerima elektron. Namun, interpretasi harus hati-hati karena hasil bergantung pada metode, basis set, geometri, dan posisi substituen.

Contoh kesimpulan yang mungkin:

Senyawa aromatik dengan substituen penarik elektron menunjukkan perubahan energi LUMO yang lebih nyata dibanding benzena, sedangkan substituen donor elektron cenderung memengaruhi HOMO. Perubahan gap HOMO-LUMO menunjukkan bahwa substituen dapat mengubah sifat elektronik molekul aromatik.

Keterbatasan:

Analisis HOMO-LUMO bersifat kualitatif. Reaktivitas tidak boleh disimpulkan hanya dari nilai gap. Faktor lain seperti muatan, pelarut, posisi substituen, efek sterik, dan mekanisme reaksi tetap perlu dipertimbangkan.

6. Pilihan Proyek 2: Perbandingan IR Komputasi dan IR Literatur

Proyek kedua adalah membandingkan spektrum IR hasil komputasi dengan data IR literatur. Proyek ini cocok untuk mahasiswa yang tertarik pada spektroskopi molekul dan identifikasi gugus fungsi.

Contoh molekul:

1. Formaldehida.
2. Aseton.
3. Etanol.

4. Asam asetat.
5. Amonia.
6. Air.
7. Karbon dioksida.
8. Metanol.

Pertanyaan proyek:

Sejauh mana frekuensi vibrasi hasil komputasi sesuai dengan wilayah serapan IR yang dilaporkan dalam literatur?

Data yang dikumpulkan:

1. Frekuensi vibrasi komputasi.
2. Intensitas IR.
3. Spektrum IR hasil Python.
4. Puncak utama.
5. Data literatur untuk wilayah gugus fungsi.
6. Frekuensi imajiner untuk validasi struktur.

Analisis yang diharapkan:

Mahasiswa tidak hanya membandingkan angka, tetapi menjelaskan alasan perbedaan. Frekuensi komputasi umumnya dihitung dengan pendekatan osilator harmonik, sedangkan spektrum eksperimen dipengaruhi oleh anharmonitas, lingkungan, fase, pelarut, suhu, dan instrumen. Karena itu, perbandingan sering dilakukan berdasarkan wilayah serapan dan pola puncak, bukan hanya pencocokan angka mutlak.

Contoh analisis:

Puncak regangan C=O pada formaldehida muncul pada wilayah bilangan gelombang tinggi yang sesuai dengan karakter ikatan karbonil. Perbedaan nilai komputasi terhadap literatur dapat disebabkan oleh pendekatan harmonik dan level teori yang digunakan.

Keterbatasan:

Jika mahasiswa tidak menggunakan faktor skala frekuensi, nilai komputasi dapat lebih tinggi daripada eksperimen. Oleh karena itu, kesimpulan harus menyebutkan bahwa hasil bersifat pendekatan.

7. Pilihan Proyek 3: Optimasi Geometri Molekul Obat Sederhana

Proyek ketiga adalah optimasi geometri molekul obat sederhana. Proyek ini cocok untuk mahasiswa yang tertarik pada kimia farmasi, struktur molekul biologis, atau hubungan struktur dengan sifat molekul.

Contoh molekul obat sederhana:

1. Aspirin.
2. Parasetamol.
3. Kafein.
4. Ibuprofen.
5. Nikotinamida.
6. Asam salisilat.
7. Urea sebagai model farmasetik sederhana.
8. Benzokain jika sumber daya memungkinkan.

Pertanyaan proyek:

Bagaimana struktur teroptimasi molekul obat sederhana menggambarkan gugus fungsi penting dan potensi interaksi molekulnya?

Data yang dikumpulkan:

1. Struktur awal.
2. Struktur teroptimasi.
3. Energi total.
4. Panjang ikatan penting.
5. Sudut ikatan penting.
6. Momen dipol.
7. HOMO-LUMO jika relevan.
8. Frekuensi imajiner untuk validasi minimum.

Analisis yang diharapkan:

Mahasiswa menjelaskan hubungan antara struktur molekul dan gugus fungsi. Misalnya, aspirin memiliki gugus karboksilat dan ester; parasetamol memiliki gugus fenol dan amida; kafein memiliki heteroatom nitrogen dan oksigen yang memengaruhi polaritas. Optimasi geometri dapat menunjukkan orientasi gugus fungsi, kemungkinan donor-akseptor ikatan hidrogen, serta distribusi polaritas.

Keterbatasan:

Molekul obat memiliki konformasi yang dapat berubah, terutama jika memiliki ikatan tunggal fleksibel. Satu hasil optimasi belum tentu mewakili seluruh populasi konformer. Oleh karena itu, proyek ini sebaiknya membatasi diri pada satu atau beberapa konformer sederhana.

8. Pilihan Proyek 4: Studi Efek Pelarut terhadap Kestabilan Ion

Proyek keempat adalah studi efek pelarut terhadap kestabilan ion. Proyek ini cocok untuk mahasiswa yang tertarik pada solvasi, ion, polaritas, dan termodinamika larutan.

Contoh spesies:

1. NH_4^+ .
2. H_3O^+ .
3. Cl^- .
4. CH_3COO^- .
5. Na^+ sebagai model sederhana jika software mendukung.
6. Amonia dan amonium sebagai pasangan netral-ion.
7. Asam asetat dan asetat.
8. Metanol terprotonasi sebagai model sederhana.

Pertanyaan proyek:

Bagaimana model pelarut kontinu memengaruhi energi relatif spesies bermuatan dibanding fase gas?

Data yang dikumpulkan:

1. Energi fase gas.
2. Energi dalam pelarut.
3. Perubahan energi solvasi relatif.
4. Momen dipol.
5. Charge dan multiplicity.
6. Model pelarut yang digunakan.
7. Validasi status job dan geometri.

Analisis yang diharapkan:

Mahasiswa menjelaskan bahwa pelarut polar dapat menstabilkan spesies bermuatan melalui interaksi elektrostatik. Model pelarut kontinu memperlakukan pelarut sebagai medium dielektrik. Spesies bermuatan biasanya mengalami stabilisasi lebih besar dalam pelarut polar dibanding spesies netral. Namun, mahasiswa harus menyadari bahwa model kontinu tidak menggambarkan molekul pelarut eksplisit, ikatan hidrogen spesifik, atau dinamika larutan.

Keterbatasan:

Hasil model pelarut kontinu tidak boleh dianggap sama persis dengan realitas larutan. Interpretasi harus menyebutkan bahwa pelarut digambarkan sebagai medium homogen, bukan kumpulan molekul pelarut individual.

9. Pilihan Proyek 5: Analisis Konformer Etanol atau Butana

Proyek kelima adalah analisis konformer etanol atau butana. Proyek ini cocok untuk mahasiswa yang tertarik pada rotasi ikatan, energi relatif, dan kestabilan konformasi.

Contoh sistem:

1. Etanol anti.
2. Etanol gauche.
3. Butana anti.
4. Butana gauche.
5. Butana eclipsed jika dianalisis sebagai struktur tidak stabil atau melalui scan torsional sederhana.

Pertanyaan proyek:

Konformer mana yang lebih stabil dan bagaimana perbedaan energi tersebut dijelaskan berdasarkan interaksi sterik atau intramolekul?

Data yang dikumpulkan:

1. Struktur awal konformer.
2. Struktur teroptimasi.
3. Energi total.
4. Energi relatif.
5. Sudut dihedral.
6. Momen dipol.
7. Frekuensi imajiner.
8. Grafik energi relatif.

Analisis yang diharapkan:

Pada butana, konformer anti biasanya lebih stabil daripada gauche karena tolakan sterik antara gugus metil lebih kecil. Pada etanol, perbedaan konformer dapat dipengaruhi oleh orientasi gugus O–H, interaksi polar, dan lingkungan pelarut. Selisih energi konformer sering kecil sehingga hasil sensitif terhadap metode dan basis set.

Keterbatasan:

Optimasi dapat menyebabkan dua struktur awal berbeda berakhir pada konformer yang sama. Karena itu, mahasiswa harus memeriksa sudut dihedral akhir. Jika sudut dihedral tidak sesuai, perbandingan energi tidak valid.

10. Pilihan Proyek 6: Simetri dan Polaritas Molekul Anorganik

Proyek keenam adalah simetri dan polaritas molekul anorganik. Proyek ini cocok untuk mahasiswa yang tertarik pada bentuk molekul, simetri, dan momen dipol.

Contoh molekul:

1. H_2O .
2. CO_2 .
3. NH_3 .
4. BF_3 .
5. CH_4 .
6. SO_2 .
7. H_2S .
8. PCl_3 .
9. BeCl_2 .
10. SF_6 jika sumber daya dan software memungkinkan.

Pertanyaan proyek:

Bagaimana simetri molekul memengaruhi momen dipol total?

Data yang dikumpulkan:

1. Geometri teroptimasi.
2. Bentuk molekul.
3. Komponen momen dipol.
4. Momen dipol total.
5. Frekuensi imajiner.
6. Tabel perbandingan molekul polar dan nonpolar.
7. Visualisasi struktur.

Analisis yang diharapkan:

Mahasiswa menjelaskan bahwa momen dipol tidak hanya bergantung pada polaritas ikatan, tetapi juga pada geometri. CO_2 memiliki ikatan $\text{C}=\text{O}$ polar, tetapi momen dipol totalnya mendekati nol karena bentuk linear simetris. H_2O memiliki ikatan $\text{O}-\text{H}$ polar dan bentuk bengkok sehingga momen dipol total tidak nol. BF_3 dapat memiliki ikatan polar tetapi momen total mendekati nol karena trigonal planar simetris.

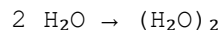
Keterbatasan:

Mahasiswa harus berhati-hati membedakan polaritas ikatan dan polaritas molekul. Molekul dengan ikatan polar tidak selalu polar secara keseluruhan jika simetrinya membuat vektor dipol saling meniadakan.

11. Pilihan Proyek 7: Energi Ikatan Hidrogen Dimer Air

Proyek ketujuh adalah energi ikatan hidrogen dimer air. Proyek ini cocok untuk mahasiswa yang tertarik pada interaksi antarmolekul, energi dimerisasi, dan ikatan hidrogen.

Sistem:



Pertanyaan proyek:

Berapa energi relatif pembentukan dimer air dan bagaimana hasil tersebut menjelaskan kestabilan ikatan hidrogen?

Data yang dikumpulkan:

1. Energi monomer air.
2. Energi dimer air.
3. Energi dimerisasi.
4. Geometri ikatan hidrogen.
5. Jarak $\text{O} \cdots \text{H}$.
6. Frekuensi imajiner.
7. Momen dipol dimer.
8. Diagram energi dimerisasi.

Perhitungan energi dimerisasi:

$$\Delta E = E_{\text{dimer}} - 2E_{\text{monomer}}$$

Jika ΔE negatif, dimer lebih stabil secara energi elektronik daripada dua monomer terpisah pada level teori yang digunakan.

Analisis yang diharapkan:

Mahasiswa menjelaskan bahwa dimer air distabilkan oleh ikatan hidrogen. Namun, energi dimerisasi sangat sensitif terhadap metode, basis set, koreksi dispersi, dan basis set superposition error. Untuk tingkat praktikum, pembahasan cukup menekankan bahwa hasil bersifat model dan digunakan untuk memahami konsep interaksi antarmolekul.

Keterbatasan:

Jika dimer tidak dioptimasi dengan benar, dua molekul air dapat terlalu jauh atau orientasi ikatan hidrogennya tidak tepat. Frekuensi imajiner juga harus diperiksa untuk memastikan struktur minimum.

12. Pilihan Proyek 8: Energi Reaksi Sederhana Fase Gas dan Pelarut

Proyek kedelapan adalah analisis energi reaksi sederhana dalam fase gas dan pelarut. Proyek ini cocok untuk mahasiswa yang ingin mengintegrasikan energi reaksi, termodinamika, dan efek pelarut.

Contoh reaksi sederhana:

1. Protonasi amonia.
2. Hidrasi formaldehida.
3. Isomerisasi cis-trans 2-butena.
4. Dimerisasi asam asetat.
5. Pembentukan dimer air.
6. Reaksi asam-basa sederhana yang secara komputasi masih realistis.

Pertanyaan proyek:

Bagaimana perbedaan fase gas dan pelarut memengaruhi ΔE , ΔH , atau ΔG suatu reaksi sederhana?

Data yang dikumpulkan:

1. Energi reaktan fase gas.
2. Energi produk fase gas.
3. Energi reaktan dalam pelarut.
4. Energi produk dalam pelarut.
5. ΔE , ΔH , atau ΔG .
6. Model pelarut.
7. Diagram energi reaksi.
8. Grafik perbandingan fase gas dan pelarut.

Analisis yang diharapkan:

Mahasiswa menjelaskan bahwa pelarut dapat menstabilkan spesies polar atau bermuatan. Akibatnya, ΔG reaksi dapat berubah antara fase gas dan pelarut. Namun, model pelarut harus diterapkan secara konsisten pada semua spesies. Jangan membandingkan reaktan fase gas dengan produk pelarut sebagai satu reaksi yang sama.

Keterbatasan:

Reaksi ionik seperti protonasi dapat sulit dimodelkan secara absolut karena spesies seperti H^+ sangat ideal. Untuk proyek mini, mahasiswa sebaiknya menekankan tren dan keterbatasan model.

13. Kriteria Kelayakan Proyek Mini

Agar proyek dapat disetujui, mahasiswa harus memastikan beberapa kriteria kelayakan.

Kriteria	Penjelasan
Topik jelas	Proyek menjawab pertanyaan tertentu
Molekul realistis	Ukuran molekul sesuai sumber daya
Metode tersedia	Software mampu menjalankan perhitungan
Output dapat divalidasi	Status job, konvergensi, dan frekuensi dapat diperiksa
Data dapat diolah	Data dapat dibuat tabel dan grafik
Literatur tersedia	Tren dapat dibandingkan dengan sumber tepercaya
Waktu cukup	Proyek dapat selesai sesuai jadwal
Risiko terkendali	Tidak terlalu banyak spesies atau metode

Jika proyek tidak memenuhi kriteria ini, proyek harus disederhanakan.

Contoh proyek terlalu luas:

Analisis semua obat analgesik dengan berbagai metode DFT dan semua model pelarut.

Proyek tersebut terlalu besar untuk praktikum.

Contoh proyek yang lebih layak:

Optimasi geometri dan analisis momen dipol aspirin menggunakan satu metode dan satu basis set.

14. Alur Kerja Proyek Mini

Alur kerja proyek mini dapat disusun sebagai berikut:

1. Memilih topik proyek.
2. Menentukan molekul target.
3. Merumuskan masalah.
4. Menentukan metode dan basis set.
5. Menyiapkan struktur awal.
6. Menjalankan optimasi geometri.
7. Menjalankan perhitungan tambahan jika diperlukan.
8. Membaca output.
9. Memvalidasi hasil.
10. Mengolah data dengan Python.
11. Membuat tabel dan grafik.
12. Membandingkan hasil dengan teori atau literatur.
13. Menulis laporan.
14. Mempresentasikan hasil.

Alur ini harus diikuti agar proyek tidak hanya menghasilkan angka, tetapi menghasilkan analisis ilmiah yang lengkap.

15. Peran Python dalam Proyek Mini

Python berperan sebagai alat bantu pengolahan data dan visualisasi. Dalam proyek mini, Python dapat digunakan untuk:

1. Membaca data CSV.
2. Menghitung energi relatif.
3. Mengonversi Hartree ke kJ/mol.
4. Menghitung gap HOMO-LUMO.
5. Membuat grafik energi.
6. Membuat diagram orbital.
7. Membuat spektrum IR.
8. Membuat diagram energi reaksi.
9. Membuat grafik perbandingan metode.
10. Menyusun ringkasan output.

Namun, Python tidak menggantikan pemahaman kimia. Python hanya membantu memproses data. Mahasiswa tetap harus menjelaskan makna data berdasarkan teori.

Prinsip penting:

Python mempercepat pengolahan
tetapi analisis tetap ditentukan oleh pemahaman kimia

16. Contoh Rancangan Proyek Mini: HOMO-LUMO Nitrobenzena dan Anilina

Judul proyek:

Perbandingan HOMO-LUMO Nitrobenzena dan Anilina sebagai Senyawa Aromatik Tersubstitusi

Rumusan masalah:

Bagaimana substituen nitro dan amino memengaruhi energi HOMO, LUMO, dan gap HOMO-LUMO pada cincin aromatik?

Molekul:

1. Benzena.
2. Nitrobenzena.
3. Anilina.

Data:

1. Energi HOMO.
2. Energi LUMO.
3. Gap HOMO-LUMO.
4. Momen dipol.
5. Struktur teroptimasi.

Analisis:

Nitrobenzena memiliki substituen penarik elektron, sedangkan anilina memiliki substituen donor elektron. Perbandingan energi orbital dapat digunakan untuk menjelaskan pengaruh substituen terhadap sifat elektronik cincin aromatik. Benzena digunakan sebagai pembandingan dasar.

Grafik:

1. Diagram HOMO-LUMO.
2. Grafik gap HOMO-LUMO.
3. Grafik momen dipol.

Keterbatasan:

Analisis ini bersifat kualitatif dan tidak langsung menunjukkan laju reaksi atau arah substitusi tanpa kajian mekanisme lebih lanjut.

17. Contoh Rancangan Proyek Mini: Spektrum IR Formaldehida

Judul proyek:

Perbandingan Spektrum IR Komputasi Formaldehida dengan Wilayah Serapan Literatur

Rumusan masalah:

Apakah frekuensi vibrasi formaldehida hasil komputasi sesuai dengan wilayah serapan khas gugus karbonil dan C-H?

Data:

1. Frekuensi vibrasi.
2. Intensitas IR.
3. Spektrum IR komputasi.
4. Frekuensi imajiner.
5. Data wilayah serapan literatur.

Analisis:

Mahasiswa mengidentifikasi puncak regangan C=O dan C–H. Puncak karbonil diharapkan muncul pada wilayah tinggi khas C=O. Jika frekuensi komputasi berbeda dari literatur, mahasiswa menjelaskan pengaruh pendekatan harmonik, metode, basis set, dan kemungkinan faktor skala.

Grafik:

1. Spektrum IR garis.
2. Spektrum IR dengan pelebaran Gaussian.

Keterbatasan:

Perbandingan dilakukan terhadap wilayah literatur, bukan pencocokan angka mutlak.

18. Contoh Rancangan Proyek Mini: Efek Pelarut pada NH_4^+

Judul proyek:

Studi Efek Pelarut terhadap Kestabilan Ion Amonium

Rumusan masalah:

Bagaimana pelarut polar memengaruhi energi relatif ion NH_4^+ dibanding fase gas?

Data:

1. Energi NH_4^+ fase gas.
2. Energi NH_4^+ dalam pelarut.
3. Momen dipol atau data solvasi jika tersedia.
4. Charge dan multiplicity.
5. Status validasi output.

Analisis:

NH_4^+ sebagai ion bermuatan positif diharapkan mengalami stabilisasi dalam pelarut polar. Model pelarut kontinu menjelaskan stabilisasi melalui respons dielektrik medium terhadap muatan. Namun, model ini tidak menampilkan molekul pelarut eksplisit atau ikatan hidrogen spesifik.

Grafik:

1. Grafik energi fase gas dan pelarut.
2. Tabel validasi output.

Keterbatasan:

Energi absolut ion dalam model pelarut harus ditafsirkan hati-hati. Proyek lebih baik menekankan tren stabilisasi, bukan klaim kuantitatif berlebihan.

19. Contoh Rancangan Proyek Mini: Dimer Air

Judul proyek:

Analisis Energi Ikatan Hidrogen pada Dimer Air

Rumusan masalah:

Bagaimana energi dimerisasi air menjelaskan kestabilan ikatan hidrogen?

Data:

1. Energi monomer air.
2. Energi dimer air.
3. Energi dimerisasi.
4. Jarak $O \cdots H$.
5. Frekuensi imajiner.
6. Struktur dimer.
7. Diagram energi.

Rumus:

$$\Delta E_{\text{dimerisasi}} = E_{(H_2O)_2} - 2E_{H_2O}$$

Analisis:

Jika ΔE bernilai negatif, dimer lebih stabil dibanding dua monomer terpisah pada level teori yang digunakan. Stabilitas ini dikaitkan dengan ikatan hidrogen. Namun, mahasiswa harus menyebut keterbatasan basis set dan kemungkinan basis set superposition error pada kajian interaksi lemah.

20. Standar Minimal Output Proyek Mini

Setiap proyek mini minimal harus menghasilkan:

1. File input.
2. File output.
3. Tabel data utama.

4. Tabel validasi.
5. Grafik utama.
6. Analisis hasil.
7. Kesimpulan.
8. Lampiran data atau kode.

Format minimal tabel validasi:

Kriteria	Status	Keterangan
Job selesai normal	Ya/Tidak	Berdasarkan output
SCF konvergen	Ya/Tidak	Jika tersedia
Optimasi konvergen	Ya/Tidak	Jika optimasi dilakukan
Frekuensi imajiner	Ada/Tidak	Untuk minimum harus tidak ada
Charge benar	Ya/Tidak	Sesuai spesies
Multiplicity benar	Ya/Tidak	Sesuai keadaan spin
Geometri masuk akal	Ya/Tidak	Dicek visual
Data layak dianalisis	Ya/Tidak	Keputusan akhir

Tanpa tabel validasi, laporan proyek mini belum lengkap.

21. Standar Minimal Analisis Proyek Mini

Analisis proyek mini harus menjawab pertanyaan berikut:

1. Apa hasil utama proyek?
2. Data apa yang mendukung hasil tersebut?
3. Apakah output valid?
4. Bagaimana hasil dijelaskan secara teori?
5. Apakah hasil sesuai dengan literatur atau tren kimia?
6. Apa keterbatasan metode?
7. Apa kesimpulan yang dapat ditarik secara hati-hati?

Analisis tidak boleh hanya menulis:

Grafik menunjukkan hasil perhitungan.

Analisis harus menjelaskan makna kimia:

Grafik menunjukkan bahwa konformer anti butana memiliki energi relatif lebih rendah dibanding konformer gauche. Hal ini dapat dijelaskan oleh tolakan sterik yang lebih kecil antara dua gugus metil pada susunan anti.

22. Kesalahan Umum dalam Proyek Mini

Kesalahan umum yang perlu dihindari:

Kesalahan	Dampak
Topik terlalu luas	Proyek tidak selesai
Molekul terlalu besar	Perhitungan terlalu berat
Tidak ada rumusan masalah	Analisis tidak fokus
Tidak mencatat metode	Hasil tidak reproduisibel
Charge salah	Sistem tidak valid
Multiplicity salah	Keadaan elektronik salah
Tidak memeriksa konvergensi	Data belum sahih
Tidak memeriksa frekuensi imajiner	Struktur minimum belum valid
Grafik tanpa satuan	Interpretasi tidak jelas
Tidak membandingkan teori/literatur	Analisis lemah
Kesimpulan terlalu berlebihan	Tidak sesuai data
Tidak melampirkan input-output	Hasil sulit diverifikasi

Kesalahan ini harus dicegah sejak tahap proposal.

23. Penilaian Proyek Mini

Penilaian proyek mini dapat mencakup beberapa aspek:

Aspek	Kriteria
Kelayakan topik	Topik jelas dan realistis
Rumusan masalah	Spesifik dan dapat dijawab
Metode	Sesuai tujuan dan terdokumentasi
Validasi	Output diperiksa secara benar
Data	Tabel lengkap dan bersatuan
Python	Digunakan untuk pengolahan/grafik
Analisis	Berbasis teori dan data
Kritis	Menyebut keterbatasan
Laporan	Rapi dan sistematis
Presentasi	Jelas dan argumentatif

Penilaian tidak hanya berfokus pada hasil akhir, tetapi juga pada proses kerja ilmiah.

24. Implikasi Keilmuan Proyek Mini

Proyek mini memiliki implikasi keilmuan karena melatih mahasiswa memahami kimia komputasi sebagai proses penelitian kecil. Dalam proyek mini, mahasiswa belajar bahwa setiap keputusan memengaruhi hasil: pemilihan molekul, metode, basis set, charge, multiplicity, model pelarut, validasi output, dan pengolahan data.

Mahasiswa juga belajar membedakan antara hasil numerik dan kesimpulan ilmiah. Angka energi, frekuensi, atau orbital belum menjadi pengetahuan sampai dijelaskan secara teoritis dan divalidasi. Proyek mini melatih proses transformasi:

```
output
→ data
→ grafik
→ analisis
→ kesimpulan
```

Inilah inti kerja ilmiah dalam kimia komputasi.

25. Implikasi Pedagogis Proyek Mini

Secara pedagogis, proyek mini membantu mahasiswa mengembangkan kemandirian. Mahasiswa tidak hanya mengikuti prosedur, tetapi mulai membuat keputusan ilmiah. Keterampilan ini penting bagi calon guru, calon peneliti, dan praktisi pendidikan kimia.

Keterampilan yang dikembangkan meliputi:

1. Memilih masalah.
2. Merancang proyek.
3. Menyusun pertanyaan ilmiah.
4. Menjalankan komputasi.
5. Membaca output.
6. Memvalidasi hasil.
7. Menggunakan Python.
8. Membuat grafik.
9. Menulis laporan.
10. Mempresentasikan hasil.

Bagi calon guru kimia, proyek mini dapat menjadi model pembelajaran yang dapat diadaptasi untuk siswa atau mahasiswa. Guru dapat menggunakan proyek sederhana seperti momen dipol H_2O dan CO_2 , spektrum IR formaldehida, atau HOMO-LUMO benzena tersubstitusi sebagai bahan pembelajaran berbasis komputasi.

26. Sintesis Subbab 12.1

Proyek mini Praktikum Kimia Kuantum merupakan puncak integrasi dari seluruh keterampilan yang telah dipelajari. Mahasiswa memilih satu proyek dari delapan pilihan: analisis HOMO-LUMO senyawa aromatik tersubstitusi, perbandingan IR komputasi dan IR literatur, optimasi geometri molekul obat sederhana, studi efek pelarut terhadap kestabilan ion, analisis konformer etanol atau butana, simetri dan polaritas molekul anorganik, energi ikatan hidrogen dimer air, atau energi reaksi sederhana fase gas dan pelarut.

Setiap proyek harus memiliki rumusan masalah, molekul target, metode komputasi, data output, validasi, pengolahan Python, grafik, analisis, dan kesimpulan. Proyek mini tidak hanya menguji kemampuan menjalankan software, tetapi juga kemampuan berpikir ilmiah, memilih metode, menilai data, dan menyusun laporan berbasis bukti.

Alur proyek mini dapat diringkas sebagai berikut:

```

topik
→ rumusan masalah
→ molekul target
→ metode
→ komputasi
→ output
→ validasi
→ Python
→ grafik
→ analisis
→ laporan
→ presentasi

```

Dengan proyek mini, mahasiswa diharapkan mampu menunjukkan bahwa mereka telah memahami kimia kuantum dan kimia komputasi bukan hanya sebagai teori, tetapi sebagai keterampilan ilmiah yang dapat diterapkan secara mandiri.

27. Transisi ke Subbab 12.2

Subbab 12.1 telah membahas proyek mini Praktikum Kimia Kuantum, mulai dari pengertian, tujuan, prinsip pemilihan topik, delapan pilihan proyek, contoh rancangan proyek, standar output, standar analisis, kesalahan umum, penilaian, hingga implikasi keilmuan dan pedagogis. Pembahasan ini menegaskan bahwa proyek mini harus dirancang secara sistematis agar hasilnya layak dianalisis dan dilaporkan.

Subbab berikutnya akan membahas format proposal proyek mini. Pada subbab 12.2, mahasiswa akan mempelajari cara menyusun proposal yang memuat judul, latar belakang, rumusan masalah, molekul target, metode komputasi, software, kode Python pendukung, data yang dikumpulkan, dan jadwal kerja. Proposal tersebut menjadi rencana awal sebelum proyek mini dijalankan.

12.2 Format Proposal Proyek Mini

Format proposal proyek mini adalah rancangan tertulis yang memuat rencana kerja mahasiswa sebelum melaksanakan proyek mini Praktikum Kimia Kuantum. Proposal ini berfungsi sebagai peta awal agar proyek tidak berjalan secara acak, terlalu luas, tidak realistis, atau tidak memiliki dasar metodologis yang jelas. Dalam praktikum kimia komputasi, proposal sangat penting karena setiap keputusan awal—pemilihan molekul, metode, basis set, software, model pelarut, jenis data, dan kode Python—akan menentukan kualitas hasil akhir.

Proposal proyek mini bukan sekadar persyaratan administratif. Proposal adalah dokumen ilmiah ringkas yang menunjukkan bahwa mahasiswa telah memahami masalah yang akan dikaji, alasan pemilihan molekul, metode yang akan digunakan, data yang akan dikumpulkan, serta cara mengolah dan melaporkan hasil. Dengan proposal, mahasiswa belajar merancang proyek secara sadar sebelum menjalankan perhitungan.

Format proposal proyek mini dalam bab ini terdiri atas sembilan komponen utama:

1. **Judul.**
2. **Latar belakang.**
3. **Rumusan masalah.**
4. **Molekul target.**
5. **Metode komputasi.**
6. **Software.**
7. **Kode Python pendukung.**
8. **Data yang dikumpulkan.**
9. **Jadwal kerja.**

Kesembilan komponen tersebut membentuk alur berpikir ilmiah:

```
judul
→ latar belakang
→ rumusan masalah
→ molekul target
→ metode
→ software
→ kode Python
→ data
→ jadwal
```

Dengan alur ini, mahasiswa tidak hanya memilih topik, tetapi juga menyusun strategi kerja yang dapat dilaksanakan, divalidasi, dan dilaporkan.

1. Pengertian Proposal Proyek Mini

Proposal proyek mini adalah dokumen perencanaan singkat yang menjelaskan apa yang akan diteliti, mengapa topik tersebut penting, molekul apa yang digunakan, metode apa yang dipilih, data apa yang akan dikumpulkan, software apa yang digunakan, kode Python apa yang mendukung, dan kapan setiap tahap dikerjakan.

Dalam konteks Praktikum Kimia Kuantum, proposal tidak perlu sepanjang proposal penelitian skripsi. Namun, proposal tetap harus menunjukkan struktur ilmiah yang jelas. Proposal yang baik harus mampu menjawab pertanyaan berikut:

Apa yang akan dikaji?
 Mengapa topik itu dipilih?
 Molekul apa yang digunakan?
 Bagaimana perhitungan dilakukan?
 Data apa yang akan diambil?
 Bagaimana data divalidasi?
 Bagaimana data diolah?
 Kapan pekerjaan diselesaikan?

Jika pertanyaan-pertanyaan tersebut belum dapat dijawab, maka proyek mini belum siap dijalankan.

Proposal proyek mini juga menjadi alat komunikasi antara mahasiswa dan dosen. Dosen dapat menilai apakah proyek terlalu luas, terlalu berat, terlalu sederhana, atau tidak sesuai dengan tujuan praktikum. Dengan demikian, proposal membantu mencegah kegagalan sejak tahap awal.

2. Latar Historis Proposal dalam Kerja Ilmiah

Dalam tradisi penelitian ilmiah, proposal muncul sebagai sarana untuk merancang kegiatan sebelum penelitian dilakukan. Proposal membantu peneliti menjelaskan masalah, tujuan, metode, sumber data, jadwal, dan keluaran yang diharapkan. Dalam kimia eksperimen, proposal membantu menentukan bahan, alat, prosedur, dan keselamatan kerja. Dalam kimia komputasi, proposal membantu menentukan molekul, metode, basis set, software, model perhitungan, validasi output, dan pengolahan data.

Kimia komputasi memiliki karakter khusus karena objek kerjanya tidak selalu berupa bahan fisik di laboratorium, tetapi berupa model molekul, input file, parameter perhitungan, output numerik, dan interpretasi berbasis teori. Karena itu, proposal kimia komputasi harus menuliskan parameter komputasi secara jelas. Tanpa parameter tersebut, hasil sulit direproduksi.

Literatur kimia komputasi yang bereputasi seperti karya Jensen, Cramer, Leach, Lewars, Levine, Szabo-Ostlund, serta panduan praktis Foresman-Frisch menekankan pentingnya pemilihan model, metode, basis set, validasi output, dan interpretasi hasil. Prinsip tersebut diterapkan dalam format

proposal proyek mini agar mahasiswa sejak awal tidak hanya berpikir tentang “molekul apa yang akan dihitung”, tetapi juga “bagaimana hasil perhitungan itu akan dinilai”.

3. Fungsi Proposal Proyek Mini

Proposal proyek mini memiliki beberapa fungsi penting.

Pertama, proposal berfungsi sebagai **alat perencanaan**. Mahasiswa menyusun langkah kerja sebelum menjalankan perhitungan.

Kedua, proposal berfungsi sebagai **alat pembatas ruang lingkup**. Mahasiswa menghindari proyek yang terlalu luas atau terlalu berat.

Ketiga, proposal berfungsi sebagai **alat validasi awal**. Dosen dapat memeriksa apakah molekul, metode, charge, multiplicity, dan data yang direncanakan sudah masuk akal.

Keempat, proposal berfungsi sebagai **alat penilaian proses**. Mahasiswa dinilai bukan hanya dari laporan akhir, tetapi juga dari kemampuan merancang pekerjaan.

Kelima, proposal berfungsi sebagai **alat reproduibilitas**. Informasi tentang metode, software, data, dan kode Python sudah direncanakan sejak awal.

Keenam, proposal berfungsi sebagai **alat pengendalian waktu**. Jadwal kerja membantu mahasiswa menyelesaikan proyek secara bertahap.

Dengan demikian, proposal bukan dokumen tambahan, melainkan bagian dari proses ilmiah.

4. Prinsip Umum Proposal yang Baik

Proposal proyek mini yang baik harus memenuhi beberapa prinsip.

Pertama, **spesifik**. Topik tidak boleh terlalu luas. Judul seperti “Analisis Kimia Komputasi Senyawa Obat” terlalu luas. Judul yang lebih baik adalah “Optimasi Geometri dan Analisis Momen Dipol Parasetamol”.

Kedua, **terukur**. Data yang dikumpulkan harus jelas. Misalnya energi total, HOMO, LUMO, gap, frekuensi IR, momen dipol, atau ΔG .

Ketiga, **realistis**. Molekul dan metode harus sesuai dengan waktu, software, dan kemampuan komputasi.

Keempat, **relevan**. Topik harus berhubungan dengan materi praktikum kimia kuantum dan kimia komputasi.

Kelima, **tervalidasi**. Proposal harus menyebut bagaimana output akan diperiksa.

Keenam, **dapat diolah dengan Python**. Minimal data dapat dimasukkan ke CSV dan dibuat grafik.

Ketujuh, **memiliki dasar teori**. Latar belakang harus menjelaskan alasan ilmiah pemilihan topik.

Prinsip ini dapat diringkas:

```
proposal baik
= topik jelas
+ metode realistis
+ data terukur
+ validasi direncanakan
+ jadwal masuk akal
```

5. Komponen 1: Judul

Judul adalah pernyataan singkat yang menggambarkan fokus proyek. Judul harus menunjukkan molekul, konsep utama, dan jenis analisis yang dilakukan. Judul tidak boleh terlalu umum, tidak boleh terlalu panjang, dan tidak boleh menjanjikan hal yang tidak dilakukan.

Judul yang kurang baik:

Studi Kimia Komputasi Molekul Organik

Judul tersebut terlalu umum. Tidak jelas molekulnya, tidak jelas metodenya, dan tidak jelas data yang dianalisis.

Judul yang lebih baik:

Analisis HOMO-LUMO Benzena, Anilina, dan Nitrobenzena Menggunakan Metode DFT

Judul ini lebih jelas karena memuat objek kajian, konsep utama, dan pendekatan umum.

Contoh judul lain:

Perbandingan Spektrum IR Komputasi Formaldehida dengan Data Literatur

Optimasi Geometri dan Analisis Momen Dipol Parasetamol

Studi Efek Pelarut terhadap Kestabilan Ion Amonium

Analisis Energi Konformer Anti dan Gauche pada Butana

Simetri dan Polaritas Molekul H_2O , CO_2 , NH_3 , dan BF_3

Analisis Energi Ikatan Hidrogen pada Dimer Air

Perbandingan Energi Reaksi Hidrasi Formaldehida pada Fase Gas dan Pelarut

Judul harus mencerminkan isi proyek. Jangan menulis “termodinamika” jika proyek hanya menghitung energi elektronik. Jangan menulis “mekanisme reaksi” jika tidak menghitung keadaan transisi atau jalur reaksi.

6. Cara Menyusun Judul yang Tepat

Judul dapat disusun dengan pola berikut:

[Jenis analisis] + [molekul/sistem] + [aspek/metode]

Contoh:

Analisis HOMO-LUMO Senyawa Aromatik Tersubstitusi Menggunakan DFT

Perbandingan Spektrum IR Komputasi dan Literatur pada Formaldehida

Studi Efek Pelarut terhadap Energi Ion Amonium

Analisis Konformer Etanol Berdasarkan Energi Relatif dan Momen Dipol

Judul juga dapat menggunakan pola:

[Konsep utama] pada [molekul target]: [pendekatan komputasi]

Contoh:

Polaritas Molekul Anorganik Sederhana: Analisis Geometri dan Momen Dipol Komputasi

Judul harus menghindari kata-kata berlebihan seperti “sangat mendalam”, “paling akurat”, atau “terbaik” karena proyek mini tidak dirancang untuk klaim seperti itu.

