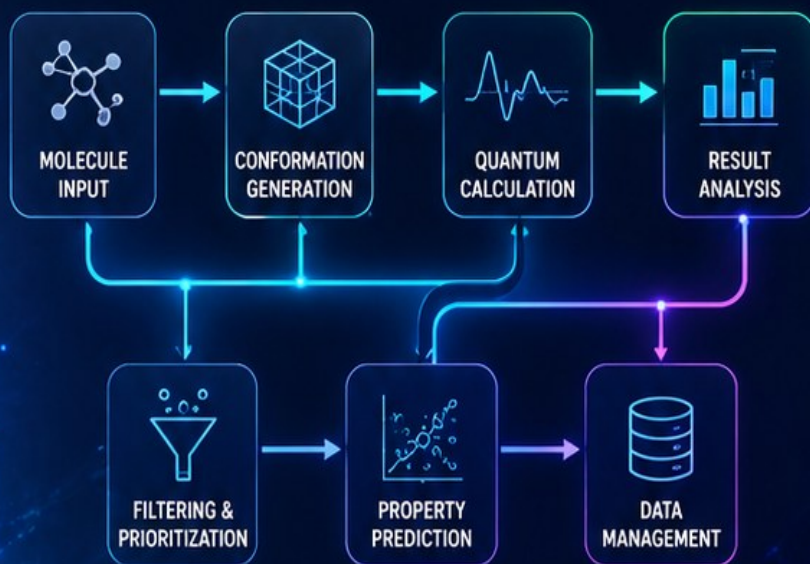
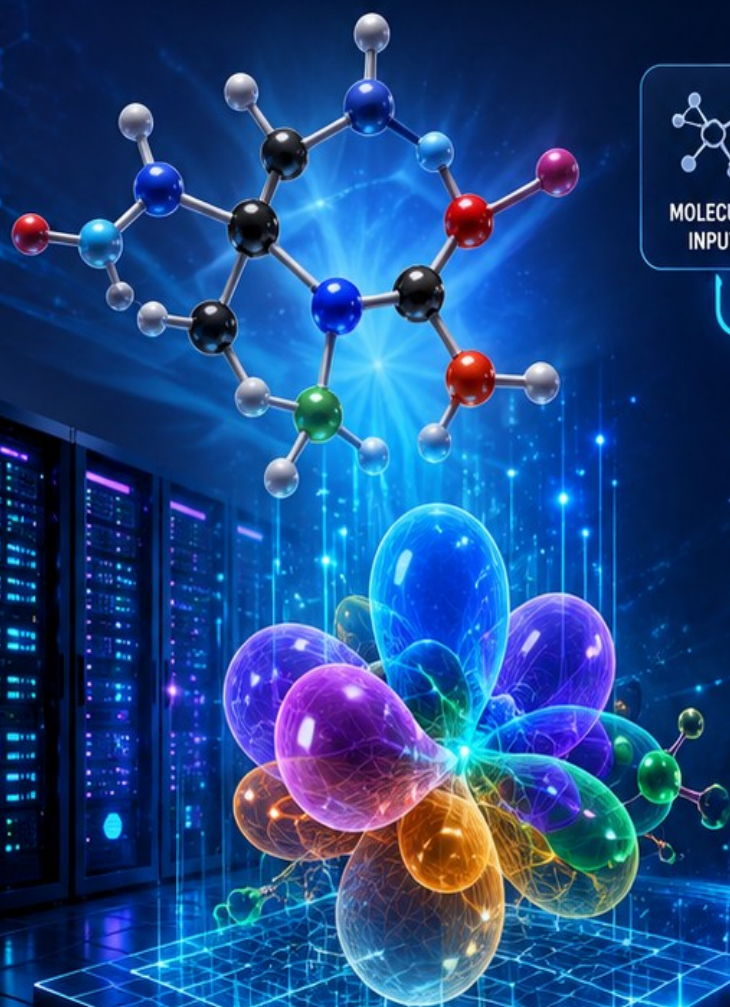


# APLIKASI QMSCREEN

High-Throughput Quantum Chemistry & Workflow Manager

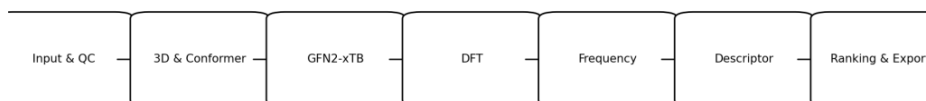


KAS MUI

# APLIKASI QMSCREEN

## HIGH-THROUGHPUT QUANTUM CHEMISTRY & WORKFLOW MANAGER

### Alur High-Throughput QMScreen



Setiap tahap menyimpan metadata, status, dan provenance.

**Kasmui**

2026

## INFORMASI BUKU

**Tabel 0.1. Informasi bibliografis dan ruang lingkup**

| Komponen              | Keterangan                                                                                                                                                               |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Judul                 | APLIKASI QMSCREEN                                                                                                                                                        |
| Subjudul              | High-Throughput Quantum Chemistry & Workflow Manager                                                                                                                     |
| Basis implementasi    | QMScreen V1.0.5                                                                                                                                                          |
| Penyusun/Pengembangan | Kasmui                                                                                                                                                                   |
| Tahun                 | 2026                                                                                                                                                                     |
| Fokus                 | Workflow high-throughput: input/QC, conformer, GFN2-xTB, PySCF/DFT, frequency/thermochemistry, descriptor, queue, cache, ranking, ekspor, benchmark, dan provenance      |
| Catatan               | Buku menjelaskan penggunaan dan konsep implementasi. Angka hasil energi atau descriptor studi kasus tidak dibuat-buat; nilai numerik harus berasal dari kalkulasi nyata. |

Hak cipta perangkat lunak yang menjadi basis buku tercantum pada antarmuka QMScreen sebagai Copyright © Kasmui, 2026. All Rights Reserved. Buku ini disusun sebagai dokumentasi teknis dan bahan pembelajaran untuk memahami prinsip ilmiah, alur komputasi, quality control, serta penggunaan aplikasi secara bertanggung jawab.

Nama pustaka, metode, dan perangkat lunak pihak ketiga tetap menjadi milik dan tanggung jawab pemegang hak masing-masing. Pengguna perlu mengikuti lisensi, dokumentasi, dan ketentuan sitasi perangkat lunak yang digunakan dalam penelitian.

## KATA PENGANTAR

Perkembangan kimia komputasi membuat jumlah struktur yang dapat dianalisis meningkat sangat cepat. Di satu sisi, ketersediaan perangkat lunak ilmiah seperti RDKit, xTB/GFN2-xTB, PySCF, dan geomeTRIC memberi peluang untuk membangun alur kerja yang semakin otomatis. Di sisi lain, otomasi yang tidak disertai validasi dapat memperbanyak kesalahan dengan kecepatan yang sama. Karena itu, buku APLIKASI QMSCREEN disusun untuk menempatkan otomasi, quality control, keterlacakan data, dan interpretasi ilmiah dalam satu kerangka kerja yang saling berhubungan.

QMScreen dirancang sebagai high-throughput quantum chemistry dan workflow manager. Istilah high-throughput di sini tidak sekadar berarti menjalankan banyak job secara paralel. Maknanya lebih luas: input harus dapat diterima secara massal, struktur harus dinormalisasi, duplikat perlu dikenali, charge dan multiplicity harus diperiksa, conformer perlu disiapkan, metode cepat dapat dipakai sebagai prescreen, dan kandidat yang lolos dapat diteruskan ke DFT. Pada setiap tahap, status harus tercatat sehingga pengguna mengetahui apakah sebuah nilai benar-benar tervalidasi atau hanya merupakan hasil parsial.

Buku ini mengikuti implementasi QMScreen V1.0.5 dan memecah fungsi aplikasi menjadi bab-bab yang dapat dipelajari secara bertahap. Pembaca tidak hanya diperkenalkan pada tombol dan menu, tetapi juga pada alasan mengapa suatu pilihan tersedia, risiko ilmiah yang harus dihindari, struktur data yang disimpan, serta bagaimana sebuah keputusan dapat ditelusuri kembali melalui manifest, log, checkpoint, database SQLite, cache, dan environment snapshot. Pendekatan tersebut dimaksudkan agar aplikasi tidak menjadi kotak hitam.

Salah satu prinsip penting yang ditekankan berulang kali adalah bahwa parameter ilmiah harus eksplisit. Functional, basis set, charge, multiplicity, toleransi SCF, jumlah conformer, penggunaan GFN2-xTB, frequency, temperatur, tekanan, dan resource komputasi harus tercatat sebagai bagian dari spesifikasi job. Dengan demikian, dua hasil yang berbeda dapat dibandingkan berdasarkan kondisi kalkulasinya, bukan hanya berdasarkan nama file atau ingatan operator. Prinsip ini juga mendukung reproducible computational research.

Buku ini sengaja memisahkan pembahasan konsep dengan tutorial penggunaan. Bab-bab awal dan tengah menjelaskan data, metode, validasi, serta arsitektur workflow. Bab tutorial kemudian menerapkan konsep tersebut pada contoh sederhana Ethanol sehingga instalasi dan pipeline dapat diuji dengan biaya rendah. Setelah itu, studi kasus menggunakan lima struktur yang memang terdapat pada katalog upload\_molekul.json, yaitu 8a, 8c, 8s, 8w, dan 8v. Studi kasus tersebut tidak mengarang nilai energi; fokusnya adalah desain batch, pemeriksaan

input, serta cara membandingkan descriptor secara benar apabila kalkulasi telah selesai.

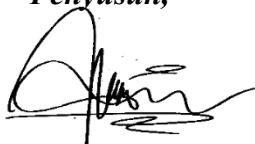
Bagian evaluasi disediakan agar pembaca dapat menilai pemahamannya sendiri. Dua puluh soal pilihan ganda, latihan audit provenance, rancangan workflow, serta proyek mini pengembangan perangkat lunak diarahkan untuk menghubungkan pengetahuan kimia komputasi dengan praktik rekayasa perangkat lunak ilmiah. Pembaca diharapkan mampu menjelaskan tidak hanya bagaimana menjalankan aplikasi, tetapi juga kapan hasil harus ditolak, kapan perlu diberi warning, dan bagaimana membedakan keberhasilan teknis dari validitas ilmiah.

Ucapan terima kasih disampaikan kepada komunitas pengembang perangkat lunak ilmiah terbuka yang memungkinkan ekosistem seperti QMScreen dibangun di atas komponen yang dapat diperiksa, diuji, dan dikembangkan bersama. Dokumentasi resmi dan literatur yang relevan dicantumkan dalam Daftar Pustaka. Pembaca disarankan selalu memeriksa dokumentasi versi perangkat lunak yang benar-benar digunakan karena API, dukungan platform, dan detail numerik dapat berubah dari waktu ke waktu.

Semoga buku ini menjadi dokumentasi yang berguna untuk pendidikan, riset, pengembangan software kimia komputasi, dan terutama pembentukan kebiasaan kerja yang lebih tertib. Kecepatan komputasi memiliki nilai ilmiah apabila disertai metodologi yang benar, data yang dapat dilacak, interpretasi yang proporsional, serta kesiapan untuk mengulang dan mengaudit kembali seluruh proses ketika diperlukan.

*Semarang, Agustus 2026*

*Penyusun,*



*Kasmui*

## PETUNJUK PENGGUNAAN BUKU

Buku dapat dibaca secara linear atau digunakan sebagai manual rujukan. Pembaca yang baru mengenal QMScreen disarankan mengikuti Bab 1 sampai Bab 14 secara berurutan. Pengguna yang telah memahami kimia komputasi dapat langsung menuju bab sesuai tahap workflow, sedangkan pengembang perangkat lunak dapat memusatkan perhatian pada Bab 11, 16, dan 17.

1. Gunakan Bab 1-3 untuk memahami posisi QMScreen, bentuk input, normalisasi, charge, multiplicity, dan deduplikasi.
2. Gunakan Bab 4-9 untuk memahami scientific pipeline dari conformer sampai descriptor kuantum.
3. Gunakan Bab 10-12 untuk mengelola batch, cache, provenance, ranking, ekspor, dan benchmark.
4. Gunakan Bab 13-14 sebagai manual antarmuka dan tutorial operasional.
5. Gunakan Bab 15 untuk latihan desain studi kasus berbasis lima molekul katalog tanpa mengarang hasil numerik.
6. Gunakan Bab 16-18 untuk troubleshooting, pengembangan kode, evaluasi, dan proyek lanjutan.

Semua perintah, nama menu, dan parameter dalam buku mengikuti implementasi QMScreen V1.0.5 sejauh didukung oleh kode sumber yang menjadi basis dokumentasi. Untuk penelitian, tetap gunakan dokumentasi resmi pustaka ilmiah yang tercantum pada Daftar Pustaka dan catat versi lingkungan yang benar-benar digunakan.

## DAFTAR ISI

|                                                                      |    |
|----------------------------------------------------------------------|----|
| INFORMASI BUKU .....                                                 | 3  |
| KATA PENGANTAR .....                                                 | 4  |
| PETUNJUK PENGGUNAAN BUKU .....                                       | 6  |
| DAFTAR ISI .....                                                     | 7  |
| BAB 1 PENGANTAR QMSCREEN DAN HIGH-THROUGHPUT QUANTUM CHEMISTRY ..... | 19 |
| 1.1 Posisi QMScreen dalam riset kimia komputasi .....                | 19 |
| 1.1.1 Prinsip, parameter, dan interpretasi .....                     | 20 |
| 1.1.2 Checklist implementasi dan quality control .....               | 21 |
| 1.2 Tujuan ilmiah dan operasional .....                              | 21 |
| 1.2.1 Prinsip, parameter, dan interpretasi .....                     | 22 |
| 1.2.2 Checklist implementasi dan quality control .....               | 22 |
| 1.3 Arsitektur multi-fidelity .....                                  | 22 |
| 1.3.1 Prinsip, parameter, dan interpretasi .....                     | 23 |
| 1.3.2 Checklist implementasi dan quality control .....               | 24 |
| 1.4 Scientific guardrails .....                                      | 24 |
| 1.4.1 Prinsip, parameter, dan interpretasi .....                     | 24 |
| 1.4.2 Checklist implementasi dan quality control .....               | 25 |
| 1.5 Ruang lingkup buku .....                                         | 25 |
| 1.5.1 Prinsip, parameter, dan interpretasi .....                     | 26 |
| 1.5.2 Checklist implementasi dan quality control .....               | 26 |
| 1.6 Rangkuman Bab .....                                              | 27 |
| BAB 2 INPUT MASSAL, FORMAT STRUKTUR, DAN KATALOG PERSISTEN .....     | 28 |
| 2.1 Sumber input yang didukung .....                                 | 28 |
| 2.1.1 Prinsip, parameter, dan interpretasi .....                     | 29 |
| 2.1.2 Checklist implementasi dan quality control .....               | 30 |
| 2.2 SMILES dan InChI sebagai representasi teks .....                 | 30 |

|                                                                                             |           |
|---------------------------------------------------------------------------------------------|-----------|
| 2.2.1 Prinsip, parameter, dan interpretasi .....                                            | 31        |
| 2.2.2 Checklist implementasi dan quality control .....                                      | 31        |
| 2.3 Upload file terstruktur .....                                                           | 32        |
| 2.3.1 Prinsip, parameter, dan interpretasi .....                                            | 32        |
| 2.3.2 Checklist implementasi dan quality control .....                                      | 33        |
| 2.4 Katalog upload_molekul.json.....                                                        | 33        |
| 2.4.1 Prinsip, parameter, dan interpretasi .....                                            | 34        |
| 2.4.2 Checklist implementasi dan quality control .....                                      | 34        |
| 2.5 Strategi QC pada tahap input.....                                                       | 35        |
| 2.5.1 Prinsip, parameter, dan interpretasi .....                                            | 35        |
| 2.5.2 Checklist implementasi dan quality control .....                                      | 36        |
| 2.6 Rangkuman Bab .....                                                                     | 36        |
| <b>BAB 3 VALIDASI STRUKTUR, NORMALISASI, CHARGE, MULTIPLICITY,<br/>DAN DEDUPLIKASI.....</b> | <b>37</b> |
| 3.1 Sanitasi dan normalisasi molekul .....                                                  | 37        |
| 3.1.1 Prinsip, parameter, dan interpretasi .....                                            | 38        |
| 3.1.2 Checklist implementasi dan quality control .....                                      | 39        |
| 3.2 Charge dan multiplicity .....                                                           | 39        |
| 3.2.1 Prinsip, parameter, dan interpretasi .....                                            | 39        |
| 3.2.2 Checklist implementasi dan quality control .....                                      | 40        |
| 3.3 Canonical structure key .....                                                           | 40        |
| 3.3.1 Prinsip, parameter, dan interpretasi .....                                            | 41        |
| 3.3.2 Checklist implementasi dan quality control .....                                      | 41        |
| 3.4 Deduplication sebelum queue .....                                                       | 42        |
| 3.4.1 Prinsip, parameter, dan interpretasi .....                                            | 42        |
| 3.4.2 Checklist implementasi dan quality control .....                                      | 43        |
| 3.5 Error taxonomy dan tindakan.....                                                        | 43        |
| 3.5.1 Prinsip, parameter, dan interpretasi .....                                            | 44        |
| 3.5.2 Checklist implementasi dan quality control .....                                      | 44        |
| 3.6 Rangkuman Bab .....                                                                     | 45        |

|                                                                       |    |
|-----------------------------------------------------------------------|----|
| BAB 4 GENERATE 3D, ETKDGV3, MMFF94/UFF, DAN CONFORMER SCREENING ..... | 47 |
| 4.1 Mengapa geometri awal penting .....                               | 47 |
| 4.1.1 Prinsip, parameter, dan interpretasi .....                      | 48 |
| 4.1.2 Checklist implementasi dan quality control .....                | 49 |
| 4.2 ETKDGV3 untuk embedding.....                                      | 49 |
| 4.2.1 Prinsip, parameter, dan interpretasi .....                      | 49 |
| 4.2.2 Checklist implementasi dan quality control .....                | 50 |
| 4.3 Pre-optimasi MMFF94 dan UFF .....                                 | 50 |
| 4.3.1 Prinsip, parameter, dan interpretasi .....                      | 51 |
| 4.3.2 Checklist implementasi dan quality control .....                | 51 |
| 4.4 Seleksi conformer.....                                            | 52 |
| 4.4.1 Prinsip, parameter, dan interpretasi .....                      | 52 |
| 4.4.2 Checklist implementasi dan quality control .....                | 53 |
| 4.5 QC conformer batch .....                                          | 53 |
| 4.5.1 Prinsip, parameter, dan interpretasi .....                      | 53 |
| 4.5.2 Checklist implementasi dan quality control .....                | 54 |
| 4.6 Rangkuman Bab .....                                               | 54 |
| BAB 5 PRESCREEN GFN2-XTB DAN STRATEGI MULTI-FIDELITY.....             | 56 |
| 5.1 Peran GFN2-xTB .....                                              | 56 |
| 5.1.1 Prinsip, parameter, dan interpretasi .....                      | 57 |
| 5.1.2 Checklist implementasi dan quality control .....                | 58 |
| 5.2 TBLite dan fallback CLI .....                                     | 58 |
| 5.2.1 Prinsip, parameter, dan interpretasi .....                      | 58 |
| 5.2.2 Checklist implementasi dan quality control .....                | 59 |
| 5.3 Seleksi kandidat setelah prescreen .....                          | 59 |
| 5.3.1 Prinsip, parameter, dan interpretasi .....                      | 60 |
| 5.3.2 Checklist implementasi dan quality control .....                | 60 |
| 5.4 Failure mode prescreen .....                                      | 61 |
| 5.4.1 Prinsip, parameter, dan interpretasi .....                      | 61 |

|                                                                          |           |
|--------------------------------------------------------------------------|-----------|
| 5.4.2 Checklist implementasi dan quality control .....                   | 62        |
| 5.5 Desain multi-fidelity yang dapat diaudit .....                       | 62        |
| 5.5.1 Prinsip, parameter, dan interpretasi .....                         | 62        |
| 5.5.2 Checklist implementasi dan quality control .....                   | 63        |
| 5.6 Rangkuman Bab .....                                                  | 63        |
| <b>BAB 6 DFT PYSCF, FUNCTIONAL, DAN BASIS SET BERTINGKAT .....</b>       | <b>65</b> |
| 6.1 PySCF sebagai engine elektronik.....                                 | 65        |
| 6.1.1 Prinsip, parameter, dan interpretasi .....                         | 66        |
| 6.1.2 Checklist implementasi dan quality control .....                   | 66        |
| 6.2 Functional yang tersedia.....                                        | 67        |
| 6.2.1 Prinsip, parameter, dan interpretasi .....                         | 67        |
| 6.2.2 Checklist implementasi dan quality control .....                   | 68        |
| 6.3 Basis optimasi dan basis final .....                                 | 69        |
| 6.3.1 Prinsip, parameter, dan interpretasi .....                         | 69        |
| 6.3.2 Checklist implementasi dan quality control .....                   | 70        |
| 6.4 SCF convergence controls.....                                        | 70        |
| 6.4.1 Prinsip, parameter, dan interpretasi .....                         | 71        |
| 6.4.2 Checklist implementasi dan quality control .....                   | 71        |
| 6.5 Density fitting dan resource .....                                   | 72        |
| 6.5.1 Prinsip, parameter, dan interpretasi .....                         | 72        |
| 6.5.2 Checklist implementasi dan quality control .....                   | 73        |
| 6.6 Rangkuman Bab .....                                                  | 73        |
| <b>BAB 7 OPTIMASI GEOMETRI, GEOMETRIC, DAN KONTROL KONVERGENSI .....</b> | <b>74</b> |
| 7.1 Optimasi geometri sebagai masalah numerik.....                       | 74        |
| 7.1.1 Prinsip, parameter, dan interpretasi .....                         | 75        |
| 7.1.2 Checklist implementasi dan quality control .....                   | 76        |
| 7.2 Integrasi geomeTRIC .....                                            | 76        |
| 7.2.1 Prinsip, parameter, dan interpretasi .....                         | 76        |
| 7.2.2 Checklist implementasi dan quality control .....                   | 77        |

|                                                                               |           |
|-------------------------------------------------------------------------------|-----------|
| 7.3 Basis bertingkat selama optimasi .....                                    | 77        |
| 7.3.1 Prinsip, parameter, dan interpretasi .....                              | 78        |
| 7.3.2 Checklist implementasi dan quality control .....                        | 78        |
| 7.4 Optimasi gagal dan recovery .....                                         | 79        |
| 7.4.1 Prinsip, parameter, dan interpretasi .....                              | 79        |
| 7.4.2 Checklist implementasi dan quality control .....                        | 80        |
| 7.5 Pemeriksaan geometri akhir .....                                          | 80        |
| 7.5.1 Prinsip, parameter, dan interpretasi .....                              | 81        |
| 7.5.2 Checklist implementasi dan quality control .....                        | 81        |
| 7.6 Rangkuman Bab .....                                                       | 81        |
| <b>BAB 8 FREQUENCY, THERMOCHEMISTRY, HESSIAN, DAN MINIMUM CHECK</b> .....     | <b>83</b> |
| 8.1 Makna perhitungan frequency .....                                         | 83        |
| 8.1.1 Prinsip, parameter, dan interpretasi .....                              | 84        |
| 8.1.2 Checklist implementasi dan quality control .....                        | 85        |
| 8.2 Minimum lokal dan frekuensi imajiner .....                                | 85        |
| 8.2.1 Prinsip, parameter, dan interpretasi .....                              | 85        |
| 8.2.2 Checklist implementasi dan quality control .....                        | 86        |
| 8.3 Termokimia .....                                                          | 86        |
| 8.3.1 Prinsip, parameter, dan interpretasi .....                              | 87        |
| 8.3.2 Checklist implementasi dan quality control .....                        | 87        |
| 8.4 Failure dan hasil parsial.....                                            | 88        |
| 8.4.1 Prinsip, parameter, dan interpretasi .....                              | 88        |
| 8.4.2 Checklist implementasi dan quality control .....                        | 89        |
| 8.5 Biaya dan strategi penggunaan.....                                        | 89        |
| 8.5.1 Prinsip, parameter, dan interpretasi .....                              | 90        |
| 8.5.2 Checklist implementasi dan quality control .....                        | 90        |
| 8.6 Rangkuman Bab .....                                                       | 90        |
| <b>BAB 9 HOMO-LUMO, DIPOLE, MULLIKEN CHARGE, DAN DESCRIPTOR KUANTUM</b> ..... | <b>92</b> |

|                                                                                  |            |
|----------------------------------------------------------------------------------|------------|
| 9.1 Orbital frontier.....                                                        | 92         |
| 9.1.1 Prinsip, parameter, dan interpretasi .....                                 | 93         |
| 9.1.2 Checklist implementasi dan quality control .....                           | 94         |
| 9.2 Momen dipol .....                                                            | 94         |
| 9.2.1 Prinsip, parameter, dan interpretasi .....                                 | 94         |
| 9.2.2 Checklist implementasi dan quality control .....                           | 95         |
| 9.3 Mulliken population analysis .....                                           | 95         |
| 9.3.1 Prinsip, parameter, dan interpretasi .....                                 | 96         |
| 9.3.2 Checklist implementasi dan quality control .....                           | 96         |
| 9.4 Descriptor molekul RDKit .....                                               | 97         |
| 9.4.1 Prinsip, parameter, dan interpretasi .....                                 | 97         |
| 9.4.2 Checklist implementasi dan quality control .....                           | 98         |
| 9.5 Interpretasi lintas kandidat .....                                           | 98         |
| 9.5.1 Prinsip, parameter, dan interpretasi .....                                 | 98         |
| 9.5.2 Checklist implementasi dan quality control .....                           | 99         |
| 9.6 Rangkuman Bab .....                                                          | 99         |
| <b>BAB 10 SCHEDULING PARALEL, RESOURCE LIMIT, FAIR QUEUE, DAN STOP AMAN.....</b> | <b>101</b> |
| 10.1 Paralelisasi batch .....                                                    | 101        |
| 10.1.1 Prinsip, parameter, dan interpretasi .....                                | 102        |
| 10.1.2 Checklist implementasi dan quality control .....                          | 102        |
| 10.2 Resource guardrail.....                                                     | 103        |
| 10.2.1 Prinsip, parameter, dan interpretasi .....                                | 104        |
| 10.2.2 Checklist implementasi dan quality control .....                          | 104        |
| 10.3 Fair queue .....                                                            | 104        |
| 10.3.1 Prinsip, parameter, dan interpretasi .....                                | 105        |
| 10.3.2 Checklist implementasi dan quality control .....                          | 105        |
| 10.4 Stop aman.....                                                              | 106        |
| 10.4.1 Prinsip, parameter, dan interpretasi .....                                | 106        |
| 10.4.2 Checklist implementasi dan quality control .....                          | 107        |

|                                                                                            |            |
|--------------------------------------------------------------------------------------------|------------|
| 10.5 Monitoring dan log.....                                                               | 107        |
| 10.5.1 Prinsip, parameter, dan interpretasi .....                                          | 108        |
| 10.5.2 Checklist implementasi dan quality control .....                                    | 108        |
| 10.6 Rangkuman Bab .....                                                                   | 109        |
| <b>BAB 11 RESUME, CHECKPOINT, CACHE, HASH, SQLITE, DAN PROVENANCE.....</b>                 | <b>110</b> |
| 11.1 Checkpoint per tahap.....                                                             | 110        |
| 11.1.1 Prinsip, parameter, dan interpretasi .....                                          | 111        |
| 11.1.2 Checklist implementasi dan quality control .....                                    | 111        |
| 11.2 Cache berbasis spec hash .....                                                        | 112        |
| 11.2.1 Prinsip, parameter, dan interpretasi .....                                          | 112        |
| 11.2.2 Checklist implementasi dan quality control .....                                    | 113        |
| 11.3 SQLite sebagai indeks proyek.....                                                     | 113        |
| 11.3.1 Prinsip, parameter, dan interpretasi .....                                          | 114        |
| 11.3.2 Checklist implementasi dan quality control .....                                    | 114        |
| 11.4 Environment snapshot .....                                                            | 114        |
| 11.4.1 Prinsip, parameter, dan interpretasi .....                                          | 115        |
| 11.4.2 Checklist implementasi dan quality control .....                                    | 116        |
| 11.5 Audit trail dan provenance .....                                                      | 116        |
| 11.5.1 Prinsip, parameter, dan interpretasi .....                                          | 116        |
| 11.5.2 Checklist implementasi dan quality control .....                                    | 117        |
| 11.6 Rangkuman Bab .....                                                                   | 117        |
| <b>BAB 12 RANKING MULTIKRITERIA, EKSPOR, BENCHMARK, DAN VALIDASI HIGH-THROUGHPUT .....</b> | <b>119</b> |
| 12.1 Normalisasi descriptor untuk ranking .....                                            | 119        |
| 12.1.1 Prinsip, parameter, dan interpretasi .....                                          | 120        |
| 12.1.2 Checklist implementasi dan quality control .....                                    | 121        |
| 12.2 Energi formula-sama .....                                                             | 121        |
| 12.2.1 Prinsip, parameter, dan interpretasi .....                                          | 121        |
| 12.2.2 Checklist implementasi dan quality control .....                                    | 122        |

|                                                                                |            |
|--------------------------------------------------------------------------------|------------|
| 12.3 QC sebagai syarat ranking.....                                            | 122        |
| 12.3.1 Prinsip, parameter, dan interpretasi .....                              | 123        |
| 12.3.2 Checklist implementasi dan quality control .....                        | 123        |
| 12.4 Ekspor riset.....                                                         | 124        |
| 12.4.1 Prinsip, parameter, dan interpretasi .....                              | 124        |
| 12.4.2 Checklist implementasi dan quality control .....                        | 125        |
| 12.5 Benchmark dan regression baseline .....                                   | 125        |
| 12.5.1 Prinsip, parameter, dan interpretasi .....                              | 126        |
| 12.5.2 Checklist implementasi dan quality control .....                        | 126        |
| 12.6 Rangkuman Bab .....                                                       | 127        |
| <b>BAB 13 ANTARMUKA STREAMLIT, SIDEBAR, DASHBOARD, DAN SESSION STATE .....</b> | <b>128</b> |
| 13.1 Peran antarmuka .....                                                     | 128        |
| 13.1.1 Prinsip, parameter, dan interpretasi .....                              | 129        |
| 13.1.2 Checklist implementasi dan quality control .....                        | 130        |
| 13.2 Sidebar pengaturan proyek .....                                           | 130        |
| 13.2.1 Prinsip, parameter, dan interpretasi .....                              | 130        |
| 13.2.2 Checklist implementasi dan quality control .....                        | 131        |
| 13.3 Tab Dashboard dan Input & QC.....                                         | 132        |
| 13.3.1 Prinsip, parameter, dan interpretasi .....                              | 132        |
| 13.3.2 Checklist implementasi dan quality control .....                        | 133        |
| 13.4 Workflow dan Queue/Monitor .....                                          | 133        |
| 13.4.1 Prinsip, parameter, dan interpretasi .....                              | 134        |
| 13.4.2 Checklist implementasi dan quality control .....                        | 134        |
| 13.5 Results dan Benchmark/Provenance .....                                    | 134        |
| 13.5.1 Prinsip, parameter, dan interpretasi .....                              | 135        |
| 13.5.2 Checklist implementasi dan quality control .....                        | 136        |
| 13.6 Rangkuman Bab .....                                                       | 136        |
| <b>BAB 14 TUTORIAL PENGGUNAAN QMSCREEN DARI INSTALASI SAMPAI EKSPOR .....</b>  | <b>137</b> |

|                                                         |     |
|---------------------------------------------------------|-----|
| 14.1 Persiapan lingkungan .....                         | 137 |
| 14.1.1 Prinsip, parameter, dan interpretasi .....       | 138 |
| 14.1.2 Checklist implementasi dan quality control ..... | 139 |
| 14.2 Menjalankan aplikasi.....                          | 139 |
| 14.2.1 Prinsip, parameter, dan interpretasi .....       | 139 |
| 14.2.2 Checklist implementasi dan quality control ..... | 140 |
| 14.3 Latihan cepat Ethanol.....                         | 140 |
| 14.3.1 Prinsip, parameter, dan interpretasi .....       | 141 |
| 14.3.2 Checklist implementasi dan quality control ..... | 141 |
| 14.4 Membaca hasil.....                                 | 142 |
| 14.4.1 Prinsip, parameter, dan interpretasi .....       | 142 |
| 14.4.2 Checklist implementasi dan quality control ..... | 143 |
| 14.5 Ekspor dan pengarsipan.....                        | 143 |
| 14.5.1 Prinsip, parameter, dan interpretasi .....       | 143 |
| 14.5.2 Checklist implementasi dan quality control ..... | 144 |
| 14.6 Rangkuman Bab .....                                | 144 |
| BAB 15 STUDI KASUS KATALOG 8A, 8C, 8S, 8W, DAN 8V ..... | 145 |
| 15.1 Karakter data katalog.....                         | 145 |
| 15.1.1 Prinsip, parameter, dan interpretasi .....       | 146 |
| 15.1.2 Checklist implementasi dan quality control ..... | 147 |
| 15.2 Persiapan batch studi kasus .....                  | 147 |
| 15.2.1 Prinsip, parameter, dan interpretasi .....       | 148 |
| 15.2.2 Checklist implementasi dan quality control ..... | 148 |
| 15.3 Strategi conformer dan prescreen.....              | 149 |
| 15.3.1 Prinsip, parameter, dan interpretasi .....       | 149 |
| 15.3.2 Checklist implementasi dan quality control ..... | 150 |
| 15.4 Perbandingan descriptor yang aman.....             | 150 |
| 15.4.1 Prinsip, parameter, dan interpretasi .....       | 151 |
| 15.4.2 Checklist implementasi dan quality control ..... | 151 |

|                                                                                  |            |
|----------------------------------------------------------------------------------|------------|
| 15.5 Pelaporan studi kasus .....                                                 | 151        |
| 15.5.1 Prinsip, parameter, dan interpretasi .....                                | 152        |
| 15.5.2 Checklist implementasi dan quality control .....                          | 153        |
| 15.6 Rangkuman Bab .....                                                         | 153        |
| <b>BAB 16 VALIDASI, TROUBLESHOOTING, ERROR CODE, DAN QUALITY ASSURANCE .....</b> | <b>154</b> |
| 16.1 Validasi berlapis .....                                                     | 154        |
| 16.1.1 Prinsip, parameter, dan interpretasi .....                                | 155        |
| 16.1.2 Checklist implementasi dan quality control .....                          | 156        |
| 16.2 Panduan QMS-E001 sampai QMS-E010 .....                                      | 156        |
| 16.2.1 Prinsip, parameter, dan interpretasi .....                                | 156        |
| 16.2.2 Checklist implementasi dan quality control .....                          | 157        |
| 16.3 Troubleshooting SCF dan optimasi .....                                      | 157        |
| 16.3.1 Prinsip, parameter, dan interpretasi .....                                | 158        |
| 16.3.2 Checklist implementasi dan quality control .....                          | 158        |
| 16.4 Regression testing .....                                                    | 159        |
| 16.4.1 Prinsip, parameter, dan interpretasi .....                                | 159        |
| 16.4.2 Checklist implementasi dan quality control .....                          | 160        |
| 16.5 Checklist sebelum publikasi .....                                           | 160        |
| 16.5.1 Prinsip, parameter, dan interpretasi .....                                | 161        |
| 16.5.2 Checklist implementasi dan quality control .....                          | 161        |
| 16.6 Rangkuman Bab .....                                                         | 162        |
| <b>BAB 17 ARSITEKTUR KODE QMSCREEN V1.0.5 DAN PANDUAN PENGEMBANGAN .....</b>     | <b>163</b> |
| 17.1 Bootstrap dan workspace writable .....                                      | 163        |
| 17.1.1 Prinsip, parameter, dan interpretasi .....                                | 164        |
| 17.1.2 Checklist implementasi dan quality control .....                          | 165        |
| 17.2 Lapisan utilitas dan database .....                                         | 165        |
| 17.2.1 Prinsip, parameter, dan interpretasi .....                                | 165        |
| 17.2.2 Checklist implementasi dan quality control .....                          | 166        |

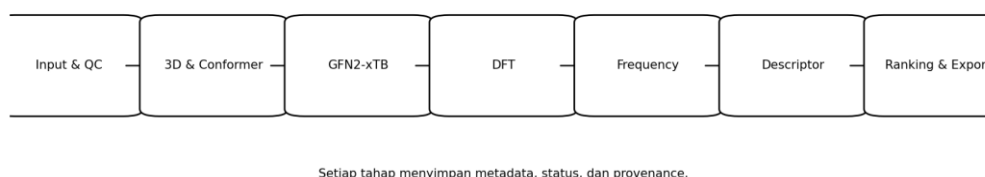
|                                                                                       |            |
|---------------------------------------------------------------------------------------|------------|
| 17.3 Lapisan struktur dan engine .....                                                | 166        |
| 17.3.1 Prinsip, parameter, dan interpretasi .....                                     | 167        |
| 17.3.2 Checklist implementasi dan quality control .....                               | 167        |
| 17.4 Scheduler dan state .....                                                        | 168        |
| 17.4.1 Prinsip, parameter, dan interpretasi .....                                     | 168        |
| 17.4.2 Checklist implementasi dan quality control .....                               | 169        |
| 17.5 Prinsip pengembangan lanjutan.....                                               | 169        |
| 17.5.1 Prinsip, parameter, dan interpretasi .....                                     | 170        |
| 17.5.2 Checklist implementasi dan quality control .....                               | 170        |
| 17.6 Rangkuman Bab .....                                                              | 171        |
| <b>BAB 18 EVALUASI PEMBELAJARAN, LATIHAN, DAN PROYEK</b><br><b>PENGEMBANGAN .....</b> | <b>172</b> |
| 18.1 Tujuan evaluasi.....                                                             | 172        |
| 18.1.1 Prinsip, parameter, dan interpretasi .....                                     | 173        |
| 18.1.2 Checklist implementasi dan quality control .....                               | 173        |
| 18.2 Latihan desain workflow .....                                                    | 174        |
| 18.2.1 Prinsip, parameter, dan interpretasi .....                                     | 174        |
| 18.2.2 Checklist implementasi dan quality control .....                               | 175        |
| 18.3 Latihan audit provenance .....                                                   | 175        |
| 18.3.1 Prinsip, parameter, dan interpretasi .....                                     | 176        |
| 18.3.2 Checklist implementasi dan quality control .....                               | 176        |
| 18.4 Proyek mini validasi.....                                                        | 176        |
| 18.4.1 Prinsip, parameter, dan interpretasi .....                                     | 177        |
| 18.4.2 Checklist implementasi dan quality control .....                               | 177        |
| 18.5 Proyek pengembangan perangkat lunak.....                                         | 178        |
| 18.5.1 Prinsip, parameter, dan interpretasi .....                                     | 178        |
| 18.5.2 Checklist implementasi dan quality control .....                               | 179        |
| 18.6 Rangkuman Bab .....                                                              | 179        |
| <b>LAMPIRAN 1 20 SOAL EVALUASI .....</b>                                              | <b>180</b> |

|                                                                                                |     |
|------------------------------------------------------------------------------------------------|-----|
| LAMPIRAN 2 TUTORIAL INSTALASI QMSCREEN    Tutorial Instalasi QMScreen 1.0.5 di WSL/Ubuntu..... | 184 |
| 1. Pendahuluan .....                                                                           | 184 |
| 3. Menempatkan File Instalasi .....                                                            | 185 |
| 11. Perintah Penting QMScreen.....                                                             | 189 |
| Menjalankan aplikasi.....                                                                      | 189 |
| Menyiapkan ulang dependency .....                                                              | 189 |
| Reset virtual environment.....                                                                 | 189 |
| Fungsi direktori.....                                                                          | 190 |
| 14. Contoh Proses Instalasi Lengkap .....                                                      | 191 |
| Penutup.....                                                                                   | 196 |
| GLOSARIUM .....                                                                                | 197 |
| DAFTAR PUSTAKA .....                                                                           | 203 |

## BAB 1 PENGANTAR QMSCREEN DAN HIGH-THROUGHPUT QUANTUM CHEMISTRY

Bab ini membahas pengantar qmscreen dan high-throughput quantum chemistry sebagai bagian dari workflow QMScreen V1.0.5. Pembahasan menempatkan konsep kimia komputasi, implementasi perangkat lunak, scientific guardrail, quality control, dan reproducibility dalam satu alur. Setiap subbab diakhiri dengan checklist praktis agar pembaca dapat menghubungkan teori dengan penggunaan aplikasi.

