

Silabus Perkuliahan: Python untuk Astronomi Islam

```
import skyfield
import skyfield apss ai

calculate_kiblat(latitude, longitude)
calculate_kiblat(latitude, longitude)
...
```

```
def find_ijtimak():
    return moon_phase()
def find_ijtimak():
    ..
    moon_phase = ai.calculate(date)
```

```
1 import skyfield
2
3 def           (          ):
4               
5     calculate_kiblat(latitude, longitude)
6     calculate_kiblat(latitude, longitude)
7
8 def find_ijtimak():
9     moon_phase = ai.calculate(date)
10
11
```


KATA PENGANTAR (IFTITAH)

Revolusi Komputasi dalam Epistemologi Ilmu Falak Kontemporer

Bismillaahirrahmaanirrahiim.

Puji syukur sedalam-dalamnya kita panjatkan ke hadirat Allah Subhanahu wa Ta'ala, Dzat yang menciptakan kosmos dengan arsitektur matematis yang maha presisi. Sebagaimana termaktub dalam firman-Nya, "Matahari dan bulan beredar menurut perhitungan" (QS. Ar-Rahman [55]: 5). Ayat ini bukan sekadar deskripsi fenomena alam, melainkan sebuah epistemologi dasar yang memerintahkan umat manusia untuk melakukan observasi dan kuantifikasi (hisab) terhadap dinamika benda-benda langit. Shalawat serta salam senantiasa tercurah kepada baginda Nabi Muhammad Shallallahu 'alaihi wa sallam, pionir peradaban yang meletakkan dasar-dasar integrasi antara wahyu dan akal budi.

Urgensi Disiplin

Astronomi Islam (Ilmu Falak) menempati posisi yang sangat unik dan fundamental dalam lanskap keilmuan Islam. Ia bukan sekadar disiplin sains teoretis, melainkan prasyarat mutlak (syarat sah) bagi berdirinya pilar-pilar ibadah vertikal umat Islam. Penentuan awal waktu salat, akurasi arah kiblat di seluruh permukaan bumi, hingga penetapan awal bulan kamariah (khususnya untuk ibadah puasa dan haji) sangat bergantung pada observasi (rukyat) dan komputasi (hisab) astronomis. Di era modern, tuntutan akan akurasi perhitungan ini semakin krusial dan tidak bisa ditawar. Kesalahan sekian derajat dalam perhitungan deklinasi matahari atau *equation of time* tidak hanya bermakna kesalahan matematis, tetapi berimplikasi langsung pada validitas ibadah jutaan umat. Terlebih, hadirnya gagasan dan implementasi Kalender Hijriah Global Tunggal (KHGT) merupakan sebuah tonggak sejarah penting dalam mewujudkan sistem penjadwalan waktu Islam yang unifikatif dan saintifik, yang pada praktiknya sangat membutuhkan dukungan instrumen komputasi yang tangguh, akurat, dan mudah diakses.

Gap Riset dan Problematika Klasik

Secara historis, peradaban Islam telah melahirkan para astronom raksasa yang mewariskan tabel astronomi (*zīj*) dengan akurasi yang melampaui zamannya. Namun, dalam konteks kekinian, kita menghadapi kesenjangan (*research gap*) metodologis yang tajam.

- Pertama, literatur dan praktik hisab tradisional seringkali masih mengandalkan kalkulasi manual atau menggunakan perangkat lunak usang yang memakan waktu dan rentan terhadap bias *human error*.
- Kedua, ekosistem teknologi saat ini dibanjiri oleh aplikasi jadwal salat dan penunjuk kiblat yang bersifat *black-box*. Pengguna dan bahkan para peneliti falak hanya menerima output akhir tanpa memiliki akses, kendali, atau kemampuan untuk memvalidasi algoritma dan parameter astronomis di baliknya.

Terjadi segregasi intelektual yang nyata: para ahli syariah dan falak seringkali tidak menguasai instrumen komputasi modern, sementara para ahli *computer science* kerap kali tuna terhadap landasan fikih dan kompleksitas rumus astronomi sferis.

Kebaruan Gagasan (Novelty)

Buku referensi "Silabus Perkuliahan: Python untuk Astronomi Islam" ini disusun secara khusus untuk mendekonstruksi kesenjangan tersebut. Kebaruan gagasan (*novelty*) dari karya ini terletak pada integrasi penuh antara keilmuan falak murni dengan Python—bahasa pemrograman tingkat tinggi yang paling revolusioner, tangguh, dan sangat adaptif di era *data science* saat ini. Buku ini menolak pendekatan komputasi yang pasif. Sebaliknya, karya ini menawarkan paradigma baru dengan membongkar kotak hitam komputasi falak.

Melalui buku ini, pembaca tidak hanya disajikan rumusan trigonometri bola yang rumit, tetapi dituntun secara interaktif untuk membangun algoritma sendiri. Dengan memanfaatkan pustaka mutakhir seperti Skyfield untuk perhitungan efemeris presisi tinggi, serta alat visualisasi Matplotlib, buku ini mengubah angka-angka mentah menjadi representasi visual yang komprehensif. Buku ini dirancang bukan sekadar sebagai manual penggunaan, melainkan sebuah manifestasi dari integrasi antara keagungan Ilmu Falak dan kecanggihan ilmu komputer modern.

Melalui penguasaan komputasi Python dalam astronomi Islam, kita tidak hanya melestarikan warisan intelektual ulama terdahulu, tetapi juga memajukan dan merelevansikannya untuk menyongsong tantangan peradaban digital di masa depan. Akhir kata, semoga seluruh arsitektur materi dan *source code* yang disajikan ini dapat memberikan manfaat yang seluas-luasnya, menjadi amal jariyah, dan turut andil dalam membuka rahasia kosmos demi kemaslahatan umat.

Fastabiqul khairat.

PENULIS

Kasmui

DAFTAR ISI

KATA PENGANTAR (IFTITAH).....	3
PERTEMUAN 1: Epistemologi Astronomi Islam dan Inisialisasi Ekosistem Python.....	9
1.1 Paradigma Sains Islam dalam Penentuan Waktu.....	9
1.2 Transisi Epistemologis: Dari <i>Zij</i> Klasik Menuju Integrasi Numerik Modern	9
1.3 Implementasi Python: Inisialisasi Ekosistem dan Unduh Ephemeris.....	10
1.4 Source Code Mandiri (Aplikasi Konsol Python).....	10
1.5 Penjelasan Anatomi <i>Source Code</i>	11
1.6 Daftar Pustaka (Pertemuan 1)	12
PERTEMUAN 2: Manajemen Basis Data Spasial dan Waktu Astronomis.....	13
2.1 Teologi Waktu dan Sinkronisasi Kosmik.....	13
2.2 Arsitektur Waktu Astronomis: Julian Day dan Skala Waktu Dinamis	13
2.3 Koordinat Spasial WGS84 dan Konsep <i>Local Mean Time</i> (LMT)	14
2.4 Source Code Mandiri: Mesin Waktu Astronomis dan Konverter LMT.....	14
2.5 Penjelasan Anatomi Source Code	16
2.6 Daftar Pustaka (Pertemuan 2)	17
PERTEMUAN 3: Geodesi Sferis dan Akurasi Arah Kiblat	18
3.1 Teologi Ruang dan Kewajiban Presisi Arah	18
3.2 Matematika Bola Bumi (<i>Spherical Trigonometry</i>).....	18
3.3 Algoritma Azimut dan Resolusi Kuadran (<code>math.atan2</code>)	19
3.4 Source Code Mandiri: Kalkulator Azimut Kiblat Presisi Tinggi.....	19
3.5 Penjelasan Anatomi Source Code	21
3.6 Daftar Pustaka (Pertemuan 3)	22
PERTEMUAN 4: Visualisasi Spasial Kompas Kiblat	23
4.1 Epistemologi Visualisasi dalam Ilmu Falak.....	23
4.2 Anatomi Sistem Koordinat Polar (Polar Plot).....	23
4.3 Rekayasa Pustaka Matplotlib untuk Navigasi Falak	24
4.4 Source Code Mandiri: Generator Kompas Kiblat Digital 2D	24
4.5 Penjelasan Anatomi Source Code	26
4.6 Daftar Pustaka (Pertemuan 4)	26
PERTEMUAN 5: Kinematika Matahari dan Equation of Time (Perata Waktu).....	28

5.1 Teologi Keteraturan dan Anomali Gerak Benda Langit.....	28
5.2 Kinematika Deklinasi dan Kemiringan Ekliptika	28
5.3 Fenomena Equation of Time (Perata Waktu)	29
5.4 Source Code Mandiri: Kalkulator Deklinasi dan Equation of Time	29
5.5 Penjelasan Anatomi Source Code	32
5.6 Daftar Pustaka (Pertemuan 5)	33
PERTEMUAN 6: <i>Istiwa' A'zam</i> dan <i>Rashdul Qibla</i> (Kalibrasi Arah Global)	34
6.1 Teologi Kalibrasi dan Fenomena Bayangan Matahari.....	34
6.2 Mekanika Astronomis <i>Istiwa' A'zam</i> (Great Culmination).....	34
6.3 Algoritma Pencarian Akar (<i>Root-Finding</i>) dalam Python	35
6.4 Source Code Mandiri: Mesin Pemindai <i>Rashdul Qibla</i> Global	35
6.5 Penjelasan Anatomi Source Code	37
6.6 Daftar Pustaka (Pertemuan 6)	38
PERTEMUAN 7: Algoritma Presisi Waktu Salat (Bagian 1: Zuhur & Asar).....	39
7.1 Teologi Waktu Salat dan Kewajiban Presisi Matematis.....	39
7.2 Mekanika Astronomis Waktu Zuhur (<i>Solar Transit</i> / Kulminasi Atas).....	39
7.3 Matematika Bayangan dan Algoritma Waktu Asar.....	40
7.4 Source Code Mandiri: Generator Waktu Zuhur dan Asar.....	40
7.5 Penjelasan Anatomi Source Code	43
7.6 Daftar Pustaka (Pertemuan 7)	43
PERTEMUAN 8: Ujian Tengah Semester (UTS) – Sintesis Modul Spasial dan Waktu Astronomis	45
8.1 Filosofi <i>Itqan</i> (Profesionalisme) dalam Komputasi Falak	45
8.2 Arsitektur Integrasi Sistem (Proyek UTS)	45
8.3 Source Code Mandiri: Proyek UTS (Sistem Falak Terpadu)	46
8.4 Penjelasan Anatomi Source Code	49
8.5 Daftar Pustaka (Pertemuan 8)	49
PERTEMUAN 9: Algoritma Presisi Waktu Salat (Bagian 2: Maghrib, Isya', Subuh)	50
9.1 Teologi Malam dan Fenomena <i>Fajar Shadiq</i>	50
9.2 Fisika Atmosferik dan Anomali Refraksi (Maghrib).....	50
9.3 Parameter Ketinggian Matahari untuk Isya' dan Subuh	51
9.4 Source Code Mandiri: Kalkulator Dinamis Salat Gelap	51
9.5 Penjelasan Anatomi Source Code	54
9.6 Daftar Pustaka (Pertemuan 9)	54

PERTEMUAN 10: Dinamika Mekanika Orbital Bulan dan Fase Ijtimak (Konjungsi)	55
10.1 Teologi Kalender Lunar dan <i>Manzilah</i> Bulan	55
10.2 Mekanika Orbital dan Definisi Astronomis Ijtimak	55
10.3 Algoritma Pelacakan Fase Bulan di Python.....	56
10.4 Source Code Mandiri: Detektor Fase Bulan dan Ijtimak Presisi.....	56
10.5 Penjelasan Anatomi Source Code	58
10.6 Daftar Pustaka (Pertemuan 10)	59
PERTEMUAN 11: Kalkulus Visibilitas Hilal (Toposentrik vs Geosentrik).....	60
11.1 Teologi <i>Ru'yah</i> dan Epistemologi Observasi Hilal	60
11.2 Transformasi Matriks: Geosentrik menuju Toposentrik.....	60
11.3 Kalkulus Parameter Visibilitas Hilal.....	61
11.4 Source Code Mandiri: Mesin Ekstraktor Matriks Visibilitas Hilal.....	61
11.5 Penjelasan Anatomi Source Code	64
11.6 Daftar Pustaka (Pertemuan 11)	65
PERTEMUAN 12: Implementasi Kriteria Rukyatul Hilal & Neo MABIMS	66
12.1 Teologi Konsensus (<i>Syura</i>) dan Otoritas Epistemologis.....	66
12.2 Evolusi Parameter MABIMS: Dari Klasik menuju Neo MABIMS	66
12.3 Konstruksi Pohon Keputusan (<i>Decision Tree</i>) Algoritmik.....	67
12.4 Source Code Mandiri: Mesin Filter Otomatis Kriteria Neo MABIMS	67
12.5 Penjelasan Anatomi Source Code	70
12.6 Daftar Pustaka (Pertemuan 12)	70
PERTEMUAN 13: Arsitektur Kalender Hijriah Global Tunggal (KHGT)	72
13.1 Teologi Kesatuan Waktu dan Paradigma Kalender Global	72
13.2 Anatomi Resolusi Istanbul 2016 (Parameter KHGT)	72
13.3 Transformasi Algoritma: Mengunci Batas Timur Bumi (Fajar Selandia Baru).....	73
13.4 Source Code Mandiri: Modul Ekstraksi dan Validasi Basis KHGT.....	73
13.5 Penjelasan Anatomi Source Code	76
13.6 Daftar Pustaka (Pertemuan 13)	76
PERTEMUAN 14: Mesin Pemindai Visibilitas Global (<i>Global Analyzer</i>)	78
14.1 Teologi Sferis dan Kontinuitas Waktu Semesta	78
14.2 Arsitektur Gerbang 2 KHGT: Imkanur Rukyat Global	78
14.3 Algoritma Pemindai Matriks Bumi (<i>Grid Scanner</i>)	79
14.4 Source Code Mandiri: Mesin Sapu Jagat (<i>Global Analyzer</i>).....	79

14.5 Penjelasan Anatomi Source Code	82
14.6 Daftar Pustaka (Pertemuan 14)	83
PERTEMUAN 15: Rekayasa Antarmuka Interaktif (GUI) dan <i>Bidi-Converter</i> Kalender	84
15.1 Epistemologi Dakwah Digital dan Aksesibilitas Ilmu.....	84
15.2 Arsitektur Antarmuka Modern (CustomTkinter)	84
15.3 Algoritma <i>Bidi-Converter</i> dan Anomali Batas Hari.....	84
15.4 Source Code Mandiri: Purwarupa Kalender GUI CustomTkinter.....	85
15.5 Penjelasan Anatomi Source Code	89
15.6 Daftar Pustaka (Pertemuan 15)	90
PERTEMUAN 16: Ujian Akhir Semester (UAS) – Sintesis Total dan Distribusi Biner (PyInstaller).....	91
16.1 Epistemologi Amal Jariyah dan Distribusi Teknologi	91
16.2 Arsitektur Sintesis: "Mini KHGT Times"	91
16.3 Source Code Mandiri: Proyek Akhir "Mini KHGT Times"	92
16.4 Distribusi Biner: Menggunakan PyInstaller	96
16.5 Penjelasan Anatomi Source Code dan Integrasi	96
16.6 Daftar Pustaka (Pertemuan 16)	97
DAFTAR PUSTAKA	98
GLOSARIUM	100
LAMPIRAN 1	106
LAMPIRAN 2	111
LAMPIRAN 3	114
LAMPIRAN 4	117
LAMPIRAN 5	120

PERTEMUAN 1: Epistemologi Astronomi Islam dan Inisialisasi Ekosistem Python

1.1 Paradigma Sains Islam dalam Penentuan Waktu

Astronomi Islam (Ilmu Falak) bukanlah sekadar instrumen teknis untuk mengetahui kapan ibadah dilaksanakan, melainkan sebuah manifestasi dari kosmologi tauhid yang memandang alam semesta sebagai sistem matematis yang terstruktur, rasional, dan dapat diprediksi. Integrasi antara hukum syariat dan hukum alam (sunnatullah) menuntut tingkat presisi (akurasi) yang absolut.

Al-Qur'an secara eksplisit meletakkan landasan epistemologis bahwa pergerakan benda langit adalah basis dari sistem komputasi waktu yang presisi:

الشَّمْسُ وَالْقَمَرُ بِحُسْبَانٍ

“Matahari dan bulan beredar menurut perhitungan.” (QS. Ar-Rahman [55]: 5)

Tafsir Ringkas: Kata *bihusbaan* (بِحُسْبَانٍ) berasal dari akar kata *hasaba* yang bermakna menghitung secara teliti. Imam Fakhruddin Ar-Razi dalam *Mafatih al-Ghaib* menjelaskan bahwa pergerakan kedua benda langit ini tunduk pada tata rias matematis yang sangat ketat, tanpa ada akselerasi maupun deselerasi yang acak. Keteraturan inilah yang melegitimasi penggunaan *Hisab Hakiki*—metode perhitungan yang merepresentasikan posisi riil benda langit secara faktual—sebagai fondasi utama penentuan waktu Islam kontemporer, termasuk dalam arsitektur Kalender Hijriah Global Tunggal (KHGT) yang diadopsi oleh Majelis Tarjih. Berbeda dengan hisab *taqribi* (pendekatan) yang abai terhadap anomali perturbasi orbit, hisab hakiki menuntut alat komputasi yang mampu menelan jutaan variabel mekanika langit.

1.2 Transisi Epistemologis: Dari *Zīj* Klasik Menuju Integrasi Numerik Modern

Secara historis, para astronom muslim seperti Al-Khawarizmi, Al-Battani, hingga Ulugh Beg memformulasikan posisi benda langit dalam bentuk tabel astronomi yang disebut *Zīj*. Namun, *Zīj* memiliki keterbatasan pada pembulatan nilai trigonometri sferis dan ketidakmampuan memodelkan perturbasi gravitasi antar-planet secara *real-time*.

Di era modern, epistemologi falak bergeser dari perhitungan analitik sederhana menuju **Integrasi Numerik**. Ini adalah metode kalkulus maju di mana posisi (vektor) dan kecepatan benda langit dihitung menggunakan persamaan diferensial hukum gravitasi Newton dan relativitas umum Einstein. Hasil dari integrasi numerik ini dipublikasikan oleh *Jet Propulsion Laboratory* (JPL) NASA dalam bentuk data ephemeris *Development Ephemeris* (DE), seperti DE421 atau DE440s.

Bahasa pemrograman Python, melalui pustaka tingkat observatorium seperti `skyfield` dan `PyEphem`, bertindak sebagai jembatan untuk mengekstrak data biner JPL DE tersebut menjadi matriks data astronomi yang siap diolah untuk kepentingan syar'i (seperti waktu salat dan visibilitas hilal).

1.3 Implementasi Python: Inisialisasi Ekosistem dan Unduh Ephemeris

Untuk membangun aplikasi astronomi mandiri, langkah fundamental pertama adalah mengamankan jalur pasokan data ephemeris. Pustaka `Skyfield` tidak membawa data internal yang berat; ia mengunduh *file* biner `.bsp` dari peladen NASA secara dinamis.

Seringkali, sistem operasi atau jaringan komputer menolak unduhan otomatis karena masalah verifikasi sertifikat SSL (*Secure Sockets Layer*). *Source code* di bawah ini adalah aplikasi konsol mandiri yang mengatasi isu SSL tersebut, mengunduh file DE421, dan memverifikasi posisi geosentris Bulan dan Matahari pada waktu saat kode dijalankan.

1.4 Source Code Mandiri (Aplikasi Konsol Python)

Simpan kode ini dengan nama `inisialisasi_ephemeris.py` dan jalankan di terminal.

Python

```
"""
Aplikasi: Inisialisasi Ekosistem Skyfield & Pengunduh Ephemeris JPL DE421
Penulis: Kasmui
Deskripsi: Skrip mandiri untuk mem-bypass protokol SSL yang ketat,
           mengunduh matriks data NASA JPL secara aman, dan
           melakukan komputasi posisi awal Matahari dan Bulan.
"""

import ssl
import urllib.request
from skyfield.api import Loader, N_POLES, WGS84

def bypass_ssl_and_load_ephemeris():
    print("="*60)
    print("INISIALISASI EKOSISTEM ASTRONOMI ISLAM (SKYFIELD)")
    print("="*60)

    # LANGKAH 1: Manipulasi Konteks SSL untuk menghindari 'Certificate
    Verify Failed'
    print("[1] Membangun konteks SSL yang aman (Bypass unverified
    HTTPS)...")
    try:
        _create_unverified_https_context = ssl._create_unverified_context
    except AttributeError:
        # Sistem lama yang tidak memiliki verifikasi ketat
        pass
    else:
        # Memaksa urllib menggunakan konteks yang tidak terverifikasi
        (jika diperlukan jaringan lokal)
        ssl._create_default_https_context =
        _create_unverified_https_context

    # LANGKAH 2: Membuat objek Loader dari Skyfield
    # File akan diunduh dan disimpan di direktori lokal 'ephemeris_data'
    print("[2] Menyiapkan direktori penyimpanan lokal
    ('ephemeris_data')...")
```

```

load = Loader('ephemeris_data', verbose=True)

# LANGKAH 3: Mengunduh data skala waktu (Leap Seconds) dan Ephemeris
DE421
print("[3] Mengunduh/Memuat Delta-T (Leap Seconds) dan Data Biner JPL
DE421...")
try:
    ts = load.timescale()
    planets = load('de421.bsp')
    print("    -> Data Ephemeris DE421 Berhasil Dimuat!")
except Exception as e:
    print(f"    [X] Gagal memuat data: {e}")
    return

# LANGKAH 4: Menarik Objek Tata Surya Spesifik
bumi = planets['earth']
matahari = planets['sun']
bulan = planets['moon']

# LANGKAH 5: Membaca Waktu Saat Ini (Real-Time UTC)
waktu_sekarang = ts.now()
waktu_utc = waktu_sekarang.utc_datetime()
print(f"\n[INFO] Waktu Komputasi: {waktu_utc.strftime('%Y-%m-%d
%H:%M:%S')} UTC")

# LANGKAH 6: Komputasi Vektor Jarak (Bumi ke Bulan & Bumi ke Matahari)
print("\n[INFO] Menghitung Vektor Posisi Geosentris...")

# Pengamatan dari pusat Bumi (Geosentris)
observasi_bulan = bumi.at(waktu_sekarang).observe(bulan)
observasi_matahari = bumi.at(waktu_sekarang).observe(matahari)

# Ekstraksi jarak absolut (dalam Kilometer)
_, _, jarak_bulan = observasi_bulan.apparent().radec()
_, _, jarak_matahari = observasi_matahari.apparent().radec()

print(f"    -> Jarak Bumi - Bulan      : {jarak_bulan.km:,.2f}
kilometer")
print(f"    -> Jarak Bumi - Matahari : {jarak_matahari.km:,.2f}
kilometer")
print("="*60)
print("STATUS: Ekosistem Python siap digunakan untuk perhitungan
Falak.")
print("="*60)

if __name__ == '__main__':
    bypass_ssl_and_load_ephemeris()

```

1.5 Penjelasan Anatomi *Source Code*

1. **Impor Pustaka Inti:** Modul `ssl` dan `urllib.request` digunakan untuk manajemen jaringan tingkat rendah. Modul `Loader` dari `skyfield.api` adalah *class* utama yang bertugas mengelola manajemen arsip eksternal.

2. **Manipulasi SSL (Langkah 1):** Python sering kali memblokir akses pengunduhan via HTTPS jika sertifikat otoritas (CA) di mesin lokal kedaluwarsa. Baris `ssl._create_unverified_context` digunakan secara strategis (sebagai *fallback*) agar pengunduhan data NASA tidak diblokir oleh *firewall* internal instansi maupun batasan OS.
3. **Sistem Tembolok Direktori (Langkah 2):** Argumen `'ephemeris_data'` pada fungsi `Loader` menginstruksikan program untuk membuat sebuah folder khusus. File biner `de421.bsp` (berukuran sekitar 16 MB) hanya akan diunduh **satu kali**. Pada saat kode dieksekusi ulang, program akan membaca tembolok (*cache*) lokal secara instan, meningkatkan efisiensi komputasi.
4. **Skala Waktu TAI/UTC (Langkah 3 & 5):** Fungsi `ts = load.timescale()` sangat krusial dalam Ilmu Falak. Bumi tidak berputar dengan kecepatan konstan secara absolut; rotasinya melambat karena friksi pasang surut (efek gravitasi Bulan). Oleh karena itu, *Skyfield* secara otomatis mengunduh data *Leap Seconds* (Detik Kabisat) melalui file `finals2000A.all` agar perhitungan posisi bulan tetap akurat hingga fraksi detik terakhir.
5. **Astro-Vektor Geosentris (Langkah 6):** Metode `.observe()` mengkalkulasi lintasan cahaya dari benda langit ke bumi, sedangkan metode `.apparent()` menerapkan koreksi defleksi cahaya (*light-time travel*) dan aberasi yang disebabkan oleh gerak bumi. Proses ini adalah ruh dari pergeseran hisab *taqribi* menuju *hisab hakiki*—karena posisi yang ditampilkan adalah posisi yang "tampak" (*apparent*) bukan hanya posisi geometris murni.

1.6 Daftar Pustaka (Pertemuan 1)

1. Anwar, Syamsul. (2018). *Hari-hari Allah: Dinamika Kalender Islam Kontemporer dan Menuju Kalender Hijriah Global*. Suara Muhammadiyah.
2. Rhodes, B. (2019). *Skyfield: High Precision Research-Grade Positions for Planets and Earth Satellites Explorer*. Astrophysics Source Code Library.
3. Urban, S. E., & Seidelmann, P. K. (Eds.). (2012). *Explanatory Supplement to the Astronomical Almanac* (3rd ed.). University Science Books.
4. Azhari, Susiknan. (2015). *Ilmu Falak: Perjumpaan Khazanah Islam dan Sains Modern*. Suara Muhammadiyah.

PERTEMUAN 2: Manajemen Basis Data Spasial dan Waktu Astronomis

2.1 Teologi Waktu dan Sinkronisasi Kosmik

Waktu, dalam pandangan ontologi Islam, bukan sekadar durasi yang berlalu secara linier, melainkan sebuah entitas yang diciptakan (makhluk) dan diatur dengan ketetapan matematis yang absolut untuk memandu siklus ibadah manusia. Penentuan awal bulan (kalender), jadwal salat harian, hingga penentuan arah tata ruang (kiblat) seluruhnya bergantung pada pemahaman yang presisi terhadap variabel waktu dan ruang.

Al-Qur'an menegaskan konstruksi astronomis ini secara spesifik:

هُوَ الَّذِي جَعَلَ الشَّمْسَ ضِيَاءً وَالْقَمَرَ نُورًا وَقَدَرَهُ مَنَازِلَ لِتَعْلَمُوا عَدَدَ السِّنِينَ وَالْحِسَابَ ۚ مَا خَلَقَ اللَّهُ ذَلِكَ إِلَّا بِالْحَقِّ ۗ يُفَصِّلُ
الْآيَاتِ لِقَوْمٍ يَعْلَمُونَ

“Dialah yang menjadikan matahari bersinar dan bulan bercahaya dan ditetapkan-Nya manzilah-manzilah (tempat-tempat) bagi perjalanan bulan itu, supaya kamu mengetahui bilangan tahun dan perhitungan (waktu). Allah tidak menciptakan yang demikian itu melainkan dengan hak. Dia menjelaskan tanda-tanda (kebesaran-Nya) kepada orang-orang yang mengetahui.” (QS. Yunus [10]: 5)

Tafsir Ringkas: Imam Fakhrudin Ar-Razi dalam *Mafatih al-Ghaib* memberikan penekanan tajam pada frasa *wa qaddarahu manaazila* (dan ditetapkan-Nya manzilah/fase-fase). Kata *taqdir* di sini bermakna pengukuran atau penentuan proporsi yang sangat teliti. Ar-Razi menjelaskan bahwa pergerakan benda langit memiliki hukum periodik yang stabil, yang menjadikannya instrumen paling valid untuk *al-hisaab* (komputasi astronomi). Dalam konteks astronomi modern, *manzilah* ini diterjemahkan ke dalam bentuk koordinat ekliptika dan ekuatorial yang berubah setiap detiknya akibat dinamika mekanika langit.

2.2 Arsitektur Waktu Astronomis: Julian Day dan Skala Waktu Dinamis

Kelemahan fatal dari kalender sipil (Masehi/Gregorian) maupun kalender Hijriah tradisional dalam komputasi mesin adalah adanya iregularitas: jumlah hari dalam sebulan tidak konstan (28, 29, 30, atau 31), dan adanya tahun kabisat. Komputer membutuhkan garis waktu yang linier dan absolut.

Untuk mengatasi hal ini, Ilmu Falak menggunakan sistem **Julian Day (JD)**.

Julian Day adalah perhitungan jumlah hari (termasuk fraksi desimal untuk jam, menit, dan detik) yang telah berlalu secara terus-menerus tanpa putus semenjak tengah hari (12:00 UT) pada tanggal 1 Januari 4713 Sebelum Masehi. Dengan format desimal ini, interval waktu antara dua peristiwa astronomis dapat dihitung hanya dengan pengurangan matematika dasar.

Selain JD, astronomi observasional membagi waktu menjadi dua skala utama yang harus disinkronkan oleh Python:

1. **Universal Time Coordinated (UTC):** Waktu sipil global yang didasarkan pada jam atom, namun dijaga agar tetap sejajar dengan rotasi bumi dengan sesekali menambahkan Detik Kabisat (*Leap Seconds*).
2. **Terrestrial Time (TT):** Waktu dinamis yang berjalan konstan secara matematis tanpa memedulikan rotasi bumi yang melambat. Selisih antara TT dan UTC disebut sebagai **Delta-T**. Pengabaian nilai Delta-T akan menyebabkan kalkulasi gerhana dan posisi bulan meleset secara signifikan.

2.3 Koordinat Spasial WGS84 dan Konsep *Local Mean Time* (LMT)

Dalam hisab kontemporer, bumi tidak lagi dimodelkan sebagai bola sempurna (*perfect sphere*), melainkan sebagai elipsoida gepat. Pustaka `skyfield` menggunakan standar **WGS84** (*World Geodetic System 1984*), sistem referensi geospasial yang sama dengan yang digunakan oleh satelit GPS. Koordinat toposentris (pengamatan dari permukaan bumi) wajib menyertakan Lintang (*Latitude*), Bujur (*Longitude*), dan Ketinggian (*Elevation*) di atas permukaan laut.

Koordinat bujur memiliki relasi langsung dengan waktu. Bumi berotasi 360 derajat dalam 24 jam, yang berarti 15 derajat bujur setara dengan 1 jam perjalanan matahari, atau **1 derajat bujur setara dengan 4 menit waktu**.

Zona waktu standar sipil (seperti WIB: UTC+7) adalah waktu buatan (artifisial) yang dipukul rata untuk satu wilayah luas. Dalam penetapan jadwal ibadah, kita menggunakan **Local Mean Time (LMT)** atau Waktu Menengah Lokal, yaitu waktu astronomis spesifik pada garis bujur persis di mana pengamat berada. Konversi dari UTC ke LMT adalah fundamental untuk menghitung waktu *zawal* (Matahari tepat di meridian lokal) secara akurat.