7. Komponen 2: Latar Belakang

Latar belakang menjelaskan alasan ilmiah mengapa proyek dipilih. Bagian ini harus menjawab pertanyaan:

Mengapa topik ini penting?

Apa konsep kimia yang terlibat?

Mengapa molekul ini layak dianalisis?
 Apa hubungan proyek dengan praktikum sebelumnya?
 Apa manfaat proyek bagi pemahaman kimia kuantum?

Latar belakang tidak boleh hanya berupa pernyataan umum. Latar belakang harus menunjukkan hubungan antara masalah kimia dan pendekatan komputasi.

Contoh latar belakang untuk HOMO-LUMO:

Senyawa aromatik tersubstitusi memiliki sifat elektronik yang dipengaruhi oleh jenis substituen. Substituen donor elektron dan penarik elektron dapat mengubah distribusi elektron pada cincin aromatik. Analisis HOMO-LUMO dapat digunakan untuk memahami perubahan energi orbital dan gap elektronik secara kualitatif. Oleh karena itu, benzena, anilina, dan nitrobenzena dipilih sebagai sistem model untuk membandingkan pengaruh substituen terhadap sifat elektronik molekul aromatik.

Contoh latar belakang untuk IR:

Spektrum inframerah digunakan untuk mengidentifikasi gugus fungsi berdasarkan frekuensi vibrasi molekul. Perhitungan kimia komputasi dapat memprediksi frekuensi dan intensitas IR melalui analisis vibrasi. Formaldehida dipilih karena memiliki gugus karbonil yang memberikan puncak khas pada spektrum IR. Perbandingan antara IR komputasi dan wilayah serapan literatur dapat membantu mahasiswa memahami hubungan antara struktur, vibrasi, dan spektrum.

Latar belakang harus singkat tetapi padat secara konsep.

8. Unsur Teori dalam Latar Belakang

Latar belakang harus memuat unsur teori yang relevan. Namun, teori dalam proposal tidak perlu terlalu panjang. Cukup jelaskan konsep inti yang menjadi dasar proyek.

Jika proyek tentang HOMO-LUMO, latar belakang harus menyebut orbital molekul, HOMO, LUMO, dan gap.

Jika proyek tentang IR, latar belakang harus menyebut frekuensi vibrasi, intensitas IR, dan gugus fungsi.

Jika proyek tentang efek pelarut, latar belakang harus menyebut polaritas, ion, medium dielektrik, dan stabilisasi elektrostatik.

Jika proyek tentang konformer, latar belakang harus menyebut rotasi ikatan, energi relatif, dan kestabilan konformasi.

Jika proyek tentang ikatan hidrogen, latar belakang harus menyebut interaksi antarmolekul, donor-akseptor, dan energi dimerisasi.

Dengan demikian, latar belakang tidak berdiri sendiri sebagai uraian umum, tetapi menjadi dasar bagi rumusan masalah.

9. Komponen 3: Rumusan Masalah

Rumusan masalah adalah pertanyaan utama yang akan dijawab melalui proyek mini. Rumusan masalah harus spesifik, dapat dijawab dengan data komputasi, dan sesuai dengan waktu praktikum.

Rumusan masalah yang kurang baik:

Bagaimana sifat molekul organik?

Pertanyaan ini terlalu luas.

Rumusan masalah yang lebih baik:

Bagaimana substituen amino dan nitro memengaruhi energi HOMO, LUMO, dan gap HOMO-LUMO pada benzena tersubstitusi?

Contoh rumusan masalah lain:

Bagaimana perbedaan frekuensi IR komputasi formaldehida dibandingkan dengan wilayah serapan karbonil dalam literatur?

Bagaimana pelarut polar memengaruhi energi relatif ion amonium dibanding fase gas?

Konformer butana manakah yang lebih stabil berdasarkan energi relatif hasil optimasi geometri?

Bagaimana simetri molekul memengaruhi momen dipol total pada H_2O , CO_2 , NH_3 , dan BF_3 ?

Berapa energi dimerisasi air dan bagaimana hasil tersebut menjelaskan kestabilan ikatan hidrogen?

Bagaimana perbedaan fase gas dan pelarut memengaruhi ΔG hidrasi formaldehida?

Rumusan masalah harus menghasilkan data yang jelas. Jika pertanyaan tidak dapat dijawab oleh output komputasi yang direncanakan, maka rumusan masalah harus diperbaiki.

10. Ciri Rumusan Masalah yang Baik

Rumusan masalah yang baik memiliki ciri-ciri:

1. Berbentuk pertanyaan.
2. Fokus pada satu masalah utama.
3. Mengandung variabel atau data yang dapat dihitung.
4. Berkaitan dengan teori kimia.
5. Sesuai dengan metode yang digunakan.
6. Tidak terlalu luas.
7. Tidak menuntut data yang tidak dapat diperoleh.

Contoh rumusan yang tidak realistis:

Apakah molekul obat ini efektif menyembuhkan penyakit tertentu?

Pertanyaan ini tidak dapat dijawab hanya dengan optimasi geometri atau HOMO-LUMO sederhana. Pertanyaan tersebut memerlukan data farmakologi, biologi, docking, dinamika molekul, dan uji eksperimental.

Rumusan yang lebih realistis:

Bagaimana geometri teroptimasi dan momen dipol parasetamol menggambarkan keberadaan gugus fungsi polar?

Pertanyaan ini dapat dijawab dengan optimasi geometri dan analisis momen dipol.

11. Komponen 4: Molekul Target

Molekul target adalah molekul atau sistem kimia yang akan dihitung. Pemilihan molekul target harus dijelaskan dalam proposal. Mahasiswa tidak boleh hanya menulis nama molekul tanpa alasan.

Bagian molekul target harus memuat:

1. Nama molekul.
2. Rumus molekul.
3. Struktur atau deskripsi gugus fungsi.
4. Charge.
5. Multiplicity.
6. Alasan pemilihan.
7. Kelayakan komputasi.
8. Data yang diharapkan.

Contoh tabel molekul target:

Molekul	Rumus	Charge	Multiplicity	Alasan Pemilihan
Benzena	C ₆ H ₆	0	1	Sistem aromatik dasar

Anilina	C_6H_7N	0	1	Aromatik dengan substituen donor elektron
Nitrobenzena	$C_6H_5NO_2$	0	1	Aromatik dengan substituen penarik elektron

Contoh untuk proyek efek pelarut:

Spesies	Rumus	Charge	Multiplicity	Keterangan
NH_3	NH_3	0	1	Basa netral
NH_4^+	NH_4^+	+1	1	Ion amonium
H_2O	H_2O	0	1	Pelarut atau pembanding polar

Molekul target harus sesuai kemampuan komputasi. Molekul terlalu besar dapat menyebabkan waktu komputasi panjang dan output sulit dianalisis.

12. Prinsip Pemilihan Molekul Target

Pemilihan molekul target harus mempertimbangkan beberapa hal.

Pertama, molekul harus relevan dengan topik. Jika proyek tentang IR karbonil, pilih molekul yang memiliki gugus karbonil, misalnya formaldehida, asetaldehida, aseton, atau asam asetat.

Kedua, molekul harus cukup sederhana. Untuk proyek mini, molekul kecil atau sedang lebih baik daripada molekul besar.

Ketiga, molekul harus memiliki sifat yang dapat dianalisis. Jika proyek tentang momen dipol, pilih molekul dengan perbedaan simetri atau polaritas.

Keempat, molekul harus dapat divalidasi. Struktur hasil optimasi harus dapat dinilai masuk akal.

Kelima, molekul harus memiliki data literatur atau teori pembanding. Misalnya, wilayah IR gugus karbonil atau polaritas H_2O dan CO_2 .

Prinsipnya:

```
molekul target
= relevan
+ sederhana
+ dapat dihitung
+ dapat divalidasi
+ dapat dianalisis
```


13. Komponen 5: Metode Komputasi

Metode komputasi adalah pendekatan teori yang digunakan untuk menghitung struktur dan sifat molekul. Dalam proposal, mahasiswa harus menuliskan metode dengan jelas, misalnya HF, DFT, B3LYP, PBE0, semiempiris, atau metode lain yang tersedia.

Metode komputasi tidak boleh ditulis secara kabur. Menulis “menggunakan komputer” atau “menggunakan software” tidak cukup. Software adalah alat, sedangkan metode adalah pendekatan teorinya.

Contoh penulisan yang kurang tepat:

Perhitungan dilakukan dengan Gaussian.

Contoh penulisan yang lebih baik:

Perhitungan dilakukan menggunakan metode DFT dengan fungsional B3LYP dan basis set 6-31G(d).

Metode komputasi harus disesuaikan dengan tujuan proyek. Untuk pembelajaran dasar, metode seperti HF/STO-3G dapat digunakan untuk demonstrasi konsep, tetapi untuk interpretasi struktur dan frekuensi yang lebih wajar, DFT dengan basis set yang memadai lebih dianjurkan jika tersedia.

14. Basis Set dalam Proposal

Selain metode, proposal harus mencantumkan basis set. Basis set adalah kumpulan fungsi matematika yang digunakan untuk merepresentasikan orbital atom. Contoh basis set:

STO-3G
3-21G
6-31G
6-31G(d)
6-31G(d,p)
def2-SVP
def2-TZVP

Pemilihan basis set harus realistis. Basis set kecil lebih cepat tetapi kurang akurat. Basis set lebih besar lebih baik tetapi membutuhkan waktu komputasi lebih besar.

Contoh penulisan:

Basis set 6-31G(d) dipilih karena memiliki fungsi polarisasi yang lebih baik dibanding basis set minimal, sehingga lebih sesuai untuk menggambarkan perubahan distribusi elektron pada molekul organik kecil.

Jika menggunakan basis set kecil karena keterbatasan waktu, tuliskan keterbatasannya:

Basis set STO-3G digunakan untuk tujuan pembelajaran dan efisiensi komputasi. Oleh karena itu, hasil dianalisis terutama secara kualitatif, bukan sebagai prediksi kuantitatif presisi tinggi.

Pernyataan seperti ini menunjukkan kejujuran metodologis.

15. Jenis Job Komputasi

Proposal juga harus menuliskan jenis job yang akan dijalankan. Jenis job bergantung pada data yang dibutuhkan.

Tujuan	Jenis Job
Mendapatkan struktur stabil	Optimasi geometri
Mendapatkan energi pada geometri tertentu	Single point
Memvalidasi minimum dan mendapatkan IR	Frekuensi
Mendapatkan energi reaksi termodinamika	Optimasi + frekuensi
Membandingkan orbital	Optimasi + analisis orbital
Menguji efek pelarut	Perhitungan gas dan pelarut
Membandingkan konformer	Optimasi tiap konformer

Contoh penulisan:

Jenis job yang digunakan adalah optimasi geometri dilanjutkan dengan perhitungan frekuensi. Optimasi digunakan untuk memperoleh struktur minimum, sedangkan frekuensi digunakan untuk memvalidasi tidak adanya frekuensi imajiner dan menghasilkan data IR.

Jika proyek tidak memerlukan frekuensi, jelaskan alasannya. Namun, untuk validasi struktur minimum, frekuensi sangat dianjurkan jika sumber daya memungkinkan.

16. Model Pelarut dalam Proposal

Jika proyek membahas efek pelarut, model pelarut harus ditulis secara jelas. Contoh model pelarut:

PCM
CPCM
SMD
COSMO

Proposal harus menyebutkan:

1. Apakah perhitungan dilakukan dalam fase gas.
2. Apakah perhitungan dilakukan dalam pelarut.
3. Nama pelarut yang digunakan.

4. Model pelarut.
5. Alasan penggunaan model.
6. Keterbatasan model.

Contoh penulisan:

Perhitungan dilakukan dalam fase gas dan dalam pelarut air menggunakan model pelarut kontinu. Perbandingan energi fase gas dan pelarut digunakan untuk menilai pengaruh pelarut terhadap kestabilan ion. Model pelarut kontinu dipilih karena praktis untuk proyek mini, tetapi tidak menggambarkan molekul pelarut eksplisit.

Jika semua spesies dalam reaksi dibandingkan, model pelarut harus diterapkan konsisten pada semua spesies.

17. Komponen 6: Software

Software adalah perangkat yang digunakan untuk menjalankan perhitungan atau mengolah data. Proposal harus mencantumkan software komputasi dan software pendukung.

Contoh software kimia komputasi:

1. ChemCompute.
2. Gaussian.
3. ORCA.
4. GAMESS.
5. Psi4.
6. PySCF.
7. Avogadro.
8. GaussView.
9. Molden.

Contoh software pengolahan data:

1. Python.
2. pandas.
3. matplotlib.
4. numpy.
5. spreadsheet jika diperlukan.

Bagian software harus menjelaskan fungsi masing-masing.

Contoh:

Avogadro digunakan untuk membuat struktur awal molekul. ChemCompute digunakan untuk menjalankan optimasi geometri dan perhitungan frekuensi. Python digunakan untuk mengolah data energi dan membuat grafik. Pustaka pandas digunakan untuk membaca CSV, sedangkan matplotlib digunakan untuk visualisasi grafik.

Jangan menulis daftar software tanpa fungsi. Proposal harus menunjukkan alasan pemakaian software.

18. Software Bukan Metode

Mahasiswa sering mencampurkan software dan metode. Ini harus dihindari.

Software adalah alat, misalnya Gaussian atau ORCA. Metode adalah pendekatan teori, misalnya HF, DFT, B3LYP, atau MP2. Basis set adalah representasi matematika orbital, misalnya 6-31G(d) atau def2-SVP.

Perbedaan:

Komponen	Contoh	Fungsi
Software	ORCA	Menjalankan perhitungan
Metode	B3LYP	Pendekatan teori
Basis set	6-31G(d)	Representasi orbital
Job	Opt Freq	Jenis perhitungan
Python	pandas/matplotlib	Pengolahan dan grafik

Penulisan proposal yang benar:

Perhitungan dijalankan dengan ORCA menggunakan metode B3LYP dan basis set def2-SVP untuk optimasi geometri dan frekuensi.

Penulisan yang kurang tepat:

Metode yang digunakan adalah ORCA.

ORCA bukan metode, melainkan software.

19. Komponen 7: Kode Python Pendukung

Kode Python pendukung adalah skrip yang digunakan untuk mengolah data proyek mini. Kode ini dapat digunakan untuk menghitung energi relatif, mengonversi satuan, menghitung gap HOMO-LUMO, membuat spektrum IR, membuat diagram energi, membaca output, atau menyusun tabel CSV.

Proposal harus menjelaskan kode Python yang akan digunakan.

Contoh:

Kode Python pendukung digunakan untuk membaca data HOMO dan LUMO dari CSV, menghitung gap HOMO-LUMO, serta membuat diagram orbital. Grafik disimpan dalam format PNG dengan resolusi 300 dpi.

Contoh lain:

Kode Python digunakan untuk menghitung energi dimerisasi air dengan rumus $\Delta E = E_{\text{dimer}} - 2E_{\text{monomer}}$. Energi dalam Hartree dikonversi ke kJ/mol menggunakan faktor 2625,5.

Kode Python pendukung tidak harus sangat kompleks. Yang penting, kode membantu pengolahan data secara konsisten dan reproduisibel.

20. Contoh Kode Python yang Direncanakan dalam Proposal

Untuk proyek HOMO-LUMO, kode Python dapat memuat:

1. Membaca file data_orbital.csv.
2. Mengambil kolom HOMO dan LUMO.
3. Menghitung gap HOMO-LUMO.
4. Membuat diagram orbital.
5. Menyimpan grafik.

Untuk proyek IR:

1. Membaca data frekuensi dan intensitas.
2. Membuat spektrum IR garis.
3. Membuat spektrum IR dengan pelebaran Gaussian.
4. Menyimpan grafik spektrum.

Untuk proyek energi reaksi:

1. Membaca energi reaktan dan produk.
2. Menghitung ΔE , ΔH , atau ΔG .
3. Mengonversi Hartree ke kJ/mol.
4. Membuat diagram energi reaksi.
5. Menyimpan CSV hasil perhitungan.

Untuk proyek efek pelarut:

1. Membaca energi fase gas dan pelarut.
2. Menghitung perubahan energi.
3. Membuat grafik perbandingan gas-pelarut.
4. Menulis tabel ringkasan.

Dalam proposal, mahasiswa cukup menjelaskan rencana kode. Kode lengkap dapat dilampirkan pada laporan akhir.

21. Komponen 8: Data yang Dikumpulkan

Data yang dikumpulkan harus ditentukan sejak proposal. Jangan menjalankan perhitungan tanpa mengetahui data apa yang dicari.

Contoh data untuk masing-masing proyek:

Proyek	Data Utama
HOMO-LUMO	HOMO, LUMO, gap, momen dipol
IR	Frekuensi, intensitas, spektrum, frekuensi imajiner
Optimasi obat	Energi total, geometri, panjang ikatan, momen dipol
Efek pelarut	Energi gas, energi pelarut, perubahan energi
Konformer	Energi relatif, sudut dihedral, momen dipol
Simetri-polaritas	Geometri, momen dipol, bentuk molekul
Dimer air	Energi monomer, energi dimer, jarak O...H
Energi reaksi	Energi reaktan-produk, ΔE , ΔH , ΔG

Proposal harus menyebut data mentah dan data olahan. Data mentah berasal dari output. Data olahan berasal dari perhitungan lanjutan, misalnya konversi satuan atau selisih energi.

Contoh:

Data mentah yang dikumpulkan adalah energi total dalam Hartree, frekuensi vibrasi dalam cm^{-1} , dan momen dipol dalam Debye. Data olahan meliputi energi relatif dalam kJ/mol dan grafik spektrum IR.

22. Validasi Data dalam Proposal

Proposal harus menyebut rencana validasi. Ini penting karena data yang dikumpulkan harus dipastikan layak digunakan.

Rencana validasi dapat mencakup:

1. Memeriksa job selesai normal.
2. Memeriksa SCF konvergen.
3. Memeriksa optimasi geometri konvergen.
4. Memeriksa frekuensi imajiner.
5. Memeriksa charge dan multiplicity.
6. Memeriksa geometri akhir.
7. Memeriksa konsistensi metode dan basis set.

8. Membandingkan tren dengan teori atau literatur.

Contoh penulisan:

Validasi dilakukan dengan memeriksa status job, konvergensi optimasi, jumlah frekuensi imajiner, geometri akhir, charge, dan multiplicity. Data hanya digunakan jika job selesai normal dan struktur minimum tidak memiliki frekuensi imajiner.

Proposal yang tidak menyebut validasi berisiko menghasilkan laporan yang lemah.

23. Komponen 9: Jadwal Kerja

Jadwal kerja adalah pembagian waktu pelaksanaan proyek. Jadwal membantu mahasiswa menghindari pekerjaan yang menumpuk di akhir.

Contoh jadwal kerja satu minggu:

Hari/Tahap	Kegiatan
Hari 1	Memilih topik dan molekul target
Hari 2	Menyusun struktur awal dan input
Hari 3	Menjalankan optimasi geometri
Hari 4	Menjalankan frekuensi atau perhitungan tambahan
Hari 5	Membaca output dan validasi
Hari 6	Mengolah data dengan Python dan membuat grafik
Hari 7	Menulis laporan dan menyiapkan presentasi

Contoh jadwal kerja dua minggu:

Minggu Kegiatan

Minggu 1 Proposal, struktur awal, input, perhitungan awal

Minggu 2 Validasi, pengolahan Python, grafik, laporan, presentasi

Jadwal harus realistis. Jika proyek memiliki banyak molekul, jadwal harus memberi waktu cukup untuk perhitungan ulang jika terjadi error.

24. Contoh Format Proposal Proyek Mini

Berikut format proposal yang dapat digunakan mahasiswa.

PROPOSAL PROYEK MINI PRAKTIKUM KIMIA KUANTUM

1. Judul
[Tulis judul proyek secara spesifik.]
2. Latar Belakang
[Jelaskan alasan pemilihan topik, konsep kimia yang terlibat, dan relevansi proyek.]
3. Rumusan Masalah
[Tulis 1-3 pertanyaan utama yang akan dijawab.]
4. Molekul Target
[Tulis nama molekul, rumus, charge, multiplicity, dan alasan pemilihan.]
5. Metode Komputasi
[Tulis metode, basis set, jenis job, model pelarut jika digunakan, dan alasan pemilihan.]
6. Software
[Tulis software kimia komputasi, editor molekul, dan Python/pustaka pendukung.]
7. Kode Python Pendukung
[Jelaskan fungsi kode Python yang akan digunakan.]
8. Data yang Dikumpulkan
[Tulis data mentah dan data olahan yang akan digunakan.]
9. Rencana Validasi
[Tulis cara memeriksa status job, konvergensi, frekuensi imajiner, charge, multiplicity, dan geometri.]
10. Jadwal Kerja
[Tulis jadwal kerja dalam bentuk tabel.]
11. Luaran Proyek
[Tulis bentuk output akhir: laporan, CSV, grafik, file input-output, dan presentasi.]

Meskipun silabus memuat sembilan komponen utama, format di atas menambahkan rencana validasi dan luaran proyek agar proposal lebih kuat. Rencana validasi dapat dimasukkan ke bagian data atau metode jika format harus dibuat lebih ringkas.

25. Contoh Proposal Singkat Proyek HOMO-LUMO

Judul:

Analisis HOMO-LUMO Benzena, Anilina, dan Nitrobenzena Menggunakan Metode DFT

Latar belakang:

Senyawa aromatik tersubstitusi memiliki sifat elektronik yang dipengaruhi oleh substituen. Substituen donor elektron dan penarik elektron dapat mengubah

distribusi elektron cincin aromatik. Analisis HOMO-LUMO dapat digunakan untuk membandingkan energi orbital terisi tertinggi, orbital kosong terendah, dan gap HOMO-LUMO. Benzena dipilih sebagai pembanding dasar, sedangkan anilina mewakili substituen donor elektron dan nitrobenzena mewakili substituen penarik elektron.

Rumusan masalah:

Bagaimana substituen amino dan nitro memengaruhi energi HOMO, LUMO, dan gap HOMO-LUMO dibanding benzena?

Molekul target:

Molekul	Rumus	Charge	Multiplicity
Benzena	C ₆ H ₆	0	1
Anilina	C ₆ H ₇ N	0	1
Nitrobenzena	C ₆ H ₅ NO ₂	0	1

Metode:

Perhitungan dilakukan dengan metode DFT menggunakan fungsional B3LYP dan basis set 6-31G(d). Struktur dioptimasi terlebih dahulu, kemudian energi HOMO dan LUMO diambil dari output.

Software:

Avogadro digunakan untuk membuat struktur awal. ChemCompute atau ORCA digunakan untuk menjalankan perhitungan. Python digunakan untuk menghitung gap HOMO-LUMO dan membuat diagram orbital.

Kode Python:

Kode Python membaca CSV berisi HOMO dan LUMO, menghitung gap, membuat diagram orbital, dan menyimpan grafik.

Data:

Energi HOMO, energi LUMO, gap HOMO-LUMO, energi total, momen dipol, status konvergensi.

Jadwal:

Tahap	Kegiatan
1	Membuat struktur dan input
2	Menjalankan optimasi
3	Membaca output
4	Mengolah data Python
5	Menulis laporan

Proposal ini cukup fokus dan realistis untuk proyek mini.

26. Contoh Proposal Singkat Proyek IR

Judul:

Perbandingan Spektrum IR Komputasi Formaldehida dengan Wilayah Serapan Literatur

Latar belakang:

Spektrum IR digunakan untuk mengidentifikasi gugus fungsi berdasarkan frekuensi vibrasi molekul. Formaldehida memiliki gugus karbonil yang menghasilkan puncak khas pada spektrum IR. Perhitungan frekuensi komputasi dapat digunakan untuk memprediksi bilangan gelombang dan intensitas IR. Perbandingan hasil komputasi dengan wilayah serapan literatur membantu memahami hubungan antara struktur, ikatan, dan vibrasi molekul.

Rumusan masalah:

Apakah frekuensi vibrasi formaldehida hasil komputasi sesuai dengan wilayah serapan khas gugus karbonil dan C-H dalam literatur?

Molekul target:

Molekul	Rumus	Charge	Multiplicity
Formaldehida	CH ₂ O	0	1

Metode:

Perhitungan dilakukan melalui optimasi geometri dan frekuensi. Frekuensi digunakan untuk memvalidasi struktur minimum dan membuat spektrum IR.

Software:

ChemCompute atau ORCA digunakan untuk perhitungan. Python digunakan untuk membuat spektrum IR dari data frekuensi dan intensitas.

Data:

Frekuensi vibrasi, intensitas IR, jumlah frekuensi imajiner, energi total, struktur akhir.

Validasi:

Struktur dianggap valid jika optimasi konvergen dan tidak memiliki frekuensi imajiner.

27. Contoh Proposal Singkat Proyek Dimer Air

Judul:

Analisis Energi Ikatan Hidrogen pada Dimer Air

Latar belakang:

Ikatan hidrogen merupakan interaksi antarmolekul penting yang memengaruhi sifat air dan banyak sistem kimia. Dimer air dapat digunakan sebagai model sederhana untuk memahami pembentukan ikatan hidrogen. Dengan membandingkan energi dimer air dan dua monomer air, energi dimerisasi dapat dihitung secara komputasi.

Rumusan masalah:

Berapa energi dimerisasi air dan bagaimana hasil tersebut menjelaskan kestabilan ikatan hidrogen?

Molekul target:

Spesies	Rumus	Charge	Multiplicity
Monomer air	H ₂ O	0	1
Dimer air	(H ₂ O) ₂	0	1

Metode:

Monomer dan dimer air dioptimasi menggunakan metode dan basis set yang sama. Energi dimerisasi dihitung dengan rumus $\Delta E = E_{\text{dimer}} - 2E_{\text{monomer}}$.

Software:

Avogadro digunakan untuk membuat struktur awal dimer. Software komputasi digunakan untuk optimasi dan energi. Python digunakan untuk menghitung ΔE dan membuat diagram energi.

Data:

Energi monomer, energi dimer, jarak O...H, jumlah frekuensi imajiner, diagram energi dimerisasi.

Keterbatasan:

Energi ikatan hidrogen sensitif terhadap basis set dan kemungkinan basis set superposition error. Oleh karena itu, hasil ditafsirkan sebagai model pembelajaran.

28. Analisis Kritis terhadap Proposal Proyek Mini

Proposal yang baik tidak hanya lengkap secara bagian, tetapi juga logis secara ilmiah. Ada beberapa pertanyaan kritis yang harus diajukan sebelum proposal disetujui.

Pertama, apakah rumusan masalah dapat dijawab oleh data yang direncanakan?

Jika rumusan masalah tentang efek pelarut, tetapi metode tidak mencantumkan model pelarut, maka proposal tidak konsisten.

Kedua, apakah molekul target sesuai dengan kemampuan komputasi?

Jika mahasiswa memilih molekul obat besar dengan banyak konformer tetapi hanya memiliki waktu singkat, proyek terlalu berat.

Ketiga, apakah metode dan basis set konsisten?

Jika reaktan dan produk akan dibandingkan, semuanya harus dihitung dengan metode dan basis set yang sama.

Keempat, apakah validasi direncanakan?

Jika proposal tidak menyebutkan status job, konvergensi, frekuensi imajiner, charge, atau multiplicity, hasil akhir dapat lemah.

Kelima, apakah Python benar-benar digunakan untuk mengolah data?

Jika Python hanya disebutkan tetapi tidak jelas fungsinya, proposal belum kuat.

Dengan demikian, proposal harus dinilai bukan hanya dari kelengkapan format, tetapi juga dari koherensi antara masalah, metode, data, dan analisis.

29. Kesalahan Umum dalam Menyusun Proposal

Beberapa kesalahan umum dalam proposal proyek mini adalah:

Kesalahan	Dampak
Judul terlalu luas	Fokus proyek tidak jelas
Latar belakang terlalu umum	Tidak menunjukkan alasan ilmiah
Rumusan masalah tidak spesifik	Data sulit diarahkan
Molekul terlalu besar	Perhitungan sulit selesai
Charge tidak ditulis	Sistem tidak jelas
Multiplicity tidak ditulis	Keadaan spin tidak jelas
Metode tidak lengkap	Hasil tidak reproduibel

Basis set tidak disebut	Level teori tidak jelas
Software dianggap metode	Kekeliruan konsep
Data tidak dirinci	Output yang dicari tidak jelas
Tidak ada rencana validasi	Hasil berisiko tidak sah
Jadwal tidak realistis	Proyek sulit selesai
Python tidak jelas perannya	Otomatisasi tidak terarah

Kesalahan ini harus diperbaiki sebelum proyek dijalankan.

30. Rubrik Penilaian Proposal Proyek Mini

Proposal dapat dinilai dengan rubrik berikut:

Aspek	Kriteria
Judul	Spesifik, jelas, mencerminkan isi proyek
Latar belakang	Relevan, berbasis konsep kimia, tidak terlalu umum
Rumusan masalah	Fokus, dapat dijawab dengan data komputasi
Molekul target	Layak, jelas charge dan multiplicity
Metode komputasi	Metode, basis set, dan job ditulis lengkap
Software	Software dan fungsi penggunaannya jelas
Python	Peran kode Python dijelaskan
Data	Data mentah dan olahan dirinci
Validasi	Rencana validasi output jelas
Jadwal	Realistis dan bertahap
Kelayakan keseluruhan	Proyek dapat dilaksanakan dalam waktu praktikum

Rubrik ini membantu dosen menilai proposal secara objektif dan membantu mahasiswa memperbaiki proposal sebelum proyek dimulai.

31. Implikasi Keilmuan Proposal Proyek Mini

Proposal proyek mini memiliki implikasi keilmuan karena mengajarkan bahwa kerja komputasi ilmiah harus dirancang sebelum dijalankan. Dalam kimia komputasi, hasil sangat bergantung pada keputusan awal. Molekul yang dipilih, metode yang digunakan, basis set, model pelarut, charge, multiplicity, dan jenis job semuanya memengaruhi hasil.

Dengan menyusun proposal, mahasiswa belajar bahwa sains bukan sekadar memperoleh angka, tetapi merancang cara memperoleh angka yang bermakna. Proposal menghubungkan pertanyaan ilmiah dengan metode yang sesuai.

Alur keilmuan proposal dapat diringkas:

pertanyaan kimia
 → model molekul
 → metode komputasi
 → output
 → validasi
 → analisis

Jika alur ini benar, proyek memiliki dasar ilmiah yang kuat.

32. Implikasi Pedagogis Proposal Proyek Mini

Secara pedagogis, proposal melatih mahasiswa berpikir sebelum bekerja. Mahasiswa tidak langsung menjalankan software, tetapi terlebih dahulu menyusun rencana. Ini penting karena dalam kimia komputasi, perhitungan yang salah sejak awal dapat menghabiskan waktu dan menghasilkan data yang tidak berguna.

Keterampilan yang dilatih melalui proposal:

1. Merumuskan masalah.
2. Membatasi ruang lingkup.
3. Memilih molekul.
4. Menentukan charge dan multiplicity.
5. Memilih metode.
6. Menentukan data.
7. Merancang validasi.
8. Mengatur jadwal.
9. Menyiapkan pengolahan Python.
10. Mengantisipasi kesalahan.

Keterampilan ini penting bagi calon guru kimia karena guru perlu mampu merancang aktivitas komputasi yang jelas, realistis, dan bermakna bagi peserta didik.

33. Hubungan Proposal dengan Laporan Akhir

Proposal adalah dasar laporan akhir. Bagian-bagian proposal akan berkembang menjadi bagian laporan.

Hubungannya dapat dilihat sebagai berikut:

Proposal	Laporan Akhir
Judul	Judul laporan
Latar belakang	Pendahuluan
Rumusan masalah	Tujuan dan fokus analisis

Molekul target	Sistem yang dikaji
Metode komputasi	Metode penelitian
Software	Alat komputasi
Kode Python	Lampiran kode dan pengolahan data
Data yang dikumpulkan	Hasil dan tabel
Jadwal kerja	Dokumentasi proses
Validasi	Validasi data
Luaran proyek	Laporan, grafik, presentasi

Jika proposal disusun dengan baik, laporan akhir akan lebih mudah ditulis. Sebaliknya, proposal yang lemah sering menghasilkan laporan yang tidak fokus.

34. Template Ringkas Proposal Proyek Mini Siap Pakai

Berikut template ringkas yang dapat langsung digunakan mahasiswa.

PROPOSAL PROYEK MINI PRAKTIKUM KIMIA KUANTUM

A. Judul

.....

B. Latar Belakang

.....

C. Rumusan Masalah

1.
2.

D. Molekul Target

Nama molekul:
 Rumus molekul:
 Charge:
 Multiplicity:
 Alasan pemilihan:

E. Metode Komputasi

Metode:
 Basis set:
 Jenis job:
 Model pelarut:
 Alasan pemilihan metode:

F. Software

Editor molekul:
 Software komputasi:
 Software/Pustaka Python:
 Fungsi masing-masing software:

G. Kode Python Pendukung

Nama file kode:

Fungsi kode:

Output kode:

H. Data yang Dikumpulkan

Data mentah:

Data olahan:

Satuan:

Grafik yang dibuat:

I. Rencana Validasi

Status job:

SCF:

Optimasi:

Frekuensi imajiner:

Geometri:

Charge dan multiplicity:

Perbandingan teori/literatur:

J. Jadwal Kerja

Tahap 1:

Tahap 2:

Tahap 3:

Tahap 4:

Tahap 5:

K. Luaran Proyek

File input:

File output:

CSV:

Grafik:

Laporan:

Presentasi:

Template ini dapat disesuaikan dengan topik proyek mini yang dipilih.

35. Sintesis Subbab 12.2

Proposal proyek mini merupakan dokumen perencanaan yang mengarahkan mahasiswa sebelum menjalankan proyek Praktikum Kimia Kuantum. Proposal harus memuat judul, latar belakang, rumusan masalah, molekul target, metode komputasi, software, kode Python pendukung, data yang dikumpulkan, dan jadwal kerja. Proposal yang baik juga mencantumkan rencana validasi dan luaran proyek agar pekerjaan lebih terarah.

Alur proposal dapat diringkas sebagai berikut:

judul

→ latar belakang

→ rumusan masalah

→ molekul target

- metode komputasi
- software
- kode Python
- data
- validasi
- jadwal
- luaran

Dengan proposal, mahasiswa belajar merancang proyek secara ilmiah. Proposal membantu membatasi topik, memilih metode yang realistis, menentukan data yang akan dikumpulkan, menyiapkan validasi, dan mengatur waktu kerja. Proposal juga menjadi dasar bagi laporan akhir proyek mini.

36. Transisi ke Subbab 12.3

Subbab 12.2 telah membahas format proposal proyek mini, mulai dari judul, latar belakang, rumusan masalah, molekul target, metode komputasi, software, kode Python pendukung, data yang dikumpulkan, validasi, jadwal kerja, hingga luaran proyek. Proposal ini menjadi rencana awal sebelum mahasiswa menjalankan proyek mini secara penuh.

Subbab berikutnya akan membahas format laporan proyek mini. Pada subbab 12.3, mahasiswa akan mempelajari cara menyusun laporan akhir proyek yang memuat abstrak, pendahuluan, dasar teori, metode komputasi, kode Python, hasil dan pembahasan, kesimpulan, daftar pustaka, serta lampiran input-output. Dengan demikian, proposal yang disusun pada subbab 12.2 akan dikembangkan menjadi laporan ilmiah lengkap pada subbab 12.3.

12.3 Format Laporan Proyek Mini

Format laporan proyek mini adalah susunan sistematis untuk menuliskan hasil proyek mini Praktikum Kimia Kuantum setelah mahasiswa menyelesaikan tahap perencanaan, perhitungan, validasi output, pengolahan data, pembuatan grafik, dan analisis. Jika proposal pada subbab 12.2 berfungsi sebagai **rencana kerja**, maka laporan proyek mini berfungsi sebagai **dokumen pertanggungjawaban ilmiah**. Laporan menunjukkan apa yang dilakukan mahasiswa, bagaimana perhitungan dilakukan, data apa yang diperoleh, apakah data tersebut valid, bagaimana data diolah dengan Python, dan kesimpulan apa yang dapat ditarik berdasarkan hasil.

Laporan proyek mini tidak boleh hanya berisi kumpulan gambar struktur molekul dan tabel energi. Laporan harus menunjukkan alur ilmiah yang utuh:

- masalah kimia
- dasar teori
- metode komputasi
- output
- validasi
- pengolahan Python
- grafik

- pembahasan
- kesimpulan
- lampiran input-output

Dalam proyek mini kimia kuantum, laporan menjadi bukti bahwa mahasiswa tidak sekadar menjalankan software, tetapi mampu melakukan kerja ilmiah berbasis data komputasi. Oleh karena itu, laporan harus jelas, rapi, argumentatif, kritis, dan reproduibel.

Format laporan proyek mini terdiri atas sembilan bagian utama:

1. **Abstrak.**
2. **Pendahuluan.**
3. **Dasar teori.**
4. **Metode komputasi.**
5. **Kode Python.**
6. **Hasil dan pembahasan.**
7. **Kesimpulan.**
8. **Daftar pustaka.**
9. **Lampiran input-output.**

Kesembilan bagian ini tidak boleh dipahami sebagai formalitas. Setiap bagian memiliki fungsi metodologis. Abstrak merangkum keseluruhan proyek. Pendahuluan menjelaskan masalah. Dasar teori memberi landasan konsep. Metode komputasi menjelaskan cara memperoleh data. Kode Python menjelaskan pengolahan data. Hasil dan pembahasan menafsirkan output. Kesimpulan menjawab rumusan masalah. Daftar pustaka menunjukkan sumber akademik yang digunakan. Lampiran input-output memastikan keterlacakan dan reproduibilitas.