**Alur High-Throughput QMScreen**



*Gambar 1.1. Peta konseptual pengantar qmscreen dan high-throughput quantum chemistry.*

**Tabel 1.1. Peta materi Bab 1**

| Subbab | Fokus                                       | Keluaran utama                                                   |
|--------|---------------------------------------------|------------------------------------------------------------------|
| 1.1    | Posisi QMScreen dalam riset kimia komputasi | project terstruktur, hasil per molekul, log, dan data provenance |
| 1.2    | Tujuan ilmiah dan operasional               | daftar kandidat yang lolos QC dan status setiap tahap            |
| 1.3    | Arsitektur multi-fidelity                   | alur kandidat dari struktur mentah sampai hasil final            |
| 1.4    | Scientific guardrails                       | QC record dan error code                                         |
| 1.5    | Ruang lingkup buku                          | peta belajar dari konsep sampai evaluasi                         |

### 1.1 Posisi QMScreen dalam riset kimia komputasi

Pembahasan posisi qmscreen dalam riset kimia komputasi dimulai dari prinsip bahwa satu angka komputasi hanya bermakna jika struktur, metode, unit, dan status validasinya diketahui. QMScreen diposisikan sebagai pengelola workflow high-throughput yang menghubungkan persiapan struktur, metode semiempirik,

DFT, validasi hasil, dan pelaporan. Elemen yang paling relevan pada tahap ini meliputi workflow bertingkat, otomasi yang dapat diaudit dan pemisahan scientific logic dan presentation layer. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Cramer, 2004; Jensen, 2017).

Implementasi perangkat lunak untuk bagian ini mengikuti pola parse/validate/execute/record. Input terlebih dahulu dibaca, kemudian diverifikasi, setelah itu engine dijalankan, dan akhirnya keluaran dipaketkan menjadi result record. Parameter yang terkait mencakup workflow bertingkat, otomasi yang dapat diaudit, pemisahan scientific logic dan presentation layer. Keluaran yang diharapkan adalah project terstruktur, hasil per molekul, log, dan data provenance. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### ***1.1.1 Prinsip, parameter, dan interpretasi***

Quality control harus membedakan kondisi tidak valid, kondisi meragukan, dan kondisi yang sekadar berbeda dari default. Aturan utama yang perlu dipertahankan adalah bahwa parameter ilmiah tidak disembunyikan dan energi total lintas komposisi tidak diperlakukan sebagai afinitas universal. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Strategi operasional yang aman untuk posisi qmscreen dalam riset kimia komputasi adalah memulai dari konfigurasi sederhana, memeriksa log, lalu menambah kompleksitas satu parameter pada satu waktu. Setelah tahap selesai, periksa apakah project terstruktur, hasil per molekul, log, dan data provenance benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang

sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***1.1.2 Checklist implementasi dan quality control***

- Pastikan parameter workflow bertingkat telah ditentukan dan disimpan.
- Pastikan parameter otomasi yang dapat diaudit telah ditentukan dan disimpan.
- Pastikan parameter pemisahan scientific logic dan presentation layer telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: parameter ilmiah tidak disembunyikan dan energi total lintas komposisi tidak diperlakukan sebagai afinitas universal.
- Pastikan keluaran tahap tersedia sebagai project terstruktur, hasil per molekul, log, dan data provenance dan status QC tercatat.

## **1.2 Tujuan ilmiah dan operasional**

Pada QMScreen, tujuan ilmiah dan operasional bukan fitur dekoratif antarmuka. Tujuan utama bukan sekadar mempercepat klik, tetapi menyeragamkan proses agar banyak kandidat diperlakukan dengan aturan yang konsisten. Pendekatan ini membuat tahap tersebut dapat diuji sebagai komponen ilmiah yang mandiri. Elemen yang paling relevan pada tahap ini meliputi screening banyak molekul, konsistensi parameter, pengurangan kesalahan input dan reuse hasil. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Cramer, 2004; Jensen, 2017).

Secara arsitektural, bagian tujuan ilmiah dan operasional ditempatkan di luar logika tampilan sebanyak mungkin. Data dari widget hanya menjadi sumber konfigurasi, sedangkan fungsi inti bekerja pada struktur Python yang dapat diuji. Titik kendalanya meliputi screening banyak molekul, konsistensi parameter, pengurangan kesalahan input, reuse hasil. Keluaran yang diharapkan adalah daftar kandidat yang lolos QC dan status setiap tahap. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### ***1.2.1 Prinsip, parameter, dan interpretasi***

Risiko terbesar dalam otomasi adalah membuat proses tampak berhasil ketika asumsi ilmiahnya tidak terpenuhi. Aturan utama yang perlu dipertahankan adalah bahwa otomasi harus berhenti atau memberi peringatan ketika prasyarat ilmiah tidak terpenuhi. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Untuk tujuan riset, hasil dari tujuan ilmiah dan operasional sebaiknya ditinjau bersama data sebelum dan sesudah tahap tersebut agar perubahan dapat dilacak. Setelah tahap selesai, periksa apakah daftar kandidat yang lolos QC dan status setiap tahap benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***1.2.2 Checklist implementasi dan quality control***

- Pastikan parameter screening banyak molekul telah ditentukan dan disimpan.
- Pastikan parameter konsistensi parameter telah ditentukan dan disimpan.
- Pastikan parameter pengurangan kesalahan input telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: otomasi harus berhenti atau memberi peringatan ketika prasyarat ilmiah tidak terpenuhi.
- Pastikan keluaran tahap tersedia sebagai daftar kandidat yang lolos QC dan status setiap tahap dan status QC tercatat.

## **1.3 Arsitektur multi-fidelity**

High-throughput chemistry menuntut aturan yang konsisten ketika jumlah molekul bertambah. Dalam konteks arsitektur multi-fidelity, QMScreen memakai tahapan berbiaya berbeda sehingga perhitungan murah dapat mendahului perhitungan mahal. Elemen yang paling relevan pada tahap ini meliputi RDKit,

GFN2-xTB, DFT, frequency opsional dan descriptor. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Cramer, 2004; Jensen, 2017).

Untuk batch besar, implementasi harus tahan terhadap kegagalan parsial. QMScreen karena itu menyimpan status per molekul dan per tahap, sambil mempertahankan parameter seperti `RDKit`, `GFN2-xTB`, `DFT`, frequency opsional, `descriptor` pada manifest atau provenance. Keluaran yang diharapkan adalah alur kandidat dari struktur mentah sampai hasil final. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### ***1.3.1 Prinsip, parameter, dan interpretasi***

Ketika workflow dipercepat, kontrol ilmiah tidak boleh ikut dipangkas. Aturan utama yang perlu dipertahankan adalah bahwa setiap penyaringan harus mempertahankan alasan keputusan dan tidak menganggap metode cepat identik dengan DFT. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Pada komputer dengan resource terbatas, arsitektur multi-fidelity perlu diatur bersama jumlah worker, thread, memory, dan biaya tahap berikutnya. Setelah tahap selesai, periksa apakah alur kandidat dari struktur mentah sampai hasil final benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***1.3.2 Checklist implementasi dan quality control***

- Pastikan parameter RDKit telah ditentukan dan disimpan.
- Pastikan parameter GFN2-xTB telah ditentukan dan disimpan.
- Pastikan parameter DFT telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: setiap penyaringan harus mempertahankan alasan keputusan dan tidak menganggap metode cepat identik dengan DFT.
- Pastikan keluaran tahap tersedia sebagai alur kandidat dari struktur mentah sampai hasil final dan status QC tercatat.

## **1.4 Scientific guardrails**

Guardrail adalah aturan eksplisit yang mencegah kombinasi parameter yang jelas tidak konsisten dan memaksa metadata disimpan. Dalam workflow komputasi skala besar, konsep ini perlu didefinisikan sebelum job pertama dijalankan karena keputusan awal akan memengaruhi biaya, keterbandingan, dan kemampuan reproduksi hasil. Elemen yang paling relevan pada tahap ini meliputi charge, multiplicity, konvergensi SCF, minimum check, unit dan provenance. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Cramer, 2004; Jensen, 2017).

Di tingkat implementasi, QMScreen mengubah pilihan pengguna menjadi record konfigurasi dan menjalankan fungsi scientific logic yang sesuai. Komponen seperti 'charge', 'multiplicity', konvergensi SCF, minimum check, 'unit', 'provenance' tidak diperlakukan sebagai nilai sementara, tetapi sebagai bagian dari state proyek yang dapat ditulis ke disk. Keluaran yang diharapkan adalah QC record dan error code. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### ***1.4.1 Prinsip, parameter, dan interpretasi***

Scientific guardrail pada bagian ini sengaja dibuat konservatif. Aturan utama yang perlu dipertahankan adalah bahwa warning tidak boleh disamakan dengan error; hasil parsial harus diberi status yang benar. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai

belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Dalam penggunaan sehari-hari, pengguna sebaiknya menguji scientific guardrails pada satu atau dua molekul kecil sebelum memperbesar batch. Setelah tahap selesai, periksa apakah QC record dan error code benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

#### ***1.4.2 Checklist implementasi dan quality control***

- Pastikan parameter charge telah ditentukan dan disimpan.
- Pastikan parameter multiplicity telah ditentukan dan disimpan.
- Pastikan parameter konvergensi SCF telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: warning tidak boleh disamakan dengan error; hasil parsial harus diberi status yang benar.
- Pastikan keluaran tahap tersedia sebagai QC record dan error code dan status QC tercatat.

### **1.5 Ruang lingkup buku**

Pembahasan ruang lingkup buku dimulai dari prinsip bahwa satu angka komputasi hanya bermakna jika struktur, metode, unit, dan status validasinya diketahui. Buku mengikuti implementasi QMScreen V1.0.5 dan membahas pemakaian, konsep, risiko, interpretasi, serta pengembangan kode. Elemen yang paling relevan pada tahap ini meliputi teori seperlunya, tutorial operasional, studi kasus, validasi dan troubleshooting. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Cramer, 2004; Jensen, 2017).

Implementasi perangkat lunak untuk bagian ini mengikuti pola parse/validate/execute/record. Input terlebih dahulu dibaca, kemudian diverifikasi, setelah itu engine dijalankan, dan akhirnya keluaran dipaketkan menjadi result record. Parameter yang terkait mencakup teori seperlunya, tutorial operasional, studi kasus, `validasi`, `troubleshooting`. Keluaran yang diharapkan adalah peta belajar dari konsep sampai evaluasi. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### ***1.5.1 Prinsip, parameter, dan interpretasi***

Quality control harus membedakan kondisi tidak valid, kondisi meragukan, dan kondisi yang sekadar berbeda dari default. Aturan utama yang perlu dipertahankan adalah bahwa buku tidak menggantikan perancangan metode riset atau validasi domain kimia spesifik. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Strategi operasional yang aman untuk ruang lingkup buku adalah memulai dari konfigurasi sederhana, memeriksa log, lalu menambah kompleksitas satu parameter pada satu waktu. Setelah tahap selesai, periksa apakah peta belajar dari konsep sampai evaluasi benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***1.5.2 Checklist implementasi dan quality control***

- Pastikan parameter teori seperlunya telah ditentukan dan disimpan.
- Pastikan parameter tutorial operasional telah ditentukan dan disimpan.
- Pastikan parameter studi kasus telah ditentukan dan disimpan.

- Pastikan guardrail berikut dipenuhi: buku tidak menggantikan perancangan metode riset atau validasi domain kimia spesifik.
- Pastikan keluaran tahap tersedia sebagai peta belajar dari konsep sampai evaluasi dan status QC tercatat.

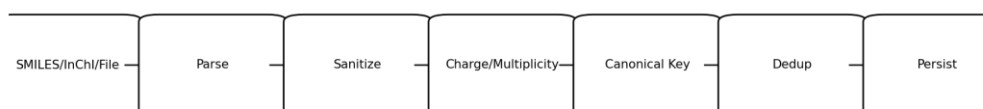
## 1.6 Rangkuman Bab

Bab 1 menunjukkan bahwa pengantar qmscreen dan high-throughput quantum chemistry harus dipahami sebagai satu sistem yang menghubungkan keputusan ilmiah dan keputusan perangkat lunak. Keandalan QMScreen bergantung pada kemampuan setiap tahap untuk menerima input yang jelas, menjalankan metode dengan parameter eksplisit, mencatat status, mempertahankan hasil parsial secara benar, dan menyediakan provenance. Dengan pola ini, batch yang besar tetap dapat diaudit dan dikembangkan tanpa menjadikan GUI sebagai satu-satunya sumber kebenaran. Pembaca sebaiknya menggunakan checklist pada setiap subbab sebagai kendali sebelum melanjutkan ke tahap berikutnya.

## BAB 2 INPUT MASSAL, FORMAT STRUKTUR, DAN KATALOG PERSISTEN

Bab ini membahas input massal, format struktur, dan katalog persisten sebagai bagian dari workflow QMScreen V1.0.5. Pembahasan menempatkan konsep kimia komputasi, implementasi perangkat lunak, scientific guardrail, quality control, dan reproducibility dalam satu alur. Setiap subbab diakhiri dengan checklist praktis agar pembaca dapat menghubungkan teori dengan penggunaan aplikasi.

**Pipeline Input dan Quality Control**



*Gambar 2.1. Peta konseptual input massal, format struktur, dan katalog persisten.*

**Tabel 2.1. Peta materi Bab 2**

| Subbab | Fokus                                      | Keluaran utama                             |
|--------|--------------------------------------------|--------------------------------------------|
| 2.1    | Sumber input yang didukung                 | objek spesifikasi molekul yang seragam     |
| 2.2    | SMILES dan InChI sebagai representasi teks | structure key dan metadata sumber          |
| 2.3    | Upload file terstruktur                    | manifest input dan daftar error per file   |
| 2.4    | Katalog upload_molekul.json                | katalog permanen yang dapat dibaca kembali |
| 2.5    | Strategi QC pada tahap input               | tabel Input & QC dan pesan tindakan        |

### 2.1 Sumber input yang didukung

Pembahasan sumber input yang didukung dimulai dari prinsip bahwa satu angka komputasi hanya bermakna jika struktur, metode, unit, dan status validasinya diketahui. Input high-throughput harus menerima representasi molekul yang umum tanpa mengorbankan identitas struktur. Elemen yang paling relevan pada

tahap ini meliputi SMILES, InChI, CSV, SDF, MOL, PDB, XYZ, SMI, TXT dan katalog JSON. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (RDKit Documentation, 2026).

Implementasi perangkat lunak untuk bagian ini mengikuti pola parse/validate/execute/record. Input terlebih dahulu dibaca, kemudian diverifikasi, setelah itu engine dijalankan, dan akhirnya keluaran dipaketkan menjadi result record. Parameter yang terkait mencakup `SMILES`, `InChI`, `CSV`, `SDF`, `MOL`, `PDB`, `XYZ`, `SMI`, `TXT`, katalog JSON. Keluaran yang diharapkan adalah objek spesifikasi molekul yang seragam. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### ***2.1.1 Prinsip, parameter, dan interpretasi***

Quality control harus membedakan kondisi tidak valid, kondisi meragukan, dan kondisi yang sekadar berbeda dari default. Aturan utama yang perlu dipertahankan adalah bahwa parser harus membedakan kegagalan sintaks, kegagalan sanitasi, dan informasi stereokimia yang hilang. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Strategi operasional yang aman untuk sumber input yang didukung adalah memulai dari konfigurasi sederhana, memeriksa log, lalu menambah kompleksitas satu parameter pada satu waktu. Setelah tahap selesai, periksa apakah objek spesifikasi molekul yang seragam benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### 2.1.2 Checklist implementasi dan quality control

- Pastikan parameter SMILES telah ditentukan dan disimpan.
- Pastikan parameter InChI telah ditentukan dan disimpan.
- Pastikan parameter CSV telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: parser harus membedakan kegagalan sintaks, kegagalan sanitasi, dan informasi stereokimia yang hilang.
- Pastikan keluaran tahap tersedia sebagai objek spesifikasi molekul yang seragam dan status QC tercatat.

**Tabel 2.2. Format input yang dibahas QMScreen**

| Format     | Kandungan utama            | Catatan QC                                |
|------------|----------------------------|-------------------------------------------|
| SMILES/SMI | Konektivitas dalam teks    | Periksa sanitasi dan stereokimia          |
| InChI      | Identifier struktur teks   | Gunakan InChIKey untuk identitas tambahan |
| CSV        | Tabel banyak molekul       | Kolom struktur harus diidentifikasi       |
| SDF/MOL    | Konektivitas dan koordinat | SDF dapat memuat banyak record            |
| PDB        | Koordinat 3D               | Konektivitas/atom perlu diperiksa         |
| XYZ        | Elemen dan koordinat       | Ikatan tidak selalu eksplisit             |
| TXT        | Input teks generik         | Parser harus membedakan SMILES/InChI      |

## 2.2 SMILES dan InChI sebagai representasi teks

Pada QMScreen, smiles dan inchi sebagai representasi teks bukan fitur dekoratif antarmuka. Representasi teks memudahkan input massal dan pencatatan ulang, tetapi tetap harus dinormalisasi sebelum dipakai sebagai identitas. Pendekatan ini membuat tahap tersebut dapat diuji sebagai komponen ilmiah yang mandiri. Elemen yang paling relevan pada tahap ini meliputi canonical SMILES, InChIKey, source\_ref dan nama input. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (RDKit Documentation, 2026).

Secara arsitektural, bagian smiles dan inchi sebagai representasi teks ditempatkan di luar logika tampilan sebanyak mungkin. Data dari widget hanya menjadi sumber konfigurasi, sedangkan fungsi inti bekerja pada struktur Python yang dapat diuji. Titik kendalinya meliputi canonical SMILES, `InChIKey`, `source\_ref`, nama input. Keluaran yang diharapkan adalah structure key dan metadata sumber. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### ***2.2.1 Prinsip, parameter, dan interpretasi***

Risiko terbesar dalam otomasi adalah membuat proses tampak berhasil ketika asumsi ilmiahnya tidak terpenuhi. Aturan utama yang perlu dipertahankan adalah bahwa nama pengguna tidak boleh dipakai sebagai satu-satunya kunci identitas karena dapat berbeda untuk struktur sama. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Untuk tujuan riset, hasil dari smiles dan inchi sebagai representasi teks sebaiknya ditinjau bersama data sebelum dan sesudah tahap tersebut agar perubahan dapat dilacak. Setelah tahap selesai, periksa apakah structure key dan metadata sumber benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***2.2.2 Checklist implementasi dan quality control***

- Pastikan parameter canonical SMILES telah ditentukan dan disimpan.
- Pastikan parameter InChIKey telah ditentukan dan disimpan.
- Pastikan parameter source\_ref telah ditentukan dan disimpan.

- Pastikan guardrail berikut dipenuhi: nama pengguna tidak boleh dipakai sebagai satu-satunya kunci identitas karena dapat berbeda untuk struktur sama.
- Pastikan keluaran tahap tersedia sebagai structure key dan metadata sumber dan status QC tercatat.

## 2.3 Upload file terstruktur

High-throughput chemistry menuntut aturan yang konsisten ketika jumlah molekul bertambah. Dalam konteks upload file terstruktur, File kimia membawa koordinat, konektivitas, atau kumpulan molekul sehingga parser harus mempertahankan informasi yang valid. Elemen yang paling relevan pada tahap ini meliputi SDF multirecord, MOL, PDB, XYZ dan CSV kolom struktur. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (RDKit Documentation, 2026).

Untuk batch besar, implementasi harus tahan terhadap kegagalan parsial. QMScreen karena itu menyimpan status per molekul dan per tahap, sambil mempertahankan parameter seperti SDF multirecord, `MOL`, `PDB`, `XYZ`, CSV kolom struktur pada manifest atau provenance. Keluaran yang diharapkan adalah manifest input dan daftar error per file. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### 2.3.1 Prinsip, parameter, dan interpretasi

Ketika workflow dipercepat, kontrol ilmiah tidak boleh ikut dipangkas. Aturan utama yang perlu dipertahankan adalah bahwa jangan menganggap semua XYZ memiliki konektivitas yang eksplisit; sanitasi dan inferensi ikatan perlu dipantau. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat

menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Pada komputer dengan resource terbatas, upload file terstruktur perlu diatur bersama jumlah worker, thread, memory, dan biaya tahap berikutnya. Setelah tahap selesai, periksa apakah manifest input dan daftar error per file benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***2.3.2 Checklist implementasi dan quality control***

- Pastikan parameter SDF multirecord telah ditentukan dan disimpan.
- Pastikan parameter MOL telah ditentukan dan disimpan.
- Pastikan parameter PDB telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: jangan menganggap semua XYZ memiliki konektivitas yang eksplisit; sanitasi dan inferensi ikatan perlu dipantau.
- Pastikan keluaran tahap tersedia sebagai manifest input dan daftar error per file dan status QC tercatat.

## **2.4 Katalog upload\_molekul.json**

QMScreen menyimpan input pengguna secara persisten agar molekul yang pernah ditambahkan dapat digunakan kembali. Dalam workflow komputasi skala besar, konsep ini perlu didefinisikan sebelum job pertama dijalankan karena keputusan awal akan memengaruhi biaya, keterbandingan, dan kemampuan reproduksi hasil. Elemen yang paling relevan pada tahap ini meliputi entry terbaru di atas, idempotensi struktur, molblock, formula, MW, charge dan multiplicity. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (RDKit Documentation, 2026).

Di tingkat implementasi, QMScreen mengubah pilihan pengguna menjadi record konfigurasi dan menjalankan fungsi scientific logic yang sesuai. Komponen seperti entry terbaru di atas, idempotensi struktur, 'molblock', 'formula', 'MW', 'charge', 'multiplicity' tidak diperlakukan sebagai nilai sementara, tetapi sebagai bagian dari state proyek yang dapat ditulis ke disk. Keluaran yang diharapkan

adalah katalog permanen yang dapat dibaca kembali. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

#### ***2.4.1 Prinsip, parameter, dan interpretasi***

Scientific guardrail pada bagian ini sengaja dibuat konservatif. Aturan utama yang perlu dipertahankan adalah bahwa katalog bukan pengganti provenance proyek; perubahan struktur harus menghasilkan digest baru. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Dalam penggunaan sehari-hari, pengguna sebaiknya menguji katalog `upload_molekul.json` pada satu atau dua molekul kecil sebelum memperbesar batch. Setelah tahap selesai, periksa apakah katalog permanen yang dapat dibaca kembali benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

#### ***2.4.2 Checklist implementasi dan quality control***

- Pastikan parameter entry terbaru di atas telah ditentukan dan disimpan.
- Pastikan parameter idempotensi struktur telah ditentukan dan disimpan.
- Pastikan parameter molblock telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: katalog bukan pengganti provenance proyek; perubahan struktur harus menghasilkan digest baru.
- Pastikan keluaran tahap tersedia sebagai katalog permanen yang dapat dibaca kembali dan status QC tercatat.

## 2.5 Strategi QC pada tahap input

Pembahasan strategi qc pada tahap input dimulai dari prinsip bahwa satu angka komputasi hanya bermakna jika struktur, metode, unit, dan status validasinya diketahui. QC dilakukan sedini mungkin karena kesalahan struktur yang diteruskan ke DFT jauh lebih mahal. Elemen yang paling relevan pada tahap ini meliputi sanitize, formula, atom count, charge, multiplicity dan duplicate. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (RDKit Documentation, 2026).

Implementasi perangkat lunak untuk bagian ini mengikuti pola parse/validate/execute/record. Input terlebih dahulu dibaca, kemudian diverifikasi, setelah itu engine dijalankan, dan akhirnya keluaran dipaketkan menjadi result record. Parameter yang terkait mencakup 'sanitize', 'formula', atom count, 'charge', 'multiplicity', 'duplicate'. Keluaran yang diharapkan adalah tabel Input & QC dan pesan tindakan. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### 2.5.1 Prinsip, parameter, dan interpretasi

Quality control harus membedakan kondisi tidak valid, kondisi meragukan, dan kondisi yang sekadar berbeda dari default. Aturan utama yang perlu dipertahankan adalah bahwa input invalid harus ditolak sebelum resource komputasi dialokasikan. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Strategi operasional yang aman untuk strategi qc pada tahap input adalah memulai dari konfigurasi sederhana, memeriksa log, lalu menambah kompleksitas satu parameter pada satu waktu. Setelah tahap selesai, periksa apakah tabel Input &

QC dan pesan tindakan benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***2.5.2 Checklist implementasi dan quality control***

- Pastikan parameter sanitize telah ditentukan dan disimpan.
- Pastikan parameter formula telah ditentukan dan disimpan.
- Pastikan parameter atom count telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: input invalid harus ditolak sebelum resource komputasi dialokasikan.
- Pastikan keluaran tahap tersedia sebagai tabel Input & QC dan pesan tindakan dan status QC tercatat.

#### **Sketsa implementasi Python - Bab 2 (pseudocode original)**

```
for record in incoming_records:
    spec = parse_and_normalize(record)
    validate_structure(spec)
    catalog.store_if_new(spec)
```

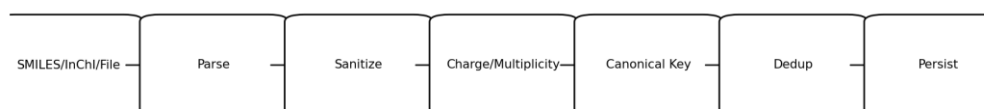
## **2.6 Rangkuman Bab**

Bab 2 menunjukkan bahwa input massal, format struktur, dan katalog persisten harus dipahami sebagai satu sistem yang menghubungkan keputusan ilmiah dan keputusan perangkat lunak. Keandalan QMScreen bergantung pada kemampuan setiap tahap untuk menerima input yang jelas, menjalankan metode dengan parameter eksplisit, mencatat status, mempertahankan hasil parsial secara benar, dan menyediakan provenance. Dengan pola ini, batch yang besar tetap dapat diaudit dan dikembangkan tanpa menjadikan GUI sebagai satu-satunya sumber kebenaran. Pembaca sebaiknya menggunakan checklist pada setiap subbab sebagai kendali sebelum melanjutkan ke tahap berikutnya.

## BAB 3 VALIDASI STRUKTUR, NORMALISASI, CHARGE, MULTIPLICITY, DAN DEDUPLIKASI

Bab ini membahas validasi struktur, normalisasi, charge, multiplicity, dan deduplikasi sebagai bagian dari workflow QMScreen V1.0.5. Pembahasan menempatkan konsep kimia komputasi, implementasi perangkat lunak, scientific guardrail, quality control, dan reproducibility dalam satu alur. Setiap subbab diakhiri dengan checklist praktis agar pembaca dapat menghubungkan teori dengan penggunaan aplikasi.

**Pipeline Input dan Quality Control**



Gambar 3.1. Peta konseptual validasi struktur, normalisasi, charge, multiplicity, dan deduplikasi.

**Tabel 3.1. Peta materi Bab 3**

| Subbab | Fokus                            | Keluaran utama                            |
|--------|----------------------------------|-------------------------------------------|
| 3.1    | Sanitasi dan normalisasi molekul | molekul kerja dan catatan normalisasi     |
| 3.2    | Charge dan multiplicity          | parameter spin yang diteruskan ke backend |
| 3.3    | Canonical structure key          | spec_hash dan structure key               |
| 3.4    | Deduplication sebelum queue      | input unik dan tabel duplikat             |
| 3.5    | Error taxonomy dan tindakan      | error record, log, dan status job         |

### 3.1 Sanitasi dan normalisasi molekul

Pembahasan sanitasi dan normalisasi molekul dimulai dari prinsip bahwa satu angka komputasi hanya bermakna jika struktur, metode, unit, dan status validasinya diketahui. Normalisasi mengubah berbagai bentuk input menjadi representasi kerja yang konsisten tanpa mengubah identitas kimia secara sembarangan. Elemen yang paling relevan pada tahap ini meliputi sanitasi valensi, hidrogen, aromatisitas, canonicalization dan metadata. Masing-masing elemen

harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (RDKit Documentation, 2026).

Implementasi perangkat lunak untuk bagian ini mengikuti pola parse/validate/execute/record. Input terlebih dahulu dibaca, kemudian diverifikasi, setelah itu engine dijalankan, dan akhirnya keluaran dipaketkan menjadi result record. Parameter yang terkait mencakup sanitasi valensi, 'hidrogen', 'aromatisitas', 'canonicalization', 'metadata'. Keluaran yang diharapkan adalah molekul kerja dan catatan normalisasi. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### ***3.1.1 Prinsip, parameter, dan interpretasi***

Quality control harus membedakan kondisi tidak valid, kondisi meragukan, dan kondisi yang sekadar berbeda dari default. Aturan utama yang perlu dipertahankan adalah bahwa normalisasi tidak boleh menebak tautomer, protonasi, atau stereokimia yang tidak dinyatakan tanpa kebijakan eksplisit. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Strategi operasional yang aman untuk sanitasi dan normalisasi molekul adalah memulai dari konfigurasi sederhana, memeriksa log, lalu menambah kompleksitas satu parameter pada satu waktu. Setelah tahap selesai, periksa apakah molekul kerja dan catatan normalisasi benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### 3.1.2 Checklist implementasi dan quality control

- Pastikan parameter sanitasi valensi telah ditentukan dan disimpan.
- Pastikan parameter hidrogen telah ditentukan dan disimpan.
- Pastikan parameter aromatisitas telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: normalisasi tidak boleh menebak tautomer, protonasi, atau stereokimia yang tidak dinyatakan tanpa kebijakan eksplisit.
- Pastikan keluaran tahap tersedia sebagai molekul kerja dan catatan normalisasi dan status QC tercatat.

## 3.2 Charge dan multiplicity

Pada QMScreen, charge dan multiplicity bukan fitur dekoratif antarmuka. Jumlah elektron menentukan apakah kombinasi charge dan multiplicity masuk akal untuk perhitungan elektron. Pendekatan ini membuat tahap tersebut dapat diuji sebagai komponen ilmiah yang mandiri. Elemen yang paling relevan pada tahap ini meliputi electron count, spin, multiplicity =  $2S+1$  dan closed-shell versus open-shell. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (RDKit Documentation, 2026).

Secara arsitektural, bagian charge dan multiplicity ditempatkan di luar logika tampilan sebanyak mungkin. Data dari widget hanya menjadi sumber konfigurasi, sedangkan fungsi inti bekerja pada struktur Python yang dapat diuji. Titik kendalinya meliputi electron count, `spin`, multiplicity =  $2S+1$ , closed-shell versus open-shell. Keluaran yang diharapkan adalah parameter spin yang diteruskan ke backend. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### 3.2.1 Prinsip, parameter, dan interpretasi

Risiko terbesar dalam otomasi adalah membuat proses tampak berhasil ketika asumsi ilmiahnya tidak terpenuhi. Aturan utama yang perlu dipertahankan adalah bahwa kombinasi yang tidak konsisten harus menghasilkan QMS-E002 atau peringatan yang dapat ditindaklanjuti. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum

tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Untuk tujuan riset, hasil dari charge dan multiplicity sebaiknya ditinjau bersama data sebelum dan sesudah tahap tersebut agar perubahan dapat dilacak. Setelah tahap selesai, periksa apakah parameter spin yang diteruskan ke backend benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

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

### ***3.2.2 Checklist implementasi dan quality control***

- Pastikan parameter electron count telah ditentukan dan disimpan.
- Pastikan parameter spin telah ditentukan dan disimpan.
- Pastikan parameter multiplicity =  $2S+1$  telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: kombinasi yang tidak konsisten harus menghasilkan QMS-E002 atau peringatan yang dapat ditindaklanjuti.
- Pastikan keluaran tahap tersedia sebagai parameter spin yang diteruskan ke backend dan status QC tercatat.

### **3.3 Canonical structure key**

High-throughput chemistry menuntut aturan yang konsisten ketika jumlah molekul bertambah. Dalam konteks canonical structure key, Kunci struktur kanonik membuat deduplikasi dan cache tidak bergantung pada nama file atau urutan input. Elemen yang paling relevan pada tahap ini meliputi canonical SMILES, InChIKey, charge, multiplicity dan hash. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (RDKit Documentation, 2026).

Untuk batch besar, implementasi harus tahan terhadap kegagalan parsial. QMScreen karena itu menyimpan status per molekul dan per tahap, sambil mempertahankan parameter seperti canonical SMILES, `InChIKey`, `charge`, `multiplicity`, `hash` pada manifest atau provenance. Keluaran yang diharapkan adalah spec\_hash dan structure key. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### ***3.3.1 Prinsip, parameter, dan interpretasi***

Ketika workflow dipercepat, kontrol ilmiah tidak boleh ikut dipangkas. Aturan utama yang perlu dipertahankan adalah bahwa hash harus memasukkan parameter yang mengubah makna kalkulasi; jika tidak, cache dapat mengembalikan hasil yang salah. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Pada komputer dengan resource terbatas, canonical structure key perlu diatur bersama jumlah worker, thread, memory, dan biaya tahap berikutnya. Setelah tahap selesai, periksa apakah spec\_hash dan structure key benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***3.3.2 Checklist implementasi dan quality control***

- Pastikan parameter canonical SMILES telah ditentukan dan disimpan.
- Pastikan parameter InChIKey telah ditentukan dan disimpan.
- Pastikan parameter charge telah ditentukan dan disimpan.

- Pastikan guardrail berikut dipenuhi: hash harus memasukkan parameter yang mengubah makna kalkulasi; jika tidak, cache dapat mengembalikan hasil yang salah.
- Pastikan keluaran tahap tersedia sebagai spec\_hash dan structure key dan status QC tercatat.

### 3.4 Deduplication sebelum queue

Deduplikasi mengurangi pekerjaan yang sama dan mencegah bias jumlah kandidat pada tabel hasil. Dalam workflow komputasi skala besar, konsep ini perlu didefinisikan sebelum job pertama dijalankan karena keputusan awal akan memengaruhi biaya, keterbandingan, dan kemampuan reproduksi hasil. Elemen yang paling relevan pada tahap ini meliputi kelompok struktur sama, daftar duplicate dan sumber pertama/terbaru. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (RDKit Documentation, 2026).

Di tingkat implementasi, QMScreen mengubah pilihan pengguna menjadi record konfigurasi dan menjalankan fungsi scientific logic yang sesuai. Komponen seperti kelompok struktur sama, daftar duplicate, sumber pertama/terbaru tidak diperlakukan sebagai nilai sementara, tetapi sebagai bagian dari state proyek yang dapat ditulis ke disk. Keluaran yang diharapkan adalah input unik dan tabel duplikat. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

#### 3.4.1 Prinsip, parameter, dan interpretasi

Scientific guardrail pada bagian ini sengaja dibuat konservatif. Aturan utama yang perlu dipertahankan adalah bahwa duplikat harus dilaporkan, bukan dihapus diam-diam, agar pengguna mengetahui perubahan jumlah input. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat

karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Dalam penggunaan sehari-hari, pengguna sebaiknya menguji deduplication sebelum queue pada satu atau dua molekul kecil sebelum memperbesar batch. Setelah tahap selesai, periksa apakah input unik dan tabel duplikat benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***3.4.2 Checklist implementasi dan quality control***

- Pastikan parameter kelompok struktur sama telah ditentukan dan disimpan.
- Pastikan parameter daftar duplicate telah ditentukan dan disimpan.
- Pastikan parameter sumber pertama/terbaru telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: duplikat harus dilaporkan, bukan dihapus diam-diam, agar pengguna mengetahui perubahan jumlah input.
- Pastikan keluaran tahap tersedia sebagai input unik dan tabel duplikat dan status QC tercatat.