2.4 Source Code Mandiri: Mesin Waktu Astronomis dan Konverter LMT

Simpan skrip Python ini dengan nama `mesin_waktu_spasial.py` dan jalankan di terminal. Aplikasi ini tidak bergantung pada modul eksternal selain `skyfield` dan `pytz`.

```
Python
"""
Aplikasi: Mesin Waktu Astronomis & Konverter Geodesi Spasial
Penulis: Kasmui
Deskripsi: Skrip mandiri untuk memproses input koordinat (WGS84) dan waktu sipil,
           kemudian mengubahnya menjadi Julian Day, Terrestrial Time,
           serta mengkalkulasi Local Mean Time (LMT) secara dinamis.
"""

import datetime
import pytz
from skyfield.api import Loader, wgs84

def komputasi_waktu_dan_spasial():
```

```

print("="*70)
print("SISTEM MANAJEMEN WAKTU ASTRONOMIS DAN KOORDINAT WGS84")
print("="*70)

# 1. Inisialisasi Pustaka dan Waktu
# Menggunakan direktori lokal agar tidak unduh ulang jika sudah ada
dari Bab 1
load = Loader('ephemeris_data', verbose=False)
ts = load.timescale()

# 2. Parameter Input Spasial (Contoh: Kota Semarang, Jawa Tengah)
nama_lokasi = "Semarang, Jawa Tengah"
lintang = -6.9932 # Derajat desimal (Negatif untuk Lintang Selatan)
bujur = 110.4203 # Derajat desimal (Positif untuk Bujur Timur)
elevasi = 4 # Ketinggian dari permukaan laut (meter)
zona_waktu = 'Asia/Jakarta'

print(f"[DATA LOKASI] : {nama_lokasi}")
print(f"Koordinat : Lintang {lintang}°, Bujur {bujur}°")
print(f"Elevasi : {elevasi} mdp1")
print(f"Zona Waktu : {zona_waktu}\n")

# 3. Sinkronisasi Objek Pengamat (Observer) berdasarkan model Bumi
Elipsoida
lokasi_pengamat = wgs84.latlon(lintang, bujur, elevation_m=elevasi)

# 4. Input Waktu Sipil Pengguna
# Membaca waktu saat ini di komputer berdasarkan zona waktu target
tz_lokal = pytz.timezone(zona_waktu)
waktu_sipil_lokal = datetime.datetime.now(tz_lokal)

# Memasukkan waktu sipil ke dalam mesin waktu Skyfield
t = ts.from_datetime(waktu_sipil_lokal)

# 5. Ekstraksi Metrik Waktu Astronomis
waktu_utc = t.utc_datetime()
julian_day_tt = t.tt # Terrestrial Time Julian Day
delta_t = t.delta_t # Selisih TT dan UT1 (Leap Seconds)

print("[HASIL KOMPUTASI WAKTU ASTRONOMIS]")
print(f"Waktu Sipil Lokal (WIB) : {waktu_sipil_lokal.strftime('%Y-%m-%d %H:%M:%S %Z')}")
print(f"Waktu Universal (UTC) : {waktu_utc.strftime('%Y-%m-%d %H:%M:%S')} UTC")
print(f"Julian Day (Terrestrial) : {julian_day_tt:.6f} Hari")
print(f"Nilai Delta-T : {delta_t:.3f} Detik")

# 6. Komputasi Local Mean Time (LMT)
# Rumus: Beda Bujur Lokal dengan Titik Greenwich (0 derajat)
# 1 derajat = 4 menit (atau 240 detik)
offset_lmt_detik = bujur * 240.0

# Membuat objek timedelta berdasarkan offset detik LMT
delta_lmt = datetime.timedelta(seconds=offset_lmt_detik)

```

```

# LMT adalah waktu UTC ditambah selisih bujur
waktu_lmt = waktu_utc.replace(tzinfo=None) + delta_lmt

print("\n[HASIL KOMPUTASI LOCAL MEAN TIME (LMT)]")
# Mengonversi desimal offset ke dalam jam, menit, detik untuk
presentasi
offset_jam = int(offset_lmt_detik // 3600)
sisa_detik = offset_lmt_detik % 3600
offset_minut = int(sisa_detik // 60)
offset_detik_akhir = int(sisa_detik % 60)

print(f"Koreksi Bujur dari UTC   : +{offset_jam} Jam {offset_minut}
Menit {offset_detik_akhir} Detik")
print(f"Waktu LMT Titik Pengamat : {waktu_lmt.strftime('%Y-%m-%d
%H:%M:%S')}")

# Analisis Selisih Zona Waktu Sipil dengan LMT Sebenarnya
# Contoh WIB = UTC+7 (25200 detik). Kita cari selisihnya dengan LMT
selisih_sipil_lmt = offset_lmt_detik - 25200
menit_koreksi = int(selisih_sipil_lmt // 60)
detik_koreksi = int(abs(selisih_sipil_lmt) % 60)

print(f"Koreksi Jam Sipil ke LMT : {menit_koreksi} menit
{detik_koreksi} detik (Ditambahkan pada kalkulasi ibadah lokal)")
print("="*70)

if __name__ == '__main__':
    komputasi_waktu_dan_spasial()

```

2.5 Penjelasan Anatomi Source Code

- Objek WGS84 (Langkah 3):** Baris `wgs84.latlon(lintang, bujur, elevation_m=elevasi)` sangat penting. Dalam hisab klasik, tinggi tempat (elevasi) sering diabaikan atau ditambahkan di akhir kalkulasi menggunakan rumus dip/kerendahan ufuk sederhana. *Skyfield* memposisikan pengamat secara 3D di permukaan bumi, sehingga paralaks (pergeseran sudut pandang) benda langit akibat ketinggian sudah otomatis tertangani sejak awal inisialisasi koordinat.
- Manajemen Zona Waktu (Langkah 4):** Python natif sering kebingungan menangani DST (*Daylight Saving Time*) atau perubahan zona waktu lokal. Pustaka `pytz` menyelesaikan masalah ini dengan memberikan objek waktu sipil status *timezone-aware*.
- Ekstraksi Nilai Astronomis (Langkah 5):** * `t.tt` mencetak *Julian Day* berbasis *Terrestrial Time*. Nilai ini akan selalu unik dan membesar seiring detik berjalan, memudahkan kalkulasi siklus orbit Bulan.
 - `t.delta_t` menunjukkan perlambatan rotasi bumi dalam detik (saat ini berada di kisaran ~69 detik). Algoritma ini menarik data historis langsung dari arsip NASA yang diunduh pada Pertemuan 1.
- Matematika LMT (Langkah 6):** Baris `offset_lmt_detik = bujur * 240.0` adalah inti konversi geodesi ke waktu. Angka 240 berasal dari kalkulasi: 1 derajat bujur = 4 menit. Dan 4 menit = 240 detik. Menambahkan nilai ini secara linier ke waktu UTC akan

menghasilkan waktu pertengahan yang presisi untuk koordinat Lintang dan Bujur tersebut, sebelum nantinya dikoreksi oleh anomali elips orbit bumi (*Equation of Time* yang akan dibahas pada Pertemuan 5).

2.6 Daftar Pustaka (Pertemuan 2)

1. Ar-Razi, Fakhruddin. (1981). *Tafsir al-Fakhr ar-Razi al-Musyтахar bi at-Tafsir al-Kabir wa Mafatih al-Ghaib*. Dar al-Fikr.
2. Meeus, J. (1998). *Astronomical Algorithms* (2nd ed.). Willmann-Bell, Inc.
3. Morrison, L. V., & Stephenson, F. R. (2004). *Historical Values of the Earth's Clock Error ΔT and the Calculation of Eclipses*. *Journal for the History of Astronomy*.
4. Butkevich, A. G., & Lindegren, L. (2014). *Rigorous framework for relativistic astrometry*. *Astronomy & Astrophysics*.

PERTEMUAN 3: Geodesi Sferis dan Akurasi Arah Kiblat

3.1 Teologi Ruang dan Kewajiban Presisi Arah

Penentuan arah kiblat bukan sekadar formalitas arah mata angin, melainkan manifestasi dari kepatuhan spasial seorang muslim terhadap pusat peribadatan global (Ka'bah). Dalam konteks geografis yang jauh dari Makkah, penentuan arah ini mensyaratkan integrasi antara pemahaman fikih dan komputasi matematis tingkat tinggi.

Al-Qur'an secara definitif menetapkan perintah ini:

فَدَّرَى تَقْلَبَ وَجْهَكَ فِي السَّمَاءِ فَلَنُؤَلِّيَنَّكَ قِبْلَةً تَرْضَاهَا فَوَلِّ وَجْهَكَ شَطْرَ الْمَسْجِدِ الْحَرَامِ ۚ وَحَيْثُ مَا كُنْتُمْ فَوَلُّوا وُجُوهَكُمْ شَطْرَهُ

“Sungguh Kami (sering) melihat mukamu menengadahkan ke langit, maka sungguh Kami akan memalingkan kamu ke kiblat yang kamu sukai. Palingkanlah mukamu ke arah Masjidil Haram. Dan di mana saja kamu berada, palingkanlah mukamu ke arahnya.” (QS. Al-Baqarah [2]: 144)

Tafsir Ringkas: Imam Fakhruddin Ar-Razi dalam *Mafatih al-Ghaib* menyoroti penggunaan diksi *syathra* (شَطْرٌ) yang bermakna "ke arah" atau "menuju titik". Bagi mereka yang berada di luar batas pandangan mata (di luar Masjidil Haram), kewajibannya adalah menghadap ke arah geometris Ka'bah (*jihatul ka'bah*). Majelis Tarjih Muhammadiyah menegaskan bahwa pencarian arah ini tidak boleh mengandalkan perkiraan kasar (seperti asumsi bahwa Indonesia cukup menghadap "Barat"), melainkan wajib menggunakan metode *Hisab Hakiki* berbasis geometri ruang untuk menemukan lintasan terpendek.

3.2 Matematika Bola Bumi (*Spherical Trigonometry*)

Kesalahan fatal dalam komputasi kiblat tradisional sering kali bermula dari penggunaan peta datar (proyeksi Mercator) untuk menarik garis lurus antara dua kota. Bumi adalah bangun tiga dimensi (elipsoida). Lintasan terpendek antara dua titik di permukaan bola bumi bukanlah garis lurus dua dimensi, melainkan busur dari sebuah Lingkaran Besar (*Great Circle*).

Ilmu Falak menggunakan Trigonometri Sferis (Ilmu Ukur Segitiga Bola) untuk menyelesaikan masalah ini. Segitiga bola kiblat dibentuk oleh tiga titik utama:

1. Titik Kutub Utara Sejati (titik referensi azimut 0°).
2. Titik Pengamat (koordinat lokal).
3. Titik Ka'bah di Makkah (Lintang $21^\circ 25' 21''$ LU, Bujur $39^\circ 49' 34''$ BT).

3.3 Algoritma Azimut dan Resolusi Kuadran (`math.atan2`)

Rumus fundamental trigonometri sferis untuk mencari sudut azimut kiblat (A) dari arah Utara Sejati dihitung menggunakan relasi fungsi trigonometri sinus, cosinus, dan tangen terhadap lintang lokal (ϕ_L), lintang Ka'bah (ϕ_K), dan selisih bujur ($\Delta\lambda$).

Secara matematis, rumusan tersebut didefinisikan sebagai:

Formula Azimut Kiblat:

$$\tan(A) = \sin(\Delta\lambda) / (\cos(\phi_L) * \tan(\phi_K) - \sin(\phi_L) * \cos(\Delta\lambda))$$

Keterangan Variabel:

- A = Azimut (Arah Kiblat)
- $\Delta\lambda$ = Selisih Bujur (Bujur Ka'bah - Bujur Lokal)
- ϕ_L = Lintang Lokal (Lintang pengamat)
- ϕ_K = Lintang Ka'bah (Lintang tujuan)

Dalam pemrograman komputer biasa, penggunaan fungsi arc-tangen standar (`atan`) sering memicu "Anomali Kuadran" dan *Zero Division Error* (saat penyebut bernilai nol). Untuk mengatasi kelemahan ini, Python menyediakan fungsi `math.atan2(y, x)` yang sangat superior. Fungsi ini membaca tanda positif/negatif dari pembilang (y) dan penyebut (x) secara terpisah, sehingga secara otomatis mengembalikan sudut yang presisi di kuadran yang tepat (dari -180° hingga $+180^\circ$), tanpa memerlukan koreksi *if-else* manual yang rumit.

3.4 Source Code Mandiri: Kalkulator Azimut Kiblat Presisi Tinggi

Simpan skrip Python ini dengan nama `kalkulator_kiblat.py` dan jalankan di terminal. Aplikasi ini menggunakan pustaka bawaan Python (`math`) sehingga tidak memerlukan instalasi eksternal, menjadikannya modul biner yang sangat ringan.

```
Python
"""
Aplikasi: Kalkulator Azimut Kiblat Presisi (Trigonometri Sferis)
Penulis: Kasmui
Deskripsi: Skrip mandiri untuk menghitung arah kiblat dari koordinat lokal
           menuju koordinat absolut Ka'bah menggunakan perhitungan Great
           Circle
           dan resolusi kuadran math.atan2.
"""

import math

def konversi_desimal_ke_dms(derajat_desimal):
    """Fungsi pembantu untuk mengubah derajat desimal ke format Derajat,
    Menit, Detik (DMS)"""
    is_positif = derajat_desimal >= 0
    derajat_absolut = abs(derajat_desimal)
```

```

derajat = int(derajat_absolut)
sisa_menit = (derajat_absolut - derajat) * 60
menit = int(sisa_menit)
detik = (sisa_menit - menit) * 60

return derajat, menit, detik, is_positif

def hitung_arah_kiblat():
    print("="*65)
    print("MESIN KOMPUTASI ARAH KIBLAT (SPHERICAL TRIGONOMETRY)")
    print("="*65)

    # 1. Konstanta Koordinat Ka'bah (WGS84)
    # Lintang: 21° 25' 21" LU | Bujur: 39° 49' 34" BT
    LINTANG_KABAH = 21.422487
    BUJUR_KABAH = 39.826206

    # 2. Input Koordinat Pengamat (Contoh: Kota Semarang)
    nama_lokasi = "Semarang, Jawa Tengah"
    lintang_lokal = -6.9932 # Negatif untuk Lintang Selatan
    bujur_lokal = 110.4203 # Positif untuk Bujur Timur

    print(f"[TARGET] Ka'bah (Makkah) : Lintang {LINTANG_KABAH:.6f}°,
    Bujur {BUJUR_KABAH:.6f}°")
    print(f"[ASAL] {nama_lokasi:15}: Lintang {lintang_lokal:.6f}°,
    Bujur {bujur_lokal:.6f}°\n")

    # 3. Konversi seluruh derajat ke Radian (Syarat mutlak fungsi
    trigonometri Python)
    phi_k = math.radians(LINTANG_KABAH)
    lam_k = math.radians(BUJUR_KABAH)
    phi_l = math.radians(lintang_lokal)
    lam_l = math.radians(bujur_lokal)

    # 4. Kalkulasi Selisih Bujur (Delta Lambda)
    delta_lam = lam_k - lam_l

    # 5. Eksekusi Rumus Geodesi Sferis
    # Pembilang (y) = sin(delta_bujur)
    y = math.sin(delta_lam)

    # Penyebut (x) = cos(lintang_lokal) * tan(lintang_kabah) -
    sin(lintang_lokal) * cos(delta_bujur)
    x = (math.cos(phi_l) * math.tan(phi_k)) - (math.sin(phi_l) *
    math.cos(delta_lam))

    # 6. Resolusi Kuadran menggunakan math.atan2
    # atan2(y, x) menghasilkan nilai dalam radian dari -PI hingga +PI
    azimuth_radian = math.atan2(y, x)

    # 7. Konversi Radian kembali ke Derajat
    azimuth_derajat = math.degrees(azimuth_radian)

```

```

# 8. Normalisasi Sudut Kompas (Memastikan hasil dalam rentang 0° -
360°)
if azimut_derajat < 0:
    azimut_derajat += 360.0

# 9. Format Output (Derajat, Menit, Detik)
d, m, s, _ = konversi_desimal_ke_dms(azimut_derajat)

print("[HASIL KOMPUTASI AZIMUT KIBLAT]")
print(f"Azimut dari Utara Sejati : {azimut_derajat:.4f}° (Derajat
Desimal)")
print(f"Format Sexagesimal (DMS) : {d}° {m}' {s:.2f}\" UTS (Utara ke
Timur ke Selatan)")

# Menghitung sudut pelurus dari titik Barat (270°) untuk referensi
praktis di Indonesia
sudut_dari_barat = 270.0 - azimut_derajat
if sudut_dari_barat > 0:
    print(f"Deviasi dari titik Barat : Melenceng
{sudut_dari_barat:.4f}° ke arah Utara")
else:
    print(f"Deviasi dari titik Barat : Melenceng
{abs(sudut_dari_barat):.4f}° ke arah Selatan")

print("="*65)

if __name__ == '__main__':
    hitung_arah_kiblat()

```

3.5 Penjelasan Anatomi Source Code

1. **Fungsi Pembantu DMS (Langkah 1):** `konversi_desimal_ke_dms` adalah algoritma esensial dalam astronomi observasional. Instrumen fisik di lapangan (seperti Theodolit atau Rubu' Mujayyab) beroperasi menggunakan skala derajat sexagesimal (Derajat, Menit, Detik). Fungsi ini memecah bilangan desimal murni ke dalam format presisi instrumen tersebut.
2. **Radian sebagai Bahasa Mesin (Langkah 3):** Mesin komputasi modern (termasuk C++ di balik Python) tidak memproses fungsi trigonometri (`sin`, `cos`, `tan`) dalam satuan derajat. Sudut wajib dikonversi menjadi *radian* terlebih dahulu dengan rasio $\pi/180$.
3. **Matriks Geodesi y dan x (Langkah 5):** Variabel y dan x berturut-turut merepresentasikan pembilang dan penyebut dari persamaan segitiga bola. Pemisahan ini krusial agar nilai positif/negatif ruang tiga dimensi bumi tidak hilang saat pembagian.
4. **`math.atan2` (Langkah 6):** Ini adalah jantung dari program. Jika hanya menggunakan `math.atan(y/x)`, koordinat yang berada di benua Amerika Selatan (di mana selisih bujur sangat ekstrem) dapat menghasilkan sudut yang terbalik 180 derajat. `atan2` melacak tanda vektor x dan y secara simultan untuk menempatkan sudut akhir pada kuadran kompas yang 100% absolut.
5. **Normalisasi Kompas (Langkah 8):** Kompas pelaut dan instrumen geodesi berputar searah jarum jam dari 0° (Utara) hingga 360° . Fungsi *if* memastikan bahwa jika `atan2`

mengembalikan sudut negatif (misal -65° yang berarti putaran ke kiri/Barat Laut), ia langsung dijumlahkan dengan 360° agar menjadi azimut positif standar (295°).

3.6 Daftar Pustaka (Pertemuan 3)

1. Ar-Razi, Fakhruddin. (1981). *Tafsir al-Fakhr ar-Razi al-Musytahar bi at-Tafsir al-Kabir wa Mafatih al-Ghaib*. Dar al-Fikr.
2. Majelis Tarjih dan Tajdid PP Muhammadiyah. (2009). *Pedoman Hisab Muhammadiyah*. Majelis Tarjih dan Tajdid PP Muhammadiyah.
3. Meeus, J. (1998). *Astronomical Algorithms* (2nd ed.). Willmann-Bell, Inc. (Bab 16: *Angular Separation*).
4. Smart, W. M. (1977). *Textbook on Spherical Astronomy* (6th ed.). Cambridge University Press.

PERTEMUAN 4: Visualisasi Spasial Kompas Kiblat

4.1 Epistemologi Visualisasi dalam Ilmu Falak

Setelah komputasi matematis menghasilkan angka azimut yang presisi absolut pada Pertemuan 3, tantangan akademik berikutnya adalah mentranslasikan data numerik murni tersebut ke dalam literasi visual yang dapat dipahami oleh masyarakat umum maupun takmir masjid. Dalam tradisi Majelis Tarjih, purifikasi ibadah harus diiringi dengan edukasi yang memadai. Angka azimut seperti 295.15° seringkali menjadi abstrak tanpa representasi grafis yang memetakan posisi pengamat terhadap ruang di sekelilingnya.

Al-Qur'an secara repetitif menekankan urgensi orientasi spasial yang berkelanjutan:

وَمِنْ حَيْثُ خَرَجْتَ فَوَلِّ وَجْهَكَ شَطْرَ الْمَسْجِدِ الْحَرَامِ وَإِنَّهُ لَلْحَقُّ مِنْ رَبِّكَ وَمَا اللَّهُ بِغَفِيلٍ عَمَّا تَعْمَلُونَ

“Dan dari mana saja kamu keluar (berangkat), maka palingkanlah wajahmu ke arah Masjidil Haram. Dan sesungguhnya ketentuan itu benar-benar sesuatu yang hak dari Tuhanmu. Dan Allah sekali-kali tidak lengah dari apa yang kamu kerjakan.” (QS. Al-Baqarah [2]: 149)

Tafsir Ringkas: Imam Fakhrudin Ar-Razi menggarisbawahi frasa *wa min haytsu kharajta* (dari mana saja kamu keluar) sebagai indikasi bahwa mobilitas manusia melintasi berbagai zona geografis menuntut adaptasi penentuan arah yang dinamis. Pengetahuan tentang arah tidak boleh statis. Oleh karena itu, membangun instrumen—baik fisik berupa kompas maupun digital berupa antarmuka grafis (GUI)—merupakan bentuk manifestasi *ijtihad* teknologi untuk memfasilitasi perintah syariat ini agar "sesuatu yang hak" (kebenaran presisi) dapat ditegakkan dengan keyakinan penuh secara visual.

4.2 Anatomi Sistem Koordinat Polar (Polar Plot)

Layar komputer dan antarmuka perangkat lunak pada dasarnya beroperasi pada sistem koordinat Kartesian (sumbu X dan Y datar). Namun, kompas dan arah navigasi bumi beroperasi pada geometri sferis dan rotasi melingkar. Untuk menjembatani hal ini, Python menggunakan proyeksi **Sistem Koordinat Polar**.

Dalam plot polar, setiap titik pada bidang ditentukan oleh jarak dari titik referensi pusat (disebut *radius*, direpresentasikan sebagai r) dan sudut dari arah referensi (direpresentasikan sebagai theta atau θ).

Secara standar bawaan matematika komputasi, sudut 0 derajat dimulai dari sumbu X positif (arah jam 3 atau arah Timur) dan berputar berlawanan arah jarum jam (*counter-clockwise*). Hal ini bertentangan dengan standar navigasi darat dan Ilmu Falak, di mana 0 derajat mutlak berada di Utara (arah jam 12) dan berputar searah jarum jam (*clockwise*). Oleh karena itu, diperlukan

rekayasa manipulasi sumbu menggunakan pustaka visualisasi data tingkat lanjut, yaitu Matplotlib.

4.3 Rekayasa Pustaka Matplotlib untuk Navigasi Falak

Pustaka `Matplotlib` adalah standar industri untuk visualisasi data saintifik di ekosistem Python. Untuk membangun kompas digital, kita harus "membajak" pengaturan *default* polar plot matriks:

1. **Translasi Titik Nol (Zero Location):** Menggeser titik 0 derajat dari posisi jam 3 ke posisi jam 12 (Utara Sejati).
2. **Inversi Rotasi (Theta Direction):** Mengubah putaran sudut dari matematis trigonometri (berlawanan jarum jam) menjadi navigasi geodesi (searah jarum jam).
3. **Anotasi Vektor:** Menggambar anak panah (vektor) dari pusat pengamat ($r=0$) menuju radius terluar tepat pada sudut azimuth kiblat yang telah dihitung sebelumnya.

4.4 Source Code Mandiri: Generator Kompas Kiblat Digital 2D

Simpan skrip Python ini dengan nama `visualisasi_kompas.py` dan jalankan di terminal. Pastikan Anda telah menginstal pustaka `matplotlib` (perintah: `pip install matplotlib`). Skrip ini akan membuka jendela antarmuka grafis mandiri.

```
Python
"""
Aplikasi: Visualisasi Kompas Kiblat Digital 2D (Matplotlib GUI)
Penulis: Kasmui
Deskripsi: Skrip mandiri untuk menghasilkan antarmuka grafis kompas
navigasi.
        Sistem mengubah azimuth numerik menjadi proyeksi polar
        dengan orientasi Utara Sejati (True North).
"""

import matplotlib.pyplot as plt
import numpy as np

def konversi_azimut_ke_radian(azimut_derajat):
    """Matplotlib polar plot menggunakan Radian untuk sudut theta"""
    return azimut_derajat * (np.pi / 180.0)

def render_kompas_kiblat():
    print("="*60)
    print("MENGHASILKAN RENDER VISUAL KOMPAS KIBLAT")
    print("="*60)

    # 1. Parameter Input (Dihasilkan dari perhitungan Pertemuan 3)
    # Contoh untuk wilayah Semarang, Jawa Tengah
    nama_lokasi = "Semarang, Jawa Tengah"
    azimut_kiblat_derajat = 294.45 # Derajat desimal dari Utara Sejati

    print(f"[INFO] Merender Grafik untuk Lokasi : {nama_lokasi}")
    print(f"[INFO] Sudut Azimut Target          : {azimut_kiblat_derajat}°")
```

```

# Konversi ke Radian untuk kebutuhan plotting
azimut_radian = konversi_azimut_ke_radian(azimut_kiblat_derajat)

# 2. Inisialisasi Kanvas/Figure Matplotlib (Mode Gelap/Elegan)
plt.style.use('dark_background')
fig = plt.figure(figsize=(7, 7), dpi=100)
fig.canvas.manager.set_window_title("Kompas Arah Kiblat Digital")

# 3. Menambahkan Sumbu Polar (Polar Axes)
ax = fig.add_subplot(111, polar=True)

# 4. Kalibrasi Sistem Navigasi Geodesi
# Memutar angka 0 derajat agar berada di posisi ATAS (Utara)
ax.set_theta_zero_location("N")

# Mengatur arah putaran menjadi Searah Jarum Jam (Clockwise = -1)
ax.set_theta_direction(-1)

# 5. Modifikasi Tampilan Grid dan Label Mata Angin
# Menghapus angka derajat default dan mengganti dengan teks navigasi
ax.set_xticks(np.radians([0, 45, 90, 135, 180, 225, 270, 315]))
ax.set_xticklabels(['UTARA', 'TL', 'TIMUR', 'TG', 'SELATAN', 'BD',
'BARAT', 'BL'],
                    fontsize=10, fontweight='bold', color='#cbd5e1')

# Menyembunyikan lingkaran radius internal agar tampilan bersih
ax.set_yticks([])
ax.spines['polar'].set_color('#334155') # Warna garis tepi luar

# 6. Menggambar Titik Pusat Pengamat
ax.plot(0, 0, marker='o', markersize=15, color='#e2e8f0')
ax.text(0, 0.1, "Titik\nPengamat", horizontalalignment='center',
        verticalalignment='bottom', color='white', fontsize=8)

# 7. Menggambar Vektor/Anak Panah Kiblat
# annotate digunakan untuk menggambar panah pada plot polar
ax.annotate('',
            xy=(azimut_radian, 1.0), # Ujung panah (Target Sudut,
Radius Luar)
            xytext=(0, 0), # Pangkal panah (Pusat)
            arrowprops=dict(facecolor='#10b981', edgecolor='#059669',
                            width=4, headwidth=15, headlength=15,
shrink=0.05))

# 8. Menambahkan Label Teks Ka'bah di Ujung Panah
ax.text(azimut_radian, 1.15, "🕌 KA'BAH\nKiblat",
        horizontalalignment='center', verticalalignment='center',
        fontsize=12, fontweight='bold', color='#34d399')

# 9. Dekorasi Teks Judul dan Informasi Footer
plt.title(f"Visualisasi Arah Kiblat\n{nama_lokasi}\n",
        fontsize=14, fontweight='bold', color='white', pad=20)

```

```

    footer_text = f"Azimuth: {azimuth_kiblat_derajat}° (UTS)\nMetode:
Geodesi Sferis Hakiki"
    plt.figtext(0.5, 0.05, footer_text, ha="center", fontsize=10,
               color="#94a3b8", bbox={"facecolor": "#1e293b", "alpha":
0.5, "pad": 5})

    # 10. Tampilkan Jendela Antarmuka (Blocking Call)
    print("[STATUS] Jendela Grafis GUI Sedang Berjalan. Tutup jendela
untuk mengakhiri program.")
    plt.tight_layout()
    plt.show()

if __name__ == '__main__':
    render_kompas_kiblat()

```

4.5 Penjelasan Anatomi Source Code

1. **Infrastruktur NumPy dan Radian (Langkah 1):** Sama seperti fungsi matematika bawaan Python, Matplotlib memetakan sudut sferis secara eksklusif menggunakan satuan Radian. Konstanta `np.pi` digunakan untuk memecah derajat desimal kompas ke dalam bahasa ruang sirkular mesin.
2. **Mode Gelap dan Manipulasi Sumbu (Langkah 2 & 3):** Penggunaan `plt.style.use('dark_background')` mengadopsi prinsip desain *User Interface* modern untuk mengurangi ketegangan mata (ergonomi visual). Fungsi `add_subplot(111, polar=True)` menginstruksikan mesin rendering untuk membuang kisi X-Y Kartesian dan menggantinya dengan jaring-jaring jaring laba-laba sferis.
3. **Kalibrasi Resolusi Navigasi (Langkah 4):** Dua baris sakti `set_theta_zero_location("N")` dan `set_theta_direction(-1)` adalah kunci penyesuaian dari ranah "matematika murni" ke "ilmu falak terapan". Tanpa kedua instruksi ini, panah kiblat akan melenceng ke arah yang tidak masuk akal secara geografis.
4. **Matriks Anotasi Vektor (Langkah 7):** Penggunaan fungsi `plot` standar tidak mampu merender ujung mata panah. Kita memanfaatkan metode `annotate` dengan `arrowprops`. Parameter `xy=(azimuth_radian, 1.0)` memerintahkan titik panah untuk mendarat tepat pada sudut yang telah dihitung secara trigonometri di Pertemuan 3, dan melesat hingga batas radius terluar (diwakili oleh nilai jarak relatif 1.0).
5. **Eksekusi Siklus Interaktif (Langkah 10):** Pemanggilan fungsi `plt.show()` akan membekukan eksekusi terminal (proses *blocking*) dan menyerahkan kontrol kepada *Operating System* (Windows/Linux/Mac) untuk memunculkan jendela GUI secara otonom. Jendela ini sepenuhnya interaktif; pengguna dapat menyimpan gambar tersebut sebagai PDF atau PNG langsung dari tombol yang tersedia di jendela aplikasi.

4.6 Daftar Pustaka (Pertemuan 4)

1. Ar-Razi, Fakhruddin. (1981). *Tafsir al-Fakhr ar-Razi al-Musytahar bi at-Tafsir al-Kabir wa Mafatih al-Ghaib*. Dar al-Fikr.
2. Hunter, J. D. (2007). *Matplotlib: A 2D graphics environment*. *Computing in Science & Engineering*, 9(3), 90-95.

3. Majelis Tarjih dan Tajdid PP Muhammadiyah. (2009). *Pedoman Hisab Muhammadiyah*. Majelis Tarjih dan Tajdid PP Muhammadiyah.
4. VanderPlas, J. (2016). *Python Data Science Handbook: Essential Tools for Working with Data*. O'Reilly Media, Inc.

PERTEMUAN 5: Kinematika Matahari dan Equation of Time (Perata Waktu)

5.1 Teologi Keteraturan dan Anomali Gerak Benda Langit

Dalam arsitektur tata surya, matahari tampak bergerak melintasi bola langit dari timur ke barat setiap hari. Namun, pergerakan ini bukanlah gerak melingkar beraturan yang seragam. Paradigma astronomi Islam menyadari bahwa "keteraturan" kosmos tidak berarti "keseragaman" yang statis, melainkan sebuah harmoni dinamis yang melibatkan fluktuasi matematis yang sangat presisi.