1. Pengertian Laporan Proyek Mini

Laporan proyek mini adalah dokumen akademik yang memuat hasil pelaksanaan proyek komputasi dalam ruang lingkup terbatas. Laporan ini menggabungkan unsur laporan praktikum, laporan penelitian kecil, dan dokumentasi komputasi.

Dalam konteks Praktikum Kimia Kuantum, laporan proyek mini harus mampu menjawab pertanyaan berikut:

Apa topik proyek?
 Mengapa topik itu dipilih?
 Molekul apa yang dihitung?
 Metode apa yang digunakan?
 Software apa yang dipakai?
 Output apa yang diperoleh?
 Apakah output valid?
 Bagaimana data diolah?
 Grafik apa yang dibuat?
 Apa makna kimianya?
 Apa kesimpulan akhirnya?

Jika laporan tidak mampu menjawab pertanyaan tersebut, maka laporan belum lengkap.

Laporan proyek mini juga menjadi tempat mahasiswa menunjukkan keterampilan integratif. Mahasiswa menggabungkan teori kimia kuantum, teknik komputasi, validasi output, analisis data, visualisasi Python, dan komunikasi ilmiah. Dengan demikian, laporan proyek mini adalah hasil akhir dari seluruh rangkaian pembelajaran praktikum.

2. Latar Historis Penulisan Laporan Komputasi

Dalam sains modern, laporan ilmiah berkembang sebagai sarana untuk menyampaikan hasil penelitian secara terbuka, terstruktur, dan dapat diuji ulang. Pada kimia eksperimen, laporan memuat bahan, alat, prosedur, pengamatan, perhitungan, dan pembahasan. Pada kimia komputasi, objek yang dilaporkan berbeda: struktur molekul, file input, parameter metode, basis set, output numerik, status konvergensi, kode pengolahan data, dan visualisasi hasil.

Sejak berkembangnya kimia komputasi sebagai disiplin yang menghubungkan mekanika kuantum, kimia fisik, metode numerik, dan pemodelan molekul, dokumentasi menjadi sangat penting. Buku-buku dan panduan bereputasi dalam kimia komputasi, seperti karya Jensen, Cramer, Leach, Lewars, Levine, Szabo dan Ostlund, serta Foresman dan Frisch, menekankan bahwa hasil komputasi tidak dapat dilepaskan dari metode, basis set, dan asumsi yang digunakan. Hasil energi, geometri, orbital, dan frekuensi selalu bergantung pada level teori.

Karena itu, laporan komputasi harus mencantumkan informasi yang cukup agar pembaca dapat memahami dan menilai hasil. Laporan yang hanya menulis “hasil dihitung dengan software tertentu” belum memadai. Laporan harus mencantumkan metode, basis set, charge, multiplicity, jenis job, model pelarut jika ada, status validasi, dan cara pengolahan data. Inilah perbedaan mendasar antara laporan visualisasi molekul biasa dan laporan kimia komputasi yang ilmiah.

3. Fungsi Laporan Proyek Mini

Laporan proyek mini memiliki beberapa fungsi penting.

Pertama, laporan berfungsi sebagai **dokumen hasil kerja**. Semua aktivitas proyek dirangkum dalam satu naskah.

Kedua, laporan berfungsi sebagai **alat komunikasi ilmiah**. Pembaca dapat memahami masalah, metode, hasil, dan kesimpulan tanpa harus menanyakan kembali kepada penulis.

Ketiga, laporan berfungsi sebagai **alat evaluasi**. Dosen dapat menilai kemampuan mahasiswa dalam merancang, menjalankan, memvalidasi, dan menafsirkan proyek.

Keempat, laporan berfungsi sebagai **arsip reproduibilitas**. File input, output, CSV, grafik, dan kode Python dapat dilacak melalui laporan dan lampiran.

Kelima, laporan berfungsi sebagai **latihan berpikir kritis**. Mahasiswa dilatih untuk tidak hanya menampilkan hasil, tetapi juga menilai keterbatasan hasil.

Keenam, laporan berfungsi sebagai **jembatan menuju penelitian**. Walaupun proyek mini kecil, struktur laporannya melatih pola penulisan ilmiah yang lebih luas.

Dengan demikian, laporan proyek mini bukan sekadar tugas akhir praktikum, tetapi latihan akademik yang penting.

4. Prinsip Umum Laporan Proyek Mini

Laporan proyek mini harus mengikuti beberapa prinsip.

Pertama, laporan harus **sistematis**. Bagian-bagiannya tersusun urut dari abstrak sampai lampiran.

Kedua, laporan harus **reproduibel**. Pembaca dapat mengetahui bagaimana perhitungan dilakukan dan data diperoleh.

Ketiga, laporan harus **berbasis data**. Pembahasan dan kesimpulan harus didukung oleh output, tabel, dan grafik.

Keempat, laporan harus **tervalidasi**. Data yang digunakan harus berasal dari output yang telah diperiksa.

Kelima, laporan harus **argumentatif**. Mahasiswa harus menjelaskan mengapa hasil terjadi, bukan hanya menyebut angka.

Keenam, laporan harus **kritis**. Keterbatasan metode, basis set, model pelarut, pendekatan harmonik, atau data literatur harus disebutkan jika relevan.

Ketujuh, laporan harus **jujur**. Kesalahan, output gagal, frekuensi imajiner, atau keterbatasan data tidak boleh disembunyikan.

Prinsip tersebut dapat diringkas:

```

laporan baik
= data valid
+ metode jelas
+ analisis kuat
+ grafik informatif
+ kesimpulan hati-hati
+ lampiran lengkap

```

5. Struktur Umum Laporan Proyek Mini

Struktur laporan proyek mini yang dianjurkan adalah sebagai berikut:

```
JUDUL LAPORAN
IDENTITAS MAHASISWA

ABSTRAK
1. PENDAHULUAN
2. DASAR TEORI
3. METODE KOMPUTASI
4. KODE PYTHON
5. HASIL DAN PEMBAHASAN
6. KESIMPULAN
7. DAFTAR PUSTAKA
8. LAMPIRAN INPUT-OUTPUT
```

Struktur ini dapat digunakan untuk semua pilihan proyek mini, baik proyek HOMO-LUMO, IR, optimasi geometri, efek pelarut, konformer, simetri-polaritas, dimer air, maupun energi reaksi.

Perbedaan antarproyek terutama terletak pada isi hasil, jenis grafik, dan fokus pembahasan. Struktur utamanya tetap sama agar laporan mudah dinilai dan dibandingkan.

6. Bagian Judul dan Identitas

Sebelum abstrak, laporan harus memuat judul dan identitas mahasiswa. Judul harus sama dengan judul proyek yang telah disetujui dalam proposal, kecuali ada revisi yang disetujui dosen.

Format identitas:

```
LAPORAN PROYEK MINI PRAKTIKUM KIMIA KUANTUM
```

```
Judul:
[Nama proyek mini]
```

```
Nama Mahasiswa:
[Nama]
```

```
NIM:
[NIM]
```

```
Kelas:
[Kelas]
```

```
Program Studi:
[Program Studi]
```

Dosen/Instruktur:
[Nama dosen/instruktur]

Tanggal Pelaksanaan:
[Tanggal]

Tanggal Pengumpulan:
[Tanggal]

Judul harus spesifik. Contoh judul yang baik:

Analisis HOMO-LUMO Benzena, Anilina, dan Nitrobenzena Menggunakan Metode DFT

atau:

Perbandingan Spektrum IR Komputasi Formaldehida dengan Wilayah Serapan Literatur

Judul yang terlalu umum harus dihindari, misalnya:

Laporan Kimia Komputasi Molekul Organik

Judul tersebut tidak menunjukkan molekul, konsep, atau metode.

7. Bagian Abstrak

Abstrak adalah ringkasan singkat dari keseluruhan proyek. Abstrak harus memuat tujuan, metode, molekul target, data utama, hasil penting, dan kesimpulan. Panjang abstrak cukup sekitar 150–250 kata. Abstrak tidak perlu memuat tabel, gambar, atau uraian teori panjang.

Struktur abstrak:

Kalimat 1: tujuan proyek.
Kalimat 2: molekul target dan metode.
Kalimat 3: data yang dianalisis.
Kalimat 4: hasil utama.
Kalimat 5: kesimpulan dan keterbatasan singkat.

Contoh abstrak untuk proyek HOMO-LUMO:

Proyek mini ini bertujuan menganalisis pengaruh substituen terhadap energi HOMO, LUMO, dan gap HOMO-LUMO pada senyawa aromatik sederhana. Molekul yang dikaji adalah benzena, anilina, dan nitrobenzena. Perhitungan dilakukan melalui optimasi geometri menggunakan metode DFT pada level teori yang ditetapkan, kemudian energi orbital diambil dari output komputasi dan diolah menggunakan Python. Hasil menunjukkan bahwa substituen dapat mengubah energi orbital dan gap HOMO-LUMO dibanding benzena. Anilina sebagai senyawa dengan substituen donor elektron dan nitrobenzena sebagai senyawa dengan substituen penarik elektron menunjukkan perbedaan sifat elektronik yang dapat dijelaskan melalui pengaruh

substituen terhadap distribusi elektron cincin aromatik. Analisis dilakukan secara kualitatif karena hasil bergantung pada metode, basis set, dan asumsi komputasi yang digunakan.

Abstrak harus ditulis setelah semua bagian laporan selesai agar ringkasannya sesuai isi.

8. Kesalahan Umum dalam Menulis Abstrak

Beberapa kesalahan abstrak yang harus dihindari:

Kesalahan	Dampak
Terlalu umum	Tidak menggambarkan isi proyek
Tidak menyebut molekul	Objek kajian tidak jelas
Tidak menyebut metode	Hasil tidak terhubung dengan cara perhitungan
Tidak menyebut hasil utama	Abstrak tidak informatif
Terlalu panjang	Tidak lagi menjadi ringkasan
Berisi teori terlalu banyak	Mengaburkan hasil
Berisi klaim berlebihan	Tidak sesuai data
Menyebut kesimpulan yang tidak dibahas	Tidak konsisten

Abstrak yang baik harus padat, jelas, dan berbasis hasil.

9. Bagian Pendahuluan

Pendahuluan menjelaskan latar belakang masalah, alasan pemilihan topik, rumusan masalah, tujuan, dan batasan proyek. Pendahuluan adalah bagian yang menjawab mengapa proyek dilakukan.

Struktur pendahuluan:

1. Konteks umum topik.
2. Pentingnya topik dalam kimia kuantum/kimia komputasi.
3. Alasan pemilihan molekul target.
4. Rumusan masalah.
5. Tujuan proyek.
6. Batasan proyek.

Contoh pendahuluan untuk proyek IR:

Spektrum inframerah merupakan salah satu metode penting untuk mengidentifikasi gugus fungsi dalam molekul. Dalam kimia komputasi, frekuensi vibrasi dapat dihitung melalui analisis frekuensi setelah optimasi geometri. Formaldehida dipilih sebagai molekul target karena memiliki gugus karbonil yang memberikan serapan khas pada wilayah IR. Proyek ini dilakukan untuk membandingkan frekuensi

vibrasi hasil komputasi dengan wilayah serapan literatur, terutama untuk mode regangan C=O dan C-H. Proyek dibatasi pada analisis frekuensi komputasi dan intensitas IR dari satu molekul formaldehida pada level teori yang digunakan.

Pendahuluan harus mengarahkan pembaca pada fokus proyek. Jangan menulis pendahuluan terlalu luas, misalnya membahas seluruh sejarah kimia kuantum jika proyek hanya membandingkan IR formaldehida.

10. Rumusan Masalah dan Tujuan dalam Pendahuluan

Rumusan masalah dan tujuan harus jelas. Rumusan masalah berbentuk pertanyaan, sedangkan tujuan berbentuk pernyataan kerja.

Contoh:

Rumusan masalah:

Bagaimana perbandingan frekuensi IR komputasi formaldehida dengan wilayah serapan literatur untuk gugus karbonil dan C-H?

Tujuan:

Menganalisis frekuensi vibrasi dan intensitas IR formaldehida hasil perhitungan komputasi serta membandingkannya dengan wilayah serapan literatur.

Untuk proyek HOMO-LUMO:

Rumusan masalah:

Bagaimana substituen amino dan nitro memengaruhi energi HOMO, LUMO, dan gap HOMO-LUMO pada senyawa aromatik?

Tujuan:

Membandingkan energi HOMO, LUMO, dan gap HOMO-LUMO benzena, anilina, dan nitrobenzena berdasarkan hasil perhitungan komputasi.

Rumusan masalah dan tujuan harus sejalan. Jangan membuat tujuan yang tidak menjawab rumusan masalah.

11. Bagian Dasar Teori

Dasar teori memuat konsep-konsep yang digunakan untuk menafsirkan hasil. Bagian ini tidak perlu terlalu panjang, tetapi harus cukup kuat untuk menjadi landasan analisis.

Dasar teori harus disesuaikan dengan topik proyek.

Untuk proyek HOMO-LUMO, dasar teori memuat:

1. Orbital molekul.
2. HOMO dan LUMO.
3. Gap HOMO-LUMO.
4. Pengaruh substituen.
5. Interpretasi reaktivitas secara kualitatif.

Untuk proyek IR, dasar teori memuat:

1. Vibrasi molekul.
2. Frekuensi vibrasi.
3. Intensitas IR.
4. Gugus fungsi.
5. Pendekatan harmonik.
6. Perbedaan IR komputasi dan eksperimen.

Untuk proyek optimasi geometri, dasar teori memuat:

1. Permukaan energi potensial.
2. Optimasi geometri.
3. Struktur minimum.
4. Konvergensi.
5. Frekuensi imajiner.

Untuk proyek efek pelarut, dasar teori memuat:

1. Polaritas.
2. Ion dan solvasi.
3. Model pelarut kontinu.
4. Konstanta dielektrik.
5. Stabilitas relatif.

Untuk proyek konformer, dasar teori memuat:

1. Rotasi ikatan tunggal.
2. Konformasi.
3. Energi relatif.
4. Tolakan sterik.
5. Sudut dihedral.

Untuk proyek simetri-polaritas, dasar teori memuat:

1. Simetri molekul.
2. Vektor dipol.
3. Momen dipol total.
4. Bentuk molekul.
5. Polaritas ikatan dan polaritas molekul.

Untuk proyek dimer air, dasar teori memuat:

1. Ikatan hidrogen.
2. Interaksi antarmolekul.
3. Energi dimerisasi.
4. Basis set dan keterbatasan interaksi lemah.

Untuk proyek energi reaksi, dasar teori memuat:

1. Energi elektronik.
2. Entalpi.
3. Energi bebas Gibbs.
4. Stoikiometri.
5. Efek pelarut.

Dasar teori harus menggunakan referensi akademik yang valid, seperti buku kimia fisik, kimia kuantum, kimia komputasi, artikel ilmiah, dokumentasi software resmi, atau sumber spektroskopi bereputasi.

12. Cara Menulis Dasar Teori yang Baik

Dasar teori yang baik tidak hanya menyalin definisi. Dasar teori harus menghubungkan konsep dengan proyek.

Contoh kurang baik:

HOMO adalah orbital terisi tertinggi. LUMO adalah orbital kosong terendah.

Contoh lebih baik:

HOMO merupakan orbital molekul terisi tertinggi yang secara kualitatif berkaitan dengan kemampuan molekul mendonorkan elektron, sedangkan LUMO merupakan orbital kosong terendah yang berkaitan dengan kemampuan menerima elektron. Dalam proyek ini, energi HOMO, LUMO, dan gap HOMO-LUMO digunakan untuk membandingkan pengaruh substituen amino dan nitro terhadap sifat elektronik senyawa aromatik.

Contoh kurang baik:

IR adalah spektrum inframerah.

Contoh lebih baik:

Spektrum IR muncul akibat interaksi radiasi inframerah dengan perubahan momen dipol selama vibrasi molekul. Oleh karena itu, mode vibrasi yang menyebabkan perubahan momen dipol signifikan akan memiliki intensitas IR lebih kuat. Konsep

ini digunakan untuk menafsirkan puncak utama pada spektrum IR komputasi formaldehida.

Dasar teori harus langsung mendukung pembahasan.

13. Bagian Metode Komputasi

Metode komputasi adalah bagian yang menjelaskan bagaimana perhitungan dilakukan. Bagian ini harus rinci karena menentukan reproduibilitas laporan.

Metode komputasi harus memuat:

1. Molekul atau sistem yang dihitung.
2. Software yang digunakan.
3. Metode komputasi.
4. Basis set.
5. Jenis job.
6. Charge.
7. Multiplicity.
8. Model pelarut jika ada.
9. Prosedur optimasi.
10. Prosedur frekuensi jika ada.
11. Cara membaca output.
12. Cara validasi.
13. Cara mengolah data.

Format tabel metode:

Komponen	Keterangan
Molekul target	Benzena, anilina, nitrobenzena
Software	ORCA / Gaussian / ChemCompute / GAMESS
Metode	DFT/B3LYP
Basis set	6-31G(d)
Job	Optimasi geometri
Charge	0
Multiplicity	1
Model pelarut	Fase gas / PCM air
Validasi	Konvergensi, frekuensi imajiner, geometri
Pengolahan data	Python, CSV, grafik

Metode harus ditulis secara jelas agar orang lain dapat mengulang perhitungan.

14. Contoh Penulisan Metode Komputasi

Contoh untuk proyek HOMO-LUMO:

Struktur benzena, anilina, dan nitrobenzena dibuat menggunakan editor molekul, kemudian dioptimasi menggunakan metode DFT dengan fungsional B3LYP dan basis set 6-31G(d). Seluruh molekul dihitung sebagai spesies netral dengan multiplicity 1. Energi HOMO dan LUMO diambil dari output hasil perhitungan. Gap HOMO-LUMO dihitung sebagai selisih antara energi LUMO dan HOMO. Data orbital kemudian diolah menggunakan Python untuk membuat diagram orbital dan grafik perbandingan gap.

Contoh untuk proyek IR:

Struktur formaldehida dioptimasi terlebih dahulu, kemudian dilakukan perhitungan frekuensi pada level teori yang sama. Frekuensi vibrasi dan intensitas IR diambil dari output. Validasi struktur dilakukan dengan memastikan tidak terdapat frekuensi imajiner. Data frekuensi dan intensitas dimasukkan ke CSV dan divisualisasikan menggunakan Python sebagai spektrum IR komputasi.

Contoh untuk proyek dimer air:

Monomer air dan dimer air dioptimasi menggunakan metode dan basis set yang sama. Energi dimerisasi dihitung menggunakan persamaan $\Delta E = E_{\text{dimer}} - 2E_{\text{monomer}}$. Struktur dimer diperiksa untuk memastikan adanya interaksi ikatan hidrogen. Data energi dikonversi dari Hartree ke kJ/mol menggunakan faktor konversi 1 Hartree = 2625,5 kJ/mol.

15. Validasi dalam Bagian Metode

Validasi harus masuk dalam metode, bukan hanya hasil. Mahasiswa harus menjelaskan bagaimana data akan dinilai.

Contoh:

Validasi dilakukan dengan memeriksa status job, konvergensi SCF, konvergensi optimasi, geometri akhir, charge, multiplicity, dan frekuensi imajiner. Struktur minimum hanya dianggap valid jika optimasi konvergen dan tidak terdapat frekuensi imajiner.

Untuk proyek yang tidak menjalankan frekuensi karena keterbatasan, harus ditulis secara jujur:

Karena keterbatasan waktu komputasi, validasi frekuensi tidak dilakukan. Oleh karena itu, hasil optimasi ditafsirkan sebagai struktur hasil optimasi pada level teori yang digunakan, tetapi belum dikonfirmasi sepenuhnya sebagai minimum lokal melalui analisis frekuensi.

Pernyataan seperti ini lebih baik daripada mengklaim validitas tanpa pemeriksaan.

16. Bagian Kode Python

Bagian kode Python menjelaskan skrip yang digunakan untuk mengolah data. Kode Python dapat dilampirkan penuh di lampiran, sedangkan bagian utama laporan cukup menjelaskan fungsi kode dan potongan penting.

Bagian kode Python harus memuat:

1. Nama file kode.
2. Fungsi kode.
3. Input kode.
4. Output kode.
5. Pustaka yang digunakan.
6. Rumus atau perhitungan yang dilakukan.
7. Grafik yang dihasilkan.

Contoh:

Kode Python yang digunakan bernama `analisis_homo_lumo.py`. Kode ini membaca file `data_orbital.csv`, menghitung gap HOMO-LUMO, membuat diagram orbital, dan menyimpan grafik dalam format PNG. Pustaka yang digunakan adalah `pandas` untuk membaca data dan `matplotlib` untuk membuat grafik.

Untuk proyek IR:

Kode Python membaca data frekuensi dan intensitas dari file `data_ir.csv`. Spektrum IR dibuat menggunakan grafik garis dan pelebaran Gaussian sederhana. Grafik disimpan sebagai `spektrum_ir.png`.

Untuk proyek energi reaksi:

Kode Python menghitung ΔE dari energi reaktan dan produk dalam Hartree, kemudian mengonversinya ke kJ/mol. Hasil perhitungan disimpan dalam CSV dan divisualisasikan sebagai diagram energi reaksi.

17. Prinsip Penulisan Kode Python dalam Laporan

Kode Python dalam laporan harus ditulis secara informatif, bukan sekadar ditempel panjang tanpa penjelasan. Jika kode terlalu panjang, letakkan di lampiran. Dalam bagian utama, cukup tuliskan potongan kode inti dan jelaskan fungsinya.

Prinsip penulisan:

1. Sebutkan nama file kode.
2. Jelaskan input.
3. Jelaskan proses.
4. Jelaskan output.

5. Jelaskan rumus yang digunakan.
6. Jangan menampilkan kode yang tidak relevan.
7. Pastikan kode yang dilampirkan sama dengan kode yang digunakan.
8. Jangan mengklaim grafik dibuat dengan Python jika sebenarnya dibuat manual.

Contoh ringkas:

Kode Python digunakan untuk menjaga konsistensi perhitungan dan visualisasi. Dengan kode ini, perubahan data pada CSV dapat langsung diperbarui menjadi grafik baru tanpa menggambar ulang secara manual.

Pernyataan ini menunjukkan fungsi reproduibilitas Python.

18. Bagian Hasil dan Pembahasan

Hasil dan pembahasan adalah bagian terpenting laporan. Bagian ini memuat data, tabel, grafik, validasi, interpretasi, perbandingan teori/literatur, dan analisis kritis.

Bagian hasil dan pembahasan harus memuat:

1. Ringkasan output.
2. Tabel data utama.
3. Tabel validasi.
4. Grafik utama.
5. Interpretasi hasil.
6. Perbandingan dengan teori atau literatur.
7. Keterbatasan hasil.

Bagian ini tidak boleh hanya menyajikan tabel tanpa analisis. Tabel menunjukkan data, tetapi pembahasan menjelaskan makna data.

Alur hasil dan pembahasan:

```
data output
→ validasi
→ tabel hasil
→ grafik
→ interpretasi kimia
→ perbandingan teori/literatur
→ keterbatasan
```

19. Tabel Hasil Utama

Tabel hasil harus disesuaikan dengan proyek.

Untuk proyek HOMO-LUMO:

Molekul	HOMO (eV)	LUMO (eV)	Gap (eV)	Momen Dipol (Debye)
Benzena	...	...	...	...
Anilina	...	...	...	...
Nitrobenzena	...	...	...	...

Untuk proyek IR:

Mode	Frekuensi Komputasi (cm ⁻¹)	Intensitas	Dugaan Vibrasi
1	...	...	Regangan C=O
2	...	...	Regangan C-H
3	...	...	Tekukan H-C-H

Untuk proyek dimer air:

Sistem	Energi Total (Hartree)	Energi Relatif (kJ/mol)	Keterangan
Monomer H ₂ O	...	0	Acuan
Dimer H ₂ O	...	...	Kompleks ikatan hidrogen

Untuk proyek efek pelarut:

Spesies	Energi Fase Gas	Energi Pelarut	Selisih
NH ₄ ⁺	...	...	...
NH ₃	...	...	...

Setiap tabel harus memiliki judul, satuan, dan keterangan.

20. Tabel Validasi Output

Tabel validasi wajib dimasukkan sebelum pembahasan mendalam. Contoh:

Kriteria Validasi	Status	Keterangan
Job selesai normal	Ya	Output berakhir normal
SCF konvergen	Ya	Energi elektronik stabil
Optimasi konvergen	Ya	Struktur memenuhi kriteria
Frekuensi imajiner	Tidak ada	Struktur minimum valid
Charge benar	Ya	Sesuai spesies
Multiplicity benar	Ya	Singlet
Geometri masuk akal	Ya	Dicek visual
Data layak dianalisis	Ya	Digunakan dalam pembahasan

Jika ada masalah:

Kriteria Validasi	Status	Keterangan
Frekuensi imajiner	Ada	Struktur belum valid sebagai minimum
Data layak dianalisis	Tidak	Perhitungan perlu diulang

Tabel validasi menunjukkan tanggung jawab ilmiah mahasiswa. Data yang tidak valid tidak boleh dipaksakan untuk mendukung kesimpulan.

21. Grafik dalam Hasil dan Pembahasan

Grafik harus dipilih sesuai proyek.

Jenis Proyek	Grafik Utama
HOMO-LUMO	Diagram orbital, grafik gap
IR	Spektrum IR
Optimasi obat	Grafik energi atau momen dipol
Efek pelarut	Grafik gas-pelarut
Konformer	Grafik energi relatif
Simetri-polaritas	Grafik momen dipol
Dimer air	Diagram energi dimerisasi
Energi reaksi	Diagram energi reaksi

Setiap grafik harus diberi nomor dan keterangan.

Contoh:

Gambar 1. Diagram HOMO-LUMO benzena, anilina, dan nitrobenzena berdasarkan hasil perhitungan komputasi.

atau:

Gambar 2. Spektrum IR komputasi formaldehida dengan pelebaran Gaussian sederhana.

Grafik harus dijelaskan dalam teks. Jangan hanya menampilkan gambar.

22. Cara Menulis Pembahasan yang Argumentatif

Pembahasan yang baik harus menjawab “mengapa”. Bukan hanya “apa”.

Contoh kurang baik:

Nilai gap nitrobenzena lebih kecil daripada benzena.

Contoh lebih baik:

Nilai gap HOMO-LUMO nitrobenzena lebih kecil dibanding benzena. Hal ini dapat dikaitkan dengan keberadaan gugus nitro sebagai substituen penarik elektron yang memengaruhi energi orbital kosong. Penurunan LUMO membuat selisih energi antara HOMO dan LUMO berubah. Namun, interpretasi ini bersifat kualitatif dan perlu mempertimbangkan metode serta basis set yang digunakan.

Contoh kurang baik:

Puncak IR muncul pada 1700 cm^{-1} .

Contoh lebih baik:

Puncak kuat pada sekitar wilayah karbonil dapat dikaitkan dengan regangan $\text{C}=\text{O}$. Intensitas yang besar menunjukkan bahwa mode vibrasi tersebut menyebabkan perubahan momen dipol yang signifikan. Perbedaan nilai komputasi terhadap wilayah literatur dapat disebabkan oleh pendekatan harmonik dan belum digunakannya faktor skala.

Pembahasan harus menghubungkan data, teori, dan keterbatasan.

23. Perbandingan dengan Literatur

Laporan proyek mini harus membandingkan hasil dengan teori atau literatur yang valid. Perbandingan tidak harus selalu berupa angka identik. Dalam proyek pendidikan, perbandingan tren sering lebih realistis.

Contoh perbandingan literatur:

1. Wilayah IR gugus fungsi.
2. Bentuk molekul berdasarkan VSEPR atau data struktur umum.
3. Polaritas H_2O dan CO_2 .
4. Kestabilan trans-2-butena dibanding cis-2-butena.
5. Konformer anti butana lebih stabil dibanding gauche.
6. Dimer air distabilkan oleh ikatan hidrogen.
7. Ion lebih stabil dalam pelarut polar dibanding fase gas.
8. HOMO-LUMO sebagai indikator kualitatif sifat elektronik.

Sumber yang digunakan harus valid, misalnya buku kimia fisik, kimia kuantum, kimia komputasi, spektroskopi, artikel jurnal, dokumentasi software resmi, atau database akademik. Hindari sumber tidak jelas, blog tanpa rujukan, atau artikel populer yang tidak memiliki dasar ilmiah.

Contoh penulisan:

Hasil komputasi menunjukkan puncak kuat pada wilayah karbonil, sesuai dengan literatur spektroskopi yang menyatakan bahwa regangan C=O berada pada wilayah bilangan gelombang tinggi. Namun, nilai komputasi tidak harus sama persis dengan eksperimen karena perhitungan menggunakan pendekatan harmonik.

24. Keterbatasan Hasil

Setiap laporan proyek mini harus menyebut keterbatasan. Keterbatasan bukan kelemahan laporan, tetapi tanda kejujuran ilmiah.

Contoh keterbatasan:

1. Metode dan basis set sederhana.
2. Tidak menggunakan faktor skala frekuensi.
3. Tidak menghitung semua konformer.
4. Tidak menggunakan pelarut eksplisit.
5. Tidak menghitung koreksi dispersi.
6. Tidak melakukan koreksi basis set superposition error.
7. Hanya membandingkan tren, bukan angka presisi tinggi.
8. Tidak menghitung keadaan transisi.
9. Tidak membahas dinamika molekul.
10. Tidak membandingkan banyak metode.

Contoh penulisan:

Hasil proyek ini ditafsirkan secara kualitatif karena perhitungan hanya menggunakan satu level teori dan satu basis set. Selain itu, efek pelarut diperlakukan dengan model kontinu sehingga interaksi spesifik antara molekul pelarut dan zat terlarut tidak tergambarkan secara eksplisit.

Keterbatasan membuat kesimpulan lebih hati-hati dan ilmiah.

25. Bagian Kesimpulan

Kesimpulan harus menjawab rumusan masalah dan tujuan. Kesimpulan tidak boleh berisi data baru. Kesimpulan harus singkat tetapi jelas.

Format kesimpulan:

1. Hasil utama proyek adalah ...
2. Data validasi menunjukkan bahwa ...
3. Interpretasi kimia menunjukkan bahwa ...
4. Keterbatasan utama proyek adalah ...

Contoh kesimpulan untuk proyek HOMO-LUMO:

Berdasarkan hasil perhitungan, substituen pada cincin aromatik memengaruhi energi HOMO, LUMO, dan gap HOMO-LUMO. Anilina dan nitrobenzena menunjukkan pola energi orbital yang berbeda dibanding benzena karena perbedaan karakter substituen. Hasil ini menunjukkan bahwa analisis HOMO-LUMO dapat digunakan untuk membandingkan sifat elektronik senyawa aromatik secara kualitatif. Namun, interpretasi reaktivitas tidak boleh hanya didasarkan pada gap HOMO-LUMO.

Contoh kesimpulan untuk proyek IR:

Spektrum IR komputasi formaldehida menunjukkan puncak yang sesuai dengan mode vibrasi gugus karbonil dan C-H. Tidak adanya frekuensi imajiner menunjukkan bahwa struktur hasil optimasi dapat dianggap sebagai minimum lokal. Perbedaan nilai frekuensi terhadap literatur dapat dijelaskan oleh pendekatan harmonik dan level teori yang digunakan.

Kesimpulan harus konsisten dengan pembahasan.

26. Bagian Daftar Pustaka dalam Laporan Proyek

Bagian daftar pustaka dalam laporan proyek mini memuat sumber yang digunakan dalam pendahuluan, dasar teori, metode, dan pembahasan. Daftar pustaka tidak boleh diisi sembarangan. Semua sumber harus relevan, valid, dan dapat dipertanggungjawabkan.

Sumber yang dapat digunakan:

1. Buku kimia fisik.
2. Buku kimia kuantum.
3. Buku kimia komputasi.
4. Buku spektroskopi.
5. Artikel jurnal ilmiah.
6. Dokumentasi resmi software.
7. Database akademik atau sumber data ilmiah.
8. Panduan resmi pustaka Python jika membahas kode.

Contoh sumber yang lazim dan bereputasi dalam bidang ini meliputi karya Atkins dan de Paula untuk kimia fisik, Levine untuk kimia kuantum, Jensen dan Cramer untuk kimia komputasi, Leach dan Lewars untuk pemodelan molekul, Foresman-Frisch untuk praktik Gaussian, serta dokumentasi resmi ORCA, GAMESS, Psi4, PySCF, Python, NumPy, pandas, dan Matplotlib jika relevan.

Namun, dalam laporan proyek mini, daftar pustaka harus disusun sesuai sumber yang benar-benar digunakan. Jangan memasukkan sumber yang tidak dibaca atau tidak dirujuk.

Prinsip daftar pustaka:

sumber digunakan

- dirujuk dalam teks
- dicantumkan dalam daftar pustaka

Jangan mencantumkan sumber hanya agar laporan tampak panjang.

27. Cara Menggunakan Referensi dalam Laporan

Referensi digunakan untuk mendukung teori dan pembahasan. Referensi tidak boleh menggantikan analisis mahasiswa. Mahasiswa tetap harus menjelaskan hasil dengan kata-kata sendiri.

Contoh penggunaan referensi yang baik:

Konsep HOMO dan LUMO digunakan untuk menafsirkan sifat elektronik molekul secara kualitatif. Berdasarkan konsep orbital molekul dalam kimia kuantum, HOMO berkaitan dengan orbital terisi tertinggi, sedangkan LUMO berkaitan dengan orbital kosong terendah. Dalam proyek ini, selisih energi HOMO-LUMO digunakan sebagai indikator perbandingan sifat elektronik senyawa aromatik.

Contoh yang kurang baik:

Menurut buku, HOMO dan LUMO penting. Maka hasil saya benar.

Referensi harus memperkuat analisis, bukan menjadi pengganti analisis.

28. Bagian Lampiran Input-Output

Lampiran input-output adalah bagian yang memuat bukti teknis proyek. Lampiran sangat penting dalam kimia komputasi karena hasil harus dapat ditelusuri.

Lampiran dapat memuat:

1. File input.
2. Potongan output penting.
3. Koordinat XYZ.
4. Tabel CSV.
5. Kode Python.
6. Grafik tambahan.
7. Struktur molekul.
8. Pesan status job.
9. Bagian energi output.
10. Bagian frekuensi output.
11. Bagian HOMO-LUMO output.

12. Tabel validasi lengkap.

Lampiran tidak harus memuat seluruh output jika output sangat panjang. Cukup lampirkan potongan penting dan sebutkan bahwa file lengkap disimpan secara digital.

Contoh format lampiran:

Lampiran 1. Struktur awal dan koordinat XYZ benzena, anilina, dan nitrobenzena.
 Lampiran 2. Potongan output energi HOMO-LUMO.
 Lampiran 3. File CSV data orbital.
 Lampiran 4. Kode Python diagram HOMO-LUMO.
 Lampiran 5. Grafik hasil visualisasi.

Lampiran memastikan laporan dapat diperiksa.

29. Cara Memilih Potongan Output untuk Lampiran

Karena output komputasi dapat sangat panjang, mahasiswa perlu memilih potongan output yang relevan. Potongan output yang penting meliputi:

1. Bagian metode dan basis set.
2. Charge dan multiplicity.
3. Status normal termination.
4. Energi total.
5. Geometri akhir.
6. Momen dipol.
7. Frekuensi dan intensitas IR.
8. Frekuensi imajiner.
9. HOMO-LUMO.
10. Koreksi termokimia jika digunakan.

Potongan output harus diberi keterangan. Jangan menempel teks output tanpa penjelasan.

Contoh keterangan:

Lampiran 2 menunjukkan bagian output yang memuat energi total akhir setelah optimasi geometri. Energi ini digunakan sebagai dasar perhitungan energi relatif.

atau:

Lampiran 3 menunjukkan bagian output frekuensi. Tidak terdapat frekuensi negatif, sehingga struktur dianggap sebagai minimum lokal.

30. Format Lengkap Laporan Proyek Mini Siap Pakai

Berikut format lengkap laporan proyek mini yang dapat digunakan mahasiswa.

LAPORAN PROYEK MINI PRAKTIKUM KIMIA KUANTUM

Judul:
Nama:
NIM:
Kelas:
Program Studi:
Dosen/Instruktur:
Tanggal:

ABSTRAK

[Tulis ringkasan tujuan, metode, molekul target, hasil utama, dan kesimpulan.]

1. PENDAHULUAN

1.1 Latar Belakang

[Tulis alasan ilmiah pemilihan topik.]

1.2 Rumusan Masalah

1. ...
2. ...

1.3 Tujuan

1. ...
2. ...

1.4 Batasan Proyek

[Tulis ruang lingkup proyek.]

2. DASAR TEORI

[Tulis konsep yang relevan dengan topik proyek.]

3. METODE KOMPUTASI

3.1 Molekul Target

[Tulis molekul, rumus, charge, multiplicity.]

3.2 Software

[Tulis software dan fungsinya.]

3.3 Metode dan Basis Set

[Tulis metode, basis set, jenis job, model pelarut jika ada.]

3.4 Prosedur Perhitungan

[Tulis langkah komputasi.]

3.5 Validasi Output

[Tulis cara memeriksa validitas hasil.]