## **3.5 Error taxonomy dan tindakan**

Pembahasan error taxonomy dan tindakan dimulai dari prinsip bahwa satu angka komputasi hanya bermakna jika struktur, metode, unit, dan status validasinya diketahui. Kode error konsisten memisahkan kegagalan input, 3D, xTB, SCF, optimasi, frequency, resource, parser, dan ekspor. Elemen yang paling relevan pada tahap ini meliputi QMS-E001 sampai QMS-E010. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (RDKit Documentation, 2026).

Implementasi perangkat lunak untuk bagian ini mengikuti pola parse/validate/execute/record. Input terlebih dahulu dibaca, kemudian diverifikasi, setelah itu engine dijalankan, dan akhirnya keluaran dipaketkan menjadi result record. Parameter yang terkait mencakup QMS-E001 sampai QMS-E010. Keluaran yang diharapkan adalah error record, log, dan status job. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali

keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### ***3.5.1 Prinsip, parameter, dan interpretasi***

Quality control harus membedakan kondisi tidak valid, kondisi meragukan, dan kondisi yang sekadar berbeda dari default. Aturan utama yang perlu dipertahankan adalah bahwa pesan error harus menjelaskan tindakan korektif dan menjaga hasil parsial yang masih valid. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Strategi operasional yang aman untuk error taxonomy dan tindakan adalah memulai dari konfigurasi sederhana, memeriksa log, lalu menambah kompleksitas satu parameter pada satu waktu. Setelah tahap selesai, periksa apakah error record, log, dan status job benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***3.5.2 Checklist implementasi dan quality control***

- Pastikan parameter QMS-E001 sampai QMS-E010 telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: pesan error harus menjelaskan tindakan korektif dan menjaga hasil parsial yang masih valid.
- Pastikan keluaran tahap tersedia sebagai error record, log, dan status job dan status QC tercatat.

**Tabel 3.2. Kode error QMScreen dan makna ringkas**

| Kode     | Makna                               | Tindakan awal                                   |
|----------|-------------------------------------|-------------------------------------------------|
| QMS-E001 | Struktur tidak dapat dibaca         | Periksa SMILES/InChI/file dan stereokimia       |
| QMS-E002 | Charge/multiplicity tidak konsisten | Periksa jumlah elektron dan spin                |
| QMS-E003 | Generate 3D gagal                   | Coba seed/parameter conformer lain              |
| QMS-E004 | GFN2-xTB/TB Lite gagal              | Periksa backend, struktur, charge               |
| QMS-E005 | SCF tidak konvergen                 | Tinjau struktur, metode, max cycle              |
| QMS-E006 | Optimasi geometri gagal             | Periksa geometri dan batas langkah              |
| QMS-E007 | Frequency/Hessian gagal             | Energi dapat dipertahankan; minimum belum valid |
| QMS-E008 | Resource limit                      | Kurangi worker/thread/memory request            |
| QMS-E009 | Result/parser tidak lengkap         | Periksa raw output dan parser                   |
| QMS-E010 | Ekspor gagal                        | Data database tetap dipertahankan               |

**Sketsa implementasi Python - Bab 3 (pseudocode original)**

```

key = canonical_key(smiles, charge, multiplicity)
if key in seen:
    duplicates.append(key)
else:
    seen.add(key)

```

**3.6 Rangkuman Bab**

Bab 3 menunjukkan bahwa validasi struktur, normalisasi, charge, multiplicity, dan deduplikasi harus dipahami sebagai satu sistem yang menghubungkan keputusan ilmiah dan keputusan perangkat lunak. Keandalan QMScreen bergantung pada kemampuan setiap tahap untuk menerima input yang jelas, menjalankan metode dengan parameter eksplisit, mencatat status, mempertahankan hasil parsial secara benar, dan menyediakan provenance. Dengan pola ini, batch yang besar tetap

dapat diaudit dan dikembangkan tanpa menjadikan GUI sebagai satu-satunya sumber kebenaran. Pembaca sebaiknya menggunakan checklist pada setiap subbab sebagai kendali sebelum melanjutkan ke tahap berikutnya.

## BAB 4 GENERATE 3D, ETKDGV3, MMFF94/UFF, DAN CONFORMER SCREENING

Bab ini membahas generate 3d, etkdgv3, mmff94/uff, dan conformer screening sebagai bagian dari workflow QMScreen V1.0.5. Pembahasan menempatkan konsep kimia komputasi, implementasi perangkat lunak, scientific guardrail, quality control, dan reproducibility dalam satu alur. Setiap subbab diakhiri dengan checklist praktis agar pembaca dapat menghubungkan teori dengan penggunaan aplikasi.

**Funnel Conformer**



*Gambar 4.1. Peta konseptual generate 3d, etkdgv3, mmff94/uff, dan conformer screening.*

**Tabel 4.1. Peta materi Bab 4**

| Subbab | Fokus                           | Keluaran utama                        |
|--------|---------------------------------|---------------------------------------|
| 4.1    | Mengapa geometri awal penting   | ensemble conformer awal               |
| 4.2    | ETKDGv3 untuk embedding         | daftar conformer dan status embedding |
| 4.3    | Pre-optimalisasi MMFF94 dan UFF | energi relatif dan ranking conformer  |
| 4.4    | Seleksi conformer               | subset conformer terpilih             |
| 4.5    | QC conformer batch              | log conformer, final chosen geometry  |

### 4.1 Mengapa geometri awal penting

Pembahasan mengapa geometri awal penting dimulai dari prinsip bahwa satu angka komputasi hanya bermakna jika struktur, metode, unit, dan status validasinya diketahui. Metode kuantum mengoptimasi dari titik awal tertentu; struktur 3D yang buruk dapat memperlambat atau mengarahkan optimasi ke

minimum yang tidak diinginkan. Elemen yang paling relevan pada tahap ini meliputi embed 3D, stereokimia, hidrogen dan random seed. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (RDKit Documentation, 2026).

Implementasi perangkat lunak untuk bagian ini mengikuti pola parse/validate/execute/record. Input terlebih dahulu dibaca, kemudian diverifikasi, setelah itu engine dijalankan, dan akhirnya keluaran dipaketkan menjadi result record. Parameter yang terkait mencakup embed 3D, 'stereokimia', 'hidrogen', random seed. Keluaran yang diharapkan adalah ensemble conformer awal. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

#### ***4.1.1 Prinsip, parameter, dan interpretasi***

Quality control harus membedakan kondisi tidak valid, kondisi meragukan, dan kondisi yang sekadar berbeda dari default. Aturan utama yang perlu dipertahankan adalah bahwa konektivitas valid tidak menjamin geometri 3D layak; clash dan koordinat aneh harus dideteksi. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Strategi operasional yang aman untuk mengapa geometri awal penting adalah memulai dari konfigurasi sederhana, memeriksa log, lalu menambah kompleksitas satu parameter pada satu waktu. Setelah tahap selesai, periksa apakah ensemble conformer awal benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### 4.1.2 Checklist implementasi dan quality control

- Pastikan parameter embed 3D telah ditentukan dan disimpan.
- Pastikan parameter stereokimia telah ditentukan dan disimpan.
- Pastikan parameter hidrogen telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: konektivitas valid tidak menjamin geometri 3D layak; clash dan koordinat aneh harus dideteksi.
- Pastikan keluaran tahap tersedia sebagai ensemble conformer awal dan status QC tercatat.

## 4.2 ETKDGv3 untuk embedding

Pada QMScreen, etkdgv3 untuk embedding bukan fitur dekoratif antarmuka. ETKDG menggabungkan distance geometry dan pengetahuan torsional untuk menghasilkan conformer yang lebih realistis bagi banyak molekul organik. Pendekatan ini membuat tahap tersebut dapat diuji sebagai komponen ilmiah yang mandiri. Elemen yang paling relevan pada tahap ini meliputi jumlah conformer, seed, prune threshold dan embedding failures. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (RDKit Documentation, 2026).

Secara arsitektural, bagian etkdgv3 untuk embedding ditempatkan di luar logika tampilan sebanyak mungkin. Data dari widget hanya menjadi sumber konfigurasi, sedangkan fungsi inti bekerja pada struktur Python yang dapat diuji. Titik kendalinya meliputi jumlah conformer, `seed`, prune threshold, embedding failures. Keluaran yang diharapkan adalah daftar conformer dan status embedding. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### 4.2.1 Prinsip, parameter, dan interpretasi

Risiko terbesar dalam otomasi adalah membuat proses tampak berhasil ketika asumsi ilmiahnya tidak terpenuhi. Aturan utama yang perlu dipertahankan adalah bahwa jumlah conformer adalah kompromi coverage dan biaya; tidak ada angka universal. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status

QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Untuk tujuan riset, hasil dari etkdgv3 untuk embedding sebaiknya ditinjau bersama data sebelum dan sesudah tahap tersebut agar perubahan dapat dilacak. Setelah tahap selesai, periksa apakah daftar conformer dan status embedding benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

#### ***4.2.2 Checklist implementasi dan quality control***

- Pastikan parameter jumlah conformer telah ditentukan dan disimpan.
- Pastikan parameter seed telah ditentukan dan disimpan.
- Pastikan parameter prune threshold telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: jumlah conformer adalah kompromi coverage dan biaya; tidak ada angka universal.
- Pastikan keluaran tahap tersedia sebagai daftar conformer dan status embedding dan status QC tercatat.

### **4.3 Pre-optimasi MMFF94 dan UFF**

High-throughput chemistry menuntut aturan yang konsisten ketika jumlah molekul bertambah. Dalam konteks pre-optimasi mmff94 dan uff, Force field murah dipakai untuk mengurangi strain sebelum prescreen semiempirik atau DFT. Elemen yang paling relevan pada tahap ini meliputi MMFF94 bila parameter tersedia dan UFF sebagai alternatif. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (RDKit Documentation, 2026).

Untuk batch besar, implementasi harus tahan terhadap kegagalan parsial. QMScreen karena itu menyimpan status per molekul dan per tahap, sambil

mempertahankan parameter seperti MMFF94 bila parameter tersedia, UFF sebagai alternatif pada manifest atau provenance. Keluaran yang diharapkan adalah energi relatif dan ranking conformer. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

#### ***4.3.1 Prinsip, parameter, dan interpretasi***

Ketika workflow dipercepat, kontrol ilmiah tidak boleh ikut dipangkas. Aturan utama yang perlu dipertahankan adalah bahwa energi force field hanya untuk seleksi internal tahap awal, bukan pengganti energi elektronik DFT. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Pada komputer dengan resource terbatas, pre-optimasi mmff94 dan uff perlu diatur bersama jumlah worker, thread, memory, dan biaya tahap berikutnya. Setelah tahap selesai, periksa apakah energi relatif dan ranking conformer benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

#### ***4.3.2 Checklist implementasi dan quality control***

- Pastikan parameter MMFF94 bila parameter tersedia telah ditentukan dan disimpan.
- Pastikan parameter UFF sebagai alternatif telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: energi force field hanya untuk seleksi internal tahap awal, bukan pengganti energi elektronik DFT.
- Pastikan keluaran tahap tersedia sebagai energi relatif dan ranking conformer dan status QC tercatat.

## 4.4 Seleksi conformer

Conformer dapat dipilih berdasarkan energi, keragaman, dan batas jumlah kandidat yang diteruskan. Dalam workflow komputasi skala besar, konsep ini perlu didefinisikan sebelum job pertama dijalankan karena keputusan awal akan memengaruhi biaya, keterbandingan, dan kemampuan reproduksi hasil. Elemen yang paling relevan pada tahap ini meliputi energy window, RMSD concept dan top-N. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (RDKit Documentation, 2026).

Di tingkat implementasi, QMScreen mengubah pilihan pengguna menjadi record konfigurasi dan menjalankan fungsi scientific logic yang sesuai. Komponen seperti energy window, RMSD concept, 'top-N' tidak diperlakukan sebagai nilai sementara, tetapi sebagai bagian dari state proyek yang dapat ditulis ke disk. Keluaran yang diharapkan adalah subset conformer terpilih. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### 4.4.1 Prinsip, parameter, dan interpretasi

Scientific guardrail pada bagian ini sengaja dibuat konservatif. Aturan utama yang perlu dipertahankan adalah bahwa seleksi terlalu agresif dapat kehilangan basin penting; terlalu longgar memboroskan resource. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Dalam penggunaan sehari-hari, pengguna sebaiknya menguji seleksi conformer pada satu atau dua molekul kecil sebelum memperbesar batch. Setelah tahap selesai, periksa apakah subset conformer terpilih benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang

dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

#### ***4.4.2 Checklist implementasi dan quality control***

- Pastikan parameter energy window telah ditentukan dan disimpan.
- Pastikan parameter RMSD concept telah ditentukan dan disimpan.
- Pastikan parameter top-N telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: seleksi terlalu agresif dapat kehilangan basin penting; terlalu longgar memboroskan resource.
- Pastikan keluaran tahap tersedia sebagai subset conformer terpilih dan status QC tercatat.

### **4.5 QC conformer batch**

Pembahasan qc conformer batch dimulai dari prinsip bahwa satu angka komputasi hanya bermakna jika struktur, metode, unit, dan status validasinya diketahui. Batch besar membutuhkan pencatatan conformer yang gagal tanpa menggagalkan molekul lain. Elemen yang paling relevan pada tahap ini meliputi exception per conformer, fallback, seed alternatif dan checkpoint. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (RDKit Documentation, 2026).

Implementasi perangkat lunak untuk bagian ini mengikuti pola parse/validate/execute/record. Input terlebih dahulu dibaca, kemudian diverifikasi, setelah itu engine dijalankan, dan akhirnya keluaran dipaketkan menjadi result record. Parameter yang terkait mencakup exception per conformer, 'fallback', seed alternatif, 'checkpoint'. Keluaran yang diharapkan adalah log conformer, final chosen geometry. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

#### ***4.5.1 Prinsip, parameter, dan interpretasi***

Quality control harus membedakan kondisi tidak valid, kondisi meragukan, dan kondisi yang sekadar berbeda dari default. Aturan utama yang perlu dipertahankan

adalah bahwa kegagalan satu conformer harus dipisahkan dari kegagalan struktur keseluruhan. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Strategi operasional yang aman untuk qc conformer batch adalah memulai dari konfigurasi sederhana, memeriksa log, lalu menambah kompleksitas satu parameter pada satu waktu. Setelah tahap selesai, periksa apakah log conformer, final chosen geometry benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

#### ***4.5.2 Checklist implementasi dan quality control***

- Pastikan parameter exception per conformer telah ditentukan dan disimpan.
- Pastikan parameter fallback telah ditentukan dan disimpan.
- Pastikan parameter seed alternatif telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: kegagalan satu conformer harus dipisahkan dari kegagalan struktur keseluruhan.
- Pastikan keluaran tahap tersedia sebagai log conformer, final chosen geometry dan status QC tercatat.

**Sketsa implementasi Python - Bab 4 (pseudocode original)**

```
conformers = embed_3d(molecule, n=n_conf)
scored = preoptimize_forcefield(conformers)
selected = choose_low_energy(scored)
```

## **4.6 Rangkuman Bab**

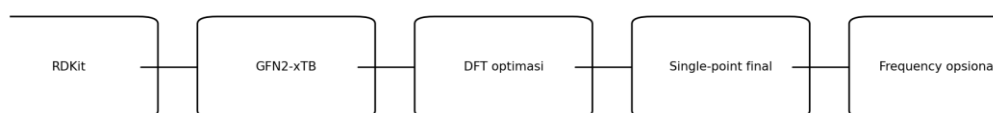
Bab 4 menunjukkan bahwa generate 3d, etkdgv3, mmff94/uff, dan conformer screening harus dipahami sebagai satu sistem yang menghubungkan keputusan ilmiah dan keputusan perangkat lunak. Keandalan QMScreen bergantung pada

kemampuan setiap tahap untuk menerima input yang jelas, menjalankan metode dengan parameter eksplisit, mencatat status, mempertahankan hasil parsial secara benar, dan menyediakan provenance. Dengan pola ini, batch yang besar tetap dapat diaudit dan dikembangkan tanpa menjadikan GUI sebagai satu-satunya sumber kebenaran. Pembaca sebaiknya menggunakan checklist pada setiap subbab sebagai kendali sebelum melanjutkan ke tahap berikutnya.

## BAB 5 PRESCREEN GFN2-XTB DAN STRATEGI MULTI-FIDELITY

Bab ini membahas prescreen gfn2-xtb dan strategi multi-fidelity sebagai bagian dari workflow QMScreen V1.0.5. Pembahasan menempatkan konsep kimia komputasi, implementasi perangkat lunak, scientific guardrail, quality control, dan reproducibility dalam satu alur. Setiap subbab diakhiri dengan checklist praktis agar pembaca dapat menghubungkan teori dengan penggunaan aplikasi.

### Strategi Multi-Fidelity



Gambar 5.1. Peta konseptual prescreen gfn2-xtb dan strategi multi-fidelity.

Tabel 5.1. Peta materi Bab 5

| Subbab | Fokus                                    | Keluaran utama                         |
|--------|------------------------------------------|----------------------------------------|
| 5.1    | Peran GFN2-xTB                           | energi prescreen dan status backend    |
| 5.2    | TBLite dan fallback CLI                  | record backend dan raw output          |
| 5.3    | Seleksi kandidat setelah prescreen       | subset kandidat menuju DFT             |
| 5.4    | Failure mode prescreen                   | status prescreen dan tindakan korektif |
| 5.5    | Desain multi-fidelity yang dapat diaudit | audit trail seleksi kandidat           |

### 5.1 Peran GFN2-xTB

Pembahasan peran gfn2-xtb dimulai dari prinsip bahwa satu angka komputasi hanya bermakna jika struktur, metode, unit, dan status validasinya diketahui. GFN2-xTB menyediakan perhitungan tight-binding yang jauh lebih murah dibanding DFT dan dapat digunakan sebagai prescreen. Elemen yang paling relevan pada tahap ini meliputi TBLite Python backend, xTB CLI fallback dan energi single-point. Masing-masing elemen harus memiliki representasi data yang

eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Bannwarth, Ehlert, & Grimme, 2019).

Implementasi perangkat lunak untuk bagian ini mengikuti pola parse/validate/execute/record. Input terlebih dahulu dibaca, kemudian diverifikasi, setelah itu engine dijalankan, dan akhirnya keluaran dipaketkan menjadi result record. Parameter yang terkait mencakup TBLite Python backend, xTB CLI fallback, energi single-point. Keluaran yang diharapkan adalah energi prescreen dan status backend. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### ***5.1.1 Prinsip, parameter, dan interpretasi***

Quality control harus membedakan kondisi tidak valid, kondisi meragukan, dan kondisi yang sekadar berbeda dari default. Aturan utama yang perlu dipertahankan adalah bahwa hasil xTB tidak boleh disamakan dengan energi DFT atau dipakai sebagai bukti akhir tanpa validasi. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Strategi operasional yang aman untuk peran gfn2-xtb adalah memulai dari konfigurasi sederhana, memeriksa log, lalu menambah kompleksitas satu parameter pada satu waktu. Setelah tahap selesai, periksa apakah energi prescreen dan status backend benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***5.1.2 Checklist implementasi dan quality control***

- Pastikan parameter TBLite Python backend telah ditentukan dan disimpan.
- Pastikan parameter xTB CLI fallback telah ditentukan dan disimpan.
- Pastikan parameter energi single-point telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: hasil xTB tidak boleh disamakan dengan energi DFT atau dipakai sebagai bukti akhir tanpa validasi.
- Pastikan keluaran tahap tersedia sebagai energi prescreen dan status backend dan status QC tercatat.

## **5.2 TBLite dan fallback CLI**

Pada QMScreen, tblite dan fallback cli bukan fitur dekoratif antarmuka. Implementasi memilih backend Python ketika tersedia dan dapat menggunakan xTB CLI sebagai fallback. Pendekatan ini membuat tahap tersebut dapat diuji sebagai komponen ilmiah yang mandiri. Elemen yang paling relevan pada tahap ini meliputi deteksi backend, atomic numbers, coordinates, charge dan uhf. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Bannwarth, Ehlert, & Grimme, 2019).

Secara arsitektural, bagian tblite dan fallback cli ditempatkan di luar logika tampilan sebanyak mungkin. Data dari widget hanya menjadi sumber konfigurasi, sedangkan fungsi inti bekerja pada struktur Python yang dapat diuji. Titik kendalinya meliputi deteksi backend, atomic numbers, 'coordinates', 'charge', 'uhf'. Keluaran yang diharapkan adalah record backend dan raw output. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### ***5.2.1 Prinsip, parameter, dan interpretasi***

Risiko terbesar dalam otomasi adalah membuat proses tampak berhasil ketika asumsi ilmiahnya tidak terpenuhi. Aturan utama yang perlu dipertahankan adalah bahwa fallback harus mempertahankan unit, parameter, dan parser yang konsisten. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini

juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Untuk tujuan riset, hasil dari `tblite` dan `fallback cli` sebaiknya ditinjau bersama data sebelum dan sesudah tahap tersebut agar perubahan dapat dilacak. Setelah tahap selesai, periksa apakah record backend dan raw output benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***5.2.2 Checklist implementasi dan quality control***

- Pastikan parameter deteksi backend telah ditentukan dan disimpan.
- Pastikan parameter atomic numbers telah ditentukan dan disimpan.
- Pastikan parameter coordinates telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: fallback harus mempertahankan unit, parameter, dan parser yang konsisten.
- Pastikan keluaran tahap tersedia sebagai record backend dan raw output dan status QC tercatat.

## **5.3 Seleksi kandidat setelah prescreen**

High-throughput chemistry menuntut aturan yang konsisten ketika jumlah molekul bertambah. Dalam konteks seleksi kandidat setelah prescreen, Prescreen dapat mengurutkan conformer atau kandidat sebelum DFT untuk memfokuskan biaya komputasi. Elemen yang paling relevan pada tahap ini meliputi ranking energi internal, top-N dan threshold relatif. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Bannwarth, Ehlert, & Grimme, 2019).

Untuk batch besar, implementasi harus tahan terhadap kegagalan parsial. QMScreen karena itu menyimpan status per molekul dan per tahap, sambil mempertahankan parameter seperti ranking energi internal, 'top-N', threshold

relatif pada manifest atau provenance. Keluaran yang diharapkan adalah subset kandidat menuju DFT. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### ***5.3.1 Prinsip, parameter, dan interpretasi***

Ketika workflow dipercepat, kontrol ilmiah tidak boleh ikut dipangkas. Aturan utama yang perlu dipertahankan adalah bahwa energi total antar formula berbeda tidak boleh dipakai sebagai skor lintas komposisi. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Pada komputer dengan resource terbatas, seleksi kandidat setelah prescreen perlu diatur bersama jumlah worker, thread, memory, dan biaya tahap berikutnya. Setelah tahap selesai, periksa apakah subset kandidat menuju DFT benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***5.3.2 Checklist implementasi dan quality control***

- Pastikan parameter ranking energi internal telah ditentukan dan disimpan.
- Pastikan parameter top-N telah ditentukan dan disimpan.
- Pastikan parameter threshold relatif telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: energi total antar formula berbeda tidak boleh dipakai sebagai skor lintas komposisi.
- Pastikan keluaran tahap tersedia sebagai subset kandidat menuju DFT dan status QC tercatat.

## 5.4 Failure mode prescreen

Kegagalan dapat berasal dari struktur, charge, backend, elemen, atau proses eksternal. Dalam workflow komputasi skala besar, konsep ini perlu didefinisikan sebelum job pertama dijalankan karena keputusan awal akan memengaruhi biaya, keterbandingan, dan kemampuan reproduksi hasil. Elemen yang paling relevan pada tahap ini meliputi QMS-E004, timeout concept dan parser energy. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Bannwarth, Ehlert, & Grimme, 2019).

Di tingkat implementasi, QMScreen mengubah pilihan pengguna menjadi record konfigurasi dan menjalankan fungsi scientific logic yang sesuai. Komponen seperti 'QMS-E004', timeout concept, parser energy tidak diperlakukan sebagai nilai sementara, tetapi sebagai bagian dari state proyek yang dapat ditulis ke disk. Keluaran yang diharapkan adalah status prescreen dan tindakan korektif. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### 5.4.1 Prinsip, parameter, dan interpretasi

Scientific guardrail pada bagian ini sengaja dibuat konservatif. Aturan utama yang perlu dipertahankan adalah bahwa opsi menonaktifkan prescreen berguna untuk diagnosis tetapi harus dicatat di provenance. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Dalam penggunaan sehari-hari, pengguna sebaiknya menguji failure mode prescreen pada satu atau dua molekul kecil sebelum memperbesar batch. Setelah tahap selesai, periksa apakah status prescreen dan tindakan korektif benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan

mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

#### ***5.4.2 Checklist implementasi dan quality control***

- Pastikan parameter QMS-E004 telah ditentukan dan disimpan.
- Pastikan parameter timeout concept telah ditentukan dan disimpan.
- Pastikan parameter parser energy telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: opsi menonaktifkan prescreen berguna untuk diagnosis tetapi harus dicatat di provenance.
- Pastikan keluaran tahap tersedia sebagai status prescreen dan tindakan korektif dan status QC tercatat.

### **5.5 Desain multi-fidelity yang dapat diaudit**

Pembahasan desain multi-fidelity yang dapat diaudit dimulai dari prinsip bahwa satu angka komputasi hanya bermakna jika struktur, metode, unit, dan status validasinya diketahui. Setiap tahap murah harus diperlakukan sebagai filter dengan alasan yang eksplisit. Elemen yang paling relevan pada tahap ini meliputi stage record, threshold, method label dan hash. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Bannwarth, Ehlert, & Grimme, 2019).

Implementasi perangkat lunak untuk bagian ini mengikuti pola parse/validate/execute/record. Input terlebih dahulu dibaca, kemudian diverifikasi, setelah itu engine dijalankan, dan akhirnya keluaran dipaketkan menjadi result record. Parameter yang terkait mencakup stage record, `threshold`, method label, `hash`. Keluaran yang diharapkan adalah audit trail seleksi kandidat. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

#### ***5.5.1 Prinsip, parameter, dan interpretasi***

Quality control harus membedakan kondisi tidak valid, kondisi meragukan, dan kondisi yang sekadar berbeda dari default. Aturan utama yang perlu dipertahankan adalah bahwa jangan mengubah threshold setelah melihat hasil tanpa mencatat

perubahan karena dapat menciptakan bias seleksi. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Strategi operasional yang aman untuk desain multi-fidelity yang dapat diaudit adalah memulai dari konfigurasi sederhana, memeriksa log, lalu menambah kompleksitas satu parameter pada satu waktu. Setelah tahap selesai, periksa apakah audit trail seleksi kandidat benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***5.5.2 Checklist implementasi dan quality control***

- Pastikan parameter stage record telah ditentukan dan disimpan.
- Pastikan parameter threshold telah ditentukan dan disimpan.
- Pastikan parameter method label telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: jangan mengubah threshold setelah melihat hasil tanpa mencatat perubahan karena dapat menciptakan bias seleksi.
- Pastikan keluaran tahap tersedia sebagai audit trail seleksi kandidat dan status QC tercatat.

**Sketsa implementasi Python - Bab 5 (pseudocode original)**

```
if tblite_available():
    energy = gfn2_singlepoint(structure)
else:
    energy = xtb_cli_singlepoint(structure)
```

## **5.6 Rangkuman Bab**

Bab 5 menunjukkan bahwa prescreen gfn2-xtb dan strategi multi-fidelity harus dipahami sebagai satu sistem yang menghubungkan keputusan ilmiah dan

keputusan perangkat lunak. Keandalan QMScreen bergantung pada kemampuan setiap tahap untuk menerima input yang jelas, menjalankan metode dengan parameter eksplisit, mencatat status, mempertahankan hasil parsial secara benar, dan menyediakan provenance. Dengan pola ini, batch yang besar tetap dapat diaudit dan dikembangkan tanpa menjadikan GUI sebagai satu-satunya sumber kebenaran. Pembaca sebaiknya menggunakan checklist pada setiap subbab sebagai kendali sebelum melanjutkan ke tahap berikutnya.

## BAB 6 DFT PYSCF, FUNCTIONAL, DAN BASIS SET BERTINGKAT

Bab ini membahas dft pyscf, functional, dan basis set bertingkat sebagai bagian dari workflow QMScreen V1.0.5. Pembahasan menempatkan konsep kimia komputasi, implementasi perangkat lunak, scientific guardrail, quality control, dan reproducibility dalam satu alur. Setiap subbab diakhiri dengan checklist praktis agar pembaca dapat menghubungkan teori dengan penggunaan aplikasi.

**Basis Bertingkat dan DFT**



*Gambar 6.1. Peta konseptual dft pyscf, functional, dan basis set bertingkat.*

**Tabel 6.1. Peta materi Bab 6**

| Subbab | Fokus                           | Keluaran utama                           |
|--------|---------------------------------|------------------------------------------|
| 6.1    | PySCF sebagai engine elektronik | energi SCF, orbital, density matrix, log |
| 6.2    | Functional yang tersedia        | method label dan parameter XC            |
| 6.3    | Basis optimasi dan basis final  | basis_opt dan basis_final                |
| 6.4    | SCF convergence controls        | converged flag dan QMS-E005              |
| 6.5    | Density fitting dan resource    | resource record dan runtime              |

### 6.1 PySCF sebagai engine elektronik

Pembahasan pyscf sebagai engine elektronik dimulai dari prinsip bahwa satu angka komputasi hanya bermakna jika struktur, metode, unit, dan status validasinya diketahui. QMScreen menggunakan PySCF untuk membangun molekul, menjalankan SCF/DFT, dan menurunkan besaran elektronik. Elemen yang paling relevan pada tahap ini meliputi gto, dft, scf, memory, threads dan density fitting. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila

relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Sun et al., 2018; Sun et al., 2020).

Implementasi perangkat lunak untuk bagian ini mengikuti pola parse/validate/execute/record. Input terlebih dahulu dibaca, kemudian diverifikasi, setelah itu engine dijalankan, dan akhirnya keluaran dipaketkan menjadi result record. Parameter yang terkait mencakup `gto`, `dft`, `scf`, `memory`, `threads`, density fitting. Keluaran yang diharapkan adalah energi SCF, orbital, density matrix, log. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### ***6.1.1 Prinsip, parameter, dan interpretasi***

Quality control harus membedakan kondisi tidak valid, kondisi meragukan, dan kondisi yang sekadar berbeda dari default. Aturan utama yang perlu dipertahankan adalah bahwa engine harus menerima charge, spin, basis, functional, dan toleransi yang tersimpan di job specification. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Strategi operasional yang aman untuk pyscf sebagai engine elektronik adalah memulai dari konfigurasi sederhana, memeriksa log, lalu menambah kompleksitas satu parameter pada satu waktu. Setelah tahap selesai, periksa apakah energi SCF, orbital, density matrix, log benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***6.1.2 Checklist implementasi dan quality control***

- Pastikan parameter gto telah ditentukan dan disimpan.

- Pastikan parameter dft telah ditentukan dan disimpan.
- Pastikan parameter scf telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: engine harus menerima charge, spin, basis, functional, dan toleransi yang tersimpan di job specification.
- Pastikan keluaran tahap tersedia sebagai energi SCF, orbital, density matrix, log dan status QC tercatat.

## 6.2 Functional yang tersedia

Pada QMScreen, functional yang tersedia bukan fitur dekoratif antarmuka. Antarmuka menyediakan B3LYP, PBE0, M06-2X, CAM-B3LYP, dan WB97X sebagai pilihan DFT. Pendekatan ini membuat tahap tersebut dapat diuji sebagai komponen ilmiah yang mandiri. Elemen yang paling relevan pada tahap ini meliputi hybrid, meta-hybrid dan range-separated concept. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Sun et al., 2018; Sun et al., 2020).

Secara arsitektural, bagian functional yang tersedia ditempatkan di luar logika tampilan sebanyak mungkin. Data dari widget hanya menjadi sumber konfigurasi, sedangkan fungsi inti bekerja pada struktur Python yang dapat diuji. Titik kendalinya meliputi 'hybrid', 'meta-hybrid', range-separated concept. Keluaran yang diharapkan adalah method label dan parameter XC. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### 6.2.1 Prinsip, parameter, dan interpretasi

Risiko terbesar dalam otomasi adalah membuat proses tampak berhasil ketika asumsi ilmiahnya tidak terpenuhi. Aturan utama yang perlu dipertahankan adalah bahwa functional tidak dipilih berdasarkan popularitas saja; domain kimia dan properti target menentukan validasi. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam

laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Untuk tujuan riset, hasil dari functional yang tersedia sebaiknya ditinjau bersama data sebelum dan sesudah tahap tersebut agar perubahan dapat dilacak. Setelah tahap selesai, periksa apakah method label dan parameter XC benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### 6.2.2 Checklist implementasi dan quality control

- Pastikan parameter hybrid telah ditentukan dan disimpan.
- Pastikan parameter meta-hybrid telah ditentukan dan disimpan.
- Pastikan parameter range-separated concept telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: functional tidak dipilih berdasarkan popularitas saja; domain kimia dan properti target menentukan validasi.
- Pastikan keluaran tahap tersedia sebagai method label dan parameter XC dan status QC tercatat.

**Tabel 6.2. Functional DFT yang tersedia pada antarmuka**

| Functional | Keluarga umum          | Catatan penggunaan                                          |
|------------|------------------------|-------------------------------------------------------------|
| B3LYP      | Hybrid GGA             | Pilihan umum; tetap perlu validasi untuk properti target    |
| PBE0       | Hybrid GGA             | Parameter-free hybrid berdasarkan PBE                       |
| M06-2X     | Hybrid meta-GGA        | Pertukaran nonlokal tinggi; banyak dipakai untuk main-group |
| CAM-B3LYP  | Range-separated hybrid | Memisahkan kontribusi jarak pendek/panjang                  |
| WB97X      | Range-separated hybrid | Functional long-range corrected                             |

### 6.3 Basis optimasi dan basis final

High-throughput chemistry menuntut aturan yang konsisten ketika jumlah molekul bertambah. Dalam konteks basis optimasi dan basis final, Basis bertingkat memisahkan biaya optimasi dari akurasi single-point final. Elemen yang paling relevan pada tahap ini meliputi STO-3G sampai def2-QZVP, cc-pVXZ dan aug-cc-pVDZ. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Sun et al., 2018; Sun et al., 2020).

Untuk batch besar, implementasi harus tahan terhadap kegagalan parsial. QMScreen karena itu menyimpan status per molekul dan per tahap, sambil mempertahankan parameter seperti STO-3G sampai def2-QZVP, `cc-pVXZ`, `aug-cc-pVDZ` pada manifest atau provenance. Keluaran yang diharapkan adalah `basis_opt` dan `basis_final`. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

#### 6.3.1 Prinsip, parameter, dan interpretasi

Ketika workflow dipercepat, kontrol ilmiah tidak boleh ikut dipangkas. Aturan utama yang perlu dipertahankan adalah bahwa basis lebih besar meningkatkan biaya dan tidak selalu memperbaiki semua observabel secara monoton. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Pada komputer dengan resource terbatas, basis optimasi dan basis final perlu diatur bersama jumlah worker, thread, memory, dan biaya tahap berikutnya. Setelah tahap selesai, periksa apakah `basis_opt` dan `basis_final` benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil

akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### 6.3.2 Checklist implementasi dan quality control

- Pastikan parameter STO-3G sampai def2-QZVP telah ditentukan dan disimpan.
- Pastikan parameter cc-pVXZ telah ditentukan dan disimpan.
- Pastikan parameter aug-cc-pVDZ telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: basis lebih besar meningkatkan biaya dan tidak selalu memperbaiki semua observabel secara monoton.
- Pastikan keluaran tahap tersedia sebagai basis\_opt dan basis\_final dan status QC tercatat.

**Tabel 6.3.** Basis set yang ditawarkan QMScreen

| Kelompok               | Pilihan                                               |
|------------------------|-------------------------------------------------------|
| Minimal/split-valence  | STO-3G; 3-21G; 6-31G; 6-31G*; 6-31G**                 |
| Karlsruhe def2         | def2-SVP; def2-SVPD; def2-TZVP; def2-TZVPP; def2-QZVP |
| Correlation-consistent | cc-pVDZ; cc-pVTZ; aug-cc-pVDZ                         |

## 6.4 SCF convergence controls

Konvergensi SCF dipengaruhi struktur, metode, spin, dan parameter numerik. Dalam workflow komputasi skala besar, konsep ini perlu didefinisikan sebelum job pertama dijalankan karena keputusan awal akan memengaruhi biaya, keterbandingan, dan kemampuan reproduksi hasil. Elemen yang paling relevan pada tahap ini meliputi conv\_tol 1e-3 sampai 1e-10, max cycle, grid level dan fallback. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Sun et al., 2018; Sun et al., 2020).