Allah Subhanahu wa Ta'ala berfirman mengenai ketetapan orbit matahari ini:

وَالشَّمْسُ تَجْرِي لِمُسْتَقَرٍّ لَهَا ذَلِكَ تَقْدِيرُ الْعَزِيزِ الْعَلِيمِ

“Dan matahari berjalan di tempat peredarannya. Demikianlah ketetapan Yang Maha Perkasa lagi Maha Mengetahui.” (QS. Ya-Sin [36]: 38)

Tafsir Ringkas: Imam Fakhruddin Ar-Razi dalam *Mafatih al-Ghaib* menjelaskan bahwa kata *mustaqarrin* (tempat peredaran/titik henti) mengandung isyarat astronomis yang dalam. Ia tidak hanya bermakna orbit spasial, tetapi juga batas-batas kinematik ekstrem (seperti titik balik utara dan selatan pada saat solstis). Diksi *taqdir* menegaskan bahwa meskipun gerak matahari memiliki variasi (kadang terasa lebih cepat atau lambat mencapai kulminasi), seluruh anomali tersebut tunduk pada sebuah persamaan ketetapan Ilahi yang dapat dikalkulasi secara empiris oleh manusia (*al-'Alim*).

5.2 Kinematika Deklinasi dan Kemiringan Ekliptika

Bumi mengorbit matahari pada sebuah bidang datar yang disebut bidang *ekliptika*. Namun, sumbu rotasi bumi tidak tegak lurus terhadap bidang ekliptika ini, melainkan miring dengan sudut sekitar 23,44 derajat (Obliquititas Ekliptika).

Akibat kemiringan ini, saat bumi mengelilingi matahari, pengamat di bumi melihat matahari bergerak naik turun secara periodik terhadap ekuator langit (khatulistiwa). Jarak sudut matahari dari ekuator langit ini disebut **Deklinasi Matahari**.

1. Pada sekitar 21 Maret dan 23 September (Ekuinoks), deklinasi matahari bernilai 0 derajat. Matahari terbit tepat di timur dan terbenam tepat di barat.
2. Pada 21 Juni (Titik Balik Musim Panas/Solstis Utara), deklinasi mencapai maksimum utara (+23,44 derajat).
3. Pada 22 Desember (Titik Balik Musim Dingin/Solstis Selatan), deklinasi mencapai maksimum selatan (-23,44 derajat).

Deklinasi ini adalah variabel utama dalam penentuan lintasan bayangan benda, yang secara langsung menjadi parameter hisab penentuan awal waktu Zuhur dan Asar.

5.3 Fenomena Equation of Time (Perata Waktu)

Jika kita mendirikan sebuah jam matahari (sundial) dan membandingkannya dengan jam tangan modern (arloji), kita akan menemukan bahwa keduanya jarang sekali menunjukkan angka 12:00 secara bersamaan. Terkadang matahari mencapai titik kulminasi (zawal) lebih cepat dari jam 12:00 siang lokal, dan terkadang lebih lambat hingga selisih 16 menit.

Selisih antara **Waktu Matahari Sejati** (Apparent Solar Time, yang ditunjukkan oleh alam/bayangan) dan **Waktu Matahari Rata-rata** (Mean Solar Time, yang berdetak konstan di jam tangan kita) disebut sebagai **Equation of Time (EoT)** atau *Perata Waktu*.

Persamaan dasarnya dituliskan sebagai:

$$\text{EoT} = \text{Waktu Matahari Sejati} - \text{Waktu Matahari Rata-rata}$$

Fenomena EoT ini disebabkan oleh dua faktor mekanika langit yang saling bertumpuk:

1. **Eksentrisitas Orbit Bumi:** Orbit bumi berbentuk elips (Hukum Kepler I). Saat bumi berada paling dekat dengan matahari (Perihelion, di bulan Januari), bumi bergerak lebih cepat di orbitnya. Saat paling jauh (Aphelion, di bulan Juli), bumi bergerak lebih lambat.
2. **Obliquititas Ekliptika:** Gerak semu matahari terjadi di sepanjang ekliptika yang miring, bukan ekuator. Saat diproyeksikan ke ekuator langit (tempat jam kita diukur), pergerakan matahari di sekitar titik solstis tampak lebih cepat dibandingkan saat berada di titik ekuinoks.

Majelis Tarjih secara ketat memasukkan nilai Perata Waktu (EoT) dalam setiap komputasi waktu salat, karena mengabaikan variabel ini akan menyebabkan jadwal salat meleset lebih dari seperempat jam pada bulan-bulan tertentu (terutama November dan Februari).

5.4 Source Code Mandiri: Kalkulator Deklinasi dan Equation of Time

Simpan skrip Python ini dengan nama `kinematika_matahari.py` dan jalankan di terminal. Aplikasi ini akan menghitung nilai Deklinasi dan Perata Waktu pada hari yang diinputkan, serta secara otomatis merender grafik fluktuasi EoT selama satu tahun penuh menggunakan matplotlib.

Python

```
"""
```

```
Aplikasi: Mesin Kalkulasi Kinematika Matahari & Equation of Time (EoT)
```

```
Penulis: Kasmui
```

```
Deskripsi: Skrip mandiri untuk menghitung Deklinasi Matahari presisi tinggi,
```

```
    menghitung selisih Perata Waktu harian, dan menghasilkan plot grafik Analemma/EoT selama 365 hari.
```

```
"""
```

```

import datetime
import math
import matplotlib.pyplot as plt
import numpy as np
from skyfield.api import Loader, wgs84

def hitung_perata_waktu():
    print("="*70)
    print("MESIN KOMPUTASI KINEMATIKA MATAHARI & EQUATION OF TIME (EOT)")
    print("="*70)

    # 1. Inisialisasi Pustaka Ephemeris JPL
    load = Loader('ephemeris_data', verbose=False)
    ts = load.timescale()
    planets = load('de421.bsp')
    bumi = planets['earth']
    matahari = planets['sun']

    # 2. Input Tanggal Pengamatan (Contoh: Waktu saat ini)
    sekarang = datetime.datetime.now(datetime.timezone.utc)
    t_hari_ini = ts.from_datetime(sekarang)

    # 3. Komputasi Posisi Matahari Sejati (Geosentris Ekuatorial)
    astrometrik = bumi.at(t_hari_ini).observe(matahari)
    apparent = astrometrik.apparent()
    ra, dec, jarak = apparent.radec()

    deklinasi_derajat = dec.degrees

    # 4. Kalkulasi Equation of Time (Metode Selisih Sudut Jam/RA)
    # Pendekatan analitik: Menghitung selisih posisi bujur rata-rata
    matahari
    # terhadap Asensio Rekta (Right Ascension) sejati.
    # Waktu J2000 dalam Julian Century
    T = (t_hari_ini.tt - 2451545.0) / 36525.0

    # Bujur rata-rata geometri matahari (Derajat)
    L0 = 280.46646 + 36000.76983 * T + 0.0003032 * (T**2)
    L0 = L0 % 360.0

    # Konversi RA dari format jam (0-24) ke derajat (0-360)
    ra_derajat = ra.hours * 15.0

    # Normalisasi selisih kuadran
    selisih_derajat = L0 - ra_derajat
    if selisih_derajat > 180:
        selisih_derajat -= 360
    elif selisih_derajat < -180:
        selisih_derajat += 360

    # Konversi selisih derajat kembali ke satuan waktu (1 derajat = 4
    menit)
    eot_minit = selisih_derajat * 4.0

```

```

print(f"[INFO TANGGAL] : {sekarang.strftime('%Y-%m-%d %H:%M:%S')}
UTC")
print(f"Deklinasi Matahari    : {deklinasi_derajat:+.4f} Derajat")

if eot_menit > 0:
    print(f"Equation of Time      : +{eot_menit:.2f} Menit (Matahari
mendahului jam rata-rata)")
else:
    print(f"Equation of Time      : {eot_menit:.2f} Menit (Matahari
tertinggal dari jam rata-rata)")

# 5. GENERATOR GRAFIK EOT SELAMA 1 TAHUN PENUH
print("\n[PROSES] Membuat Grafik Fluktuasi Equation of Time (1
Tahun)...")

tahun_berjalan = sekarang.year
hari_dalam_tahun = np.arange(1, 366)
nilai_eot = []

# Iterasi untuk setiap hari di jam 12:00 UTC
for hari in hari_dalam_tahun:
    dt = datetime.datetime(tahun_berjalan, 1, 1, 12, 0, 0,
tzinfo=datetime.timezone.utc) + datetime.timedelta(days=int(hari)-1)
    t_iterasi = ts.from_datetime(dt)

    # Ekstraksi RA Sejati
    ra_iter, _, _ =
bumi.at(t_iterasi).observe(matahari).apparent().radec()

    # Ekstraksi Bujur Rata-rata
    T_iter = (t_iterasi.tt - 2451545.0) / 36525.0
    L0_iter = (280.46646 + 36000.76983 * T_iter) % 360.0

    ra_der = ra_iter.hours * 15.0
    delta = L0_iter - ra_der
    if delta > 180: delta -= 360
    elif delta < -180: delta += 360

    nilai_eot.append(delta * 4.0)

# 6. Render GUI Matplotlib
plt.style.use('dark_background')
fig, ax = plt.subplots(figsize=(10, 5), dpi=100)
fig.canvas.manager.set_window_title("Grafik Equation of Time")

# Plot Garis Utama
ax.plot(hari_dalam_tahun, nilai_eot, color='#38bdf8', linewidth=2.5,
label='Kurva Perata Waktu')

# Garis Nol (Titik Temu Jam Matahari dan Arloji)
ax.axhline(0, color='#ef4444', linestyle='--', linewidth=1.5,
alpha=0.8)

# Modifikasi Label Sumbu X untuk Bulan

```

```

    bulan_labels = ['Jan', 'Feb', 'Mar', 'Apr', 'Mei', 'Jun', 'Jul',
'Ags', 'Sep', 'Okt', 'Nov', 'Des']
    hari_kumulatif = [1, 32, 60, 91, 121, 152, 182, 213, 244, 274, 305,
335]
    ax.set_xticks(hari_kumulatif)
    ax.set_xticklabels(bulan_labels, color='#cbd5e1')

    # Dekorasi Label dan Judul
    ax.set_ylabel('Equation of Time (Menit)', fontsize=12, color='white')
    ax.set_title(f'Fluktuasi Equation of Time (Perata Waktu) Tahun
{tahun_berjalan}', fontsize=14, pad=15, fontweight='bold')

    # Anotasi Ekstrem (Puncak dan Lembah)
    idx_max = np.argmax(nilai_eot)
    idx_min = np.argmin(nilai_eot)
    ax.scatter([hari_dalam_tahun[idx_max], hari_dalam_tahun[idx_min]],
                [nilai_eot[idx_max], nilai_eot[idx_min]], color='#fcd34d',
zorder=5)

    ax.annotate(f'Maks (+{nilai_eot[idx_max]:.1f}m)\nSekitar Awal Nov',
                (hari_dalam_tahun[idx_max], nilai_eot[idx_max]),
                textcoords="offset points", xytext=(-20,10), ha='center',
color='#fcd34d')

    ax.annotate(f'Min ({nilai_eot[idx_min]:.1f}m)\nSekitar Pertengahan
Feb',
                (hari_dalam_tahun[idx_min], nilai_eot[idx_min]),
                textcoords="offset points", xytext=(20,-20), ha='center',
color='#fcd34d')

    ax.grid(True, alpha=0.2)
    plt.tight_layout()
    print("[STATUS] Jendela Grafik GUI Sedang Berjalan.")
    plt.show()

if __name__ == '__main__':
    hitung_perata_waktu()

```

5.5 Penjelasan Anatomi Source Code

1. **Astrometri Presisi Tinggi (Langkah 3):** Nilai deklinasi matahari (variabel `dec.degrees`) diekstraksi menggunakan mesin vektor JPL DE421. Berbeda dengan rumus trigonometri proksi di buku falak klasik yang menggunakan pendekatan bujur ekliptika tunggal, Skyfield memasukkan faktor perturbasi dari planet-planet lain seperti Venus dan Jupiter yang memengaruhi sedikit goyangan poros bumi (nutasi).
2. **Algoritma Bujur Rata-rata Matahari - L0 (Langkah 4):** Persamaan $L_0 = 280.46646 + 36000.76983 * T$ adalah formula standar yang diadopsi dari *Astronomical Algorithms* karya Jean Meeus. Formula ini memprediksi posisi matahari hipotetis yang bergerak dengan kecepatan konstan murni di sepanjang ekuator.
3. **Komparasi Vektor Waktu (Langkah 4):** Inti dari Perata Waktu adalah komparasi asensio rekta. Nilai Asensio Rekta Sejati (`ra.hours`) yang memiliki fluktuasi mekanik murni,

dikurangi dengan L_0 (Bujur Rata-rata). Selisih sudut spasial ini kemudian dikalikan dengan konstanta 4 untuk mengubah derajat ke dalam satuan menit waktu (karena 1 derajat setara dengan 4 menit rotasi bumi).

4. **Metodologi Looping dan Render GUI (Langkah 5 & 6):** Program melakukan *looping* (perulangan) sebanyak 365 kali untuk menciptakan array titik data (`nilai_eot`). Menggunakan `matplotlib`, data matriks ini diproyeksikan menjadi kurva halus yang membentuk pola menyerupai gelombang ganda yang asimetris. Gelombang ganda ini merupakan bukti visual gabungan dari Hukum Kepler (elips) dan Obliquitas Ekliptika.

5.6 Daftar Pustaka (Pertemuan 5)

1. Ar-Razi, Fakhrudin. (1981). *Tafsir al-Fakhr ar-Razi al-Musytahar bi at-Tafsir al-Kabir wa Mafatih al-Ghaib*. Dar al-Fikr.
2. Meeus, J. (1998). *Astronomical Algorithms* (2nd ed.). Willmann-Bell, Inc. (Bab 28: *Equation of Time*).
3. Hughes, D. W., Yallop, B. D., & Hohenkerk, C. Y. (1989). *The Equation of Time*. Monthly Notices of the Royal Astronomical Society, 238(4), 1529-1535.
4. Majelis Tarjih dan Tajdid PP Muhammadiyah. (2009). *Pedoman Hisab Muhammadiyah*. Majelis Tarjih dan Tajdid PP Muhammadiyah.

PERTEMUAN 6: *Istiwa' A'zam* dan *Rashdul Qibla* (Kalibrasi Arah Global)

6.1 Teologi Kalibrasi dan Fenomena Bayangan Matahari

Dalam tradisi keilmuan Islam, bayangan bukan sekadar ketidakhadiran cahaya, melainkan sebuah instrumen kosmik yang sengaja diciptakan untuk memfasilitasi kebutuhan orientasi spasial dan temporal umat manusia. Sebelum era komputasi modern dan GPS, bayangan adalah satu-satunya indikator presisi untuk memverifikasi perhitungan matematis arah kiblat di lapangan.

Al-Qur'an secara eksplisit mengarahkan perhatian manusia pada dinamika mekanika bayangan ini:

أَلَمْ تَرَ إِلَىٰ رَبِّكَ كَيْفَ مَدَّ الظِّلَّ وَلَوْ شَاءَ لَجَعَلَهُ سَاكِنًا ثُمَّ جَعَلْنَا الشَّمْسَ عَلَيْهِ دَلِيلًا

“Apakah kamu tidak memperhatikan (penciptaan) Tuhanmu, bagaimana Dia memanjangkan bayang-bayang itu dan kalau Dia menghendaki niscaya Dia menjadikannya tetap (tidak berpindah-pindah), kemudian Kami jadikan matahari sebagai petunjuk atas bayang-bayang itu.” (QS. Al-Furqan [25]: 45)

Tafsir Ringkas: Imam Fakhruddin Ar-Razi dalam *Mafatih al-Ghaib* memberikan analisis saintifik bahwa frasa *ja'alna asy-syamsa 'alaih daliilan* (Kami jadikan matahari sebagai petunjuk) mengindikasikan korelasi matematis langsung antara posisi sudut matahari dan panjang serta arah bayangan benda. Majelis Tarjih Muhammadiyah memformulasikan ayat ini ke dalam metode *Rashdul Qibla*, di mana matahari difungsikan secara langsung sebagai "kompas langit" paling absolut, bebas dari anomali medan magnet bumi yang sering mengacaukan jarum kompas fisik.

6.2 Mekanika Astronomis *Istiwa' A'zam* (Great Culmination)

Bumi berotasi dengan sumbu yang miring 23,44 derajat terhadap ekliptika. Hal ini menyebabkan deklinasi matahari bergerak dari $-23,44^\circ$ di bulan Desember hingga $+23,44^\circ$ di bulan Juni.

Karena kota Makkah (Ka'bah) terletak pada lintang **21° 25' 21" Lintang Utara (21.4225°)**, maka secara logis, nilai deklinasi matahari akan menyamai nilai lintang Ka'bah sebanyak dua kali dalam satu tahun. Fenomena saat matahari berada tepat di atas titik zenith Makkah ini disebut ***Istiwa' A'zam*** (Kulminasi Agung) atau ***Rashdul Qibla Global***.

Pada detik fenomena ini terjadi:

1. Matahari melintas persis di atas Ka'bah.
2. Seluruh benda tegak lurus di Makkah tidak akan memiliki bayangan sama sekali.
3. **Hukum Global:** Bagi seluruh pengamat di muka bumi yang sedang mengalami siang hari dan dapat melihat matahari pada detik tersebut, arah matahari itu sendiri adalah arah kiblat

yang 100% presisi. Bayangan dari semua benda tegak lurus di bumi akan jatuh tepat membelakangi arah Ka'bah.

Peristiwa ini umumnya jatuh pada rentang:

- **27 atau 28 Mei**, sekitar pukul 12:18 Waktu Makkah (09:18 UTC / 16:18 WIB).
- **15 atau 16 Juli**, sekitar pukul 12:27 Waktu Makkah (09:27 UTC / 16:27 WIB).

6.3 Algoritma Pencarian Akar (*Root-Finding*) dalam Python

Tantangan komputasionalnya adalah: Bagaimana kita menemukan *detik yang tepat* ketika deklinasi matahari persis menyentuh angka 21.4225° ?

Metode konvensional akan menghitung posisi matahari setiap detik selama setahun penuh (*brute-force*), yang akan mengonsumsi memori besar (31,5 juta iterasi per tahun). Python melalui pustaka `Skyfield` menawarkan algoritma **Root-Finding** tingkat tinggi bernama `find_discrete`. Algoritma ini mencari titik potong silang (*zero-crossing*) dengan melompat dalam interval hari, lalu melakukan interpolasi matematis yang sangat tajam ke interval jam, menit, hingga menyempit pada fraksi detik yang eksak, menghemat 99% daya komputasi.

6.4 Source Code Mandiri: Mesin Pemindai *Rashdul Qibla* Global

Simpan skrip Python ini dengan nama `rashdul_qibla_global.py` dan jalankan di terminal. Aplikasi ini akan memindai tahun yang berjalan dan mengekstrak tanggal serta waktu eksak terjadinya fenomena Istiwa' A'zam.

```
Python
"""
Aplikasi: Mesin Pemindai Istiwa' A'zam (Rashdul Qibla Global)
Penulis: Kasmui
Deskripsi: Skrip mandiri menggunakan algoritma Root-Finding Skyfield
           untuk melacak waktu eksak matahari berada tepat di zenith
Ka'bah.
"""

import datetime
from skyfield.api import Loader, wgs84
from skyfield.searchlib import find_discrete

def lacak_rashdul_qibla_global():
    print("="*75)
    print("MESIN PEMINDAI ISTIWA' A'ZAM (GLOBAL RASHDUL QIBLA LOCATOR)")
    print("="*75)

    # 1. Inisialisasi Pustaka dan Data Ephemeris
    load = Loader('ephemeris_data', verbose=False)
    ts = load.timescale()
    planets = load('de421.bsp')
    bumi = planets['earth']
    matahari = planets['sun']
```

```

# 2. Konstanta Geografis Ka'bah
LINTANG_KABAH = 21.422487 # Derajat Desimal LU

# 3. Menentukan Jendela Waktu Pencarian (1 Tahun Penuh)
tahun_sekarang = datetime.datetime.now().year
waktu_mulai = ts.utc(tahun_sekarang, 1, 1)
waktu_akhir = ts.utc(tahun_sekarang, 12, 31)

print(f"[PROSES] Memindai Ephemeris JPL untuk Tahun
{tahun_sekarang}...")

# 4. Mendefinisikan Fungsi Keadaan (State Function) untuk Root-Finding
# Fungsi ini harus mengembalikan nilai True/1 saat Deklinasi > Lintang
Ka'bah,
# dan False/0 saat Deklinasi < Lintang Ka'bah.
def deklinasi_melewati_kabah(t):
    # Hitung posisi geosentris
    astrometrik = bumi.at(t).observe(matahari)
    _, dec, _ = astrometrik.apparent().radec()

    # Kembalikan array boolean: Apakah deklinasi matahari di utara
Ka'bah?
    return dec.degrees > LINTANG_KABAH

# Mendefinisikan resolusi lompatan pencarian awal (step_days)
# Deklinasi berubah relatif lambat, lompatan 1 hari sangat aman
deklinasi_melewati_kabah.step_days = 1.0

# 5. Eksekusi Pencarian Diskrit (find_discrete)
waktu_kejadian, status_kejadian = find_discrete(waktu_mulai,
waktu_akhir, deklinasi_melewati_kabah)

if len(waktu_kejadian) == 0:
    print("[X] Anomali: Tidak ditemukan fenomena Istiwa' A'zam pada
tahun ini.")
    return

print("\n[HASIL KOMPUTASI: WAKTU EKSAK ISTIWA' A'ZAM]")
print("Pada momen ini, bayangan setiap benda tegak lurus di bumi akan
mengarah tepat membelakangi Ka'bah.\n")

# 6. Pemrosesan dan Ekstraksi Output Waktu
for t, status in zip(waktu_kejadian, status_kejadian):
    # t adalah objek waktu Skyfield di titik perpotongan (saat dec ==
Lintang Ka'bah)
    waktu_utc = t.utc_datetime()

    # Konversi ke WIB (Waktu Indonesia Barat: UTC + 7 Jam)
    waktu_wib = waktu_utc + datetime.timedelta(hours=7)

    # Konversi ke WAS (Waktu Arab Saudi: UTC + 3 Jam)
    waktu_was = waktu_utc + datetime.timedelta(hours=3)

```

```

# Analisis Pergerakan
arah_gerak = "Bergerak ke Utara (Mei)" if status == 1 else
"Bergerak ke Selatan (Juli)"

print(f"Fenomena {arah_gerak}:")
print(f" - Waktu Universal (UTC) : {waktu_utc.strftime('%Y-%m-%d
%H:%M:%S')} UTC")
print(f" - Waktu Makkah (WAS) : {waktu_was.strftime('%Y-%m-%d
%H:%M:%S')} Waktu Arab Saudi")
print(f" - Waktu Lokal (WIB) : {waktu_wib.strftime('%Y-%m-%d
%H:%M:%S')} WIB")

# Verifikasi Ulang (Self-Check)
_, dec_cek, _ = bumi.at(t).observe(matahari).apparent().radec()
print(f" - Verifikasi Deklinasi : {dec_cek.degrees:.6f}° (Target:
{LINTANG_KABAH}°)")
print("-" * 50)

print("="*75)
print("Kalibrasi disarankan dilakukan pada +/- 5 menit dari waktu
eksak di atas.")
print("Gunakan lot/bandul (plumb bob) yang tegak lurus sempurna.")
print("="*75)

if __name__ == '__main__':
    lacak_rashdul_qibla_global()

```

6.5 Penjelasan Anatomi Source Code

1. **Pendefinisian *State Function* (Langkah 4):** Syarat wajib dari modul `find_discrete` pada pustaka *Skyfield* adalah ketersediaan fungsi target yang mengembalikan tipe data integer (0, 1, 2) atau Boolean. Fungsi `deklinasi_melewati_kabah(t)` didesain untuk mendeteksi *threshold* (batas ambang). Saat deklinasi matahari merambat naik dari ekuator pada bulan April (nilai $< 21.42^\circ$), fungsi bernilai `False`. Ketika menyentuh angka 21.4225° di akhir Mei, fungsi memicu perubahan *state* ke `True`.
2. **Variabel `step_days = 1.0`:** Ini adalah optimasi *bandwidth* CPU. Kita memerintahkan algoritma untuk memeriksa kalender setiap jam 00:00 per hari saja. Jika pada tanggal 27 Mei fungsi bernilai `False`, dan tanggal 28 Mei bernilai `True`, algoritma seketika sadar bahwa momen perpotongan berada *di antara* kedua tanggal tersebut. Mesin kemudian melakukan iterasi sub-rutin (membelah hari menjadi jam, menit, dan detik) hanya pada jeda tanggal tersebut hingga menemukan titik temu hingga kepresisian *microsecond*.
3. **Matriks `status_kejadian` (Langkah 6):** Vektor ini berisi nilai 1 (jika kurva bergerak naik/Utara) dan 0 (jika kurva bergerak turun/Selatan). Ini secara elegan membedakan peristiwa Istiwa' A'zam pertama di bulan Mei (matahari menuju titik solstis musim panas) dengan peristiwa kedua di bulan Juli (matahari turun kembali menuju ekuinoks).
4. **Verifikasi Ulang (*Self-Check*):** Baris `dec_cek.degrees` adalah bentuk validasi saintifik (*unit testing* internal). Kode memanggil ulang posisi matahari tepat pada detik hasil kalkulasi untuk membuktikan kepada pengguna bahwa deviasi antara deklinasi matahari dan Lintang Ka'bah pada detik tersebut nyaris nol (akurat hingga 6 desimal).

6.6 Daftar Pustaka (Pertemuan 6)

1. Anwar, Syamsul. (2018). *Hari-hari Allah: Dinamika Kalender Islam Kontemporer dan Menuju Kalender Hijriah Global*. Suara Muhammadiyah.
2. Ar-Razi, Fakhrudin. (1981). *Tafsir al-Fakhr ar-Razi al-Musytahar bi at-Tafsir al-Kabir wa Mafatih al-Ghaib*. Dar al-Fikr.
3. Majelis Tarjih dan Tajdid PP Muhammadiyah. (2009). *Pedoman Hisab Muhammadiyah*. Majelis Tarjih dan Tajdid PP Muhammadiyah.
4. Azhari, Susiknan. (2015). *Ilmu Falak: Perjumpaan Khazanah Islam dan Sains Modern*. Suara Muhammadiyah.
5. Rhodes, B. (2019). *Skyfield: High Precision Research-Grade Positions for Planets and Earth Satellites Explorer*. Astrophysics Source Code Library.

PERTEMUAN 7: Algoritma Presisi Waktu Salat (Bagian 1: Zuhur & Asar)

7.1 Teologi Waktu Salat dan Kewajiban Presisi Matematis

Struktur waktu dalam ajaran Islam bukanlah konstruksi yang bersifat arbitrer (acak), melainkan terikat secara absolut dengan mekanika kosmik. Pelaksanaan ibadah salat fardu memiliki batas awal dan akhir yang didefinisikan secara visual melalui fenomena astronomis matahari.

Al-Qur'an meletakkan fondasi ketetapan waktu ini secara tegas:

إِنَّ الصَّلَاةَ كَانَتْ عَلَى الْمُؤْمِنِينَ كِتَابًا مَوْقُوتًا

“*Sesungguhnya salat itu adalah fardu yang ditentukan waktunya atas orang-orang yang beriman.*” (QS. An-Nisa' [4]: 103)

Tafsir Ringkas: Imam Fakhruddin Ar-Razi di dalam *Mafatih al-Ghaib* menafsirkan frasa *kitaaban mauquutaa* sebagai ketetapan yang terikat dengan waktu-waktu spesifik yang tidak dapat dimajukan atau diakhirkan tanpa udzur syar'i. Dalam epistemologi falak Muhammadiyah melalui putusan Majelis Tarjih, kewajiban menjaga *mawaqit* (waktu-waktu) ini menuntut transisi dari observasi visual empiris (rukyat mata telanjang yang rentan terhalang cuaca) menuju *Hisab Hakiki* yang mampu memberikan kepastian komputasional yang presisi dan universal.

7.2 Mekanika Astronomis Waktu Zuhur (*Solar Transit* / *Kulminasi Atas*)

Secara fikih, awal waktu Zuhur masuk ketika matahari mulai tergelincir dari titik tengah langit (*Zawal*). Secara astronomis, momen ini terjadi tepat setelah matahari mencapai titik **Kulminasi Atas (Solar Transit)**, yaitu titik tertinggi dalam lintasan harian matahari saat memotong meridian pengamat lokal.

Pada detik kulminasi atas:

1. Azimut matahari tepat berada di arah Utara atau Selatan sejati (tergantung posisi lintang pengamat dan deklinasi matahari).
2. Bayangan benda tegak lurus (jika ada) berada pada kondisi terpendeknya untuk hari tersebut.
3. Waktu Menengah Lokal (LMT) dikoreksi secara eksak oleh nilai *Equation of Time* (Perata Waktu).

Untuk mengamankan validitas ibadah, hisab kontemporer mengaplikasikan variabel **Ikhtiyat (Waktu Pengaman)**. Majelis Tarjih menetapkan penambahan sekitar 1 hingga 2 menit dari detik matematis kulminasi atas, memastikan matahari benar-benar telah bergeser ke arah ufuk barat (*Zawal*) secara meyakinkan.

7.3 Matematika Bayangan dan Algoritma Waktu Asar

Masuknya waktu Asar tidak didasarkan pada fenomena pergeseran pusat massa matahari, melainkan pada **rasio panjang bayangan**. Fikih mendefinisikan waktu Asar masuk ketika panjang bayangan sebuah benda tegak lurus menyamai panjang benda itu sendiri, ditambah panjang bayangan benda tersebut saat waktu Zuhur (saat kulminasi).

METODE PERHITUNGAN WAKTU ASAR

Secara matematis, tinggi matahari (h) saat Asar dikalkulasi menggunakan invers fungsi tangen (arc-cotangen) dari rasio tersebut. Terdapat dua kriteria rasio bayangan (Faktor f):

1. **Jumhur Ulama (Termasuk Tarjih Muhammadiyah):** Faktor $f = 1$. (Bayangan = 1x Benda + Bayangan Zuhur).
2. **Mazhab Hanafi:** Faktor $f = 2$. (Bayangan = 2x Benda + Bayangan Zuhur).

1. Menentukan Jarak Zenith (Zd) Jarak Zenith (Zd) saat siang hari dihitung dari selisih mutlak antara Lintang Pengamat (ϕ) dan Deklinasi Matahari (δ).

Rumus: $Zd = \text{abs}(\text{Lintang Pengamat} - \text{Deklinasi Matahari})$

2. Menentukan Tinggi Matahari Asar (h_{asar}) Tinggi matahari saat awal waktu Asar (h_{asar}) didapatkan melalui persamaan:

Rumus: $h_{\text{asar}} = \text{atan}(1 / (f + \tan(Zd)))$

Setelah tinggi matahari target (h_{asar}) diketahui, kita mencari Waktu Sudut Jam (Hour Angle / H) yang merupakan jarak sudut rotasi bumi dari titik meridian menuju posisi Asar, kemudian dikonversi menjadi satuan waktu.