4. KODE PYTHON

4.1 Nama File Kode

[Tulis nama file kode.]

4.2 Fungsi Kode

[Jelaskan input, proses, output.]

4.3 Data CSV

[Jelaskan data yang dibaca atau dihasilkan.]

4.4 Grafik

[Jelaskan grafik yang dibuat.]

5. HASIL DAN PEMBAHASAN

5.1 Hasil Output

[Tampilkan tabel hasil utama.]

5.2 Validasi Data

[Tampilkan tabel validasi.]

5.3 Grafik Hasil

[Tampilkan grafik dan keterangan.]

5.4 Pembahasan

[Jelaskan hasil berdasarkan teori, data, literatur, dan keterbatasan.]

6. KESIMPULAN

1. ...
2. ...
3. ...

7. DAFTAR PUSTAKA

[Tulis sumber akademik yang benar-benar digunakan.]

8. LAMPIRAN

- Lampiran 1. File input
- Lampiran 2. Potongan output
- Lampiran 3. CSV
- Lampiran 4. Kode Python
- Lampiran 5. Grafik

Template ini dapat digunakan untuk semua jenis proyek mini dengan penyesuaian pada bagian hasil dan pembahasan.

31. Contoh Alur Laporan Proyek HOMO-LUMO

Untuk proyek HOMO-LUMO, laporan dapat disusun dengan alur berikut:

Judul:

Analisis HOMO-LUMO Benzena, Anilina, dan Nitrobenzena

Pendahuluan:

Membahas pengaruh substituen donor dan penarik elektron.

Dasar teori:

Orbital molekul, HOMO, LUMO, gap, substituen aromatik.

Metode:

Optimasi geometri dan analisis orbital.

Kode Python:

Menghitung gap dan membuat diagram orbital.

Hasil:

Tabel HOMO, LUMO, gap, momen dipol.

Pembahasan:

Menjelaskan pengaruh substituen terhadap energi orbital.

Kesimpulan:

Substituen mengubah sifat elektronik aromatik secara kualitatif.

Lampiran:

Output orbital, CSV, kode Python.

Alur ini menunjukkan keterpaduan antara teori, output, Python, dan analisis.

32. Contoh Alur Laporan Proyek IR

Untuk proyek IR, laporan dapat disusun sebagai berikut:

Judul:

Perbandingan Spektrum IR Komputasi Formaldehida dengan Literatur

Pendahuluan:

Membahas IR sebagai alat identifikasi gugus fungsi.

Dasar teori:

Vibrasi molekul, intensitas IR, gugus karbonil, pendekatan harmonik.

Metode:

Optimasi geometri dan frekuensi.

Kode Python:

Membuat spektrum IR dari frekuensi dan intensitas.

Hasil:

Tabel frekuensi dan intensitas, spektrum IR.

Pembahasan:

Menghubungkan puncak dengan mode vibrasi C=O dan C-H.

Kesimpulan:

Pola IR komputasi sesuai wilayah gugus fungsi, dengan keterbatasan metode.

Lampiran:

Output frekuensi, CSV, kode spektrum IR.

33. Contoh Alur Laporan Proyek Dimer Air

Untuk proyek dimer air, laporan dapat disusun sebagai berikut:

Judul:

Analisis Energi Ikatan Hidrogen pada Dimer Air

Pendahuluan:

Membahas ikatan hidrogen sebagai interaksi antarmolekul.

Dasar teori:

Ikatan hidrogen, energi dimerisasi, energi relatif, validasi minimum.

Metode:

Optimasi monomer dan dimer dengan level teori sama.

Kode Python:

Menghitung $\Delta E = E_{\text{dimer}} - 2E_{\text{monomer}}$ dan membuat diagram energi.

Hasil:

Energi monomer, energi dimer, jarak $O \cdots H$, diagram energi.

Pembahasan:

Menjelaskan stabilisasi dimer oleh ikatan hidrogen.

Kesimpulan:

Dimer air menunjukkan stabilisasi relatif melalui interaksi ikatan hidrogen.

Lampiran:

Input monomer dan dimer, output energi, CSV, kode Python.

34. Kesalahan Umum dalam Menulis Laporan Proyek Mini

Beberapa kesalahan umum dalam laporan proyek mini adalah:

Kesalahan	Dampak
Abstrak terlalu umum	Isi proyek tidak tergambar
Pendahuluan tidak fokus	Masalah tidak jelas
Dasar teori tidak relevan	Analisis lemah
Metode tidak mencantumkan basis set	Hasil tidak reproduibel
Charge dan multiplicity tidak ditulis	Sistem tidak dapat diverifikasi
Validasi output tidak ada	Data belum layak dianalisis
Grafik tanpa satuan	Interpretasi kabur

Kode Python tidak dijelaskan	Pengolahan data tidak transparan
Pembahasan hanya mengulang tabel	Tidak argumentatif
Kesimpulan tidak menjawab rumusan masalah	Laporan tidak tuntas
Daftar pustaka tidak relevan	Landasan akademik lemah
Lampiran tidak memuat output	Hasil sulit diperiksa

Kesalahan ini harus dihindari dengan mengikuti template dan rubrik penilaian.

35. Rubrik Penilaian Laporan Proyek Mini

Laporan proyek mini dapat dinilai dengan rubrik berikut:

Aspek	Bobot Anjuran	Kriteria
Abstrak	10%	Ringkas, mencakup tujuan, metode, hasil, kesimpulan
Pendahuluan	10%	Latar belakang, rumusan masalah, tujuan jelas
Dasar teori	15%	Relevan, berbasis referensi valid, mendukung analisis
Metode komputasi	15%	Software, metode, basis set, charge, multiplicity lengkap
Kode Python	10%	Fungsi kode jelas, data dan output terhubung
Hasil dan validasi	15%	Tabel, output, validasi, satuan lengkap
Pembahasan	15%	Argumentatif, kritis, berbasis data dan teori
Kesimpulan	5%	Menjawab tujuan dan tidak berlebihan
Lampiran	5%	Input-output, CSV, grafik, kode tersedia

Rubrik ini menekankan bahwa nilai laporan tidak hanya ditentukan oleh banyaknya data, tetapi oleh kejelasan metode, validitas hasil, dan kualitas analisis.

36. Implikasi Keilmuan Laporan Proyek Mini

Laporan proyek mini memiliki implikasi keilmuan karena mengajarkan mahasiswa bahwa kerja kimia komputasi harus dapat ditelusuri dan dipertanggungjawabkan. Dalam kimia komputasi, angka tidak berdiri sendiri. Energi, frekuensi, orbital, dan momen dipol harus selalu dikaitkan dengan metode, basis set, struktur, charge, multiplicity, dan validasi.

Laporan yang baik mengubah output software menjadi pengetahuan ilmiah. Prosesnya adalah:

```

output mentah
→ data tervalidasi
→ tabel terstruktur
→ grafik informatif
→ pembahasan teoritis
→ kesimpulan ilmiah

```

Dengan menulis laporan, mahasiswa belajar bahwa hasil komputasi tidak hanya perlu dihitung, tetapi juga perlu dikomunikasikan secara benar.

37. Implikasi Pedagogis Laporan Proyek Mini

Secara pedagogis, laporan proyek mini melatih mahasiswa menggabungkan kemampuan menulis, menghitung, memvalidasi, dan menafsirkan. Mahasiswa tidak hanya belajar kimia kuantum, tetapi juga belajar literasi data, literasi komputasi, dan komunikasi ilmiah.

Keterampilan yang dilatih:

1. Menulis latar belakang akademik.
2. Merumuskan masalah.
3. Menjelaskan teori.
4. Mendokumentasikan metode.
5. Membaca output.
6. Memvalidasi data.
7. Menggunakan Python.
8. Membuat grafik.
9. Menyusun argumen.
10. Menarik kesimpulan.
11. Menyusun lampiran.
12. Menggunakan referensi valid.

Bagi calon guru kimia, keterampilan ini sangat penting karena guru tidak hanya perlu memahami konsep, tetapi juga perlu mampu membimbing peserta didik menyusun laporan ilmiah berbasis data.

38. Analisis Kritis terhadap Format Laporan

Format laporan membantu mahasiswa menyusun pekerjaan secara rapi, tetapi format tidak boleh membuat laporan menjadi mekanis. Mahasiswa tidak boleh hanya mengisi bagian-bagian laporan tanpa pemahaman. Setiap bagian harus memiliki fungsi.

Abstrak harus merangkum, bukan mengulang pendahuluan. Pendahuluan harus mengarahkan masalah, bukan membahas terlalu luas. Dasar teori harus mendukung analisis, bukan menjadi kumpulan definisi. Metode harus menjamin reproduktibilitas. Kode Python harus menjelaskan pengolahan data. Hasil harus divalidasi. Pembahasan harus argumentatif. Kesimpulan harus hati-hati. Lampiran harus mendukung keterlacakan.

Prinsip kritisnya:

format memberi struktur
tetapi kualitas laporan ditentukan oleh validitas data dan kekuatan analisis

39. Sintesis Subbab 12.3

Format laporan proyek mini merupakan kerangka penulisan ilmiah yang mengubah hasil proyek menjadi dokumen akademik yang rapi, valid, dan reproduibel. Laporan memuat abstrak, pendahuluan, dasar teori, metode komputasi, kode Python, hasil dan pembahasan, kesimpulan, daftar pustaka, serta lampiran input-output.

Alur laporan dapat diringkas sebagai berikut:

```
abstrak
→ pendahuluan
→ dasar teori
→ metode komputasi
→ kode Python
→ hasil dan validasi
→ pembahasan
→ kesimpulan
→ daftar pustaka
→ lampiran input-output
```

Dengan format ini, mahasiswa dilatih untuk menulis laporan yang tidak hanya lengkap secara administratif, tetapi juga kuat secara ilmiah. Laporan proyek mini menjadi bukti bahwa mahasiswa mampu merancang, menjalankan, memvalidasi, mengolah, menafsirkan, dan mengomunikasikan hasil praktikum kimia kuantum secara bertanggung jawab.

40. Transisi ke Subbab 12.4

Subbab 12.3 telah membahas format laporan proyek mini, mulai dari abstrak, pendahuluan, dasar teori, metode komputasi, kode Python, hasil dan pembahasan, kesimpulan, daftar pustaka, hingga lampiran input-output. Format ini menjadi panduan agar hasil proyek mini dapat ditulis secara sistematis, valid, kritis, dan reproduibel.

Subbab berikutnya akan membahas evaluasi akhir praktikum. Pada subbab 12.4, mahasiswa akan mempelajari bentuk evaluasi yang meliputi ujian teori singkat, ujian membaca output, ujian menjalankan kode Python, ujian interpretasi grafik, presentasi proyek mini, dan laporan akhir. Dengan demikian, laporan proyek mini pada subbab ini akan menjadi salah satu komponen penting dalam evaluasi akhir Praktikum Kimia Kuantum.

12.4 Evaluasi Akhir Praktikum

Evaluasi akhir praktikum adalah proses penilaian menyeluruh terhadap capaian belajar mahasiswa setelah menyelesaikan rangkaian Praktikum Kimia Kuantum. Evaluasi ini tidak hanya mengukur

kemampuan mengingat teori, tetapi juga menilai kemampuan mahasiswa dalam membaca output komputasi, memvalidasi data, menjalankan kode Python, menafsirkan grafik, mempresentasikan proyek mini, dan menyusun laporan akhir secara ilmiah.

Dalam praktikum kimia kuantum dan kimia komputasi, evaluasi tidak cukup dilakukan dengan ujian tulis biasa. Hal ini karena kompetensi yang dibangun sepanjang praktikum bersifat multidimensional. Mahasiswa tidak hanya belajar konsep, tetapi juga belajar prosedur komputasi, keterampilan teknis, validasi output, pengolahan data, visualisasi, dan komunikasi ilmiah. Oleh karena itu, evaluasi akhir harus dirancang sebagai penilaian terpadu yang mencerminkan seluruh proses belajar.

Evaluasi akhir pada bab ini terdiri atas enam komponen utama:

1. **Ujian teori singkat.**
2. **Ujian membaca output.**
3. **Ujian menjalankan kode Python.**
4. **Ujian interpretasi grafik.**
5. **Presentasi proyek mini.**
6. **Laporan akhir.**

Keenam komponen tersebut saling melengkapi. Ujian teori menilai pemahaman konsep. Ujian membaca output menilai ketelitian terhadap data komputasi. Ujian Python menilai kemampuan otomatisasi dan pengolahan data. Ujian interpretasi grafik menilai kemampuan analisis visual. Presentasi proyek mini menilai komunikasi ilmiah. Laporan akhir menilai kemampuan dokumentasi, validasi, dan argumentasi tertulis.

Alur evaluasi akhir dapat diringkas sebagai berikut:

```
konsep
→ output
→ validasi
→ Python
→ grafik
→ presentasi
→ laporan
→ penilaian akhir
```

Dengan alur tersebut, evaluasi akhir tidak hanya menilai hasil, tetapi juga menilai proses berpikir ilmiah mahasiswa.

1. Pengertian Evaluasi Akhir Praktikum

Evaluasi akhir praktikum adalah penilaian puncak yang dilakukan setelah seluruh rangkaian praktikum selesai. Evaluasi ini bertujuan memastikan bahwa mahasiswa benar-benar menguasai kompetensi utama yang telah dilatihkan.

Dalam Praktikum Kimia Kuantum, kompetensi utama tersebut meliputi:

1. Memahami konsep dasar kimia kuantum dan kimia komputasi.
2. Menentukan metode, basis set, charge, dan multiplicity.
3. Menjalankan perhitungan sederhana.
4. Membaca file output.
5. Memvalidasi status job, SCF, optimasi, dan frekuensi.
6. Mengambil data energi, orbital, frekuensi, dan momen dipol.
7. Mengolah data menggunakan Python.
8. Membuat tabel dan grafik.
9. Menafsirkan hasil berdasarkan teori.
10. Menyusun laporan ilmiah.
11. Mempresentasikan hasil secara ringkas dan argumentatif.

Evaluasi akhir yang baik harus mampu mengukur seluruh kemampuan tersebut, bukan hanya sebagian.

2. Latar Historis Evaluasi dalam Praktikum Komputasi

Dalam tradisi laboratorium kimia, evaluasi praktikum biasanya mencakup kehadiran, keterampilan kerja, laporan, dan ujian. Pada laboratorium kimia basah, keterampilan yang dinilai dapat berupa penggunaan alat, keselamatan kerja, teknik titrasi, pengamatan perubahan warna, atau ketelitian menimbang bahan.

Dalam kimia komputasi, bentuk keterampilan berbeda. Mahasiswa tidak menimbang bahan kimia, tetapi membuat struktur molekul. Mahasiswa tidak membaca buret, tetapi membaca output. Mahasiswa tidak mengamati endapan, tetapi mengamati konvergensi, energi total, frekuensi, dan momen dipol. Mahasiswa tidak hanya membuat tabel hasil eksperimen, tetapi juga mengolah file digital, CSV, kode Python, dan grafik.

Karena itu, evaluasi praktikum kimia komputasi harus menyesuaikan karakter bidangnya. Evaluasi harus menilai kemampuan mahasiswa sebagai pengguna software, analisis output, pengolahan data, dan penulis laporan ilmiah. Prinsip ini sejalan dengan praktik modern dalam pendidikan sains komputasi, yaitu penilaian harus mencerminkan proses kerja nyata yang dilakukan dalam bidang tersebut.

Dalam konteks ini, evaluasi akhir Praktikum Kimia Kuantum dirancang sebagai evaluasi autentik. Mahasiswa tidak hanya menjawab definisi, tetapi juga diberi output untuk dibaca, kode untuk dijalankan, grafik untuk ditafsirkan, proyek untuk dipresentasikan, dan laporan untuk disusun. Dengan demikian, evaluasi lebih dekat dengan praktik nyata kimia komputasi.

3. Prinsip Evaluasi Akhir yang Baik

Evaluasi akhir harus memenuhi beberapa prinsip.

Pertama, **valid**. Evaluasi harus benar-benar mengukur kompetensi yang ingin dinilai. Jika tujuan praktikum adalah membaca output, maka evaluasi harus memuat tugas membaca output.

Kedua, **reliabel**. Penilaian harus konsisten. Mahasiswa yang memiliki kemampuan serupa seharusnya memperoleh nilai yang relatif sebanding. Untuk itu, rubrik penilaian diperlukan.

Ketiga, **autentik**. Evaluasi harus menyerupai tugas nyata dalam kimia komputasi, seperti memvalidasi output, menghitung energi relatif, membuat grafik, dan menjelaskan hasil.

Keempat, **transparan**. Mahasiswa harus mengetahui komponen penilaian dan kriteria keberhasilan.

Kelima, **terintegrasi**. Evaluasi tidak boleh terpisah dari proses praktikum. Apa yang diuji harus sesuai dengan apa yang dipelajari.

Keenam, **kritis**. Evaluasi harus menilai kemampuan mahasiswa mengenali kesalahan dan keterbatasan, bukan hanya menghasilkan jawaban angka.

Ketujuh, **reproduksibel**. Dalam tugas komputasi, data, kode, dan output harus dapat diperiksa ulang.

Prinsip ini dapat diringkas:

```

evaluasi baik
= mengukur konsep
+ mengukur keterampilan
+ mengukur validasi
+ mengukur analisis
+ mengukur komunikasi ilmiah
  
```

4. Komponen 1: Ujian Teori Singkat

Ujian teori singkat menilai pemahaman konsep dasar yang mendukung praktikum. Ujian ini tidak perlu terlalu panjang, tetapi harus mencakup konsep inti yang digunakan dalam seluruh praktikum.

Topik yang dapat diuji meliputi:

1. Fungsi gelombang.
2. Energi kuantum.
3. Hamiltonian.
4. Orbital atom dan orbital molekul.
5. Hartree-Fock.

6. DFT.
7. Basis set.
8. Optimasi geometri.
9. SCF.
10. HOMO dan LUMO.
11. Momen dipol.
12. Frekuensi vibrasi.
13. Frekuensi imajiner.
14. Energi total.
15. Entalpi dan energi bebas Gibbs.
16. Model pelarut.
17. Validasi output.
18. Reprodusibilitas data.

Ujian teori sebaiknya menilai pemahaman, bukan hafalan istilah semata.

Contoh soal teori singkat:

1. Jelaskan perbedaan antara metode Hartree-Fock dan DFT secara konseptual.
2. Mengapa basis set memengaruhi hasil perhitungan kimia komputasi?
3. Apa makna frekuensi imajiner dalam optimasi geometri?
4. Mengapa charge dan multiplicity harus ditentukan dengan benar?
5. Apa perbedaan energi elektronik dan energi bebas Gibbs?

Soal seperti ini menuntut mahasiswa menjelaskan konsep dengan bahasa sendiri.

5. Contoh Bentuk Soal Ujian Teori

Ujian teori dapat menggunakan kombinasi soal pilihan ganda beralasan, isian singkat, dan uraian pendek.

Contoh pilihan ganda beralasan:

Sebuah struktur hasil optimasi memiliki satu frekuensi bernilai negatif. Jika struktur tersebut dimaksudkan sebagai minimum, maka kesimpulan yang tepat adalah ...

- A. Struktur valid karena hanya ada satu frekuensi negatif.
- B. Struktur belum valid sebagai minimum dan perlu diperiksa ulang.
- C. Struktur pasti merupakan produk paling stabil.
- D. Frekuensi negatif selalu boleh diabaikan.

Jawaban disertai alasan singkat.

Contoh uraian pendek:

Jelaskan mengapa output yang berakhir dengan pesan normal termination belum tentu otomatis valid untuk analisis.

Contoh soal aplikasi:

Mahasiswa menghitung NH_4^+ dengan charge 0 dan multiplicity 1. Jelaskan mengapa hasil tersebut tidak valid.

Soal teori harus menghubungkan konsep dengan praktik.

6. Rubrik Ujian Teori Singkat

Rubrik penilaian ujian teori dapat mencakup:

Kriteria	Skor Tinggi	Skor Rendah
Ketepatan konsep	Menjelaskan konsep dengan benar	Definisi salah atau kabur
Keterkaitan dengan praktikum	Menghubungkan teori dengan output/praktik	Hanya hafalan tanpa konteks
Ketelitian istilah	Menggunakan istilah tepat	Mencampur software, metode, basis set
Argumentasi	Memberi alasan logis	Jawaban pendek tanpa alasan
Contoh	Memberi contoh relevan	Tidak ada contoh

Ujian teori tidak boleh hanya menguji ingatan, tetapi harus menilai pemahaman konseptual.

7. Komponen 2: Ujian Membaca Output

Ujian membaca output menilai kemampuan mahasiswa menemukan dan menafsirkan informasi penting dalam file output komputasi. Ini adalah komponen evaluasi yang sangat penting karena seluruh laporan praktikum bergantung pada kemampuan membaca output secara benar.

Mahasiswa dapat diberi potongan output yang memuat:

1. Metode.
2. Basis set.
3. Charge.
4. Multiplicity.
5. Status SCF.
6. Status optimasi.
7. Energi total.
8. Momen dipol.
9. Frekuensi.

10. HOMO-LUMO.
11. Pesan normal termination.
12. Pesan error.
13. Frekuensi imajiner.
14. Data termokimia.

Tugas mahasiswa adalah mengidentifikasi informasi tersebut dan menyimpulkan apakah output layak digunakan.

Contoh instruksi:

Bacalah potongan output berikut. Tentukan metode, basis set, charge, multiplicity, energi total, status job, jumlah frekuensi imajiner, dan apakah data layak dianalisis.

Ujian ini melatih ketelitian dan kemampuan validasi.

8. Contoh Soal Membaca Output

Contoh potongan output sederhana:

```
Charge = 0 Multiplicity = 1
SCF Done: E(RB3LYP) = -76.345678 A.U.
Optimization completed.
Frequencies -- 1595.2 3650.4 3756.1
Dipole moment (Debye):
Tot= 1.85
Normal termination
```

Pertanyaan:

1. Berapa charge dan multiplicity sistem?
2. Berapa energi total dalam Hartree?
3. Apakah struktur memiliki frekuensi imajiner?
4. Berapa momen dipol total?
5. Apakah output layak digunakan untuk analisis struktur minimum? Jelaskan.

Contoh output bermasalah:

```
Charge = 0 Multiplicity = 1
SCF failed to converge
Optimization stopped: Maximum number of cycles exceeded
Frequencies -- -120.4 450.8 1660.2
Error termination
```

Pertanyaan:

1. Apa masalah utama pada output tersebut?

2. Apakah energi dari output boleh digunakan untuk analisis? Jelaskan.
3. Apa tindakan perbaikan yang perlu dilakukan?

Soal seperti ini menilai kemampuan mahasiswa mengenali output valid dan tidak valid.

9. Rubrik Ujian Membaca Output

Rubrik penilaian:

Aspek	Kriteria
Identifikasi data	Mampu menemukan energi, metode, basis set, charge, multiplicity
Validasi status	Mampu mengenali job normal/error
Frekuensi	Mampu menentukan ada/tidaknya frekuensi imajiner
Interpretasi	Mampu menyimpulkan kelayakan data
Tindakan korektif	Mampu menyarankan perbaikan jika output bermasalah
Ketelitian	Tidak salah membaca satuan atau bagian output

Komponen ini sangat penting karena mahasiswa sering keliru mengambil angka dari output tanpa memahami statusnya.

10. Komponen 3: Ujian Menjalankan Kode Python

Ujian menjalankan kode Python menilai kemampuan mahasiswa menggunakan Python untuk mengolah data praktikum. Ujian ini tidak harus menuntut mahasiswa membuat kode sangat kompleks dari awal. Fokusnya adalah memastikan mahasiswa mampu menjalankan, memahami, dan memodifikasi kode sederhana yang relevan dengan praktikum.

Kemampuan yang dinilai:

1. Menjalankan skrip Python.
2. Membaca file CSV.
3. Menghitung energi relatif.
4. Mengonversi Hartree ke kJ/mol.
5. Menghitung gap HOMO-LUMO.
6. Membuat grafik sederhana.
7. Menyimpan output.
8. Memperbaiki error sederhana.
9. Memahami fungsi kode.
10. Menjelaskan hasil kode.

Contoh tugas:

Diberikan file `data_energi.csv` berisi energi beberapa molekul dalam Hartree. Jalankan kode Python untuk menghitung energi relatif dalam kJ/mol dan simpan hasilnya ke file CSV baru.

Atau:

Diberikan data HOMO dan LUMO dalam CSV. Buat kolom gap HOMO-LUMO dan tampilkan grafik perbandingan gap.

Ujian ini menilai literasi komputasi mahasiswa.

11. Contoh Tugas Python: Energi Relatif

Data CSV:

```
molekul,energi_Hartree
konformer_A,-154.123456
konformer_B,-154.120000
konformer_C,-154.118500
```

Tugas:

1. Baca file CSV menggunakan pandas.
2. Tentukan energi terendah.
3. Hitung energi relatif setiap konformer dalam kJ/mol.
4. Simpan hasil ke `energi_relatif.csv`.
5. Buat grafik energi relatif.

Rumus:

Energi relatif = $(E_i - E_{\text{minimum}}) \times 2625,5$

Kriteria keberhasilan:

1. Data berhasil dibaca.
2. Energi minimum dikenali.
3. Konversi satuan benar.
4. CSV baru terbentuk.
5. Grafik memiliki label dan satuan.
6. Mahasiswa dapat menjelaskan arti grafik.

12. Contoh Tugas Python: Spektrum IR

Data CSV:

```
frekuensi_cml,intensitas
520.5,10
1050.2,22
1710.8,85
2960.1,40
```

Tugas:

1. Baca data frekuensi dan intensitas.
2. Buat spektrum IR garis.
3. Balik sumbu x sesuai konvensi spektrum IR.
4. Simpan grafik sebagai spektrum_ir.png.
5. Jelaskan puncak paling kuat.

Kriteria keberhasilan:

1. File terbaca.
2. Grafik dibuat.
3. Sumbu x ditulis sebagai bilangan gelombang.
4. Sumbu y ditulis sebagai intensitas.
5. Grafik disimpan.
6. Mahasiswa mampu menghubungkan puncak dengan mode vibrasi.

13. Rubrik Ujian Python

Rubrik penilaian:

Aspek	Kriteria
Eksekusi kode	Kode dapat dijalankan tanpa error utama
Pembacaan data	CSV atau file input terbaca benar
Perhitungan	Rumus dan konversi satuan benar
Output	CSV/grafik berhasil dibuat
Interpretasi	Mahasiswa dapat menjelaskan hasil
Debugging	Mampu memperbaiki error sederhana
Kerapian kode	Nama variabel, komentar, dan struktur cukup jelas
Reprodusibilitas	File input-output tersimpan dengan baik

Ujian Python tidak bertujuan menjadikan mahasiswa programmer profesional, tetapi memastikan mahasiswa mampu menggunakan Python sebagai alat ilmiah.

14. Komponen 4: Ujian Interpretasi Grafik

Ujian interpretasi grafik menilai kemampuan mahasiswa membaca dan menjelaskan grafik hasil praktikum. Grafik dapat berupa diagram energi, diagram HOMO-LUMO, spektrum IR, grafik momen dipol, grafik perbandingan metode, atau grafik efek pelarut.

Kemampuan yang dinilai:

1. Membaca sumbu dan satuan.
2. Menentukan tren.
3. Menghubungkan grafik dengan teori.
4. Menjelaskan makna perbedaan data.
5. Mengenali grafik yang menyesatkan.
6. Menyebut keterbatasan interpretasi.
7. Menarik kesimpulan yang sesuai data.

Contoh tugas:

Diberikan grafik energi relatif tiga konformer butana. Tentukan konformer paling stabil, jelaskan alasan kimianya, dan sebutkan keterbatasan interpretasi.

Contoh lain:

Diberikan diagram HOMO-LUMO beberapa senyawa aromatik. Tentukan molekul dengan gap paling kecil dan jelaskan makna kualitatifnya.

15. Contoh Soal Interpretasi Grafik

Contoh soal spektrum IR:

Grafik spektrum IR formaldehida menunjukkan puncak kuat pada wilayah sekitar 1700 cm^{-1} dan beberapa puncak pada wilayah sekitar $2800\text{--}3100\text{ cm}^{-1}$. Jelaskan kemungkinan mode vibrasi yang berkaitan dengan puncak tersebut dan mengapa nilai komputasi dapat berbeda dari literatur eksperimen.

Contoh soal diagram energi reaksi:

Diagram energi menunjukkan produk berada 18 kJ/mol lebih rendah daripada reaktan. Apa makna termodinamik dari hasil tersebut? Apakah grafik ini dapat digunakan untuk menyimpulkan bahwa reaksi berlangsung cepat? Jelaskan.

Jawaban yang diharapkan:

1. Produk lebih stabil secara termodinamika.
2. ΔG atau ΔE negatif, bergantung data yang digunakan.
3. Grafik reaktan-produk tidak menunjukkan energi aktivasi.
4. Laju reaksi tidak dapat disimpulkan tanpa data keadaan transisi atau mekanisme.

16. Rubrik Interpretasi Grafik

Rubrik penilaian:

Aspek	Kriteria
Membaca grafik	Mampu membaca sumbu, satuan, dan nilai utama
Menemukan tren	Mampu menentukan pola atau perbedaan utama
Penjelasan kimia	Menghubungkan grafik dengan teori
Ketepatan kesimpulan	Tidak membuat klaim berlebihan
Keterbatasan	Menyebut batas interpretasi grafik
Bahasa ilmiah	Menjelaskan secara jelas dan sistematis

Komponen ini penting karena grafik sering menjadi pusat pembahasan laporan.

17. Komponen 5: Presentasi Proyek Mini

Presentasi proyek mini menilai kemampuan mahasiswa menyampaikan hasil proyek secara lisan. Presentasi bukan sekadar membaca laporan. Mahasiswa harus mampu menjelaskan masalah, metode, hasil, validasi, grafik, kesimpulan, dan keterbatasan secara ringkas.

Struktur presentasi:

1. Judul dan identitas.
2. Latar belakang singkat.
3. Rumusan masalah.
4. Molekul target.
5. Metode komputasi.
6. Validasi output.
7. Hasil utama.
8. Grafik utama.
9. Pembahasan.
10. Kesimpulan.
11. Keterbatasan.

Durasi presentasi dapat disesuaikan. Untuk proyek mini, durasi 7–10 menit cukup, diikuti 3–5 menit tanya jawab.

Presentasi harus menekankan hasil utama. Jangan menampilkan terlalu banyak tabel output yang panjang. Pilih data yang paling relevan.

18. Kriteria Presentasi yang Baik

Presentasi proyek mini yang baik memiliki ciri-ciri:

1. Topik jelas sejak awal.
2. Slide tidak terlalu padat.
3. Molekul target ditampilkan.
4. Metode ditulis ringkas.
5. Validasi disebutkan.
6. Grafik mudah dibaca.
7. Analisis dijelaskan dengan bahasa sendiri.
8. Kesimpulan menjawab rumusan masalah.
9. Keterbatasan disebutkan.
10. Mahasiswa mampu menjawab pertanyaan.

Contoh alur slide:

Slide	Isi
1	Judul dan identitas
2	Latar belakang dan rumusan masalah
3	Molekul target dan metode
4	Validasi output
5	Tabel hasil utama
6	Grafik utama
7	Pembahasan
8	Kesimpulan dan keterbatasan

Presentasi harus menampilkan data yang sudah divalidasi.

19. Rubrik Presentasi Proyek Mini

Rubrik penilaian:

Aspek	Kriteria
Kejelasan masalah	Rumusan masalah dan tujuan jelas
Ketepatan metode	Metode, basis set, software, charge, multiplicity dijelaskan
Validasi	Status output dan kelayakan data disampaikan
Penyajian data	Tabel dan grafik jelas
Analisis	Penjelasan kimia kuat dan tidak hanya membaca angka
Kesimpulan	Menjawab tujuan
Keterbatasan	Disampaikan secara jujur
Komunikasi	Suara, bahasa, alur, dan waktu baik
Tanya jawab	Mampu menjawab secara logis

Presentasi menilai kemampuan komunikasi ilmiah, bukan hanya isi laporan.

20. Komponen 6: Laporan Akhir

Laporan akhir adalah dokumen lengkap proyek mini. Laporan ini telah dibahas pada subbab 12.3. Dalam evaluasi akhir, laporan akhir menjadi komponen penting karena menunjukkan kemampuan mahasiswa menyusun seluruh proses proyek secara tertulis.

Laporan akhir harus memuat:

1. Abstrak.
2. Pendahuluan.
3. Dasar teori.
4. Metode komputasi.
5. Kode Python.
6. Hasil dan pembahasan.
7. Kesimpulan.
8. Daftar pustaka.
9. Lampiran input-output.

Laporan akhir harus menunjukkan:

1. Metode jelas.
2. Data valid.
3. Tabel bersatuan.
4. Grafik informatif.
5. Pembahasan argumentatif.
6. Kesimpulan berbasis data.
7. Lampiran lengkap.
8. Referensi valid.

Laporan akhir adalah bukti tertulis bahwa mahasiswa telah menjalankan proyek secara ilmiah.

21. Rubrik Laporan Akhir

Rubrik laporan akhir dapat menggunakan komponen berikut:

Aspek	Bobot Anjuran	Kriteria
Abstrak	10%	Ringkas, memuat tujuan, metode, hasil, kesimpulan
Pendahuluan	10%	Latar belakang, masalah, tujuan jelas
Dasar teori	10%	Relevan dan mendukung analisis
Metode komputasi	15%	Software, metode, basis set, charge, multiplicity lengkap
Kode Python	10%	Fungsi kode, input-output, grafik jelas
Validasi data	15%	Output, konvergensi, frekuensi, geometri diperiksa

Hasil pembahasan	dan	20%	Data, grafik, analisis, keterbatasan kuat
Kesimpulan		5%	Menjawab tujuan
Lampiran pustaka	dan	5%	Input-output, CSV, kode, referensi lengkap

Bobot dapat disesuaikan, tetapi validasi dan pembahasan sebaiknya diberi bobot besar karena keduanya menunjukkan kematangan ilmiah.

22. Rancangan Bobot Evaluasi Akhir Praktikum

Evaluasi akhir dapat menggunakan bobot berikut:

Komponen Evaluasi	Bobot Anjuran
Ujian teori singkat	15%
Ujian membaca output	20%
Ujian menjalankan kode Python	15%
Ujian interpretasi grafik	15%
Presentasi proyek mini	15%
Laporan akhir	20%
Total	100%

Bobot ini menempatkan membaca output dan laporan akhir sebagai komponen besar karena keduanya sangat penting dalam kimia komputasi. Namun, bobot dapat disesuaikan oleh dosen sesuai konteks kelas.

Jika praktikum lebih menekankan pemrograman, bobot Python dapat dinaikkan. Jika praktikum lebih menekankan komunikasi ilmiah, bobot presentasi dapat dinaikkan. Jika praktikum lebih konseptual, bobot teori dapat dinaikkan.

23. Evaluasi Formatif dan Sumatif

Evaluasi akhir bersifat sumatif karena dilakukan pada akhir praktikum. Namun, agar mahasiswa berhasil, proses evaluasi sebaiknya didahului evaluasi formatif selama praktikum.

Evaluasi formatif dapat berupa:

1. Kuis singkat setiap praktikum.
2. Pemeriksaan file input.
3. Pemeriksaan output sementara.
4. Umpan balik terhadap grafik.
5. Umpan balik terhadap proposal.

6. Umpan balik terhadap draf laporan.
7. Diskusi kesalahan umum.
8. Latihan membaca output.
9. Latihan menjalankan kode Python.

Evaluasi formatif membantu mahasiswa memperbaiki kesalahan sebelum evaluasi akhir. Tanpa evaluasi formatif, evaluasi akhir dapat menjadi terlalu berat.

Alur yang baik:

```

latihan
→ umpan balik
→ perbaikan
→ proyek mini
→ evaluasi akhir

```

Dengan alur ini, evaluasi akhir menjadi puncak pembelajaran, bukan kejutan.

24. Analisis Kritis terhadap Evaluasi Akhir

Evaluasi akhir harus dirancang hati-hati agar tidak hanya menilai hasil akhir, tetapi juga proses. Dalam kimia komputasi, mahasiswa dapat memperoleh angka yang benar secara kebetulan, tetapi tidak memahami maknanya. Sebaliknya, mahasiswa dapat mengalami error tetapi menunjukkan kemampuan diagnostik yang baik. Oleh karena itu, evaluasi harus menilai pemahaman dan proses berpikir.

Contoh:

Mahasiswa A memperoleh grafik yang rapi tetapi tidak memvalidasi output.
 Mahasiswa B mengalami SCF gagal, menjelaskan penyebabnya, memperbaiki input, dan menjalankan ulang perhitungan.

Dalam evaluasi ilmiah, mahasiswa B menunjukkan keterampilan yang sangat penting. Karena itu, rubrik harus memberi nilai pada kemampuan mendiagnosis dan memperbaiki kesalahan, bukan hanya pada hasil yang tampak bagus.

Evaluasi juga harus menghindari penilaian yang terlalu bergantung pada kemampuan teknis komputer. Jika mahasiswa mengalami kendala instalasi atau perangkat, dosen perlu membedakan antara kendala teknis lingkungan dan ketidakpahaman konsep. Solusinya adalah menyediakan data output alternatif, platform web, atau file contoh untuk evaluasi.