Di tingkat implementasi, QMScreen mengubah pilihan pengguna menjadi record konfigurasi dan menjalankan fungsi scientific logic yang sesuai. Komponen seperti `conv_tol`  $1e-3$  sampai  $1e-10$ , `max cycle`, `grid level`, `'fallback'` tidak diperlakukan sebagai nilai sementara, tetapi sebagai bagian dari state proyek yang dapat ditulis ke disk. Keluaran yang diharapkan adalah converged flag dan QMS-E005. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

#### ***6.4.1 Prinsip, parameter, dan interpretasi***

Scientific guardrail pada bagian ini sengaja dibuat konservatif. Aturan utama yang perlu dipertahankan adalah bahwa SCF yang tidak konvergen tidak boleh dilabeli sukses hanya karena menghasilkan angka energi terakhir. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Dalam penggunaan sehari-hari, pengguna sebaiknya menguji scf convergence controls pada satu atau dua molekul kecil sebelum memperbesar batch. Setelah tahap selesai, periksa apakah converged flag dan QMS-E005 benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

#### ***6.4.2 Checklist implementasi dan quality control***

- Pastikan parameter `conv_tol`  $1e-3$  sampai  $1e-10$  telah ditentukan dan disimpan.
- Pastikan parameter `max cycle` telah ditentukan dan disimpan.
- Pastikan parameter `grid level` telah ditentukan dan disimpan.

- Pastikan guardrail berikut dipenuhi: SCF yang tidak konvergen tidak boleh dilabeli sukses hanya karena menghasilkan angka energi terakhir.
- Pastikan keluaran tahap tersedia sebagai converged flag dan QMS-E005 dan status QC tercatat.

## 6.5 Density fitting dan resource

Pembahasan density fitting dan resource dimulai dari prinsip bahwa satu angka komputasi hanya bermakna jika struktur, metode, unit, dan status validasinya diketahui. Density fitting dapat menurunkan biaya beberapa langkah tetapi perlu dicatat sebagai bagian metode. Elemen yang paling relevan pada tahap ini meliputi RI/DF, memory per job dan CPU threads. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Sun et al., 2018; Sun et al., 2020).

Implementasi perangkat lunak untuk bagian ini mengikuti pola parse/validate/execute/record. Input terlebih dahulu dibaca, kemudian diverifikasi, setelah itu engine dijalankan, dan akhirnya keluaran dipaketkan menjadi result record. Parameter yang terkait mencakup 'RI/DF', memory per job, CPU threads. Keluaran yang diharapkan adalah resource record dan runtime. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### 6.5.1 Prinsip, parameter, dan interpretasi

Quality control harus membedakan kondisi tidak valid, kondisi meragukan, dan kondisi yang sekadar berbeda dari default. Aturan utama yang perlu dipertahankan adalah bahwa optimasi performa tidak boleh mengubah definisi metode tanpa metadata. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada

ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Strategi operasional yang aman untuk density fitting dan resource adalah memulai dari konfigurasi sederhana, memeriksa log, lalu menambah kompleksitas satu parameter pada satu waktu. Setelah tahap selesai, periksa apakah resource record dan runtime benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### 6.5.2 Checklist implementasi dan quality control

- Pastikan parameter RI/DF telah ditentukan dan disimpan.
- Pastikan parameter memory per job telah ditentukan dan disimpan.
- Pastikan parameter CPU threads telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: optimasi performa tidak boleh mengubah definisi metode tanpa metadata.
- Pastikan keluaran tahap tersedia sebagai resource record dan runtime dan status QC tercatat.

#### Sketsa implementasi Python - Bab 6 (pseudocode original)

```
mf = build_dft(molecule, functional, basis_opt)
energy_opt = optimize_geometry(mf)
energy_final = singlepoint(basis_final)
```

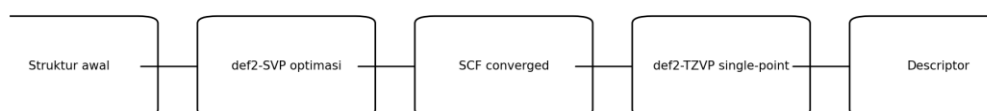
## 6.6 Rangkuman Bab

Bab 6 menunjukkan bahwa dft pyscf, functional, dan basis set bertingkat harus dipahami sebagai satu sistem yang menghubungkan keputusan ilmiah dan keputusan perangkat lunak. Keandalan QMScreen bergantung pada kemampuan setiap tahap untuk menerima input yang jelas, menjalankan metode dengan parameter eksplisit, mencatat status, mempertahankan hasil parsial secara benar, dan menyediakan provenance. Dengan pola ini, batch yang besar tetap dapat diaudit dan dikembangkan tanpa menjadikan GUI sebagai satu-satunya sumber kebenaran. Pembaca sebaiknya menggunakan checklist pada setiap subbab sebagai kendali sebelum melanjutkan ke tahap berikutnya.

## BAB 7 OPTIMASI GEOMETRI, GEOMETRIC, DAN KONTROL KONVERGENSI

Bab ini membahas optimasi geometri, geometric, dan kontrol konvergensi sebagai bagian dari workflow QMScreen V1.0.5. Pembahasan menempatkan konsep kimia komputasi, implementasi perangkat lunak, scientific guardrail, quality control, dan reproducibility dalam satu alur. Setiap subbab diakhiri dengan checklist praktis agar pembaca dapat menghubungkan teori dengan penggunaan aplikasi.

### Basis Bertingkat dan DFT



Gambar 7.1. Peta konseptual optimasi geometri, geometric, dan kontrol konvergensi.

Tabel 7.1. Peta materi Bab 7

| Subbab | Fokus                                     | Keluaran utama                         |
|--------|-------------------------------------------|----------------------------------------|
| 7.1    | Optimasi geometri sebagai masalah numerik | geometri final dan status optimasi     |
| 7.2    | Integrasi geomeTRIC                       | trajectory dan convergence metadata    |
| 7.3    | Basis bertingkat selama optimasi          | energy_opt dan energy_final            |
| 7.4    | Optimasi gagal dan recovery               | checkpoint optimasi dan failure record |
| 7.5    | Pemeriksaan geometri akhir                | final geometry dan preview             |

### 7.1 Optimasi geometri sebagai masalah numerik

Pembahasan optimasi geometri sebagai masalah numerik dimulai dari prinsip bahwa satu angka komputasi hanya bermakna jika struktur, metode, unit, dan status validasinya diketahui. Optimasi mencari titik stasioner pada permukaan energi potensial dengan menggunakan energi dan gradien. Elemen yang paling relevan pada tahap ini meliputi iteration, gradient norm, step dan convergence.

Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Wang, Song, & Martínez, 2016).

Implementasi perangkat lunak untuk bagian ini mengikuti pola parse/validate/execute/record. Input terlebih dahulu dibaca, kemudian diverifikasi, setelah itu engine dijalankan, dan akhirnya keluaran dipaketkan menjadi result record. Parameter yang terkait mencakup `iteration`, gradient norm, `step`, `convergence`. Keluaran yang diharapkan adalah geometri final dan status optimasi. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### ***7.1.1 Prinsip, parameter, dan interpretasi***

Quality control harus membedakan kondisi tidak valid, kondisi meragukan, dan kondisi yang sekadar berbeda dari default. Aturan utama yang perlu dipertahankan adalah bahwa energi menurun saja tidak cukup untuk menyatakan optimasi konvergen. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Strategi operasional yang aman untuk optimasi geometri sebagai masalah numerik adalah memulai dari konfigurasi sederhana, memeriksa log, lalu menambah kompleksitas satu parameter pada satu waktu. Setelah tahap selesai, periksa apakah geometri final dan status optimasi benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***7.1.2 Checklist implementasi dan quality control***

- Pastikan parameter iteration telah ditentukan dan disimpan.
- Pastikan parameter gradient norm telah ditentukan dan disimpan.
- Pastikan parameter step telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: energi menurun saja tidak cukup untuk menyatakan optimasi konvergen.
- Pastikan keluaran tahap tersedia sebagai geometri final dan status optimasi dan status QC tercatat.

## **7.2 Integrasi geomeTRIC**

Pada QMScreen, integrasi geometric bukan fitur dekoratif antarmuka. geomeTRIC menggunakan koordinat internal translasi-rotasi untuk optimasi yang robust dan berinteraksi dengan engine energi/gradien. Pendekatan ini membuat tahap tersebut dapat diuji sebagai komponen ilmiah yang mandiri. Elemen yang paling relevan pada tahap ini meliputi PySCF interface, max steps dan coordinates. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Wang, Song, & Martínez, 2016).

Secara arsitektural, bagian integrasi geometric ditempatkan di luar logika tampilan sebanyak mungkin. Data dari widget hanya menjadi sumber konfigurasi, sedangkan fungsi inti bekerja pada struktur Python yang dapat diuji. Titik kendalinya meliputi PySCF interface, max steps, 'coordinates'. Keluaran yang diharapkan adalah trajectory dan convergence metadata. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### ***7.2.1 Prinsip, parameter, dan interpretasi***

Risiko terbesar dalam otomasi adalah membuat proses tampak berhasil ketika asumsi ilmiahnya tidak terpenuhi. Aturan utama yang perlu dipertahankan adalah bahwa optimizer adalah lapisan numerik; kualitas metode elektronik tetap menentukan permukaan energi. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari

keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Untuk tujuan riset, hasil dari integrasi geometric sebaiknya ditinjau bersama data sebelum dan sesudah tahap tersebut agar perubahan dapat dilacak. Setelah tahap selesai, periksa apakah trajectory dan convergence metadata benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***7.2.2 Checklist implementasi dan quality control***

- Pastikan parameter PySCF interface telah ditentukan dan disimpan.
- Pastikan parameter max steps telah ditentukan dan disimpan.
- Pastikan parameter coordinates telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: optimizer adalah lapisan numerik; kualitas metode elektronik tetap menentukan permukaan energi.
- Pastikan keluaran tahap tersedia sebagai trajectory dan convergence metadata dan status QC tercatat.

## **7.3 Basis bertingkat selama optimasi**

High-throughput chemistry menuntut aturan yang konsisten ketika jumlah molekul bertambah. Dalam konteks basis bertingkat selama optimasi, Optimasi pada basis moderat diikuti single-point lebih tinggi mengurangi total biaya. Elemen yang paling relevan pada tahap ini meliputi def2-SVP -> def2-TZVP sebagai default antarmuka. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Wang, Song, & Martínez, 2016).

Untuk batch besar, implementasi harus tahan terhadap kegagalan parsial. QMScreen karena itu menyimpan status per molekul dan per tahap, sambil mempertahankan parameter seperti def2-SVP -> def2-TZVP sebagai default

antarmuka pada manifest atau provenance. Keluaran yang diharapkan adalah `energy_opt` dan `energy_final`. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### ***7.3.1 Prinsip, parameter, dan interpretasi***

Ketika workflow dipercepat, kontrol ilmiah tidak boleh ikut dipangkas. Aturan utama yang perlu dipertahankan adalah bahwa strategi dua tingkat harus divalidasi untuk sistem yang sangat fleksibel, bermuatan, atau memiliki interaksi lemah. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Pada komputer dengan resource terbatas, basis bertingkat selama optimasi perlu diatur bersama jumlah worker, thread, memory, dan biaya tahap berikutnya. Setelah tahap selesai, periksa apakah `energy_opt` dan `energy_final` benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***7.3.2 Checklist implementasi dan quality control***

- Pastikan parameter `def2-SVP` -> `def2-TZVP` sebagai default antarmuka telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: strategi dua tingkat harus divalidasi untuk sistem yang sangat fleksibel, bermuatan, atau memiliki interaksi lemah.
- Pastikan keluaran tahap tersedia sebagai `energy_opt` dan `energy_final` dan status QC tercatat.

## 7.4 Optimasi gagal dan recovery

Kegagalan dapat dipicu geometri buruk, SCF tidak stabil, batas langkah, atau masalah backend. Dalam workflow komputasi skala besar, konsep ini perlu didefinisikan sebelum job pertama dijalankan karena keputusan awal akan memengaruhi biaya, keterbandingan, dan kemampuan reproduksi hasil. Elemen yang paling relevan pada tahap ini meliputi QMS-E006, restart, seed/conformer alternatif dan max steps. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Wang, Song, & Martínez, 2016).

Di tingkat implementasi, QMScreen mengubah pilihan pengguna menjadi record konfigurasi dan menjalankan fungsi scientific logic yang sesuai. Komponen seperti 'QMS-E006', 'restart', seed/conformer alternatif, max steps tidak diperlakukan sebagai nilai sementara, tetapi sebagai bagian dari state proyek yang dapat ditulis ke disk. Keluaran yang diharapkan adalah checkpoint optimasi dan failure record. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### 7.4.1 Prinsip, parameter, dan interpretasi

Scientific guardrail pada bagian ini sengaja dibuat konservatif. Aturan utama yang perlu dipertahankan adalah bahwa recovery tidak boleh menghapus jejak upaya gagal; setiap percobaan perlu status dan log. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Dalam penggunaan sehari-hari, pengguna sebaiknya menguji optimasi gagal dan recovery pada satu atau dua molekul kecil sebelum memperbesar batch. Setelah tahap selesai, periksa apakah checkpoint optimasi dan failure record benar-benar

tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

#### ***7.4.2 Checklist implementasi dan quality control***

- Pastikan parameter QMS-E006 telah ditentukan dan disimpan.
- Pastikan parameter restart telah ditentukan dan disimpan.
- Pastikan parameter seed/conformer alternatif telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: recovery tidak boleh menghapus jejak upaya gagal; setiap percobaan perlu status dan log.
- Pastikan keluaran tahap tersedia sebagai checkpoint optimasi dan failure record dan status QC tercatat.

### **7.5 Pemeriksaan geometri akhir**

Pembahasan pemeriksaan geometri akhir dimulai dari prinsip bahwa satu angka komputasi hanya bermakna jika struktur, metode, unit, dan status validasinya diketahui. Geometri final perlu diuji konektivitas, koordinat, serta kesiapan untuk frequency dan descriptor. Elemen yang paling relevan pada tahap ini meliputi final molblock, 2D connectivity dan 3D view. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Wang, Song, & Martínez, 2016).

Implementasi perangkat lunak untuk bagian ini mengikuti pola parse/validate/execute/record. Input terlebih dahulu dibaca, kemudian diverifikasi, setelah itu engine dijalankan, dan akhirnya keluaran dipaketkan menjadi result record. Parameter yang terkait mencakup final molblock, 2D connectivity, 3D view. Keluaran yang diharapkan adalah final geometry dan preview. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### ***7.5.1 Prinsip, parameter, dan interpretasi***

Quality control harus membedakan kondisi tidak valid, kondisi meragukan, dan kondisi yang sekadar berbeda dari default. Aturan utama yang perlu dipertahankan adalah bahwa optimasi dapat mengubah jarak secara besar tanpa mengubah konektivitas formal; inspeksi tetap diperlukan. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Strategi operasional yang aman untuk pemeriksaan geometri akhir adalah memulai dari konfigurasi sederhana, memeriksa log, lalu menambah kompleksitas satu parameter pada satu waktu. Setelah tahap selesai, periksa apakah final geometry dan preview benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***7.5.2 Checklist implementasi dan quality control***

- Pastikan parameter final molblock telah ditentukan dan disimpan.
- Pastikan parameter 2D connectivity telah ditentukan dan disimpan.
- Pastikan parameter 3D view telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: optimasi dapat mengubah jarak secara besar tanpa mengubah konektivitas formal; inspeksi tetap diperlukan.
- Pastikan keluaran tahap tersedia sebagai final geometry dan preview dan status QC tercatat.

## **7.6 Rangkuman Bab**

Bab 7 menunjukkan bahwa optimasi geometri, geometric, dan kontrol konvergensi harus dipahami sebagai satu sistem yang menghubungkan keputusan ilmiah dan keputusan perangkat lunak. Keandalan QMScreen bergantung pada kemampuan setiap tahap untuk menerima input yang jelas, menjalankan metode dengan parameter eksplisit, mencatat status, mempertahankan hasil parsial secara

benar, dan menyediakan provenance. Dengan pola ini, batch yang besar tetap dapat diaudit dan dikembangkan tanpa menjadikan GUI sebagai satu-satunya sumber kebenaran. Pembaca sebaiknya menggunakan checklist pada setiap subbab sebagai kendali sebelum melanjutkan ke tahap berikutnya.

## BAB 8 FREQUENCY, THERMOCHEMISTRY, HESSIAN, DAN MINIMUM CHECK

Bab ini membahas frequency, thermochemistry, hessian, dan minimum check sebagai bagian dari workflow QMScreen V1.0.5. Pembahasan menempatkan konsep kimia komputasi, implementasi perangkat lunak, scientific guardrail, quality control, dan reproducibility dalam satu alur. Setiap subbab diakhiri dengan checklist praktis agar pembaca dapat menghubungkan teori dengan penggunaan aplikasi.

### Validasi Minimum Lokal



Gambar 8.1. Peta konseptual frequency, thermochemistry, hessian, dan minimum check.

Tabel 8.1. Peta materi Bab 8

| Subbab | Fokus                                | Keluaran utama                      |
|--------|--------------------------------------|-------------------------------------|
| 8.1    | Makna perhitungan frequency          | daftar frekuensi dan mode           |
| 8.2    | Minimum lokal dan frekuensi imajiner | minimum check status                |
| 8.3    | Termokimia                           | thermochemistry dictionary          |
| 8.4    | Failure dan hasil parsial            | partial result dengan QC            |
| 8.5    | Biaya dan strategi penggunaan        | daftar kandidat tervalidasi minimum |

### 8.1 Makna perhitungan frequency

Pembahasan makna perhitungan frequency dimulai dari prinsip bahwa satu angka komputasi hanya bermakna jika struktur, metode, unit, dan status validasinya diketahui. Frequency berasal dari analisis Hessian dan memberi informasi mode vibrasi serta karakter titik stasioner. Elemen yang paling relevan pada tahap ini meliputi Hessian, normal modes dan imaginary frequency. Masing-masing elemen

harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Cramer, 2004; Jensen, 2017).

Implementasi perangkat lunak untuk bagian ini mengikuti pola parse/validate/execute/record. Input terlebih dahulu dibaca, kemudian diverifikasi, setelah itu engine dijalankan, dan akhirnya keluaran dipaketkan menjadi result record. Parameter yang terkait mencakup `Hessian`, normal modes, imaginary frequency. Keluaran yang diharapkan adalah daftar frekuensi dan mode. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### ***8.1.1 Prinsip, parameter, dan interpretasi***

Quality control harus membedakan kondisi tidak valid, kondisi meragukan, dan kondisi yang sekadar berbeda dari default. Aturan utama yang perlu dipertahankan adalah bahwa frequency jauh lebih mahal daripada single-point dan harus diaktifkan sesuai tujuan. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Strategi operasional yang aman untuk makna perhitungan frequency adalah memulai dari konfigurasi sederhana, memeriksa log, lalu menambah kompleksitas satu parameter pada satu waktu. Setelah tahap selesai, periksa apakah daftar frekuensi dan mode benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***8.1.2 Checklist implementasi dan quality control***

- Pastikan parameter Hessian telah ditentukan dan disimpan.
- Pastikan parameter normal modes telah ditentukan dan disimpan.
- Pastikan parameter imaginary frequency telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: frequency jauh lebih mahal daripada single-point dan harus diaktifkan sesuai tujuan.
- Pastikan keluaran tahap tersedia sebagai daftar frekuensi dan mode dan status QC tercatat.

## **8.2 Minimum lokal dan frekuensi imajiner**

Pada QMScreen, minimum lokal dan frekuensi imajiner bukan fitur dekoratif antarmuka. Minimum lokal ideal tidak memiliki frekuensi imajiner dalam kerangka harmonik yang digunakan. Pendekatan ini membuat tahap tersebut dapat diuji sebagai komponen ilmiah yang mandiri. Elemen yang paling relevan pada tahap ini meliputi imaginary count, numerical noise dan low modes. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Cramer, 2004; Jensen, 2017).

Secara arsitektural, bagian minimum lokal dan frekuensi imajiner ditempatkan di luar logika tampilan sebanyak mungkin. Data dari widget hanya menjadi sumber konfigurasi, sedangkan fungsi inti bekerja pada struktur Python yang dapat diuji. Titik kendalinya meliputi imaginary count, numerical noise, low modes. Keluaran yang diharapkan adalah minimum check status. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### ***8.2.1 Prinsip, parameter, dan interpretasi***

Risiko terbesar dalam otomasi adalah membuat proses tampak berhasil ketika asumsi ilmiahnya tidak terpenuhi. Aturan utama yang perlu dipertahankan adalah bahwa satu nilai negatif kecil perlu interpretasi hati-hati; aturan threshold harus eksplisit jika diterapkan. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari

keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Untuk tujuan riset, hasil dari minimum lokal dan frekuensi imajiner sebaiknya ditinjau bersama data sebelum dan sesudah tahap tersebut agar perubahan dapat dilacak. Setelah tahap selesai, periksa apakah minimum check status benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

Minimum lokal: jumlah frekuensi imajiner = 0 (setelah mempertimbangkan toleransi numerik)

### 8.2.2 Checklist implementasi dan quality control

- Pastikan parameter imaginary count telah ditentukan dan disimpan.
- Pastikan parameter numerical noise telah ditentukan dan disimpan.
- Pastikan parameter low modes telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: satu nilai negatif kecil perlu interpretasi hati-hati; aturan threshold harus eksplisit jika diterapkan.
- Pastikan keluaran tahap tersedia sebagai minimum check status dan status QC tercatat.

## 8.3 Termokimia

High-throughput chemistry menuntut aturan yang konsisten ketika jumlah molekul bertambah. Dalam konteks termokimia, Frequency dapat digunakan untuk membentuk koreksi vibrasi dan besaran termokimia pada temperatur serta tekanan tertentu. Elemen yang paling relevan pada tahap ini meliputi 298.15 K, 101325 Pa default, ZPE dan thermal correction. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Cramer, 2004; Jensen, 2017).

Untuk batch besar, implementasi harus tahan terhadap kegagalan parsial. QMScreen karena itu menyimpan status per molekul dan per tahap, sambil mempertahankan parameter seperti 298.15 K, 101325 Pa default, `ZPE`, thermal correction pada manifest atau provenance. Keluaran yang diharapkan adalah thermochemistry dictionary. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### ***8.3.1 Prinsip, parameter, dan interpretasi***

Ketika workflow dipercepat, kontrol ilmiah tidak boleh ikut dipangkas. Aturan utama yang perlu dipertahankan adalah bahwa besaran termokimia bergantung pada model statistik dan kondisi input; unit harus jelas. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Pada komputer dengan resource terbatas, termokimia perlu diatur bersama jumlah worker, thread, memory, dan biaya tahap berikutnya. Setelah tahap selesai, periksa apakah thermochemistry dictionary benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***8.3.2 Checklist implementasi dan quality control***

- Pastikan parameter 298.15 K telah ditentukan dan disimpan.
- Pastikan parameter 101325 Pa default telah ditentukan dan disimpan.
- Pastikan parameter ZPE telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: besaran termokimia bergantung pada model statistik dan kondisi input; unit harus jelas.

- Pastikan keluaran tahap tersedia sebagai thermochemistry dictionary dan status QC tercatat.

## 8.4 Failure dan hasil parsial

QMS-E007 memisahkan kegagalan Hessian dari energi yang mungkin sudah berhasil dihitung. Dalam workflow komputasi skala besar, konsep ini perlu didefinisikan sebelum job pertama dijalankan karena keputusan awal akan memengaruhi biaya, keterbandingan, dan kemampuan reproduksi hasil. Elemen yang paling relevan pada tahap ini meliputi preserve energy, failed frequency dan retry. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Cramer, 2004; Jensen, 2017).

Di tingkat implementasi, QMScreen mengubah pilihan pengguna menjadi record konfigurasi dan menjalankan fungsi scientific logic yang sesuai. Komponen seperti preserve energy, failed frequency, 'retry' tidak diperlakukan sebagai nilai sementara, tetapi sebagai bagian dari state proyek yang dapat ditulis ke disk. Keluaran yang diharapkan adalah partial result dengan QC. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### 8.4.1 Prinsip, parameter, dan interpretasi

Scientific guardrail pada bagian ini sengaja dibuat konservatif. Aturan utama yang perlu dipertahankan adalah bahwa hasil energi tidak boleh dibuang hanya karena frequency gagal, tetapi status minimum harus tetap belum tervalidasi. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Dalam penggunaan sehari-hari, pengguna sebaiknya menguji failure dan hasil parsial pada satu atau dua molekul kecil sebelum memperbesar batch. Setelah tahap selesai, periksa apakah partial result dengan QC benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

#### ***8.4.2 Checklist implementasi dan quality control***

- Pastikan parameter preserve energy telah ditentukan dan disimpan.
- Pastikan parameter failed frequency telah ditentukan dan disimpan.
- Pastikan parameter retry telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: hasil energi tidak boleh dibuang hanya karena frequency gagal, tetapi status minimum harus tetap belum tervalidasi.
- Pastikan keluaran tahap tersedia sebagai partial result dengan QC dan status QC tercatat.

### **8.5 Biaya dan strategi penggunaan**

Pembahasan biaya dan strategi penggunaan dimulai dari prinsip bahwa satu angka komputasi hanya bermakna jika struktur, metode, unit, dan status validasinya diketahui. Untuk screening besar, frequency dapat dibatasi pada kandidat final agar biaya tetap terkendali. Elemen yang paling relevan pada tahap ini meliputi stage gating, top candidates dan batch cost. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Cramer, 2004; Jensen, 2017).

Implementasi perangkat lunak untuk bagian ini mengikuti pola parse/validate/execute/record. Input terlebih dahulu dibaca, kemudian diverifikasi, setelah itu engine dijalankan, dan akhirnya keluaran dipaketkan menjadi result record. Parameter yang terkait mencakup stage gating, top candidates, batch cost. Keluaran yang diharapkan adalah daftar kandidat tervalidasi minimum. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### ***8.5.1 Prinsip, parameter, dan interpretasi***

Quality control harus membedakan kondisi tidak valid, kondisi meragukan, dan kondisi yang sekadar berbeda dari default. Aturan utama yang perlu dipertahankan adalah bahwa pemilihan kandidat untuk frequency harus ditentukan sebelum analisis akhir bila digunakan untuk inferensi statistik. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Strategi operasional yang aman untuk biaya dan strategi penggunaan adalah memulai dari konfigurasi sederhana, memeriksa log, lalu menambah kompleksitas satu parameter pada satu waktu. Setelah tahap selesai, periksa apakah daftar kandidat tervalidasi minimum benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***8.5.2 Checklist implementasi dan quality control***

- Pastikan parameter stage gating telah ditentukan dan disimpan.
- Pastikan parameter top candidates telah ditentukan dan disimpan.
- Pastikan parameter batch cost telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: pemilihan kandidat untuk frequency harus ditentukan sebelum analisis akhir bila digunakan untuk inferensi statistik.
- Pastikan keluaran tahap tersedia sebagai daftar kandidat tervalidasi minimum dan status QC tercatat.

## **8.6 Rangkuman Bab**

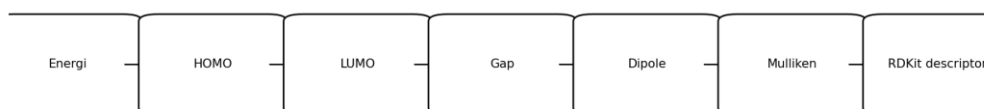
Bab 8 menunjukkan bahwa frequency, thermochemistry, hessian, dan minimum check harus dipahami sebagai satu sistem yang menghubungkan keputusan ilmiah dan keputusan perangkat lunak. Keandalan QMScreen bergantung pada

kemampuan setiap tahap untuk menerima input yang jelas, menjalankan metode dengan parameter eksplisit, mencatat status, mempertahankan hasil parsial secara benar, dan menyediakan provenance. Dengan pola ini, batch yang besar tetap dapat diaudit dan dikembangkan tanpa menjadikan GUI sebagai satu-satunya sumber kebenaran. Pembaca sebaiknya menggunakan checklist pada setiap subbab sebagai kendali sebelum melanjutkan ke tahap berikutnya.

## BAB 9 HOMO-LUMO, DIPOLE, MULLIKEN CHARGE, DAN DESCRIPTOR KUANTUM

Bab ini membahas homo-lumo, dipole, mulliken charge, dan descriptor kuantum sebagai bagian dari workflow QMScreen V1.0.5. Pembahasan menempatkan konsep kimia komputasi, implementasi perangkat lunak, scientific guardrail, quality control, dan reproducibility dalam satu alur. Setiap subbab diakhiri dengan checklist praktis agar pembaca dapat menghubungkan teori dengan penggunaan aplikasi.

**Descriptor Kuantum**



*Gambar 9.1. Peta konseptual homo-lumo, dipole, mulliken charge, dan descriptor kuantum.*

**Tabel 9.1. Peta materi Bab 9**

| Subbab | Fokus                        | Keluaran utama              |
|--------|------------------------------|-----------------------------|
| 9.1    | Orbital frontier             | HOMO, LUMO, gap eV          |
| 9.2    | Momen dipol                  | dipole total dan komponen   |
| 9.3    | Mulliken population analysis | tabel muatan per atom       |
| 9.4    | Descriptor molekul RDKit     | tabel descriptor gabungan   |
| 9.5    | Interpretasi lintas kandidat | dataframe hasil tervalidasi |

### 9.1 Orbital frontier

Pembahasan orbital frontier dimulai dari prinsip bahwa satu angka komputasi hanya bermakna jika struktur, metode, unit, dan status validasinya diketahui. HOMO dan LUMO merupakan orbital frontier yang sering dipakai sebagai descriptor elektronik, sedangkan gap dihitung dari selisih energinya. Elemen yang paling relevan pada tahap ini meliputi orbital energies, occupation dan spin cases. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan,

pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Becke, 1993; Lee, Yang, & Parr, 1988; Parr & Yang, 1989).

Implementasi perangkat lunak untuk bagian ini mengikuti pola parse/validate/execute/record. Input terlebih dahulu dibaca, kemudian diverifikasi, setelah itu engine dijalankan, dan akhirnya keluaran dipaketkan menjadi result record. Parameter yang terkait mencakup orbital energies, `occupation`, spin cases. Keluaran yang diharapkan adalah HOMO, LUMO, gap eV. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### ***9.1.1 Prinsip, parameter, dan interpretasi***

Quality control harus membedakan kondisi tidak valid, kondisi meragukan, dan kondisi yang sekadar berbeda dari default. Aturan utama yang perlu dipertahankan adalah bahwa gap orbital bukan otomatis band gap eksperimen atau energi eksitasi. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Strategi operasional yang aman untuk orbital frontier adalah memulai dari konfigurasi sederhana, memeriksa log, lalu menambah kompleksitas satu parameter pada satu waktu. Setelah tahap selesai, periksa apakah HOMO, LUMO, gap eV benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

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

### ***9.1.2 Checklist implementasi dan quality control***

- Pastikan parameter orbital energies telah ditentukan dan disimpan.
- Pastikan parameter occupation telah ditentukan dan disimpan.
- Pastikan parameter spin cases telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: gap orbital bukan otomatis band gap eksperimen atau energi eksitasi.
- Pastikan keluaran tahap tersedia sebagai HOMO, LUMO, gap eV dan status QC tercatat.

## **9.2 Momen dipol**

Pada QMScreen, momen dipol bukan fitur dekoratif antarmuka. Dipole menggambarkan distribusi muatan total dalam representasi yang dipakai perhitungan. Pendekatan ini membuat tahap tersebut dapat diuji sebagai komponen ilmiah yang mandiri. Elemen yang paling relevan pada tahap ini meliputi vector components, magnitude dan Debye. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Becke, 1993; Lee, Yang, & Parr, 1988; Parr & Yang, 1989).

Secara arsitektural, bagian momen dipol ditempatkan di luar logika tampilan sebanyak mungkin. Data dari widget hanya menjadi sumber konfigurasi, sedangkan fungsi inti bekerja pada struktur Python yang dapat diuji. Titik kendalinya meliputi vector components, `magnitude`, `Debye`. Keluaran yang diharapkan adalah dipole total dan komponen. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### ***9.2.1 Prinsip, parameter, dan interpretasi***

Risiko terbesar dalam otomasi adalah membuat proses tampak berhasil ketika asumsi ilmiahnya tidak terpenuhi. Aturan utama yang perlu dipertahankan adalah bahwa nilai sensitif terhadap geometri, metode, basis, dan keadaan elektronik. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini

juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Untuk tujuan riset, hasil dari momen dipol sebaiknya ditinjau bersama data sebelum dan sesudah tahap tersebut agar perubahan dapat dilacak. Setelah tahap selesai, periksa apakah dipole total dan komponen benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***9.2.2 Checklist implementasi dan quality control***

- Pastikan parameter vector components telah ditentukan dan disimpan.
- Pastikan parameter magnitude telah ditentukan dan disimpan.
- Pastikan parameter Debye telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: nilai sensitif terhadap geometri, metode, basis, dan keadaan elektronik.
- Pastikan keluaran tahap tersedia sebagai dipole total dan komponen dan status QC tercatat.

## **9.3 Mulliken population analysis**

High-throughput chemistry menuntut aturan yang konsisten ketika jumlah molekul bertambah. Dalam konteks mulliken population analysis, Mulliken charge membagi populasi elektron berdasarkan basis orbital dan mudah dihitung tetapi bersifat basis-dependent. Elemen yang paling relevan pada tahap ini meliputi atomic charges, population dan sum charge. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Becke, 1993; Lee, Yang, & Parr, 1988; Parr & Yang, 1989).

Untuk batch besar, implementasi harus tahan terhadap kegagalan parsial. QMScreen karena itu menyimpan status per molekul dan per tahap, sambil

mempertahankan parameter seperti atomic charges, `population`, sum charge pada manifest atau provenance. Keluaran yang diharapkan adalah tabel muatan per atom. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### ***9.3.1 Prinsip, parameter, dan interpretasi***

Ketika workflow dipercepat, kontrol ilmiah tidak boleh ikut dipangkas. Aturan utama yang perlu dipertahankan adalah bahwa muatan parsial bukan observabel tunggal yang unik; interpretasi harus komparatif dan hati-hati. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Pada komputer dengan resource terbatas, muliken population analysis perlu diatur bersama jumlah worker, thread, memory, dan biaya tahap berikutnya. Setelah tahap selesai, periksa apakah tabel muatan per atom benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***9.3.2 Checklist implementasi dan quality control***

- Pastikan parameter atomic charges telah ditentukan dan disimpan.
- Pastikan parameter population telah ditentukan dan disimpan.
- Pastikan parameter sum charge telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: muatan parsial bukan observabel tunggal yang unik; interpretasi harus komparatif dan hati-hati.
- Pastikan keluaran tahap tersedia sebagai tabel muatan per atom dan status QC tercatat.

## 9.4 Descriptor molekul RDKit

QMScreen juga menyimpan MW, TPSA, LogP, HBD, HBA, dan rotatable bonds untuk konteks tambahan. Dalam workflow komputasi skala besar, konsep ini perlu didefinisikan sebelum job pertama dijalankan karena keputusan awal akan memengaruhi biaya, keterbandingan, dan kemampuan reproduksi hasil. Elemen yang paling relevan pada tahap ini meliputi molecular descriptors dan cheminformatics. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Becke, 1993; Lee, Yang, & Parr, 1988; Parr & Yang, 1989).

Di tingkat implementasi, QMScreen mengubah pilihan pengguna menjadi record konfigurasi dan menjalankan fungsi scientific logic yang sesuai. Komponen seperti molecular descriptors, 'cheminformatics' tidak diperlakukan sebagai nilai sementara, tetapi sebagai bagian dari state proyek yang dapat ditulis ke disk. Keluaran yang diharapkan adalah tabel descriptor gabungan. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### 9.4.1 Prinsip, parameter, dan interpretasi

Scientific guardrail pada bagian ini sengaja dibuat konservatif. Aturan utama yang perlu dipertahankan adalah bahwa descriptor 2D tidak menggantikan properti kuantum dan sebaliknya; gabungan harus sesuai tujuan. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Dalam penggunaan sehari-hari, pengguna sebaiknya menguji descriptor molekul rdkit pada satu atau dua molekul kecil sebelum memperbesar batch. Setelah tahap selesai, periksa apakah tabel descriptor gabungan benar-benar tersedia, apakah

unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

#### ***9.4.2 Checklist implementasi dan quality control***

- Pastikan parameter molecular descriptors telah ditentukan dan disimpan.
- Pastikan parameter cheminformatics telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: descriptor 2D tidak menggantikan properti kuantum dan sebaliknya; gabungan harus sesuai tujuan.
- Pastikan keluaran tahap tersedia sebagai tabel descriptor gabungan dan status QC tercatat.

### **9.5 Interpretasi lintas kandidat**

Pembahasan interpretasi lintas kandidat dimulai dari prinsip bahwa satu angka komputasi hanya bermakna jika struktur, metode, unit, dan status validasinya diketahui. Perbandingan descriptor hanya bermakna ketika status QC dan metode setara. Elemen yang paling relevan pada tahap ini meliputi same workflow, units, formula groups dan missing values. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Becke, 1993; Lee, Yang, & Parr, 1988; Parr & Yang, 1989).

Implementasi perangkat lunak untuk bagian ini mengikuti pola parse/validate/execute/record. Input terlebih dahulu dibaca, kemudian diverifikasi, setelah itu engine dijalankan, dan akhirnya keluaran dipaketkan menjadi result record. Parameter yang terkait mencakup same workflow, 'units', formula groups, missing values. Keluaran yang diharapkan adalah dataframe hasil tervalidasi. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

#### ***9.5.1 Prinsip, parameter, dan interpretasi***

Quality control harus membedakan kondisi tidak valid, kondisi meragukan, dan kondisi yang sekadar berbeda dari default. Aturan utama yang perlu dipertahankan

adalah bahwa nilai dari job gagal atau metode berbeda tidak boleh dicampur tanpa penandaan. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Strategi operasional yang aman untuk interpretasi lintas kandidat adalah memulai dari konfigurasi sederhana, memeriksa log, lalu menambah kompleksitas satu parameter pada satu waktu. Setelah tahap selesai, periksa apakah dataframe hasil tervalidasi benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***9.5.2 Checklist implementasi dan quality control***

- Pastikan parameter same workflow telah ditentukan dan disimpan.
- Pastikan parameter units telah ditentukan dan disimpan.
- Pastikan parameter formula groups telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: nilai dari job gagal atau metode berbeda tidak boleh dicampur tanpa penandaan.
- Pastikan keluaran tahap tersedia sebagai dataframe hasil tervalidasi dan status QC tercatat.