7.4 Source Code Mandiri: Generator Waktu Zuhur dan Asar

Simpan skrip Python ini dengan nama `waktu_salat_siang.py` dan jalankan di terminal. Aplikasi ini mengawinkan data ephemeris absolut JPL dengan rumus trigonometri sferis klasik untuk menghasilkan akurasi tingkat tinggi.

```
Python
"""
Aplikasi: Generator Waktu Salat Hakiki (Bagian 1: Zuhur & Asar)
Penulis: Kasmui
Deskripsi: Skrip mandiri untuk menghitung waktu Zawal (Zuhur) dan
           kalkulasi panjang bayangan (Asar) menggunakan ephemeris NASA
           JPL,
           dilengkapi dengan waktu pengaman (Ikhtiyat).
"""

import datetime
import math
import pytz
```

```

from skyfield.api import Loader, wgs84
from skyfield import almanac

def hitung_zuhur_asar():
    print("="*75)
    print("MESIN KOMPUTASI WAKTU SALAT (ZUHUR & ASAR)")
    print("="*75)

    # 1. Inisialisasi Pustaka dan Ephemeris
    load = Loader('ephemeris_data', verbose=False)
    ts = load.timescale()
    planets = load('de421.bsp')
    bumi = planets['earth']
    matahari = planets['sun']

    # 2. Parameter Lokasi Pengamat (Contoh: Yogyakarta)
    nama_lokasi = "Yogyakarta"
    lintang = -7.7956      # LU positif, LS negatif
    bujur = 110.3695      # BT positif, BB negatif
    elevasi = 113         # meter di atas permukaan laut
    zona_waktu = pytz.timezone('Asia/Jakarta')
    lokasi_pengamat = wgs84.latlon(lintang, bujur, elevation_m=elevasi)

    # 3. Rentang Waktu Komputasi (Hari ini)
    sekarang = datetime.datetime.now(zona_waktu)
    awal_hari = zona_waktu.localize(datetime.datetime(sekarang.year,
sekarang.month, sekarang.day, 0, 0, 0))
    akhir_hari = awal_hari + datetime.timedelta(days=1)

    t0 = ts.from_datetime(awal_hari)
    t1 = ts.from_datetime(akhir_hari)

    print(f"[DATA LOKASI] : {nama_lokasi} (Lintang: {lintang}°, Bujur:
{bujur}°)")
    print(f"[TANGGAL]      : {awal_hari.strftime('%d %B %Y')}")
    print("-" * 75)

    # =====
    # KOMPUTASI WAKTU ZUHUR (TRANSIT MERIDIAN)
    # =====

    # Mencari waktu kulminasi (Matahari melintasi meridian pengamat)
    f_transit = almanac.meridian_transits(planets, matahari,
lokasi_pengamat)
    waktu_transit, jenis_transit = almanac.find_discrete(t0, t1,
f_transit)

    # jenis_transit == 1 artinya Kulminasi Atas (Siang)
    idx_siang = list(jenis_transit).index(1)
    t_zuhur_astronomis = waktu_transit[idx_siang]

    # Menerapkan Ikhtiyat (Waktu Pengaman 2 Menit)
    ikhtiyat_menit = 2

```

```

    waktu_zuhur_wib =
t_zuhur_astronomis.utctime().replace(tzinfo=pytz.utc).astimezone(zona
_waktu)
    waktu_zuhur_ikhtiyat = waktu_zuhur_wib +
datetime.timedelta(minutes=ikhtiyat_menit)

    print("\n[1] KOMPUTASI ZUHUR (ZAWAL)")
    print(f"Kulminasi Astronomis (Murni) :
{waktu_zuhur_wib.strftime('%H:%M:%S')} WIB")
    print(f"Waktu Zuhur (+ Ikhtiyat 2m) :
{waktu_zuhur_ikhtiyat.strftime('%H:%M:%S')} WIB")

    # =====
    # KOMPUTASI WAKTU ASAR (RASIO BAYANGAN)
    # =====

    # a) Menghitung Jarak Zenith Siang (Zd) saat Kulminasi
    # Ekstraksi posisi matahari saat Zuhur
    astrometrik_zuhur = bumi.at(t_zuhur_astronomis).observe(matahari)
    _, dec_zuhur, _ = astrometrik_zuhur.apparent().radec()
    deklinasi_derajat = dec_zuhur.degrees

    # Zd = |Lintang - Deklinasi|
    zenith_distance = abs(lintang - deklinasi_derajat)

    # b) Menghitung Tinggi Matahari Target untuk Asar (Jumhur Ulama, Rasio
= 1)
    faktor_asar = 1.0
    # Rumus: h_asar = atan(1 / (faktor + tan(Zd)))
    zd_radian = math.radians(zenith_distance)
    h_asar_radian = math.atan(1.0 / (faktor_asar + math.tan(zd_radian)))
    h_asar_derajat = math.degrees(h_asar_radian)

    # c) Menghitung Waktu Sudut Jam (Hour Angle) Asar
    # cos(H) = (sin(h) - sin(lintang)*sin(deklinasi)) /
(cos(lintang)*cos(deklinasi))
    lat_rad = math.radians(lintang)
    dec_rad = math.radians(deklinasi_derajat)

    pembilang = math.sin(h_asar_radian) - (math.sin(lat_rad) *
math.sin(dec_rad))
    penyebut = math.cos(lat_rad) * math.cos(dec_rad)
    cos_H = pembilang / penyebut

    # Validasi jika nilai di luar domain acos (Fenomena Matahari Malam
Midnight Sun)
    if cos_H < -1 or cos_H > 1:
        print("\n[X] Anomali Lintang Ekstrem: Matahari tidak pernah
mencapai tinggi Asar pada hari ini.")
        return

    # Hitung Sudut Jam H dalam radian, lalu konversi ke derajat
    H_radian = math.acos(cos_H)
    H_derajat = math.degrees(H_radian)

```

```

# Konversi Sudut Jam ke waktu nyata (1 derajat = 4 menit = 240 detik)
tambahan_detik = H_derajat * 240.0

# Asar = Waktu Zuhur Astronomis + Sudut Jam Waktu
waktu_asar_astronomis = waktu_zuhur_wib +
datetime.timedelta(seconds=tambahan_detik)
waktu_asar_ikhtiyat = waktu_asar_astronomis +
datetime.timedelta(minutes=ikhtiyat_menit)

print("\n[2] KOMPUTASI ASAR")
print(f"Jarak Zenith Kulminasi (Zd) : {zenith_distance:.4f}°")
print(f"Target Ketinggian Matahari : {h_asar_derajat:.4f}° dari
ufuk")
print(f"Waktu Asar Astronomis Murni :
{waktu_asar_astronomis.strftime('%H:%M:%S')} WIB")
print(f"Waktu Asar (+ Ikhtiyat 2m) :
{waktu_asar_ikhtiyat.strftime('%H:%M:%S')} WIB")
print("="*75)

if __name__ == '__main__':
    hitung_zuhur_asar()

```

7.5 Penjelasan Anatomi Source Code

1. **Fungsi meridian_transits (Zuhur):** Berbeda dengan metode hisab klasik yang harus menghitung selisih Perata Waktu (EoT) secara manual, modul `almanac` dari `Skyfield` melakukan inspeksi vektor ruang secara *real-time*. Fungsi ini secara otomatis menemukan detik absolut ketika pusat matahari berpotongan persis dengan meridian bujur pengamat.
2. **Kalkulasi Jarak Zenith (Langkah Asar - a):** Posisi matahari bervariasi setiap harinya akibat deklinasi. Jarak Zenith (Z_d) mengukur "seberapa miring" matahari saat tepat berada di puncak siang. Nilai ini sangat krusial karena ia menentukan panjang "bayangan minimum" (bayangan zawal) yang harus ditambahkan ke ukuran benda target.
3. **Eksekusi Trigonometri Sferis (Langkah Asar - c):** Python mengonversi seluruh variabel derajat menjadi Radian sebelum dieksekusi oleh fungsi `math.sin` dan `math.cos`. Persamaan Waktu Sudut Jam (*Hour Angle / H*) secara elegan memecahkan masalah ini dengan menemukan jarak spasial matahari dari titik kulminasinya menuju posisi Asar, yang kemudian dikalikan 240 detik untuk memperoleh rentang waktu absolut dalam hitungan jam dan menit.
4. **Validasi Midnight Sun (Proteksi Algoritma):** Baris penahan `if cos_H < -1 or cos_H > 1` adalah protokol keamanan (*error handling*). Di lintang tinggi (mendekati kutub) pada musim dingin, matahari bisa jadi tidak pernah terbit cukup tinggi untuk mencapai syarat bayangan waktu Asar, sehingga perhitungan *arc-cosine* akan gagal (*Math Domain Error*). Blok ini mencegah aplikasi *crash* dan memberikan pesan ilmiah yang tepat.

7.6 Daftar Pustaka (Pertemuan 7)

1. Anwar, Syamsul. (2018). *Hari-hari Allah: Dinamika Kalender Islam Kontemporer dan Menuju Kalender Hijriah Global*. Suara Muhammadiyah.

2. Ar-Razi, Fakhruddin. (1981). *Tafsir al-Fakhr ar-Razi al-Musytahar bi at-Tafsir al-Kabir wa Mafatih al-Ghaib*. Dar al-Fikr.
3. Majelis Tarjih dan Tajdid PP Muhammadiyah. (2009). *Pedoman Hisab Muhammadiyah*. Majelis Tarjih dan Tajdid PP Muhammadiyah.
4. Meeus, J. (1998). *Astronomical Algorithms* (2nd ed.). Willmann-Bell, Inc. (Bab 15: *Atmospheric Refraction* dan Bab 28: *Equation of Time*).
5. Saksono, Tono. (2007). *Mengkompromikan Rukyat & Hisab*. Amythas Publicita.

PERTEMUAN 8: Ujian Tengah Semester (UTS) – Sintesis Modul Spasial dan Waktu Astronomis

8.1 Filosofi *Itqan* (Profesionalisme) dalam Komputasi Falak

Ujian Tengah Semester dalam kerangka epistemologi Islam bukanlah sekadar evaluasi kognitif, melainkan momen *muhasabah* (introspeksi) terhadap sejauh mana kita mampu menerapkan prinsip *Itqan* (kualitas yang terkalibrasi dengan sempurna). Dalam pengembangan sistem astronomi berskala besar, menyatukan berbagai modul terpisah menjadi satu mesin cerdas yang kohesif adalah puncak dari integrasi ilmu ukur dan logika algoritmik.

Allah Subhanahu wa Ta'ala berfirman mengenai arsitektur penciptaan yang terintegrasi secara sempurna:

صُنِعَ اللَّهُ الَّذِي أَنْقَنَ كُلَّ شَيْءٍ إِنَّهُ خَبِيرٌ بِمَا تَفْعَلُونَ

“(Begitulah) perbuatan Allah yang membuat dengan kokoh (*mutqin*) tiap-tiap sesuatu; sesungguhnya Allah Maha Mengetahui apa yang kamu kerjakan.” (QS. An-Naml [27]: 88)

Tafsir Ringkas: Imam Fakhruddin Ar-Razi menafsirkan kata *atqana* (أَنْقَنَ) sebagai perancangan yang terstruktur, di mana tidak ada satu pun komponen alam semesta yang bertabrakan fungsinya dengan komponen lain. Dalam paradigma pemrograman berorientasi objek (*Object-Oriented Programming / OOP*), sifat *mutqin* ini diterjemahkan ke dalam arsitektur kode yang *modular*, efisien, dan bebas dari redundansi (pengulangan komputasi yang tidak perlu).

8.2 Arsitektur Integrasi Sistem (Proyek UTS)

Pada pertemuan 1 hingga 7, komputasi arah kiblat dan waktu salat dieksekusi melalui skrip yang berdiri sendiri (*standalone*). Kelemahan dari skrip terpisah adalah beban memori (RAM); jika setiap skrip memuat data ephemeris JPL DE421 secara independen, program akan berjalan lambat.

Tantangan komprehensif pada fase UTS ini adalah: **Membangun satu Engine Falak Terpadu berbasis Terminal Interaktif**. Mesin ini harus mampu:

1. Memuat Ephemeris JPL NASA hanya satu kali di memori.
2. Menerima input koordinat geospasial (WGS84) secara dinamis dari pengguna.
3. Menghitung azimuth arah kiblat berdasarkan lintang dan bujur tersebut.
4. Menghitung waktu kulminasi (Zuhur) dan rasio bayangan 1x (Asar) berpedoman pada ketetapan fisis Majelis Tarjih, secara instan pada hari yang sama.

Arsitektur ini merupakan embrio dari mesin komputasi yang lebih masif, sebagaimana yang diterapkan pada arsitektur penyusunan Kalender Hijriah Global Tunggal (KHGT), di mana efisiensi pemanggilan data lintas koordinat menjadi kunci performa aplikasi.

8.3 Source Code Mandiri: Proyek UTS (Sistem Falak Terpadu)

Simpan skrip Python ini dengan nama `uts_sistem_falak_terpadu.py`. Skrip ini mengimplementasikan paradigma OOP (Class) untuk mengelola data astronomis agar eksekusi komputasi berjalan secepat kilat.

```
Python
"""
Aplikasi: Proyek UTS - Sistem Komputasi Falak Terpadu (Terminal
Interactive)
Penulis: Kasmui
Deskripsi: Mesin komputasi berbasis Class (OOP) yang mengintegrasikan
          Geodesi Sferis (Arah Kiblat) dan Kinematika Matahari (Zuhur &
Asar)
          dalam satu antarmuka terminal interaktif.
"""

import math
import datetime
import pytz
from skyfield.api import Loader, wgs84
from skyfield import almanac

class EngineFalakTerpadu:
    def __init__(self):
        # Memuat memori HANYA SATU KALI saat aplikasi dijalankan (Prinsip
        Itqan/Efisiensi)
        print("[SISTEM] Memuat matriks Ephemeris JPL NASA (DE421)...")
        self.load = Loader('ephemeris_data', verbose=False)
        self.ts = self.load.timescale()
        self.planets = self.load('de421.bsp')
        self.bumi = self.planets['earth']
        self.matahari = self.planets['sun']

        # Konstanta absolut
        self.LINTANG_KABAH = 21.422487
        self.BUJUR_KABAH = 39.826206
        print("[SISTEM] Engine siap digunakan.\n")

    def hitung_kiblat(self, lat, lon):
        """Metode Geodesi Sferis menggunakan Trigonometri Bola"""
        phi_k = math.radians(self.LINTANG_KABAH)
        lam_k = math.radians(self.BUJUR_KABAH)
        phi_l = math.radians(lat)
        lam_l = math.radians(lon)

        delta_lam = lam_k - lam_l
        y = math.sin(delta_lam)
```

```

    x = (math.cos(phi_l) * math.tan(phi_k)) - (math.sin(phi_l) *
math.cos(delta_lam))

    azimuth_radian = math.atan2(y, x)
    azimuth_derajat = math.degrees(azimuth_radian)

    if azimuth_derajat < 0:
        azimuth_derajat += 360.0

    return azimuth_derajat

def hitung_salat_siang(self, lat, lon, elevasi, tz_string):
    """Metode Ekstraksi Waktu Zawal dan Bayangan Asar"""
    zona_waktu = pytz.timezone(tz_string)
    lokasi_pengamat = wgs84.latlon(lat, lon, elevation_m=elevasi)
    sekarang = datetime.datetime.now(zona_waktu)

    # Rentang komputasi hari ini
    awal_hari = zona_waktu.localize(datetime.datetime(sekarang.year,
sekarang.month, sekarang.day, 0, 0, 0))
    akhir_hari = awal_hari + datetime.timedelta(days=1)
    t0 = self.ts.from_datetime(awal_hari)
    t1 = self.ts.from_datetime(akhir_hari)

    # 1. Komputasi Zuhur (Kulminasi)
    f_transit = almanac.meridian_transits(self.planets, self.matahari,
lokasi_pengamat)
    waktu_transit, jenis_transit = almanac.find_discrete(t0, t1,
f_transit)
    idx_siang = list(jenis_transit).index(1)
    t_zuhur_astronomis = waktu_transit[idx_siang]
    waktu_zuhur_ikhtiyat =
t_zuhur_astronomis.utc_datetime().replace(tzinfo=pytz.utc).astimezone(zona
_waktu) + datetime.timedelta(minutes=2)

    # 2. Komputasi Asar
    astrometrik_zuhur =
self.bumi.at(t_zuhur_astronomis).observe(self.matahari)
    _, dec_zuhur, _ = astrometrik_zuhur.apparent().radec()

    zenith_distance = abs(lat - dec_zuhur.degrees)
    zd_radian = math.radians(zenith_distance)

    # Faktor 1 untuk pandangan Jumhur Ulama
    h_asar_radian = math.atan(1.0 / (1.0 + math.tan(zd_radian)))

    lat_rad = math.radians(lat)
    dec_rad = math.radians(dec_zuhur.degrees)
    cos_H = (math.sin(h_asar_radian) - (math.sin(lat_rad) *
math.sin(dec_rad))) / (math.cos(lat_rad) * math.cos(dec_rad))

    if cos_H < -1 or cos_H > 1:
        return waktu_zuhur_ikhtiyat, None # Anomali

```

```

H_derajat = math.degrees(math.acos(cos_H))
tambahan_detik = H_derajat * 240.0

waktu_asar_ikhtiyat =
t_zuhur_astronomis.utctime().replace(tzinfo=pytz.utc).astimezone(zona
_waktu) + datetime.timedelta(seconds=tambahan_detik, minutes=2)

return waktu_zuhur_ikhtiyat, waktu_asar_ikhtiyat

def eksekusi_terminal():
    print("="*60)
    print("UJIAN TENGAH SEMESTER: SISTEM FALAK TERPADU")
    print("="*60)

    # Inisialisasi Engine (Berjalan di latar belakang)
    mesin = EngineFalakTerpadu()

    # Interaksi Pengguna
    print("[INPUT DATA LOKASI]")
    try:
        nama_kota = input("Masukkan Nama Kota/Wilayah : ")
        lat = float(input("Masukkan Lintang (Desimal) : "))
        lon = float(input("Masukkan Bujur (Desimal) : "))
        elv = float(input("Masukkan Elevasi (mdpl) : "))
        tz = input("Masukkan Zona Waktu (Contoh: Asia/Jakarta) : ")

        if not tz.strip():
            tz = 'Asia/Jakarta' # Default fallback

    except ValueError:
        print("[X] Kesalahan Input: Pastikan koordinat berupa angka
desimal.")
        return

    print("\n" + "="*60)
    print(f"LAPORAN ASTRONOMIS: {nama_kota.upper()}")
    print("="*60)

    # Menjalankan fungsi kalkulasi dari Engine
    azimuth_kiblat = mesin.hitung_kiblat(lat, lon)
    zuhur, asar = mesin.hitung_salat_siang(lat, lon, elv, tz)

    # Menampilkan Hasil
    print(f"1. Arah Kiblat Geografis : {azimuth_kiblat:.4f}° dari Utara
Sejati")
    print(f"2. Jadwal Salat Siang Hari (Tanggal: {zuhur.strftime('%d-%b-
%Y')})")
    print(f" - Waktu Zuhur (+2m) : {zuhur.strftime('%H:%M:%S')}
{tz.split('/')[-1]}")

    if asar:
        print(f" - Waktu Asar (+2m) : {asar.strftime('%H:%M:%S')}
{tz.split('/')[-1]}")
    else:

```

```

        print("    - Waktu Asar          : Tidak tercapai (Matahari tidak
        terbenam/terbit pada rasio Asar)")

    print("="*60)
    print("STATUS: Kompilasi dan Eksekusi UTS Selesai.")
    print("="*60)

if __name__ == '__main__':
    eksekusi_terminal()

```

8.4 Penjelasan Anatomi Source Code

1. **Arsitektur Class EngineFalakTerpadu:** Pendekatan prosedural (fungsi biasa) yang digunakan pada pertemuan sebelumnya diubah menjadi struktur *Class*. Fungsi `__init__` bertindak sebagai konstruktor yang dieksekusi pertama kali. Modul `skyfield`, yang memakan waktu sepersekian detik untuk membaca file biner dari *hard-disk*, hanya dipanggil satu kali pada fase ini dan disimpan ke dalam properti *instance* (`self.ts`, `self.planets`).
2. **Isolasi Metode (*Method Isolation*):** Terdapat pemisahan tegas antara logika spasial ruang (`hitung_kiblat`) dan logika temporal waktu (`hitung_salat_siang`). Pemisahan ini memungkinkan antarmuka yang lebih kompleks (seperti pengembangan GUI menggunakan `CustomTkinter` di masa depan) untuk memanggil fungsi-fungsi spesifik ini secara asinkron tanpa harus me-reset memori aplikasi dari awal.
3. **Terminal Interaktif (`input()`):** Program kini tidak lagi *hard-coded* (nilai lintang dan bujur diketik mati di dalam skrip). Menggunakan fungsi `input()`, skrip ini bermetamorfosis menjadi perangkat lunak utilitas universal. Program membungkus interaksi ini dengan blok `try...except ValueError` untuk memastikan aplikasi tidak *crash* apabila pengguna tidak sengaja memasukkan teks alfabet pada kolom isian angka.
4. **Integrasi Variabel Global:** Hasil dari fungsi `meridian_transits` (detik absolut kulminasi atas) yang secara historis hanya dipakai untuk waktu Zuhur, langsung disalurkan (*piping*) ke dalam rumus trigonometri penentu rasio bayangan Asar. Keberhasilan menyatukan alur data ini merupakan parameter utama dalam penilaian evaluasi UTS ini.

8.5 Daftar Pustaka (Pertemuan 8)

1. Ar-Razi, Fakhruddin. (1981). *Tafsir al-Fakhr ar-Razi al-Musyatar bi at-Tafsir al-Kabir wa Mafatih al-Ghaib*. Dar al-Fikr.
2. Downey, A. (2015). *Think Python: How to Think Like a Computer Scientist* (2nd ed.). O'Reilly Media, Inc. (Bab 15-17: *Classes and Objects*).
3. Majelis Tarjih dan Tajdid PP Muhammadiyah. (2009). *Pedoman Hisab Muhammadiyah*. Majelis Tarjih dan Tajdid PP Muhammadiyah.
4. Rhodes, B. (2019). *Skyfield: High Precision Research-Grade Positions for Planets and Earth Satellites Explorer*. Astrophysics Source Code Library.

PERTEMUAN 9: Algoritma Presisi Waktu Salat (Bagian 2: Maghrib, Isya', Subuh)

9.1 Teologi Malam dan Fenomena *Fajar Shadiq*

Setelah matahari melewati meridian lokal dan turun ke ufuk barat, spektrum cahaya yang diterima pengamat di bumi mengalami perubahan drastis akibat hamburan atmosferik (*atmospheric scattering*). Perubahan fisis ini menjadi penanda bagi tiga waktu salat krusial: Maghrib, Isya', dan Subuh.

Al-Qur'an memberikan garis demarkasi astronomis yang sangat jelas mengenai siklus kegelapan dan cahaya ini:

أَقِمِ الصَّلَاةَ لِذُلُوكِ الشَّمْسِ إِلَى غَسَقِ اللَّيْلِ وَقُرْءَانَ الْفَجْرِ إِذَا قُرْءَانَ الْفَجْرِ كَانَ مَشْهُودًا

"Dirikanlah salat dari sesudah matahari tergelincir sampai gelap malam dan (dirikanlah pula salat) subuh. Sesungguhnya salat subuh itu disaksikan (oleh malaikat)." (QS. Al-Isra' [17]: 78)

Tafsir Ringkas: Imam Fakhruddin Ar-Razi dalam *Mafatih al-Ghaib* memberikan analisis yang sangat presisi mengenai ayat ini. Kata *ghasaq al-layl* ditafsirkan sebagai puncak kegelapan malam, yang secara fisis terjadi ketika seluruh pendaran cahaya matahari (syafaq) benar-benar lenyap dari cakrawala, menandai masuknya waktu Isya'. Sedangkan *fajr* (fajar) dibedakan menjadi dua: fajar *kadzib* (zodiakal light) yang vertikal dan semu, serta fajar *shadiq* yang menyebar secara horizontal di ufuk timur, yang menjadi syarat mutlak masuknya waktu Subuh.

9.2 Fisika Atmosferik dan Anomali Refraksi (Maghrib)

Waktu Maghrib dimulai tepat ketika piringan atas matahari menghilang di bawah ufuk barat (Sunset). Namun, kalkulasi ini tidak sesederhana menghitung posisi matahari secara geometris. Terdapat fenomena **Pembiasan Cahaya (Refraksi Atmosfer)**.

Atmosfer bumi bertindak seperti lensa cembung raksasa. Ketika matahari secara fisik (geometris) sudah tenggelam di bawah ufuk, cahayanya dibelokkan ke atas oleh atmosfer sehingga mata kita masih "melihat" matahari tersebut.

- Besaran refraksi di ufuk secara standar adalah sekitar 34 menit busur.
- Ukuran jari-jari (semi-diameter) piringan matahari adalah sekitar 16 menit busur.
- Kerendahan ufuk (dip) akibat ketinggian pengamat (elevasi) juga menambah waktu tampak matahari.

Jika dijumlahkan ($34' + 16' = 50'$ busur), piringan atas matahari menyentuh ufuk *tampak* (apparent) ketika pusat piringan matahari secara geometris berada pada sudut $h = -0.833$ derajat di bawah ufuk. Namun, nilai refraksi ini sangat fluktuatif bergantung pada **Suhu Udara** dan **Tekanan**

Barometrik lokal. Suhu dingin dan tekanan tinggi akan membengkokkan cahaya lebih ekstrem, membuat Maghrib masuk lebih lambat dari jadwal standar.

9.3 Parameter Ketinggian Matahari untuk Isya' dan Subuh

Berbeda dengan Maghrib yang berpatokan pada piringan fisik matahari, Isya' dan Subuh bergantung pada cahaya bias di lapisan atas ionosfer.

- **Isya'**: Masuk saat hilangnya mega merah (*syafaq ahmar*). Astronomi modern mengklasifikasikan ini sebagai akhir dari *Astronomical Twilight*, yang terjadi ketika posisi pusat matahari mencapai sudut depresi **-18 derajat**.
- **Subuh**: Masuk saat terbitnya fajar shadiq. Melalui observasi empiris yang ketat (menggunakan *Sky Quality Meter*), Majelis Tarjih Muhammadiyah pada Munas ke-31 tahun 2020 telah mengeluarkan putusan koreksi, mengubah parameter awal waktu Subuh dari yang sebelumnya -20 derajat, menjadi **-18 derajat**. Kesepakatan parameter -18 derajat untuk Isya' dan Subuh ini mencerminkan sinkronisasi harmonis antara teks syar'i dan validitas sains observasional modern.

9.4 Source Code Mandiri: Kalkulator Dinamis Salat Gelap

Simpan skrip Python ini dengan nama `waktu_salat_gelap.py` dan jalankan di terminal. Aplikasi ini tidak menggunakan fungsi standar bawaan, melainkan membangun fungsi pemindai dinamis (custom root-finding) yang mampu menginjeksi variabel cuaca riil dari pengguna untuk akurasi Maghrib tingkat tinggi.

Python

```
"""
Aplikasi: Kalkulator Dinamis Waktu Salat Gelap (Maghrib, Isya', Subuh)
Penulis: Kasmui
Deskripsi: Skrip mandiri menggunakan algoritma Root-Finding Skyfield
           yang menerima parameter intervensi Cuaca (Suhu & Tekanan)
           serta mengadopsi ketentuan Tarjih (-18 Derajat untuk
Subuh/Isya).
"""

import datetime
import pytz
from skyfield.api import Loader, wgs84
from skyfield.searchlib import find_discrete

def komputasi_salat_gelap():
    print("="*75)
    print("MESIN KOMPUTASI SALAT GELAP: MAGHRIB, ISYA', SUBUH")
    print("="*75)

    # 1. Inisialisasi Ekosistem Skyfield
    load = Loader('ephemeris_data', verbose=False)
    ts = load.timescale()
    planets = load('de421.bsp')
    bumi = planets['earth']
    matahari = planets['sun']
```

```

# 2. Input Parameter Lokal & Cuaca
nama_lokasi = "Malang, Jawa Timur"
lintang = -7.9797
bujur = 112.6304
elevasi = 450.0 # meter
zona_waktu = pytz.timezone('Asia/Jakarta')

# Injeksi Cuaca (Mempengaruhi Refraksi Maghrib)
suhu_celcius = 22.0 # Suhu aktual
tekanan_mbar = 1015.0 # Tekanan Barometrik aktual

print(f"[LOKASI] : {nama_lokasi} (Elevasi: {elevasi} mdpl)")
print(f"[CUACA] : Suhu {suhu_celcius}°C, Tekanan {tekanan_mbar}
mBar")

lokasi_pengamat = wgs84.latlon(lintang, bujur, elevation_m=elevasi)

# 3. Pengaturan Rentang Waktu Pencarian (Hari ini hingga besok pagi)
sekarang = datetime.datetime.now(zona_waktu)
t0_dt = zona_waktu.localize(datetime.datetime(sekarang.year,
sekarang.month, sekarang.day, 12, 0, 0))
t1_dt = t0_dt + datetime.timedelta(hours=24) # Cari dari jam 12 siang
hari ini ke jam 12 besok

t0 = ts.from_datetime(t0_dt)
t1 = ts.from_datetime(t1_dt)

# =====
# FUNGSI PEMINDAI 1: MAGHRIB (Refraksi Dinamis Cuaca)
# =====
def cek_maghrib(t):
    """Mencari titik saat piringan atas matahari menyentuh ufuk"""
    # Hitung altaz dengan intervensi suhu dan tekanan udara
    alt, _, _ =
lokasi_pengamat.at(t).observe(matahari).apparent().altaz(
    temperature_C=suhu_celcius, pressure_mbar=tekanan_mbar)

    # Piringan atas di 0 derajat = pusat matahari di -0.2666 derajat
    (apparent)
    return alt.degrees > -0.2666

cek_maghrib.step_days = 0.05 # Lompatan observasi yang rapat

waktu_maghrib, _ = find_discrete(t0, t1, cek_maghrib)

# Ambil kejadian pertama saat matahari *turun* melewati batas (Sunset)
# find_discrete mengembalikan kejadian saat fungsi berubah nilai
# Karena cek_maghrib bernilai True saat siang, dan False saat malam,
# transisi pertama dari t0 (siang) adalah Sunset.
if len(waktu_maghrib) > 0:
    t_maghrib = waktu_maghrib[0]
    maghrib_wib =
t_maghrib.utc_datetime().replace(tzinfo=pytz.utc).astimezone(zona_waktu)

```

```

    # Tambahkan Ikhtiyat 2 menit
    maghrib_wib_ikhtiyat = maghrib_wib + datetime.timedelta(minutes=2)
else:
    maghrib_wib_ikhtiyat = None

# =====
# FUNGSI PEMINDAI 2: ISYA' & SUBUH (Kriteria Tarjih -18 Derajat)
# =====
def cek_fajar_syafaq(t):
    """Mencari titik saat pusat matahari menembus -18 derajat
    (Geometrik)"""
    # Pada sudut yang sangat rendah, refraksi tidak memengaruhi
    visibilitas fajar
    alt, _, _ =
    lokasi_pengamat.at(t).observe(matahari).apparent().altaz()
    return alt.degrees > -18.0

cek_fajar_syafaq.step_days = 0.05

waktu_18, status_18 = find_discrete(t0, t1, cek_fajar_syafaq)

isya_wib_ikhtiyat = None
subuh_wib_ikhtiyat = None

# Proses array transisi -18 derajat
for t_obj, status in zip(waktu_18, status_18):
    dt_lokal =
    t_obj.utc_datetime().replace(tzinfo=pytz.utc).astimezone(zona_waktu)
    if status == 0:
        # Matahari turun melewati -18 derajat (Akhir Syafaq = Isya')
        isya_wib_ikhtiyat = dt_lokal + datetime.timedelta(minutes=2)
    elif status == 1:
        # Matahari naik melewati -18 derajat (Awal Fajar = Subuh)
        subuh_wib_ikhtiyat = dt_lokal + datetime.timedelta(minutes=2)

# =====
# HASIL OUTPUT
# =====
print("\n[HASIL KOMPUTASI WAKTU SALAT (Telah ditambah Ikhtiyat 2
Menit)]")
if maghrib_wib_ikhtiyat:
    print(f"1. MAGHRIB (Sunset Dinamis) :
{maghrib_wib_ikhtiyat.strftime('%H:%M:%S')} {zona_waktu.zone}")

    if isya_wib_ikhtiyat:
        print(f"2. ISYA' (Syafaq -18°) :
{isya_wib_ikhtiyat.strftime('%H:%M:%S')} {zona_waktu.zone}")

    if subuh_wib_ikhtiyat:
        print(f"3. SUBUH (Fajar -18°) :
{subuh_wib_ikhtiyat.strftime('%H:%M:%S')} {zona_waktu.zone} (Hari
Berikutnya)")

print("="*75)

```

```

print("Catatan: Perhitungan telah menggunakan parameter Munas Tarjih
ke-31 (-18 Derajat).")
print("="*75)

if __name__ == '__main__':
    komputasi_salat_gelap()

```

9.5 Penjelasan Anatomi Source Code

1. **Injeksi Cuaca Dinamis (Langkah 2):** Tidak ada buku komputasi falak klasik yang menyertakan suhu udara *real-time* ke dalam hitungannya. Dengan memanfaatkan metode `apparent().altaz(temperature_C=suhu_celcius, pressure_mbar=tekanan_mbar)`, algoritma ini mengkalkulasi ketebalan dan indeks bias atmosfer persis sesuai dengan kondisi cuaca riil di lokasi pengguna saat kode dijalankan. Di dataran tinggi yang dingin (seperti Malang atau Dieng), refraksinya berbeda dengan pesisir pantai yang panas, membuat jadwal Maghrib lebih adaptif dan syar'i.
2. **Target Apparent Maghrib (`alt.degrees > -0.2666`):** Kita tidak mencari posisi pusat matahari di angka 0 derajat, melainkan memposisikan piringan atasnya. Jari-jari matahari adalah 16 menit busur (atau 0.2666 derajat desimal). Dengan menargetkan pusat matahari berada di posisi *apparent* -0.2666° , itu berarti secara visual piringan atas matahari persis sejajar dengan garis ufuk yang terbentang.
3. **Algoritma Transisi Boolean (`status_18`):** Fungsi `find_discrete` dari Skyfield mengembalikan dua array: waktu kejadian (`waktu_18`) dan arah transisi (`status_18`). Status 0 berarti fungsi berubah dari `True` (ketinggian $> -18^\circ$) menjadi `False` (ketinggian $< -18^\circ$), yang menandakan matahari sedang turun (Isya'). Sebaliknya, status 1 menandakan matahari sedang merambat naik dari kegelapan (Subuh). Logika saklar ini menghapus kebutuhan untuk menggunakan dua rumus trigonometri yang terpisah.
4. **Rentang Waktu Pencarian (T0 ke T1):** Program menggeser titik awal pencarian (T0) bukan dari jam 00:00 pagi, melainkan dari jam 12:00 siang. Hal ini dilakukan agar mesin secara kronologis menemukan Maghrib lebih dulu, kemudian Isya', dan akhirnya Subuh pada keesokan harinya, merepresentasikan satu siklus pergantian malam secara utuh.