25. Evaluasi Kejujuran Akademik

Dalam proyek komputasi, kejujuran akademik sangat penting. Mahasiswa harus menggunakan data yang benar-benar berasal dari output perhitungan, bukan angka yang dibuat atau disalin tanpa sumber. Kode Python yang digunakan juga harus sesuai dengan data yang dilaporkan.

Hal yang harus diperhatikan:

1. File input harus sesuai dengan molekul yang dilaporkan.
2. File output harus sesuai dengan metode yang ditulis.
3. Energi yang dilaporkan harus dapat ditemukan di output.
4. Grafik harus berasal dari data CSV yang sesuai.
5. Kode Python harus benar-benar digunakan.
6. Referensi harus dicantumkan jika data literatur digunakan.
7. Kesalahan tidak boleh disembunyikan.
8. Data tidak boleh dimanipulasi agar sesuai harapan.

Kejujuran akademik dapat diperkuat dengan meminta mahasiswa melampirkan:

```
file input
file output
CSV
kode Python
grafik
catatan validasi
```

Lampiran ini membuat laporan lebih mudah diperiksa.

26. Evaluasi Reprodusibilitas

Reprodusibilitas adalah kemampuan orang lain untuk mengulang perhitungan atau memeriksa kembali hasil. Dalam evaluasi akhir, reprodusibilitas harus menjadi salah satu kriteria.

Pertanyaan reprodusibilitas:

```
Apakah metode ditulis jelas?
Apakah basis set dicantumkan?
Apakah charge dan multiplicity dicatat?
Apakah file input tersedia?
Apakah output tersedia?
Apakah kode Python tersedia?
Apakah CSV tersedia?
Apakah grafik dapat dibuat ulang?
```

Jika jawabannya banyak “tidak”, laporan belum reprodusibel.

Rubrik reprodusibilitas:

Kriteria	Status
Metode dan basis set lengkap	Ya/Tidak
Charge dan multiplicity lengkap	Ya/Tidak
File input tersedia	Ya/Tidak
File output tersedia	Ya/Tidak
CSV tersedia	Ya/Tidak
Kode Python tersedia	Ya/Tidak
Grafik dapat dibuat ulang	Ya/Tidak

Reprodusibilitas penting karena kimia komputasi berbasis data digital. Data digital harus dapat ditelusuri.

27. Evaluasi Keterampilan Diagnostik

Keterampilan diagnostik adalah kemampuan mengenali masalah dan memperbaikinya. Dalam praktikum kimia komputasi, keterampilan ini sangat penting karena error sering terjadi.

Masalah yang dapat diuji:

1. Struktur awal buruk.
2. Charge salah.
3. Multiplicity salah.
4. Basis set tidak tersedia.
5. SCF tidak konvergen.
6. Optimasi gagal.
7. Frekuensi imajiner muncul.
8. Satuan energi salah.
9. Grafik salah label.
10. Python gagal membaca CSV.

Contoh soal diagnostik:

Sebuah output menunjukkan pesan SCF failed to converge. Sebutkan tiga kemungkinan penyebab dan dua langkah perbaikan yang dapat dilakukan.

Contoh jawaban yang baik:

Penyebab dapat berupa struktur awal buruk, charge atau multiplicity salah, atau sistem elektronik sulit. Perbaikan dapat dilakukan dengan memperbaiki struktur awal, menjalankan optimasi awal dengan metode lebih sederhana, menambah siklus SCF, atau memeriksa kembali charge dan multiplicity.

Evaluasi keterampilan diagnostik menunjukkan apakah mahasiswa benar-benar memahami proses komputasi.

28. Evaluasi Kemampuan Menarik Kesimpulan

Kesimpulan adalah bagian penting dalam evaluasi. Mahasiswa harus mampu menarik kesimpulan yang sesuai data. Kesimpulan tidak boleh terlalu luas, terlalu pasti, atau tidak didukung hasil.

Contoh kesimpulan berlebihan:

Molekul ini pasti paling reaktif dalam semua kondisi.

Kesimpulan tersebut terlalu kuat jika hanya berdasarkan gap HOMO-LUMO.

Kesimpulan yang lebih tepat:

Berdasarkan gap HOMO-LUMO pada level teori yang digunakan, molekul ini menunjukkan kecenderungan sifat elektronik yang berbeda dibanding pembanding. Namun, reaktivitas sebenarnya masih dipengaruhi oleh faktor struktur, lingkungan, dan mekanisme reaksi.

Evaluasi harus memberi nilai lebih pada kesimpulan yang hati-hati dan berbasis bukti.

29. Contoh Paket Evaluasi Akhir

Berikut contoh paket evaluasi akhir yang dapat digunakan:

Bagian A: Ujian teori singkat
- 5 soal uraian pendek.

Bagian B: Membaca output
- 1 potongan output valid.
- 1 potongan output bermasalah.

Bagian C: Python
- Baca CSV energi.
- Hitung energi relatif.
- Buat grafik.

Bagian D: Interpretasi grafik
- Jelaskan diagram HOMO-LUMO atau spektrum IR.

Bagian E: Presentasi proyek mini
- Presentasi 7-10 menit.
- Tanya jawab 3-5 menit.

Bagian F: Laporan akhir
- Dikumpulkan bersama file input, output, CSV, kode, dan grafik.

Paket ini mencakup teori, praktik, analisis, dan komunikasi.

30. Contoh Instruksi Evaluasi untuk Mahasiswa

Instruksi evaluasi dapat ditulis sebagai berikut:

Mahasiswa wajib mengikuti evaluasi akhir Praktikum Kimia Kuantum yang terdiri atas ujian teori singkat, ujian membaca output, ujian menjalankan kode Python, ujian interpretasi grafik, presentasi proyek mini, dan pengumpulan laporan akhir. Semua data yang digunakan dalam laporan harus berasal dari output yang telah divalidasi. Mahasiswa wajib melampirkan file input, file output, CSV, grafik, dan kode Python. Kesimpulan harus ditulis berdasarkan data dan tidak boleh melebihi bukti yang tersedia.

Instruksi seperti ini membantu mahasiswa memahami ekspektasi evaluasi.

31. Implikasi Keilmuan Evaluasi Akhir

Evaluasi akhir memiliki implikasi keilmuan karena menilai apakah mahasiswa telah memahami kimia komputasi sebagai proses ilmiah yang utuh. Mahasiswa dinilai bukan hanya karena mampu memperoleh angka, tetapi karena mampu menilai validitas angka tersebut.

Evaluasi akhir menegaskan bahwa:

angka tanpa validasi

→ belum menjadi data ilmiah

grafik tanpa analisis

→ belum menjadi argumen ilmiah

laporan tanpa metode

→ tidak reproduibel

kesimpulan tanpa bukti

→ tidak sah

Dengan demikian, evaluasi akhir menjaga agar praktikum kimia komputasi tetap berada dalam kerangka sains, bukan sekadar penggunaan software.

32. Implikasi Pedagogis Evaluasi Akhir

Secara pedagogis, evaluasi akhir membantu mahasiswa menyadari bahwa pembelajaran kimia kuantum tidak berhenti pada teori. Mahasiswa harus mampu menerapkan konsep dalam kerja nyata: membaca output, mengolah data, membuat grafik, menjelaskan hasil, dan menyusun laporan.

Evaluasi akhir juga membantu dosen menilai bagian mana yang perlu diperbaiki dalam pembelajaran. Jika banyak mahasiswa gagal membaca output, maka latihan output perlu ditambah. Jika banyak mahasiswa salah mengonversi satuan, maka materi satuan energi perlu diperkuat. Jika banyak mahasiswa membuat grafik tanpa analisis, maka pembelajaran visualisasi ilmiah perlu diperbaiki.

Dengan demikian, evaluasi akhir tidak hanya menilai mahasiswa, tetapi juga memberi umpan balik bagi perbaikan praktikum.

33. Hubungan Evaluasi Akhir dengan Proyek Mini

Evaluasi akhir sangat terkait dengan proyek mini. Proyek mini menyediakan konteks nyata untuk menilai kemampuan mahasiswa. Dalam proyek mini, mahasiswa harus menunjukkan bahwa mereka dapat:

1. Merancang masalah.
2. Memilih molekul.
3. Menentukan metode.
4. Menjalankan komputasi.
5. Memvalidasi output.
6. Mengolah data.
7. Membuat grafik.
8. Menyusun laporan.
9. Mempresentasikan hasil.

Karena itu, presentasi dan laporan proyek mini menjadi bagian penting dalam evaluasi akhir. Ujian teori, output, Python, dan grafik membantu memastikan bahwa mahasiswa tidak hanya menyerahkan proyek, tetapi benar-benar memahami prosesnya.

34. Sintesis Subbab 12.4

Evaluasi akhir Praktikum Kimia Kuantum dirancang untuk menilai kompetensi mahasiswa secara menyeluruh. Evaluasi terdiri atas ujian teori singkat, ujian membaca output, ujian menjalankan kode Python, ujian interpretasi grafik, presentasi proyek mini, dan laporan akhir. Keenam komponen ini mencerminkan karakter kimia komputasi sebagai bidang yang menggabungkan teori, metode numerik, validasi data, pemrograman, visualisasi, dan komunikasi ilmiah.

Alur evaluasi akhir dapat diringkas sebagai berikut:

```
teori
→ output
→ Python
→ grafik
```


- presentasi
- laporan
- penilaian akhir

Evaluasi yang baik harus valid, reliabel, autentik, transparan, terintegrasi, kritis, dan reproduibel. Dengan evaluasi seperti ini, mahasiswa tidak hanya dinilai dari hasil akhir, tetapi juga dari proses berpikir, keterampilan teknis, kemampuan validasi, dan kedalaman analisis.

35. Transisi ke Subbab 12.5

Subbab 12.4 telah membahas evaluasi akhir praktikum yang terdiri atas ujian teori singkat, ujian membaca output, ujian menjalankan kode Python, ujian interpretasi grafik, presentasi proyek mini, dan laporan akhir. Evaluasi ini menjadi alat untuk memastikan bahwa mahasiswa telah menguasai kompetensi utama Praktikum Kimia Kuantum secara konseptual, teknis, analitis, dan komunikatif.

Subbab berikutnya akan membahas glosarium kimia kuantum dan komputasi. Pada subbab 12.5, istilah-istilah utama seperti *ab initio*, basis set, ChemCompute, DFT, energi total, frekuensi imajiner, fungsi gelombang, Hartree-Fock, HOMO, LUMO, momen dipol, optimasi geometri, orbital molekul, SCF, simetri molekul, dan zero-point energy akan dijelaskan secara ringkas dan sistematis. Glosarium tersebut akan membantu mahasiswa memperkuat pemahaman istilah penting yang digunakan dalam seluruh buku praktikum ini.

12.5 Glosarium Kimia Kuantum dan Komputasi

Glosarium kimia kuantum dan komputasi adalah daftar istilah penting yang digunakan dalam Praktikum Kimia Kuantum, disertai penjelasan ringkas, sistematis, dan kontekstual. Glosarium berfungsi membantu mahasiswa memahami istilah teknis yang sering muncul dalam teori, input, output, kode Python, grafik, validasi data, laporan praktikum, dan proyek mini.

Dalam kimia kuantum dan kimia komputasi, istilah teknis tidak boleh dipahami sebagai hafalan lepas. Istilah seperti *ab initio*, basis set, DFT, Hartree-Fock, HOMO, LUMO, SCF, frekuensi imajiner, optimasi geometri, dan zero-point energy memiliki makna konseptual, matematis, dan praktis. Jika istilah tersebut dipahami secara dangkal, mahasiswa dapat salah membaca output, salah memilih metode, salah menafsirkan grafik, atau salah menarik kesimpulan.

Glosarium dalam subbab ini disusun untuk memperkuat pemahaman atas istilah utama yang digunakan dalam seluruh rangkaian praktikum. Setiap istilah dijelaskan dengan orientasi praktikum, bukan hanya definisi kamus. Artinya, setiap istilah dikaitkan dengan bagaimana istilah itu muncul dalam perhitungan, output, validasi, analisis, dan laporan.

Istilah utama yang dibahas dalam subbab ini meliputi:

1. *Ab initio*.
2. Basis set.

3. ChemCompute.
4. DFT.
5. Energi total.
6. Frekuensi imajiner.
7. Fungsi gelombang.
8. Hartree-Fock.
9. HOMO.
10. LUMO.
11. Momen dipol.
12. Optimasi geometri.
13. Orbital molekul.
14. SCF.
15. Simetri molekul.
16. *Zero-point energy*.

Glosarium ini tidak menggantikan pembahasan teori dalam bab-bab sebelumnya. Glosarium berfungsi sebagai peta istilah. Mahasiswa dapat kembali ke glosarium ketika membaca output, menulis laporan, menyusun proposal proyek mini, atau mempresentasikan hasil praktikum.

1. Fungsi Glosarium dalam Praktikum Kimia Kuantum

Glosarium memiliki beberapa fungsi penting.

Pertama, glosarium membantu menyamakan pemahaman istilah. Dalam praktikum, istilah yang sama dapat muncul di teori, software, output, dan laporan. Misalnya, istilah “energi total” dapat muncul dalam output sebagai SCF Done, FINAL SINGLE POINT ENERGY, atau Total Energy. Mahasiswa harus memahami bahwa istilah tersebut tidak sekadar label, tetapi merujuk pada besaran energi yang memiliki konteks metode tertentu.

Kedua, glosarium membantu menghindari kesalahan konsep. Misalnya, software bukan metode. ORCA, Gaussian, GAMESS, Psi4, PySCF, dan ChemCompute adalah perangkat atau platform, sedangkan HF, DFT, B3LYP, dan MP2 adalah metode atau pendekatan teori. Kesalahan membedakan software dan metode dapat membuat laporan tidak tepat.

Ketiga, glosarium membantu membaca output. Output komputasi berisi banyak istilah teknis seperti SCF, convergence, basis, frequency, dipole, HOMO, LUMO, energy, dan normal termination. Jika istilah tersebut tidak dipahami, mahasiswa akan kesulitan menentukan apakah hasil valid.

Keempat, glosarium membantu menulis laporan. Laporan yang baik menggunakan istilah secara tepat. Misalnya, “frekuensi imajiner” tidak boleh disebut sekadar “frekuensi negatif” tanpa penjelasan. “Gap HOMO-LUMO” tidak boleh langsung disamakan dengan reaktivitas mutlak. “DFT” tidak boleh ditulis sebagai software.

Kelima, glosarium membantu mahasiswa melakukan evaluasi akhir. Pada ujian teori, membaca output, interpretasi grafik, dan presentasi proyek mini, istilah-istilah ini akan muncul berulang kali.

Dengan demikian, glosarium berfungsi sebagai alat bantu konseptual dan praktis.

2. Latar Historis Istilah Kimia Kuantum dan Komputasi

Istilah dalam kimia kuantum dan komputasi berkembang dari sejarah panjang mekanika kuantum, kimia fisik, matematika terapan, dan sains komputasi. Istilah seperti fungsi gelombang, orbital, energi kuantum, dan Hamiltonian berakar pada perkembangan mekanika kuantum awal abad ke-20. Istilah Hartree-Fock berkembang dari upaya menyelesaikan persoalan banyak elektron secara aproksimatif. Istilah basis set muncul karena fungsi gelombang molekul harus direpresentasikan dalam bentuk fungsi matematika yang dapat dihitung.

Istilah DFT berkembang dari pendekatan yang menggunakan kerapatan elektron sebagai variabel utama. Istilah SCF muncul dari prosedur iteratif untuk mendapatkan medan elektron yang konsisten. Istilah optimasi geometri berkembang dari kebutuhan mencari struktur molekul dengan energi minimum pada permukaan energi potensial. Istilah frekuensi imajiner berkaitan dengan analisis kurvatur permukaan energi untuk membedakan minimum dan keadaan transisi. Istilah HOMO dan LUMO berkembang dari teori orbital molekul yang digunakan untuk menafsirkan sifat elektronik dan reaktivitas secara kualitatif.

Dalam literatur standar seperti kimia kuantum, kimia fisik, dan kimia komputasi, istilah-istilah tersebut tidak berdiri sendiri. Semuanya saling terkait. Misalnya, optimasi geometri memerlukan energi dan gradien; energi diperoleh melalui SCF; SCF bergantung pada metode dan basis set; orbital molekul muncul dari penyelesaian persamaan elektronik; HOMO-LUMO berasal dari orbital molekul; frekuensi dihitung dari turunan kedua energi terhadap koordinat inti.

Karena itu, glosarium ini tidak hanya menjelaskan arti kata, tetapi juga hubungan antarkonsep.

3. Prinsip Membaca Glosarium

Mahasiswa sebaiknya membaca glosarium dengan tiga cara.

Pertama, baca definisinya untuk memahami makna dasar.

Kedua, baca konteks praktikum untuk memahami bagaimana istilah itu digunakan dalam perhitungan.

Ketiga, baca catatan kritis untuk mengetahui kesalahan umum yang harus dihindari.

Misalnya, ketika membaca istilah “basis set”, mahasiswa tidak cukup mengetahui bahwa basis set adalah kumpulan fungsi matematika. Mahasiswa juga harus memahami bahwa basis set memengaruhi energi, geometri, frekuensi, momen dipol, dan waktu komputasi. Mahasiswa juga perlu tahu bahwa membandingkan energi dari basis set berbeda dapat menghasilkan kesimpulan yang tidak valid.

Prinsipnya:

```
istilah
→ definisi
→ teori
→ output
→ praktik
→ kesalahan umum
→ laporan
```

Dengan cara ini, glosarium menjadi alat belajar yang aktif.

Glosarium Utama

1. *Ab initio*

Ab initio berarti “dari prinsip pertama”. Dalam kimia komputasi, metode *ab initio* adalah metode yang berusaha menghitung sifat molekul berdasarkan prinsip dasar mekanika kuantum tanpa menggunakan parameter empiris utama yang disesuaikan langsung dari data eksperimen untuk sistem tertentu.

Metode *ab initio* menggunakan persamaan kuantum sebagai dasar, terutama persamaan Schrödinger elektronik, lalu menerapkan berbagai aproksimasi agar dapat dihitung. Contoh metode yang sering dikaitkan dengan pendekatan *ab initio* adalah Hartree-Fock dan metode pasca-Hartree-Fock seperti MP2, CI, CCSD, atau metode lanjutan lain.

Dalam praktikum dasar, istilah *ab initio* penting untuk membedakan metode berbasis teori kuantum dari metode semiempiris atau mekanika molekuler. Metode semiempiris juga memakai bentuk persamaan kuantum, tetapi memasukkan parameter empiris untuk mempercepat perhitungan. Mekanika molekuler lebih jauh lagi karena tidak menyelesaikan struktur elektronik secara eksplisit, melainkan menggunakan medan gaya.

Secara praktikum, metode *ab initio* memiliki beberapa ciri:

1. Berbasis mekanika kuantum.
2. Tidak bergantung langsung pada parameter empiris sistem spesifik.
3. Umumnya lebih mahal secara komputasi dibanding metode semiempiris.

4. Dapat memberikan dasar teori yang kuat.
5. Hasilnya tetap bergantung pada basis set dan aproksimasi metode.

Contoh penggunaan dalam laporan:

Perhitungan dilakukan menggunakan pendekatan *ab initio* Hartree-Fock dengan basis set tertentu untuk memperoleh energi elektronik dan geometri molekul.

Catatan kritis:

Istilah *ab initio* tidak berarti hasil pasti benar mutlak. Semua metode komputasi tetap menggunakan aproksimasi. Bahkan metode *ab initio* tetap dipengaruhi oleh basis set, korelasi elektron, kualitas geometri, dan keterbatasan numerik. Oleh karena itu, hasil *ab initio* tetap harus divalidasi dan dianalisis secara kritis.

2. Basis Set

Basis set adalah kumpulan fungsi matematika yang digunakan untuk merepresentasikan orbital atom atau orbital molekul dalam perhitungan kimia kuantum. Karena fungsi gelombang molekul yang sebenarnya sulit ditulis secara eksak, software komputasi menggunakan basis set sebagai pendekatan matematis.

Secara sederhana:

basis set
= kumpulan fungsi matematika untuk membangun orbital

Contoh basis set:

STO-3G
3-21G
6-31G
6-31G (d)
6-31G (d, p)
def2-SVP
def2-TZVP
cc-pVDZ

Basis set kecil biasanya cepat tetapi kurang fleksibel. Basis set besar lebih fleksibel tetapi membutuhkan waktu komputasi lebih lama. Fungsi polarisasi, seperti tanda (d) atau (d, p), membantu menggambarkan perubahan bentuk distribusi elektron. Fungsi difusi, seperti tanda +, penting untuk anion, keadaan tereksitasi tertentu, dan interaksi lemah.

Dalam praktikum, basis set memengaruhi:

1. Energi total.

2. Panjang ikatan.
3. Sudut ikatan.
4. Momen dipol.
5. Frekuensi vibrasi.
6. HOMO-LUMO.
7. Energi reaksi.
8. Stabilitas konformer.
9. Energi ikatan hidrogen.
10. Waktu komputasi.

Contoh penggunaan dalam laporan:

Basis set 6-31G(d) digunakan karena menyediakan fungsi polarisasi yang lebih baik dibanding basis set minimal, sehingga lebih sesuai untuk menggambarkan geometri molekul organik kecil.

Catatan kritis:

Kesalahan umum adalah membandingkan energi reaktan dan produk yang dihitung dengan basis set berbeda. Perbandingan seperti itu tidak valid karena perbedaan energi dapat berasal dari perbedaan basis set, bukan dari perbedaan kimia. Untuk perbandingan energi, metode dan basis set harus konsisten.

3. ChemCompute

ChemCompute adalah platform berbasis web yang menyediakan akses ke perhitungan kimia komputasi melalui antarmuka yang lebih mudah digunakan oleh mahasiswa. Platform semacam ini membantu pembelajaran karena mahasiswa dapat menjalankan perhitungan tanpa harus selalu menginstal software komputasi kompleks di komputer pribadi.

Dalam praktikum, ChemCompute dapat digunakan untuk:

1. Membuat atau mengunggah struktur molekul.
2. Menjalankan optimasi geometri.
3. Menjalankan perhitungan frekuensi.
4. Mengambil energi total.
5. Membaca output.
6. Menghasilkan data untuk laporan.
7. Membantu praktikum berbasis web.

Secara pedagogis, ChemCompute memudahkan mahasiswa pemula memasuki kimia komputasi. Mahasiswa dapat fokus pada konsep metode, basis set, output, validasi, dan interpretasi tanpa terlalu banyak terhambat instalasi software.

Namun, ChemCompute harus dipahami sebagai platform atau antarmuka, bukan metode teori. Metode tetap harus disebutkan secara terpisah, misalnya HF, DFT, B3LYP, atau metode lain yang tersedia.

Penulisan yang kurang tepat:

Metode yang digunakan adalah ChemCompute.

Penulisan yang benar:

Perhitungan dijalankan melalui ChemCompute menggunakan metode DFT/B3LYP dengan basis set 6-31G(d).

Catatan kritis:

Kemudahan platform web tidak menghapus kewajiban validasi. Output dari ChemCompute tetap harus diperiksa: job selesai normal atau tidak, optimasi konvergen atau tidak, frekuensi imajiner ada atau tidak, charge dan multiplicity benar atau tidak, serta geometri akhir masuk akal atau tidak.

4. DFT

DFT atau *Density Functional Theory* adalah pendekatan kimia kuantum yang menggunakan kerapatan elektron sebagai variabel utama untuk menghitung energi dan sifat molekul. Berbeda dari metode fungsi gelombang yang berfokus pada fungsi gelombang banyak elektron, DFT menggunakan kerapatan elektron yang bergantung pada tiga koordinat ruang.

DFT menjadi sangat populer dalam kimia komputasi karena memberikan keseimbangan yang baik antara ketelitian dan biaya komputasi. Untuk banyak sistem molekul, DFT lebih efisien dibanding metode pasca-Hartree-Fock dan sering lebih baik daripada Hartree-Fock murni karena memasukkan efek korelasi elektron melalui fungsional pertukaran-korelasi.

Contoh fungsional DFT:

B3LYP
PBE
PBE0
M06
M06-2X
CAM-B3LYP

Dalam praktikum, DFT sering digunakan untuk:

1. Optimasi geometri.
2. Analisis HOMO-LUMO.
3. Perhitungan frekuensi IR.
4. Perhitungan momen dipol.

5. Energi reaksi sederhana.
6. Kajian efek pelarut.
7. Studi molekul organik kecil dan sedang.

Contoh penggunaan dalam laporan:

Metode DFT dengan fungsional B3LYP digunakan untuk mengoptimasi struktur molekul dan memperoleh energi orbital HOMO-LUMO.

Catatan kritis:

DFT bukan satu metode tunggal. Hasil DFT bergantung pada fungsional yang digunakan. B3LYP, PBE, PBE0, dan M06-2X dapat menghasilkan nilai yang berbeda. Oleh karena itu, laporan harus menyebut fungsional secara jelas. Menulis “menggunakan DFT” saja belum cukup.

DFT juga memiliki keterbatasan, terutama pada interaksi lemah, keadaan tereksitasi tertentu, sistem multireferensi, dan transfer muatan jarak jauh jika fungsional tidak sesuai. Karena itu, hasil DFT harus ditafsirkan berdasarkan konteks.

5. Energi Total

Energi total adalah energi yang dihitung untuk suatu sistem molekul pada level teori tertentu. Dalam output kimia komputasi, energi total biasanya muncul dalam satuan Hartree. Energi ini dapat berupa energi elektronik, energi SCF, energi single point, atau energi dengan koreksi tertentu tergantung jenis perhitungan.

Contoh label output:

```
SCF Done
FINAL SINGLE POINT ENERGY
Total Energy
Total SCF Energy
```

Energi total penting karena menjadi dasar untuk menghitung:

1. Energi relatif.
2. Energi isomerisasi.
3. Energi konformer.
4. Energi dimerisasi.
5. Energi reaksi.
6. Perbandingan metode.
7. Efek pelarut.
8. Stabilitas relatif.

Namun, energi total absolut sering tidak langsung bermakna bagi pembahasan kimia. Yang lebih sering dianalisis adalah selisih energi.

Contoh:

$$\Delta E = E_{\text{produk}} - E_{\text{reaktan}}$$

atau:

$$\Delta E_{\text{dimerisasi}} = E_{\text{dimer}} - 2E_{\text{monomer}}$$

Konversi satuan penting:

$$1 \text{ Hartree} \approx 2625,5 \text{ kJ/mol}$$

Catatan kritis:

Kesalahan umum adalah membandingkan energi total dari metode atau basis set berbeda. Kesalahan lain adalah mengambil energi dari langkah optimasi awal, bukan energi akhir setelah konvergensi. Mahasiswa harus memastikan energi yang digunakan berasal dari bagian output yang benar.

6. Frekuensi Imajiner

Frekuensi imajiner adalah frekuensi vibrasi yang secara matematis muncul ketika struktur berada pada arah kurvatur negatif pada permukaan energi potensial. Dalam banyak output, frekuensi imajiner ditampilkan sebagai nilai negatif.

Contoh:

```
Frequencies -- -250.4 120.5 360.8
```

Angka -250.4 menunjukkan adanya frekuensi imajiner.

Dalam optimasi geometri, frekuensi imajiner digunakan untuk memvalidasi apakah struktur merupakan minimum lokal atau bukan. Untuk struktur minimum, tidak boleh ada frekuensi imajiner. Jika ada frekuensi imajiner, struktur belum valid sebagai minimum.

Aturan praktikum:

```
struktur minimum
→ 0 frekuensi imajiner
```

```
keadaan transisi
→ 1 frekuensi imajiner yang sesuai koordinat reaksi
```

Karena sebagian besar praktikum dalam buku ini berfokus pada struktur minimum, frekuensi imajiner harus dianggap sebagai tanda bahwa struktur perlu diperiksa ulang.

Contoh penggunaan dalam laporan:

Hasil perhitungan frekuensi tidak menunjukkan frekuensi imajiner, sehingga struktur hasil optimasi dapat dianggap sebagai minimum lokal.

Jika bermasalah:

Output menunjukkan satu frekuensi imajiner. Karena struktur dimaksudkan sebagai minimum, hasil belum digunakan untuk analisis energi dan perlu dilakukan optimasi ulang.

Catatan kritis:

Frekuensi imajiner tidak selalu berarti software gagal total. Maknanya tergantung tujuan perhitungan. Untuk keadaan transisi, satu frekuensi imajiner justru diharapkan. Namun, untuk struktur minimum, frekuensi imajiner menunjukkan masalah validasi.

7. Fungsi Gelombang

Fungsi gelombang adalah fungsi matematika yang memuat informasi tentang keadaan kuantum suatu sistem. Dalam mekanika kuantum, fungsi gelombang biasanya dilambangkan dengan simbol Ψ . Untuk sistem molekul, fungsi gelombang idealnya memuat informasi tentang elektron dan inti. Namun, karena persoalan banyak partikel sangat kompleks, berbagai pendekatan digunakan untuk menyelesaikannya.

Dalam kimia kuantum, fungsi gelombang berkaitan dengan:

1. Probabilitas keberadaan elektron.
2. Energi sistem.
3. Orbital.
4. Distribusi elektron.
5. Keadaan elektronik.
6. Spin.
7. Interaksi elektron-elektron.

Persamaan Schrödinger secara umum dapat ditulis:

$$\hat{H}\Psi = E\Psi$$

dengan $\hat{H}$ adalah operator Hamiltonian, Ψ adalah fungsi gelombang, dan E adalah energi.

Dalam praktikum, mahasiswa tidak selalu menghitung fungsi gelombang secara eksplisit, tetapi konsep ini menjadi dasar untuk memahami orbital molekul, Hartree-Fock, dan metode berbasis fungsi gelombang.

Contoh hubungan konsep:

fungsi gelombang
 → orbital molekul
 → energi elektronik
 → sifat molekul

Catatan kritis:

Fungsi gelombang bukan lintasan elektron seperti dalam mekanika klasik. Elektron dalam mekanika kuantum tidak digambarkan sebagai partikel yang bergerak pada orbit pasti seperti planet mengelilingi matahari. Fungsi gelombang memberikan informasi probabilistik.

8. Hartree-Fock

Hartree-Fock adalah metode kimia kuantum yang menyelesaikan persoalan banyak elektron dengan pendekatan medan rata-rata. Dalam metode ini, setiap elektron dianggap bergerak dalam medan rata-rata yang dihasilkan oleh elektron-elektron lain. Hartree-Fock menghasilkan orbital molekul dan energi elektronik melalui prosedur SCF.

Hartree-Fock penting secara historis karena menjadi dasar banyak metode kimia kuantum modern. Metode ini memperkenalkan cara sistematis untuk memperkirakan fungsi gelombang banyak elektron menggunakan determinan Slater dan prinsip variasi.

Dalam praktikum, Hartree-Fock dapat digunakan untuk:

1. Memahami konsep orbital molekul.
2. Memahami prosedur SCF.
3. Membandingkan hasil dengan DFT.
4. Menghitung energi dan geometri molekul kecil.
5. Menjelaskan keterbatasan korelasi elektron.

Keterbatasan utama Hartree-Fock adalah tidak memasukkan korelasi elektron secara penuh. Hartree-Fock memperlakukan interaksi elektron secara rata-rata, sehingga energi yang dihasilkan biasanya kurang stabil dibanding metode yang memasukkan korelasi elektron.

Contoh penggunaan dalam laporan:

Metode Hartree-Fock digunakan sebagai pendekatan dasar untuk menghitung energi elektronik dan orbital molekul. Hasilnya dibandingkan dengan metode DFT untuk melihat pengaruh perlakuan korelasi elektron.

Catatan kritis:

Hartree-Fock bukan metode “salah”. Ia adalah metode dasar yang sangat penting. Namun, untuk banyak sistem kimia, hasil Hartree-Fock dapat kurang akurat karena korelasi elektron tidak dijelaskan secara memadai. Karena itu, interpretasi hasil HF harus mempertimbangkan keterbatasan ini.

9. HOMO

HOMO adalah singkatan dari *Highest Occupied Molecular Orbital*, yaitu orbital molekul terisi tertinggi. Secara kualitatif, HOMO sering dikaitkan dengan kemampuan molekul mendonorkan elektron karena orbital ini merupakan orbital terisi dengan energi paling tinggi.

Dalam analisis kimia komputasi, energi HOMO digunakan untuk membandingkan sifat elektronik molekul. Molekul dengan HOMO lebih tinggi secara kualitatif dapat lebih mudah mendonorkan elektron dalam konteks tertentu. Namun, interpretasi ini harus dilakukan hati-hati dan tidak boleh dilepaskan dari mekanisme reaksi, lingkungan, muatan, dan struktur molekul.

Dalam praktikum, HOMO digunakan pada:

1. Analisis senyawa aromatik tersubstitusi.
2. Perbandingan sifat elektronik molekul.
3. Diagram orbital.
4. Perhitungan gap HOMO-LUMO.
5. Diskusi reaktivitas kualitatif.

Contoh penggunaan dalam laporan:

Energi HOMO anilina lebih tinggi dibanding benzena pada level teori yang digunakan, yang secara kualitatif dapat dikaitkan dengan pengaruh substituen donor elektron.

Catatan kritis:

HOMO tidak boleh digunakan sendirian untuk menyimpulkan seluruh reaktivitas molekul. Reaktivitas kimia dipengaruhi oleh banyak faktor, seperti LUMO, distribusi orbital, muatan parsial, sterik, pelarut, suhu, dan mekanisme reaksi. HOMO adalah indikator, bukan satu-satunya penentu.

10. LUMO

LUMO adalah singkatan dari *Lowest Unoccupied Molecular Orbital*, yaitu orbital molekul kosong terendah. Secara kualitatif, LUMO sering dikaitkan dengan kemampuan molekul menerima elektron karena orbital ini merupakan orbital kosong dengan energi paling rendah.

LUMO penting dalam analisis sifat elektronik, terutama ketika membahas akseptor elektron, interaksi donor-akseptor, transisi elektronik, dan kecenderungan reaktivitas tertentu.

Dalam praktikum, LUMO digunakan bersama HOMO untuk menghitung gap HOMO-LUMO:

$$\text{gap HOMO-LUMO} = E_{\text{LUMO}} - E_{\text{HOMO}}$$

Semakin kecil gap, semakin mudah secara kualitatif terjadi eksitasi atau interaksi elektronik tertentu. Namun, interpretasi ini tetap terbatas.

Contoh penggunaan dalam laporan:

Nitrobenzena memiliki LUMO yang lebih rendah dibanding benzena, yang dapat dikaitkan dengan pengaruh gugus nitro sebagai substituen penarik elektron.

Catatan kritis:

LUMO dari perhitungan orbital molekul bergantung pada metode dan basis set. Dalam beberapa metode, nilai orbital virtual tidak selalu dapat ditafsirkan sebagai energi observabel langsung. Karena itu, analisis LUMO harus bersifat kualitatif dalam praktikum dasar.

11. Momen Dipol

Momen dipol adalah besaran yang menyatakan pemisahan muatan dalam suatu molekul. Momen dipol bergantung pada polaritas ikatan dan geometri molekul. Satuan yang sering digunakan adalah Debye.

Secara konseptual:

momen dipol
= ukuran pemisahan muatan positif dan negatif dalam molekul

Molekul dengan ikatan polar belum tentu memiliki momen dipol total besar. Hal ini karena momen dipol adalah besaran vektor. Jika vektor dipol ikatan saling meniadakan karena simetri, momen dipol total dapat mendekati nol.

Contoh:

1. H₂O memiliki momen dipol tidak nol karena bentuknya bengkok.

2. CO_2 memiliki momen dipol total mendekati nol karena linear dan simetris.
3. NH_3 memiliki momen dipol tidak nol karena bentuk trigonal piramida.
4. BF_3 dapat memiliki ikatan polar tetapi momen total mendekati nol karena trigonal planar simetris.

Dalam praktikum, momen dipol digunakan untuk:

1. Menganalisis polaritas molekul.
2. Menghubungkan geometri dan simetri.
3. Membandingkan efek substituen.
4. Menjelaskan interaksi dengan pelarut.
5. Membahas sifat molekul polar dan nonpolar.

Contoh penggunaan dalam laporan:

Momen dipol CO_2 mendekati nol karena molekul memiliki geometri linear simetris sehingga vektor dipol kedua ikatan $\text{C}=\text{O}$ saling meniadakan.

Catatan kritis:

Nilai momen dipol harus ditafsirkan bersama geometri. Jika geometri salah, momen dipol juga dapat salah. Oleh karena itu, validasi struktur tetap diperlukan sebelum momen dipol dianalisis.

12. Optimasi Geometri

Optimasi geometri adalah proses mencari susunan atom yang menghasilkan energi minimum atau titik stasioner pada permukaan energi potensial. Dalam optimasi geometri, software mengubah posisi atom secara bertahap sampai kriteria konvergensi terpenuhi.

Secara praktikum:

```
struktur awal
→ optimasi
→ struktur akhir
→ validasi
```

Optimasi geometri sangat penting karena banyak sifat molekul bergantung pada struktur. Energi, momen dipol, frekuensi, orbital, dan spektrum IR dapat berubah jika geometri berubah.

Kriteria optimasi biasanya melibatkan:

1. Perubahan energi.
2. Gaya maksimum.
3. Gaya RMS.
4. Perpindahan maksimum.

5. Perpindahan RMS.
6. Jumlah langkah optimasi.
7. Status konvergensi.

Contoh pesan output:

```
Optimization completed
Geometry optimization converged
Stationary point found
```

Contoh penggunaan dalam laporan:

Struktur molekul dioptimasi terlebih dahulu sebelum analisis frekuensi agar data vibrasi diperoleh dari geometri yang telah mencapai titik stasioner.