## **9.6 Rangkuman Bab**

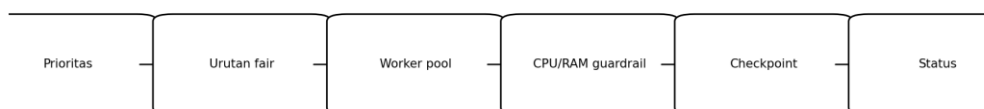
Bab 9 menunjukkan bahwa homo-lumo, dipole, mulliken charge, dan descriptor kuantum harus dipahami sebagai satu sistem yang menghubungkan keputusan ilmiah dan keputusan perangkat lunak. Keandalan QMScreen bergantung pada kemampuan setiap tahap untuk menerima input yang jelas, menjalankan metode dengan parameter eksplisit, mencatat status, mempertahankan hasil parsial secara benar, dan menyediakan provenance. Dengan pola ini, batch yang besar tetap dapat diaudit dan dikembangkan tanpa menjadikan GUI sebagai satu-satunya

sumber kebenaran. Pembaca sebaiknya menggunakan checklist pada setiap subbab sebagai kendali sebelum melanjutkan ke tahap berikutnya.

## BAB 10 SCHEDULING PARALEL, RESOURCE LIMIT, FAIR QUEUE, DAN STOP AMAN

Bab ini membahas scheduling paralel, resource limit, fair queue, dan stop aman sebagai bagian dari workflow QMScreen V1.0.5. Pembahasan menempatkan konsep kimia komputasi, implementasi perangkat lunak, scientific guardrail, quality control, dan reproducibility dalam satu alur. Setiap subbab diakhiri dengan checklist praktis agar pembaca dapat menghubungkan teori dengan penggunaan aplikasi.

**Fair Queue dan Resource Guardrail**



*Gambar 10.1. Peta konseptual scheduling paralel, resource limit, fair queue, dan stop aman.*

**Tabel 10.1. Peta materi Bab 10**

| Subbab | Fokus              | Keluaran utama                    |
|--------|--------------------|-----------------------------------|
| 10.1   | Paralelisasi batch | worker status dan runtime         |
| 10.2   | Resource guardrail | QMS-E008 atau izin eksekusi       |
| 10.3   | Fair queue         | queue order dan priority metadata |
| 10.4   | Stop aman          | stopped status dan checkpoint     |
| 10.5   | Monitoring dan log | status ringkas dan detail job     |

### 10.1 Paralelisasi batch

Pembahasan paralelisasi batch dimulai dari prinsip bahwa satu angka komputasi hanya bermakna jika struktur, metode, unit, dan status validasinya diketahui. QMScreen menjalankan beberapa job melalui worker pool sehingga kandidat dapat diproses serentak sesuai resource. Elemen yang paling relevan pada tahap ini meliputi ThreadPoolExecutor, parallel workers dan threads per job. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci

hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Sandve et al., 2013; Wilkinson et al., 2016; Wilson et al., 2014).

Implementasi perangkat lunak untuk bagian ini mengikuti pola parse/validate/execute/record. Input terlebih dahulu dibaca, kemudian diverifikasi, setelah itu engine dijalankan, dan akhirnya keluaran dipaketkan menjadi result record. Parameter yang terkait mencakup `ThreadPoolExecutor`, parallel workers, threads per job. Keluaran yang diharapkan adalah worker status dan runtime. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### ***10.1.1 Prinsip, parameter, dan interpretasi***

Quality control harus membedakan kondisi tidak valid, kondisi meragukan, dan kondisi yang sekadar berbeda dari default. Aturan utama yang perlu dipertahankan adalah bahwa oversubscription CPU dan RAM dapat memperlambat seluruh batch atau memicu kegagalan. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Strategi operasional yang aman untuk paralelisasi batch adalah memulai dari konfigurasi sederhana, memeriksa log, lalu menambah kompleksitas satu parameter pada satu waktu. Setelah tahap selesai, periksa apakah worker status dan runtime benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***10.1.2 Checklist implementasi dan quality control***

- Pastikan parameter ThreadPoolExecutor telah ditentukan dan disimpan.

- Pastikan parameter parallel workers telah ditentukan dan disimpan.
- Pastikan parameter threads per job telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: oversubscription CPU dan RAM dapat memperlambat seluruh batch atau memicu kegagalan.
- Pastikan keluaran tahap tersedia sebagai worker status dan runtime dan status QC tercatat.

**Tabel 10.2. Parameter resource dan fungsi operasional**

| Parameter           | Makna                  | Risiko bila terlalu besar              |
|---------------------|------------------------|----------------------------------------|
| Parallel workers    | Jumlah job bersamaan   | Oversubscription proses                |
| CPU threads per job | Thread per kalkulasi   | CPU contention                         |
| Memory per job      | Batas memori backend   | Swap/OOM jika total melampaui RAM      |
| Max opt steps       | Batas iterasi optimasi | Runtime membengkak pada struktur sulit |

## 10.2 Resource guardrail

Pada QMScreen, resource guardrail bukan fitur dekoratif antarmuka. Estimasi memory dan batas resource digunakan sebelum job berat dijalankan. Pendekatan ini membuat tahap tersebut dapat diuji sebagai komponen ilmiah yang mandiri. Elemen yang paling relevan pada tahap ini meliputi memory per job, logical CPU dan available RAM. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Sandve et al., 2013; Wilkinson et al., 2016; Wilson et al., 2014).

Secara arsitektural, bagian resource guardrail ditempatkan di luar logika tampilan sebanyak mungkin. Data dari widget hanya menjadi sumber konfigurasi, sedangkan fungsi inti bekerja pada struktur Python yang dapat diuji. Titik kendalinya meliputi memory per job, logical CPU, available RAM. Keluaran yang diharapkan adalah QMS-E008 atau izin eksekusi. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### ***10.2.1 Prinsip, parameter, dan interpretasi***

Risiko terbesar dalam otomasi adalah membuat proses tampak berhasil ketika asumsi ilmiahnya tidak terpenuhi. Aturan utama yang perlu dipertahankan adalah bahwa resource guardrail harus konservatif dan tidak menggantikan monitoring sistem nyata. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Untuk tujuan riset, hasil dari resource guardrail sebaiknya ditinjau bersama data sebelum dan sesudah tahap tersebut agar perubahan dapat dilacak. Setelah tahap selesai, periksa apakah QMS-E008 atau izin eksekusi benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***10.2.2 Checklist implementasi dan quality control***

- Pastikan parameter memory per job telah ditentukan dan disimpan.
- Pastikan parameter logical CPU telah ditentukan dan disimpan.
- Pastikan parameter available RAM telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: resource guardrail harus konservatif dan tidak menggantikan monitoring sistem nyata.
- Pastikan keluaran tahap tersedia sebagai QMS-E008 atau izin eksekusi dan status QC tercatat.

## **10.3 Fair queue**

High-throughput chemistry menuntut aturan yang konsisten ketika jumlah molekul bertambah. Dalam konteks fair queue, Urutan fair mencegah satu kelompok prioritas atau satu pola job memonopoli worker. Elemen yang paling relevan pada tahap ini meliputi High/Normal/Low priority, ordering dan starvation avoidance. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI.

Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Sandve et al., 2013; Wilkinson et al., 2016; Wilson et al., 2014).

Untuk batch besar, implementasi harus tahan terhadap kegagalan parsial. QMScreen karena itu menyimpan status per molekul dan per tahap, sambil mempertahankan parameter seperti High/Normal/Low priority, `ordering`, starvation avoidance pada manifest atau provenance. Keluaran yang diharapkan adalah queue order dan priority metadata. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### ***10.3.1 Prinsip, parameter, dan interpretasi***

Ketika workflow dipercepat, kontrol ilmiah tidak boleh ikut dipangkas. Aturan utama yang perlu dipertahankan adalah bahwa prioritas adalah kebijakan operasional, bukan kualitas ilmiah kandidat. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Pada komputer dengan resource terbatas, fair queue perlu diatur bersama jumlah worker, thread, memory, dan biaya tahap berikutnya. Setelah tahap selesai, periksa apakah queue order dan priority metadata benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***10.3.2 Checklist implementasi dan quality control***

- Pastikan parameter High/Normal/Low priority telah ditentukan dan disimpan.
- Pastikan parameter ordering telah ditentukan dan disimpan.

- Pastikan parameter starvation avoidance telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: prioritas adalah kebijakan operasional, bukan kualitas ilmiah kandidat.
- Pastikan keluaran tahap tersedia sebagai queue order dan priority metadata dan status QC tercatat.

## 10.4 Stop aman

Tombol stop mengirim flag agar proses berhenti pada titik/checkpoint aman ketimbang mematikan proses secara kasar. Dalam workflow komputasi skala besar, konsep ini perlu didefinisikan sebelum job pertama dijalankan karena keputusan awal akan memengaruhi biaya, keterbandingan, dan kemampuan reproduksi hasil. Elemen yang paling relevan pada tahap ini meliputi `threading.Event`, checkpoint boundaries dan state. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Sandve et al., 2013; Wilkinson et al., 2016; Wilson et al., 2014).

Di tingkat implementasi, QMScreen mengubah pilihan pengguna menjadi record konfigurasi dan menjalankan fungsi scientific logic yang sesuai. Komponen seperti ``threading.Event``, checkpoint boundaries, ``state`` tidak diperlakukan sebagai nilai sementara, tetapi sebagai bagian dari state proyek yang dapat ditulis ke disk. Keluaran yang diharapkan adalah stopped status dan checkpoint. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### 10.4.1 Prinsip, parameter, dan interpretasi

Scientific guardrail pada bagian ini sengaja dibuat konservatif. Aturan utama yang perlu dipertahankan adalah bahwa stop tidak menjamin semua backend eksternal berhenti seketika; file harus tetap konsisten. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan

kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Dalam penggunaan sehari-hari, pengguna sebaiknya menguji stop aman pada satu atau dua molekul kecil sebelum memperbesar batch. Setelah tahap selesai, periksa apakah stopped status dan checkpoint benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

#### ***10.4.2 Checklist implementasi dan quality control***

- Pastikan parameter threading.Event telah ditentukan dan disimpan.
- Pastikan parameter checkpoint boundaries telah ditentukan dan disimpan.
- Pastikan parameter state telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: stop tidak menjamin semua backend eksternal berhenti seketika; file harus tetap konsisten.
- Pastikan keluaran tahap tersedia sebagai stopped status dan checkpoint dan status QC tercatat.

### **10.5 Monitoring dan log**

Pembahasan monitoring dan log dimulai dari prinsip bahwa satu angka komputasi hanya bermakna jika struktur, metode, unit, dan status validasinya diketahui. Dashboard queue harus memisahkan running, done, failed/QC, dan stopped. Elemen yang paling relevan pada tahap ini meliputi batch log, progress dan state file. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Sandve et al., 2013; Wilkinson et al., 2016; Wilson et al., 2014).

Implementasi perangkat lunak untuk bagian ini mengikuti pola parse/validate/execute/record. Input terlebih dahulu dibaca, kemudian diverifikasi, setelah itu engine dijalankan, dan akhirnya keluaran dipaketkan menjadi result record. Parameter yang terkait mencakup batch log, `progress`, state file. Keluaran yang diharapkan adalah status ringkas dan detail job. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada

sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### ***10.5.1 Prinsip, parameter, dan interpretasi***

Quality control harus membedakan kondisi tidak valid, kondisi meragukan, dan kondisi yang sekadar berbeda dari default. Aturan utama yang perlu dipertahankan adalah bahwa jumlah done tanpa konteks QC dapat menyesatkan; pengguna harus membuka log dan result record. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Strategi operasional yang aman untuk monitoring dan log adalah memulai dari konfigurasi sederhana, memeriksa log, lalu menambah kompleksitas satu parameter pada satu waktu. Setelah tahap selesai, periksa apakah status ringkas dan detail job benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***10.5.2 Checklist implementasi dan quality control***

- Pastikan parameter batch log telah ditentukan dan disimpan.
- Pastikan parameter progress telah ditentukan dan disimpan.
- Pastikan parameter state file telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: jumlah done tanpa konteks QC dapat menyesatkan; pengguna harus membuka log dan result record.
- Pastikan keluaran tahap tersedia sebagai status ringkas dan detail job dan status QC tercatat.

**Sketsa implementasi Python - Bab 10 (pseudocode original)**

```
queue = fair_order(jobs)
with WorkerPool(max_workers=n_workers) as pool:
    pool.submit(run_job, job, resource_limits)
```

## 10.6 Rangkuman Bab

Bab 10 menunjukkan bahwa scheduling paralel, resource limit, fair queue, dan stop aman harus dipahami sebagai satu sistem yang menghubungkan keputusan ilmiah dan keputusan perangkat lunak. Keandalan QMScreen bergantung pada kemampuan setiap tahap untuk menerima input yang jelas, menjalankan metode dengan parameter eksplisit, mencatat status, mempertahankan hasil parsial secara benar, dan menyediakan provenance. Dengan pola ini, batch yang besar tetap dapat diaudit dan dikembangkan tanpa menjadikan GUI sebagai satu-satunya sumber kebenaran. Pembaca sebaiknya menggunakan checklist pada setiap subbab sebagai kendali sebelum melanjutkan ke tahap berikutnya.

## BAB 11 RESUME, CHECKPOINT, CACHE, HASH, SQLITE, DAN PROVENANCE

Bab ini membahas resume, checkpoint, cache, hash, sqlite, dan provenance sebagai bagian dari workflow QMScreen V1.0.5. Pembahasan menempatkan konsep kimia komputasi, implementasi perangkat lunak, scientific guardrail, quality control, dan reproducibility dalam satu alur. Setiap subbab diakhiri dengan checklist praktis agar pembaca dapat menghubungkan teori dengan penggunaan aplikasi.

### Resume dan Cache



Gambar 11.1. Peta konseptual resume, checkpoint, cache, hash, sqlite, dan provenance.

Tabel 11.1. Peta materi Bab 11

| Subbab | Fokus                        | Keluaran utama             |
|--------|------------------------------|----------------------------|
| 11.1   | Checkpoint per tahap         | checkpoint JSON            |
| 11.2   | Cache berbasis spec hash     | cache result terverifikasi |
| 11.3   | SQLite sebagai indeks proyek | qmscreen.sqlite3           |
| 11.4   | Environment snapshot         | metadata lingkungan        |
| 11.5   | Audit trail dan provenance   | jejak audit end-to-end     |

### 11.1 Checkpoint per tahap

Pembahasan checkpoint per tahap dimulai dari prinsip bahwa satu angka komputasi hanya bermakna jika struktur, metode, unit, dan status validasinya diketahui. Checkpoint menyimpan progres agar kalkulasi dapat dilanjutkan setelah interupsi. Elemen yang paling relevan pada tahap ini meliputi stage marker, partial result dan atomic write. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada

spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Sandve et al., 2013; Wilkinson et al., 2016; Wilson et al., 2014).

Implementasi perangkat lunak untuk bagian ini mengikuti pola parse/validate/execute/record. Input terlebih dahulu dibaca, kemudian diverifikasi, setelah itu engine dijalankan, dan akhirnya keluaran dipaketkan menjadi result record. Parameter yang terkait mencakup stage marker, partial result, atomic write. Keluaran yang diharapkan adalah checkpoint JSON. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### ***11.1.1 Prinsip, parameter, dan interpretasi***

Quality control harus membedakan kondisi tidak valid, kondisi meragukan, dan kondisi yang sekadar berbeda dari default. Aturan utama yang perlu dipertahankan adalah bahwa checkpoint harus ditulis atomik agar file setengah tulis tidak dianggap valid. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Strategi operasional yang aman untuk checkpoint per tahap adalah memulai dari konfigurasi sederhana, memeriksa log, lalu menambah kompleksitas satu parameter pada satu waktu. Setelah tahap selesai, periksa apakah checkpoint JSON benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***11.1.2 Checklist implementasi dan quality control***

- Pastikan parameter stage marker telah ditentukan dan disimpan.
- Pastikan parameter partial result telah ditentukan dan disimpan.

- Pastikan parameter atomic write telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: checkpoint harus ditulis atomik agar file setengah tulis tidak dianggap valid.
- Pastikan keluaran tahap tersedia sebagai checkpoint JSON dan status QC tercatat.

## 11.2 Cache berbasis spec hash

Pada QMScreen, cache berbasis spec hash bukan fitur dekoratif antarmuka. Cache hanya aman jika kunci memasukkan struktur dan parameter ilmiah yang relevan. Pendekatan ini membuat tahap tersebut dapat diuji sebagai komponen ilmiah yang mandiri. Elemen yang paling relevan pada tahap ini meliputi canonical JSON, SHA-256 dan validated flag. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Sandve et al., 2013; Wilkinson et al., 2016; Wilson et al., 2014).

Secara arsitektural, bagian cache berbasis spec hash ditempatkan di luar logika tampilan sebanyak mungkin. Data dari widget hanya menjadi sumber konfigurasi, sedangkan fungsi inti bekerja pada struktur Python yang dapat diuji. Titik kendalinya meliputi canonical JSON, 'SHA-256', validated flag. Keluaran yang diharapkan adalah cache result terverifikasi. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### 11.2.1 Prinsip, parameter, dan interpretasi

Risiko terbesar dalam otomasi adalah membuat proses tampak berhasil ketika asumsi ilmiahnya tidak terpenuhi. Aturan utama yang perlu dipertahankan adalah bahwa hasil gagal tidak boleh masuk cache valid; perubahan basis atau functional harus menghasilkan hash berbeda. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang

terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Untuk tujuan riset, hasil dari cache berbasis spec hash sebaiknya ditinjau bersama data sebelum dan sesudah tahap tersebut agar perubahan dapat dilacak. Setelah tahap selesai, periksa apakah cache result terverifikasi benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***11.2.2 Checklist implementasi dan quality control***

- Pastikan parameter canonical JSON telah ditentukan dan disimpan.
- Pastikan parameter SHA-256 telah ditentukan dan disimpan.
- Pastikan parameter validated flag telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: hasil gagal tidak boleh masuk cache valid; perubahan basis atau functional harus menghasilkan hash berbeda.
- Pastikan keluaran tahap tersedia sebagai cache result terverifikasi dan status QC tercatat.

## **11.3 SQLite sebagai indeks proyek**

High-throughput chemistry menuntut aturan yang konsisten ketika jumlah molekul bertambah. Dalam konteks sqlite sebagai indeks proyek, Database menyimpan project, molecule, job, result, cache, dan audit trail secara terstruktur. Elemen yang paling relevan pada tahap ini meliputi transactions, IDs, timestamps dan JSON payload. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Sandve et al., 2013; Wilkinson et al., 2016; Wilson et al., 2014).

Untuk batch besar, implementasi harus tahan terhadap kegagalan parsial. QMScreen karena itu menyimpan status per molekul dan per tahap, sambil mempertahankan parameter seperti 'transactions', 'IDs', 'timestamps', JSON payload pada manifest atau provenance. Keluaran yang diharapkan adalah qmscreen.sqlite3. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test

memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### ***11.3.1 Prinsip, parameter, dan interpretasi***

Ketika workflow dipercepat, kontrol ilmiah tidak boleh ikut dipangkas. Aturan utama yang perlu dipertahankan adalah bahwa database adalah indeks dan provenance; raw output tetap perlu dipertahankan terpisah. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Pada komputer dengan resource terbatas, sqlite sebagai indeks proyek perlu diatur bersama jumlah worker, thread, memory, dan biaya tahap berikutnya. Setelah tahap selesai, periksa apakah qmscreen.sqlite3 benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***11.3.2 Checklist implementasi dan quality control***

- Pastikan parameter transactions telah ditentukan dan disimpan.
- Pastikan parameter IDs telah ditentukan dan disimpan.
- Pastikan parameter timestamps telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: database adalah indeks dan provenance; raw output tetap perlu dipertahankan terpisah.
- Pastikan keluaran tahap tersedia sebagai qmscreen.sqlite3 dan status QC tercatat.

## **11.4 Environment snapshot**

Versi Python, platform, package, dan backend membantu reproduksi dan diagnosis perbedaan hasil. Dalam workflow komputasi skala besar, konsep ini perlu didefinisikan sebelum job pertama dijalankan karena keputusan awal akan

memengaruhi biaya, keterbandingan, dan kemampuan reproduksi hasil. Elemen yang paling relevan pada tahap ini meliputi `environment.json`, platform dan dependency status. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Sandve et al., 2013; Wilkinson et al., 2016; Wilson et al., 2014).

Di tingkat implementasi, QMScreen mengubah pilihan pengguna menjadi record konfigurasi dan menjalankan fungsi scientific logic yang sesuai. Komponen seperti `'environment.json'`, `'platform'`, dependency status tidak diperlakukan sebagai nilai sementara, tetapi sebagai bagian dari state proyek yang dapat ditulis ke disk. Keluaran yang diharapkan adalah metadata lingkungan. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

#### ***11.4.1 Prinsip, parameter, dan interpretasi***

Scientific guardrail pada bagian ini sengaja dibuat konservatif. Aturan utama yang perlu dipertahankan adalah bahwa snapshot tidak menjamin determinisme absolut tetapi mempersempit sumber variasi. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Dalam penggunaan sehari-hari, pengguna sebaiknya menguji environment snapshot pada satu atau dua molekul kecil sebelum memperbesar batch. Setelah tahap selesai, periksa apakah metadata lingkungan benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***11.4.2 Checklist implementasi dan quality control***

- Pastikan parameter `environment.json` telah ditentukan dan disimpan.
- Pastikan parameter platform telah ditentukan dan disimpan.
- Pastikan parameter dependency status telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: snapshot tidak menjamin determinisme absolut tetapi mempersempit sumber variasi.
- Pastikan keluaran tahap tersedia sebagai metadata lingkungan dan status QC tercatat.

## **11.5 Audit trail dan provenance**

Pembahasan audit trail dan provenance dimulai dari prinsip bahwa satu angka komputasi hanya bermakna jika struktur, metode, unit, dan status validasinya diketahui. Setiap keputusan otomatis dan perubahan state perlu dapat ditelusuri dari input sampai ekspor. Elemen yang paling relevan pada tahap ini meliputi `project_id`, `job_id`, `spec_hash`, `source` dan `timestamps`. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Sandve et al., 2013; Wilkinson et al., 2016; Wilson et al., 2014).

Implementasi perangkat lunak untuk bagian ini mengikuti pola `parse/validate/execute/record`. Input terlebih dahulu dibaca, kemudian diverifikasi, setelah itu engine dijalankan, dan akhirnya keluaran dipaketkan menjadi `result record`. Parameter yang terkait mencakup ``project_id``, ``job_id``, ``spec_hash``, ``source``, ``timestamps``. Keluaran yang diharapkan adalah jejak audit end-to-end. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### ***11.5.1 Prinsip, parameter, dan interpretasi***

Quality control harus membedakan kondisi tidak valid, kondisi meragukan, dan kondisi yang sekadar berbeda dari default. Aturan utama yang perlu dipertahankan adalah bahwa provenance harus dibentuk saat komputasi berjalan, bukan direkonstruksi manual setelah selesai. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan

error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Strategi operasional yang aman untuk audit trail dan provenance adalah memulai dari konfigurasi sederhana, memeriksa log, lalu menambah kompleksitas satu parameter pada satu waktu. Setelah tahap selesai, periksa apakah jejak audit end-to-end benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***11.5.2 Checklist implementasi dan quality control***

- Pastikan parameter `project_id` telah ditentukan dan disimpan.
- Pastikan parameter `job_id` telah ditentukan dan disimpan.
- Pastikan parameter `spec_hash` telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: provenance harus dibentuk saat komputasi berjalan, bukan direkonstruksi manual setelah selesai.
- Pastikan keluaran tahap tersedia sebagai jejak audit end-to-end dan status QC tercatat.

**Sketsa implementasi Python - Bab 11 (pseudocode original)**

```
spec_hash = sha256(canonical_config)
if cache.has_valid(spec_hash):
    return cache.load(spec_hash)
checkpoint.save(stage, partial_result)
```

## **11.6 Rangkuman Bab**

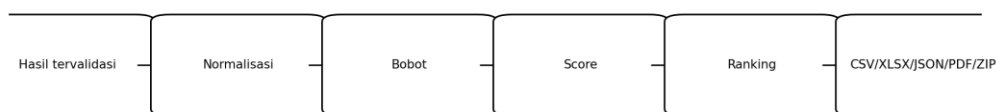
Bab 11 menunjukkan bahwa resume, checkpoint, cache, hash, sqlite, dan provenance harus dipahami sebagai satu sistem yang menghubungkan keputusan ilmiah dan keputusan perangkat lunak. Keandalan QMScreen bergantung pada kemampuan setiap tahap untuk menerima input yang jelas, menjalankan metode dengan parameter eksplisit, mencatat status, mempertahankan hasil parsial secara benar, dan menyediakan provenance. Dengan pola ini, batch yang besar tetap dapat diaudit dan dikembangkan tanpa menjadikan GUI sebagai satu-satunya

sumber kebenaran. Pembaca sebaiknya menggunakan checklist pada setiap subbab sebagai kendali sebelum melanjutkan ke tahap berikutnya.

## BAB 12 RANKING MULTIKRITERIA, EKSPOR, BENCHMARK, DAN VALIDASI HIGH-THROUGHPUT

Bab ini membahas ranking multikriteria, ekspor, benchmark, dan validasi high-throughput sebagai bagian dari workflow QMScreen V1.0.5. Pembahasan menempatkan konsep kimia komputasi, implementasi perangkat lunak, scientific guardrail, quality control, dan reproducibility dalam satu alur. Setiap subbab diakhiri dengan checklist praktis agar pembaca dapat menghubungkan teori dengan penggunaan aplikasi.

### Ranking dan Ekspor



Gambar 12.1. Peta konseptual ranking multikriteria, ekspor, benchmark, dan validasi high-throughput.

Tabel 12.1. Peta materi Bab 12

| Subbab | Fokus                                | Keluaran utama                           |
|--------|--------------------------------------|------------------------------------------|
| 12.1   | Normalisasi descriptor untuk ranking | score dan rank                           |
| 12.2   | Energi formula-sama                  | energy_same_formula component            |
| 12.3   | QC sebagai syarat ranking            | ranked dataframe dengan QC               |
| 12.4   | Ekspor riset                         | artefak ekspor dan checksum              |
| 12.5   | Benchmark dan regression baseline    | laporan benchmark dan peringatan regresi |

### 12.1 Normalisasi descriptor untuk ranking

Pembahasan normalisasi descriptor untuk ranking dimulai dari prinsip bahwa satu angka komputasi hanya bermakna jika struktur, metode, unit, dan status validasinya diketahui. Ranking menggabungkan descriptor dengan skala berbeda melalui normalisasi dan bobot. Elemen yang paling relevan pada tahap ini

meliputi min-max, direction high/low dan weights. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Sandve et al., 2013; Wilkinson et al., 2016; Wilson et al., 2014).

Implementasi perangkat lunak untuk bagian ini mengikuti pola parse/validate/execute/record. Input terlebih dahulu dibaca, kemudian diverifikasi, setelah itu engine dijalankan, dan akhirnya keluaran dipaketkan menjadi result record. Parameter yang terkait mencakup 'min-max', direction high/low, 'weights'. Keluaran yang diharapkan adalah score dan rank. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### ***12.1.1 Prinsip, parameter, dan interpretasi***

Quality control harus membedakan kondisi tidak valid, kondisi meragukan, dan kondisi yang sekadar berbeda dari default. Aturan utama yang perlu dipertahankan adalah bahwa skor gabungan bukan hukum kimia universal; bobot harus merefleksikan tujuan screening. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Strategi operasional yang aman untuk normalisasi descriptor untuk ranking adalah memulai dari konfigurasi sederhana, memeriksa log, lalu menambah kompleksitas satu parameter pada satu waktu. Setelah tahap selesai, periksa apakah score dan rank benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

$$\text{Score} = \Sigma(w_i \cdot x_{i,\text{normalized}}) / \Sigma w_i$$

### ***12.1.2 Checklist implementasi dan quality control***

- Pastikan parameter min-max telah ditentukan dan disimpan.
- Pastikan parameter direction high/low telah ditentukan dan disimpan.
- Pastikan parameter weights telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: skor gabungan bukan hukum kimia universal; bobot harus merefleksikan tujuan screening.
- Pastikan keluaran tahap tersedia sebagai score dan rank dan status QC tercatat.

## **12.2 Energi formula-sama**

Pada QMScreen, energi formula-sama bukan fitur dekoratif antarmuka. Energi total hanya dinormalisasi di dalam kelompok formula sama untuk menghindari perbandingan lintas komposisi yang tidak bermakna. Pendekatan ini membuat tahap tersebut dapat diuji sebagai komponen ilmiah yang mandiri. Elemen yang paling relevan pada tahap ini meliputi groupby formula dan lower energy preferred. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Sandve et al., 2013; Wilkinson et al., 2016; Wilson et al., 2014).

Secara arsitektural, bagian energi formula-sama ditempatkan di luar logika tampilan sebanyak mungkin. Data dari widget hanya menjadi sumber konfigurasi, sedangkan fungsi inti bekerja pada struktur Python yang dapat diuji. Titik kendalinya meliputi groupby formula, lower energy preferred. Keluaran yang diharapkan adalah `energy_same_formula` component. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### ***12.2.1 Prinsip, parameter, dan interpretasi***

Risiko terbesar dalam otomasi adalah membuat proses tampak berhasil ketika asumsi ilmiahnya tidak terpenuhi. Aturan utama yang perlu dipertahankan adalah bahwa aturan ini tidak mengubah energi total menjadi afinitas; ia hanya mencegah misuse dasar. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status

QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Untuk tujuan riset, hasil dari energi formula-sama sebaiknya ditinjau bersama data sebelum dan sesudah tahap tersebut agar perubahan dapat dilacak. Setelah tahap selesai, periksa apakah `energy_same_formula` component benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***12.2.2 Checklist implementasi dan quality control***

- Pastikan parameter `groupby` formula telah ditentukan dan disimpan.
- Pastikan parameter `lower energy preferred` telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: aturan ini tidak mengubah energi total menjadi afinitas; ia hanya mencegah misuse dasar.
- Pastikan keluaran tahap tersedia sebagai `energy_same_formula` component dan status QC tercatat.

## **12.3 QC sebagai syarat ranking**

High-throughput chemistry menuntut aturan yang konsisten ketika jumlah molekul bertambah. Dalam konteks qc sebagai syarat ranking, Kandidat invalid tidak boleh mengungguli kandidat tervalidasi hanya karena descriptor numeriknya ekstrem. Elemen yang paling relevan pada tahap ini meliputi `validated` flag, `missing data` dan `penalty/exclusion`. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Sandve et al., 2013; Wilkinson et al., 2016; Wilson et al., 2014).

Untuk batch besar, implementasi harus tahan terhadap kegagalan parsial. QMScreen karena itu menyimpan status per molekul dan per tahap, sambil mempertahankan parameter seperti `validated` flag, `missing data`,

`penalty/exclusion` pada manifest atau provenance. Keluaran yang diharapkan adalah ranked dataframe dengan QC. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### ***12.3.1 Prinsip, parameter, dan interpretasi***

Ketika workflow dipercepat, kontrol ilmiah tidak boleh ikut dipangkas. Aturan utama yang perlu dipertahankan adalah bahwa ranking harus transparan terhadap data hilang dan kegagalan job. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Pada komputer dengan resource terbatas, qc sebagai syarat ranking perlu diatur bersama jumlah worker, thread, memory, dan biaya tahap berikutnya. Setelah tahap selesai, periksa apakah ranked dataframe dengan QC benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***12.3.2 Checklist implementasi dan quality control***

- Pastikan parameter validated flag telah ditentukan dan disimpan.
- Pastikan parameter missing data telah ditentukan dan disimpan.
- Pastikan parameter penalty/exclusion telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: ranking harus transparan terhadap data hilang dan kegagalan job.
- Pastikan keluaran tahap tersedia sebagai ranked dataframe dengan QC dan status QC tercatat.

## 12.4 Ekspor riset

QMScreen menyediakan CSV, XLSX, JSON, PDF, ZIP proyek, dan database sebagai bentuk keluaran berbeda. Dalam workflow komputasi skala besar, konsep ini perlu didefinisikan sebelum job pertama dijalankan karena keputusan awal akan memengaruhi biaya, keterbandingan, dan kemampuan reproduksi hasil. Elemen yang paling relevan pada tahap ini meliputi table export, archive dan raw project. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Sandve et al., 2013; Wilkinson et al., 2016; Wilson et al., 2014).

Di tingkat implementasi, QMScreen mengubah pilihan pengguna menjadi record konfigurasi dan menjalankan fungsi scientific logic yang sesuai. Komponen seperti table export, 'archive', raw project tidak diperlakukan sebagai nilai sementara, tetapi sebagai bagian dari state proyek yang dapat ditulis ke disk. Keluaran yang diharapkan adalah artefak ekspor dan checksum. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### 12.4.1 Prinsip, parameter, dan interpretasi

Scientific guardrail pada bagian ini sengaja dibuat konservatif. Aturan utama yang perlu dipertahankan adalah bahwa semua format harus berasal dari result record yang sama agar tidak terjadi inkonsistensi. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Dalam penggunaan sehari-hari, pengguna sebaiknya menguji ekspor riset pada satu atau dua molekul kecil sebelum memperbesar batch. Setelah tahap selesai, periksa apakah artefak ekspor dan checksum benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar

kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

#### 12.4.2 Checklist implementasi dan quality control

- Pastikan parameter table export telah ditentukan dan disimpan.
- Pastikan parameter archive telah ditentukan dan disimpan.
- Pastikan parameter raw project telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: semua format harus berasal dari result record yang sama agar tidak terjadi inkonsistensi.
- Pastikan keluaran tahap tersedia sebagai artefak ekspor dan checksum dan status QC tercatat.

**Tabel 12.2. Format ekspor QMScreen**

| Format     | Tujuan utama                   | Isi tipikal                                  |
|------------|--------------------------------|----------------------------------------------|
| CSV        | Analisis tabel lintas aplikasi | Kolom hasil/ranking                          |
| XLSX       | Analisis spreadsheet           | Tabel hasil terformat                        |
| JSON       | Pertukaran data terstruktur    | Record hasil                                 |
| PDF        | Ringkasan human-readable       | Tabel ringkas                                |
| ZIP proyek | Arsip lengkap                  | config, manifest, hasil, log                 |
| SQLite     | Indeks/provenance              | project, molecule, job, result, cache, audit |

### 12.5 Benchmark dan regression baseline

Pembahasan benchmark dan regression baseline dimulai dari prinsip bahwa satu angka komputasi hanya bermakna jika struktur, metode, unit, dan status validasinya diketahui. Benchmark mengukur throughput, runtime, success rate, dan perubahan terhadap baseline untuk mendeteksi regresi. Elemen yang paling relevan pada tahap ini meliputi metrics, baseline JSON dan compare. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian

tak terpisahkan dari hasil (Sandve et al., 2013; Wilkinson et al., 2016; Wilson et al., 2014).

Implementasi perangkat lunak untuk bagian ini mengikuti pola parse/validate/execute/record. Input terlebih dahulu dibaca, kemudian diverifikasi, setelah itu engine dijalankan, dan akhirnya keluaran dipaketkan menjadi result record. Parameter yang terkait mencakup `metrics`, baseline JSON, `compare`. Keluaran yang diharapkan adalah laporan benchmark dan peringatan regresi. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### ***12.5.1 Prinsip, parameter, dan interpretasi***

Quality control harus membedakan kondisi tidak valid, kondisi meragukan, dan kondisi yang sekadar berbeda dari default. Aturan utama yang perlu dipertahankan adalah bahwa benchmark performa harus dibedakan dari validasi ilmiah terhadap data referensi. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Strategi operasional yang aman untuk benchmark dan regression baseline adalah memulai dari konfigurasi sederhana, memeriksa log, lalu menambah kompleksitas satu parameter pada satu waktu. Setelah tahap selesai, periksa apakah laporan benchmark dan peringatan regresi benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***12.5.2 Checklist implementasi dan quality control***

- Pastikan parameter metrics telah ditentukan dan disimpan.
- Pastikan parameter baseline JSON telah ditentukan dan disimpan.

- Pastikan parameter compare telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: benchmark performa harus dibedakan dari validasi ilmiah terhadap data referensi.
- Pastikan keluaran tahap tersedia sebagai laporan benchmark dan peringatan regresi dan status QC tercatat.

#### Sketsa implementasi Python - Bab 12 (pseudocode original)

```
normalized = normalize_descriptors(results)
score = weighted_score(normalized, weights)
ranked = sort_validated_first(score)
```

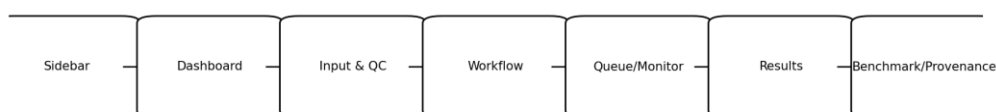
## 12.6 Rangkuman Bab

Bab 12 menunjukkan bahwa ranking multikriteria, ekspor, benchmark, dan validasi high-throughput harus dipahami sebagai satu sistem yang menghubungkan keputusan ilmiah dan keputusan perangkat lunak. Keandalan QMScreen bergantung pada kemampuan setiap tahap untuk menerima input yang jelas, menjalankan metode dengan parameter eksplisit, mencatat status, mempertahankan hasil parsial secara benar, dan menyediakan provenance. Dengan pola ini, batch yang besar tetap dapat diaudit dan dikembangkan tanpa menjadikan GUI sebagai satu-satunya sumber kebenaran. Pembaca sebaiknya menggunakan checklist pada setiap subbab sebagai kendali sebelum melanjutkan ke tahap berikutnya.