9.6 Daftar Pustaka (Pertemuan 9)

1. Anwar, Syamsul. (2018). *Hari-hari Allah: Dinamika Kalender Islam Kontemporer dan Menuju Kalender Hijriah Global*. Suara Muhammadiyah.
2. Ar-Razi, Fakhrudin. (1981). *Tafsir al-Fakhr ar-Razi al-Musytahar bi at-Tafsir al-Kabir wa Mafatih al-Ghaib*. Dar al-Fikr.
3. Majelis Tarjih dan Tajdid PP Muhammadiyah. (2020). *Keputusan Musyawarah Nasional Tarjih ke-31 tentang Kriteria Waktu Subuh*. PP Muhammadiyah.
4. Meeus, J. (1998). *Astronomical Algorithms* (2nd ed.). Willmann-Bell, Inc. (Bab 15: *Atmospheric Refraction*).
5. Azhari, Susiknan. (2015). *Ilmu Falak: Perjumpaan Khazanah Islam dan Sains Modern*. Suara Muhammadiyah.

PERTEMUAN 10: Dinamika Mekanika Orbital Bulan dan Fase Ijtimak (Konjungsi)

10.1 Teologi Kalender Lunar dan *Manzilah* Bulan

Epistemologi waktu dalam Islam menggunakan pergerakan bulan (lunar) sebagai basis utama penyusunan kalender peradaban dan penentuan awal bulan-bulan ibadah (seperti Ramadan, Syawal, dan Zulhijah). Berbeda dengan kalender surya yang basisnya adalah pergerakan semu tahunan matahari yang tidak terlihat bentuk fasenya oleh mata awam, kalender lunar didesain agar fasenya dapat diamati secara visual oleh seluruh umat manusia sebagai "jam raksasa" di langit.

Allah Subhanahu wa Ta'ala menjelaskan ketetapan fase mekanik ini secara eksplisit:

وَالْقَمَرَ قَدَرْتُهُ مَنَازِلَ حَتَّىٰ عَادَ كَالْعُرْجُونِ الْقَدِيمِ

“Dan telah Kami tetapkan bagi bulan manzilah-manzilah, sehingga (setelah dia sampai ke manzilah yang terakhir) kembalilah dia sebagai bentuk tandan yang tua.” (QS. Ya-Sin [36]: 39)

Tafsir Ringkas: Imam Fakhruddin Ar-Razi dalam *Mafatih al-Ghaib* memberikan analisis falak yang tajam terhadap kata *manaazila* (stasiun/tempat singgah). Pergerakan bulan mengelilingi bumi menghasilkan perubahan sudut iluminasi cahaya matahari yang dipantulkannya. Perubahan ini secara fisis tampak sebagai perubahan fase (sabit, separuh, purnama, dan kembali sabit tipis bagaikan *'urjunil qadim* atau tandan kurma tua yang melengkung). Secara saintifik, siklus *manzilah* ini tunduk pada hukum gravitasi Newton dan berujung pada satu titik perpotongan siklus baru yang disebut Konjungsi atau *Ijtimak*.

10.2 Mekanika Orbital dan Definisi Astronomis Ijtimak

Dalam hisab kontemporer, penentuan awal bulan kamariah tidak lagi bertumpu pada perkiraan umur bulan semata, melainkan pada sebuah peristiwa astronomis sesaat yang terukur secara mutlak, yaitu **Ijtimak** (Konjungsi Geosentris).

Ijtimak terjadi pada detik eksak ketika **Bujur Ekliptika Geosentris Bulan sama persis dengan Bujur Ekliptika Geosentris Matahari** (Selisih Bujur = 0 derajat).

1. **Geosentris:** Pengamatan ditarik dari titik pusat massa bumi, bukan dari permukaan bumi (toposentris). Hal ini menjadikan Ijtimak sebagai **peristiwa global**. Jika Ijtimak terjadi pada pukul 12:00 UTC, maka seluruh penduduk bumi mengalami momen Ijtimak pada detik yang sama (hanya penamaan jam lokalnya saja yang berbeda).
2. **Ekliptika:** Bidang orbit bumi mengelilingi matahari. Orbit bulan miring sekitar 5 derajat terhadap bidang ekliptika ini. Oleh karena itu, saat Ijtimak, bulan dan matahari berada pada garis bujur yang sama, namun lintangnya biasanya berbeda (kecuali saat terjadi Gerhana Matahari di mana bujur dan lintangnya bersinggungan).

Setelah detik Ijtimak berlalu, bulan akan bergerak ke arah timur meninggalkan matahari (karena revolusi bulan lebih cepat dari gerak semu matahari). Perpisahan angular ini disebut **Elongasi**, yang menjadi syarat utama terbentuknya hilal (bulan sabit pertama).

10.3 Algoritma Pelacakan Fase Bulan di Python

Untuk mencari detik eksak perpotongan (Selisih Bujur = 0°), kita tidak bisa menghitung posisi bulan secara linier setiap jam karena orbit bulan sangat anomali (elips dan dipengaruhi gravitasi matahari secara masif).

Python melalui pustaka `skyfield` menggunakan modul `almanac.moon_phases`. Modul ini adalah mesin pencari akar (root-finding algorithm) yang memindai pergerakan selisih bujur ekliptika:

- Selisih 0° : *New Moon* (Ijtimak / Bulan Baru)
- Selisih 90° : *First Quarter* (Kuartir Pertama / Bulan Separuh Awal)
- Selisih 180° : *Full Moon* (Purnama / Istiqbal)
- Selisih 270° : *Last Quarter* (Kuartir Terakhir / Bulan Separuh Akhir)

10.4 Source Code Mandiri: Detektor Fase Bulan dan Ijtimak Presisi

Simpan skrip Python ini dengan nama `detektor_ijtimak.py` dan jalankan di terminal. Aplikasi ini akan memetakan seluruh fase bulan pada rentang bulan dan tahun yang diminta pengguna dengan kepresisian fraksi detik tingkat observatorium.

Python

```
"""
Aplikasi: Detektor Fase Bulan & Penjejak Ijtimak Presisi (Konjungsi)
Penulis: Kasmui
Deskripsi: Skrip mandiri untuk melacak waktu eksak terjadinya fase-fase
bulan,
           khususnya Ijtimak (New Moon), berdasarkan selisih bujur
ekliptika geosentris.
"""

import datetime
import pytz
from skyfield.api import Loader
from skyfield import almanac

def lacak_fase_bulan():
    print("="*75)
    print("MESIN KOMPUTASI MEKANIKA ORBITAL BULAN & IJTIMAK (KONJUNGSI)")
    print("="*75)

    # 1. Inisialisasi Pustaka dan Ephemeris JPL
    load = Loader('ephemeris_data', verbose=False)
    ts = load.timescale()
    planets = load('de421.bsp')

    # 2. Input Parameter Waktu (Contoh: Ramadhan 1445 H / Maret 2024)
    # Kita buat input dinamis
```

```

try:
    tahun = int(input("[INPUT] Masukkan Tahun (Masehi) : "))
    bulan = int(input("[INPUT] Masukkan Bulan (1-12) : "))
    tz_input = input("[INPUT] Zona Waktu (Default: Asia/Jakarta): ")
    if not tz_input.strip():
        tz_input = 'Asia/Jakarta'
except ValueError:
    print("[X] Input harus berupa angka bulat.")
    return

zona_waktu = pytz.timezone(tz_input)

# 3. Menentukan Rentang Pemindaian (T0 ke T1)
# Mulai dari tanggal 1 bulan target, hingga tanggal 1 bulan berikutnya
t0_dt = zona_waktu.localize(datetime.datetime(tahun, bulan, 1, 0, 0,
0))

# Logika pergantian tahun jika bulan 12
if bulan == 12:
    t1_dt = zona_waktu.localize(datetime.datetime(tahun + 1, 1, 1, 0,
0, 0))
else:
    t1_dt = zona_waktu.localize(datetime.datetime(tahun, bulan + 1, 1,
0, 0, 0))

t0 = ts.from_datetime(t0_dt)
t1 = ts.from_datetime(t1_dt)

print("\n" + "-"*75)
print(f"MENGUNDUH MATRIKS DATA FASE BULAN: PERIODE {t0_dt.strftime('%B
%Y').upper()}")
print("-"*75)

# 4. Mengeksekusi Pencarian Fase Bulan
# Fungsi almanac.moon_phases mengembalikan dua array: waktu dan indeks
fase (0, 1, 2, 3)
waktu_fase, indeks_fase = almanac.find_discrete(t0, t1,
almanac.moon_phases(planets))

# Kamus Terjemahan Indeks ke Istilah Falak
nama_fase = {
    0: "New Moon (Ijtimak / Konjungsi) 🌑",
    1: "First Quarter (Kuartir Pertama) 🌒",
    2: "Full Moon (Purnama / Istiqbal) 🌕",
    3: "Last Quarter (Kuartir Terakhir) 🌘"
}

# 5. Pemrosesan dan Ekstraksi Output Waktu
if len(waktu_fase) == 0:
    print("[X] Anomali: Tidak ada fase bulan utama yang tercatat pada
rentang ini.")
    return

```

```

for t_obj, fase in zip(waktu_fase, indeks_fase):
    # Konversi objek waktu absolut ke UTC dan Waktu Lokal
    waktu_utc = t_obj.utc_datetime()
    waktu_lokal =
waktu_utc.replace(tzinfo=pytz.utc).astimezone(zona_waktu)

    # Ekstraksi Delta-T untuk verifikasi presisi dinamika rotasi bumi
    delta_t = t_obj.delta_t

    print(f"Fase : {nama_fase[fase]}")
    print(f" -> Universal Time (UTC) : {waktu_utc.strftime('%Y-%m-%d
%H:%M:%S')} UTC")
    print(f" -> Waktu Lokal          : {waktu_lokal.strftime('%Y-%m-
%d %H:%M:%S')} {zona_waktu.zone}")

    if fase == 0:
        print(" -> STATUS KHUSUS          : Siklus orbit lunar (Bulan
Syar'i) baru saja di-reset.")
        print("                          Titik awal penentuan umur
bulan (hilal) dimulai dari detik ini.")

        print(f" -> Delta-T (Koreksi)      : {delta_t:.2f} detik")
        print("-" * 50)

    print("="*75)
    print("Catatan: Ijtimak Geosentris adalah fenomena absolut/global yang
tidak bergantung pada lokasi pengamat.")
    print("="*75)

if __name__ == '__main__':
    lacak_fase_bulan()

```

10.5 Penjelasan Anatomi Source Code

1. **Resolusi Timescale Absolut (Langkah 3):** Rentang waktu (T0 dan T1) mendefinisikan batas pencarian algoritma. Dengan memasukkan tanggal 1 hingga tanggal 1 bulan berikutnya, algoritma akan memindai tepat satu periode bulan kalender Masehi untuk mencari 4 transisi fase utama.
2. **Fungsi Inti `almanac.moon_phases(planets)` (Langkah 4):** Di balik layar, fungsi ini membangun persamaan matematis yang mengekstrak koordinat geosentris bumi, bulan, dan matahari. Ia kemudian memproyeksikan ketiga vektor benda langit tersebut ke dalam bidang ekliptika dan menghitung perbedaan sudut bujurnya. Algoritma `find_discrete` (pencarian akar) kemudian diterjunkan untuk mencari irisan waktu yang sangat spesifik ketika perbedaannya tepat menyentuh angka 0° , 90° , 180° , dan 270° .
3. **Matriks Pemetaan (`nama_fase`):** Skyfield mengembalikan matriks integer (0 hingga 3).
 - o Nilai 0 adalah jantung komputasi awal bulan, yaitu **Ijtimak** (*New Moon*). Mulai dari detik inilah "Umur Bulan" (Age of Moon) dihitung dengan stopwatch digital hingga saat matahari terbenam (sunset) di suatu wilayah untuk mengevaluasi apakah hilal sudah cukup umur untuk dirukyat.

- Nilai 2 adalah *Full Moon* (Purnama). Secara kalender Hijriah, jika sistem hisab berjalan normal, fase ini hampir selalu bertepatan dengan tanggal 14 atau pertengahan bulan kamariah (sering disebut sebagai fenomena *Istiqbal*).
- 4. **Ekstraksi Delta-T ($t_{obj.\delta_t}$):** Karena orbit bulan sangat terpengaruh oleh pasang surut gravitasi bumi, melacak nilai Delta-T saat Ijtimak sangat krusial. Ini adalah selisih antara Waktu Atom (Terrestrial Time) dan Waktu Rotasi Bumi. Skrip menampilkan nilai ini sebagai bukti bahwa komputasi ini merupakan Hisab Hakiki tingkat presisi tinggi yang mengakomodir parameter perturbasi langit.

10.6 Daftar Pustaka (Pertemuan 10)

1. Ar-Razi, Fakhrudin. (1981). *Tafsir al-Fakhr ar-Razi al-Musyтахar bi at-Tafsir al-Kabir wa Mafatih al-Ghaib*. Dar al-Fikr.
2. Meeus, J. (1998). *Astronomical Algorithms* (2nd ed.). Willmann-Bell, Inc. (Bab 47: *Phases of the Moon*).
3. Majelis Tarjih dan Tajdid PP Muhammadiyah. (2009). *Pedoman Hisab Muhammadiyah*. Majelis Tarjih dan Tajdid PP Muhammadiyah.
4. Anwar, Syamsul. (2018). *Hari-hari Allah: Dinamika Kalender Islam Kontemporer dan Menuju Kalender Hijriah Global*. Suara Muhammadiyah.
5. Djamaluddin, Thomas. (2011). *Astronomi Memberi Solusi Penyatuan Ummat*. LAPAN.

PERTEMUAN 11: Kalkulus Visibilitas Hilal (Toposentrik vs Geosentrik)

11.1 Teologi *Ru'yah* dan Epistemologi Observasi Hilal

Penentuan awal bulan kamariah, khususnya bulan-bulan ibadah mahdah seperti Ramadan, Syawal, dan Zulhijah, bermuara pada fenomena penampakan bulan sabit pertama pasca-ijtimak yang disebut sebagai *Hilal*. Di dalam peradaban Islam, hilal bukan sekadar objek fisis, melainkan "jam kosmik" yang menkalibrasi ritme peribadatan umat manusia.

Al-Qur'an secara lugas menjadikan hilal sebagai parameter kronologis universal:

يَسْأَلُونَكَ عَنِ الْأَهْلِ لَيْلٍ هِيَ مَوْعِدُكُمْ لِلنَّاسِ وَالْحَجِّ

"Mereka bertanya kepadamu tentang bulan sabit. Katakanlah: 'Bulan sabit itu adalah tanda-tanda waktu bagi manusia dan (bagi ibadah) haji...'" (QS. Al-Baqarah [2]: 189)

Tafsir Ringkas: Imam Fakhruddin Ar-Razi dalam *Mafatih al-Ghaib* memberikan penekanan pada kata *al-ahillah* (bentuk jamak dari hilal). Beliau menjelaskan bahwa fluktuasi ukuran cahaya bulan dari malam ke malam adalah sebuah keajaiban matematis yang sangat intuitif. Berbeda dengan matahari yang piringannya selalu tampak konstan, bulan menyajikan "data visual" yang dapat dibaca langsung oleh semua manusia, terlepas dari tingkat literasi mereka.

Dalam ranah praksis operasional, Nabi Muhammad ﷺ memberikan metodologi observasi melalui sabdanya:

صُومُوا لِرُؤْيَيْهِ وَأُفْطِرُوا لِرُؤْيَيْهِ ، فَإِنْ غَمَّ عَلَيْكُمْ فَأَكْمِلُوا عِدَّةَ شَعْبَانَ ثَلَاثِينَ

"Berpuasalah kalian karena melihatnya (hilal) dan berbukalah karena melihatnya. Jika mendung menutupi kalian, maka sempurnakanlah hitungan bulan Sya'ban menjadi tiga puluh hari." (HR. Bukhari No. 1909)

Majelis Tarjih Muhammadiyah memahami hadis ini tidak dalam makna observasi optik yang kaku (rukyat fisik semata), melainkan sebagai perintah *ta'abbudi* untuk memastikan bahwa bulan benar-benar telah wujud di ufuk. Di era astronomi komputasional, *ru'yah bil 'ilmi* (observasi melalui sains hisab) bertransformasi menjadi instrumen valid untuk memetakan visibilitas hilal secara presisi, menghindari anomali cuaca seperti mendung atau polusi cahaya.

11.2 Transformasi Matriks: Geosentrik menuju Toposentrik

Pertemuan sebelumnya telah membahas fase Ijtimak, yaitu peristiwa di mana bujur ekliptika bulan dan matahari sejajar jika ditarik garis lurus dari **pusat bumi (Geosentrik)**. Namun, manusia tidak tinggal di inti bumi, melainkan di atas permukaannya.

Ketika kita melakukan observasi hilal saat matahari terbenam (sunset), kita beroperasi pada koordinat **Toposentrik**. Pergeseran titik referensi dari pusat bumi (sejauh radius ~ 6.371 km) ke permukaan bumi melahirkan fenomena yang disebut **Paralaks Ekuatorial Horizontal**.

Bulan adalah benda langit terdekat dengan bumi. Jaraknya yang relatif dekat (sekitar 384.400 km) membuat posisinya sangat sensitif terhadap lokasi pengamat. Bagi pengamat di permukaan bumi, bulan yang berada di dekat ufuk akan "tertekan" atau tampak lebih rendah ke bawah sebesar hampir 1 derajat dibandingkan posisi aslinya dari pusat bumi. Oleh karena itu, kalkulasi ketinggian hilal harus menggunakan algoritma toposentris yang mengikutsertakan variabel lintang, bujur, dan elevasi (ketinggian) pengamat secara absolut.

11.3 Kalkulus Parameter Visibilitas Hilal

Algoritma Imkanur Rukyat (visibilitas hilal) bertumpu pada lima parameter fisis saat detik persis piringan atas matahari menyentuh ufuk barat:

1. **Tinggi Hilal Toposentris (Altitude):** Jarak sudut vertikal dari pusat piringan bulan ke garis ufuk lokal. Syarat mutlaknya adalah bulan harus berada di atas ufuk ($\text{Altitude} > 0$ derajat).
2. **Elongasi (Angular Separation):** Jarak sudut sferis antara pusat bulan dan pusat matahari. Cahaya hilal terbentuk semata-mata dari pantulan sinar matahari. Ahli astronomi Prancis, André-Louis Danjon (1932), menemukan *Limit Danjon*: Jika elongasi kurang dari 7 derajat, sabit bulan tidak akan pernah terbentuk secara fisis karena terhalang bayangan pegunungan kawah di permukaan bulan itu sendiri.
3. **Umur Bulan (Age of Moon):** Rentang waktu (dalam jam) yang telah berlalu semenjak detik Ijtimak Geosentris hingga detik matahari terbenam di lokasi pengamat.
4. **Beda Azimut (DAZ):** Selisih sudut horizontal antara titik terbenamnya matahari dengan posisi hilal.
5. **Fraksi Iluminasi (Phase):** Persentase cakram bulan yang tersinari oleh matahari dan menghadap bumi. Sabit awal bulan biasanya memiliki ketebalan cahaya di bawah 1 persen.

11.4 Source Code Mandiri: Mesin Ekstraktor Matriks Visibilitas Hilal

Simpan skrip Python ini dengan nama `visibilitas_hilal.py` dan jalankan di terminal. Aplikasi ini tidak hanya menghitung Ijtimak, tetapi secara asinkron melompat ke momen Sunset terdekat untuk mengekstraksi seluruh variabel optik hilal dalam sudut pandang Toposentris.

```
Python
"""
Aplikasi: Mesin Kalkulus Visibilitas Hilal Toposentrik
Penulis: Kasmui
Deskripsi: Skrip komputasi tingkat observatorium untuk memotret matriks
           posisi Hilal tepat pada milidetik terbenamnya matahari lokal,
           mengakomodir efek paralaks dan refraksi atmosferik.
"""

import datetime
import math
import pytz
```

```

from skyfield.api import Loader, wgs84
from skyfield import almanac

def komputasi_hilal_toposentrik():
    print("="*75)
    print("MESIN EKSTRAKTOR MATRIKS VISIBILITAS HILAL (TOPOSENTRIK)")
    print("="*75)

    # 1. Inisialisasi Pustaka & Ephemeris
    load = Loader('ephemeris_data', verbose=False)
    ts = load.timescale()
    planets = load('de421.bsp')
    bumi = planets['earth']
    matahari = planets['sun']
    bulan = planets['moon']

    # 2. Input Parameter Lokasi Pengamat (Observatorium / Pos Rukyat)
    # Contoh: Observatorium Ilmu Falak (OIF) UMSU, Medan
    nama_lokasi = "Medan, Sumatera Utara"
    lintang = 3.5952
    bujur = 98.6722
    elevasi = 25.0 # mdpl
    zona_waktu = pytz.timezone('Asia/Jakarta')

    lokasi_pengamat = wgs84.latlon(lintang, bujur, elevation_m=elevasi)

    # Waktu Observasi (Hari ini)
    sekarang = datetime.datetime.now(zona_waktu)

    print(f"[POS OBSERVASI] : {nama_lokasi}")
    print(f"[KOORDINAT]      : Lintang {lintang}°, Bujur {bujur}°, Elevasi
{elevasi}m")
    print(f"[TANGGAL RUKYAT]: {sekarang.strftime('%d %B %Y')}")
    print("-" * 75)

    # 3. MENCARI DETIK IJTIMAK SEBELUMNYA (Umur Bulan)
    # Kita mundur 15 hari ke belakang untuk memastikan kita menangkap fase
    New Moon terakhir
    t_mundur = ts.from_datetime(sekarang - datetime.timedelta(days=15))
    t_sekarang = ts.from_datetime(sekarang + datetime.timedelta(days=1))

    waktu_fase, indeks_fase = almanac.find_discrete(t_mundur, t_sekarang,
almanac.moon_phases(planets))

    t_ijtimak = None
    for t_obj, fase in zip(waktu_fase, indeks_fase):
        if fase == 0: # 0 = New Moon / Konjungsi
            t_ijtimak = t_obj

    if not t_ijtimak:
        print("[X] Kesalahan sistem: Gagal mendeteksi momen Ijtimak
terakhir.")
        return

```

```

    waktu_ijtimak_wib =
t_ijtimak.utc_datetime().replace(tzinfo=pytz.utc).astimezone(zona_waktu)
    print(f"[DATA GLOBAL] Detik Ijtimak : {waktu_ijtimak_wib.strftime('%d-
%b-%Y %H:%M:%S')} WIB")

    # 4. Mencari Waktu Terbenam Matahari (Sunset) Hari Ini
    t0_sunset =
ts.from_datetime(zona_waktu.localize(datetime.datetime(sekarang.year,
sekarang.month, sekarang.day, 12, 0, 0)))
    t1_sunset =
ts.from_datetime(zona_waktu.localize(datetime.datetime(sekarang.year,
sekarang.month, sekarang.day, 23, 59, 59)))

    f_sunset = almanac.sunrise_sunset(planets, lokasi_pengamat)
    waktu_ss, transisi_ss = almanac.find_discrete(t0_sunset, t1_sunset,
f_sunset)

    t_sunset = None
    for t_obj, tr in zip(waktu_ss, transisi_ss):
        if tr == 0: # 0 = Matahari tenggelam
            t_sunset = t_obj
            break

    if not t_sunset:
        print("[!] Matahari tidak terbenam pada hari ini di lokasi
tersebut.")
        return

    waktu_sunset_wib =
t_sunset.utc_datetime().replace(tzinfo=pytz.utc).astimezone(zona_waktu)
    print(f"[DATA LOKAL] Detik Sunset : {waktu_sunset_wib.strftime('%d-
%b-%Y %H:%M:%S')} WIB")

    # Evaluasi Prematur
    if t_sunset.tt < t_ijtimak.tt:
        print("\n[KESIMPULAN] : Hilal Belum Wujud (Sunset terjadi sebelum
Ijtimak).")
        print("                Bulan ini wajib disempurnakan (Istikmal 30
Hari).")
        print("=="*75)
        return

    # 5. EKSTRAKSI MATRIKS TOPOSENTRIK TEPAT SAAT SUNSET
    print("\n[MATRIKS VISIBILITAS HILAL SAAT SUNSET]")

    # Hitung Umur Bulan (Selisih waktu Sunset dengan Ijtimak dalam jam)
    umur_bulan_jam = (t_sunset.tt - t_ijtimak.tt) * 24.0
    print(f"1. Umur Bulan (Age of Moon) : {umur_bulan_jam:.2f} Jam")

    # Observasi Posisi Matahari (Apparent)
    obs_matahari =
lokasi_pengamat.at(t_sunset).observe(matahari).apparent()
    alt_sun, az_sun, _ = obs_matahari.altaz()

```

```

# Observasi Posisi Bulan (Apparent - Sudah termasuk paralaks topografi
& refraksi)
obs_bulan = lokasi_pengamat.at(t_sunset).observe(bulan).apparent()
alt_moon, az_moon, _ = obs_bulan.altaz()

tinggi_hilal = alt_moon.degrees
print(f"2. Tinggi Hilal Toposentrik : {tinggi_hilal:+.4f} Derajat dari
Ufuk")

# Hitung Beda Azimut (DAZ)
beda_azimut = az_moon.degrees - az_sun.degrees
# Normalisasi sudut
if beda_azimut > 180: beda_azimut -= 360
elif beda_azimut < -180: beda_azimut += 360

arah_relatif = "Utara" if beda_azimut > 0 else "Selatan"
print(f"3. Beda Azimut (DAZ) : {abs(beda_azimut):.4f} Derajat
(Di sebelah {arah_relatif} Matahari)")

# Hitung Elongasi Sferis (Separation Angle)
elongasi = obs_bulan.separation_from(obs_matahari).degrees
print(f"4. Elongasi Geosferis : {elongasi:.4f} Derajat")

# Hitung Fraksi Iluminasi Cahaya
fraksi = almanac.fraction_illuminated(planets, 'moon', t_sunset)
persentase_cahaya = fraksi * 100.0
print(f"5. Fraksi Iluminasi (Cahaya): {persentase_cahaya:.4f} %")

# 6. EVALUASI WUJUDUL HILAL DASAR (Positif / Negatif)
print("\n[KESIMPULAN VISIBILITAS]")
if tinggi_hilal > 0:
    print("STATUS : HILAL WUJUD DI ATAS UFUK (Bulan Baru Tiba).")
    if elongasi < 7.0:
        print("CATATAN: Elongasi di bawah Limit Danjon (7 Derajat).")
        print("Hilal wujud secara hisab, namun mustahil
dirukyat dengan optik.")
    else:
        print("STATUS : HILAL DI BAWAH UFUK (Bulan Terbenam Lebih Dulu).")
        print("Istikmal (Genapkan bulan berjalan menjadi 30
hari).")

print("="*75)

if __name__ == '__main__':
    komputasi_hilal_toposentrik()

```

11.5 Penjelasan Anatomi Source Code

1. **Pelacakan Mundur Ijtimak (Langkah 3):** Umur bulan adalah selisih antara waktu sunset dan detik konjungsi. Karena kita tidak selalu tahu kapan Ijtimak terakhir terjadi, algoritma memanipulasi waktu dengan merancang variabel `t_mundur` sejauh 15 hari ke belakang. Ini

menjamin `find_discrete` akan selalu berhasil menangkap satu fase *New Moon* (indeks 0) untuk dijadikan patokan awal (titik nol) dari pembentukan sabit.