Catatan kritis:

Optimasi yang selesai belum tentu selalu menghasilkan minimum. Untuk memastikan struktur adalah minimum, perhitungan frekuensi perlu dilakukan. Jika ada frekuensi imajiner, struktur belum valid sebagai minimum.

13. Orbital Molekul

Orbital molekul adalah fungsi matematika yang menggambarkan distribusi elektron dalam molekul. Orbital molekul terbentuk dari kombinasi orbital atom. Dalam teori orbital molekul, elektron tidak hanya dimiliki oleh satu atom, tetapi dapat terdelokalisasi pada seluruh molekul.

Orbital molekul dapat berupa:

1. Orbital ikatan.
2. Orbital antiikatan.
3. Orbital nonikatan.
4. Orbital σ .
5. Orbital π .
6. Orbital terisi.
7. Orbital virtual.

HOMO dan LUMO adalah bagian dari orbital molekul. HOMO adalah orbital molekul terisi tertinggi, sedangkan LUMO adalah orbital molekul kosong terendah.

Dalam praktikum, orbital molekul digunakan untuk:

1. Menjelaskan ikatan kimia.
2. Menganalisis HOMO-LUMO.
3. Membahas substituen aromatik.

4. Menafsirkan donor-akseptor elektron.
5. Menjelaskan sifat elektronik molekul.

Contoh penggunaan dalam laporan:

Analisis orbital molekul menunjukkan bahwa substituen pada cincin aromatik dapat mengubah distribusi orbital HOMO dan LUMO.

Catatan kritis:

Visualisasi orbital molekul bergantung pada kontur dan fase yang ditampilkan. Warna atau bentuk orbital dalam visualizer harus ditafsirkan sebagai representasi matematika, bukan objek fisik padat. Orbital adalah fungsi, bukan “awan” yang dapat dipegang.

14. SCF

SCF adalah singkatan dari *Self-Consistent Field*. SCF adalah prosedur iteratif untuk memperoleh solusi elektronik yang konsisten. Dalam perhitungan Hartree-Fock dan banyak perhitungan DFT, SCF digunakan untuk mencari distribusi elektron dan energi yang stabil.

Alur SCF sederhana:

```
tebakan awal orbital
→ hitung medan elektron
→ perbarui orbital
→ hitung energi
→ ulangi
→ konvergen
```

SCF dianggap konvergen jika perubahan energi atau kerapatan elektron sudah berada di bawah batas tertentu.

Jika SCF gagal konvergen, energi elektronik tidak dapat dipercaya. Output dapat memuat pesan seperti:

```
SCF failed to converge
SCF NOT CONVERGED
Maximum SCF cycles exceeded
```

Dalam praktikum, SCF penting karena:

1. Menentukan energi elektronik.
2. Menjadi dasar optimasi geometri.
3. Menentukan orbital HOMO-LUMO.
4. Memengaruhi momen dipol.
5. Memengaruhi frekuensi dan sifat turunan lain.

Contoh penggunaan dalam laporan:

SCF dinyatakan konvergen pada output, sehingga energi elektronik digunakan untuk analisis lanjutan.

Catatan kritis:

Jika SCF tidak konvergen, hasil tidak boleh langsung digunakan. Mahasiswa harus memperbaiki struktur awal, charge, multiplicity, metode, basis set, atau parameter SCF jika diperlukan.

15. Simetri Molekul

Simetri molekul adalah keteraturan geometri molekul terhadap operasi simetri seperti rotasi, refleksi, inversi, atau kombinasi operasi tertentu. Simetri sangat penting dalam kimia karena memengaruhi polaritas, orbital molekul, spektrum, degenerasi energi, dan seleksi transisi.

Dalam praktikum dasar, simetri paling sering digunakan untuk menjelaskan momen dipol dan polaritas molekul.

Contoh:

1. CO₂ linear dan simetris, sehingga momen dipol total mendekati nol.
2. H₂O bengkok dan tidak simetris secara penuh, sehingga memiliki momen dipol.
3. BF₃ trigonal planar simetris, sehingga momen dipol total dapat saling meniadakan.
4. NH₃ trigonal piramida, sehingga memiliki momen dipol.
5. CH₄ tetrahedral simetris, sehingga nonpolar secara total.

Simetri juga berhubungan dengan spektrum IR. Mode vibrasi IR aktif jika menyebabkan perubahan momen dipol selama vibrasi. Oleh karena itu, simetri dapat memengaruhi puncak spektrum.

Contoh penggunaan dalam laporan:

Perbedaan momen dipol H₂O dan CO₂ dapat dijelaskan berdasarkan simetri molekul. H₂O memiliki bentuk bengkok sehingga vektor dipol tidak saling meniadakan, sedangkan CO₂ linear sehingga vektor dipol kedua ikatan C=O saling meniadakan.

Catatan kritis:

Mahasiswa harus membedakan polaritas ikatan dan polaritas molekul. Molekul dengan ikatan polar dapat memiliki momen dipol total nol jika simetrinya membuat vektor dipol saling meniadakan.

16. Zero-Point Energy

Zero-point energy atau ZPE adalah energi vibrasi minimum yang tetap dimiliki molekul bahkan pada suhu 0 K. Dalam mekanika kuantum, osilator harmonik tidak dapat memiliki energi vibrasi nol sepenuhnya. Energi terendahnya tetap bernilai tertentu.

Secara konsep:

ZPE

= energi vibrasi dasar yang tidak hilang bahkan pada 0 K

Dalam perhitungan frekuensi, software dapat memberikan koreksi ZPE. Koreksi ini dapat digunakan untuk memperbaiki energi elektronik menjadi energi yang lebih sesuai untuk analisis termokimia.

Hubungan sederhana:

$E_{\text{dengan_ZPE}} = E_{\text{elektronik}} + \text{ZPE}$

Dalam praktikum, ZPE penting untuk:

1. Analisis frekuensi.
2. Koreksi energi reaksi.
3. Termodinamika.
4. Perbandingan kestabilan.
5. Perhitungan energi dimerisasi atau isomerisasi yang lebih baik.
6. Pemahaman energi vibrasi kuantum.

Contoh penggunaan dalam laporan:

Energi elektronik dikoreksi dengan zero-point energy untuk memperoleh estimasi energi yang memasukkan kontribusi vibrasi dasar molekul.

Catatan kritis:

ZPE bergantung pada frekuensi vibrasi yang dihitung. Jika struktur belum valid atau memiliki frekuensi imajiner, koreksi ZPE juga tidak dapat digunakan secara aman. Selain itu, ZPE bukan satu-satunya koreksi termokimia. Untuk menghitung entalpi atau energi bebas Gibbs, kontribusi termal dan entropi juga perlu diperhitungkan.

Ringkasan Hubungan Antaristilah

Istilah-istilah dalam glosarium ini saling terhubung. Hubungan konseptualnya dapat diringkaskan sebagai berikut:

fungsi gelombang
 → orbital molekul
 → HOMO dan LUMO
 → sifat elektronik

metode komputasi
 → Hartree-Fock / DFT / ab initio
 → SCF
 → energi total

basis set
 → representasi orbital
 → kualitas energi dan geometri

optimasi geometri
 → struktur akhir
 → frekuensi
 → validasi minimum

frekuensi
 → frekuensi imajiner / ZPE / IR

geometri + distribusi muatan
 → momen dipol
 → polaritas

simetri molekul
 → momen dipol
 → spektrum
 → interpretasi struktur

software/platform
 → ChemCompute
 → output
 → validasi
 → laporan

Ringkasan ini menunjukkan bahwa kimia komputasi bukan kumpulan istilah terpisah. Setiap istilah berada dalam jaringan konsep yang saling mendukung.

Analisis Kritis terhadap Penggunaan Istilah

Penggunaan istilah teknis harus dilakukan secara tepat. Kesalahan istilah sering mencerminkan kesalahan pemahaman. Misalnya, jika mahasiswa menulis “metode ChemCompute”, berarti mahasiswa belum membedakan platform dan metode. Jika mahasiswa menulis “DFT adalah software”, berarti mahasiswa belum memahami posisi DFT sebagai pendekatan teori. Jika mahasiswa menulis “frekuensi imajiner kecil boleh diabaikan” tanpa konteks, berarti mahasiswa belum memahami validasi minimum.

Beberapa pasangan istilah yang sering tertukar:

Istilah 1	Istilah 2	Perbedaan
Software	Metode	Software menjalankan perhitungan; metode adalah teori
Metode	Basis set	Metode adalah pendekatan; basis set adalah fungsi matematika
Energi total	Energi relatif	Energi total adalah nilai absolut; energi relatif adalah selisih
HOMO	LUMO	HOMO terisi; LUMO kosong
Frekuensi negatif	Frekuensi imajiner	Dalam output, frekuensi imajiner sering ditampilkan sebagai angka negatif
Polaritas ikatan	Polaritas molekul	Polaritas molekul bergantung pada vektor total dan simetri
Optimasi selesai	Struktur minimum valid	Minimum perlu diperiksa dengan frekuensi
Python	Kimia	Python mengolah data; interpretasi tetap berdasarkan teori kimia

Ketelitian istilah adalah bagian dari ketelitian ilmiah.

Aplikasi Glosarium dalam Laporan Praktikum

Glosarium ini dapat digunakan mahasiswa saat menulis laporan. Contoh penerapan istilah yang tepat:

Perhitungan dilakukan menggunakan metode DFT dengan fungsional B3LYP dan basis set 6-31G(d). Optimasi geometri dilakukan untuk memperoleh struktur minimum. Validasi struktur dilakukan melalui analisis frekuensi. Tidak ditemukan frekuensi imajiner, sehingga struktur dianggap sebagai minimum lokal. Energi HOMO dan LUMO digunakan untuk menghitung gap HOMO-LUMO. Momen dipol dianalisis berdasarkan geometri dan simetri molekul.

Paragraf tersebut menggunakan istilah secara tepat dan saling terhubung. Mahasiswa harus berlatih menulis seperti ini agar laporan tidak hanya berisi angka, tetapi juga konsep yang benar.

Implikasi Keilmuan Glosarium

Glosarium memiliki implikasi keilmuan karena membantu mahasiswa memahami bahasa kimia komputasi. Setiap disiplin ilmu memiliki bahasa teknis. Dalam kimia komputasi, bahasa teknis tersebut mencerminkan konsep matematika, fisika, dan kimia. Memahami istilah berarti memahami cara berpikir bidang tersebut.

Jika mahasiswa memahami istilah dengan benar, mereka lebih mampu:

1. Membaca literatur.

2. Membaca output.
3. Memilih metode.
4. Memvalidasi hasil.
5. Menulis laporan.
6. Mengkritisi hasil.
7. Berkomunikasi secara ilmiah.
8. Mengembangkan proyek mini.

Sebaliknya, jika istilah dipahami secara salah, analisis dapat keliru sejak awal.

Implikasi Pedagogis Glosarium

Secara pedagogis, glosarium membantu mahasiswa belajar secara mandiri. Mahasiswa dapat menggunakan glosarium sebagai rujukan cepat ketika mengerjakan praktikum, menyusun proposal, menulis laporan, atau mempersiapkan evaluasi akhir.

Bagi dosen atau instruktur, glosarium dapat digunakan sebagai dasar kuis istilah, latihan membaca output, diskusi konsep, atau refleksi akhir praktikum. Misalnya, mahasiswa dapat diminta menjelaskan perbedaan DFT dan Hartree-Fock, membedakan basis set dan metode, atau menjelaskan mengapa frekuensi imajiner penting untuk validasi struktur.

Glosarium juga membantu calon guru kimia. Dalam pembelajaran, guru perlu mampu menjelaskan istilah teknis dengan bahasa yang dapat dipahami peserta didik tanpa menghilangkan ketepatan ilmiah. Oleh karena itu, glosarium ini tidak hanya bermanfaat untuk mahasiswa, tetapi juga untuk pengembangan bahan ajar kimia komputasi.

Sintesis Subbab 12.5

Subbab 12.5 telah menyajikan glosarium utama kimia kuantum dan komputasi yang mencakup *ab initio*, basis set, ChemCompute, DFT, energi total, frekuensi imajiner, fungsi gelombang, Hartree-Fock, HOMO, LUMO, momen dipol, optimasi geometri, orbital molekul, SCF, simetri molekul, dan *zero-point energy*. Setiap istilah dijelaskan tidak hanya sebagai definisi, tetapi juga dalam konteks teori, praktikum, output, validasi, laporan, dan kesalahan umum.

Glosarium ini menegaskan bahwa pemahaman istilah adalah bagian dari kompetensi ilmiah. Mahasiswa yang memahami istilah secara tepat akan lebih mampu membaca output, memvalidasi data, menggunakan Python secara terarah, membuat grafik yang benar, menyusun laporan yang reproduktibel, dan mempresentasikan hasil proyek mini secara argumentatif.

Alur fungsi glosarium dapat diringkas sebagai berikut:

istilah teknis

- pemahaman konsep
- pembacaan output
- validasi data
- analisis
- laporan
- presentasi

Dengan demikian, glosarium menjadi pengikat konseptual bagi seluruh rangkaian Praktikum Kimia Kuantum.

Transisi ke Subbab 12.6

Subbab 12.5 telah membahas glosarium kimia kuantum dan komputasi sebagai alat bantu memahami istilah utama yang digunakan dalam praktikum, proyek mini, evaluasi akhir, laporan, dan analisis output. Glosarium ini memperkuat pemahaman konseptual mahasiswa sebelum memasuki bagian akhir bab.

Subbab berikutnya akan membahas lampiran kode Python. Pada subbab 12.6, akan disajikan daftar dan fungsi kode lengkap yang mendukung seluruh praktikum, mulai dari praktikum_01_partikel_dalam_kotak.py, praktikum_02_optimasi_geometri.py, praktikum_03_hartree_fock_basis_set.py, praktikum_04_dft_homo_lumo.py, praktikum_05_simetri_momen_dipol.py, praktikum_06_spektrum_ir.py, praktikum_07_energi_reaksi_termodinamika.py, parser_output_komputasi.py, hingga template_laporan_otomatis.py. Lampiran kode tersebut menjadi arsip praktikum yang dapat digunakan ulang, dimodifikasi, dan dikembangkan.

12.6 Lampiran Kode Python

Lampiran kode Python adalah bagian akhir dari Bab 12 yang memuat arsip kode pendukung untuk seluruh rangkaian Praktikum Kimia Kuantum. Kode-kode ini disusun sebagai alat bantu pembelajaran, pengolahan data, visualisasi grafik, parsing output, dan penyusunan laporan otomatis. Dengan adanya lampiran kode, mahasiswa tidak hanya memahami konsep kimia kuantum secara teoritis, tetapi juga memiliki perangkat praktis untuk mengolah hasil perhitungan secara sistematis dan reproduibel.

Dalam praktikum kimia kuantum modern, Python memiliki posisi penting karena dapat menghubungkan data komputasi dengan analisis ilmiah. Software kimia komputasi menghasilkan output berupa energi, geometri, frekuensi, orbital, momen dipol, dan data termodinamika. Namun, data tersebut sering tersebar dalam file log yang panjang. Python membantu membaca, menyaring, menghitung ulang, mengonversi satuan, membuat grafik, dan menyusun laporan.

Lampiran kode dalam subbab ini mencakup sembilan file utama:

1. praktikum_01_partikel_dalam_kotak.py
2. praktikum_02_optimasi_geometri.py

3. `praktikum_03_hartree_fock_basis_set.py`
4. `praktikum_04_dft_homo_lumo.py`
5. `praktikum_05_simetri_momen_dipol.py`
6. `praktikum_06_spektrum_ir.py`
7. `praktikum_07_energi_reaksi_termodinamika.py`
8. `parser_output_komputasi.py`
9. `template_laporan_otomatis.py`

Kode-kode ini tidak dimaksudkan untuk menggantikan software kimia kuantum profesional seperti ORCA, Gaussian, GAMESS, Psi4, PySCF, atau ChemCompute. Kode dalam lampiran ini berfungsi sebagai **pendamping praktikum**: membantu memahami konsep, membaca hasil, mengolah data, dan membuat laporan. Untuk perhitungan elektronik tingkat lanjut, mahasiswa tetap memerlukan software kimia komputasi khusus.

1. Pengertian Lampiran Kode Python

Lampiran kode Python adalah kumpulan skrip yang mendukung kegiatan praktikum. Setiap skrip memiliki tujuan tertentu. Ada skrip yang bersifat konseptual, seperti partikel dalam kotak. Ada skrip yang bersifat pengolahan data, seperti analisis HOMO-LUMO dan energi reaksi. Ada skrip yang bersifat visualisasi, seperti spektrum IR. Ada pula skrip yang bersifat otomatisasi, seperti parser output dan template laporan otomatis.

Dalam konteks buku ini, lampiran kode memiliki empat fungsi utama.

Pertama, sebagai **alat pembelajaran konsep**. Mahasiswa dapat melihat bagaimana persamaan kuantum sederhana diterjemahkan menjadi perhitungan numerik.

Kedua, sebagai **alat pengolahan data**. Mahasiswa dapat menghitung energi relatif, gap HOMO-LUMO, energi reaksi, dan perubahan energi akibat pelarut.

Ketiga, sebagai **alat visualisasi**. Mahasiswa dapat membuat grafik energi, spektrum IR, diagram orbital, dan diagram energi reaksi.

Keempat, sebagai **alat reproduibilitas**. Data dan grafik dapat dibuat ulang dari kode yang sama.

Dengan demikian, lampiran kode bukan sekadar tambahan teknis, tetapi bagian dari budaya kerja ilmiah.

2. Latar Historis Penggunaan Python dalam Kimia Komputasi

Dalam sejarah kimia komputasi, pemrograman selalu menjadi bagian penting. Pada masa awal, banyak program kimia kuantum dikembangkan menggunakan Fortran karena efisiensi

numeriknya. Seiring perkembangan komputasi ilmiah, Python menjadi semakin populer sebagai bahasa penghubung karena sintaksnya mudah dibaca, pustaka ilmiahnya luas, dan cocok untuk pengolahan data.

Python banyak digunakan dalam pendidikan dan penelitian karena menyediakan pustaka seperti NumPy untuk array numerik, pandas untuk data tabel, Matplotlib untuk visualisasi, SciPy untuk komputasi ilmiah, dan berbagai pustaka khusus kimia. Dalam praktikum, Python tidak harus digunakan untuk membangun program kimia kuantum penuh dari awal. Python dapat digunakan untuk tugas yang lebih realistis bagi mahasiswa, seperti membaca CSV, menghitung selisih energi, membuat grafik, dan memeriksa output.

Pendekatan ini sejalan dengan praktik komputasi modern: software khusus digunakan untuk menjalankan perhitungan berat, sedangkan Python digunakan untuk mengelola data, mengotomatisasi analisis, dan membuat laporan. Dengan demikian, mahasiswa belajar dua hal sekaligus: memahami konsep kimia kuantum dan mengembangkan literasi data komputasi.

3. Prinsip Umum Penulisan Kode Praktikum

Kode Python dalam lampiran ini mengikuti beberapa prinsip.

Pertama, kode harus **mudah dibaca**. Nama variabel dibuat jelas, komentar ditambahkan, dan struktur kode dibagi menjadi fungsi.

Kedua, kode harus **dapat dijalankan ulang**. Jika data input tersedia, mahasiswa dapat menjalankan kode dan memperoleh output yang sama.

Ketiga, kode harus **tidak terlalu bergantung pada sistem tertentu**. Kode menggunakan pustaka umum seperti `numpy`, `pandas`, dan `matplotlib`.

Keempat, kode harus **bersifat edukatif**. Kode tidak hanya menghasilkan angka, tetapi membantu mahasiswa memahami proses perhitungan.

Kelima, kode harus **jujur secara metodologis**. Jika kode bersifat model sederhana, maka harus disebutkan sebagai model sederhana. Misalnya, kode optimasi geometri dalam lampiran ini menggunakan model potensial sederhana untuk tujuan pembelajaran, bukan menggantikan optimasi geometri kuantum profesional.

Keenam, kode harus **mendukung laporan**. Setiap kode sebaiknya menghasilkan file CSV atau grafik yang dapat dimasukkan ke laporan praktikum.

Prinsip ini dapat diringkas sebagai berikut:

```
kode praktikum yang baik
= jelas
+ dapat dijalankan
```


- + terstruktur
- + edukatif
- + reprodusiabel
- + terhubung dengan laporan

4. Struktur Folder yang Dianjurkan

Agar semua kode mudah digunakan, struktur folder berikut dianjurkan:

```
praktikum_kimia_kuantum/
├── data/
│   ├── data_basis_set.csv
│   ├── data_orbital.csv
│   ├── data_ir.csv
│   ├── data_reaksi.csv
│   └── data_output.csv
├── output_log/
│   ├── h2o.log
│   ├── co2.log
│   └── nh3.log
├── kode/
│   ├── praktikum_01_partikel_dalam_kotak.py
│   ├── praktikum_02_optimasi_geometri.py
│   ├── praktikum_03_hartree_fock_basis_set.py
│   ├── praktikum_04_dft_homo_lumo.py
│   ├── praktikum_05_simetri_momen_dipol.py
│   ├── praktikum_06_spektrum_ir.py
│   ├── praktikum_07_energi_reaksi_termodinamika.py
│   ├── parser_output_komputasi.py
│   └── template_laporan_otomatis.py
├── grafik/
│   ├── partikel_dalam_kotak.png
│   ├── optimasi_geometri.png
│   ├── diagram_homo_lumo.png
│   ├── spektrum_ir.png
│   └── diagram_energi_reaksi.png
├── ringkasan/
│   ├── ringkasan_output_komputasi.csv
│   └── laporan_otomatis.md
└── README.txt
```

Struktur folder ini membantu mahasiswa memisahkan data mentah, kode, grafik, output, dan laporan.

5. Kebutuhan Pustaka Python

Kode dalam lampiran ini menggunakan pustaka berikut:

```
numpy
pandas
matplotlib
pathlib
re
```

Pustaka `pathlib` dan `re` merupakan bagian dari pustaka standar Python. Pustaka `numpy`, `pandas`, dan `matplotlib` biasanya perlu dipasang terlebih dahulu.

Perintah instalasi:

```
pip install numpy pandas matplotlib
```

Jika menggunakan Anaconda atau Miniconda:

```
conda install numpy pandas matplotlib
```

Mahasiswa perlu memastikan Python sudah terpasang dengan benar sebelum menjalankan kode.

Lampiran Kode 1

`praktikum_01_partikel_dalam_kotak.py`

Kode ini digunakan untuk memvisualisasikan model partikel dalam kotak satu dimensi. Model ini penting dalam kimia kuantum karena memperkenalkan konsep kuantisasi energi, fungsi gelombang, bilangan kuantum, dan probabilitas.

```
# =====
# praktikum_01_partikel_dalam_kotak.py
# Praktikum Kimia Kuantum
# Model: Partikel dalam kotak satu dimensi
# =====

from pathlib import Path
import numpy as np
import matplotlib.pyplot as plt

# Konstanta fisika
h = 6.62607015e-34          # konstanta Planck (J s)
m_e = 9.1093837015e-31     # massa elektron (kg)
eV = 1.602176634e-19       # 1 eV dalam joule
```

```

def energi_partikel_kotak(n, L, massa=m_e):
    """
    Menghitung energi partikel dalam kotak 1D.
     $E_n = n^2 h^2 / (8 m L^2)$ 

    n      : bilangan kuantum utama
    L      : panjang kotak dalam meter
    massa  : massa partikel dalam kg
    """
    return (n**2 * h**2) / (8 * massa * L**2)

def fungsi_gelombang(n, x, L):
    """
    Fungsi gelombang ternormalisasi untuk partikel dalam kotak 1D.
     $\psi_n(x) = \sqrt{2/L} \sin(n \pi x / L)$ 
    """
    return np.sqrt(2 / L) * np.sin(n * np.pi * x / L)

def main():
    folder_grafik = Path("grafik")
    folder_grafik.mkdir(exist_ok=True)

    # Panjang kotak 1 nm
    L = 1.0e-9

    # Grid posisi
    x = np.linspace(0, L, 1000)

    # Bilangan kuantum yang ditampilkan
    daftar_n = [1, 2, 3, 4]

    print("Energi partikel dalam kotak 1D")
    print(f"Panjang kotak = {L:.2e} m")
    print()

    for n in daftar_n:
        E_joule = energi_partikel_kotak(n, L)
        E_ev = E_joule / eV
        print(f"n = {n:2d} | Energi = {E_ev:10.5f} eV")

    plt.figure(figsize=(8, 5))

    for n in daftar_n:
        psi = fungsi_gelombang(n, x, L)
        # Dinormalisasi untuk visualisasi agar skala grafik mudah dibaca
        psi_plot = psi / np.max(np.abs(psi))
        plt.plot(x * 1e9, psi_plot + 1.5 * (n - 1), label=f"n = {n}")

    plt.xlabel("Posisi x (nm)")
    plt.ylabel("Fungsi gelombang relatif")
    plt.title("Fungsi Gelombang Partikel dalam Kotak 1D")
    plt.legend()
    plt.tight_layout()
    plt.savefig(folder_grafik / "partikel_dalam_kotak.png", dpi=300)

```

```
plt.close()

print()
print("Grafik disimpan: grafik/partikel_dalam_kotak.png")

if __name__ == "__main__":
    main()
```

Kode ini menunjukkan bahwa energi meningkat sebanding dengan n^2 . Mahasiswa dapat mengubah panjang kotak untuk melihat bagaimana ukuran sistem memengaruhi energi. Semakin kecil kotak, semakin besar energi kuantisasi.

Lampiran Kode 2

praktikum_02_optimasi_geometri.py

Kode ini digunakan untuk memperkenalkan konsep optimasi geometri secara sederhana. Dalam software kimia kuantum profesional, optimasi geometri dilakukan berdasarkan energi elektronik. Dalam kode pembelajaran ini, optimasi ditunjukkan menggunakan model potensial Morse untuk ikatan diatomik.

Kode ini tidak menggantikan optimasi geometri kuantum profesional. Tujuannya adalah memperlihatkan gagasan bahwa struktur stabil berkaitan dengan energi minimum.

```
# =====
# praktikum_02_optimasi_geometri.py
# Praktikum Kimia Kuantum
# Model: Optimasi jarak ikatan menggunakan potensial Morse
# =====

from pathlib import Path
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

def morse_potential(r, D_e=430.0, a=1.8, r_e=0.96):
    """
    Potensial Morse sederhana.
    r : jarak ikatan dalam Angstrom
    D_e : kedalaman potensial dalam kJ/mol
    a : parameter lebar potensial
    r_e : jarak kesetimbangan dalam Angstrom
    """
    return D_e * (1 - np.exp(-a * (r - r_e)))**2

def main():
```

```

folder_data = Path("data")
folder_grafik = Path("grafik")
folder_ringkasan = Path("ringkasan")

folder_data.mkdir(exist_ok=True)
folder_grafik.mkdir(exist_ok=True)
folder_ringkasan.mkdir(exist_ok=True)

# Rentang jarak ikatan O-H sederhana
r = np.linspace(0.5, 2.5, 800)
energi = morse_potential(r)

indeks_min = np.argmin(energi)
r_min = r[indeks_min]
e_min = energi[indeks_min]

df = pd.DataFrame({
    "jarak_Angstrom": r,
    "energi_kJmol": energi
})

df.to_csv(folder_ringkasan / "optimasi_geometri_morse.csv", index=False)

plt.figure(figsize=(8, 5))
plt.plot(r, energi)
plt.scatter([r_min], [e_min], s=80)
plt.xlabel("Jarak ikatan (Å)")
plt.ylabel("Energi relatif (kJ/mol)")
plt.title("Model Optimasi Geometri dengan Potensial Morse")
plt.tight_layout()
plt.savefig(folder_grafik / "optimasi_geometri.png", dpi=300)
plt.close()

print("Hasil optimasi geometri sederhana")
print(f"Jarak ikatan minimum = {r_min:.4f} Å")
print(f"Energi minimum      = {e_min:.4f} kJ/mol")
print("CSV disimpan          = ringkasan/optimasi_geometri_morse.csv")
print("Grafik disimpan         = grafik/optimasi_geometri.png")

if __name__ == "__main__":
    main()

```

Kode ini mengajarkan bahwa optimasi geometri adalah pencarian titik energi minimum. Dalam perhitungan molekul nyata, koordinat yang dioptimasi jauh lebih banyak daripada satu jarak ikatan. Namun, prinsip dasarnya tetap sama: struktur stabil berkaitan dengan energi minimum pada permukaan energi potensial.

Lampiran Kode 3

praktikum_03_hartree_fock_basis_set.py

Kode ini digunakan untuk menganalisis pengaruh basis set terhadap hasil perhitungan Hartree-Fock atau metode lain berdasarkan data CSV. Kode tidak menjalankan perhitungan Hartree-Fock dari awal, tetapi mengolah hasil output yang sudah diperoleh dari software komputasi.

Format data yang digunakan:

```
basis_set,energi_Hartree,panjang_ikatan_Angstrom
STO-3G,-1.1167,0.735
3-21G,-1.1229,0.741
6-31G,-1.1268,0.742
6-31G(d),-1.1285,0.741
```

Kode:

```
# =====
# praktikum_03_hartree_fock_basis_set.py
# Praktikum Kimia Kuantum
# Analisis pengaruh basis set terhadap energi dan geometri
# =====

from pathlib import Path
import pandas as pd
import matplotlib.pyplot as plt

HARTREE_TO_KJMOL = 2625.5

def buat_data_contoh(file_data):
    """
    Membuat data contoh jika file CSV belum tersedia.
    Angka hanya contoh edukatif, bukan data referensi mutlak.
    """
    df = pd.DataFrame({
        "basis_set": ["STO-3G", "3-21G", "6-31G", "6-31G(d)"],
        "energi_Hartree": [-1.1167, -1.1229, -1.1268, -1.1285],
        "panjang_ikatan_Angstrom": [0.735, 0.741, 0.742, 0.741]
    })
    df.to_csv(file_data, index=False)

def main():
    folder_data = Path("data")
    folder_grafik = Path("grafik")
    folder_ringkasan = Path("ringkasan")

    folder_data.mkdir(exist_ok=True)
    folder_grafik.mkdir(exist_ok=True)
    folder_ringkasan.mkdir(exist_ok=True)
```

```

file_data = folder_data / "data_basis_set.csv"

if not file_data.exists():
    buat_data_contoh(file_data)
    print("File data_basis_set.csv belum ada, dibuat data contoh.")

df = pd.read_csv(file_data)

energi_min = df["energi_Hartree"].min()
df["energi_relatif_kJmol"] = (df["energi_Hartree"] - energi_min) *
HARTREE_TO_KJMOL

df.to_csv(folder_ringkasan / "ringkasan_basis_set.csv", index=False)

plt.figure(figsize=(8, 5))
plt.bar(df["basis_set"], df["energi_relatif_kJmol"])
plt.xlabel("Basis set")
plt.ylabel("Energi relatif (kJ/mol)")
plt.title("Pengaruh Basis Set terhadap Energi Relatif")
plt.xticks(rotation=30, ha="right")
plt.tight_layout()
plt.savefig(folder_grafik / "pengaruh_basis_set_energi.png", dpi=300)
plt.close()

plt.figure(figsize=(8, 5))
plt.plot(df["basis_set"], df["panjang_ikatan_Angstrom"], marker="o")
plt.xlabel("Basis set")
plt.ylabel("Panjang ikatan (Å)")
plt.title("Pengaruh Basis Set terhadap Panjang Ikatan")
plt.xticks(rotation=30, ha="right")
plt.tight_layout()
plt.savefig(folder_grafik / "pengaruh_basis_set_geometri.png", dpi=300)
plt.close()

print("Ringkasan basis set:")
print(df.to_string(index=False))
print()
print("CSV disimpan      = ringkasan/ringkasan_basis_set.csv")
print("Grafik energi      = grafik/pengaruh_basis_set_energi.png")
print("Grafik geometri     = grafik/pengaruh_basis_set_geometri.png")

if __name__ == "__main__":
    main()

```

Kode ini menunjukkan bahwa basis set dapat memengaruhi energi dan geometri. Dalam laporan, mahasiswa harus menjelaskan bahwa basis set lebih besar tidak otomatis selalu “benar mutlak”, tetapi biasanya memberikan representasi orbital yang lebih fleksibel dengan biaya komputasi lebih tinggi.

Lampiran Kode 4

praktikum_04_dft_homo_lumo.py

Kode ini digunakan untuk menghitung gap HOMO-LUMO dan membuat diagram orbital dari data CSV.

Format data:

```
molekul,HOMO_eV,LUMO_eV
benzena,-6.20,-1.10
anilina,-5.70,-0.80
nitrobenzena,-6.80,-2.50
```

Kode:

```
# =====
# praktikum_04_dft_homo_lumo.py
# Praktikum Kimia Kuantum
# Analisis HOMO-LUMO dan gap energi
# =====

from pathlib import Path
import pandas as pd
import matplotlib.pyplot as plt

def buat_data_contoh(file_data):
    """
    Membuat data contoh jika file belum tersedia.
    Data hanya untuk latihan pengolahan, bukan data rujukan mutlak.
    """
    df = pd.DataFrame({
        "molekul": ["benzena", "anilina", "nitrobenzena"],
        "HOMO_eV": [-6.20, -5.70, -6.80],
        "LUMO_eV": [-1.10, -0.80, -2.50]
    })
    df.to_csv(file_data, index=False)

def main():
    folder_data = Path("data")
    folder_grafik = Path("grafik")
    folder_ringkasan = Path("ringkasan")

    folder_data.mkdir(exist_ok=True)
    folder_grafik.mkdir(exist_ok=True)
    folder_ringkasan.mkdir(exist_ok=True)

    file_data = folder_data / "data_orbital.csv"

    if not file_data.exists():
        buat_data_contoh(file_data)
```



```

    print("File data_orbital.csv belum ada, dibuat data contoh.")

df = pd.read_csv(file_data)
df["gap_HOMO_LUMO_eV"] = df["LUMO_eV"] - df["HOMO_eV"]

df.to_csv(folder_ringkasan / "ringkasan_homo_lumo.csv", index=False)

# Grafik gap
plt.figure(figsize=(8, 5))
plt.bar(df["molekul"], df["gap_HOMO_LUMO_eV"])
plt.xlabel("Molekul")
plt.ylabel("Gap HOMO-LUMO (eV)")
plt.title("Perbandingan Gap HOMO-LUMO")
plt.xticks(rotation=30, ha="right")
plt.tight_layout()
plt.savefig(folder_grafik / "gap_homo_lumo.png", dpi=300)
plt.close()

# Diagram orbital
plt.figure(figsize=(8, 6))
x_pos = range(len(df))

plt.scatter(x_pos, df["HOMO_eV"], marker="_", s=700, label="HOMO")
plt.scatter(x_pos, df["LUMO_eV"], marker="_", s=700, label="LUMO")

for i, baris in df.iterrows():
    plt.plot([i, i], [baris["HOMO_eV"], baris["LUMO_eV"]], linewidth=1)
    plt.text(
        i,
        (baris["HOMO_eV"] + baris["LUMO_eV"]) / 2,
        f"{baris['gap_HOMO_LUMO_eV']:.2f} eV",
        ha="center",
        va="center"
    )

plt.xticks(list(x_pos), df["molekul"], rotation=30, ha="right")
plt.ylabel("Energi orbital (eV)")
plt.title("Diagram HOMO-LUMO")
plt.legend()
plt.tight_layout()
plt.savefig(folder_grafik / "diagram_homo_lumo.png", dpi=300)
plt.close()

print("Ringkasan HOMO-LUMO:")
print(df.to_string(index=False))
print()
print("CSV disimpan          = ringkasan/ringkasan_homo_lumo.csv")
print("Grafik gap             = grafik/gap_homo_lumo.png")
print("Diagram HOMO-LUMO       = grafik/diagram_homo_lumo.png")

if __name__ == "__main__":
    main()

```

Kode ini mengajarkan bahwa HOMO, LUMO, dan gap dapat dihitung serta divisualisasikan secara sederhana. Dalam analisis, mahasiswa harus tetap hati-hati: gap HOMO-LUMO adalah indikator kualitatif, bukan satu-satunya ukuran reaktivitas.