## BAB 13 ANTARMUKA STREAMLIT, SIDEBAR, DASHBOARD, DAN SESSION STATE

Bab ini membahas antarmuka streamlit, sidebar, dashboard, dan session state sebagai bagian dari workflow QMScreen V1.0.5. Pembahasan menempatkan konsep kimia komputasi, implementasi perangkat lunak, scientific guardrail, quality control, dan reproducibility dalam satu alur. Setiap subbab diakhiri dengan checklist praktis agar pembaca dapat menghubungkan teori dengan penggunaan aplikasi.

**Peta Antarmuka**



*Gambar 13.1. Peta konseptual antarmuka streamlit, sidebar, dashboard, dan session state.*

**Tabel 13.1. Peta materi Bab 13**

| Subbab | Fokus                            | Keluaran utama                          |
|--------|----------------------------------|-----------------------------------------|
| 13.1   | Peran antarmuka                  | config object dan state                 |
| 13.2   | Sidebar pengaturan proyek        | konfigurasi proyek                      |
| 13.3   | Tab Dashboard dan Input & QC     | selected_specs                          |
| 13.4   | Workflow dan Queue/Monitor       | state running dan lock widget           |
| 13.5   | Results dan Benchmark/Provenance | tampilan yang konsisten dengan database |

### 13.1 Peran antarmuka

Pembahasan peran antarmuka dimulai dari prinsip bahwa satu angka komputasi hanya bermakna jika struktur, metode, unit, dan status validasinya diketahui. Streamlit menyajikan kontrol proyek dan hasil tanpa memindahkan scientific logic ke lapisan presentasi. Elemen yang paling relevan pada tahap ini meliputi sidebar, tabs, forms dan session state. Masing-masing elemen harus memiliki

representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Sandve et al., 2013; Wilkinson et al., 2016; Wilson et al., 2014).

Implementasi perangkat lunak untuk bagian ini mengikuti pola parse/validate/execute/record. Input terlebih dahulu dibaca, kemudian diverifikasi, setelah itu engine dijalankan, dan akhirnya keluaran dipaketkan menjadi result record. Parameter yang terkait mencakup `sidebar`, `tabs`, `forms`, session state. Keluaran yang diharapkan adalah config object dan state. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### ***13.1.1 Prinsip, parameter, dan interpretasi***

Quality control harus membedakan kondisi tidak valid, kondisi meragukan, dan kondisi yang sekadar berbeda dari default. Aturan utama yang perlu dipertahankan adalah bahwa nilai widget harus dikonversi menjadi konfigurasi eksplisit yang disimpan; GUI bukan provenance. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Strategi operasional yang aman untuk peran antarmuka adalah memulai dari konfigurasi sederhana, memeriksa log, lalu menambah kompleksitas satu parameter pada satu waktu. Setelah tahap selesai, periksa apakah config object dan state benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***13.1.2 Checklist implementasi dan quality control***

- Pastikan parameter sidebar telah ditentukan dan disimpan.
- Pastikan parameter tabs telah ditentukan dan disimpan.
- Pastikan parameter forms telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: nilai widget harus dikonversi menjadi konfigurasi eksplisit yang disimpan; GUI bukan provenance.
- Pastikan keluaran tahap tersedia sebagai config object dan state dan status QC tercatat.

## **13.2 Sidebar pengaturan proyek**

Pada QMScreen, sidebar pengaturan proyek bukan fitur dekoratif antarmuka. Sidebar menampung input sampel, resource, conformer, prescreen, DFT, basis, dan frequency. Pendekatan ini membuat tahap tersebut dapat diuji sebagai komponen ilmiah yang mandiri. Elemen yang paling relevan pada tahap ini meliputi project name, workers, threads, memory dan method options. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Sandve et al., 2013; Wilkinson et al., 2016; Wilson et al., 2014).

Secara arsitektural, bagian sidebar pengaturan proyek ditempatkan di luar logika tampilan sebanyak mungkin. Data dari widget hanya menjadi sumber konfigurasi, sedangkan fungsi inti bekerja pada struktur Python yang dapat diuji. Titik kendalinya meliputi project name, 'workers', 'threads', 'memory', method options. Keluaran yang diharapkan adalah konfigurasi proyek. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### ***13.2.1 Prinsip, parameter, dan interpretasi***

Risiko terbesar dalam otomasi adalah membuat proses tampak berhasil ketika asumsi ilmiahnya tidak terpenuhi. Aturan utama yang perlu dipertahankan adalah bahwa default harus terlihat dan tersimpan sehingga hasil tidak bergantung pada state tersembunyi. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status

QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Untuk tujuan riset, hasil dari sidebar pengaturan proyek sebaiknya ditinjau bersama data sebelum dan sesudah tahap tersebut agar perubahan dapat dilacak. Setelah tahap selesai, periksa apakah konfigurasi proyek benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### 13.2.2 Checklist implementasi dan quality control

- Pastikan parameter project name telah ditentukan dan disimpan.
- Pastikan parameter workers telah ditentukan dan disimpan.
- Pastikan parameter threads telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: default harus terlihat dan tersimpan sehingga hasil tidak bergantung pada state tersembunyi.
- Pastikan keluaran tahap tersedia sebagai konfigurasi proyek dan status QC tercatat.

**Tabel 13.2. Pengaturan penting di sidebar**

| Pengaturan       | Arti                 | Saran awal untuk tes             |
|------------------|----------------------|----------------------------------|
| Parallel workers | Job paralel          | 1-2                              |
| CPU threads/job  | Thread CPU per job   | 1-2                              |
| Memory/job       | Batas memori         | Sesuaikan RAM                    |
| Jumlah conformer | Struktur 3D awal     | Kecil dahulu                     |
| GFN2-xTB         | Prescreen cepat      | Aktif bila backend tersedia      |
| Basis optimasi   | Basis tahap optimasi | def2-SVP default UI              |
| Basis final      | Single-point final   | def2-TZVP default UI             |
| SCF conv_tol     | Toleransi SCF        | Default lalu perketat bila perlu |

| Pengaturan | Arti                  | Saran awal untuk tes  |
|------------|-----------------------|-----------------------|
| Frequency  | Hessian/minimum check | Aktif bila diperlukan |

### 13.3 Tab Dashboard dan Input & QC

High-throughput chemistry menuntut aturan yang konsisten ketika jumlah molekul bertambah. Dalam konteks tab dashboard dan input & qc, Dashboard memberi ringkasan input unik, duplikat, CPU, RAM; Input & QC memungkinkan koreksi charge, multiplicity, priority. Elemen yang paling relevan pada tahap ini meliputi data editor, validation rows dan metrics. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Sandve et al., 2013; Wilkinson et al., 2016; Wilson et al., 2014).

Untuk batch besar, implementasi harus tahan terhadap kegagalan parsial. QMScreen karena itu menyimpan status per molekul dan per tahap, sambil mempertahankan parameter seperti data editor, validation rows, 'metrics' pada manifest atau provenance. Keluaran yang diharapkan adalah `selected_specs`. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

#### 13.3.1 Prinsip, parameter, dan interpretasi

Ketika workflow dipercepat, kontrol ilmiah tidak boleh ikut dipangkas. Aturan utama yang perlu dipertahankan adalah bahwa editable fields harus divalidasi ulang sebelum run. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Pada komputer dengan resource terbatas, tab dashboard dan input & qc perlu diatur bersama jumlah worker, thread, memory, dan biaya tahap berikutnya. Setelah tahap selesai, periksa apakah `selected_specs` benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***13.3.2 Checklist implementasi dan quality control***

- Pastikan parameter data editor telah ditentukan dan disimpan.
- Pastikan parameter validation rows telah ditentukan dan disimpan.
- Pastikan parameter metrics telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: editable fields harus divalidasi ulang sebelum run.
- Pastikan keluaran tahap tersedia sebagai `selected_specs` dan status QC tercatat.

## **13.4 Workflow dan Queue/Monitor**

Tab workflow merangkum parameter, sedangkan monitor menjalankan batch dan menampilkan status. Dalam workflow komputasi skala besar, konsep ini perlu didefinisikan sebelum job pertama dijalankan karena keputusan awal akan memengaruhi biaya, keterbandingan, dan kemampuan reproduksi hasil. Elemen yang paling relevan pada tahap ini meliputi start batch, stop flag, log dan progress. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Sandve et al., 2013; Wilkinson et al., 2016; Wilson et al., 2014).

Di tingkat implementasi, QMScreen mengubah pilihan pengguna menjadi record konfigurasi dan menjalankan fungsi scientific logic yang sesuai. Komponen seperti start batch, stop flag, 'log', 'progress' tidak diperlakukan sebagai nilai sementara, tetapi sebagai bagian dari state proyek yang dapat ditulis ke disk. Keluaran yang diharapkan adalah state running dan lock widget. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil

dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

#### ***13.4.1 Prinsip, parameter, dan interpretasi***

Scientific guardrail pada bagian ini sengaja dibuat konservatif. Aturan utama yang perlu dipertahankan adalah bahwa UI harus mencegah perubahan parameter kritis saat job sedang berjalan. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Dalam penggunaan sehari-hari, pengguna sebaiknya menguji workflow dan queue/monitor pada satu atau dua molekul kecil sebelum memperbesar batch. Setelah tahap selesai, periksa apakah state running dan lock widget benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

#### ***13.4.2 Checklist implementasi dan quality control***

- Pastikan parameter start batch telah ditentukan dan disimpan.
- Pastikan parameter stop flag telah ditentukan dan disimpan.
- Pastikan parameter log telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: UI harus mencegah perubahan parameter kritis saat job sedang berjalan.
- Pastikan keluaran tahap tersedia sebagai state running dan lock widget dan status QC tercatat.

### **13.5 Results dan Benchmark/Provenance**

Pembahasan results dan benchmark/provenance dimulai dari prinsip bahwa satu angka komputasi hanya bermakna jika struktur, metode, unit, dan status validasinya diketahui. Hasil menampilkan metric, 3D, orbital, charge, frequency, thermochemistry, dan provenance; benchmark menguji kinerja. Elemen yang

paling relevan pada tahap ini meliputi detail tabs, downloads dan baseline. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Sandve et al., 2013; Wilkinson et al., 2016; Wilson et al., 2014).

Implementasi perangkat lunak untuk bagian ini mengikuti pola parse/validate/execute/record. Input terlebih dahulu dibaca, kemudian diverifikasi, setelah itu engine dijalankan, dan akhirnya keluaran dipaketkan menjadi result record. Parameter yang terkait mencakup detail tabs, 'downloads', 'baseline'. Keluaran yang diharapkan adalah tampilan yang konsisten dengan database. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### ***13.5.1 Prinsip, parameter, dan interpretasi***

Quality control harus membedakan kondisi tidak valid, kondisi meragukan, dan kondisi yang sekadar berbeda dari default. Aturan utama yang perlu dipertahankan adalah bahwa visualisasi harus mengikuti record tersimpan, bukan menghitung angka berbeda di sisi UI. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Strategi operasional yang aman untuk results dan benchmark/provenance adalah memulai dari konfigurasi sederhana, memeriksa log, lalu menambah kompleksitas satu parameter pada satu waktu. Setelah tahap selesai, periksa apakah tampilan yang konsisten dengan database benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***13.5.2 Checklist implementasi dan quality control***

- Pastikan parameter detail tabs telah ditentukan dan disimpan.
- Pastikan parameter downloads telah ditentukan dan disimpan.
- Pastikan parameter baseline telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: visualisasi harus mengikuti record tersimpan, bukan menghitung angka berbeda di sisi UI.
- Pastikan keluaran tahap tersedia sebagai tampilan yang konsisten dengan database dan status QC tercatat.

## **13.6 Rangkuman Bab**

Bab 13 menunjukkan bahwa antarmuka streamlit, sidebar, dashboard, dan session state harus dipahami sebagai satu sistem yang menghubungkan keputusan ilmiah dan keputusan perangkat lunak. Keandalan QMScreen bergantung pada kemampuan setiap tahap untuk menerima input yang jelas, menjalankan metode dengan parameter eksplisit, mencatat status, mempertahankan hasil parsial secara benar, dan menyediakan provenance. Dengan pola ini, batch yang besar tetap dapat diaudit dan dikembangkan tanpa menjadikan GUI sebagai satu-satunya sumber kebenaran. Pembaca sebaiknya menggunakan checklist pada setiap subbab sebagai kendali sebelum melanjutkan ke tahap berikutnya.

## BAB 14 TUTORIAL PENGGUNAAN QMSCREEN DARI INSTALASI SAMPAI EKSPOR

Bab ini membahas tutorial penggunaan qmscreen dari instalasi sampai ekspor sebagai bagian dari workflow QMScreen V1.0.5. Pembahasan menempatkan konsep kimia komputasi, implementasi perangkat lunak, scientific guardrail, quality control, dan reproducibility dalam satu alur. Setiap subbab diakhiri dengan checklist praktis agar pembaca dapat menghubungkan teori dengan penggunaan aplikasi.

### Tutorial End-to-End Ethanol



Gambar 14.1. Peta konseptual tutorial penggunaan qmscreen dari instalasi sampai ekspor.

Tabel 14.1. Peta materi Bab 14

| Subbab | Fokus                  | Keluaran utama                             |
|--------|------------------------|--------------------------------------------|
| 14.1   | Persiapan lingkungan   | environment snapshot dan backend checklist |
| 14.2   | Menjalankan aplikasi   | halaman QMScreen V1.0.5                    |
| 14.3   | Latihan cepat Ethanol  | satu job end-to-end                        |
| 14.4   | Membaca hasil          | result record yang dapat diaudit           |
| 14.5   | Ekspor dan pengarsipan | paket arsip proyek                         |

### 14.1 Persiapan lingkungan

Pembahasan persiapan lingkungan dimulai dari prinsip bahwa satu angka komputasi hanya bermakna jika struktur, metode, unit, dan status validasinya diketahui. QMScreen memerlukan Python dan stack ilmiah seperti Streamlit, NumPy, pandas, SciPy, RDKit, PySCF, geomeTRIC, serta paket ekspor. Elemen yang paling relevan pada tahap ini meliputi auto-install package, backend status dan writable data directory. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI.

Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Sun et al., 2018; Sun et al., 2020).

Implementasi perangkat lunak untuk bagian ini mengikuti pola parse/validate/execute/record. Input terlebih dahulu dibaca, kemudian diverifikasi, setelah itu engine dijalankan, dan akhirnya keluaran dipaketkan menjadi result record. Parameter yang terkait mencakup auto-install package, backend status, writable data directory. Keluaran yang diharapkan adalah environment snapshot dan backend checklist. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

#### ***14.1.1 Prinsip, parameter, dan interpretasi***

Quality control harus membedakan kondisi tidak valid, kondisi meragukan, dan kondisi yang sekadar berbeda dari default. Aturan utama yang perlu dipertahankan adalah bahwa instalasi otomatis membantu tetapi lingkungan riset final sebaiknya dicatat dan dikendalikan. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Strategi operasional yang aman untuk persiapan lingkungan adalah memulai dari konfigurasi sederhana, memeriksa log, lalu menambah kompleksitas satu parameter pada satu waktu. Setelah tahap selesai, periksa apakah environment snapshot dan backend checklist benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***14.1.2 Checklist implementasi dan quality control***

- Pastikan parameter auto-install package telah ditentukan dan disimpan.
- Pastikan parameter backend status telah ditentukan dan disimpan.
- Pastikan parameter writable data directory telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: instalasi otomatis membantu tetapi lingkungan riset final sebaiknya dicatat dan dikendalikan.
- Pastikan keluaran tahap tersedia sebagai environment snapshot dan backend checklist dan status QC tercatat.

## **14.2 Menjalankan aplikasi**

Pada QMScreen, menjalankan aplikasi bukan fitur dekoratif antarmuka. Aplikasi dapat diluncurkan melalui Python sehingga Streamlit dijalankan otomatis, atau langsung melalui perintah `streamlit run`. Pendekatan ini membuat tahap tersebut dapat diuji sebagai komponen ilmiah yang mandiri. Elemen yang paling relevan pada tahap ini meliputi local browser, server address dan sidebar. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Sun et al., 2018; Sun et al., 2020).

Secara arsitektural, bagian menjalankan aplikasi ditempatkan di luar logika tampilan sebanyak mungkin. Data dari widget hanya menjadi sumber konfigurasi, sedangkan fungsi inti bekerja pada struktur Python yang dapat diuji. Titik kendalinya meliputi local browser, server address, `sidebar`. Keluaran yang diharapkan adalah halaman QMScreen V1.0.5. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### ***14.2.1 Prinsip, parameter, dan interpretasi***

Risiko terbesar dalam otomasi adalah membuat proses tampak berhasil ketika asumsi ilmiahnya tidak terpenuhi. Aturan utama yang perlu dipertahankan adalah bahwa jangan menjalankan batch berat sebelum backend dan direktori data dinyatakan siap. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan

interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Untuk tujuan riset, hasil dari menjalankan aplikasi sebaiknya ditinjau bersama data sebelum dan sesudah tahap tersebut agar perubahan dapat dilacak. Setelah tahap selesai, periksa apakah halaman QMScreen V1.0.5 benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

#### ***14.2.2 Checklist implementasi dan quality control***

- Pastikan parameter local browser telah ditentukan dan disimpan.
- Pastikan parameter server address telah ditentukan dan disimpan.
- Pastikan parameter sidebar telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: jangan menjalankan batch berat sebelum backend dan direktori data dinyatakan siap.
- Pastikan keluaran tahap tersedia sebagai halaman QMScreen V1.0.5 dan status QC tercatat.

### **14.3 Latihan cepat Ethanol**

High-throughput chemistry menuntut aturan yang konsisten ketika jumlah molekul bertambah. Dalam konteks latihan cepat ethanol, Ethanol dipakai sebagai molekul kecil untuk memeriksa keseluruhan pipeline sebelum batch besar. Elemen yang paling relevan pada tahap ini meliputi Ethanol=CCO, charge 0, multiplicity 1 dan small conformer count. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Sun et al., 2018; Sun et al., 2020).

Untuk batch besar, implementasi harus tahan terhadap kegagalan parsial. QMScreen karena itu menyimpan status per molekul dan per tahap, sambil mempertahankan parameter seperti `Ethanol=CCO`, charge 0, multiplicity 1,

small conformer count pada manifest atau provenance. Keluaran yang diharapkan adalah satu job end-to-end. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### ***14.3.1 Prinsip, parameter, dan interpretasi***

Ketika workflow dipercepat, kontrol ilmiah tidak boleh ikut dipangkas. Aturan utama yang perlu dipertahankan adalah bahwa uji cepat bukan parameter publikasi; setelah pipeline sehat, metode riset perlu disetel. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Pada komputer dengan resource terbatas, latihan cepat ethanol perlu diatur bersama jumlah worker, thread, memory, dan biaya tahap berikutnya. Setelah tahap selesai, periksa apakah satu job end-to-end benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***14.3.2 Checklist implementasi dan quality control***

- Pastikan parameter Ethanol=CCO telah ditentukan dan disimpan.
- Pastikan parameter charge 0 telah ditentukan dan disimpan.
- Pastikan parameter multiplicity 1 telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: uji cepat bukan parameter publikasi; setelah pipeline sehat, metode riset perlu disetel.
- Pastikan keluaran tahap tersedia sebagai satu job end-to-end dan status QC tercatat.

## 14.4 Membaca hasil

Interpretasi dimulai dari status QC, energi, gap, dipole, frequency, kemudian provenance dan log. Dalam workflow komputasi skala besar, konsep ini perlu didefinisikan sebelum job pertama dijalankan karena keputusan awal akan memengaruhi biaya, keterbandingan, dan kemampuan reproduksi hasil. Elemen yang paling relevan pada tahap ini meliputi VALID/QC/FAILED, metric cards dan detail tabs. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Sun et al., 2018; Sun et al., 2020).

Di tingkat implementasi, QMScreen mengubah pilihan pengguna menjadi record konfigurasi dan menjalankan fungsi scientific logic yang sesuai. Komponen seperti `VALID/QC/FAILED`, metric cards, detail tabs tidak diperlakukan sebagai nilai sementara, tetapi sebagai bagian dari state proyek yang dapat ditulis ke disk. Keluaran yang diharapkan adalah result record yang dapat diaudit. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### 14.4.1 Prinsip, parameter, dan interpretasi

Scientific guardrail pada bagian ini sengaja dibuat konservatif. Aturan utama yang perlu dipertahankan adalah bahwa angka tanpa status konvergensi dan metode tidak boleh dijadikan kesimpulan. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Dalam penggunaan sehari-hari, pengguna sebaiknya menguji membaca hasil pada satu atau dua molekul kecil sebelum memperbesar batch. Setelah tahap selesai, periksa apakah result record yang dapat diaudit benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan

antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

#### ***14.4.2 Checklist implementasi dan quality control***

- Pastikan parameter VALID/QC/FAILED telah ditentukan dan disimpan.
- Pastikan parameter metric cards telah ditentukan dan disimpan.
- Pastikan parameter detail tabs telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: angka tanpa status konvergensi dan metode tidak boleh dijadikan kesimpulan.
- Pastikan keluaran tahap tersedia sebagai result record yang dapat diaudit dan status QC tercatat.

### **14.5 Ekspor dan pengarsipan**

Pembahasan ekspor dan pengarsipan dimulai dari prinsip bahwa satu angka komputasi hanya bermakna jika struktur, metode, unit, dan status validasinya diketahui. Unduh tabel dan ZIP proyek setelah hasil diperiksa agar analisis lanjutan tetap memiliki konteks. Elemen yang paling relevan pada tahap ini meliputi CSV, XLSX, JSON, PDF, ZIP dan SQLite. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Sun et al., 2018; Sun et al., 2020).

Implementasi perangkat lunak untuk bagian ini mengikuti pola parse/validate/execute/record. Input terlebih dahulu dibaca, kemudian diverifikasi, setelah itu engine dijalankan, dan akhirnya keluaran dipaketkan menjadi result record. Parameter yang terkait mencakup `CSV`, `XLSX`, `JSON`, `PDF`, `ZIP`, `SQLite`. Keluaran yang diharapkan adalah paket arsip proyek. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

#### ***14.5.1 Prinsip, parameter, dan interpretasi***

Quality control harus membedakan kondisi tidak valid, kondisi meragukan, dan kondisi yang sekadar berbeda dari default. Aturan utama yang perlu dipertahankan

adalah bahwa jangan hanya menyimpan screenshot; simpan data mentah dan konfigurasi. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Strategi operasional yang aman untuk ekspor dan pengarsipan adalah memulai dari konfigurasi sederhana, memeriksa log, lalu menambah kompleksitas satu parameter pada satu waktu. Setelah tahap selesai, periksa apakah paket arsip proyek benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

#### ***14.5.2 Checklist implementasi dan quality control***

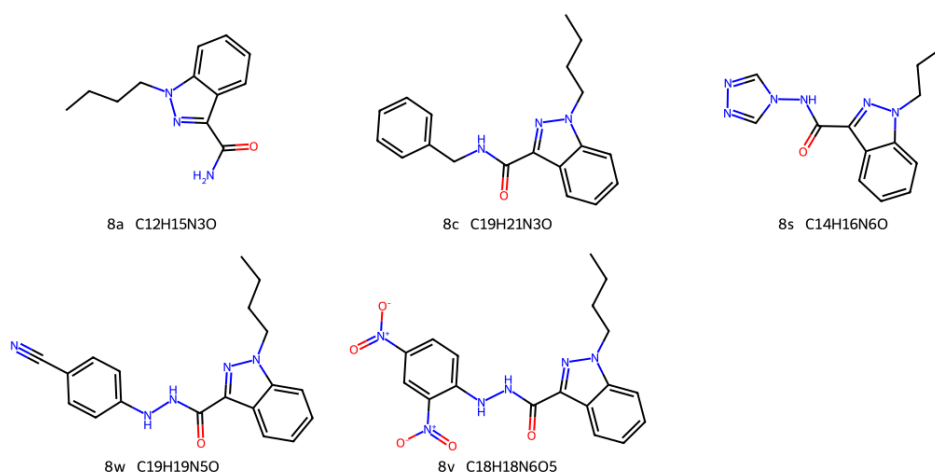
- Pastikan parameter CSV telah ditentukan dan disimpan.
- Pastikan parameter XLSX telah ditentukan dan disimpan.
- Pastikan parameter JSON telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: jangan hanya menyimpan screenshot; simpan data mentah dan konfigurasi.
- Pastikan keluaran tahap tersedia sebagai paket arsip proyek dan status QC tercatat.

### **14.6 Rangkuman Bab**

Bab 14 menunjukkan bahwa tutorial penggunaan qmscreen dari instalasi sampai ekspor harus dipahami sebagai satu sistem yang menghubungkan keputusan ilmiah dan keputusan perangkat lunak. Keandalan QMScreen bergantung pada kemampuan setiap tahap untuk menerima input yang jelas, menjalankan metode dengan parameter eksplisit, mencatat status, mempertahankan hasil parsial secara benar, dan menyediakan provenance. Dengan pola ini, batch yang besar tetap dapat diaudit dan dikembangkan tanpa menjadikan GUI sebagai satu-satunya sumber kebenaran. Pembaca sebaiknya menggunakan checklist pada setiap subbab sebagai kendali sebelum melanjutkan ke tahap berikutnya.

## BAB 15 STUDI KASUS KATALOG 8A, 8C, 8S, 8W, DAN 8V

Bab ini membahas studi kasus katalog 8a, 8c, 8s, 8w, dan 8v sebagai bagian dari workflow QMScreen V1.0.5. Pembahasan menempatkan konsep kimia komputasi, implementasi perangkat lunak, scientific guardrail, quality control, dan reproducibility dalam satu alur. Setiap subbab diakhiri dengan checklist praktis agar pembaca dapat menghubungkan teori dengan penggunaan aplikasi.



Gambar 15.1. Peta konseptual studi kasus katalog 8a, 8c, 8s, 8w, dan 8v.

Tabel 15.1. Peta materi Bab 15

| Subbab | Fokus                             | Keluaran utama                       |
|--------|-----------------------------------|--------------------------------------|
| 15.1   | Karakter data katalog             | dataset kasus lima molekul           |
| 15.2   | Persiapan batch studi kasus       | batch input tervalidasi              |
| 15.3   | Strategi conformer dan prescreen  | conformer terpilih per molekul       |
| 15.4   | Perbandingan descriptor yang aman | tabel komparatif terstruktur         |
| 15.5   | Pelaporan studi kasus             | template pelaporan tanpa data fiktif |

### 15.1 Karakter data katalog

Pembahasan karakter data katalog dimulai dari prinsip bahwa satu angka komputasi hanya bermakna jika struktur, metode, unit, dan status validasinya diketahui. Katalog terlampir menyimpan lima turunan indazole dengan SMILES, InChI, formula, massa molekul, charge, spin, dan molblock 3D. Elemen yang paling relevan pada tahap ini meliputi 8a, 8c, 8s, 8w, 8v dan persisted SMILES.

Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Cramer, 2004; Jensen, 2017).

Implementasi perangkat lunak untuk bagian ini mengikuti pola parse/validate/execute/record. Input terlebih dahulu dibaca, kemudian diverifikasi, setelah itu engine dijalankan, dan akhirnya keluaran dipaketkan menjadi result record. Parameter yang terkait mencakup `8a`, `8c`, `8s`, `8w`, `8v`, persisted SMILES. Keluaran yang diharapkan adalah dataset kasus lima molekul. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### ***15.1.1 Prinsip, parameter, dan interpretasi***

Quality control harus membedakan kondisi tidak valid, kondisi meragukan, dan kondisi yang sekadar berbeda dari default. Aturan utama yang perlu dipertahankan adalah bahwa data katalog adalah input, bukan hasil DFT; descriptor kuantum harus dihitung melalui workflow. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Strategi operasional yang aman untuk karakter data katalog adalah memulai dari konfigurasi sederhana, memeriksa log, lalu menambah kompleksitas satu parameter pada satu waktu. Setelah tahap selesai, periksa apakah dataset kasus lima molekul benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### 15.1.2 Checklist implementasi dan quality control

- Pastikan parameter 8a telah ditentukan dan disimpan.
- Pastikan parameter 8c telah ditentukan dan disimpan.
- Pastikan parameter 8s telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: data katalog adalah input, bukan hasil DFT; descriptor kuantum harus dihitung melalui workflow.
- Pastikan keluaran tahap tersedia sebagai dataset kasus lima molekul dan status QC tercatat.

**Tabel 15.2. Data katalog molekul terlampir**

| Nama | Formula                                                          | MW          | Charge | Spin | SMILES                                                                   |
|------|------------------------------------------------------------------|-------------|--------|------|--------------------------------------------------------------------------|
| 8a   | C <sub>12</sub> H <sub>15</sub> N <sub>3</sub><br>O              | 217.27<br>2 | 0      | 0    | <chem>CCCCn1nc(C(N)=O)c2cccc21</chem>                                    |
| 8c   | C <sub>19</sub> H <sub>21</sub> N <sub>3</sub><br>O              | 307.39<br>7 | 0      | 0    | <chem>CCCCn1nc(C(=O)NCc2cccc2)c2cccc21</chem>                            |
| 8s   | C <sub>14</sub> H <sub>16</sub> N <sub>6</sub><br>O              | 284.32<br>3 | 0      | 0    | <chem>CCCCn1nc(C(=O)Nn2cnnc2)c2cccc21</chem>                             |
| 8w   | C <sub>19</sub> H <sub>19</sub> N <sub>5</sub><br>O              | 333.39<br>5 | 0      | 0    | <chem>CCCCn1nc(C(=O)NNc2ccc(C#N)cc2)c2cccc21</chem>                      |
| 8v   | C <sub>18</sub> H <sub>18</sub> N <sub>6</sub><br>O <sub>5</sub> | 398.37<br>9 | 0      | 0    | <chem>CCCCn1nc(C(=O)NNc2ccc([N+](=O)[O-])cc2[N+](=O)[O-])c2cccc21</chem> |

## 15.2 Persiapan batch studi kasus

Pada QMScreen, persiapan batch studi kasus bukan fitur dekoratif antarmuka. Kelima struktur dapat dipilih bersama lalu diperiksa charge, multiplicity, stereokimia, dan duplikasi. Pendekatan ini membuat tahap tersebut dapat diuji sebagai komponen ilmiah yang mandiri. Elemen yang paling relevan pada tahap ini meliputi charge 0, spin 0 pada katalog dan formula berbeda. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Cramer, 2004; Jensen, 2017).

Secara arsitektural, bagian persiapan batch studi kasus ditempatkan di luar logika tampilan sebanyak mungkin. Data dari widget hanya menjadi sumber konfigurasi, sedangkan fungsi inti bekerja pada struktur Python yang dapat diuji. Titik kendalinya meliputi charge 0, spin 0 pada katalog, formula berbeda. Keluaran yang diharapkan adalah batch input tervalidasi. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### ***15.2.1 Prinsip, parameter, dan interpretasi***

Risiko terbesar dalam otomasi adalah membuat proses tampak berhasil ketika asumsi ilmiahnya tidak terpenuhi. Aturan utama yang perlu dipertahankan adalah bahwa karena formula berbeda, energi total final tidak boleh diranking langsung antar kelima molekul. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Untuk tujuan riset, hasil dari persiapan batch studi kasus sebaiknya ditinjau bersama data sebelum dan sesudah tahap tersebut agar perubahan dapat dilacak. Setelah tahap selesai, periksa apakah batch input tervalidasi benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***15.2.2 Checklist implementasi dan quality control***

- Pastikan parameter charge 0 telah ditentukan dan disimpan.
- Pastikan parameter spin 0 pada katalog telah ditentukan dan disimpan.
- Pastikan parameter formula berbeda telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: karena formula berbeda, energi total final tidak boleh diranking langsung antar kelima molekul.

- Pastikan keluaran tahap tersedia sebagai batch input tervalidasi dan status QC tercatat.

### 15.3 Strategi conformer dan prescreen

High-throughput chemistry menuntut aturan yang konsisten ketika jumlah molekul bertambah. Dalam konteks strategi conformer dan prescreen, Ukuran dan fleksibilitas substituen berbeda sehingga kebutuhan conformer dapat berbeda. Elemen yang paling relevan pada tahap ini meliputi butyl chain, aromatic rings, hydrazide dan nitrile/nitro substituents. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Cramer, 2004; Jensen, 2017).

Untuk batch besar, implementasi harus tahan terhadap kegagalan parsial. QMScreen karena itu menyimpan status per molekul dan per tahap, sambil mempertahankan parameter seperti butyl chain, aromatic rings, `hydrazide`, nitrile/nitro substituents pada manifest atau provenance. Keluaran yang diharapkan adalah conformer terpilih per molekul. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

#### 15.3.1 Prinsip, parameter, dan interpretasi

Ketika workflow dipercepat, kontrol ilmiah tidak boleh ikut dipangkas. Aturan utama yang perlu dipertahankan adalah bahwa jumlah conformer harus cukup untuk fleksibilitas tetapi tetap realistis terhadap resource. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Pada komputer dengan resource terbatas, strategi conformer dan prescreen perlu diatur bersama jumlah worker, thread, memory, dan biaya tahap berikutnya. Setelah tahap selesai, periksa apakah conformer terpilih per molekul benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***15.3.2 Checklist implementasi dan quality control***

- Pastikan parameter butyl chain telah ditentukan dan disimpan.
- Pastikan parameter aromatic rings telah ditentukan dan disimpan.
- Pastikan parameter hydrazide telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: jumlah conformer harus cukup untuk fleksibilitas tetapi tetap realistis terhadap resource.
- Pastikan keluaran tahap tersedia sebagai conformer terpilih per molekul dan status QC tercatat.

## **15.4 Perbandingan descriptor yang aman**

Gap, dipole, MW, TPSA, dan descriptor lain dapat dibandingkan dengan catatan metode serta QC yang sama. Dalam workflow komputasi skala besar, konsep ini perlu didefinisikan sebelum job pertama dijalankan karena keputusan awal akan memengaruhi biaya, keterbandingan, dan kemampuan reproduksi hasil. Elemen yang paling relevan pada tahap ini meliputi same DFT functional/basis, validated jobs dan missing data. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Cramer, 2004; Jensen, 2017).

Di tingkat implementasi, QMScreen mengubah pilihan pengguna menjadi record konfigurasi dan menjalankan fungsi scientific logic yang sesuai. Komponen seperti same DFT functional/basis, validated jobs, missing data tidak diperlakukan sebagai nilai sementara, tetapi sebagai bagian dari state proyek yang dapat ditulis ke disk. Keluaran yang diharapkan adalah tabel komparatif terstruktur. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika

hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

#### ***15.4.1 Prinsip, parameter, dan interpretasi***

Scientific guardrail pada bagian ini sengaja dibuat konservatif. Aturan utama yang perlu dipertahankan adalah bahwa perbandingan bukan prediksi aktivitas biologis otomatis; tujuan riset menentukan makna descriptor. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Dalam penggunaan sehari-hari, pengguna sebaiknya menguji perbandingan descriptor yang aman pada satu atau dua molekul kecil sebelum memperbesar batch. Setelah tahap selesai, periksa apakah tabel komparatif terstruktur benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

#### ***15.4.2 Checklist implementasi dan quality control***

- Pastikan parameter same DFT functional/basis telah ditentukan dan disimpan.
- Pastikan parameter validated jobs telah ditentukan dan disimpan.
- Pastikan parameter missing data telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: perbandingan bukan prediksi aktivitas biologis otomatis; tujuan riset menentukan makna descriptor.
- Pastikan keluaran tahap tersedia sebagai tabel komparatif terstruktur dan status QC tercatat.

### **15.5 Pelaporan studi kasus**

Pembahasan pelaporan studi kasus dimulai dari prinsip bahwa satu angka komputasi hanya bermakna jika struktur, metode, unit, dan status validasinya diketahui. Laporan harus menyertakan input canonical, parameter, status setiap

tahap, hasil, dan batas interpretasi. Elemen yang paling relevan pada tahap ini meliputi project manifest, result table, provenance dan archive. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Cramer, 2004; Jensen, 2017).

Implementasi perangkat lunak untuk bagian ini mengikuti pola parse/validate/execute/record. Input terlebih dahulu dibaca, kemudian diverifikasi, setelah itu engine dijalankan, dan akhirnya keluaran dipaketkan menjadi result record. Parameter yang terkait mencakup project manifest, result table, 'provenance', 'archive'. Keluaran yang diharapkan adalah template pelaporan tanpa data fiktif. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

#### ***15.5.1 Prinsip, parameter, dan interpretasi***

Quality control harus membedakan kondisi tidak valid, kondisi meragukan, dan kondisi yang sekadar berbeda dari default. Aturan utama yang perlu dipertahankan adalah bahwa jangan melaporkan angka komputasi yang tidak benar-benar dihitung; buku ini tidak mengarang hasil energi. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Strategi operasional yang aman untuk pelaporan studi kasus adalah memulai dari konfigurasi sederhana, memeriksa log, lalu menambah kompleksitas satu parameter pada satu waktu. Setelah tahap selesai, periksa apakah template pelaporan tanpa data fiktif benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu

disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***15.5.2 Checklist implementasi dan quality control***

- Pastikan parameter project manifest telah ditentukan dan disimpan.
- Pastikan parameter result table telah ditentukan dan disimpan.
- Pastikan parameter provenance telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: jangan melaporkan angka komputasi yang tidak benar-benar dihitung; buku ini tidak mengarang hasil energi.
- Pastikan keluaran tahap tersedia sebagai template pelaporan tanpa data fiktif dan status QC tercatat.