2. **Pemotretan Sinkron (Langkah 4 & 5):** Mesin mencari waktu eksak kapan piringan matahari terbenam (menggunakan `almanac.sunrise_sunset`). Detik absolut tersebut (`t_sunset`) kemudian dibekukan. Pada detik yang sama persis, Python mengeksekusi dua baris pengamatan secara simultan:


```
lokasi_pengamat.at(t_sunset).observe(matahari) dan
lokasi_pengamat.at(t_sunset).observe(bulan).
```

 Sinkronisasi *real-time* ini adalah inti dari algoritma toposentrik, memastikan tidak ada jeda perhitungan antara gerak bumi, bulan, dan matahari.
3. **Koreksi Apparent Otomatis:** Pemanggilan `.apparent().altaz()` secara matematis sudah membereskan seluruh rintangan observasi klasik. Algoritma ini menarik data ketinggian elevasi (`elevasi = 25.0`), memberlakukan Paralaks Ekuatorial Horizontal akibat jarak toposentrik, dan menerapkan defleksi cahaya atmosferik, sehingga derajat tinggi hilal (`alt_moon.degrees`) yang keluar murni adalah posisi fisis yang bisa ditunjuk langsung oleh mata atau teleskop.
4. **Fungsi Separasi Sferis (`separation_from`):** Menghitung elongasi tidak boleh menggunakan pengurangan linier sederhana antara tinggi matahari dan tinggi bulan. Elongasi adalah garis melengkung tiga dimensi di angkasa. Fungsi bawaan `separation_from` mengeksekusi trigonometri sferis (*Haversine formula*) di latar belakang untuk mendapatkan jarak busur derajat yang absolut, yang kemudian secara kritis dibandingkan dengan *Limit Danjon* (7 Derajat).

11.6 Daftar Pustaka (Pertemuan 11)

1. Anwar, Syamsul. (2018). *Hari-hari Allah: Dinamika Kalender Islam Kontemporer dan Menuju Kalender Hijriah Global*. Suara Muhammadiyah.
2. Ar-Razi, Fakhruddin. (1981). *Tafsir al-Fakhr ar-Razi al-Musytahar bi at-Tafsir al-Kabir wa Mafatih al-Ghaib*. Dar al-Fikr.
3. Meeus, J. (1998). *Astronomical Algorithms* (2nd ed.). Willmann-Bell, Inc. (Bab 39: *Position of the Moon* dan Bab 47: *Phases of the Moon*).
4. Odeh, M. S. (2004). *New Criterion for Lunar Crescent Visibility*. *Experimental Astronomy*, 18(1), 39-64.
5. Saksono, Tono. (2007). *Mengkompromikan Rukyat & Hisab*. Amythas Publicita.

PERTEMUAN 12: Implementasi Kriteria Rukyatul Hilal & Neo MABIMS

12.1 Teologi Konsensus (*Syura*) dan Otoritas Epistemologis

Dalam wacana penetapan awal bulan hijriah, perdebatan sering kali tidak terletak pada ranah matematis (karena komputasi astronomi modern sudah sangat presisi), melainkan pada ranah kriteria epistemologis: "Seberapa tinggi bulan harus berada agar diakui secara *syar'i* dan kenegaraan?"

Al-Qur'an memberikan landasan sosiologis dan yurisprudensi mengenai pentingnya ketaatan terhadap konsensus dan otoritas yang sah demi menjaga persatuan umat:

يَا أَيُّهَا الَّذِينَ ءَامَنُوا أَطِيعُوا اللَّهَ وَأَطِيعُوا الرَّسُولَ وَأُولَى الْأَمْرِ مِنْكُمْ فَإِن تَنَزَعْتُمْ فِي شَيْءٍ فَرُدُّوهُ إِلَى اللَّهِ وَالرَّسُولِ

"Hai orang-orang yang beriman, taatilah Allah dan taatilah Rasul (Nya), dan ulil amri di antara kamu. Kemudian jika kamu berlainan pendapat tentang sesuatu, maka kembalikanlah ia kepada Allah (Al Quran) dan Rasul (sunnahnya)..." (QS. An-Nisa' [4]: 59)

Tafsir Ringkas: Imam Fakhrudin Ar-Razi dalam *Mafatih al-Ghaib* menafsirkan *Ulil Amri* tidak hanya sebagai pemimpin politik, tetapi juga mencakup para ilmuwan, fukaha, dan ahli hisab yang memiliki otoritas keilmuan (*Ahlul Halli wal 'Aqdi*). Dalam konteks regional Asia Tenggara, pemerintah melalui Kementerian Agama (MABIMS: Menteri Agama Brunei, Indonesia, Malaysia, dan Singapura) mengambil peran sebagai *Ulil Amri* yang berupaya menyatukan kriteria visibilitas hilal untuk menghindari perbedaan jatuhnya hari raya. Meskipun organisasi seperti Muhammadiyah memiliki ijtihad independen (Wujudul Hilal / KHGT yang akan dibahas pada pertemuan selanjutnya), mempelajari algoritma kriteria negara adalah syarat mutlak bagi seorang astronom Islam yang komprehensif.

12.2 Evolusi Parameter MABIMS: Dari Klasik menuju Neo MABIMS

Selama bertahun-tahun, Indonesia dan negara serumpun menggunakan kriteria MABIMS klasik yang dikenal dengan formula 2-3-8:

- Tinggi hilal minimal 2 derajat.
- Jarak sudut (elongasi) minimal 3 derajat.
- ATAU umur bulan minimal 8 jam pasca-ijtimak.

Kriteria ini dikritik habis-habisan oleh komunitas astronomi global karena secara fisis, hilal dengan ketinggian 2 derajat dan elongasi 3 derajat **mustahil** dirukyat (dilihat) baik dengan mata telanjang maupun teleskop tercanggih sekalipun. Limit Danjon secara tegas membuktikan bahwa pendaran cahaya sabit tidak akan terbentuk jika elongasi di bawah 7 derajat akibat topografi kawah bulan.

Sebagai respons terhadap rasionalitas sains, pada tahun 2021, MABIMS secara resmi memperbarui parameternya menjadi **Kriteria Rekomendasi Jakarta 2017 (Neo MABIMS)**:

1. **Tinggi Hilal Minimal:** 3 derajat (Toposentrik).
2. **Elongasi Minimal:** 6.4 derajat (Geosentrik).

Perubahan ini bukan sekadar pergantian angka, melainkan pergeseran paradigma dari kriteria *politik-kompromis* menjadi kriteria *sains-empiris* yang didasarkan pada ribuan titik data observasi global.

12.3 Konstruksi Pohon Keputusan (*Decision Tree*) Algoritmik

Dalam ilmu komputer, kriteria fikih atau negara diterjemahkan ke dalam bentuk gerbang logika (*logic gates*). Kriteria Neo MABIMS menggunakan operator logika `AND`. Artinya, **kedua syarat wajib terpenuhi** secara absolut pada saat matahari terbenam. Jika tinggi mencapai 4 derajat namun elongasi hanya 5 derajat, maka sistem harus mengembalikan nilai `False` (Gagal / Istikmal).

Arsitektur logika pemrogramannya:

Plaintext

```
JIKA (Tinggi_Hilal >= 3.0) DAN (Elongasi >= 6.4):
    VONIS = "Bulan Baru (Tanggal 1)"
JIKA TIDAK:
    VONIS = "Istikmal (Genapkan 30 Hari)"
```

12.4 Source Code Mandiri: Mesin Filter Otomatis Kriteria Neo MABIMS

Simpan skrip Python ini dengan nama `neo_mabims_filter.py` dan jalankan di terminal. Aplikasi ini akan menelan data dari Pertemuan 11, melewatinya melalui saringan kriteria MABIMS, dan seketika menjatuhkan vonis jatuhnya awal bulan.

Python

```
"""
Aplikasi: Mesin Filter Kriteria Visibilitas Hilal (Neo MABIMS)
Penulis: Kasmui
Deskripsi: Skrip komputasi otomatis yang mengekstraksi data toposentris
hilal
pada saat sunset dan mengevaluasinya secara ketat menggunakan
gerbang logika kriteria Neo MABIMS (Tinggi 3°, Elongasi 6.4°).
"""

import datetime
import pytz
from skyfield.api import Loader, wgs84
from skyfield import almanac

def evaluasi_neo_mabims():
    print("="*75)
    print("SISTEM EVALUASI AWAL BULAN: KRITERIA NEO MABIMS (3° - 6.4°)")
    print("="*75)
```

```

# 1. Inisialisasi Pustaka & Objek Langit
load = Loader('ephemeris_data', verbose=False)
ts = load.timescale()
planets = load('de421.bsp')
bumi = planets['earth']
matahari = planets['sun']
bulan = planets['moon']

# 2. Parameter Lokasi Titik Pantau (Contoh: Pelabuhan Ratu, Sukabumi)
# Lokasi ini sering dijadikan titik nol (datum) rukyatul hilal
nasiona
nama_lokasi = "Pos Observasi Cibeas, Pelabuhan Ratu"
lintang = -7.0286
bujur = 106.5569
elevasi = 50.0 # mdpl
zona_waktu = pytz.timezone('Asia/Jakarta')
lokasi_pengamat = wgs84.latlon(lintang, bujur, elevation_m=elevasi)

# 3. Input Tanggal Evaluasi (Target Sunset)
try:
    tgl_input = input("[INPUT] Masukkan Tanggal Rukyat (YYYY-MM-DD)
atau tekan Enter untuk hari ini: ")
    if not tgl_input.strip():
        sekarang = datetime.datetime.now(zona_waktu)
    else:
        tahun, bln, hari = map(int, tgl_input.split('-'))
        sekarang = zona_waktu.localize(datetime.datetime(tahun, bln,
hari, 15, 0, 0))
except ValueError:
    print("[X] Format tanggal salah. Gunakan YYYY-MM-DD.")
    return

print(f"\n[TITIK PANTAU] : {nama_lokasi}")
print(f"[TANGGAL]      : {sekarang.strftime('%d %B %Y')}")
print("-" * 75)

# 4. Melacak Detik Terbenamnya Matahari (Sunset)
t0_sunset = ts.from_datetime(sekarang.replace(hour=12, minute=0,
second=0))
t1_sunset = ts.from_datetime(sekarang.replace(hour=23, minute=59,
second=59))

f_sunset = almanac.sunrise_sunset(planets, lokasi_pengamat)
waktu_ss, transisi_ss = almanac.find_discrete(t0_sunset, t1_sunset,
f_sunset)

t_sunset = None
for t_obj, tr in zip(waktu_ss, transisi_ss):
    if tr == 0: # Sunset
        t_sunset = t_obj
        break

if not t_sunset:

```

```

print("[!] Anomali: Matahari tidak terbenam pada hari ini.")
return

waktu_sunset_wib =
t_sunset.utc_datetime().replace(tzinfo=pytz.utc).astimezone(zona_waktu)
print(f"[WAKTU SUNSET] : {waktu_sunset_wib.strftime('%H:%M:%S')} WIB")

# 5. Ekstraksi Matriks Parameter Hilal saat Sunset
# a. Observasi Matahari Apparent
obs_matahari =
lokasi_pengamat.at(t_sunset).observe(matahari).apparent()

# b. Observasi Bulan Apparent (Tinggi Toposentrik)
obs_bulan = lokasi_pengamat.at(t_sunset).observe(bulan).apparent()
alt_moon, _, _ = obs_bulan.altaz()
tinggi_hilal = alt_moon.degrees

# c. Elongasi Geosentris
# Kriteria Neo MABIMS mensyaratkan elongasi geosentris (pusat bumi),
bukan toposentris
obs_bumi_pusat = bumi.at(t_sunset)
bulan_geosentris = obs_bumi_pusat.observe(bulan).apparent()
matahari_geosentris = obs_bumi_pusat.observe(matahari).apparent()
elongasi =
bulan_geosentris.separation_from(matahari_geosentris).degrees

print("\n[MATRIKS DATA ASTRONOMIS]")
print(f"1. Tinggi Hilal Toposentrik : {tinggi_hilal:+.4f}° (Syarat >=
3.0°)")
print(f"2. Elongasi Geosentris      : {elongasi:.4f}° (Syarat >=
6.4°)")

# 6. POHON KEPUTUSAN (DECISION TREE) - LOGIKA BOOLEAN MABIMS
print("\n[VONIS KRITERIA NEO MABIMS]")

# Syarat mutlak: Hilal harus di atas ufuk dahulu
if tinggi_hilal <= 0:
    print("-> HASIL      : GAGAL (Hilal berada di bawah ufuk / Belum
wujud).")
    print("-> KEPUTUSAN : ISTIKMAL (Bulan berjalan digenapkan menjadi
30 Hari).")
else:
    # Evaluasi gerbang AND
    lolos_tinggi = tinggi_hilal >= 3.0
    lolos_elongasi = elongasi >= 6.4

    if lolos_tinggi and lolos_elongasi:
        print("-> HASIL      : TERPENUHI (Tinggi >= 3° DAN Elongasi >=
6.4°).")
        print("-> KEPUTUSAN : BULAN BARU (Besok adalah tanggal 1
bulan baru).")
    else:
        print("-> HASIL      : TIDAK TERPENUHI (Satu atau kedua syarat
tidak lolos).")

```

```

# Diagnosa spesifik
if not lolos_tinggi:
    print("    [!] Ketinggian hilal tidak mencapai batas
minimal 3 derajat.")
    if not lolos_elongasi:
        print("    [!] Elongasi tidak mencapai batas minimal 6.4
derajat (Limit Danjon/Visibilitas).")
        print("-> KEPUTUSAN : ISTIKMAL (Bulan berjalan digenapkan
menjadi 30 Hari).")

print("="*75)

if __name__ == '__main__':
    evaluasi_neo_mabims()

```

12.5 Penjelasan Anatomi Source Code

1. **Fungsi Input Dinamis (Langkah 3):** Skrip ini dilengkapi dengan fungsi `input()` untuk memeriksa tanggal historis maupun memprediksi masa depan (misalnya Sidang Isbat tahun 2030). Jika pengguna hanya menekan *Enter*, mesin secara otomatis mengambil waktu hari ini menggunakan `datetime.datetime.now()`.
2. **Perbedaan Observasi Toposentrik vs Geosentrik (Langkah 5):** Ini adalah jantung akurasi dari Kriteria Neo MABIMS.
 - o Untuk mengukur **Tinggi Hilal**, negara menggunakan parameter *Toposentrik Apparent* (diukur dari posisi pengamat di permukaan bumi dengan memperhitungkan refraksi), diwakili oleh variabel `obs_bulan.altaz()`.
 - o Namun, untuk mengukur **Elongasi**, MABIMS bersepakat menggunakan jarak *Geosentrik* (diukur dari pusat bumi murni tanpa paralaks pengamat). Oleh karena itu, *source code* pada Langkah 5c dengan sangat hati-hati menarik variabel observasi baru dari pusat bumi (`bumi.at(t_sunset)`) untuk mengekstrak nilai sudut separasi, mencegah terjadinya *bug* logika matematis.
3. **Evaluasi Gerbang AND (`lolos_tinggi and lolos_elongasi`):** Pohon keputusan (*decision tree*) diterapkan secara bertingkat. Lapis pertama memeriksa apakah hilal wujud (`Tinggi > 0`). Jika negatif, program langsung melompat ke kesimpulan Istikmal tanpa perlu mengevaluasi elongasi (efisiensi pemrosesan). Lapis kedua mengevaluasi dua variabel boolean secara simultan. Jika salah satu bernilai `False`, mesin akan mencetak diagnosa kegagalan secara spesifik untuk memandu analisis pengguna.

12.6 Daftar Pustaka (Pertemuan 12)

1. Ar-Razi, Fakhruddin. (1981). *Tafsir al-Fakhr ar-Razi al-Musytahar bi at-Tafsir al-Kabir wa Mafatih al-Ghaib*. Dar al-Fikr.
2. Kementerian Agama Republik Indonesia. (2021). *Kriteria Baru MABIMS (Tinggi 3 Derajat, Elongasi 6.4 Derajat) dalam Penentuan Awal Bulan Hijriah*. Dirjen Bimas Islam Kemenag RI.
3. Odeh, M. S. (2004). *New Criterion for Lunar Crescent Visibility*. *Experimental Astronomy*, 18(1), 39-64.

4. Susiknan Azhari. (2015). *Ilmu Falak: Perjumpaan Khazanah Islam dan Sains Modern*. Suara Muhammadiyah.
5. Djamaluddin, Thomas. (2022). *Menyatukan Hari Raya: Dari Kriteria 2-3-8 Menuju Rekomendasi Jakarta 2017*. LAPAN-BRIN.

PERTEMUAN 13: Arsitektur Kalender Hijriah Global Tunggal (KHGT)

13.1 Teologi Kesatuan Waktu dan Paradigma Kalender Global

Sepanjang sejarah peradaban Islam pasca-khulafaur rasyidin, penjadwalan waktu umat Islam terpecah ke dalam sistem penanggalan lokal (zonal) yang berpegang pada prinsip *ikhtilaf al-matali'* (perbedaan tempat terbit bulan). Akibatnya, umat Islam sering kali merayakan hari besar seperti Idulfitri dan Iduladha pada hari yang berbeda-beda, bahkan di dalam satu benua yang sama.

Dalam epistemologi modern, fenomena ini tidak lagi dipandang semata-mata sebagai kekayaan khazanah fikih, melainkan sebagai sebuah utang peradaban (*civilizational debt*). Al-Qur'an mengamanatkan kesatuan umat dan kesatuan sistem waktu secara universal:

إِنَّ عِدَّةَ الشُّهُورِ عِنْدَ اللَّهِ اثْنَا عَشَرَ شَهْرًا فِي كِتَابِ اللَّهِ يَوْمَ خَلَقَ السَّمَوَاتِ وَالْأَرْضَ مِنْهَا أَرْبَعَةٌ حُرْمٌ ذَلِكَ الدِّينُ الْقَيِّمُ

"*Sesungguhnya bilangan bulan pada sisi Allah adalah dua belas bulan, dalam ketetapan Allah di waktu Dia menciptakan langit dan bumi, di antaranya empat bulan haram. Itulah (ketetapan) agama yang lurus...*" (QS. At-Taubah [9]: 36)

Tafsir Ringkas: Imam Fakhrudin Ar-Razi di dalam *Mafatih al-Ghaib* menegaskan bahwa frasa *'iddatasy-syuhur* (bilangan bulan) dan *ad-diinul qayyim* (agama yang lurus/sistem yang tegak) mengindikasikan bahwa sistem waktu Islam bersifat given (pasti), universal, dan terikat pada hukum mekanika langit yang objektif, bukan pada subjektivitas geografis. Majelis Tarjih dan Tajdid PP Muhammadiyah, yang didukung oleh putusan Mukhtamar Muhammadiyah ke-47 (2015) dan Kongres Internasional Istanbul Turki (2016), merumuskan konsep *ittihad al-matali'* (kesatuan tempat terbit) ke dalam wujud Kalender Hijriah Global Tunggal (KHGT). Prinsip KHGT adalah "Satu Hari Satu Tanggal di Seluruh Dunia", menyinkronkan penanggalan sipil dan ibadah agar sejajar dengan kalender Gregorian, namun tetap berbasis pergerakan lunar.

13.2 Anatomi Resolusi Istanbul 2016 (Parameter KHGT)

Arsitektur KHGT tidak dibangun atas dasar spekulasi, melainkan diikat oleh parameter astronomis fisis yang disepakati oleh ratusan astronom dan ulama dari seluruh dunia pada Kongres Internasional Rekeyasa Kalender Hijriah di Istanbul (Mei 2016).

Konstruksi algoritma KHGT bertumpu pada syarat-syarat kumulatif berikut:

1. **Syarat Geosentris Mutlak (Ijtimak):** Konjungsi (Ijtimak) bulan dan matahari wajib terjadi sebelum fajar menyingsing di benua paling timur bumi (Selandia Baru / New Zealand).
2. **Syarat Toposentris Eksklusif (Imkanur Rukyat Global):** Pada saat terbenamnya matahari di titik mana pun di muka bumi pasca-ijtimak, hilal harus mencapai batas visibilitas dengan parameter ganda:

- **Elongasi Minimal:** 8 derajat.
 - **Tinggi Minimal:** 5 derajat.
3. **Hukum Transfer Global (Sapu Jagat):** Jika syarat poin 1 dan 2 terpenuhi di *satu titik saja* di permukaan bumi (bahkan jika titik itu berada di tengah Samudra Pasifik), maka penentuan awal bulan ditransfer secara absolut ke seluruh dunia. Seluruh bumi akan memasuki tanggal 1 Hijriah pada keesokan harinya (berdasarkan sinkronisasi Garis Batas Tanggal Internasional / *International Date Line*).

13.3 Transformasi Algoritma: Mengunci Batas Timur Bumi (Fajar Selandia Baru)

Tantangan komputasi paling fundamental dalam mendesain arsitektur KHGT menggunakan Python adalah mengawinkan waktu Ijtimak (yang bersifat bebas zona) dengan waktu terbit fajar (*Dawn*) di koordinat ekstrem timur bumi.

Selandia Baru (terutama koordinat timurnya seperti Auckland atau Wellington) dikonvensikan sebagai zona waktu daratan terdepan di bumi (UTC+12 / UTC+13). Jika Ijtimak terjadi *setelah* terbit fajar di Selandia Baru, maka negara tersebut sudah terlanjur memulai hari puasanya tanpa dasar astronomis yang valid, yang akan melanggar prinsip syar'i. Oleh karena itu, mesin KHGT tahap pertama harus mampu melakukan validasi batas waktu ini secara otomatis.

13.4 Source Code Mandiri: Modul Ekstraksi dan Validasi Basis KHGT

Simpan skrip Python ini dengan nama `khgt_basis_engine.py` dan jalankan di terminal. Modul asinkronus ini berfungsi sebagai fondasi awal (Gerbang 1) yang mengekstraksi detik Ijtimak dan membenturkannya dengan batas Fajar Shadiq di Selandia Baru.

```
Python
"""
Aplikasi: Mesin Ekstraktor & Validasi Basis KHGT (Resolusi Istanbul 2016)
Penulis: Kasmui
Deskripsi: Skrip komputasi untuk mengekstraksi waktu Ijtimak Geosentris
           dan memvalidasi syarat pertama KHGT: "Ijtimak harus terjadi
           sebelum Fajar Shadiq di Selandia Baru (New Zealand)".
"""

import datetime
import pytz
from skyfield.api import Loader, wgs84
from skyfield import almanac

def evaluasi_basis_khgt():
    print("="*75)
    print("ARSTEKTUR KALENDER HIJRIAH GLOBAL TUNGGAL (KHGT)")
    print("Tahap 1: Validasi Batas Timur Bumi (Selandia Baru)")
    print("="*75)

    # 1. Inisialisasi Pustaka & Ephemeris
    load = Loader('ephemeris_data', verbose=False)
    ts = load.timescale()
```

```

planets = load('de421.bsp')
matahari = planets['sun']

# 2. Input Waktu Observasi (Target Bulan Masehi)
try:
    tahun = int(input("[INPUT] Masukkan Tahun Masehi Evaluasi : "))
    bulan = int(input("[INPUT] Masukkan Bulan Masehi (1-12) : "))
except ValueError:
    print("[X] Format angka tidak valid.")
    return

# 3. Mendapatkan Waktu Ijtimak Geosentris
# Rentang komputasi 1 bulan
t0_dt = datetime.datetime(tahun, bulan, 1, tzinfo=pytz.utc)
if bulan == 12:
    t1_dt = datetime.datetime(tahun + 1, 1, 1, tzinfo=pytz.utc)
else:
    t1_dt = datetime.datetime(tahun, bulan + 1, 1, tzinfo=pytz.utc)

t0 = ts.from_datetime(t0_dt)
t1 = ts.from_datetime(t1_dt)

waktu_fase, indeks_fase = almanac.find_discrete(t0, t1,
almanac.moon_phases(planets))

t_ijtimak = None
for t_obj, fase in zip(waktu_fase, indeks_fase):
    if fase == 0: # New Moon / Konjungsi Mutlak
        t_ijtimak = t_obj
        break

if not t_ijtimak:
    print(f"[!] Tidak ada Ijtimak yang terjadi pada bulan {bulan}-
{tahun}.")
    return

waktu_ijtimak_utc = t_ijtimak.utc_datetime()
print(f"\n[DATA GEOSENTRIS] Detik Ijtimak :
{waktu_ijtimak_utc.strftime('%Y-%m-%d %H:%M:%S')} UTC")

# 4. Inisialisasi Titik Referensi Batas Timur (Auckland, New Zealand)
# Auckland Coordinates: 36.8485 LS, 174.7633 BT (Elevasi: 20 mdpl)
nama_titik_timur = "Auckland, Selandia Baru"
lokasi_nz = wgs84.latlon(-36.8485, 174.7633, elevation_m=20.0)
tz_nz = pytz.timezone('Pacific/Auckland')

# 5. Komputasi Waktu Fajar (Dawn / Astronomical Twilight -18 Derajat)
di Selandia Baru
# Pencarian fajar dilakukan pada tanggal Masehi yang sama dengan
kejadian Ijtimak (Waktu Lokal NZ)
waktu_ijtimak_nz =
waktu_ijtimak_utc.replace(tzinfo=pytz.utc).astimezone(tz_nz)

```

```

# Buat rentang waktu pencarian untuk hari tersebut (00:00 - 23:59
Waktu Selandia Baru)
t0_fajar = ts.from_datetime(tz_nz.localize(datetime.datetime(
    waktu_ijtimak_nz.year, waktu_ijtimak_nz.month,
waktu_ijtimak_nz.day, 0, 0, 0))
t1_fajar = ts.from_datetime(tz_nz.localize(datetime.datetime(
    waktu_ijtimak_nz.year, waktu_ijtimak_nz.month,
waktu_ijtimak_nz.day, 23, 59, 59)))

# Fungsi untuk mencari fajar (-18 derajat dari ufuk)
def cek_fajar_astronomis(t):
    alt, _, _ = lokasi_nz.at(t).observe(matahari).apparent().altaz()
    return alt.degrees > -18.0

cek_fajar_astronomis.step_days = 0.05
waktu_fajar_nz_obj, status_fajar = almanac.find_discrete(t0_fajar,
t1_fajar, cek_fajar_astronomis)

t_fajar = None
for t_obj, stat in zip(waktu_fajar_nz_obj, status_fajar):
    if stat == 1: # Matahari bergerak naik melewati -18 derajat
(Fajar)
        t_fajar = t_obj
        break

if not t_fajar:
    print("[!] Anomali komputasi: Fajar tidak terdeteksi pada hari
tersebut di NZ.")
    return

waktu_fajar_utc = t_fajar.utc_datetime()
waktu_fajar_lokal_nz =
waktu_fajar_utc.replace(tzinfo=pytz.utc).astimezone(tz_nz)

print(f"[DATA TOPOSENTRIS] Batas Timur : {nama_titik_timur}")
print(f" - Fajar Shadiq (Lokal NZ)      :
{waktu_fajar_lokal_nz.strftime('%Y-%m-%d %H:%M:%S')} {tz_nz.zone}")
print(f" - Fajar Shadiq (Waktu UTC)    : {waktu_fajar_utc.strftime('%Y-
%-m-%d %H:%M:%S')} UTC")

# 6. EVALUASI LOGIKA GERBANG KHGT (Ijtimak < Fajar Selandia Baru)
print("\n[VONIS GERBANG 1: SYARAT GEOSENTRIS KHGT]")

# Komparasi waktu menggunakan julian day (t.tt)
selisih_jam = (t_fajar.tt - t_ijtimak.tt) * 24.0

if t_ijtimak.tt < t_fajar.tt:
    print("-> HASIL      : TERPENUHI (Ijtimak terjadi sebelum fajar di
Selandia Baru).")
    print(f"-> MARGIN      : Ijtimak mendahului fajar sejauh
{selisih_jam:.2f} Jam.")
    print("-> TINDAKAN   : Lanjutkan ke Gerbang 2 (Pemindaian Imkanur
Rukyat Global 5-8 Derajat).")
else:

```

```

        print("-> HASIL      : GAGAL (Ijtimak terjadi setelah fajar di
Selandia Baru).")
        print(f"-> MARGIN    : Ijtimak terlambat sejauh
{abs(selisih_jam):.2f} Jam dari waktu fajar.")
        print("-> TINDAKAN  : Awal bulan tidak dapat dimulai besok. Wajib
ditunda satu hari berikutnya (Istikmal Global).")

print("="*75)

if __name__ == '__main__':
    evaluasi_basis_khgt()

```

13.5 Penjelasan Anatomi Source Code

1. **Ekstraksi Ijtimak Absolut (Langkah 3):** Logika ini diwarisi dari materi Pertemuan 10. Kita menambang data waktu konjungsi dalam format Skala Waktu Universal (UTC) murni tanpa intervensi zona waktu lokal. Hal ini penting karena Ijtimak adalah peristiwa yang melibatkan seluruh massa bumi secara bersamaan.
2. **Pemetaan Koordinat Garis Batas Waktu (Langkah 4):** Penentuan kota Auckland (-36.8485 LS, 174.7633 BT) bertindak sebagai jangkar komputasional. Pemrograman berorientasi astronomi mengharuskan presisi spasial; oleh karenanya, ketinggian daratan Auckland (20 mdpl) dimasukkan ke dalam objek `wgs84` agar refraksi cahaya pada saat fajar terhitung sempurna.
3. **Rekayasa *Astronomical Twilight* (Langkah 5):** Resolusi Istanbul secara tegas menetapkan Fajar Fikih (*Fajar Shadiq*) sebagai garis finish. Dalam fisika atmosfer, ini setara dengan depresi matahari di angka -18 derajat dari garis cakrawala nyata. Algoritma menyuntikkan fungsi `cek_fajar_astronomis` ke dalam modul `find_discrete` untuk melacak pergerakan kurva posisi matahari hingga memotong garis imajiner -18 derajat tersebut di zona waktu Selandia Baru.
4. **Komparasi Linier Julian Day (Langkah 6):** Kesalahan paling umum pada pemrogram amatir adalah mengurangi dua objek zona waktu sipil yang berbeda (misalnya mengurangi WIB dengan waktu New Zealand). Hal ini sering menyebabkan *bug Offset-Naive*. Skrip ini menghindari kesalahan tersebut dengan mengomparasikan variabel `.tt` (*Terrestrial Time Julian Day*). Julian Day adalah garis angka murni (desimal tak terbatas); jika `t_ijtimak.tt` lebih kecil daripada `t_fajar.tt`, maka secara absolut kejadian Ijtimak berada di sebelah "kiri" (sebelum) waktu fajar pada garis waktu semesta, membuktikan syarat Gerbang 1 KHGT terpenuhi tanpa celah keraguan.

13.6 Daftar Pustaka (Pertemuan 13)

1. Anwar, Syamsul. (2018). *Hari-hari Allah: Dinamika Kalender Islam Kontemporer dan Menuju Kalender Hijriah Global*. Suara Muhammadiyah.
2. Ar-Razi, Fakhruddin. (1981). *Tafsir al-Fakhr ar-Razi al-Musyatar bi at-Tafsir al-Kabir wa Mafatih al-Ghaib*. Dar al-Fikr.
3. Majelis Tarjih dan Tajdid PP Muhammadiyah. (2024). *Pedoman Kalender Hijriah Global Tunggal (KHGT)*. PP Muhammadiyah.

4. Butkevich, A. G., & Lindegren, L. (2014). *Rigorous framework for relativistic astrometry*. *Astronomy & Astrophysics*.
5. Azhari, Susiknan. (2015). *Ilmu Falak: Perjumpaan Khazanah Islam dan Sains Modern*. Suara Muhammadiyah.

PERTEMUAN 14: Mesin Pemindai Visibilitas Global (*Global Analyzer*)

14.1 Teologi Sferis dan Kontinuitas Waktu Semesta

Setelah Gerbang 1 (Syarat Geosentris) tervalidasi pada Pertemuan 13, arsitektur Kalender Hijriah Global Tunggal (KHGT) melangkah pada fase observasional global. Bumi bukanlah hamparan datar yang mengalami siang dan malam secara serentak, melainkan sebuah bola (elipsoida) yang berputar, di mana garis waktu (*terminator* siang dan malam) terus merambat dari Timur ke Barat tanpa jeda.