Lampiran Kode 5

praktikum_05_simetri_momen_dipol.py

Kode ini digunakan untuk menghitung momen dipol sederhana berdasarkan koordinat atom dan muatan parsial. Kode ini bersifat edukatif untuk menunjukkan bahwa momen dipol adalah besaran vektor.

```
# =====
# praktikum_05_simetri_momen_dipol.py
# Praktikum Kimia Kuantum
# Analisis sederhana momen dipol dari muatan parsial
# =====

from pathlib import Path
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

# Faktor konversi:
# 1 e * Angstrom = 4.80320427 Debye
EANGSTROM_TO_DEBYE = 4.80320427

def hitung_momen_dipol(koordinat, muatan):
    """
    Menghitung vektor momen dipol sederhana:
    mu = sum(q_i * r_i)

    koordinat : array Nx3 dalam Angstrom
    muatan     : array N dalam satuan muatan elementer
    """
    koordinat = np.array(koordinat, dtype=float)
    muatan = np.array(muatan, dtype=float)

    mu_vector_eA = np.sum(muatan[:, None] * koordinat, axis=0)
    mu_vector_D = mu_vector_eA * EANGSTROM_TO_DEBYE
    mu_total_D = np.linalg.norm(mu_vector_D)

    return mu_vector_D, mu_total_D

def data_molekul_contoh():
    """
    Data muatan parsial sederhana untuk latihan.
```

Nilai hanya contoh edukatif.
 """

```
molekul = {
    "H2O": {
        "atom": ["O", "H", "H"],
        "koordinat": [
            [0.0000, 0.0000, 0.0000],
            [0.9572, 0.0000, 0.0000],
            [-0.2390, 0.9270, 0.0000]
        ],
        "muatan": [-0.84, 0.42, 0.42]
    },
    "CO2": {
        "atom": ["O", "C", "O"],
        "koordinat": [
            [-1.1600, 0.0000, 0.0000],
            [0.0000, 0.0000, 0.0000],
            [1.1600, 0.0000, 0.0000]
        ],
        "muatan": [-0.35, 0.70, -0.35]
    },
    "NH3": {
        "atom": ["N", "H", "H", "H"],
        "koordinat": [
            [0.0000, 0.0000, 0.1000],
            [0.9400, 0.0000, -0.2500],
            [-0.4700, 0.8140, -0.2500],
            [-0.4700, -0.8140, -0.2500]
        ],
        "muatan": [-0.90, 0.30, 0.30, 0.30]
    }
}

return molekul

def main():
    folder_grafik = Path("grafik")
    folder_ringkasan = Path("ringkasan")

    folder_grafik.mkdir(exist_ok=True)
    folder_ringkasan.mkdir(exist_ok=True)

    hasil = []

    for nama, data in data_molekul_contoh().items():
        mu_vec, mu_total = hitung_momen_dipol(data["koordinat"],
        data["muatan"])
        hasil.append({
            "molekul": nama,
            "mu_x_Debye": mu_vec[0],
            "mu_y_Debye": mu_vec[1],
            "mu_z_Debye": mu_vec[2],
            "momen_dipol_total_Debye": mu_total
        })
```

```

df = pd.DataFrame(hasil)
df.to_csv(folder_ringkasan / "ringkasan_momen_dipol.csv", index=False)

plt.figure(figsize=(8, 5))
plt.bar(df["molekul"], df["momen_dipol_total_Debye"])
plt.xlabel("Molekul")
plt.ylabel("Momen dipol total (Debye)")
plt.title("Perbandingan Momen Dipol Molekul")
plt.tight_layout()
plt.savefig(folder_grafik / "momen_dipol.png", dpi=300)
plt.close()

print("Ringkasan momen dipol:")
print(df.to_string(index=False))
print()
print("CSV disimpan      = ringkasan/ringkasan_momen_dipol.csv")
print("Grafik disimpan = grafik/momen_dipol.png")

if __name__ == "__main__":
    main()

```

Kode ini menekankan bahwa momen dipol total bergantung pada vektor. Molekul dengan ikatan polar dapat memiliki momen total mendekati nol jika susunannya simetris.

Lampiran Kode 6

praktikum_06_spektrum_ir.py

Kode ini digunakan untuk membuat spektrum IR dari data frekuensi dan intensitas. Kode dapat membuat spektrum garis dan spektrum dengan pelebaran Gaussian.

Format data:

```

frekuensi_cm1,intensitas
520.5,10
1050.2,22
1710.8,85
2960.1,40

```

Kode:

```

# =====
# praktikum_06_spektrum_ir.py
# Praktikum Kimia Kuantum
# Visualisasi spektrum IR dari frekuensi dan intensitas
# =====

from pathlib import Path

```

```

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

def buat_data_contoh(file_data):
    """
    Membuat data contoh IR jika file belum tersedia.
    """
    df = pd.DataFrame({
        "frekuensi_cm1": [520.5, 1050.2, 1710.8, 2960.1, 3050.0],
        "intensitas": [10, 22, 85, 40, 25]
    })
    df.to_csv(file_data, index=False)

def spektrum_gaussian(frekuensi, intensitas, x, lebar=15.0):
    """
    Membuat spektrum IR dengan pelebaran Gaussian.
    """
    y = np.zeros_like(x)

    for f, inten in zip(frekuensi, intensitas):
        y += inten * np.exp(-0.5 * ((x - f) / lebar) ** 2)

    return y

def main():
    folder_data = Path("data")
    folder_grafik = Path("grafik")
    folder_ringkasan = Path("ringkasan")

    folder_data.mkdir(exist_ok=True)
    folder_grafik.mkdir(exist_ok=True)
    folder_ringkasan.mkdir(exist_ok=True)

    file_data = folder_data / "data_ir.csv"

    if not file_data.exists():
        buat_data_contoh(file_data)
        print("File data_ir.csv belum ada, dibuat data contoh.")

    df = pd.read_csv(file_data)
    df.to_csv(folder_ringkasan / "ringkasan_ir.csv", index=False)

    # Spektrum garis
    plt.figure(figsize=(9, 5))
    plt.vlines(df["frekuensi_cm1"], 0, df["intensitas"], linewidth=1)
    plt.xlabel("Bilangan gelombang (cm-1)")
    plt.ylabel("Intensitas IR")
    plt.title("Spektrum IR Komputasi - Garis")
    plt.gca().invert_xaxis()
    plt.tight_layout()
    plt.savefig(folder_grafik / "spektrum_ir_garis.png", dpi=300)
    plt.close()

```

```

# Spektrum Gaussian
x = np.linspace(400, 4000, 4000)
y = spektrum_gaussian(df["frekuensi_cm1"], df["intensitas"], x, lebar=18.0)

plt.figure(figsize=(9, 5))
plt.plot(x, y)
plt.xlabel("Bilangan gelombang (cm-1)")
plt.ylabel("Intensitas relatif")
plt.title("Spektrum IR Komputasi - Pelebaran Gaussian")
plt.gca().invert_xaxis()
plt.tight_layout()
plt.savefig(folder_grafik / "spektrum_ir_gaussian.png", dpi=300)
plt.close()

print("Data IR:")
print(df.to_string(index=False))
print()
print("CSV disimpan      = ringkasan/ringkasan_ir.csv")
print("Spektrum garis    = grafik/spektrum_ir_garis.png")
print("Spektrum Gaussian  = grafik/spektrum_ir_gaussian.png")

if __name__ == "__main__":
    main()

```

Kode ini membantu mahasiswa mengubah daftar frekuensi dan intensitas menjadi spektrum yang mudah dibaca. Dalam laporan, mahasiswa harus menjelaskan bahwa pelebaran Gaussian adalah visualisasi, bukan pengganti data eksperimen.

Lampiran Kode 7

praktikum_07_energi_reaksi_termodinamika.py

Kode ini digunakan untuk menghitung perubahan energi reaksi berdasarkan data reaktan dan produk. Kode dapat menghitung ΔE , ΔH , dan ΔG jika data tersedia.

Format data:

```

spesies, koefisien, energi_Hartree, entalpi_Hartree, gibbs_Hartree
reaktan_A, -1, -100.0000, -99.9900, -99.9700
produk_B, 1, -100.0200, -100.0100, -99.9950

```

Koefisien reaktan ditulis negatif, produk ditulis positif.

Kode:

```

# =====
# praktikum_07_energi_reaksi_termodinamika.py
# Praktikum Kimia Kuantum

```

```

# Analisis energi reaksi, entalpi, dan Gibbs
# =====

from pathlib import Path
import pandas as pd
import matplotlib.pyplot as plt

HARTREE_TO_KJMOL = 2625.5

def buat_data_contoh(file_data):
    """
    Membuat data contoh reaksi sederhana jika file belum tersedia.
    Koefisien reaktan negatif, produk positif.
    """
    df = pd.DataFrame({
        "spesies": ["Reaktan", "Produk"],
        "koefisien": [-1, 1],
        "energi_Hartree": [-100.0000, -100.0200],
        "entalpi_Hartree": [-99.9900, -100.0100],
        "gibbs_Hartree": [-99.9700, -99.9950]
    })
    df.to_csv(file_data, index=False)

def hitung_delta(df, kolom):
    """
    Menghitung perubahan besaran reaksi:
    Delta = jumlah koefisien * nilai spesies
    """
    return (df["koefisien"] * df[kolom]).sum()

def main():
    folder_data = Path("data")
    folder_grafik = Path("grafik")
    folder_ringkasan = Path("ringkasan")

    folder_data.mkdir(exist_ok=True)
    folder_grafik.mkdir(exist_ok=True)
    folder_ringkasan.mkdir(exist_ok=True)

    file_data = folder_data / "data_reaksi.csv"

    if not file_data.exists():
        buat_data_contoh(file_data)
        print("File data_reaksi.csv belum ada, dibuat data contoh.")

    df = pd.read_csv(file_data)

    delta_E = hitung_delta(df, "energi_Hartree") * HARTREE_TO_KJMOL
    delta_H = hitung_delta(df, "entalpi_Hartree") * HARTREE_TO_KJMOL
    delta_G = hitung_delta(df, "gibbs_Hartree") * HARTREE_TO_KJMOL

    ringkasan = pd.DataFrame({
        "besaran": ["Delta_E", "Delta_H", "Delta_G"],

```

```

        "nilai_kJmol": [delta_E, delta_H, delta_G]
    })

    ringkasan.to_csv(folder_ringkasan / "ringkasan_energi_reaksi.csv",
index=False)

    plt.figure(figsize=(7, 5))
    plt.bar(ringkasan["besaran"], ringkasan["nilai_kJmol"])
    plt.axhline(0, linewidth=1)
    plt.xlabel("Besaran termodinamika")
    plt.ylabel("Nilai (kJ/mol)")
    plt.title("Perubahan Energi Reaksi")
    plt.tight_layout()
    plt.savefig(folder_grafik / "energi_reaksi_termodinamika.png", dpi=300)
    plt.close()

    # Diagram energi Gibbs sederhana
    reaktan_G = 0.0
    produk_G = delta_G

    plt.figure(figsize=(7, 5))
    plt.plot(["Reaktan", "Produk"], [reaktan_G, produk_G], marker="o")
    plt.axhline(0, linewidth=1)
    plt.ylabel("Energi Gibbs relatif (kJ/mol)")
    plt.title("Diagram Energi Gibbs Reaksi")
    plt.tight_layout()
    plt.savefig(folder_grafik / "diagram_energi_gibbs.png", dpi=300)
    plt.close()

    print("Data reaksi:")
    print(df.to_string(index=False))
    print()
    print("Ringkasan:")
    print(ringkasan.to_string(index=False))
    print()
    print("CSV disimpan      = ringkasan/ringkasan_energi_reaksi.csv")
    print("Grafik bar        = grafik/energi_reaksi_termodinamika.png")
    print("Diagram Gibbs      = grafik/diagram_energi_gibbs.png")

if __name__ == "__main__":
    main()

```

Kode ini mengajarkan bahwa perubahan energi reaksi dihitung dari selisih produk dan reaktan dengan memperhatikan koefisien stoikiometri. Mahasiswa harus membedakan ΔE , ΔH , dan ΔG .

Lampiran Kode 8

parser_output_komputasi.py

Kode ini digunakan untuk membaca file output komputasi dan mengekstrak data penting seperti energi total, momen dipol, frekuensi, status job, charge, dan multiplicity. Kode ini bersifat umum dan dapat dikembangkan sesuai format output software yang digunakan.

```
# =====
# parser_output_komputasi.py
# Praktikum Kimia Kuantum
# Parser sederhana untuk file output komputasi
# =====

from pathlib import Path
import re
import pandas as pd

FOLDER_LOG = Path("output_log")
FOLDER_RINGKASAN = Path("ringkasan")
FOLDER_RINGKASAN.mkdir(exist_ok=True)

FILE_CSV = FOLDER_RINGKASAN / "ringkasan_output_komputasi.csv"

def baca_teks(path_file):
    """
    Membaca file sebagai teks.
    """
    return path_file.read_text(encoding="utf-8", errors="ignore")

def cari_status_normal(teks):
    """
    Mencari status job selesai normal atau bermasalah.
    """
    pola_normal = [
        "Normal termination",
        "ORCA TERMINATED NORMALLY",
        "EXECUTION OF GAMESS TERMINATED NORMALLY",
        "Job finished successfully",
        "Psi4 exiting successfully",
    ]

    for pola in pola_normal:
        if pola.lower() in teks.lower():
            return "Selesai normal"

    pola_error = [
        "Error termination",
        "SCF failed",
        "SCF NOT CONVERGED",
        "Optimization failed",
    ]
```

```

        "Maximum number of cycles exceeded",
        "abnormally",
    ]

    for pola in pola_error:
        if pola.lower() in teks.lower():
            return "Ada indikasi masalah"

    return "Tidak terdeteksi"

def cari_charge_multiplicity(teks):
    """
    Mencari charge dan multiplicity dari output.
    """
    charge = None
    multiplicity = None

    pola = r"Charge\s*=\s*([-+]?\d+)\s+Multiplicity\s*=\s*(\d+)"
    cocok = re.search(pola, teks, re.IGNORECASE)

    if cocok:
        charge = int(cocok.group(1))
        multiplicity = int(cocok.group(2))
        return charge, multiplicity

    pola_charge = r"Total\s+Charge\s*[:=]\s*([-+]?\d+)"
    cocok = re.search(pola_charge, teks, re.IGNORECASE)
    if cocok:
        charge = int(cocok.group(1))

    pola_mult = r"Multiplicity\s*[:=]\s*(\d+)"
    cocok = re.search(pola_mult, teks, re.IGNORECASE)
    if cocok:
        multiplicity = int(cocok.group(1))

    return charge, multiplicity

def cari_energi_total(teks):
    """
    Mencari energi total dari beberapa pola umum.
    """
    pola_list = [
        r"SCF Done:\s+E\([^\)]+\)\s*=\s*([-+]?\d+\.\d+(?:[Ee]([-+]?\d+)?)",
        r"FINAL SINGLE POINT ENERGY\s+([-+]?\d+\.\d+(?:[Ee]([-+]?\d+)?)",
        r"Total Energy\s*[:=]\s*([-+]?\d+\.\d+(?:[Ee]([-+]?\d+)?)",
        r"Total SCF energy\s*=\s*([-+]?\d+\.\d+(?:[Ee]([-+]?\d+)?)",
        r"E\ (TOTAL\)\s*=\s*([-+]?\d+\.\d+(?:[Ee]([-+]?\d+)?)",
    ]

    for pola in pola_list:
        hasil = re.findall(pola, teks, re.IGNORECASE)
        if hasil:
            return float(hasil[-1])

    return None

```

```

def cari_momen_dipol(teks):
    """
    Mencari momen dipol total dalam Debye.
    """
    pola_list = [
        r"Tot=\s*([-+]?\d+\.\d+?(:[Ee][-+]?\d+)?)",
        r"Total\s+Dipole\s+Moment\s*[:=]? \s*([-+]?\d+\.\d+?(:[Ee][-+]?
+]? \d+)?)",
        r"Magnitude\s*\ (Debye\)\s*[:=]? \s*([-+]?\d+\.\d+?(:[Ee][-+]? \d+)?)",
    ]

    for pola in pola_list:
        hasil = re.findall(pola, teks, re.IGNORECASE)
        if hasil:
            return float(hasil[-1])

    return None

def cari_frekuensi(teks):
    """
    Mengekstrak frekuensi vibrasi.
    """
    frekuensi = []

    pola_gaussian = r"Frequencies\s+--\s+(.+) "
    baris_frekuensi = re.findall(pola_gaussian, teks, re.IGNORECASE)

    for baris in baris_frekuensi:
        angka = re.findall(r"([-+]?\d+\.\d+?(:[Ee][-+]? \d+)?)", baris)
        frekuensi.extend([float(a) for a in angka])

    if not frekuensi:
        for baris in teks.splitlines():
            if "cm*-1" in baris.lower() or "cm^-1" in baris.lower():
                angka = re.findall(r"([-+]?\d+\.\d+?(:[Ee][-+]? \d+)?)", baris)
                frekuensi.extend([float(a) for a in angka])

    return frekuensi

def ringkas_frekuensi(frekuensi):
    """
    Membuat ringkasan frekuensi.
    """
    if not frekuensi:
        return {
            "jumlah_frekuensi": 0,
            "jumlah_frekuensi_imajiner": None,
            "frekuensi_min_cm-1": None,
            "frekuensi_max_cm-1": None,
        }

    return {
        "jumlah_frekuensi": len(frekuensi),
    }

```

```

        "jumlah_frekuensi_imajiner": sum(1 for f in frekuensi if f < 0),
        "frekuensi_min_cm-1": min(frekuensi),
        "frekuensi_max_cm-1": max(frekuensi),
    }

def parsing_satu_file(path_file):
    """
    Parsing satu file output.
    """
    teks = baca_teks(path_file)

    status = cari_status_normal(teks)
    charge, multiplicity = cari_charge_multiplicity(teks)
    energi = cari_energi_total(teks)
    dipol = cari_momen_dipol(teks)
    frekuensi = cari_frekuensi(teks)
    ringkas = ringkas_frekuensi(frekuensi)

    catatan = []

    if status != "Selesai normal":
        catatan.append("Status normal tidak terkonfirmasi")
    if energi is None:
        catatan.append("Energi tidak ditemukan")
    if dipol is None:
        catatan.append("Momen dipol tidak ditemukan")
    if charge is None:
        catatan.append("Charge tidak ditemukan")
    if multiplicity is None:
        catatan.append("Multiplicity tidak ditemukan")
    if ringkas["jumlah_frekuensi_imajiner"] not in [0, None]:
        catatan.append("Ada frekuensi imajiner")

    return {
        "nama_file": path_file.name,
        "status_job": status,
        "charge": charge,
        "multiplicity": multiplicity,
        "energi_total_Hartree": energi,
        "momen_dipol_Debye": dipol,
        **ringkas,
        "catatan": "; ".join(catatan) if catatan else "OK",
    }

def parsing_semua_file():
    """
    Parsing semua file .log, .out, dan .txt.
    """
    if not FOLDER_LOG.exists():
        raise FileNotFoundError("Folder output_log tidak ditemukan.")

    daftar_file = []
    daftar_file.extend(FOLDER_LOG.glob("*.log"))
    daftar_file.extend(FOLDER_LOG.glob("*.out"))
    daftar_file.extend(FOLDER_LOG.glob("*.txt"))

```

```

        if not daftar_file:
            raise FileNotFoundError("Tidak ada file output dalam folder
output_log.")

        hasil = [parsing_satu_file(path) for path in sorted(daftar_file)]
        df = pd.DataFrame(hasil)
        df.to_csv(FILE_CSV, index=False)

        print("Ringkasan output:")
        print(df.to_string(index=False))
        print()
        print(f"CSV disimpan = {FILE_CSV}")

if __name__ == "__main__":
    parsing_semua_file()

```

Kode ini harus diuji pada output nyata. Jika format output berbeda, pola `regex` perlu ditambahkan. Parser otomatis membantu mempercepat kerja, tetapi hasil parsing tetap harus divalidasi manual.

Lampiran Kode 9

template_laporan_otomatis.py

Kode ini digunakan untuk membuat draf laporan otomatis dalam format Markdown berdasarkan data proyek. Kode ini tidak menggantikan penulisan analisis mahasiswa. Kode hanya membantu menyusun kerangka laporan dan memasukkan tabel data.

Format data proyek:

```

komponen, isi
judul, Analisis HOMO-LUMO Benzena dan Nitrobenzena
nama, Kasmui
metode, DFT/B3LYP
basis_set, 6-31G(d)
software, ChemCompute

```

Kode:

```

# =====
# template_laporan_otomatis.py
# Praktikum Kimia Kuantum
# Membuat draf laporan otomatis dalam format Markdown
# =====

from pathlib import Path
import pandas as pd
from datetime import datetime

```

```

FOLDER_DATA = Path("data")
FOLDER_RINGKASAN = Path("ringkasan")
FOLDER_RINGKASAN.mkdir(exist_ok=True)

def buat_metadata_contoh(file_metadata):
    """
    Membuat metadata contoh jika file belum tersedia.
    """
    df = pd.DataFrame({
        "komponen": [
            "judul",
            "nama",
            "nim",
            "kelas",
            "metode",
            "basis_set",
            "software",
            "jenis_job"
        ],
        "isi": [
            "Analisis HOMO-LUMO Senyawa Aromatik Tersubstitusi",
            "Nama Mahasiswa",
            "NIM Mahasiswa",
            "Kelas",
            "DFT/B3LYP",
            "6-31G(d)",
            "ChemCompute/ORCA",
            "Optimasi geometri dan analisis orbital"
        ]
    })
    df.to_csv(file_metadata, index=False)

def baca_metadata(file_metadata):
    """
    Membaca metadata laporan dari CSV menjadi dictionary.
    """
    df = pd.read_csv(file_metadata)
    return dict(zip(df["komponen"], df["isi"]))

def baca_tabel_opsional(nama_file):
    """
    Membaca tabel CSV jika tersedia.
    Jika tidak ada, mengembalikan None.
    """
    path = FOLDER_DATA / nama_file
    if path.exists():
        return pd.read_csv(path)
    return None

def dataframe_to_markdown(df):
    """

```

```

Mengubah DataFrame menjadi tabel Markdown.
"""
if df is None or df.empty:
    return "_Data belum tersedia._"

return df.to_markdown(index=False)

def buat_laporan(metadata, tabel_hasil=None, tabel_validasi=None):
    """
    Membuat isi laporan dalam format Markdown.
    """
    tanggal = datetime.now().strftime("%d-%m-%Y")

    judul = metadata.get("judul", "Judul Proyek Mini")
    nama = metadata.get("nama", "Nama Mahasiswa")
    nim = metadata.get("nim", "NIM")
    kelas = metadata.get("kelas", "Kelas")
    metode = metadata.get("metode", "Metode belum diisi")
    basis_set = metadata.get("basis_set", "Basis set belum diisi")
    software = metadata.get("software", "Software belum diisi")
    jenis_job = metadata.get("jenis_job", "Jenis job belum diisi")

    laporan = f"""# LAPORAN PROYEK MINI PRAKTIKUM KIMIA KUANTUM

## Judul
{judul}

## Identitas
- Nama: {nama}
- NIM: {nim}
- Kelas: {kelas}
- Tanggal pembuatan draf: {tanggal}

---

## ABSTRAK
Tuliskan ringkasan tujuan, metode, molekul target, hasil utama, dan kesimpulan
proyek mini.

---

## 1. PENDAHULUAN
### 1.1 Latar Belakang
Tuliskan latar belakang ilmiah proyek.

### 1.2 Rumusan Masalah
1. Tuliskan rumusan masalah pertama.
2. Tuliskan rumusan masalah kedua jika ada.

### 1.3 Tujuan
1. Tuliskan tujuan proyek.
2. Tuliskan tujuan tambahan jika ada.

---

## 2. DASAR TEORI

```

Tuliskan teori yang relevan dengan proyek, misalnya HOMO-LUMO, IR, momen dipol, energi reaksi, efek pelarut, atau konformer.

3. METODE KOMPUTASI

- Software: {software}
- Metode: {metode}
- Basis set: {basis_set}
- Jenis job: {jenis_job}

Tuliskan charge, multiplicity, model pelarut, dan prosedur perhitungan secara lengkap.

4. KODE PYTHON

Tuliskan nama file kode Python, fungsi kode, input, output, dan grafik yang dihasilkan.

5. HASIL DAN PEMBAHASAN

5.1 Tabel Hasil

```
{dataframe_to_markdown(tabel_hasil)}
```

5.2 Validasi Data

```
{dataframe_to_markdown(tabel_validasi)}
```

5.3 Grafik

Sisipkan grafik hasil Python di bagian ini.

5.4 Pembahasan

Jelaskan makna hasil berdasarkan teori, data, grafik, dan keterbatasan metode.

6. KESIMPULAN

1. Tuliskan kesimpulan pertama.
2. Tuliskan kesimpulan kedua.
3. Tuliskan keterbatasan utama jika relevan.

7. DAFTAR PUSTAKA

Tuliskan referensi yang benar-benar digunakan.

8. LAMPIRAN

- Lampiran 1. File input
- Lampiran 2. Potongan output
- Lampiran 3. CSV
- Lampiran 4. Kode Python
- Lampiran 5. Grafik

"""


```

return laporan

def main():
    FOLDER_DATA.mkdir(exist_ok=True)

    file_metadata = FOLDER_DATA / "metadata_laporan.csv"

    if not file_metadata.exists():
        buat_metadata_contoh(file_metadata)
        print("File metadata_laporan.csv belum ada, dibuat data contoh.")

    metadata = baca_metadata(file_metadata)

    tabel_hasil = baca_tabel_opsional("tabel_hasil.csv")
    tabel_validasi = baca_tabel_opsional("tabel_validasi.csv")

    laporan = buat_laporan(metadata, tabel_hasil, tabel_validasi)

    file_laporan = FOLDER_RINGKASAN / "laporan_otomatis.md"
    file_laporan.write_text(laporan, encoding="utf-8")

    print(f"Draf laporan berhasil dibuat: {file_laporan}")

if __name__ == "__main__":
    main()

```

Kode ini menghasilkan draf laporan otomatis dalam format Markdown. Mahasiswa tetap harus menulis pendahuluan, teori, pembahasan, kesimpulan, dan daftar pustaka secara ilmiah. Kode hanya membantu menyusun kerangka laporan.

6. Cara Menjalankan Kode

Untuk menjalankan kode, mahasiswa dapat menggunakan terminal atau PowerShell. Misalnya:

```
python praktikum_01_partikel_dalam_kotak.py
```

Jika file berada dalam folder `kode`, jalankan dari folder utama:

```
python kode/praktikum_01_partikel_dalam_kotak.py
```

Pastikan folder `data`, `grafik`, `ringkasan`, dan `output_log` berada pada lokasi yang sesuai. Jika kode membuat data contoh, mahasiswa harus mengganti data contoh dengan data hasil praktikum sendiri sebelum menyusun laporan akhir.

7. Cara Menggunakan Kode dalam Laporan

Kode Python dapat disebutkan dalam laporan dengan format berikut:

Data hasil perhitungan diolah menggunakan Python. File kode yang digunakan adalah `praktikum_04_dft_homo_lumo.py`. Kode tersebut membaca data HOMO dan LUMO dari CSV, menghitung gap HOMO-LUMO, dan membuat diagram orbital dalam format PNG.

Untuk parser output:

File output komputasi dianalisis menggunakan `parser_output_komputasi.py`. Parser digunakan untuk mengekstrak energi total, momen dipol, frekuensi, charge, multiplicity, dan status job. Hasil parsing disimpan dalam `ringkasan_output_komputasi.csv`.

Untuk laporan otomatis:

Draf laporan disusun menggunakan `template_laporan_otomatis.py`, kemudian bagian pendahuluan, dasar teori, pembahasan, dan kesimpulan dilengkapi secara manual berdasarkan hasil analisis.

Mahasiswa harus menjelaskan fungsi kode, bukan hanya mencantumkan nama file.

8. Analisis Kritis terhadap Lampiran Kode

Lampiran kode sangat membantu, tetapi harus digunakan secara kritis. Kode dapat mempercepat pengolahan data, tetapi tidak menjamin bahwa data yang dimasukkan sudah benar. Jika input salah, output kode juga salah.

Prinsip yang harus diingat:

garbage in
→ garbage out

Artinya, data yang tidak valid akan menghasilkan grafik atau laporan yang tidak valid. Oleh karena itu, sebelum menjalankan kode pengolahan data, mahasiswa harus memastikan bahwa output komputasi sudah valid.

Beberapa risiko penggunaan kode:

1. Mahasiswa memakai data contoh tetapi menganggapnya sebagai data praktikum.
2. Mahasiswa lupa mengganti file CSV contoh.
3. Mahasiswa salah memasukkan satuan.
4. Mahasiswa salah menulis koefisien reaksi.
5. Mahasiswa tidak memeriksa frekuensi imajiner.
6. Mahasiswa membuat grafik dari output tidak konvergen.
7. Mahasiswa menyalin kode tanpa memahami fungsinya.

8. Mahasiswa tidak melampirkan data sumber.

Risiko ini harus dicegah dengan validasi data dan penjelasan kode dalam laporan.

9. Kesalahan Umum dalam Penggunaan Kode Python

Beberapa kesalahan umum adalah:

Kesalahan	Dampak
File CSV tidak berada di folder <code>data</code>	Kode gagal membaca input
Nama kolom CSV berbeda	Kode menghasilkan error
Satuan energi tidak konsisten	Perhitungan salah
Data contoh belum diganti	Laporan tidak sesuai praktikum
Grafik tidak diberi label	Hasil sulit ditafsirkan
Output tidak divalidasi	Data bermasalah tetap dianalisis
Kode tidak disimpan	Analisis tidak reproduisibel
Kode diedit tanpa catatan	Sulit melacak perubahan
Tidak mencantumkan kode di lampiran	Laporan sulit diperiksa
Tidak memahami fungsi kode	Mahasiswa tidak mampu menjawab evaluasi

Mahasiswa harus memeriksa kembali data, kode, dan grafik sebelum laporan dikumpulkan.

10. Implikasi Keilmuan Lampiran Kode

Lampiran kode memiliki implikasi keilmuan karena mendukung reproduibilitas. Dalam kimia komputasi, hasil tidak cukup hanya ditulis sebagai tabel. Data harus dapat ditelusuri kembali ke output dan kode pengolahan. Dengan adanya kode, pembaca dapat melihat bagaimana energi relatif dihitung, bagaimana spektrum dibuat, bagaimana gap HOMO-LUMO diperoleh, dan bagaimana laporan disusun.

Lampiran kode juga memperkuat integrasi antara kimia dan komputasi. Mahasiswa belajar bahwa kimia kuantum bukan hanya persamaan di papan tulis, tetapi juga praktik pengolahan data numerik. Kode menjadi jembatan antara teori dan laporan.

Alur ilmiahnya adalah:

```
teori
→ perhitungan
→ output
→ parsing
→ CSV
→ grafik
→ laporan
```

→ kesimpulan

Kode membantu menjaga agar alur tersebut dapat diperiksa ulang.

11. Implikasi Pedagogis Lampiran Kode

Secara pedagogis, lampiran kode melatih mahasiswa mengembangkan literasi komputasi. Mahasiswa belajar membaca kode, menjalankan kode, memahami input-output, dan menghubungkan hasil kode dengan konsep kimia.

Keterampilan yang dilatih meliputi:

1. Menjalankan Python.
2. Membaca CSV.
3. Mengolah data numerik.
4. Mengonversi satuan.
5. Membuat grafik.
6. Membaca output komputasi.
7. Menyusun laporan otomatis.
8. Memahami error sederhana.
9. Menjaga reproduibilitas.
10. Menjelaskan fungsi kode dalam laporan.

Bagi calon guru kimia, keterampilan ini sangat penting karena pembelajaran kimia modern semakin membutuhkan integrasi data, visualisasi, dan teknologi komputasi.

12. Sintesis Subbab 12.6

Subbab 12.6 telah menyajikan lampiran kode Python yang mendukung seluruh rangkaian Praktikum Kimia Kuantum. Kode yang disajikan meliputi partikel dalam kotak, optimasi geometri sederhana, analisis Hartree-Fock dan basis set, DFT dan HOMO-LUMO, simetri dan momen dipol, spektrum IR, energi reaksi dan termodinamika, parser output komputasi, serta template laporan otomatis.

Kode-kode ini membantu mahasiswa memahami konsep, mengolah data, membuat grafik, membaca output, dan menyusun laporan. Namun, kode harus digunakan secara kritis. Python membantu otomatisasi, tetapi validasi kimia tetap harus dilakukan oleh mahasiswa. Data yang tidak valid tidak akan menjadi benar hanya karena diolah dengan kode.

Alur penggunaan lampiran kode dapat diringkas:

data praktikum
→ kode Python

- CSV
- grafik
- validasi
- laporan
- evaluasi

Dengan lampiran kode ini, mahasiswa diharapkan mampu menyelesaikan praktikum secara lebih sistematis, efisien, dan reproduisibel.

13. Transisi ke Penutup Bab 12

Subbab 12.6 merupakan bagian akhir dari rangkaian subbab Bab 12. Setelah mahasiswa mempelajari proyek mini, format proposal, format laporan, evaluasi akhir, glosarium, dan lampiran kode Python, seluruh komponen penutup Praktikum Kimia Kuantum telah tersusun. Tahap berikutnya adalah menyusun penutup Bab 12 yang merangkum peran proyek mini, evaluasi, glosarium, dan kode Python sebagai integrasi akhir pembelajaran kimia kuantum dan kimia komputasi.

Penutup Bab 12

Bab 12 merupakan bab penutup yang mengintegrasikan seluruh rangkaian Praktikum Kimia Kuantum dalam bentuk proyek mini, proposal, laporan akhir, evaluasi, glosarium, dan lampiran kode Python. Jika bab-bab sebelumnya membimbing mahasiswa melalui berbagai kegiatan praktikum yang terstruktur, maka Bab 12 mengarahkan mahasiswa untuk menggabungkan semua keterampilan tersebut menjadi satu pengalaman ilmiah yang lebih mandiri, kritis, dan reproduisibel.

Bab ini menegaskan bahwa pembelajaran kimia kuantum dan kimia komputasi tidak cukup berhenti pada kemampuan menjalankan software. Mahasiswa harus mampu merancang masalah, memilih molekul, menentukan metode, menjalankan perhitungan, membaca output, memvalidasi data, mengolah hasil dengan Python, membuat grafik, menulis laporan, mempresentasikan hasil, dan mengevaluasi keterbatasan. Seluruh rangkaian tersebut membentuk kompetensi yang lebih utuh daripada sekadar memperoleh angka energi atau gambar struktur molekul.

Pada bagian awal, Bab 12 membahas **proyek mini Praktikum Kimia Kuantum**. Proyek mini menjadi ruang bagi mahasiswa untuk memilih salah satu topik, seperti analisis HOMO-LUMO senyawa aromatik tersubstitusi, perbandingan IR komputasi dan IR literatur, optimasi geometri molekul obat sederhana, studi efek pelarut terhadap kestabilan ion, analisis konformer etanol atau butana, simetri dan polaritas molekul anorganik, energi ikatan hidrogen dimer air, atau energi reaksi sederhana fase gas dan pelarut. Melalui proyek mini, mahasiswa belajar bahwa kimia komputasi adalah proses ilmiah yang dimulai dari pertanyaan, bukan dari software.

Selanjutnya, Bab 12 menjelaskan **format proposal proyek mini**. Proposal berfungsi sebagai rencana kerja sebelum proyek dijalankan. Di dalam proposal, mahasiswa harus menuliskan judul, latar belakang, rumusan masalah, molekul target, metode komputasi, software, kode Python

pendukung, data yang dikumpulkan, validasi, dan jadwal kerja. Proposal membantu mahasiswa membatasi ruang lingkup proyek agar tetap realistis, terarah, dan dapat diselesaikan dalam waktu praktikum.

Bab ini kemudian membahas **format laporan proyek mini**. Laporan proyek mini merupakan dokumen pertanggungjawaban ilmiah yang memuat abstrak, pendahuluan, dasar teori, metode komputasi, kode Python, hasil dan pembahasan, kesimpulan, daftar pustaka, serta lampiran input-output. Laporan menjadi bukti bahwa mahasiswa tidak hanya menjalankan perhitungan, tetapi juga mampu menjelaskan hasil secara sistematis, argumentatif, dan berbasis data.

Bagian berikutnya membahas **evaluasi akhir praktikum**. Evaluasi akhir tidak hanya berupa ujian teori, tetapi juga mencakup ujian membaca output, ujian menjalankan kode Python, ujian interpretasi grafik, presentasi proyek mini, dan laporan akhir. Bentuk evaluasi seperti ini lebih sesuai dengan karakter kimia komputasi karena kompetensi yang dinilai bersifat konseptual, teknis, analitis, dan komunikatif sekaligus. Mahasiswa dinilai bukan hanya dari hasil akhir, tetapi juga dari proses berpikir, validasi data, ketelitian membaca output, dan kemampuan menyampaikan kesimpulan secara ilmiah.

Bab 12 juga menyajikan **glosarium kimia kuantum dan komputasi**. Glosarium membantu mahasiswa memahami istilah utama seperti *ab initio*, basis set, ChemCompute, DFT, energi total, frekuensi imajiner, fungsi gelombang, Hartree-Fock, HOMO, LUMO, momen dipol, optimasi geometri, orbital molekul, SCF, simetri molekul, dan *zero-point energy*. Pemahaman istilah yang tepat sangat penting karena kesalahan istilah sering kali mencerminkan kesalahan konsep. Dengan glosarium, mahasiswa memiliki rujukan ringkas untuk membaca output, menulis laporan, menyusun proposal, dan mempersiapkan evaluasi.

Pada bagian akhir, Bab 12 memuat **lampiran kode Python**. Lampiran kode mencakup praktikum_01_partikel_dalam_kotak.py, praktikum_02_optimasi_geometri.py, praktikum_03_hartree_fock_basis_set.py, praktikum_04_dft_homo_lumo.py, praktikum_05_simetri_momen_dipol.py, praktikum_06_spektrum_ir.py, praktikum_07_energi_reaksi_termodinamika.py, parser_output_komputasi.py, dan template_laporan_otomatis.py. Kode-kode tersebut menjadi arsip pembelajaran yang dapat digunakan ulang, dimodifikasi, dan dikembangkan sesuai kebutuhan praktikum.