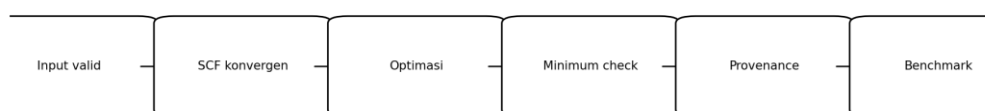
## **15.6 Rangkuman Bab**

Bab 15 menunjukkan bahwa studi kasus katalog 8a, 8c, 8s, 8w, dan 8v harus dipahami sebagai satu sistem yang menghubungkan keputusan ilmiah dan keputusan perangkat lunak. Keandalan QMScreen bergantung pada kemampuan setiap tahap untuk menerima input yang jelas, menjalankan metode dengan parameter eksplisit, mencatat status, mempertahankan hasil parsial secara benar, dan menyediakan provenance. Dengan pola ini, batch yang besar tetap dapat diaudit dan dikembangkan tanpa menjadikan GUI sebagai satu-satunya sumber kebenaran. Pembaca sebaiknya menggunakan checklist pada setiap subbab sebagai kendali sebelum melanjutkan ke tahap berikutnya.

## BAB 16 VALIDASI, TROUBLESHOOTING, ERROR CODE, DAN QUALITY ASSURANCE

Bab ini membahas validasi, troubleshooting, error code, dan quality assurance sebagai bagian dari workflow QMScreen V1.0.5. Pembahasan menempatkan konsep kimia komputasi, implementasi perangkat lunak, scientific guardrail, quality control, dan reproducibility dalam satu alur. Setiap subbab diakhiri dengan checklist praktis agar pembaca dapat menghubungkan teori dengan penggunaan aplikasi.

**Piramida Validasi**



*Gambar 16.1. Peta konseptual validasi, troubleshooting, error code, dan quality assurance.*

**Tabel 16.1. Peta materi Bab 16**

| Subbab | Fokus                            | Keluaran utama              |
|--------|----------------------------------|-----------------------------|
| 16.1   | Validasi berlapis                | QC summary                  |
| 16.2   | Panduan QMS-E001 sampai QMS-E010 | error guide                 |
| 16.3   | Troubleshooting SCF dan optimasi | catatan percobaan dan hasil |
| 16.4   | Regression testing               | laporan compare benchmark   |
| 16.5   | Checklist sebelum publikasi      | checklist publikasi         |

### 16.1 Validasi berlapis

Pembahasan validasi berlapis dimulai dari prinsip bahwa satu angka komputasi hanya bermakna jika struktur, metode, unit, dan status validasinya diketahui. Validasi QMScreen mencakup input, proses numerik, hasil ilmiah, dan integritas data. Elemen yang paling relevan pada tahap ini meliputi input QC, SCF, optimization, frequency, provenance dan benchmark. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung

pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Sandve et al., 2013; Wilkinson et al., 2016; Wilson et al., 2014).

Implementasi perangkat lunak untuk bagian ini mengikuti pola parse/validate/execute/record. Input terlebih dahulu dibaca, kemudian diverifikasi, setelah itu engine dijalankan, dan akhirnya keluaran dipaketkan menjadi result record. Parameter yang terkait mencakup input QC, `SCF`, `optimization`, `frequency`, `provenance`, `benchmark`. Keluaran yang diharapkan adalah QC summary. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### ***16.1.1 Prinsip, parameter, dan interpretasi***

Quality control harus membedakan kondisi tidak valid, kondisi meragukan, dan kondisi yang sekadar berbeda dari default. Aturan utama yang perlu dipertahankan adalah bahwa satu indikator sukses tidak cukup; validasi harus mengikuti tahap yang relevan. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Strategi operasional yang aman untuk validasi berlapis adalah memulai dari konfigurasi sederhana, memeriksa log, lalu menambah kompleksitas satu parameter pada satu waktu. Setelah tahap selesai, periksa apakah QC summary benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***16.1.2 Checklist implementasi dan quality control***

- Pastikan parameter input QC telah ditentukan dan disimpan.
- Pastikan parameter SCF telah ditentukan dan disimpan.
- Pastikan parameter optimization telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: satu indikator sukses tidak cukup; validasi harus mengikuti tahap yang relevan.
- Pastikan keluaran tahap tersedia sebagai QC summary dan status QC tercatat.

## **16.2 Panduan QMS-E001 sampai QMS-E010**

Pada QMScreen, panduan qms-e001 sampai qms-e010 bukan fitur dekoratif antarmuka. Error code mempermudah pemetaan gejala ke lokasi kegagalan. Pendekatan ini membuat tahap tersebut dapat diuji sebagai komponen ilmiah yang mandiri. Elemen yang paling relevan pada tahap ini meliputi input, spin, 3D, xTB, SCF, optimization, frequency, resource, parser dan export. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Sandve et al., 2013; Wilkinson et al., 2016; Wilson et al., 2014).

Secara arsitektural, bagian panduan qms-e001 sampai qms-e010 ditempatkan di luar logika tampilan sebanyak mungkin. Data dari widget hanya menjadi sumber konfigurasi, sedangkan fungsi inti bekerja pada struktur Python yang dapat diuji. Titik kendalinya meliputi ``input``, ``spin``, ``3D``, ``xTB``, ``SCF``, ``optimization``, ``frequency``, ``resource``, ``parser``, ``export``. Keluaran yang diharapkan adalah error guide. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### ***16.2.1 Prinsip, parameter, dan interpretasi***

Risiko terbesar dalam otomasi adalah membuat proses tampak berhasil ketika asumsi ilmiahnya tidak terpenuhi. Aturan utama yang perlu dipertahankan adalah bahwa tindakan korektif harus menjaga jejak kegagalan untuk audit. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini

juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Untuk tujuan riset, hasil dari panduan qms-e001 sampai qms-e010 sebaiknya ditinjau bersama data sebelum dan sesudah tahap tersebut agar perubahan dapat dilacak. Setelah tahap selesai, periksa apakah error guide benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***16.2.2 Checklist implementasi dan quality control***

- Pastikan parameter input telah ditentukan dan disimpan.
- Pastikan parameter spin telah ditentukan dan disimpan.
- Pastikan parameter 3D telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: tindakan korektif harus menjaga jejak kegagalan untuk audit.
- Pastikan keluaran tahap tersedia sebagai error guide dan status QC tercatat.

## **16.3 Troubleshooting SCF dan optimasi**

High-throughput chemistry menuntut aturan yang konsisten ketika jumlah molekul bertambah. Dalam konteks troubleshooting scf dan optimasi, Diagnosis dimulai dari struktur, charge/spin, basis, functional, konvergensi, dan resource. Elemen yang paling relevan pada tahap ini meliputi max cycle, conv\_tol, geometry dan alternative conformer. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Sandve et al., 2013; Wilkinson et al., 2016; Wilson et al., 2014).

Untuk batch besar, implementasi harus tahan terhadap kegagalan parsial. QMScreen karena itu menyimpan status per molekul dan per tahap, sambil mempertahankan parameter seperti max cycle, 'conv\_tol', 'geometry', alternative conformer pada manifest atau provenance. Keluaran yang diharapkan adalah

catatan percobaan dan hasil. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### ***16.3.1 Prinsip, parameter, dan interpretasi***

Ketika workflow dipercepat, kontrol ilmiah tidak boleh ikut dipangkas. Aturan utama yang perlu dipertahankan adalah bahwa jangan memperlonggar toleransi secara ekstrem hanya untuk memperoleh label converged. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Pada komputer dengan resource terbatas, troubleshooting scf dan optimasi perlu diatur bersama jumlah worker, thread, memory, dan biaya tahap berikutnya. Setelah tahap selesai, periksa apakah catatan percobaan dan hasil benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***16.3.2 Checklist implementasi dan quality control***

- Pastikan parameter max cycle telah ditentukan dan disimpan.
- Pastikan parameter conv\_tol telah ditentukan dan disimpan.
- Pastikan parameter geometry telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: jangan memperlonggar toleransi secara ekstrem hanya untuk memperoleh label converged.
- Pastikan keluaran tahap tersedia sebagai catatan percobaan dan hasil dan status QC tercatat.

## 16.4 Regression testing

Perubahan kode perlu diuji pada dataset kecil yang diketahui agar perubahan orbital, energi, parser, dan ekspor terdeteksi. Dalam workflow komputasi skala besar, konsep ini perlu didefinisikan sebelum job pertama dijalankan karena keputusan awal akan memengaruhi biaya, keterbandingan, dan kemampuan reproduksi hasil. Elemen yang paling relevan pada tahap ini meliputi baseline, hashes, success rate dan deterministic seed where possible. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Sandve et al., 2013; Wilkinson et al., 2016; Wilson et al., 2014).

Di tingkat implementasi, QMScreen mengubah pilihan pengguna menjadi record konfigurasi dan menjalankan fungsi scientific logic yang sesuai. Komponen seperti `baseline`, `hashes`, success rate, deterministic seed where possible tidak diperlakukan sebagai nilai sementara, tetapi sebagai bagian dari state proyek yang dapat ditulis ke disk. Keluaran yang diharapkan adalah laporan compare benchmark. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### 16.4.1 Prinsip, parameter, dan interpretasi

Scientific guardrail pada bagian ini sengaja dibuat konservatif. Aturan utama yang perlu dipertahankan adalah bahwa toleransi regression harus memperhitungkan variasi numerik yang wajar. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Dalam penggunaan sehari-hari, pengguna sebaiknya menguji regression testing pada satu atau dua molekul kecil sebelum memperbesar batch. Setelah tahap selesai, periksa apakah laporan compare benchmark benar-benar tersedia, apakah

unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

#### ***16.4.2 Checklist implementasi dan quality control***

- Pastikan parameter baseline telah ditentukan dan disimpan.
- Pastikan parameter hashes telah ditentukan dan disimpan.
- Pastikan parameter success rate telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: toleransi regression harus memperhitungkan variasi numerik yang wajar.
- Pastikan keluaran tahap tersedia sebagai laporan compare benchmark dan status QC tercatat.

### **16.5 Checklist sebelum publikasi**

Pembahasan checklist sebelum publikasi dimulai dari prinsip bahwa satu angka komputasi hanya bermakna jika struktur, metode, unit, dan status validasinya diketahui. Sebelum hasil dipakai di laporan ilmiah, periksa metode, unit, konvergensi, minimum, provenance, dan arsip. Elemen yang paling relevan pada tahap ini meliputi method/basis, raw output, config, environment dan QC. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Sandve et al., 2013; Wilkinson et al., 2016; Wilson et al., 2014).

Implementasi perangkat lunak untuk bagian ini mengikuti pola parse/validate/execute/record. Input terlebih dahulu dibaca, kemudian diverifikasi, setelah itu engine dijalankan, dan akhirnya keluaran dipaketkan menjadi result record. Parameter yang terkait mencakup `method/basis`, raw output, `config`, `environment`, `QC`. Keluaran yang diharapkan adalah checklist publikasi. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### 16.5.1 Prinsip, parameter, dan interpretasi

Quality control harus membedakan kondisi tidak valid, kondisi meragukan, dan kondisi yang sekadar berbeda dari default. Aturan utama yang perlu dipertahankan adalah bahwa software membantu konsistensi tetapi tanggung jawab ilmiah tetap pada peneliti. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Strategi operasional yang aman untuk checklist sebelum publikasi adalah memulai dari konfigurasi sederhana, memeriksa log, lalu menambah kompleksitas satu parameter pada satu waktu. Setelah tahap selesai, periksa apakah checklist publikasi benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### 16.5.2 Checklist implementasi dan quality control

- Pastikan parameter method/basis telah ditentukan dan disimpan.
- Pastikan parameter raw output telah ditentukan dan disimpan.
- Pastikan parameter config telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: software membantu konsistensi tetapi tanggung jawab ilmiah tetap pada peneliti.
- Pastikan keluaran tahap tersedia sebagai checklist publikasi dan status QC tercatat.

**Tabel 16.2. Checklist hasil sebelum digunakan dalam laporan**

| Pemeriksaan                  | Status yang diharapkan              |
|------------------------------|-------------------------------------|
| Input canonical dan struktur | Tersimpan dan dapat dibaca ulang    |
| Charge/multiplicity          | Konsisten dengan keadaan elektronik |
| SCF                          | Converged                           |

| Pemeriksaan            | Status yang diharapkan               |
|------------------------|--------------------------------------|
| Optimasi               | Converged atau status jelas          |
| Frequency jika dipakai | Minimum check terdokumentasi         |
| Metode/basis           | Terekam dalam config/provenance      |
| Unit                   | Eksplisit                            |
| Raw output             | Tersimpan                            |
| Environment snapshot   | Tersimpan                            |
| Ekspor                 | Berasal dari result record yang sama |

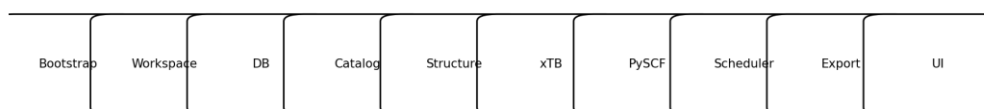
## 16.6 Rangkuman Bab

Bab 16 menunjukkan bahwa validasi, troubleshooting, error code, dan quality assurance harus dipahami sebagai satu sistem yang menghubungkan keputusan ilmiah dan keputusan perangkat lunak. Keandalan QMScreen bergantung pada kemampuan setiap tahap untuk menerima input yang jelas, menjalankan metode dengan parameter eksplisit, mencatat status, mempertahankan hasil parsial secara benar, dan menyediakan provenance. Dengan pola ini, batch yang besar tetap dapat diaudit dan dikembangkan tanpa menjadikan GUI sebagai satu-satunya sumber kebenaran. Pembaca sebaiknya menggunakan checklist pada setiap subbab sebagai kendali sebelum melanjutkan ke tahap berikutnya.

## BAB 17 ARSITEKTUR KODE QMSCREEN V1.0.5 DAN PANDUAN PENGEMBANGAN

Bab ini membahas arsitektur kode qmscreen v1.0.5 dan panduan pengembangan sebagai bagian dari workflow QMScreen V1.0.5. Pembahasan menempatkan konsep kimia komputasi, implementasi perangkat lunak, scientific guardrail, quality control, dan reproducibility dalam satu alur. Setiap subbab diakhiri dengan checklist praktis agar pembaca dapat menghubungkan teori dengan penggunaan aplikasi.

**Peta Modul Kode**



*Gambar 17.1. Peta konseptual arsitektur kode qmscreen v1.0.5 dan panduan pengembangan.*

**Tabel 17.1. Peta materi Bab 17**

| Subbab | Fokus                            | Keluaran utama                        |
|--------|----------------------------------|---------------------------------------|
| 17.1   | Bootstrap dan workspace writable | log bootstrap dan direktori aplikasi  |
| 17.2   | Lapisan utilitas dan database    | API internal data                     |
| 17.3   | Lapisan struktur dan engine      | pipeline single molecule              |
| 17.4   | Scheduler dan state              | background worker dan status          |
| 17.5   | Prinsip pengembangan lanjutan    | roadmap pengembangan yang dapat diuji |

### 17.1 Bootstrap dan workspace writable

Pembahasan bootstrap dan workspace writable dimulai dari prinsip bahwa satu angka komputasi hanya bermakna jika struktur, metode, unit, dan status validasinya diketahui. Kode memulai dengan pemeriksaan dependensi serta pemilihan direktori data yang writable pada sistem operasi berbeda. Elemen yang paling relevan pada tahap ini meliputi `ensure_python_package`, `DATA_DIR`, `RUNS_DIR` dan `CACHE_DIR`. Masing-masing elemen harus memiliki

representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Sandve et al., 2013; Wilkinson et al., 2016; Wilson et al., 2014).

Implementasi perangkat lunak untuk bagian ini mengikuti pola parse/validate/execute/record. Input terlebih dahulu dibaca, kemudian diverifikasi, setelah itu engine dijalankan, dan akhirnya keluaran dipaketkan menjadi result record. Parameter yang terkait mencakup ``ensure_python_package``, ``DATA_DIR``, ``RUNS_DIR``, ``CACHE_DIR``. Keluaran yang diharapkan adalah log bootstrap dan direktori aplikasi. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### ***17.1.1 Prinsip, parameter, dan interpretasi***

Quality control harus membedakan kondisi tidak valid, kondisi meragukan, dan kondisi yang sekadar berbeda dari default. Aturan utama yang perlu dipertahankan adalah bahwa auto-install harus terkendali dan tidak menerima nama paket dari input pengguna. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Strategi operasional yang aman untuk bootstrap dan workspace writable adalah memulai dari konfigurasi sederhana, memeriksa log, lalu menambah kompleksitas satu parameter pada satu waktu. Setelah tahap selesai, periksa apakah log bootstrap dan direktori aplikasi benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***17.1.2 Checklist implementasi dan quality control***

- Pastikan parameter `ensure_python_package` telah ditentukan dan disimpan.
- Pastikan parameter `DATA_DIR` telah ditentukan dan disimpan.
- Pastikan parameter `RUNS_DIR` telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: auto-install harus terkendali dan tidak menerima nama paket dari input pengguna.
- Pastikan keluaran tahap tersedia sebagai log bootstrap dan direktori aplikasi dan status QC tercatat.

## **17.2 Lapisan utilitas dan database**

Pada QMScreen, lapisan utilitas dan database bukan fitur dekoratif antarmuka. Fungsi hash, JSON atomik, environment snapshot, SQLite, audit, dan cache membentuk fondasi data. Pendekatan ini membuat tahap tersebut dapat diuji sebagai komponen ilmiah yang mandiri. Elemen yang paling relevan pada tahap ini meliputi `canonical_json`, `sha256`, `init_db` dan `cache_get/cache_put`. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Sandve et al., 2013; Wilkinson et al., 2016; Wilson et al., 2014).

Secara arsitektural, bagian lapisan utilitas dan database ditempatkan di luar logika tampilan sebanyak mungkin. Data dari widget hanya menjadi sumber konfigurasi, sedangkan fungsi inti bekerja pada struktur Python yang dapat diuji. Titik kendalinya meliputi `'canonical_json'`, `'sha256'`, `'init_db'`, `'cache_get/cache_put'`. Keluaran yang diharapkan adalah API internal data. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### ***17.2.1 Prinsip, parameter, dan interpretasi***

Risiko terbesar dalam otomasi adalah membuat proses tampak berhasil ketika asumsi ilmiahnya tidak terpenuhi. Aturan utama yang perlu dipertahankan adalah bahwa fungsi data sebaiknya tidak bergantung pada widget UI agar dapat diuji terpisah. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan

provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Untuk tujuan riset, hasil dari lapisan utilitas dan database sebaiknya ditinjau bersama data sebelum dan sesudah tahap tersebut agar perubahan dapat dilacak. Setelah tahap selesai, periksa apakah API internal data benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***17.2.2 Checklist implementasi dan quality control***

- Pastikan parameter canonical\_json telah ditentukan dan disimpan.
- Pastikan parameter sha256 telah ditentukan dan disimpan.
- Pastikan parameter init\_db telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: fungsi data sebaiknya tidak bergantung pada widget UI agar dapat diuji terpisah.
- Pastikan keluaran tahap tersedia sebagai API internal data dan status QC tercatat.

## **17.3 Lapisan struktur dan engine**

High-throughput chemistry menuntut aturan yang konsisten ketika jumlah molekul bertambah. Dalam konteks lapisan struktur dan engine, Parser, conformer, xTB, PySCF, frequency, descriptor disusun sebagai fungsi-fungsi terpisah. Elemen yang paling relevan pada tahap ini meliputi parse\_uploaded\_file, generate\_conformers dan run\_dft\_pipeline. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Sandve et al., 2013; Wilkinson et al., 2016; Wilson et al., 2014).

Untuk batch besar, implementasi harus tahan terhadap kegagalan parsial. QMScreen karena itu menyimpan status per molekul dan per tahap, sambil

mempertahankan parameter seperti ``parse_uploaded_file``, ``generate_conformers``, ``run_dft_pipeline`` pada manifest atau provenance. Keluaran yang diharapkan adalah pipeline single molecule. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### ***17.3.1 Prinsip, parameter, dan interpretasi***

Ketika workflow dipercepat, kontrol ilmiah tidak boleh ikut dipangkas. Aturan utama yang perlu dipertahankan adalah bahwa pemisahan tahap memudahkan unit test dan recovery. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Pada komputer dengan resource terbatas, lapisan struktur dan engine perlu diatur bersama jumlah worker, thread, memory, dan biaya tahap berikutnya. Setelah tahap selesai, periksa apakah pipeline single molecule benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***17.3.2 Checklist implementasi dan quality control***

- Pastikan parameter `parse_uploaded_file` telah ditentukan dan disimpan.
- Pastikan parameter `generate_conformers` telah ditentukan dan disimpan.
- Pastikan parameter `run_dft_pipeline` telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: pemisahan tahap memudahkan unit test dan recovery.
- Pastikan keluaran tahap tersedia sebagai pipeline single molecule dan status QC tercatat.

## 17.4 Scheduler dan state

Batch worker, fair queue, status file, dan session state menghubungkan pipeline dengan antarmuka. Dalam workflow komputasi skala besar, konsep ini perlu didefinisikan sebelum job pertama dijalankan karena keputusan awal akan memengaruhi biaya, keterbandingan, dan kemampuan reproduksi hasil. Elemen yang paling relevan pada tahap ini meliputi `run_batch_worker`, `start_batch` dan `request_stop`. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Sandve et al., 2013; Wilkinson et al., 2016; Wilson et al., 2014).

Di tingkat implementasi, QMScreen mengubah pilihan pengguna menjadi record konfigurasi dan menjalankan fungsi scientific logic yang sesuai. Komponen seperti `'run_batch_worker'`, `'start_batch'`, `'request_stop'` tidak diperlakukan sebagai nilai sementara, tetapi sebagai bagian dari state proyek yang dapat ditulis ke disk. Keluaran yang diharapkan adalah background worker dan status. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### 17.4.1 Prinsip, parameter, dan interpretasi

Scientific guardrail pada bagian ini sengaja dibuat konservatif. Aturan utama yang perlu dipertahankan adalah bahwa threading membutuhkan batas state yang jelas agar rerun Streamlit tidak menduplikasi job. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Dalam penggunaan sehari-hari, pengguna sebaiknya menguji scheduler dan state pada satu atau dua molekul kecil sebelum memperbesar batch. Setelah tahap selesai, periksa apakah background worker dan status benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan

dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

#### ***17.4.2 Checklist implementasi dan quality control***

- Pastikan parameter `run_batch_worker` telah ditentukan dan disimpan.
- Pastikan parameter `start_batch` telah ditentukan dan disimpan.
- Pastikan parameter `request_stop` telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: threading membutuhkan batas state yang jelas agar rerun Streamlit tidak menduplikasi job.
- Pastikan keluaran tahap tersedia sebagai background worker dan status dan status QC tercatat.

### **17.5 Prinsip pengembangan lanjutan**

Pembahasan prinsip pengembangan lanjutan dimulai dari prinsip bahwa satu angka komputasi hanya bermakna jika struktur, metode, unit, dan status validasinya diketahui. Pengembangan harus mempertahankan backward compatibility data, provenance, dan scientific guardrail. Elemen yang paling relevan pada tahap ini meliputi schema version, migration, tests dan modular engines. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Sandve et al., 2013; Wilkinson et al., 2016; Wilson et al., 2014).

Implementasi perangkat lunak untuk bagian ini mengikuti pola parse/validate/execute/record. Input terlebih dahulu dibaca, kemudian diverifikasi, setelah itu engine dijalankan, dan akhirnya keluaran dipaketkan menjadi result record. Parameter yang terkait mencakup schema version, 'migration', 'tests', modular engines. Keluaran yang diharapkan adalah roadmap pengembangan yang dapat diuji. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### ***17.5.1 Prinsip, parameter, dan interpretasi***

Quality control harus membedakan kondisi tidak valid, kondisi meragukan, dan kondisi yang sekadar berbeda dari default. Aturan utama yang perlu dipertahankan adalah bahwa fitur baru tidak boleh mematahkan interpretasi hasil lama atau mengubah default diam-diam. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Strategi operasional yang aman untuk prinsip pengembangan lanjutan adalah memulai dari konfigurasi sederhana, memeriksa log, lalu menambah kompleksitas satu parameter pada satu waktu. Setelah tahap selesai, periksa apakah roadmap pengembangan yang dapat diuji benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***17.5.2 Checklist implementasi dan quality control***

- Pastikan parameter schema version telah ditentukan dan disimpan.
- Pastikan parameter migration telah ditentukan dan disimpan.
- Pastikan parameter tests telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: fitur baru tidak boleh mematahkan interpretasi hasil lama atau mengubah default diam-diam.
- Pastikan keluaran tahap tersedia sebagai roadmap pengembangan yang dapat diuji dan status QC tercatat.

**Sketsa implementasi Python - Bab 17 (pseudocode original)**

```
def process_one_molecule(spec, config):
    validate(spec)
    run_stages(config)
    persist_result_and_provenance()
```

## 17.6 Rangkuman Bab

Bab 17 menunjukkan bahwa arsitektur kode qmscreen v1.0.5 dan panduan pengembangan harus dipahami sebagai satu sistem yang menghubungkan keputusan ilmiah dan keputusan perangkat lunak. Keandalan QMScreen bergantung pada kemampuan setiap tahap untuk menerima input yang jelas, menjalankan metode dengan parameter eksplisit, mencatat status, mempertahankan hasil parsial secara benar, dan menyediakan provenance. Dengan pola ini, batch yang besar tetap dapat diaudit dan dikembangkan tanpa menjadikan GUI sebagai satu-satunya sumber kebenaran. Pembaca sebaiknya menggunakan checklist pada setiap subbab sebagai kendali sebelum melanjutkan ke tahap berikutnya.

## BAB 18 EVALUASI PEMBELAJARAN, LATIHAN, DAN PROYEK PENGEMBANGAN

Bab ini membahas evaluasi pembelajaran, latihan, dan proyek pengembangan sebagai bagian dari workflow QMScreen V1.0.5. Pembahasan menempatkan konsep kimia komputasi, implementasi perangkat lunak, scientific guardrail, quality control, dan reproducibility dalam satu alur. Setiap subbab diakhiri dengan checklist praktis agar pembaca dapat menghubungkan teori dengan penggunaan aplikasi.

**Tabel 18.1. Peta materi Bab 18**

| Subbab | Fokus                               | Keluaran utama              |
|--------|-------------------------------------|-----------------------------|
| 18.1   | Tujuan evaluasi                     | kesiapan penggunaan mandiri |
| 18.2   | Latihan desain workflow             | workflow plan               |
| 18.3   | Latihan audit provenance            | audit report                |
| 18.4   | Proyek mini validasi                | mini research report        |
| 18.5   | Proyek pengembangan perangkat lunak | pull-request style package  |

### 18.1 Tujuan evaluasi

Pembahasan tujuan evaluasi dimulai dari prinsip bahwa satu angka komputasi hanya bermakna jika struktur, metode, unit, dan status validasinya diketahui. Evaluasi menguji pemahaman konsep, kemampuan mengoperasikan workflow, membaca status QC, dan menilai batas interpretasi. Elemen yang paling relevan pada tahap ini meliputi konsep, praktik, troubleshooting dan reproducibility. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Sandve et al., 2013; Wilkinson et al., 2016; Wilson et al., 2014).

Implementasi perangkat lunak untuk bagian ini mengikuti pola parse/validate/execute/record. Input terlebih dahulu dibaca, kemudian diverifikasi, setelah itu engine dijalankan, dan akhirnya keluaran dipaketkan menjadi result record. Parameter yang terkait mencakup `konsep`, `praktik`, `troubleshooting`, `reproducibility`. Keluaran yang diharapkan adalah kesiapan penggunaan mandiri. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus

membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### ***18.1.1 Prinsip, parameter, dan interpretasi***

Quality control harus membedakan kondisi tidak valid, kondisi meragukan, dan kondisi yang sekadar berbeda dari default. Aturan utama yang perlu dipertahankan adalah bahwa jawaban yang baik harus menyebut asumsi dan bukan hanya angka atau tombol. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Strategi operasional yang aman untuk tujuan evaluasi adalah memulai dari konfigurasi sederhana, memeriksa log, lalu menambah kompleksitas satu parameter pada satu waktu. Setelah tahap selesai, periksa apakah kesiapan penggunaan mandiri benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***18.1.2 Checklist implementasi dan quality control***

- Pastikan parameter konsep telah ditentukan dan disimpan.
- Pastikan parameter praktik telah ditentukan dan disimpan.
- Pastikan parameter troubleshooting telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: jawaban yang baik harus menyebut asumsi dan bukan hanya angka atau tombol.
- Pastikan keluaran tahap tersedia sebagai kesiapan penggunaan mandiri dan status QC tercatat.

## 18.2 Latihan desain workflow

Pada QMScreen, latihan desain workflow bukan fitur dekoratif antarmuka. Mahasiswa atau pengguna diminta merancang alur untuk kumpulan molekul dengan batas resource tertentu. Pendekatan ini membuat tahap tersebut dapat diuji sebagai komponen ilmiah yang mandiri. Elemen yang paling relevan pada tahap ini meliputi stage selection, conformer, method, basis dan frequency subset. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Sandve et al., 2013; Wilkinson et al., 2016; Wilson et al., 2014).

Secara arsitektural, bagian latihan desain workflow ditempatkan di luar logika tampilan sebanyak mungkin. Data dari widget hanya menjadi sumber konfigurasi, sedangkan fungsi inti bekerja pada struktur Python yang dapat diuji. Titik kendalinya meliputi stage selection, `conformer`, `method`, `basis`, frequency subset. Keluaran yang diharapkan adalah workflow plan. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### 18.2.1 Prinsip, parameter, dan interpretasi

Risiko terbesar dalam otomasi adalah membuat proses tampak berhasil ketika asumsi ilmiahnya tidak terpenuhi. Aturan utama yang perlu dipertahankan adalah bahwa desain harus menjelaskan mengapa setiap tahap dipilih atau dihilangkan. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Untuk tujuan riset, hasil dari latihan desain workflow sebaiknya ditinjau bersama data sebelum dan sesudah tahap tersebut agar perubahan dapat dilacak. Setelah tahap selesai, periksa apakah workflow plan benar-benar tersedia, apakah unit dan

metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***18.2.2 Checklist implementasi dan quality control***

- Pastikan parameter stage selection telah ditentukan dan disimpan.
- Pastikan parameter conformer telah ditentukan dan disimpan.
- Pastikan parameter method telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: desain harus menjelaskan mengapa setiap tahap dipilih atau dihilangkan.
- Pastikan keluaran tahap tersedia sebagai workflow plan dan status QC tercatat.

## **18.3 Latihan audit provenance**

High-throughput chemistry menuntut aturan yang konsisten ketika jumlah molekul bertambah. Dalam konteks latihan audit provenance, Pengguna menelusuri satu hasil dari tabel ekspor kembali ke input manifest, config, environment, job, dan raw output. Elemen yang paling relevan pada tahap ini meliputi spec\_hash, project\_id, timestamps dan database. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Sandve et al., 2013; Wilkinson et al., 2016; Wilson et al., 2014).

Untuk batch besar, implementasi harus tahan terhadap kegagalan parsial. QMScreen karena itu menyimpan status per molekul dan per tahap, sambil mempertahankan parameter seperti `spec\_hash`, `project\_id`, `timestamps`, `database` pada manifest atau provenance. Keluaran yang diharapkan adalah audit report. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### ***18.3.1 Prinsip, parameter, dan interpretasi***

Ketika workflow dipercepat, kontrol ilmiah tidak boleh ikut dipangkas. Aturan utama yang perlu dipertahankan adalah bahwa audit sukses bila hasil dapat direkonstruksi tanpa mengandalkan ingatan pengguna. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Pada komputer dengan resource terbatas, latihan audit provenance perlu diatur bersama jumlah worker, thread, memory, dan biaya tahap berikutnya. Setelah tahap selesai, periksa apakah audit report benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***18.3.2 Checklist implementasi dan quality control***

- Pastikan parameter spec\_hash telah ditentukan dan disimpan.
- Pastikan parameter project\_id telah ditentukan dan disimpan.
- Pastikan parameter timestamps telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: audit sukses bila hasil dapat direkonstruksi tanpa mengandalkan ingatan pengguna.
- Pastikan keluaran tahap tersedia sebagai audit report dan status QC tercatat.

## **18.4 Proyek mini validasi**

Dataset kecil dipakai untuk menguji perubahan functional, basis, atau parameter dengan kontrol yang jelas. Dalam workflow komputasi skala besar, konsep ini perlu didefinisikan sebelum job pertama dijalankan karena keputusan awal akan memengaruhi biaya, keterbandingan, dan kemampuan reproduksi hasil. Elemen yang paling relevan pada tahap ini meliputi baseline, one-factor-at-a-time dan regression. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu

parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Sandve et al., 2013; Wilkinson et al., 2016; Wilson et al., 2014).

Di tingkat implementasi, QMScreen mengubah pilihan pengguna menjadi record konfigurasi dan menjalankan fungsi scientific logic yang sesuai. Komponen seperti `baseline`, `one-factor-at-a-time`, `regression` tidak diperlakukan sebagai nilai sementara, tetapi sebagai bagian dari state proyek yang dapat ditulis ke disk. Keluaran yang diharapkan adalah mini research report. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

#### ***18.4.1 Prinsip, parameter, dan interpretasi***

Scientific guardrail pada bagian ini sengaja dibuat konservatif. Aturan utama yang perlu dipertahankan adalah bahwa perbandingan harus menggunakan geometri dan protokol yang dinyatakan. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Dalam penggunaan sehari-hari, pengguna sebaiknya menguji proyek mini validasi pada satu atau dua molekul kecil sebelum memperbesar batch. Setelah tahap selesai, periksa apakah mini research report benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

#### ***18.4.2 Checklist implementasi dan quality control***

- Pastikan parameter baseline telah ditentukan dan disimpan.
- Pastikan parameter one-factor-at-a-time telah ditentukan dan disimpan.

- Pastikan parameter regression telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: perbandingan harus menggunakan geometri dan protokol yang dinyatakan.
- Pastikan keluaran tahap tersedia sebagai mini research report dan status QC tercatat.

## 18.5 Proyek pengembangan perangkat lunak

Pembahasan proyek pengembangan perangkat lunak dimulai dari prinsip bahwa satu angka komputasi hanya bermakna jika struktur, metode, unit, dan status validasinya diketahui. Fitur baru dapat berupa parser, engine, visualisasi, atau ekspor dengan unit test dan dokumentasi. Elemen yang paling relevan pada tahap ini meliputi modular API, error code, test fixture dan migration. Masing-masing elemen harus memiliki representasi data yang eksplisit sehingga pipeline tidak bergantung pada keadaan visual GUI. Ketika satu parameter berubah, perubahan tersebut harus terlihat pada spesifikasi job dan, bila relevan, pada kunci hash yang membedakan hasil lama dari hasil baru. Prinsip tersebut sejalan dengan praktik kimia komputasi yang menempatkan metode dan asumsi sebagai bagian tak terpisahkan dari hasil (Sandve et al., 2013; Wilkinson et al., 2016; Wilson et al., 2014).

Implementasi perangkat lunak untuk bagian ini mengikuti pola parse/validate/execute/record. Input terlebih dahulu dibaca, kemudian diverifikasi, setelah itu engine dijalankan, dan akhirnya keluaran dipaketkan menjadi result record. Parameter yang terkait mencakup modular API, error code, test fixture, 'migration'. Keluaran yang diharapkan adalah pull-request style package. Record tersebut kemudian dapat dipakai oleh tahap berikutnya tanpa harus membaca kembali keadaan widget. Pemisahan ini penting ketika aplikasi di-resume, ketika hasil dibuka pada sesi lain, atau ketika regression test memeriksa apakah perubahan kode telah mengubah nilai yang semestinya tetap stabil.

### 18.5.1 Prinsip, parameter, dan interpretasi

Quality control harus membedakan kondisi tidak valid, kondisi meragukan, dan kondisi yang sekadar berbeda dari default. Aturan utama yang perlu dipertahankan adalah bahwa fitur diterima hanya bila tidak merusak scientific guardrail dan provenance. Jika prasyarat tidak terpenuhi, software sebaiknya menolak job, memberi warning, atau menandai hasil sebagai belum tervalidasi daripada menebak nilai yang tidak tersedia. QMScreen menggunakan error code, log, status QC, dan provenance untuk memisahkan kegagalan teknis dari keterbatasan interpretasi. Pendekatan ini juga menjaga agar angka yang kebetulan berhasil diparsing tidak otomatis dianggap benar secara ilmiah. Dalam laporan, status validasi harus ditampilkan bersama nilai numerik. Praktik tersebut sangat penting

untuk screening banyak kandidat karena kesalahan kecil yang terulang pada ratusan molekul dapat menciptakan pola semu yang terlihat meyakinkan padahal berasal dari pipeline yang tidak konsisten.

Strategi operasional yang aman untuk proyek pengembangan perangkat lunak adalah memulai dari konfigurasi sederhana, memeriksa log, lalu menambah kompleksitas satu parameter pada satu waktu. Setelah tahap selesai, periksa apakah pull-request style package benar-benar tersedia, apakah unit dan metode tercatat, dan apakah terdapat warning. Jika hasil akan dibandingkan antar kandidat, gunakan protokol yang sama dan jangan mengisi nilai hilang dengan angka buatan. Catatan keputusan, alasan perubahan parameter, serta file ekspor perlu disimpan bersama proyek supaya analisis lanjutan dapat ditelusuri tanpa mengandalkan ingatan operator.