Al-Qur'an mendeskripsikan mekanika sferis ini dengan pilihan diksi astronomis yang sangat akurat jauh sebelum era penjelajahan luar angkasa:

خَلَقَ السَّمَوَاتِ وَالْأَرْضَ بِالْحَقِّ يُكَوِّرُ اللَّيْلَ عَلَى النَّهَارِ وَيُكَوِّرُ النَّهَارَ عَلَى اللَّيْلِ وَسَخَّرَ الشَّمْسَ وَالْقَمَرَ كُلٌّ يَجْرِي لِأَجَلٍ مُّسَمًّى ۗ أَلَا هُوَ الْعَزِيزُ الْغَفُورُ

"Dia menciptakan langit dan bumi dengan (tujuan) yang benar; Dia menutupkan (yukawwir) malam atas siang dan menutupkan siang atas malam dan menundukkan matahari dan bulan, masing-masing berjalan menurut waktu yang ditentukan. Ingatlah Dialah Yang Maha Perkasa lagi Maha Pengampun." (QS. Az-Zumar [39]: 5)

Tafsir Ringkas: Imam Fakhruddin Ar-Razi dalam *Mafatih al-Ghaib* menyoroti secara khusus penggunaan kata *yukawwiru* (يُكَوِّرُ) yang berakar dari kata *al-kurrah* (bola/sesuatu yang melingkar). Proses "menutupkan" atau "menggulung" malam ke atas siang secara terus-menerus ini menjadi dalil fisis bahwa bumi berbentuk bulat. Berdasarkan epistemologi ini, Majelis Tarjih memandang bumi sebagai satu kesatuan (*matla' global*). Jika hilal telah wujud dan memenuhi kriteria di satu titik saja di belahan bumi manapun yang sedang mengalami pergulungan siang ke malam (waktu terbenam matahari), maka cahaya tersebut sudah sah menjadi penanda masuknya bulan baru bagi seluruh penduduk bumi yang terikat dalam satu sistem bola dunia.

14.2 Arsitektur Gerbang 2 KHGT: Imkanur Rukyat Global

Gerbang 2 KHGT mendiktekan bahwa setelah Ijtimak terjadi sebelum fajar di Selandia Baru, mesin komputasi harus mencari **minimal satu titik koordinat di permukaan bumi** di mana pada saat matahari terbenam pasca-Ijtimak, parameter hilal toposentrik mencapai batas:

1. **Tinggi Hilal Minimal:** 5 derajat di atas ufuk.
2. **Elongasi Minimal:** 8 derajat dari matahari.

Kriteria 5-8 derajat (Resolusi Istanbul 2016) ini lebih tinggi dibandingkan Neo MABIMS (3-6.4 derajat). Hal ini karena KHGT dirancang sebagai kalender universal yang mengayomi seluruh mazhab dan metodologi rukyat di seluruh dunia. Angka 5-8 derajat adalah parameter fisis di mana hilal tidak hanya "mungkin" terlihat oleh teleskop canggih, tetapi "sangat meyakinkan" untuk

dilihat dengan mata telanjang dalam kondisi atmosfer normal, sehingga menihilkan potensi penolakan dari kelompok *rukyyah puritan* di belahan bumi barat.

14.3 Algoritma Pemindai Matriks Bumi (*Grid Scanner*)

Tantangan komputasi utama pada tahap ini adalah skala. Permukaan bumi memiliki luas 510 juta kilometer persegi. Untuk menemukan satu titik visibilitas, kita tidak bisa hanya mengecek Makkah atau Jakarta. Kita harus menyapu seluruh garis bujur dan lintang dari Timur ke Barat.

Dalam ilmu komputer, pendekatan ini disebut *Brute-Force Grid Scanning*.

1. Mesin membagi peta bumi menjadi matriks/jaring koordinat (misal: jarak antar titik adalah 10 derajat lintang dan 15 derajat bujur).
2. Mesin melompat dari Garis Batas Tanggal Internasional (Bujur +180° di Pasifik), bergerak ke arah Barat menyusuri titik-titik bujur hingga kembali ke -180°.
3. Di setiap titik kordinat, mesin menghitung kapan detik *Sunset* (terbenam matahari) terjadi.
4. Pada detik *Sunset* tersebut, mesin mengekstraksi tinggi dan elongasi hilal.
5. Jika bernilai `True` (Tinggi ≥ 5 dan Elongasi ≥ 8), program seketika **BERHENTI (Break)**. Syarat KHGT terpenuhi secara global. Titik inilah yang disebut sebagai *Matla' Pertama*.

14.4 Source Code Mandiri: Mesin Sapu Jagat (*Global Analyzer*)

Simpan skrip Python ini dengan nama `khgt_global_analyzer.py` dan jalankan di terminal. Aplikasi ini dirancang menggunakan resolusi makro (lompatan 10 derajat lintang dan 15 derajat bujur) agar proses komputasi dapat selesai dalam hitungan detik hingga menit di komputer standar, tanpa memerlukan superkomputer.

```
Python
"""
Aplikasi: Mesin Sapu Jagat KHGT (Global Imkanur Rukyat Scanner)
Penulis: Kasmui
Deskripsi: Skrip pemindai matriks bumi secara asinkronus dari Timur ke
Barat
                untuk menemukan titik kordinat pertama yang memenuhi kriteria
                visibilitas global Istanbul (Tinggi 5°, Elongasi 8°).
"""

import datetime
import pytz
from skyfield.api import Loader, wgs84
from skyfield import almanac

def pemindai_global_khgt():
    print("="*75)
    print("MESIN SAPU JAGAT KHGT: GLOBAL ANALYZER (KRITERIA 5°- 8°)")
    print("="*75)

    # 1. Inisialisasi Pustaka
```

```

load = Loader('ephemeris_data', verbose=False)
ts = load.timescale()
planets = load('de421.bsp')
bumi = planets['earth']
matahari = planets['sun']
bulan = planets['moon']

# 2. Asumsi Waktu Eksekusi (Contoh: Pasca Ijtimak suatu bulan)
# Untuk simulasi ini, kita gunakan tanggal spesifik di mana Ijtimak
baru saja terjadi
# Misal: 8 April 2024 (Akhir Ramadan 1445 H menuju Syawal)
tahun, bln, hari = 2024, 4, 8
t0_dt = datetime.datetime(tahun, bln, hari, 0, 0, 0, tzinfo=pytz.utc)
t1_dt = t0_dt + datetime.timedelta(days=2)
t0 = ts.from_datetime(t0_dt)
t1 = ts.from_datetime(t1_dt)

# 3. Mencari Waktu Ijtimak Terakhir pada rentang tersebut
waktu_fase, indeks_fase = almanac.find_discrete(t0, t1,
almanac.moon_phases(planets))
t_ijtimak = None
for t_obj, f in zip(waktu_fase, indeks_fase):
    if f == 0:
        t_ijtimak = t_obj
        break

if not t_ijtimak:
    print("[!] Gagal mendeteksi Ijtimak pada rentang waktu yang
diberikan.")
    return

print(f"[REFERENSI] Detik Ijtimak :
{t_ijtimak.utc_datetime().strftime('%Y-%m-%d %H:%M:%S')} UTC")
print("[PROSES] Memulai pemindaian grid bumi dari Timur (+180°) ke
Barat (-180°)...")
print("      Target Parameter: Tinggi Toposentrik >= 5.0° DAN
Elongasi >= 8.0°\n")

# 4. Membangun Grid Matriks Bumi (Heuristic Macro-Grid)
# Lintang: -50 (Selatan) hingga +50 (Utara) -> area berpenghuni
dominan
# Bujur : +180 (Garis Batas Tanggal) turun ke -180
lintang_list = range(50, -51, -10)
bujur_list = range(180, -181, -15)

titik_ditemukan = False
lokasi_sukses = None
data_sukses = {}

# 5. Algoritma Sapu Jagat (Brute-Force Looping)
for lon in bujur_list:
    for lat in lintang_list:
        # Tetapkan titik pantau virtual
        pengamat_virtual = wgs84.latlon(lat, lon, elevation_m=0.0)

```

```

# Cari waktu Sunset hari ini di lokasi tersebut pasca Ijtimak
# Rentang pencarian: dari waktu Ijtimak hingga 24 jam ke depan
t_mulai_sunset = t_ijtimak
t_akhir_sunset = ts.from_datetime(t_ijtimak.utc_datetime() +
datetime.timedelta(hours=24))

f_ss = almanac.sunrise_sunset(planets, pengamat_virtual)
waktu_ss, transisi_ss = almanac.find_discrete(t_mulai_sunset,
t_akhir_sunset, f_ss)

t_sunset_lokal = None
for t_ss_obj, tr in zip(waktu_ss, transisi_ss):
    if tr == 0: # Transisi ke bawah ufuk (Sunset)
        t_sunset_lokal = t_ss_obj
        break

if not t_sunset_lokal:
    continue # Loncat ke titik berikutnya jika anomali (misal:
midnight sun di kutub)

# Ekstraksi Tinggi Toposentrik
obs_bulan =
pengamat_virtual.at(t_sunset_lokal).observe(bulan).apparent()
alt_bulan, _, _ = obs_bulan.altaz()
tinggi_hilal = alt_bulan.degrees

# Ekstraksi Elongasi Geosentris (Sesuai kaidah KHGT)
obs_pusat = bumi.at(t_sunset_lokal)
elongasi =
obs_pusat.observe(bulan).apparent().separation_from(
    obs_pusat.observe(matahari).apparent()).degrees

# 6. Evaluasi Gerbang Logika 5-8 Derajat
if tinggi_hilal >= 5.0 and elongasi >= 8.0:
    titik_ditemukan = True
    lokasi_sukses = (lat, lon)
    data_sukses = {
        'waktu_ss':
t_sunset_lokal.utc_datetime().strftime('%Y-%m-%d %H:%M:%S'),
        'tinggi': tinggi_hilal,
        'elongasi': elongasi
    }
    break # Hentikan loop lintang

if titik_ditemukan:
    break # Hentikan loop bujur (Kita hanya butuh 1 titik pertama
di dunia)

# 7. Output Kesimpulan
print("="*75)
if titik_ditemukan:
    print("[HASIL GERBANG 2 : MEMENUHI SYARAT GLOBAL KHGT]")

```

```

        print(f"Hilal telah memenuhi batas kriteria Istanbul di permukaan
bumi.")
        print(f"Lokasi Pertama (Matla') : Lintang {lokasi_sukses[0]}°,
Bujur {lokasi_sukses[1]}°")
        print(f"Waktu Sunset (UTC)      : {data_sukses['waktu_ss']} UTC")
        print(f"Tinggi Hilal           : {data_sukses['tinggi']:+.4f}°
(>= 5.0°)")
        print(f"Elongasi Geosentris      : {data_sukses['elongasi']:.4f}°
(>= 8.0°)")
        print("\nKESIMPULAN FINAL: BESOK ADALAH TANGGAL 1 BULAN BARU UNTUK
SELURUH DUNIA.")
    else:
        print("[HASIL GERBANG 2 : GAGAL (TIDAK MEMENUHI SYARAT)]")
        print("Tidak ditemukan satu titik pun di bumi (Lintang -50 sd +50)
yang")
        print("memiliki hilal dengan Tinggi >= 5° dan Elongasi >= 8° saat
sunset.")
        print("\nKESIMPULAN FINAL: ISTIKMAL. BULAN BERJALAN DIGENAPKAN 30
HARI UNTUK SELURUH DUNIA.")
        print("="*75)

if __name__ == '__main__':
    pemindai_global_khgt()

```

14.5 Penjelasan Anatomi Source Code

1. **Heuristic Macro-Grid (Langkah 4):** Komputasi tingkat observatorium biasanya menggunakan resolusi 1 derajat, menghasilkan 64.800 titik koordinat. Namun, program dengan skenario tersebut akan memakan waktu hingga beberapa jam untuk dieksekusi di prosesor *single-thread*. Algoritma di atas memodifikasi struktur dengan lompatan 15 derajat bujur dan 10 derajat lintang, sehingga iterasi direduksi menjadi hanya ratusan titik. Ini cukup presisi untuk membuktikan konsep kalender global tanpa membebani memori CPU secara berlebihan.
2. **Arah Sapuan Garis Waktu (Bujur Positif ke Negatif):** Perulangan `range(180, -181, -15)` didesain secara spesifik bergerak dari +180 ke arah angka minus. Secara fisis, ini meniru pergerakan bayangan malam yang menutupi bumi. Mesin memeriksa Selandia Baru, lalu Australia, lalu Asia, Timur Tengah, Eropa, hingga berakhir di benua Amerika (bujur negatif). Jika kriteria terpenuhi di Amerika, maka bulan baru jatuh esok harinya bagi seluruh dunia (konsep *Transfer Imkanur Rukyat*).
3. **Proteksi Midnight Sun (Langkah 5):** Kondisi `if not t_sunset_lokal: continue` adalah *error handler* yang cerdas. Di lintang ekstrem pada musim tertentu (misalnya lintang utara 70 derajat di bulan Juni), matahari tidak pernah tenggelam (*Midnight Sun*). Jika algoritma `find_discrete` tidak menemukan transisi terbenamnya matahari, mesin akan mengabaikan titik koordinat tersebut dan langsung melompat ke titik lintang berikutnya, memastikan aplikasi tidak *hang* atau *crash*.
4. **Efisiensi Break Ganda (Langkah 6):** KHGT hanya mensyaratkan "satu tempat" (*rukyyah fi makani ma*). Begitu kriteria `tinggi_hilal >= 5.0 and elongasi >= 8.0` terpenuhi, mesin melempar perintah `break` untuk keluar dari loop lintang, dan perintah `break` kedua

untuk keluar dari loop bujur. Mesin seketika berhenti mencari dan mendeklarasikan kemenangan.

14.6 Daftar Pustaka (Pertemuan 14)

1. Anwar, Syamsul. (2018). *Hari-hari Allah: Dinamika Kalender Islam Kontemporer dan Menuju Kalender Hijriah Global*. Suara Muhammadiyah.
2. Ar-Razi, Fakhruddin. (1981). *Tafsir al-Fakhr ar-Razi al-Musytahar bi at-Tafsir al-Kabir wa Mafatih al-Ghaib*. Dar al-Fikr.
3. Majelis Tarjih dan Tajdid PP Muhammadiyah. (2024). *Pedoman Kalender Hijriah Global Tunggal (KHGT)*. PP Muhammadiyah.
4. Meeus, J. (1998). *Astronomical Algorithms* (2nd ed.). Willmann-Bell, Inc.
5. Djamaluddin, Thomas. (2011). *Astronomi Memberi Solusi Penyatuan Ummat*. LAPAN.

PERTEMUAN 15: Rekayasa Antarmuka Interaktif (GUI) dan *Bidi-Converter* Kalender

15.1 Epistemologi Dakwah Digital dan Aksesibilitas Ilmu

Setelah mesin komputasi (backend) Kalender Hijriah Global Tunggal (KHGT) berhasil dibangun dan divalidasi presisinya pada tingkat terminal/konsol, tugas seorang astronom muslim belum selesai. Ilmu falak yang hanya bisa dibaca oleh mesin atau pemrogram tidak akan membawa dampak sosiologis yang luas. Mendesain Antarmuka Pengguna Grafis (*Graphical User Interface* / GUI) adalah bentuk purifikasi dakwah digital agar "kebijaksanaan" kosmik dapat diakses oleh masyarakat awam dengan mudah.

Al-Qur'an menggarisbawahi prinsip memberikan kemudahan (aksesibilitas) bagi umat manusia:

يُرِيدُ اللَّهُ بِكُمُ الْيُسْرَ وَلَا يُرِيدُ بِكُمُ الْعُسْرَ وَلِيُنْظِرُوا الْعِدَّةَ

“Allah menghendaki kemudahan bagimu, dan tidak menghendaki kesukaran bagimu. Dan hendaklah kamu mencukupkan bilangannya...” (QS. Al-Baqarah [2]: 185)

Tafsir Ringkas: Imam Fakhrudin Ar-Razi dalam *Mafatih al-Ghaib* saat menafsirkan ayat ini—yang kebetulan berada dalam konteks penentuan bilangan puasa bulan Ramadan (*litukmilu al-'iddah*)—menegaskan bahwa syariat Islam dibangun di atas fondasi rasionalitas yang menolak komplikasi yang tidak perlu (*al-'usr*). Dalam konteks rekayasa perangkat lunak, membiarkan pengguna terjebak pada layar terminal hitam yang rumit adalah sebuah "kesukaran". Mengembangkan GUI yang elegan, intuitif, dan *user-friendly* adalah manifestasi dari semangat *al-yusr* (kemudahan) untuk membantu umat secara presisi mengetahui jatuhnya waktu ibadah.

15.2 Arsitektur Antarmuka Modern (CustomTkinter)

Pustaka bawaan Python untuk GUI, yaitu `Tkinter`, secara fungsional sangat stabil namun memiliki kelemahan visual: tampilannya menyerupai perangkat lunak era Windows 95 yang kaku. Mengingat perangkat lunak astronomi sering digunakan pada malam hari (observasi hilal), antarmuka yang terlalu terang (putih) dapat merusak adaptasi gelap mata (*dark adaptation*).

Oleh karena itu, arsitektur visual kita akan dibangun menggunakan **CustomTkinter**. Ini adalah pustaka modifikasi tingkat tinggi yang memberikan kontrol penuh atas *Dark Mode* (Mode Gelap), sudut antarmuka yang melengkung (*rounded corners*), dan desain *flat-modern* yang memanjakan mata tanpa mengorbankan kecepatan eksekusi logika falak di belakang layar.

15.3 Algoritma *Bidi-Converter* dan Anomali Batas Hari

Konversi kalender Dua Arah (*Bidirectional* / *Bidi-Converter*) dari Masehi ke Hijriah dan sebaliknya bukanlah sekadar membagi jumlah hari. Tantangan terbesar dalam konversi ini adalah sinkronisasi **Batas Awal Hari**.

1. **Kalender Sipil (Masehi):** Hari berganti tepat di tengah malam (Pukul 00:00 waktu lokal).
2. **Kalender Kamariah (Hijriah):** Hari berganti pada saat **Maghrib** (matahari terbenam).

Jika pengguna memasukkan input "Masehi: 15 Ramadan 1445 H pada jam 19:00 WIB", mesin tidak boleh merender hasil yang sama dengan "15 Ramadan 1445 H pada jam 14:00 WIB". Karena jam 19:00 sudah melewati Maghrib, maka secara syar'i waktu tersebut sudah memasuki tanggal **16 Ramadan**. Algoritma harus mengekstrak data waktu Maghrib pada hari tersebut dan menjadikannya sebagai *pivot* (titik putar) penambahan tanggal.

15.4 Source Code Mandiri: Purwarupa Kalender GUI CustomTkinter

Simpan skrip Python ini dengan nama `gui_kalender_khgt.py` dan jalankan. Pastikan Anda telah menginstal pustaka GUI modern ini melalui terminal dengan perintah: `pip install customtkinter`.

Python

"""

Aplikasi: GUI Kalender KHGT & Bidi-Converter (CustomTkinter)

Penulis: Kasmui

Deskripsi: Purwarupa antarmuka grafis Mode Gelap (Dark Mode) untuk mengonversi

tanggal Masehi ke Hijriah dengan kompensasi pergantian hari
pasca-Maghrib menggunakan siklus fase bulan Skyfield.

"""

`import` datetime

`import` customtkinter `as` ctk

`import` pytz

`from` skyfield.api `import` Loader, wgs84

`from` skyfield `import` almanac

Konfigurasi Tema Global CustomTkinter

`ctk.set_appearance_mode("Dark")` # Mode observasi malam

`ctk.set_default_color_theme("blue")`

`class` KalenderKHGTApp(`ctk.CTk`):

`def` `__init__`(`self`):

`super`(`self`).`__init__`()

1. Konfigurasi Jendela Utama

`self.title("Bidi-Converter: Kalender KHGT Modern")`

`self.geometry("600x550")`

`self.resizable(False, False)`

Inisialisasi Engine Astronomi di latar belakang

`self.load_astronomy_engine()`

2. Arsitektur Antarmuka (Grid Layout)

`self.grid_columnconfigure(0, weight=1)`

`self.grid_rowconfigure(0, weight=0)`

Judul Aplikasi

```

        self.lbl_judul = ctk.CTkLabel(self, text="KONVERTER KALENDER
ASTRONOMI",
                                   font=ctk.CTkFont(family="Segoe UI",
size=22, weight="bold"))
        self.lbl_judul.grid(row=0, column=0, pady=(30, 10))

        self.lbl_sub = ctk.CTkLabel(self, text="Sinkronisasi Masehi -
Hijriah dengan Batas Maghrib",
                                   text_color="gray")
        self.lbl_sub.grid(row=1, column=0, pady=(0, 20))

        # Frame Input Data
        self.frame_input = ctk.CTkFrame(self, corner_radius=15)
        self.frame_input.grid(row=2, column=0, padx=40, pady=10,
sticky="ew")

        # Grid di dalam Frame Input
        self.frame_input.grid_columnconfigure((0, 1, 2), weight=1)

        ctk.CTkLabel(self.frame_input, text="Tanggal (DD):").grid(row=0,
column=0, padx=10, pady=(15,0), sticky="w")
        self.entry_tgl = ctk.CTkEntry(self.frame_input,
placeholder_text="Contoh: 15")
        self.entry_tgl.grid(row=1, column=0, padx=10, pady=(0, 15),
sticky="ew")

        ctk.CTkLabel(self.frame_input, text="Bulan (MM):").grid(row=0,
column=1, padx=10, pady=(15,0), sticky="w")
        self.entry_bln = ctk.CTkEntry(self.frame_input,
placeholder_text="Contoh: 08")
        self.entry_bln.grid(row=1, column=1, padx=10, pady=(0, 15),
sticky="ew")

        ctk.CTkLabel(self.frame_input, text="Tahun (YYYY):").grid(row=0,
column=2, padx=10, pady=(15,0), sticky="w")
        self.entry_thn = ctk.CTkEntry(self.frame_input,
placeholder_text="Contoh: 2026")
        self.entry_thn.grid(row=1, column=2, padx=10, pady=(0, 15),
sticky="ew")

        # Input Jam untuk Kompensasi Maghrib
        ctk.CTkLabel(self.frame_input, text="Jam Lokal
(HH:MM):").grid(row=2, column=0, columnspan=2, padx=10, pady=0,
sticky="w")
        self.entry_jam = ctk.CTkEntry(self.frame_input,
placeholder_text="Contoh: 18:30")
        self.entry_jam.grid(row=3, column=0, columnspan=3, padx=10,
pady=(0, 15), sticky="ew")

        # Tombol Eksekusi
        self.btn_konversi = ctk.CTkButton(self, text="PROSES KONVERSI
ASTRONOMIS",
                                   height=40,
font=ctk.CTkFont(weight="bold"),

```

```

                                command=self.eksekusi_konversi)
        self.btn_konversi.grid(row=3, column=0, padx=40, pady=20,
sticky="ew")

        # Frame Output Hasil
        self.frame_output = ctk.CTkFrame(self, corner_radius=15,
fg_color="#1e293b") # Dark slate
        self.frame_output.grid(row=4, column=0, padx=40, pady=10,
sticky="ew")

        self.lbl_hasil_masehi = ctk.CTkLabel(self.frame_output,
text="Masehi : -",
                                font=ctk.CTkFont(size=14))
        self.lbl_hasil_masehi.pack(padx=20, pady=(20, 5), anchor="w")

        self.lbl_hasil_hijriah = ctk.CTkLabel(self.frame_output,
text="Hijriah : -",
                                font=ctk.CTkFont(size=18,
weight="bold"), text_color="#38bdf8")
        self.lbl_hasil_hijriah.pack(padx=20, pady=5, anchor="w")

        self.lbl_status = ctk.CTkLabel(self.frame_output, text="Status : -
",
                                font=ctk.CTkFont(size=12,
slant="italic"), text_color="#94a3b8")
        self.lbl_status.pack(padx=20, pady=(5, 20), anchor="w")

def load_astronomy_engine(self):
    """Memuat data ephemeris di memori secara asinkronus virtual"""
    self.load = Loader('ephemeris_data', verbose=False)
    self.ts = self.load.timescale()
    self.planets = self.load('de421.bsp')

    # Koordinat referensi (Contoh: Yogyakarta untuk batas Maghrib)
    self.zona_waktu = pytz.timezone('Asia/Jakarta')
    self.lokasi_ref = wgs84.latlon(-7.7956, 110.3695, elevation_m=113)

def eksekusi_konversi(self):
    """Event Loop yang dipicu saat tombol ditekan"""
    try:
        # Mengambil data dari Entry Widget
        hari = int(self.entry_tgl.get())
        bulan = int(self.entry_bln.get())
        tahun = int(self.entry_thn.get())
        jam_menit = self.entry_jam.get()

        if ":" in jam_menit:
            jam, menit = map(int, jam_menit.split(':'))
        else:
            jam, menit = 12, 0 # Default tengah hari jika kosong

        # Validasi Waktu Sipil
        dt_input = self.zona_waktu.localize(datetime.datetime(tahun,
bulan, hari, jam, menit, 0))

```

```

except ValueError:
    self.lbl_status.configure(text="Error: Format input tidak
valid. Gunakan angka.", text_color="#ef4444")
    return

    # Tampilkan Input Masehi
    self.lbl_hasil_masehi.configure(text=f"Masehi :
{dt_input.strftime('%d %B %Y - Pukul %H:%M')} WIB")
    self.lbl_status.configure(text="Sedang memproses algoritma fase
bulan...", text_color="#facc15")
    self.update() # Paksa GUI refresh

    # KOMPUTASI: Mencari Waktu Maghrib pada hari tersebut
    t0_ss = self.ts.from_datetime(dt_input.replace(hour=10, minute=0,
second=0))
    t1_ss = self.ts.from_datetime(dt_input.replace(hour=23, minute=59,
second=59))

    f_ss = almanac.sunrise_sunset(self.planets, self.lokasi_ref)
    waktu_ss, transisi_ss = almanac.find_discrete(t0_ss, t1_ss, f_ss)

    waktu_maghrib = None
    for t_obj, tr in zip(waktu_ss, transisi_ss):
        if tr == 0:
            waktu_maghrib =
t_obj.utc_datetime().replace(tzinfo=pytz.utc).astimezone(self.zona_waktu)
            break

    # KOMPUTASI: Mencari Umur Bulan dari Ijtimak Terakhir
    t_target = self.ts.from_datetime(dt_input)
    t_mundur = self.ts.from_datetime(dt_input -
datetime.timedelta(days=30))

    waktu_fase, indeks_fase = almanac.find_discrete(t_mundur,
t_target, almanac.moon_phases(self.planets))
    t_ijtimak = None

    # Ambil Ijtimak yang terakhir sebelum tanggal target
    for t_obj, f in zip(waktu_fase, indeks_fase):
        if f == 0:
            t_ijtimak = t_obj

    if not t_ijtimak:
        self.lbl_status.configure(text="Error: Gagal mendeteksi
Ijtimak.")
        return

    # Konversi Umur Bulan menjadi Tanggal Hijriah (Pendekatan
Matematis GUI)
    selisih_hari = (t_target.tt - t_ijtimak.tt)
    tanggal_hijriah = int(selisih_hari) + 1 # +1 karena hari ijtimak
adalah pergantian

```

```

# LOGIKA KOMPENSASI MAGHRIB
status_teks = "Waktu Siang (Belum melampaui batas Maghrib)."
if waktu_maghrib and dt_input >= waktu_maghrib:
    tanggal_hijriah += 1
    status_teks = f"Melampaui Maghrib
({waktu_maghrib.strftime('%H:%M')}). Tanggal Hijriah +1 (Malam Hari)."

# Render Hasil ke GUI
# Catatan: Nama bulan disederhanakan sebagai representasi kalender
urutan
    nama_bulan_hijriah = ["Muharram", "Safar", "Rabi'ul Awal",
"Rabi'ul Akhir",
                                "Jumadil Awal", "Jumadil Akhir", "Rajab",
"Sya'ban",
                                "Ramadan", "Syawal", "Zulkaidah",
"Zulhijah"]

# Prediksi bulan dan tahun (memerlukan lookup tabel KHGT untuk
presisi absolut,
# ini adalah purwarupa konversi berdasarkan umur fase)

self.lbl_hasil_hijriah.configure(text=f"Hijriah : Tanggal Ke-
{tanggal_hijriah} sejak Fase Bulan Baru")
self.lbl_status.configure(text=f"Status : {status_teks}",
text_color="#10b981")

if __name__ == "__main__":
    app = KalenderKHGTApp()
    app.mainloop()

```

15.5 Penjelasan Anatomi Source Code

1. **Arsitektur *Event-Driven* (mainloop):** Berbeda dengan skrip terminal yang berjalan linier dari atas ke bawah lalu mati, aplikasi antarmuka grafis berjalan di dalam sebuah pusran abadi yang disebut `mainloop()`. Mesin GUI terus bersiaga (stanby), menunggu *event* atau pemicu, seperti klik mouse pada tombol `btn_konversi`.
2. **customtkinter dan Ergonomi Visual:** Penggunaan `ctk.CTkFrame` dengan parameter `corner_radius=15` menggantikan desain kotak bersudut tajam yang kuno. Pemilihan palet warna hex (`#1e293b` untuk *background* gelap dan `#38bdf8` untuk teks sorotan) dirancang khusus untuk mengurangi emisi *blue-light*, menjaga pupil mata astronom tetap melebar saat menatap layar komputer di lapangan observasi yang gelap.
3. **Metode Pengamanan Input (try...except):** Pada terminal, kesalahan *typo* huruf akan membuat program langsung *crash* (menutup paksa). Dalam GUI, hal itu diharamkan. Blok `try...except ValueError` akan menangkap kesalahan (*exception*) jika pengguna memasukkan huruf "April" ke dalam kotak yang meminta angka bulan "04", lalu merender pesan error halus di label `lbl_status` berwarna merah (`#ef4444`) tanpa membunuh aplikasi.
4. **Logika Kompensasi Maghrib:** Algoritma membaca input `jam_menit`. Ia memutar mesin `almanac.sunrise_sunset` untuk hari spesifik tersebut. Jika waktu input pengguna (misal: jam 19:00) lebih besar/sama dengan waktu *Sunset* yang dihitung oleh mesin (misal: jam

17:45), maka variabel `tanggal_hijriah` secara otomatis diinjeksi dengan `+= 1`. Ini adalah representasi digital dari kaidah fikih pergeseran hari yang absolut.

15.6 Daftar Pustaka (Pertemuan 15)

1. Ar-Razi, Fakhruddin. (1981). *Tafsir al-Fakhr ar-Razi al-Musytahar bi at-Tafsir al-Kabir wa Mafatih al-Ghaib*. Dar al-Fikr.
2. Schüßler, T. (2023). *CustomTkinter: A modern and customizable python UI-library based on Tkinter*. Documentation.
3. Majelis Tarjih dan Tajdid PP Muhammadiyah. (2024). *Pedoman Kalender Hijriah Global Tunggal (KHGT)*. PP Muhammadiyah.
4. Sweigart, A. (2015). *Automate the Boring Stuff with Python: Practical Programming for Total Beginners*. No Starch Press. (Bab 12: *GUI Automation*).

PERTEMUAN 16: Ujian Akhir Semester (UAS) – Sintesis Total dan Distribusi Biner (PyInstaller)

16.1 Epistemologi Amal Jariah dan Distribusi Teknologi

Pertemuan ke-16 ini menandai puncak dari perjalanan intelektual kita dalam mengintegrasikan keagungan syariat dengan presisi komputasi modern. Dalam epistemologi Islam, ilmu yang tidak didistribusikan dan diamankan ibarat pohon yang tidak berbuah. Perangkat lunak (software) astronomi yang telah kita bangun lapis demi lapis bukanlah sekadar tugas akademik, melainkan bentuk *wakaf digital* atau amal jariah kontemporer yang kemanfaatannya dapat menembus batas ruang dan waktu.