Secara keseluruhan, Bab 12 memperlihatkan bahwa praktikum kimia kuantum memiliki alur akhir sebagai berikut:

```
pemahaman konsep
→ pelaksanaan praktikum
→ proyek mini
→ proposal
→ perhitungan komputasi
→ validasi output
→ pengolahan Python
→ grafik
→ laporan
→ presentasi
→ evaluasi akhir
```

→ refleksi pembelajaran

Alur ini menunjukkan bahwa kimia komputasi tidak hanya berkaitan dengan perhitungan, tetapi juga dengan perencanaan, dokumentasi, validasi, analisis, dan komunikasi ilmiah.

Bab 12 juga menegaskan pentingnya **reprodusibilitas**. Setiap proyek mini harus dapat ditelusuri kembali ke file input, output, metode, basis set, charge, multiplicity, data CSV, kode Python, grafik, dan laporan. Reprodusibilitas menjadi salah satu ciri penting kerja ilmiah modern. Tanpa reprodusibilitas, hasil komputasi sulit diperiksa dan sulit dipercaya.

Selain reprodusibilitas, Bab 12 menekankan pentingnya **validasi**. Mahasiswa harus memahami bahwa tidak semua output yang berhasil keluar dari software otomatis valid. Output harus diperiksa: apakah job selesai normal, apakah SCF konvergen, apakah optimasi berhasil, apakah frekuensi imajiner muncul, apakah charge dan multiplicity benar, serta apakah geometri akhir masuk akal. Validasi menjadi batas antara hasil komputasi mentah dan data ilmiah yang layak dianalisis.

Bab ini juga memperkuat peran **Python sebagai alat bantu ilmiah**. Python digunakan untuk membaca data, mengolah CSV, menghitung energi relatif, membuat grafik, membuat spektrum IR, menghitung gap HOMO-LUMO, mem-parsing output, dan menyusun draf laporan. Namun, Bab 12 juga menegaskan bahwa Python bukan pengganti pemahaman kimia. Python mempercepat pengolahan data, tetapi interpretasi tetap harus didasarkan pada teori, validasi, dan argumentasi ilmiah.

Dari sisi pedagogis, Bab 12 membantu mahasiswa bergerak dari pembelajar yang mengikuti instruksi menjadi pembelajar yang mulai mampu bekerja mandiri. Mahasiswa dilatih untuk memilih topik, merumuskan masalah, mengatur jadwal, menjalankan perhitungan, mengolah data, dan mempertanggungjawabkan hasil. Keterampilan ini sangat penting, terutama bagi calon guru kimia dan calon peneliti yang akan menggunakan komputasi sebagai alat pembelajaran, penelitian, atau pengembangan bahan ajar.

Bab 12 juga menempatkan laporan dan presentasi sebagai bagian penting dari pembelajaran. Hasil komputasi yang baik tidak cukup jika tidak mampu dijelaskan. Mahasiswa harus mampu mengubah output menjadi tabel, tabel menjadi grafik, grafik menjadi pembahasan, dan pembahasan menjadi kesimpulan. Dengan demikian, pembelajaran kimia kuantum menjadi kegiatan ilmiah yang utuh: berpikir, menghitung, memvalidasi, menafsirkan, dan mengomunikasikan.

Puncak dari Bab 12 adalah kesadaran bahwa Praktikum Kimia Kuantum bukan sekadar rangkaian tugas teknis. Praktikum ini adalah latihan membangun kebiasaan ilmiah. Mahasiswa belajar teliti terhadap data, hati-hati terhadap kesimpulan, jujur terhadap keterbatasan, terbuka terhadap validasi, dan sistematis dalam dokumentasi. Kebiasaan ini jauh lebih penting daripada sekadar menyelesaikan satu perhitungan.

Dengan selesainya Bab 12, seluruh rangkaian Praktikum Kimia Kuantum Bagian 2 telah memiliki penutup yang utuh. Mahasiswa telah dibimbing dari konsep dasar, pelaksanaan praktikum, analisis output, otomatisasi Python, proyek mini, evaluasi akhir, glosarium, hingga lampiran kode. Seluruh

bagian tersebut membentuk satu kesatuan pembelajaran yang menghubungkan teori kimia kuantum dengan praktik komputasi modern.

Akhirnya, Bab 12 mengajak mahasiswa melihat kimia kuantum dan kimia komputasi sebagai keterampilan ilmiah masa depan. Kemampuan memahami teori, menggunakan software, membaca output, menulis kode sederhana, membuat grafik, menyusun laporan, dan mempresentasikan hasil merupakan bekal penting dalam pendidikan kimia modern. Dengan bekal tersebut, mahasiswa tidak hanya menjadi pengguna teknologi, tetapi juga menjadi pembelajar dan calon pendidik yang mampu menggunakan teknologi secara kritis, kreatif, dan bertanggung jawab.

Daftar Pustaka Bab 12

- Atkins, P., & de Paula, J. (2010). *Atkins' Physical Chemistry* (9th ed.). Oxford University Press.
- Beazley, D., & Jones, B. K. (2013). *Python Cookbook* (3rd ed.). O'Reilly Media.
- Bell, S. (2010). Project-based learning for the 21st century: Skills for the future. *The Clearing House: A Journal of Educational Strategies, Issues and Ideas*, 83(2), 39–43.
- Biggs, J., & Tang, C. (2011). *Teaching for Quality Learning at University* (4th ed.). Open University Press.
- Cramer, C. J. (2004). *Essentials of Computational Chemistry: Theories and Models* (2nd ed.). John Wiley & Sons.
- Foresman, J. B., & Frisch, Æ. (2015). *Exploring Chemistry with Electronic Structure Methods* (3rd ed.). Gaussian, Inc.
- Friedl, J. E. F. (2006). *Mastering Regular Expressions* (3rd ed.). O'Reilly Media.
- Gordon, M. S., & Schmidt, M. W. (2005). Advances in electronic structure theory: GAMESS a decade later. In C. E. Dykstra, G. Frenking, K. S. Kim, & G. E. Scuseria (Eds.), *Theory and Applications of Computational Chemistry: The First Forty Years* (pp. 1167–1189). Elsevier.
- Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., & Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585, 357–362.
- Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3), 90–95.
- Jensen, F. (2017). *Introduction to Computational Chemistry* (3rd ed.). John Wiley & Sons.

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B. E., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J. B., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S., & Willing, C. (2016). Jupyter Notebooks—a publishing format for reproducible computational workflows. In F. Loizides & B. Schmidt (Eds.), *Positioning and Power in Academic Publishing: Players, Agents and Agendas* (pp. 87–90). IOS Press.

Leach, A. R. (2001). *Molecular Modelling: Principles and Applications* (2nd ed.). Pearson Education.

Levine, I. N. (2014). *Quantum Chemistry* (7th ed.). Pearson.

Lewars, E. G. (2016). *Computational Chemistry: Introduction to the Theory and Applications of Molecular and Quantum Mechanics* (3rd ed.). Springer.

Linstrom, P. J., & Mallard, W. G. (Eds.). (2024). *NIST Chemistry WebBook, NIST Standard Reference Database Number 69*. National Institute of Standards and Technology.

McKinney, W. (2010). Data structures for statistical computing in Python. In S. van der Walt & J. Millman (Eds.), *Proceedings of the 9th Python in Science Conference* (pp. 56–61).

McQuarrie, D. A., & Simon, J. D. (1997). *Physical Chemistry: A Molecular Approach*. University Science Books.

Neese, F. (2012). The ORCA program system. *WIREs Computational Molecular Science*, 2(1), 73–78.

Neese, F. (2022). Software update: The ORCA program system—Version 5.0. *WIREs Computational Molecular Science*, 12(5), e1606.

O’Boyle, N. M., Tenderholt, A. L., & Langner, K. M. (2008). cclib: A library for package-independent computational chemistry algorithms. *Journal of Computational Chemistry*, 29(5), 839–845.

Parrish, R. M., Burns, L. A., Smith, D. G. A., Simmonett, A. C., DePrince, A. E., Bozkaya, U., Sokolov, A. Y., Di Remigio, R., Richard, R. M., Gonthier, J. F., James, A. M., McAlexander, H. R., Kumar, A., Saitow, M., Wang, X., Pritchard, B. P., Verma, P., Schaefer, H. F., Patkowski, K., King, R. A., Valeev, E. F., Evangelista, F. A., Turney, J. M., Crawford, T. D., & Sherrill, C. D. (2017). Psi4 1.1: An open-source electronic structure program emphasizing automation, advanced libraries, and interoperability. *Journal of Chemical Theory and Computation*, 13(7), 3185–3197.

Python Software Foundation. (2009). *Python 3 Reference Manual*. CreateSpace.

Python Software Foundation. (n.d.). *The Python Standard Library: re—Regular Expression Operations*. Python Software Foundation.

Sandve, G. K., Nekrutenko, A., Taylor, J., & Hovig, E. (2013). Ten simple rules for reproducible computational research. *PLOS Computational Biology*, 9(10), e1003285.

Schmidt, M. W., Baldridge, K. K., Boatz, J. A., Elbert, S. T., Gordon, M. S., Jensen, J. H., Koseki, S., Matsunaga, N., Nguyen, K. A., Su, S., Windus, T. L., Dupuis, M., & Montgomery, J. A. (1993). General atomic and molecular electronic structure system. *Journal of Computational Chemistry*, 14(11), 1347–1363.

Stiggins, R. J. (2005). From formative assessment to assessment for learning: A path to success in standards-based schools. *Phi Delta Kappan*, 87(4), 324–328.

Sun, Q., Berkelbach, T. C., Blunt, N. S., Booth, G. H., Guo, S., Li, Z., Liu, J., McClain, J. D., Sayfutyarova, E. R., Sharma, S., Wouters, S., & Chan, G. K.-L. (2018). PySCF: The Python-based simulations of chemistry framework. *WIREs Computational Molecular Science*, 8(1), e1340.

Sun, Q., Zhang, X., Banerjee, S., Bao, P., Barbry, M., Blunt, N. S., Bogdanov, N. A., Booth, G. H., Chen, J., Cui, Z.-H., Eriksen, J. J., Gao, Y., Guo, S., Hermann, J., Hermes, M. R., Koh, K., Koval, P., Lehtola, S., Li, Z., Liu, J., Mardirossian, N., McClain, J. D., Motta, M., Mussard, B., Pham, H. Q., Pulkin, A., Purwanto, W., Robinson, P. J., Ronca, E., Sayfutyarova, E. R., Scheurer, M., Schurkus, H. F., Smith, J. E. T., Sun, C., Sun, S.-N., Upadhyay, S., Wagner, L. K., Wang, X., White, A., Whitfield, J. D., Williamson, M. J., Wouters, S., Yang, J., Yu, J. M., Zhu, T., Berkelbach, T. C., Sharma, S., Sokolov, A. Y., & Chan, G. K.-L. (2020). Recent developments in the PySCF program package. *The Journal of Chemical Physics*, 153(2), 024109.

Szabo, A., & Ostlund, N. S. (1996). *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory*. Dover Publications.

The pandas development team. (2020). *pandas-dev/pandas: Pandas*. Zenodo.

Thomas, J. W. (2000). *A Review of Research on Project-Based Learning*. Autodesk Foundation.

VanderPlas, J. (2016). *Python Data Science Handbook: Essential Tools for Working with Data*. O'Reilly Media.

Wiggins, G. (1998). *Educative Assessment: Designing Assessments to Inform and Improve Student Performance*. Jossey-Bass.

Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., da Silva Santos, L. B., Bourne, P. E., Bouwman, J., Brookes, A. J., Clark, T., Crosas, M., Dillo, I., Dumon, O., Edmunds, S., Evelo, C. T., Finkers, R., Gonzalez-Beltran, A., Gray, A. J. G., Groth, P., Goble, C., Grethe, J. S., Heringa, J., 't Hoen, P. A. C., Hooft, R., Kuhn, T., Kok, R., Kok, J., Lusher, S. J., Martone, M. E., Mons, A., Packer, A. L., Persson, B., Rocca-Serra, P., Roos, M., van Schaik, R., Sansone, S.-A., Schultes, E., Sengstag, E., Slater, T., Strawn, G., Swertz, M. A., Thompson, M., van der Lei, J., van Mulligen, E., Velterop, J., Waagmeester, A., Wittenburg, P., Wolstencroft, K., Zhao, J., & Mons, B. (2016). The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*, 3, 160018.

Wilson, G., Aruliah, D. A., Brown, C. T., Chue Hong, N. P., Davis, M., Guy, R. T., Haddock, S. H. D., Huff, K. D., Mitchell, I. M., Plumbley, M. D., Waugh, B., White, E. P., & Wilson, P. (2014). Best practices for scientific computing. *PLOS Biology*, 12(1), e1001745.

Young, D. C. (2001). *Computational Chemistry: A Practical Guide for Applying Techniques to Real-World Problems*. John Wiley & Sons.

LAMPIRAN 1

CATATAN PENTING, STANDAR TEKNIS, DAN CHECKLIST PRAKTIKUM KIMIA KUANTUM

A. Tujuan Lampiran

Lampiran ini disusun untuk melengkapi bagian utama buku Praktikum Kimia Kuantum yang telah membahas proyek mini, proposal, laporan, evaluasi akhir, glosarium, dan lampiran kode Python. Bagian utama buku sudah menjelaskan konsep, alur praktikum, validasi output, visualisasi grafik, laporan, dan kode pendukung. Namun, dalam pelaksanaan praktikum sehari-hari, mahasiswa masih memerlukan panduan teknis tambahan yang bersifat praktis, operasional, dan mudah diperiksa.

Lampiran 1 ini memuat hal-hal penting yang sering diperlukan dalam praktikum tetapi belum dibahas secara khusus dalam bagian utama, antara lain:

1. Standar organisasi folder praktikum.
2. Standar penamaan file.
3. Checklist sebelum menjalankan perhitungan.
4. Checklist setelah perhitungan selesai.
5. Standar satuan dan konversi energi.
6. Standar validasi minimum.
7. Format metadata perhitungan.
8. Panduan troubleshooting singkat.
9. Standar kelengkapan file laporan.
10. Prinsip kejujuran akademik dan reproduibilitas.
11. Template README praktikum.
12. Checklist akhir sebelum pengumpulan.

Dengan lampiran ini, mahasiswa diharapkan dapat bekerja lebih rapi, tertib, teliti, dan bertanggung jawab.

B. Standar Organisasi Folder Praktikum

Setiap mahasiswa dianjurkan menggunakan satu folder utama untuk seluruh pekerjaan praktikum. Folder ini harus memuat data, output, kode, grafik, ringkasan, dan laporan secara terpisah.

Struktur folder yang dianjurkan:

```
praktikum_kimia_kuantum/
```

```

├── 01_input/
│   ├── h2o.xyz
│   ├── co2.xyz
│   └── nh3.xyz
├── 02_output/
│   ├── h2o.log
│   ├── co2.log
│   └── nh3.log
├── 03_data_csv/
│   ├── data_energi.csv
│   ├── data_orbital.csv
│   ├── data_ir.csv
│   └── data_validasi.csv
├── 04_kode_python/
│   ├── parser_output_komputasi.py
│   ├── grafik_laporan.py
│   └── analisis_energi.py
├── 05_grafik/
│   ├── diagram_homo_lumo.png
│   ├── spektrum_ir.png
│   └── diagram_energi_reaksi.png
├── 06_laporan/
│   ├── laporan_proyek_mini.docx
│   └── laporan_proyek_mini.pdf
├── 07_lampiran/
│   ├── potongan_output_energi.txt
│   ├── potongan_output_frekuensi.txt
│   └── kode_python_lengkap.txt
└── README.txt

```

Struktur ini membantu dosen atau instruktur memeriksa pekerjaan mahasiswa dengan cepat. Mahasiswa juga lebih mudah menemukan file jika harus melakukan perbaikan.

C. Standar Penamaan File

Penamaan file harus konsisten. Nama file sebaiknya tidak menggunakan spasi, tanda kurung berlebihan, atau karakter yang membingungkan. Gunakan huruf kecil, angka, dan garis bawah.

Contoh penamaan yang dianjurkan:

```

h2o_opt.log
h2o_freq.log
co2_opt.log

```

```
benzena_homo_lumo.csv
formaldehida_ir.csv
dimer_air_energi.csv
grafik_gap_homo_lumo.png
laporan_proyek_mini_homo_lumo.docx
```

Contoh penamaan yang kurang baik:

```
data saya final banget terbaru fix.csv
laporan akhir revisi baru lagi.docx
output(1)(2)(final).log
Molecule Calculation Result New Copy.txt
```

Penamaan file yang buruk dapat menyebabkan file tertukar, sulit dicari, atau salah dimasukkan ke laporan.

D. Checklist Sebelum Menjalankan Perhitungan

Sebelum menjalankan perhitungan, mahasiswa wajib memeriksa hal-hal berikut.

No.	Hal yang Diperiksa	Status
1	Molekul target sudah benar	Ya/Tidak
2	Struktur awal tidak rusak	Ya/Tidak
3	Tidak ada atom yang saling bertumpuk	Ya/Tidak
4	Semua atom hidrogen lengkap	Ya/Tidak
5	Charge sudah benar	Ya/Tidak
6	Multiplicity sudah benar	Ya/Tidak
7	Metode komputasi sudah dipilih	Ya/Tidak
8	Basis set sudah dipilih	Ya/Tidak
9	Jenis job sudah sesuai	Ya/Tidak
10	Model pelarut sudah ditentukan jika diperlukan	Ya/Tidak
11	Nama file input sudah jelas	Ya/Tidak
12	Tujuan data sudah diketahui	Ya/Tidak

Jika salah satu bagian penting belum terpenuhi, perhitungan sebaiknya tidak dijalankan terlebih dahulu.

E. Checklist Setelah Perhitungan Selesai

Setelah output diperoleh, mahasiswa harus memeriksa validitas hasil sebelum data dimasukkan ke laporan.

No.	Kriteria Validasi	Keterangan	Status
1	Job selesai normal	Output tidak berhenti karena error	Ya/Tidak
2	SCF konvergen	Energi elektronik stabil	Ya/Tidak
3	Optimasi konvergen	Struktur mencapai kriteria optimasi	Ya/Tidak
4	Frekuensi imajiner tidak ada	Untuk struktur minimum	Ya/Tidak
5	Geometri akhir masuk akal	Dicek visual dan logis	Ya/Tidak
6	Charge benar	Sesuai spesies kimia	Ya/Tidak
7	Multiplicity benar	Sesuai keadaan spin	Ya/Tidak
8	Energi terbaca benar	Satuan dan posisi output benar	Ya/Tidak
9	Data dapat dibandingkan	Metode dan basis set konsisten	Ya/Tidak
10	Data layak dianalisis	Keputusan akhir validasi	Ya/Tidak

Data yang belum lolos validasi tidak boleh langsung digunakan untuk menyusun kesimpulan.

F. Standar Satuan Penting

Dalam laporan kimia komputasi, satuan harus ditulis dengan jelas. Kesalahan satuan dapat menyebabkan kesimpulan salah.

Besaran	Satuan Umum	Keterangan
Energi total	Hartree	Umum dalam output komputasi
Energi relatif	kJ/mol	Umum dalam laporan kimia
Energi orbital	eV	Umum untuk HOMO-LUMO
Frekuensi vibrasi	cm ⁻¹	Umum untuk spektrum IR
Momen dipol	Debye	Umum untuk polaritas
Panjang ikatan	Å	Angstrom
Sudut ikatan	derajat	Untuk geometri molekul
Muatan	satuan muatan elementer	Charge total atau muatan parsial
Suhu	K	Kelvin

Konversi energi penting:

1 Hartree $\approx$ 2625,5 kJ/mol
 1 Hartree $\approx$ 627,5 kcal/mol
 1 Hartree $\approx$ 27,2114 eV

Contoh konversi energi relatif:

$$\Delta E \text{ (kJ/mol)} = \Delta E \text{ (Hartree)} \times 2625,5$$

Jika energi dua spesies adalah:

E_produk = -100,0200 Hartree
E_reaktan = -100,0000 Hartree

maka:

$\Delta E = -100,0200 - (-100,0000)$
 $\Delta E = -0,0200 \text{ Hartree}$
 $\Delta E = -0,0200 \times 2625,5$
 $\Delta E = -52,51 \text{ kJ/mol}$

G. Standar Minimal Validasi Struktur Minimum

Untuk struktur yang diklaim sebagai minimum lokal, kriteria minimalnya adalah:

1. Job selesai normal.
2. Optimasi geometri konvergen.
3. Tidak ada frekuensi imajiner.
4. Geometri akhir masuk akal.
5. Charge dan multiplicity benar.

Jika struktur memiliki frekuensi imajiner, struktur belum boleh langsung disebut minimum.

Aturan sederhana:

0 frekuensi imajiner
 → struktur minimum lokal

1 frekuensi imajiner
 → kemungkinan keadaan transisi, jika sesuai koordinat reaksi

lebih dari 1 frekuensi imajiner
 → struktur belum stabil atau perlu optimasi ulang

Dalam praktikum dasar, sebagian besar struktur yang dihitung adalah minimum. Oleh karena itu, keberadaan frekuensi imajiner harus dianggap sebagai tanda peringatan.

H. Format Metadata Perhitungan

Setiap perhitungan sebaiknya memiliki metadata. Metadata adalah catatan ringkas yang menjelaskan informasi teknis suatu job.

Format metadata:

Nama file:
 Nama molekul:
 Rumus molekul:

Charge:
 Multiplicity:
 Software:
 Metode:
 Basis set:
 Jenis job:
 Model pelarut:
 Tanggal perhitungan:
 Status job:
 Energi total:
 Momen dipol:
 Jumlah frekuensi imajiner:
 Catatan validasi:

Contoh:

Nama file: formaldehida_freq.log
 Nama molekul: Formaldehida
 Rumus molekul: CH₂O
 Charge: 0
 Multiplicity: 1
 Software: ORCA
 Metode: DFT/B3LYP
 Basis set: 6-31G(d)
 Jenis job: Opt Freq
 Model pelarut: Gas
 Tanggal perhitungan: 12 Mei 2026
 Status job: Selesai normal
 Energi total: -114,xxxx Hartree
 Momen dipol: x,xx Debye
 Jumlah frekuensi imajiner: 0
 Catatan validasi: Struktur valid sebagai minimum lokal

Metadata membantu reproduktibilitas dan pemeriksaan laporan.

I. Panduan Troubleshooting Singkat

1. SCF Tidak Konvergen

Kemungkinan penyebab:

1. Struktur awal buruk.
2. Charge salah.
3. Multiplicity salah.
4. Molekul terlalu reaktif.
5. Sistem memiliki keadaan elektronik sulit.
6. Basis set atau metode kurang sesuai.

Solusi awal:

Perbaiki struktur awal.
 Periksa charge dan multiplicity.
 Gunakan optimasi awal dengan metode lebih sederhana.
 Tambah jumlah siklus SCF jika software memungkinkan.
 Gunakan tebakan awal orbital yang lebih stabil jika tersedia.

2. Optimasi Geometri Gagal

Kemungkinan penyebab:

1. Struktur awal terlalu jauh dari bentuk stabil.
2. Atom bertumpuk.
3. Molekul pecah selama optimasi.
4. Konvergensi terlalu ketat.
5. Permukaan energi sulit.

Solusi awal:

Perbaiki struktur awal secara visual.
 Lakukan pre-optimasi dengan medan gaya.
 Gunakan metode lebih sederhana terlebih dahulu.
 Periksa apakah molekul sudah lengkap.
 Periksa ikatan dan orientasi gugus fungsi.

3. Frekuensi Imajiner Muncul

Kemungkinan penyebab:

1. Struktur belum minimum.
2. Optimasi berhenti terlalu cepat.
3. Struktur berada pada saddle point.
4. Konformer tidak stabil.
5. Ada mode rotasi atau translasi yang bermasalah.

Solusi awal:

Lihat mode vibrasi imajiner.
 Geser struktur mengikuti arah mode tersebut.
 Optimasi ulang.
 Pastikan optimasi benar-benar konvergen.

4. Grafik Tidak Sesuai Data

Kemungkinan penyebab:

1. CSV salah.
2. Kolom tertukar.
3. Satuan belum dikonversi.
4. Data contoh belum diganti.
5. Label grafik salah.
6. File yang dibaca bukan file terbaru.

Solusi awal:

Buka CSV dan periksa datanya.
Pastikan nama kolom sesuai kode.
Periksa satuan.
Jalankan ulang kode.
Hapus grafik lama jika perlu.

J. Standar Kelengkapan File yang Dikumpulkan

Saat mengumpulkan laporan akhir, mahasiswa sebaiknya mengumpulkan satu folder yang berisi:

No.	File/Folder	Wajib/Opsional
1	Laporan DOCX/PDF	Wajib
2	File input	Wajib
3	File output	Wajib
4	CSV data	Wajib
5	Kode Python	Wajib
6	Grafik	Wajib
7	Potongan output penting	Wajib
8	README	Wajib
9	File presentasi	Jika ada
10	Data literatur	Jika digunakan

Laporan tanpa file pendukung sulit diverifikasi.

K. Prinsip Kejujuran Akademik

Dalam praktikum kimia komputasi, kejujuran akademik sangat penting. Mahasiswa harus menjaga agar semua data yang ditulis berasal dari hasil kerja yang benar.

Hal yang tidak boleh dilakukan:

1. Mengarang energi.
2. Mengubah angka agar sesuai teori.
3. Menghapus output gagal tanpa catatan.

4. Menggunakan data teman tanpa izin dan tanpa keterangan.
5. Mengklaim kode sendiri padahal menyalin tanpa memahami.
6. Menampilkan grafik dari data contoh sebagai data praktikum.
7. Menghilangkan informasi bahwa output tidak konvergen.
8. Menulis metode berbeda dari yang benar-benar digunakan.

Hal yang harus dilakukan:

1. Menulis data sesuai output.
2. Menyebutkan jika ada error.
3. Melampirkan file input-output.
4. Menjelaskan keterbatasan.
5. Mengutip sumber literatur.
6. Menyimpan CSV dan kode Python.
7. Menulis kesimpulan sesuai bukti.

Kejujuran akademik adalah bagian dari validitas ilmiah.

L. Prinsip Reprodusibilitas

Reprodusibilitas berarti hasil dapat diperiksa dan diulang oleh orang lain. Dalam praktikum kimia komputasi, reprodusibilitas bergantung pada kelengkapan informasi.

Agar hasil reprodusibel, laporan harus mencantumkan:

```
software
metode
basis set
charge
multiplicity
jenis job
model pelarut
file input
file output
data CSV
kode Python
grafik
catatan validasi
```

Jika salah satu bagian penting hilang, hasil menjadi sulit diperiksa.

M. Template README Praktikum

Setiap folder proyek mini sebaiknya memiliki file `README.txt`.

Contoh isi README:

README PROYEK MINI PRAKTIKUM KIMIA KUANTUM

Judul proyek:

Analisis HOMO-LUMO Benzena, Anilina, dan Nitrobenzena

Nama mahasiswa:

.....

NIM:

.....

Deskripsi singkat:

Proyek ini menganalisis energi HOMO, LUMO, dan gap HOMO-LUMO pada senyawa aromatik tersubstitusi.

Software:

.....

Metode:

.....

Basis set:

.....

Jenis job:

.....

Daftar file penting:

1. input/benzena.xyz
2. output/benzena.log
3. data/data_orbital.csv
4. kode/praktikum_04_dft_homo_lumo.py
5. grafik/diagram_homo_lumo.png
6. laporan/laporan_proyek_mini.docx

Cara menjalankan kode:

python kode/praktikum_04_dft_homo_lumo.py

Catatan validasi:

Semua output selesai normal dan tidak menunjukkan frekuensi imajiner untuk struktur minimum.

Catatan keterbatasan:

Analisis HOMO-LUMO digunakan secara kualitatif dan tidak menjadi satu-satunya ukuran reaktivitas.

README membantu dosen memahami isi folder tanpa membuka semua file satu per satu.

N. Checklist Akhir Sebelum Pengumpulan

Sebelum laporan dikumpulkan, mahasiswa harus memeriksa checklist berikut.

No.	Pemeriksaan Akhir	Status
1	Judul laporan sesuai proyek	Ya/Tidak
2	Abstrak sudah memuat tujuan, metode, hasil, kesimpulan	Ya/Tidak
3	Pendahuluan memiliki rumusan masalah	Ya/Tidak
4	Dasar teori relevan	Ya/Tidak
5	Metode, basis set, charge, multiplicity ditulis lengkap	Ya/Tidak
6	Output sudah divalidasi	Ya/Tidak
7	Tabel memiliki satuan	Ya/Tidak
8	Grafik memiliki judul, sumbu, dan satuan	Ya/Tidak
9	Pembahasan tidak hanya mengulang tabel	Ya/Tidak
10	Kesimpulan menjawab rumusan masalah	Ya/Tidak
11	Keterbatasan disebutkan	Ya/Tidak
12	Daftar pustaka relevan	Ya/Tidak
13	File input dilampirkan	Ya/Tidak
14	File output dilampirkan	Ya/Tidak
15	CSV dilampirkan	Ya/Tidak
16	Kode Python dilampirkan	Ya/Tidak
17	Grafik dilampirkan	Ya/Tidak
18	README tersedia	Ya/Tidak
19	Tidak ada data contoh yang tertinggal	Ya/Tidak
20	Semua file dinamai rapi	Ya/Tidak

Jika masih ada jawaban “Tidak” pada bagian penting, laporan perlu diperbaiki sebelum dikumpulkan.

O. Ringkasan Lampiran 1

Lampiran 1 menyajikan panduan teknis penting untuk memastikan praktikum kimia kuantum berjalan tertib, valid, dan reproduibel. Bagian ini melengkapi bab utama dengan standar organisasi folder, penamaan file, checklist sebelum dan sesudah perhitungan, standar satuan, metadata, troubleshooting, kelengkapan file, kejujuran akademik, reproduibilitas, template README, dan checklist akhir.

Inti Lampiran 1 dapat diringkas sebagai berikut:

kerja rapi

- file tertata
- input benar
- output valid
- data bersatuan
- kode terdokumentasi
- grafik jelas
- laporan reproduibel
- kesimpulan dapat dipertanggungjawabkan

Dengan mengikuti lampiran ini, mahasiswa diharapkan tidak hanya mampu menjalankan praktikum, tetapi juga mampu menjaga mutu ilmiah seluruh proses dan hasil praktikum.

LAMPIRAN 2 PELENGKAP

A. Pemetaan 7 Mata Praktikum

Praktikum	Topik	Jalur Software	Jalur Python	Output Utama
1	Partikel dalam kotak	ChemCompute Jupyter / Colab	Simulasi ψ , ψ^2 , dan E_n	Grafik fungsi gelombang dan energi
2	Optimasi geometri	ChemCompute, Avogadro, GAMESS, ORCA	Analisis XYZ, jarak, sudut	Geometri akhir dan tabel ikatan
3	Hartree-Fock dan basis set	ChemCompute, GAMESS, ORCA, Gaussian	Grafik energi vs basis set	Energi total dan biaya komputasi
4	DFT dan HOMO-LUMO	ChemCompute, ORCA, Gaussian, Molden	Analisis gap dan reaktivitas	HOMO, LUMO, gap, diagram energi
5	Simetri dan momen dipol	ChemCompute, Jmol, Avogadro	Vektor dipol dari muatan parsial	Point group dan polaritas
6	Frekuensi vibrasi dan IR	ChemCompute, GAMESS, ORCA, Gaussian	Spektrum IR simulasi	Frekuensi, intensitas, spektrum
7	Energi reaksi dan pelarut	ChemCompute, ORCA, Gaussian, Psi4	ΔE , ΔH , ΔG , diagram energi	Termodinamika reaksi

B. Standar Kode Python pada Setiap Praktikum

Setiap kode Python dalam buku harus memenuhi standar berikut:

1. Dapat dijalankan di Python 3.10 atau lebih baru.
2. Menggunakan pustaka umum: numpy, pandas, matplotlib, scipy jika diperlukan.
3. Tidak memerlukan lisensi berbayar.
4. Memiliki komentar yang jelas.
5. Menghasilkan output tabel dan grafik.
6. Menyimpan hasil ke CSV atau PNG.
7. Dapat digunakan di Jupyter Notebook, Google Colab, atau komputer lokal.
8. Menggunakan satuan yang jelas.
9. Menampilkan kesimpulan numerik sederhana.
10. Disertai instruksi instalasi pustaka.

C. Alur Logis Antarbab

1. **Bab 1** memperkenalkan hakikat praktikum kimia kuantum.
2. **Bab 2** memperkenalkan ekosistem software, terutama ChemCompute sebagai laboratorium komputasi berbasis web.
3. **Bab 3** memberi dasar Python agar mahasiswa mampu membuat simulasi dan analisis mandiri.
4. **Bab 4** memulai dari model kuantum paling sederhana: partikel dalam kotak.
5. **Bab 5** masuk ke molekul nyata melalui optimasi geometri.
6. **Bab 6** memperkenalkan Hartree–Fock dan basis set.
7. **Bab 7** memperluas ke DFT dan HOMO-LUMO.
8. **Bab 8** menghubungkan struktur molekul dengan simetri dan momen dipol.
9. **Bab 9** menghubungkan struktur hasil optimasi dengan spektrum IR.
10. **Bab 10** mengintegrasikan energi, termodinamika, dan efek pelarut.
11. **Bab 11** mengajarkan validasi output dan otomatisasi laporan dengan Python.
12. **Bab 12** menutup dengan proyek mini, evaluasi akhir, glosarium, dan lampiran kode.

D. Keunggulan

Buku ini memiliki nilai tambah sebagai berikut:

1. Lebih praktis karena dapat dijalankan melalui ChemCompute tanpa instalasi rumit.
2. Lebih inovatif karena setiap praktikum memiliki kode Python sendiri.
3. Lebih mendidik karena mahasiswa memahami logika perhitungan, bukan hanya menekan tombol software.
4. Lebih fleksibel karena dapat digunakan di laboratorium komputer, laptop pribadi, Google Colab, atau platform web.
5. Lebih reproduibel karena setiap hasil praktikum dapat dianalisis ulang dengan Python.
6. Lebih sesuai untuk pembelajaran modern berbasis data, pemodelan, dan computational thinking.

E. Catatan Pelaksanaan Praktikum

Setiap mata praktikum sebaiknya dilaksanakan dalam dua tahap:

Tahap 1 — Praktikum Software

Mahasiswa menjalankan perhitungan menggunakan ChemCompute atau software kimia komputasi lain.

Tahap 2 — Praktikum Python

Mahasiswa mengolah, memvisualisasikan, dan menafsirkan data hasil praktikum menggunakan kode Python.

Dengan pola ini, mahasiswa memperoleh dua kompetensi sekaligus: **kompetensi kimia komputasi** dan **kompetensi scientific programming**.

- KASMUI -
Dosen Kimia, Komputasi, IT, AI UNNES

<https://hisabmu.com/>
<https://kasmui.cloud/>

Ketua PCM
Gunungpati 2

Anggota Majelis Tabligh
PDM Kota Semarang
& PWM Jawa Tengah

Tim Pengembang
Software KHGT
MTT PP Muhammadiyah

Praktisi
Ilmu Falak

Semarang, Juni 2026

Penulis,

Kasmui

PRAKTIKUM KIMIA KUANTUM

$$\hat{H}\psi = E\psi$$

— Bagian 3: Membuat Laporan & Penutup —

$$E_n = -\frac{R_H}{n^2}$$

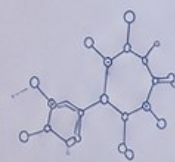
SINOPSIS

Buku PRAKTIKUM KIMIA KUANTUM Bagian 3: Membuat Laporan & Penutup membimbing mahasiswa menyelesaikan tahap akhir praktikum kimia kuantum secara sistematis, ilmiah, dan reproduibel. Bagian ini membahas cara membaca dan memvalidasi output komputasi, mengolah data dengan Python, membuat tabel dan grafik, serta menyusun laporan praktikum yang rapi, kritis, dan bertanggung jawab.

Selain itu, buku ini memuat proyek mini, evaluasi akhir, glosarium istilah penting, serta lampiran kode Python dan panduan teknis pelaporan. Dengan pendekatan ini, mahasiswa tidak hanya mampu menjalankan perhitungan, tetapi juga mampu menafsirkan hasil, menyajikannya dalam laporan yang baik, dan mempertanggungjawabkannya secara akademik.

LAPORAN PRAKTIKUM KIMIA KUANTUM

- ✓ Analisis Data
- ✓ Interpretasi Hasil
- ✓ Pembahasan
- ✓ Kesimpulan
- ✓ Penutup



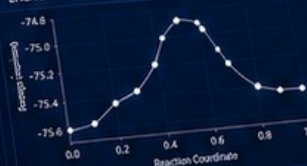
QUANTUM CHEMISTRY ANALYSIS



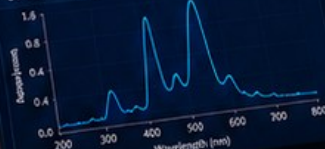
CALCULATION LOG

```
> B3LYP/6-31G(d) OPT FREQ
> SCF Converged.
> E[total] = -75.123456 Hartree
> Zero-point correction = 0.313456
> Thermal correction = 0.345678
> Gibbs free energy = -74.987654
> Normal termination of Gaussian 16
```

ENERGY PROFILE



UV-VIS SPECTRUM



Catatan Analisis

- Optimasi Geometri
- Energi & Stabilitas
- Spektrum UV-Vis
- Interpretasi Orbital



Kesimpulan

Berdasarkan hasil perhitungan dan analisis yang dilakukan, mekanisme reaksi dan sifat molekul telah dipahami dengan baik melalui pendekatan kimia kuantum.

Selesai.

KASMUI