### ***18.5.2 Checklist implementasi dan quality control***

- Pastikan parameter modular API telah ditentukan dan disimpan.
- Pastikan parameter error code telah ditentukan dan disimpan.
- Pastikan parameter test fixture telah ditentukan dan disimpan.
- Pastikan guardrail berikut dipenuhi: fitur diterima hanya bila tidak merusak scientific guardrail dan provenance.
- Pastikan keluaran tahap tersedia sebagai pull-request style package dan status QC tercatat.

## **18.6 Rangkuman Bab**

Bab 18 menunjukkan bahwa evaluasi pembelajaran, latihan, dan proyek pengembangan harus dipahami sebagai satu sistem yang menghubungkan keputusan ilmiah dan keputusan perangkat lunak. Keandalan QMScreen bergantung pada kemampuan setiap tahap untuk menerima input yang jelas, menjalankan metode dengan parameter eksplisit, mencatat status, mempertahankan hasil parsial secara benar, dan menyediakan provenance. Dengan pola ini, batch yang besar tetap dapat diaudit dan dikembangkan tanpa menjadikan GUI sebagai satu-satunya sumber kebenaran. Pembaca sebaiknya menggunakan checklist pada setiap subbab sebagai kendali sebelum melanjutkan ke tahap berikutnya.

## LAMPIRAN 1 20 SOAL EVALUASI

- 1. Tujuan utama QMScreen dalam konteks high-throughput adalah ...**
  - A. Menggantikan seluruh penilaian ilmiah dengan ranking otomatis
  - B. Menata workflow banyak molekul dengan parameter, QC, dan provenance yang konsisten
  - C. Hanya menggambar struktur 2D
  - D. Hanya mengubah SMILES menjadi InChI
  - E. Menyamakan semua metode kuantum
- 2. Mengapa deduplikasi dilakukan sebelum batch berat?**
  - A. Agar semua nama molekul menjadi sama
  - B. Agar energi DFT naik
  - C. Menghindari komputasi berulang pada struktur yang identik dan melaporkan duplikat
  - D. Agar charge selalu nol
  - E. Untuk menghapus provenance
- 3. Kombinasi charge dan multiplicity perlu diperiksa karena ...**
  - A. memengaruhi jumlah elektron dan keadaan spin
  - B. hanya memengaruhi warna UI
  - C. hanya diperlukan untuk CSV
  - D. tidak berkaitan dengan engine kuantum
  - E. selalu harus 0 dan 1
- 4. Peran ETKDGv3 pada QMScreen adalah ...**
  - A. Menghitung DFT final
  - B. Membuat kandidat geometri/conformer 3D
  - C. Menghitung database SQLite
  - D. Mengekspor PDF
  - E. Menghitung ranking akhir
- 5. MMFF94/UFF pada tahap awal terutama digunakan untuk ...**
  - A. Menggantikan semua hasil DFT
  - B. Pre-optimasi conformer dengan biaya rendah
  - C. Menentukan DOI referensi
  - D. Menghitung NUMPAGES
  - E. Membuat hash proyek

- 6. GFN2-xTB digunakan dalam QMScreen terutama sebagai ...**
- A. prescreen yang lebih murah sebelum DFT
  - B. format file
  - C. database relasional
  - D. metode ekspor
  - E. sistem operasi
- 7. Pernyataan yang benar tentang energi total antar molekul dengan formula berbeda adalah ...**
- A. Selalu dapat diranking langsung sebagai afinitas
  - B. QMScreen membatasi normalisasi energi pada kelompok formula yang sama
  - C. Harus diubah menjadi persen tanpa konteks
  - D. Tidak perlu unit
  - E. Sama dengan HOMO-LUMO gap
- 8. Basis optimasi default antarmuka QMScreen adalah ...**
- A. STO-3G
  - B. 6-31G
  - C. def2-SVP
  - D. cc-pVTZ
  - E. def2-QZVP
- 9. Basis final single-point default antarmuka QMScreen adalah ...**
- A. def2-TZVP
  - B. STO-3G
  - C. 3-21G
  - D. 6-31G\*
  - E. aug-cc-pVDZ selalu
- 10. SCF yang tidak konvergen seharusnya ...**
- A. tetap diberi label sukses karena energi terakhir tersedia
  - B. ditandai gagal/belum valid dan ditinjau parameter serta strukturnya
  - C. dihapus tanpa log
  - D. selalu diulang tanpa batas
  - E. dikonversi menjadi dipole
- 11. Minimum lokal dalam pemeriksaan frequency idealnya ...**
- A. memiliki banyak frekuensi imajiner
  - B. tidak memiliki frekuensi imajiner setelah mempertimbangkan toleransi numerik
  - C. hanya ditentukan dari MW
  - D. selalu memiliki gap nol
  - E. tidak memerlukan geometri optimum

**12. HOMO-LUMO gap dihitung sebagai ...**

- A.  $\epsilon_{\text{HOMO}} + \epsilon_{\text{LUMO}}$
- B.  $\epsilon_{\text{LUMO}} - \epsilon_{\text{HOMO}}$
- C. MW/TPSA
- D. charge  $\times$  spin
- E. runtime/worker

**13. Mulliken charge harus ditafsirkan hati-hati karena ...**

- A. tidak memiliki angka
- B. basis-dependent dan bukan satu-satunya definisi muatan parsial
- C. hanya berlaku pada file TXT
- D. tidak berasal dari elektron
- E. selalu identik untuk semua basis

**14. Fair queue terutama bertujuan ...**

- A. menghapus hasil lama
- B. mengatur urutan job agar resource tidak dimonopoli secara tidak adil
- C. menaikkan energi
- D. mengganti functional
- E. menambah stereokimia

**15. Stop aman QMScreen menggunakan gagasan ...**

- A. memutus listrik
- B. stop flag/checkpoint agar penghentian dilakukan pada titik yang lebih aman
- C. menghapus database
- D. mengganti semua hasil menjadi nol
- E. menutup file tanpa menyimpan

**16. Cache yang benar harus dikunci oleh ...**

- A. warna tema UI saja
- B. nama operator saja
- C. hash spesifikasi yang mencakup struktur dan parameter relevan
- D. nomor halaman buku
- E. tanggal tanpa parameter

**17. Environment snapshot berguna untuk ...**

- A. menggantikan hasil energi
- B. mencatat platform dan lingkungan agar reproduksi/diagnosis lebih baik
- C. menghapus log
- D. mengubah SMILES
- E. membuat multiplicity selalu 1

**18. Ranking multikriteria QMScreen sebaiknya dipahami sebagai ...**

- A. ukuran universal aktivitas kimia/biologi
- B. alat bantu prioritisasi berdasarkan descriptor dan bobot yang dipilih
- C. pengganti eksperimen mutlak
- D. metode DFT baru
- E. format file baru

**19. Format yang paling tepat untuk arsip proyek lengkap adalah ...**

- A. screenshot saja
- B. ZIP proyek bersama konfigurasi, hasil, dan log
- C. hanya satu nilai energi
- D. hanya nama molekul
- E. hanya HTML tanpa data

**20. Prinsip terpenting sebelum hasil QMScreen digunakan dalam laporan ilmiah adalah ...**

- A. semua angka harus besar
- B. metode, unit, QC, konvergensi, dan provenance harus diperiksa
- C. hanya ranking yang dibaca
- D. warning harus diabaikan
- E. semua molekul harus mempunyai formula sama

**Tabel 18.2. Kunci jawaban 20 soal evaluasi**

| No. | Jawaban | No. | Jawaban | No. | Jawaban | No. | Jawaban |
|-----|---------|-----|---------|-----|---------|-----|---------|
| 1   | B       | 6   | A       | 11  | B       | 16  | C       |
| 2   | C       | 7   | B       | 12  | B       | 17  | B       |
| 3   | A       | 8   | C       | 13  | B       | 18  | B       |
| 4   | B       | 9   | A       | 14  | B       | 19  | B       |
| 5   | B       | 10  | B       | 15  | B       | 20  | B       |

Pembahasan soal sebaiknya dilakukan dengan kembali ke bab terkait dan menelusuri alasan ilmiah di balik jawaban, bukan sekadar menghafal pilihan. Untuk tugas lanjutan, minta peserta menyertakan file konfigurasi, log, dan provenance ketika menjelaskan sebuah workflow.

## LAMPIRAN 2 TUTORIAL INSTALASI QMSCREEN

### Tutorial Instalasi QMScreen 1.0.5 di WSL/Ubuntu

#### 1. Pendahuluan

**QMScreen V1.0.5** adalah aplikasi berbasis Python dan Streamlit untuk menjalankan workflow kimia komputasi secara terintegrasi, termasuk pengelolaan struktur molekul, pembuatan konformer, prescreening, perhitungan DFT, analisis orbital, frekuensi, termokimia, pengelolaan hasil, serta ekspor data.

Untuk mempermudah instalasi pada Linux Ubuntu maupun **Windows Subsystem for Linux (WSL)**, aplikasi disediakan dalam paket Debian:

```
qmscreen_1.0.5.deb
```

Paket ini dirancang agar aplikasi dapat dipasang menggunakan sistem manajemen paket Ubuntu tanpa harus menjalankan file Python secara manual.

---

#### 2. Persyaratan Sistem

QMScreen dapat digunakan pada:

- Ubuntu Linux;
- Ubuntu pada WSL/WSL2;
- sistem berbasis Debian yang kompatibel;
- Python 3 dengan dukungan virtual environment;
- koneksi internet pada instalasi pertama untuk mengambil library Python yang diperlukan.

Untuk WSL, disarankan menggunakan **WSL2** dengan Ubuntu versi yang masih didukung.

Periksa versi Ubuntu:

```
lsb_release -a
```

Periksa versi Python:

```
python3 --version
```

---

### 3. Menempatkan File Instalasi

Misalnya file:

```
qmscreen_1.0.5.deb
```

tersimpan pada Windows di:

```
C:\Users\USWE\Desktop\RUNNING-KOMPUTASI
```

Pada WSL, folder tersebut biasanya dapat diakses melalui:

```
cd /mnt/c/Users/USWE/Desktop/RUNNING-KOMPUTASI
```

Periksa apakah file tersedia:

```
ls -lh qmscreen_1.0.5.deb
```

Jika nama paket masih menggunakan nama asli:

```
qmscreen_1.0.5-1_all.deb
```

gunakan nama tersebut dalam perintah instalasi.

---

### 4. Memperbarui Database Paket Ubuntu

Sebelum melakukan instalasi, jalankan:

```
sudo apt update
```

Perintah ini memperbarui daftar paket dari repositori Ubuntu.

Tidak perlu menjalankan `sudo apt upgrade` hanya untuk memasang QMScreen.

---

### 5. Instalasi QMScreen

Jika nama file adalah:

```
qmscreen_1.0.5.deb
```

jalankan:

```
sudo apt install ./qmscreen_1.0.5.deb
```

Jika menggunakan nama paket hasil build:

```
qmscreen_1.0.5-1_all.deb
```

jalankan:

```
sudo apt install ./qmscreen_1.0.5-1_all.deb
```

Perhatikan bahwa tanda:

```
./
```

di depan nama file penting karena menunjukkan bahwa paket DEB berasal dari direktori aktif.

---

## 6. Menjalankan QMScreen

Setelah instalasi selesai, aplikasi dapat dijalankan dari terminal:

```
qmscreen
```

Pada proses pertama, aplikasi melakukan persiapan environment QMScreen.

Secara umum QMScreen akan:

1. membuat virtual environment khusus;
  2. memasang library Python yang diperlukan;
  3. menyiapkan direktori data;
  4. menyiapkan database molekul;
  5. menjalankan server Streamlit;
  6. menyediakan antarmuka QMScreen melalui browser.
- 

## 7. Virtual Environment QMScreen

QMScreen tidak bergantung pada instalasi package Python global milik Ubuntu.

Environment aplikasi dibuat pada:

```
~/local/share/qmscreen/venv
```

Pendekatan ini menghindari konflik dengan:

- Python bawaan Ubuntu;
- environment Conda;

- Miniforge;
- Anaconda;
- paket Python sistem;
- aturan `externally-managed-environment` atau PEP 668.

Dengan demikian library QMScreen terisolasi dari package Python lain pada sistem.

## 8. Library yang Dipasang

Pada persiapan environment, QMScreen akan memasang library yang diperlukan aplikasi, antara lain:

```
Streamlit
NumPy
Pandas
SciPy
RDKit
PySCF
geomeTRIC
Plotly
psutil
openpyxl
XlsxWriter
ReportLab
py3Dmol
```

QMScreen juga mencoba menyiapkan backend:

```
TBLLite / GFN2-xTB
```

jika package tersebut tersedia dan kompatibel dengan platform yang digunakan.

Karena proses ini dilakukan pada instalasi atau running awal, waktu yang dibutuhkan bergantung pada:

- kecepatan internet;
- kecepatan CPU;
- media penyimpanan;
- versi Ubuntu;
- versi Python.

## 9. Membuka QMScreen di Browser

Setelah perintah:

```
qmscreen
```

berhasil, Streamlit biasanya dijalankan pada:

```
http://localhost:8501
```

Buka browser Windows, misalnya:

- Google Chrome;
- Microsoft Edge;
- Firefox.

Kemudian akses:

```
http://localhost:8501
```

Pada WSL2 modern, `localhost` Windows biasanya dapat langsung mengakses aplikasi yang berjalan di WSL.

---

## 10. Menjalankan dengan Port Berbeda

Jika port:

```
8501
```

sudah digunakan program lain, QMScreen dapat dijalankan menggunakan port alternatif.

Contoh:

```
QMSCREEN_PORT=8502 qmscreen
```

Kemudian buka:

```
http://localhost:8502
```

Contoh lain:

```
QMSCREEN_PORT=8503 qmscreen
```

dan akses:

```
http://localhost:8503
```

---

## 11. Perintah Penting QMScreen

Setelah paket terinstal, tersedia beberapa perintah utama.

### ***Menjalankan aplikasi***

```
qmscreen
```

Digunakan untuk menjalankan aplikasi QMScreen.

### ***Menyiapkan ulang dependency***

```
qmscreen-setup
```

Perintah ini berguna untuk menjalankan proses setup environment dan dependency secara manual.

Gunakan jika:

- dependency belum selesai terpasang;
- instalasi awal terputus;
- koneksi internet sempat gagal;
- package tertentu perlu disiapkan kembali.

### ***Reset virtual environment***

```
qmscreen-reset-env
```

Perintah tersebut menghapus environment Python khusus QMScreen sehingga dapat dibuat ulang.

Setelah reset, jalankan kembali:

```
qmscreen
```

QMScreen kemudian membangun environment baru dan memasang dependency yang dibutuhkan.

## 12. Lokasi Data QMScreen

Data pengguna ditempatkan pada:

```
~/.local/share/qmscreen
```

Direktori tersebut dapat berisi antara lain:

```

venv/
runs/
cache/
qmscreen.sqlite3
upload_molekul.json

```

Tergantung perkembangan aplikasi, isi sebenarnya dapat bertambah.

### ***Fungsi direktori***

venv/ menyimpan environment dan library Python QMScreen.

runs/ menyimpan workspace atau hasil proses komputasi.

cache/ digunakan untuk cache hasil perhitungan.

qmscreen.sqlite3 digunakan sebagai database internal QMScreen.

upload\_molekul.json merupakan penyimpanan katalog/input molekul yang telah ditambahkan.

## 13. Database Molekul

QMScreen mengintegrasikan:

```
upload_molekul.json
```

sebagai salah satu tempat penyimpanan data input molekul.

Data molekul yang tersimpan dapat mencakup informasi seperti:

```

Nama molekul
SMILES
InChI
InChIKey
Formula molekul
Muatan
Spin/multiplicity
MolBlock
Data struktur 3D
Informasi sumber
Waktu penyimpanan

```

Dengan mekanisme tersebut molekul yang sebelumnya telah dimasukkan dapat digunakan kembali tanpa harus selalu diketik ulang.

## 14. Contoh Proses Instalasi Lengkap

Misalnya file berada di:

```
/mnt/c/Users/USWE/Desktop/RUNNING-KOMPUTASI/
```

jalankan:

```
cd /mnt/c/Users/USWE/Desktop/RUNNING-KOMPUTASI/
```

Periksa file:

```
ls -lh qmscreen_1.0.5.deb
```

Update repositori:

```
sudo apt update
```

Install:

```
sudo apt install ./qmscreen_1.0.5.deb
```

Setelah berhasil:

```
qmscreen
```

Tunggu sampai Streamlit menampilkan alamat aplikasi, kemudian buka:

```
http://localhost:8501
```

---

## 15. Pemeriksaan Instalasi

Untuk memastikan perintah QMScreen tersedia:

```
which qmscreen
```

Jika terinstal dengan benar, biasanya akan ditampilkan lokasi executable, misalnya:

```
/usr/bin/qmscreen
```

Periksa informasi package:

```
dpkg -l | grep qmscreen
```

atau:

```
dpkg -s qmscreen
```

## 16. Jika Perintah `qmscreen` Tidak Ditemukan

Periksa apakah package benar-benar terpasang:

```
dpkg -l | grep qmscreen
```

Jika belum, ulangi:

```
sudo apt install ./qmscreen_1.0.5.deb
```

Kemudian:

```
hash -r
```

dan jalankan kembali:

```
qmscreen
```

---

## 17. Jika Dependency Python Bermasalah

Jika aplikasi gagal karena environment Python rusak atau instalasi library tidak lengkap, gunakan:

```
qmscreen-reset-env
```

Kemudian:

```
qmscreen
```

Environment akan disiapkan ulang.

Alternatifnya jalankan:

```
qmscreen-setup
```

untuk menjalankan proses setup dependency secara manual.

---

## 18. Jika Browser Tidak Dapat Mengakses Port 8501

Pastikan aplikasi masih berjalan di terminal.

Periksa:

```
ss -ltnp | grep 8501
```

Jika QMScreen dijalankan pada port lain:

```
QMSCREEN_PORT=8502 qmscreen
```

buka:

```
http://localhost:8502
```

Jangan menutup terminal yang sedang menjalankan QMScreen karena proses Streamlit akan ikut berhenti.

---

## 19. Menghentikan QMScreen

Jika aplikasi sedang berjalan dari terminal, tekan:

```
Ctrl + C
```

Ini akan menghentikan server QMScreen.

Data yang telah disimpan pada direktori pengguna tetap tersedia.

---

## 20. Menjalankan Kembali

Untuk penggunaan berikutnya tidak perlu menginstal ulang package DEB.

Cukup jalankan:

```
qmscreen
```

Kemudian buka:

```
http://localhost:8501
```

Environment yang sebelumnya telah dibuat akan digunakan kembali.

---

## 21. Update Versi QMScreen

Jika tersedia versi DEB yang lebih baru, misalnya:

```
qmscreen_1.0.6.deb
```

dapat dipasang dengan:

```
sudo apt install ./qmscreen_1.0.6.deb
```

APT akan menangani upgrade package apabila metadata versi paket memang lebih baru.

Data pengguna sebaiknya tetap ditempatkan di:

```
~/.local/share/qmscreen
```

sehingga terpisah dari file program yang dipasang oleh paket Debian.

---

## 22. Uninstall QMScreen

Untuk menghapus program:

```
sudo apt remove qmscreen
```

Jika ingin menghapus package beserta konfigurasi paket:

```
sudo apt purge qmscreen
```

Namun data pengguna dalam:

```
~/.local/share/qmscreen
```

sengaja tidak perlu langsung dihapus karena dapat berisi:

- hasil perhitungan;
  - database;
  - cache;
  - katalog molekul;
  - upload\_molekul.json.
- 

## 23. Menghapus Seluruh Data QMScreen

Hanya lakukan langkah berikut apabila memang ingin menghapus seluruh data aplikasi:

```
rm -rf ~/.local/share/qmscreen
```

**Perhatian:** perintah tersebut dapat menghapus database molekul, environment, cache, dan hasil komputasi yang tersimpan di direktori tersebut.

Karena itu sebaiknya lakukan backup lebih dahulu, misalnya:

```
cp -a ~/.local/share/qmscreen ~/qmscreen_backup
```

---

## 24. Instalasi Ulang Bersih

Jika dibutuhkan instalasi benar-benar dari awal:

```
sudo apt purge qmscreen
```

Kemudian hapus environment/data bila memang tidak diperlukan:

```
rm -rf ~/.local/share/qmscreen
```

Install kembali:

```
sudo apt install ./qmscreen_1.0.5.deb
```

Lalu:

```
qmscreen
```

---

## 25. Ringkasan Perintah

| Keperluan          | Perintah                                                     |
|--------------------|--------------------------------------------------------------|
| Install            | <code>sudo apt install ./qmscreen_1.0.5.deb</code>           |
| Jalankan           | <code>qmscreen</code>                                        |
| Setup dependency   | <code>qmscreen-setup</code>                                  |
| Reset environment  | <code>qmscreen-reset-env</code>                              |
| Port alternatif    | <code>QMSCREEN_PORT=8502 qmscreen</code>                     |
| Cek package        | <code>dpkg -s qmscreen</code>                                |
| Cek executable     | <code>which qmscreen</code>                                  |
| Stop aplikasi      | <code>Ctrl + C</code>                                        |
| Uninstall          | <code>sudo apt remove qmscreen</code>                        |
| Purge package      | <code>sudo apt purge qmscreen</code>                         |
| Backup data        | <code>cp -a ~/.local/share/qmscreen ~/qmscreen_backup</code> |
| Hapus seluruh data | <code>rm -rf ~/.local/share/qmscreen</code>                  |

## Penutup

Dengan paket `qmscreen_1.0.5.deb`, QMScreen dapat dipasang sebagai aplikasi Ubuntu/WSL menggunakan mekanisme paket Debian. Setelah instalasi, pengguna cukup menjalankan:

```
qmscreen
```

dan mengakses aplikasi melalui:

```
http://localhost:8501
```

Pemisahan antara **file aplikasi**, **virtual environment**, dan **data pengguna** membuat instalasi lebih terstruktur dan mengurangi konflik dengan Python sistem maupun environment kimia komputasi lain yang sudah ada.

## GLOSARIUM

Glosarium berikut mencakup istilah utama dari A sampai Z. Huruf abjad menggunakan style non-heading sehingga tidak masuk ke Daftar Isi; hanya judul GLOSARIUM yang menggunakan Heading 1.

### A

**Afinitas elektron:** Perubahan energi yang terkait penerimaan elektron; tidak boleh disamakan dengan energi total mentah lintas molekul.

**Audit trail:** Jejak kronologis keputusan, status, dan perubahan yang dapat ditelusuri.

**Atom count:** Jumlah atom pada struktur kerja yang digunakan untuk QC dan estimasi resource.

### B

**Basis set:** Kumpulan fungsi basis untuk merepresentasikan orbital dalam perhitungan elektronik.

**Benchmark:** Pengukuran kinerja atau perbandingan terhadap baseline untuk mendeteksi perubahan.

**B3LYP:** Functional hybrid DFT yang tersedia pada antarmuka QMScreen.

### C

**Cache:** Penyimpanan hasil tervalidasi agar job dengan spesifikasi identik tidak dihitung ulang.

**Canonical SMILES:** Representasi SMILES yang dinormalisasi untuk membantu identifikasi struktur.

**Charge:** Muatan total molekul yang memengaruhi jumlah elektron dalam kalkulasi.

**Conformer:** Geometri tiga dimensi berbeda dari molekul dengan konektivitas sama.

### D

**Deduplication:** Proses mengenali input yang mewakili struktur dan keadaan elektronik yang sama.

**Density fitting:** Pendekatan aproksimasi integral yang dapat mempercepat bagian kalkulasi tertentu.

**DFT:** Density Functional Theory, keluarga metode struktur elektronik berbasis densitas.

**Dipole moment:** Besaran vektor yang menggambarkan pemisahan muatan total pada molekul.

## E

**Electronic energy:** Energi elektronik dari metode kuantum pada geometri tertentu.

**Environment snapshot:** Rekaman platform dan lingkungan perangkat lunak untuk reproduksibilitas.

**ETKDGv3:** Metode embedding conformer pada RDKit berbasis distance geometry dan pengetahuan torsional.

**Error code:** Kode terstruktur seperti QMS-E005 untuk mengklasifikasikan kegagalan.

## F

**Fair queue:** Kebijakan penjadwalan yang mengurangi risiko starvation antar job/prioritas.

**Frequency:** Frekuensi vibrasi hasil analisis Hessian.

**Functional:** Bentuk aproksimasi exchange-correlation dalam DFT.

## G

**GFN2-xTB:** Metode extended tight-binding yang digunakan sebagai prescreen cepat.

**geomeTRIC:** Perangkat optimasi geometri yang dapat terintegrasi dengan PySCF.

**Gradient:** Turunan energi terhadap koordinat; digunakan optimizer untuk menentukan arah langkah.

## H

**Hash:** Digest seperti SHA-256 yang digunakan untuk identitas spesifikasi atau data.

**Hessian:** Matriks turunan kedua energi yang dipakai pada analisis vibrasi.

**HOMO:** Highest Occupied Molecular Orbital, orbital terisi dengan energi tertinggi dalam representasi tertentu.

## I

**Imaginary frequency:** Mode vibrasi dengan tanda frekuensi negatif dalam konvensi pelaporan; dapat menandakan bukan minimum lokal.

**InChI:** International Chemical Identifier untuk merepresentasikan struktur molekul.

**InChIKey:** Hash ringkas dari InChI untuk membantu identifikasi dan pencarian.

## J

**Job:** Satuan pekerjaan komputasi untuk satu molekul/spesifikasi.

**Job specification:** Record parameter eksplisit yang mendefinisikan bagaimana job harus dijalankan.

**JSON:** Format data teks terstruktur yang digunakan untuk konfigurasi, manifest, dan hasil.

## K

**Katalog molekul:** Penyimpanan persisten input pengguna pada `upload_molekul.json`.

**Konvergensi:** Kondisi ketika iterasi numerik memenuhi kriteria penghentian yang ditetapkan.

**Kuantum descriptor:** Besaran turunan perhitungan elektronik seperti energi orbital atau dipole.

## L

**Log:** Catatan urutan peristiwa dan pesan selama eksekusi.

**LUMO:** Lowest Unoccupied Molecular Orbital, orbital tak terisi dengan energi terendah dalam representasi tertentu.

**Low-energy conformer:** Conformer yang berada pada bagian bawah distribusi energi tahap screening.

## M

**Manifest:** Daftar input dan metadata yang disimpan untuk satu proyek.

**Memory per job:** Batas memori yang dialokasikan/diberitahukan untuk satu job.

**MMFF94:** Force field yang dapat dipakai untuk pre-optimasi conformer.

**Multiplicity:**  $2S+1$ , ukuran keadaan spin total.

**Mulliken charge:** Muatan parsial dari analisis populasi Mulliken.

## N

**Normalisasi:** Proses menyeragamkan representasi data sebelum perbandingan atau hashing.

**Normal mode:** Pola gerak vibrasi kolektif yang berasal dari diagonalization Hessian.

**NUMPAGES:** Field Word yang menampilkan jumlah total halaman dokumen.

## O

**Optimization:** Proses numerik mencari geometri yang memenuhi kriteria stasioner/konvergensi.

**Orbital:** Fungsi satu-elektron dalam representasi metode tertentu.

**Output mentah:** File atau teks backend sebelum diringkas menjadi result record.

## P

**Parallel worker:** Worker yang memproses job secara bersamaan.

**PBE0:** Functional hybrid DFT yang tersedia pada QMScreen.

**Prescreen:** Tahap murah untuk menyaring kandidat sebelum tahap mahal.

**Provenance:** Metadata asal-usul data, parameter, proses, dan lingkungan.

**PySCF:** Framework Python untuk simulasi struktur elektronik yang menjadi engine DFT QMScreen.

## Q

**QC:** Quality control; pemeriksaan validitas input, proses, dan hasil.

**QMS-E001–QMS-E010:** Keluarga kode error terstruktur pada QMScreen.

**Queue:** Antrean job yang menunggu resource dan worker.

## R

**Ranking multikriteria:** Pengurutan kandidat berdasarkan gabungan descriptor dan bobot.

**Regression baseline:** Rekaman metrik referensi untuk mendeteksi perubahan perilaku setelah kode berubah.

**Resume:** Melanjutkan kalkulasi dari checkpoint atau hasil tahap yang sudah valid.

**RDKit:** Toolkit cheminformatics untuk parsing, 2D/3D, conformer, dan descriptor.

## S

**SCF:** Self-consistent field, prosedur iteratif untuk memperoleh solusi elektronik konsisten.

**SMILES:** Notasi teks untuk struktur molekul.

**Spec hash:** Hash dari spesifikasi kanonik yang dipakai untuk cache/provenance.

**SQLite:** Database relasional embedded yang menyimpan project, job, result, cache, dan audit.

## T

**TBLite:** Library Python yang menyediakan kalkulator GFN-xTB.

**Thermochemistry:** Besaran termokimia yang diturunkan dari energi dan analisis vibrasi pada kondisi tertentu.

**Thread:** Unit eksekusi yang digunakan dalam konfigurasi paralel dan backend.

**TPSA:** Topological polar surface area, descriptor molekul 2D.

## U

**UFF:** Universal Force Field, alternatif pre-optimasi ketika parameter MMFF94 tidak tersedia.

**Unit:** Konvensi besaran fisik yang wajib dicatat untuk interpretasi.

**Unique input:** Input yang tersisa setelah deduplikasi.

## V

**Validasi:** Proses memastikan input, proses, atau hasil memenuhi kriteria yang dinyatakan.

**Versioning:** Pencatatan versi aplikasi, schema, dan dependency untuk menjaga kompatibilitas.

**Visualization:** Representasi 2D/3D atau tabel yang membantu inspeksi tanpa menggantikan raw data.

## W

**WB97X:** Functional range-separated yang tersedia pada pilihan DFT QMScreen.

**Worker pool:** Kumpulan worker yang menjalankan beberapa job secara paralel.

**Workflow:** Urutan tahap komputasi dari input sampai hasil dan ekspor.

## X

**XLSX:** Format spreadsheet untuk ekspor tabel hasil.

**xTB:** Keluarga metode extended tight-binding dan program yang dapat menjadi backend prescreen.

## Y

**YAML:** Format konfigurasi terstruktur yang umum pada software ilmiah; bukan format utama QMScreen V1.0.5 tetapi relevan untuk interoperabilitas konfigurasi.

**Yield komputasi:** Proporsi job yang mencapai status hasil tertentu; harus didefinisikan bersama kriteria sukses/QC.

**Z**

**Zero-point energy (ZPE):** Energi titik nol vibrasi yang dapat muncul pada analisis termokimia harmonik.

**ZIP proyek:** Arsip terkompresi yang mengumpulkan file proyek untuk penyimpanan dan pemindahan.

## DAFTAR PUSTAKA

Daftar pustaka memprioritaskan sumber yang relevan langsung dengan konsep dan perangkat yang digunakan QMScreen. Dokumentasi perangkat lunak perlu diperiksa kembali sesuai versi lingkungan yang benar-benar dipakai pada penelitian.

- Adamo, C., & Barone, V. (1999). Toward reliable density functional methods without adjustable parameters: The PBE0 model. *Journal of Chemical Physics*, 110, 6158-6170. <https://doi.org/10.1063/1.478522>
- Bannwarth, C., Ehlert, S., & Grimme, S. (2019). GFN2-xTB—An accurate and broadly parametrized self-consistent tight-binding quantum chemical method with multipole electrostatics and density-dependent dispersion contributions. *Journal of Chemical Theory and Computation*, 15, 1652-1671. <https://doi.org/10.1021/acs.jctc.8b01176>
- Becke, A. D. (1993). Density-functional thermochemistry. III. The role of exact exchange. *Journal of Chemical Physics*, 98, 5648-5652.
- Cramer, C. J. (2004). *Essentials of Computational Chemistry: Theories and Models* (2nd ed.). Wiley.
- Grimme, S., Antony, J., Ehrlich, S., & Krieg, H. (2010). A consistent and accurate ab initio parametrization of density functional dispersion correction (DFT-D) for the 94 elements H-Pu. *Journal of Chemical Physics*, 132, 154104.
- Helgaker, T., Jørgensen, P., & Olsen, J. (2000). *Molecular Electronic-Structure Theory*. Wiley.
- Hirshfeld, F. L. (1977). Bonded-atom fragments for describing molecular charge densities. *Theoretica Chimica Acta*, 44, 129-138.
- Jensen, F. (2017). *Introduction to Computational Chemistry* (3rd ed.). Wiley.
- Koch, W., & Holthausen, M. C. (2001). *A Chemist's Guide to Density Functional Theory* (2nd ed.). Wiley-VCH.
- Leach, A. R. (2001). *Molecular Modelling: Principles and Applications* (2nd ed.). Pearson.
- Lee, C., Yang, W., & Parr, R. G. (1988). Development of the Colle-Salvetti correlation-energy formula into a functional of the electron density. *Physical Review B*, 37, 785-789. <https://doi.org/10.1103/PhysRevB.37.785>
- Mulliken, R. S. (1955). Electronic population analysis on LCAO-MO molecular wave functions. *Journal of Chemical Physics*, 23, 1833-1840.
- Parr, R. G., & Yang, W. (1989). *Density-Functional Theory of Atoms and Molecules*. Oxford University Press.

- Perdew, J. P., Burke, K., & Ernzerhof, M. (1996). Generalized gradient approximation made simple. *Physical Review Letters*, 77, 3865-3868.
- Pracht, P., Bohle, F., & Grimme, S. (2020). Automated exploration of the low-energy chemical space with fast quantum chemical methods. *Physical Chemistry Chemical Physics*, 22, 7169-7192.
- Sandve, G. K., Nekrutenko, A., Taylor, J., & Hovig, E. (2013). Ten simple rules for reproducible computational research. *PLOS Computational Biology*, 9, e1003285.
- Schlegel, H. B. (1982). Optimization of equilibrium geometries and transition structures. *Journal of Computational Chemistry*, 3, 214-218.
- Stephens, P. J., Devlin, F. J., Chabalowski, C. F., & Frisch, M. J. (1994). Ab initio calculation of vibrational absorption and circular dichroism spectra using density functional force fields. *Journal of Physical Chemistry*, 98, 11623-11627.
- Sun, Q., Berkelbach, T. C., Blunt, N. S., Booth, G. H., Guo, S., Li, Z., Liu, J., McClain, J., 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, e1340. <https://doi.org/10.1002/wcms.1340>
- Sun, Q., Zhang, X., Banerjee, S., Bao, P., Barbry, M., Blunt, N. S., et al. (2020). Recent developments in the PySCF program package. *Journal of Chemical Physics*, 153, 024109.
- Szabo, A., & Ostlund, N. S. (1996). *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory*. Dover.
- Todeschini, R., & Consonni, V. (2009). *Molecular Descriptors for Chemoinformatics*. Wiley-VCH.
- Wang, L.-P., Song, C., & Martínez, T. J. (2016). Developing a robust and efficient algorithm for computational chemistry geometry optimization. *Journal of Chemical Physics*, 144, 214108. <https://doi.org/10.1063/1.4952956>
- Weigend, F., & Ahlrichs, R. (2005). Balanced basis sets of split valence, triple zeta valence and quadruple zeta valence quality for H to Rn. *Physical Chemistry Chemical Physics*, 7, 3297-3305.
- Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., et al. (2016). The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*, 3, 160018.
- Wilson, G., Aruliah, D. A., Brown, C. T., et al. (2014). Best practices for scientific computing. *PLOS Biology*, 12, e1001745.

- Zhao, Y., & Truhlar, D. G. (2008). The M06 suite of density functionals for main group thermochemistry, thermochemical kinetics, noncovalent interactions, excited states, and transition elements. *Theoretical Chemistry Accounts*, 120, 215-241. <https://doi.org/10.1007/s00214-007-0310-x>
- RDKit Documentation. (2026). The RDKit Documentation, release 2026.03.x. <https://www.rdkit.org/docs/> (diakses 22 Agustus 2026).
- PySCF Developers. (2026). PySCF User Guide and API Documentation. <https://pyscf.org/> (diakses 22 Agustus 2026).
- geomeTRIC Developers. (2026). geomeTRIC geometry optimization documentation and source repository. <https://geometric.readthedocs.io/> dan <https://github.com/leeping/geomeTRIC> (diakses 22 Agustus 2026).
- tblite Developers. (2026). tblite Python API and GFN2-xTB documentation. <https://tblite.readthedocs.io/> (diakses 22 Agustus 2026).
- Streamlit. (2026). Streamlit Documentation and API Reference. <https://docs.streamlit.io/> (diakses 22 Agustus 2026).
- SQLite Consortium. (2026). SQLite Documentation. <https://www.sqlite.org/docs.html> (diakses 22 Agustus 2026).
- NumPy Developers. (2026). NumPy Documentation. <https://numpy.org/doc/> (diakses 22 Agustus 2026).
- pandas Developers. (2026). pandas Documentation. <https://pandas.pydata.org/docs/> (diakses 22 Agustus 2026).
- SciPy Community. (2026). SciPy Documentation. <https://docs.scipy.org/doc/scipy/> (diakses 22 Agustus 2026).
- Python Software Foundation. (2026). Python 3 Documentation. <https://docs.python.org/3/> (diakses 22 Agustus 2026).

**- KASMUI -**  
Dosen Kimia, Komputasi, IT, AI UNNES

<https://hisabmu.com/>  
<https://kasmui.cloud/>

|                           |                                                                   |                                                        |                        |
|---------------------------|-------------------------------------------------------------------|--------------------------------------------------------|------------------------|
| Ketua PCM<br>Gunungpati 2 | Anggota Majelis Tabligh<br>PDM Kota Semarang<br>& PWM Jawa Tengah | Tim Pengembang<br>Software KHGT<br>MTT PP Muhammadiyah | Praktisi<br>Ilmu Falak |
|---------------------------|-------------------------------------------------------------------|--------------------------------------------------------|------------------------|