Nabi Muhammad ﷺ bersabda mengenai keberlanjutan ilmu:

إِذَا مَاتَ الْإِنْسَانُ انْقَطَعَ عَمَلُهُ إِلَّا مِنْ ثَلَاثَةٍ: صَدَقَةٍ جَارِيَةٍ، أَوْ عِلْمٍ يُنْتَفَعُ بِهِ، أَوْ وَلَدٍ صَالِحٍ يَدْعُو لَهُ

"Apabila seorang manusia meninggal dunia, terputuslah amalnya kecuali dari tiga hal: sedekah jariah, ilmu yang bermanfaat, atau anak salih yang mendoakannya." (HR. Muslim No. 1631)

Penjelasan Ringkas: Dalam konteks rekayasa perangkat lunak, "ilmu yang bermanfaat" (*ilmun yuntafa'u bihi*) bertransformasi dari sekadar lembaran kitab menjadi algoritma biner yang terus bekerja. Setiap kali seorang muslim di pelosok bumi menjalankan aplikasi yang Anda rancang untuk menemukan arah kiblat yang presisi atau mengetahui jadwal salat yang akurat, maka pahala komputasi tersebut akan terus mengalir. Oleh karena itu, tugas akhir (UAS) ini menuntut Anda untuk tidak hanya membuat kode yang "bisa jalan di komputer pemrogram", tetapi mengemasnya menjadi aplikasi biner yang siap didistribusikan ke masyarakat awam.

16.2 Arsitektur Sintesis: "Mini KHGT Times"

Pada fase Ujian Akhir Semester ini, mahasiswa ditantang untuk membangun purwarupa aplikasi desktop bernama "Mini KHGT Times". Aplikasi ini adalah sintesis dari seluruh modul yang telah dipelajari dari Pertemuan 1 hingga 15.

Tantangan arsitekturalnya meliputi:

1. **Multi-Tab Interface:** Menggunakan elemen `TabView` pada antarmuka `CustomTkinter` untuk menggabungkan modul Arah Kiblat, Jadwal Salat, dan Konverter KHGT ke dalam satu jendela yang elegan.
2. **Manajemen Memori Tunggal:** Matriks NASA JPL `de421.bsp` berukuran 16 MB. Aplikasi yang baik tidak akan memuat (*load*) file ini berkali-kali di setiap tab. Ephemeris harus dimuat satu kali di saat inisialisasi aplikasi berjalan (*singleton pattern*).

3. **Kompilasi Biner (Executable):** Masyarakat awam tidak menginstal Python di laptop mereka. Kode sumber (.py) wajib di-kompilasi (*build*) menjadi file biner eksekusi mandiri (.exe untuk Windows atau file biner ELF untuk Linux) menggunakan pustaka **PyInstaller**.

16.3 Source Code Mandiri: Proyek Akhir "Mini KHGT Times"

Simpan skrip Python ini dengan nama `mini_khgt_times.py`. Skrip ini mengimpor dan menyatukan seluruh logika Falak ke dalam satu antarmuka ber-tab (Tabview).

Python

```

"""
Aplikasi: Mini KHGT Times (Proyek UAS - Sintesis Total)
Penulis: Kasmui
Deskripsi: Aplikasi GUI berbasis CustomTkinter yang mengintegrasikan
Arah Kiblat, Jadwal Salat, dan Konverter KHGT dalam satu
jendela.
Didesain untuk dikompilasi menjadi file biner .exe.
"""

import math
import datetime
import pytz
import customtkinter as ctk
from skyfield.api import Loader, wgs84
from skyfield import almanac

# Konfigurasi Tema UI
ctk.set_appearance_mode("Dark")
ctk.set_default_color_theme("blue")

class MiniKHGTApp(ctk.CTk):
    def __init__(self):
        super().__init__()

        self.title("Mini KHGT Times - Sistem Falak Terpadu")
        self.geometry("650x550")

        # 1. Inisialisasi Engine Astronomi (Memori Tunggal)
        self.lbl_loading = ctk.CTkLabel(self, text="Memuat Engine
Astronomi NASA JPL...", text_color="#facc15")
        self.lbl_loading.pack(pady=20)
        self.update() # Render teks loading sebelum proses berat dimulai

        self.load = Loader('ephemeris_data', verbose=False)
        self.ts = self.load.timescale()
        self.planets = self.load('de421.bsp')

        self.lbl_loading.destroy() # Hapus teks loading setelah selesai

        # 2. Arsitektur Tabview
        self.tabview = ctk.CTkTabview(self, width=600, height=500,
corner_radius=10)
        self.tabview.pack(padx=20, pady=20, fill="both", expand=True)

```

```

self.tab_kiblat = self.tabview.add("Arah Kiblat")
self.tab_salat = self.tabview.add("Jadwal Salat")
self.tab_khgt = self.tabview.add("Konverter KHGT")

self.setup_tab_kiblat()
self.setup_tab_salat()
self.setup_tab_khgt()

# =====
# MODUL 1: ARAH KIBLAT
# =====
def setup_tab_kiblat(self):
    ctk.CTkLabel(self.tab_kiblat, text="Kalkulator Arah Kiblat
(Geodesi Sferis)", font=("Segoe UI", 16, "bold")).pack(pady=10)

    frame_input = ctk.CTkFrame(self.tab_kiblat)
    frame_input.pack(pady=10, padx=20, fill="x")

    ctk.CTkLabel(frame_input, text="Lintang (Desimal):").grid(row=0,
column=0, padx=10, pady=10)
    self.k_lat = ctk.CTkEntry(frame_input, placeholder_text="Contoh: -
6.9932")
    self.k_lat.grid(row=0, column=1, padx=10, pady=10)

    ctk.CTkLabel(frame_input, text="Bujur (Desimal):").grid(row=1,
column=0, padx=10, pady=10)
    self.k_lon = ctk.CTkEntry(frame_input, placeholder_text="Contoh:
110.4203")
    self.k_lon.grid(row=1, column=1, padx=10, pady=10)

    ctk.CTkButton(self.tab_kiblat, text="Hitung Azimut",
command=self.hitung_kiblat).pack(pady=10)

    self.k_hasil = ctk.CTkLabel(self.tab_kiblat, text="Hasil: -",
font=("Segoe UI", 18, "bold"), text_color="#38bdf8")
    self.k_hasil.pack(pady=20)

def hitung_kiblat(self):
    try:
        lat = float(self.k_lat.get())
        lon = float(self.k_lon.get())

        # Konstanta Ka'bah
        phi_k = math.radians(21.422487)
        lam_k = math.radians(39.826206)
        phi_l = math.radians(lat)
        lam_l = math.radians(lon)

        delta_lam = lam_k - lam_l
        y = math.sin(delta_lam)
        x = (math.cos(phi_l) * math.tan(phi_k)) - (math.sin(phi_l) *
math.cos(delta_lam))

```

```

        azimut = math.degrees(math.atan2(y, x))
        if azimut < 0: azimut += 360.0

        self.k_hasil.configure(text=f"Azimut Kiblat: {azimut:.4f}°
dari Utara Sejati")
        except ValueError:
            self.k_hasil.configure(text="Error: Masukkan angka valid!",
text_color="#ef4444")

# =====
# MODUL 2: JADWAL SALAT (ZUHUR & MAGHRIB)
# =====
def setup_tab_salat(self):
    ctk.CTkLabel(self.tab_salat, text="Pemindai Kulminasi & Sunset",
font=("Segoe UI", 16, "bold")).pack(pady=10)
    ctk.CTkLabel(self.tab_salat, text="Menggunakan koordinat Lintang &
Bujur dari Tab Kiblat.", text_color="gray").pack()

    ctk.CTkButton(self.tab_salat, text="Proses Jadwal Hari Ini",
command=self.hitung_salat).pack(pady=20)

    self.s_zuhur = ctk.CTkLabel(self.tab_salat, text="Zuhur (Transit)
: -", font=("Consolas", 16))
    self.s_zuhur.pack(pady=10)

    self.s_maghrib = ctk.CTkLabel(self.tab_salat, text="Maghrib
(Sunset): -", font=("Consolas", 16))
    self.s_maghrib.pack(pady=10)

def hitung_salat(self):
    try:
        lat = float(self.k_lat.get())
        lon = float(self.k_lon.get())
        tz = pytz.timezone('Asia/Jakarta') # Default hardcoded untuk
UAS

        sekarang = datetime.datetime.now(tz)
        pengamat = wgs84.latlon(lat, lon)

        t0 = self.ts.from_datetime(sekarang.replace(hour=0, minute=0,
second=0))
        t1 = self.ts.from_datetime(sekarang.replace(hour=23,
minute=59, second=59))

        # Zuhur
        f_transit = almanac.meridian_transits(self.planets,
self.planets['sun'], pengamat)
        w_transit, j_transit = almanac.find_discrete(t0, t1,
f_transit)
        idx = list(j_transit).index(1)
        t_zuhur =
w_transit[idx].utc_datetime().replace(tzinfo=pytz.utc).astimezone(tz)

        # Maghrib
        f_ss = almanac.sunrise_sunset(self.planets, pengamat)

```

```

w_ss, j_ss = almanac.find_discrete(t0, t1, f_ss)
t_maghrib = None
for w, j in zip(w_ss, j_ss):
    if j == 0:
        t_maghrib =
w.utc_datetime().replace(tzinfo=pytz.utc).astimezone(tz)
    break

self.s_zuhur.configure(text=f"Zuhur (Transit) :
{t_zuhur.strftime('%H:%M:%S')} WIB")
    if t_maghrib:
        self.s_maghrib.configure(text=f"Maghrib (Sunset):
{t_maghrib.strftime('%H:%M:%S')} WIB")
    except ValueError:
        self.s_zuhur.configure(text="Harap isi Lintang & Bujur di Tab
Kiblat terlebih dahulu.")

# =====
# MODUL 3: INFO KHGT
# =====
def setup_tab_khgt(self):
    ctk.CTkLabel(self.tab_khgt, text="Status KHGT (Konjungsi Global)",
font=("Segoe UI", 16, "bold")).pack(pady=10)
    ctk.CTkLabel(self.tab_khgt, text="Modul ini menampilkan status
Ijtimak geosentris terakhir.", text_color="gray").pack()

    ctk.CTkButton(self.tab_khgt, text="Cek Ijtimak Terakhir",
command=self.cek_ijtimak).pack(pady=20)

    self.khgt_hasil = ctk.CTkLabel(self.tab_khgt, text="-",
font=("Consolas", 14), text_color="#10b981")
    self.khgt_hasil.pack(pady=10)

def cek_ijtimak(self):
    sekarang = datetime.datetime.now(pytz.utc)
    t0 = self.ts.from_datetime(sekarang - datetime.timedelta(days=30))
    t1 = self.ts.from_datetime(sekarang)

    w_fase, j_fase = almanac.find_discrete(t0, t1,
almanac.moon_phases(self.planets))
    t_ijtimak = None
    for w, j in zip(w_fase, j_fase):
        if j == 0: t_ijtimak = w

    if t_ijtimak:
        waktu_utc = t_ijtimak.utc_datetime()
        umur_hari = (self.ts.now().tt - t_ijtimak.tt)

        laporan = f"Waktu Ijtimak UTC : {waktu_utc.strftime('%d-%b-%Y
%H:%M:%S')}\n"
        laporan += f"Umur Bulan Global : {umur_hari:.2f} Hari sejak
Konjungsi"
        self.khgt_hasil.configure(text=laporan)

```

```
if __name__ == "__main__":
    app = MiniKHGTApp()
    app.mainloop()
```

16.4 Distribusi Biner: Menggunakan PyInstaller

Aplikasi Python `.py` membutuhkan *interpreter* Python untuk berjalan. Agar aplikasi "Mini KHGT Times" dapat dibagikan kepada takmir masjid atau mahasiswa lain yang menggunakan sistem operasi Windows tanpa harus menginstal Python, kita menggunakan pustaka `PyInstaller`.

Langkah-langkah Kompilasi (Build):

1. Buka Terminal/Command Prompt.
2. Instal pustaka pembuat biner:

```
pip install pyinstaller
```

3. Arahkan terminal ke direktori tempat Anda menyimpan file `mini_khgt_times.py`.
4. Eksekusi perintah sakti berikut:

```
pyinstaller --noconsole --onefile mini_khgt_times.py
```

Penjelasan Parameter Command Line:

- `pyinstaller`: Memanggil compiler biner.
- `--noconsole`: Mencegah munculnya jendela terminal hitam berkedip di belakang layar aplikasi saat GUI dijalankan. Sangat penting untuk estetika aplikasi desktop modern.
- `--onefile`: Membungkus seluruh kode Python, pustaka *CustomTkinter*, *Skyfield*, dan dependensi lainnya ke dalam **satu file tunggal** berformat `.exe`.

Catatan Krusial Ephemeric:

File data ephemeric `de421.bsp` tidak akan otomatis masuk ke dalam `.exe`. Anda harus menyertakan folder `ephemeris_data` dalam satu folder yang sama dengan file `.exe` yang dihasilkan (di dalam folder `dist/`), agar saat dijalankan, aplikasi tetap dapat membaca data matriks JPL.

16.5 Penjelasan Anatomi Source Code dan Integrasi

1. **Arsitektur OOP Penuh:** `class MiniKHGTApp(ctk.CTk)` membungkus seluruh variabel global ke dalam `self`. Hal ini memastikan bahwa `self.ts` (Timescale) dan `self.planets` (Ephemeric) dapat diakses secara instan oleh seluruh tab (Kiblat, Salat, KHGT) tanpa perlu melakukan *loading* berulang kali. Ini adalah bentuk *Itqan* (efisiensi) dalam manajemen *Random Access Memory* (RAM).
2. **Transmisi Data Antar-Tab:** Perhatikan bagaimana fungsi `hitung_salat(self)` pada Modul 2 tidak meminta input lintang dan bujur lagi. Ia secara cerdas mengambilnya

langsung dari variabel antarmuka di tab pertama (`self.k_lat.get()`). Ini menciptakan *User Experience* (UX) yang berkesinambungan. Pengguna cukup mendefinisikan lokasinya satu kali, dan seluruh ekosistem astronomi menyesuaikan diri.

3. **Modularitas Fungsi Falak:** Algoritma yang semula kompleks dan panjang pada pertemuan 7 (Waktu Salat) dan 10 (Ijtimak) kini diringkas ke dalam blok-blok metode yang sangat padat (`almanac.find_discrete`). Ini mendemonstrasikan bahwa setelah fondasi epistemologi dipahami secara teoritis, Python mampu menyembunyikan kerumitan fisis ke dalam mesin komputasi yang bersih dan reaktif.

16.6 Daftar Pustaka (Pertemuan 16)

1. Anwar, Syamsul. (2018). *Hari-hari Allah: Dinamika Kalender Islam Kontemporer dan Menuju Kalender Hijriah Global*. Suara Muhammadiyah.
2. Ar-Razi, Fakhruddin. (1981). *Tafsir al-Fakhr ar-Razi al-Musytahar bi at-Tafsir al-Kabir wa Mafatih al-Ghaib*. Dar al-Fikr.
3. Cortesi, A. (2022). *PyInstaller Documentation*. PyInstaller Development Team.
4. Majelis Tarjih dan Tajdid PP Muhammadiyah. (2024). *Pedoman Kalender Hijriah Global Tunggal (KHGT)*. PP Muhammadiyah.
5. Schüßler, T. (2023). *CustomTkinter: A modern and customizable python UI-library based on Tkinter*. Documentation.

DAFTAR PUSTAKA

- Anwar, Syamsul. (2018). *Hari-hari Allah: Dinamika Kalender Islam Kontemporer dan Menuju Kalender Hijriah Global*. Suara Muhammadiyah.
- Ar-Razi, Fakhruddin. (1981). *Tafsir al-Fakhr ar-Razi al-Musytahar bi at-Tafsir al-Kabir wa Mafatih al-Ghaib*. Dar al-Fikr.
- Azhari, Susiknan. (2015). *Ilmu Falak: Perjumpaan Khazanah Islam dan Sains Modern*. Suara Muhammadiyah.
- Butkevich, A. G., & Lindegren, L. (2014). *Rigorous framework for relativistic astrometry*. *Astronomy & Astrophysics*.
- Cortesi, A. (2022). *PyInstaller Documentation*. PyInstaller Development Team.
- Djamaluddin, Thomas. (2011). *Astronomi Memberi Solusi Penyatuan Ummat*. LAPAN.
- Djamaluddin, Thomas. (2022). *Menyatukan Hari Raya: Dari Kriteria 2-3-8 Menuju Rekomendasi Jakarta 2017*. LAPAN-BRIN.
- Downey, A. (2015). *Think Python: How to Think Like a Computer Scientist* (2nd ed.). O'Reilly Media, Inc.
- Hughes, D. W., Yallop, B. D., & Hohenkerk, C. Y. (1989). *The Equation of Time*. *Monthly Notices of the Royal Astronomical Society*, 238(4), 1529-1535.
- Hunter, J. D. (2007). *Matplotlib: A 2D graphics environment*. *Computing in Science & Engineering*, 9(3), 90-95.
- Kementerian Agama Republik Indonesia. (2021). *Kriteria Baru MABIMS (Tinggi 3 Derajat, Elongasi 6.4 Derajat) dalam Penentuan Awal Bulan Hijriah*. Dirjen Bimas Islam Kemenag RI.
- Majelis Tarjih dan Tajdid PP Muhammadiyah. (2009). *Pedoman Hisab Muhammadiyah*. Majelis Tarjih dan Tajdid PP Muhammadiyah.
- Majelis Tarjih dan Tajdid PP Muhammadiyah. (2020). *Keputusan Musyawarah Nasional Tarjih ke-31 tentang Kriteria Waktu Subuh*. PP Muhammadiyah.
- Majelis Tarjih dan Tajdid PP Muhammadiyah. (2024). *Pedoman Kalender Hijriah Global Tunggal (KHGT)*. PP Muhammadiyah.
- Meeus, J. (1998). *Astronomical Algorithms* (2nd ed.). Willmann-Bell, Inc.

Morrison, L. V., & Stephenson, F. R. (2004). *Historical Values of the Earth's Clock Error Delta-T and the Calculation of Eclipses*. *Journal for the History of Astronomy*.

Odeh, M. S. (2004). *New Criterion for Lunar Crescent Visibility*. *Experimental Astronomy*, 18(1), 39-64.

Rhodes, B. (2019). *Skyfield: High Precision Research-Grade Positions for Planets and Earth Satellites Explorer*. Astrophysics Source Code Library.

Saksono, Tono. (2007). *Mengkompromikan Rukyat & Hisab*. Amythas Publicita.

Schüßler, T. (2023). *CustomTkinter: A modern and customizable python UI-library based on Tkinter*. Documentation.

Smart, W. M. (1977). *Textbook on Spherical Astronomy* (6th ed.). Cambridge University Press.

Sweigart, A. (2015). *Automate the Boring Stuff with Python: Practical Programming for Total Beginners*. No Starch Press.

Urban, S. E., & Seidelmann, P. K. (Eds.). (2012). *Explanatory Supplement to the Astronomical Almanac* (3rd ed.). University Science Books.

VanderPlas, J. (2016). *Python Data Science Handbook: Essential Tools for Working with Data*. O'Reilly Media, Inc.

GLOSARIUM

Berikut adalah **Glosarium (Daftar Istilah)** komprehensif dari buku *Silabus Perkuliahan: Python untuk Astronomi Islam*, yang merangkum persilangan antara terminologi Ilmu Falak, Fisika Astronomi, dan Ilmu Komputer (Pemrograman Python) dari huruf A hingga Z:

A

- **Algoritma:** Rangkaian instruksi logis dan matematis langkah demi langkah dalam bahasa pemrograman (seperti Python) untuk menyelesaikan masalah komputasi falak.
- **Apparent (Tampak):** Posisi benda langit sebagaimana yang *terlihat* oleh mata pengamat di bumi, setelah dikoreksi dari efek paralaks dan refraksi atmosfer (berbeda dengan posisi geometris murni).
- **Asensio Rekta (Right Ascension):** Jarak sudut benda langit diukur ke arah timur dari titik ekuinoks musim semi (Titik Aries) di sepanjang ekuator langit; ekuivalen dengan garis bujur pada peta bola bumi.
- **Azimut:** Sudut putaran horizontal yang diukur searah jarum jam dari titik Utara Sejati (0°) menuju objek yang diamati (misalnya arah Kiblat).

B

- **Bidi-Converter (Bidirectional):** Konverter dua arah dalam perangkat lunak yang mampu mengubah tanggal Masehi ke Hijriah dan sebaliknya secara dinamis.
- **Brute-Force (Grid Scanning):** Metode komputasi yang memeriksa setiap titik koordinat di permukaan bumi secara berurutan untuk mencari matriks visibilitas hilal tanpa terkecuali.
- **Bujur (Longitude):** Garis imajiner vertikal yang mengukur jarak sudut ke arah Timur atau Barat dari Meridian Utama (Greenwich). 1 derajat bujur setara dengan perbedaan waktu 4 menit.

C

- **CustomTkinter:** Pustaka (*library*) antarmuka grafis (GUI) modern di Python yang mendukung mode gelap (*dark mode*) dan desain kontemporer untuk aplikasi astronomi desktop.

D

- **Deklinasi:** Jarak sudut vertikal matahari atau bulan diukur dari ekuator langit ke arah utara (+) atau selatan (-). Ekuivalen dengan garis lintang di bumi.
- **Delta-T (ΔT):** Nilai koreksi waktu (dalam detik) yang merupakan selisih antara waktu atom mekanis (*Terrestrial Time*) dan waktu rotasi bumi riil (*Universal Time*) yang semakin melambat.

- **Decision Tree (Pohon Keputusan):** Struktur algoritma pemrograman menggunakan gerbang logika (IF/ELSE/AND) untuk mengevaluasi syarat-syarat visibilitas hilal seperti kriteria Neo MABIMS.

E

- **Ekliptika:** Bidang lintasan orbit bumi saat mengelilingi matahari.
- **Elongasi:** Jarak sudut sferis antara pusat piringan bulan dan pusat piringan matahari. Menjadi syarat utama terbentuknya cahaya hilal (Limit Danjon menetapkan minimal 7 derajat).
- **Ephemeris (Efemeris):** Kumpulan data matriks presisi tinggi yang memetakan posisi vektor benda langit dari waktu ke waktu (seperti file `de421.bsp` dari NASA JPL).
- **Equation of Time (Perata Waktu):** Selisih waktu matematis antara jam matahari sejati (bayangan fisik) dengan jam matahari rata-rata (arloji) akibat bentuk elips orbit bumi dan kemiringan sumbu bumi.

F

- **Fajar Shadiq:** Pendaran cahaya horizontal yang menyebar di ufuk timur sebagai akibat hamburan sinar matahari di atmosfer saat pusat matahari berada pada sudut depresi -18 derajat (Tanda masuknya waktu Subuh).
- **Find_discrete:** Algoritma pencarian akar (*root-finding*) tingkat lanjut dalam pustaka Skyfield Python yang secara efisien memindai waktu transisi astronomis (seperti Ijtimak atau *Sunset*) ke tingkat presisi milidetik.
- **Fraksi Iluminasi:** Persentase luasan piringan bulan yang tersinari oleh matahari dan menghadap bumi.

G

- **Geodesi Sferis (Spherical Trigonometry):** Ilmu ukur segitiga pada permukaan bola (seperti bumi) yang digunakan untuk menghitung rute terpendek (*Great Circle*) penentu arah Kiblat.
- **Geosentris:** Sistem referensi koordinat astronomi yang diukur tepat dari titik pusat massa bumi (digunakan untuk menentukan Ijtimak mutlak).
- **GUI (Graphical User Interface):** Antarmuka visual perangkat lunak (menggunakan tombol, kotak teks, dan grafik) yang menjembatani interaksi antara pengguna manusia dan logika kode Python di balik layar.

H

- **Hilal:** Penampakan bulan sabit pertama kalinya pasca-Ijtimak setelah terbenamnya matahari, yang menandai masuknya tanggal 1 bulan Hijriah baru.
- **Hisab Hakiki:** Metode komputasi astronomi Islam modern yang menggunakan data mekanika langit faktual dan mempertimbangkan perturbasi (gangguan) orbit, bukan sekadar perkiraan rata-rata.

- **Hour Angle (Sudut Jam):** Jarak sudut benda langit dari meridian lokal pengamat, diukur dalam satuan jam atau derajat (1 jam = 15 derajat).

I

- **Ijtimak (Konjungsi):** Peristiwa ketika bujur ekliptika geosentris bulan dan matahari sejajar/sama persis (Selisih 0 derajat). Pertanda berakhirnya siklus bulan kamariah yang lama.
- **Ikhtiyat:** Waktu pengaman (biasanya 1 hingga 2 menit) yang ditambahkan pada hasil hisab waktu salat untuk memastikan kehati-hatian (*sadd adz-dzari'ah*) dalam beribadah.
- **Imkanur Rukyat:** Kriteria visibilitas fisis hilal (batas kemungkinan bulan dapat dilihat oleh mata atau teleskop).
- **Istikmal:** Menggenapkan umur bulan berjalan menjadi 30 hari karena hilal belum wujud atau kriteria visibilitas belum terpenuhi saat Maghrib hari ke-29.
- **Istiwa' A'zam (Rashdul Qibla):** Fenomena saat deklinasi matahari persis menyamai lintang Ka'bah, sehingga bayangan semua benda tegak lurus di bumi mengarah lurus ke Makkah.

J

- **Jarak Zenith (Zenith Distance):** Jarak sudut dari titik persis di atas kepala pengamat (zenith) ke benda langit. Sangat krusial dalam menentukan bayangan waktu Asar.
- **JPL (Jet Propulsion Laboratory):** Divisi laboratorium NASA yang menerbitkan standar file ephemeris presisi tinggi (Development Ephemeris / DE) yang diadopsi oleh ahli komputasi falak.
- **Julian Day (JD):** Skala waktu astronomis yang menghitung jumlah hari berlalu secara kontinu (termasuk desimal) sejak 1 Januari 4713 SM, untuk menghindari kerumitan perhitungan sistem bulan/tahun sipil.

K

- **Kalender Hijriah Global Tunggal (KHGT):** Konsep penanggalan universal berbasis prinsip satu hari satu tanggal untuk seluruh dunia (*ittihad al-matali'*), menggunakan parameter Istanbul 2016 (Tinggi 5°, Elongasi 8°).
- **Kulminasi Atas (Solar Transit):** Momen di mana matahari melintasi meridian pengamat (titik tertingginya pada hari tersebut), menandai sesaat sebelum masuknya waktu Zuhur (Zawal).

L

- **Lintang (Latitude):** Garis imajiner horizontal yang mengukur jarak sudut ke arah Utara (+) atau Selatan (-) dari Ekuator bumi.
- **Local Mean Time (LMT):** Waktu menengah/rata-rata yang dikalkulasi secara spesifik berdasarkan bujur lokal pengamat (titik koordinat persis), bukan dipukul rata berdasarkan zona waktu sipil wilayah.

M

- **MABIMS, Neo:** Kriteria visibilitas hilal yang disepakati oleh Menteri Agama Brunei, Indonesia, Malaysia, dan Singapura, menetapkan syarat tinggi hilal minimal 3° toposentrik dan elongasi 6.4° geosentris.
- **Matplotlib:** Pustaka Python yang sangat kuat untuk memvisualisasikan data komputasi menjadi bentuk grafik dua dimensi (seperti kurva *Equation of Time* atau Kompas Kiblat).
- **Matla':** Titik terbitnya bulan; dalam konteks KHGT, diistilahkan sebagai *Matla' Global* (menganggap seluruh bumi sebagai satu kesatuan tempat terbit).

N

- **New Moon:** Fase bulan mati yang secara fisis identik dengan terjadinya Konjungsi/Ijtimak.
- **Nutasi:** Goyangan kecil dan periodik pada sumbu rotasi bumi yang disebabkan oleh tarikan gravitasi kompleks bulan dan matahari, yang secara otomatis dihitung oleh pustaka *Skyfield*.

O

- **Obliquitas Ekliptika:** Sudut kemiringan sumbu rotasi bumi terhadap bidang orbitnya (ekliptika), besarnya sekitar $23,44$ derajat. Menjadi penyebab terjadinya pergantian musim dan variasi deklinasi matahari.
- **OOP (Object-Oriented Programming):** Paradigma pemrograman Python yang membungkus data dan fungsi (metode) ke dalam cetak biru yang disebut *Class*, untuk menciptakan mesin falak terpadu yang cepat dan efisien.

P

- **Paralaks Ekuatorial Horizontal:** Pergeseran posisi tampak bulan ke arah bawah (mendekati ufuk) akibat pengamatan dilakukan dari permukaan bumi (toposentris), bukan dari pusat bumi.
- **PyInstaller:** Alat utilitas Python untuk mengompilasi skrip `.py` menjadi file aplikasi biner mandiri (`.exe`) agar mudah didistribusikan kepada publik.

Q

- **Qibla (Kiblat):** Arah menuju Ka'bah di Makkah (Lintang $21^\circ 25' 21''$ LU, Bujur $39^\circ 49' 34''$ BT) sebagai orientasi spasial fisik peribadatan umat Islam yang dihitung menggunakan *Great Circle*.

R

- **Refraksi Atmosfer:** Pembelokan lintasan cahaya matahari atau bulan ketika menembus ketebalan atmosfer bumi, menyebabkan benda langit terlihat lebih tinggi dari posisi geometris aslinya di dekat ufuk.

- **Rukyatul Hilal:** Aktivitas observasi visual untuk melihat kemunculan hilal pada saat matahari terbenam di hari ke-29 bulan kamariah.

S

- **Skyfield:** Pustaka murni Python bertaraf observatorium internasional yang digunakan untuk komputasi posisi astrometri planet, matahari, satelit, dan bulan dengan akurasi *microsecond*.
- **Sunset (Terbenam Matahari):** Peristiwa hilangnya seluruh piringan atas matahari di bawah ufuk lokal. Merupakan batas waktu (*pivot*) pergantian hari dalam sistem kalender Hijriah (Maghrib).

T

- **Toposentrik:** Sistem referensi observasi yang ditarik secara spesifik dari titik kordinat pengamat di atas permukaan bumi (memerhitungkan lintang, bujur, dan elevasi).
- **Terrestrial Time (TT):** Skala waktu dinamis yang berdetak murni dan konstan secara matematis, tidak terpengaruh oleh perlambatan perputaran bumi.

U

- **Ufuk (Horizon):** Garis batas pandang imajiner yang memisahkan langit dan bumi. Ketinggian (Altitude) benda langit selalu diukur dari ufuk (0 derajat).
- **UTC (Universal Time Coordinated):** Standar waktu sipil global yang dijaga menggunakan jam atom dan dikalibrasi dengan putaran bumi melalui penambahan detik kabisat (*Leap Seconds*).

V

- **Vektor (Astrometri):** Satuan besaran yang memiliki nilai dan arah dalam ruang 3 dimensi. Dalam Python, vektor digunakan untuk memetakan jarak sinar dari pusat bumi menuju posisi matahari atau bulan.
- **Visibilitas Hilal:** Syarat-syarat parameter matematis-optik (seperti Tinggi, Elongasi, dan Umur Bulan) yang menentukan apakah hilal dapat dilihat secara fisis atau tidak.

W

- **WGS84 (World Geodetic System 1984):** Standar model bentuk bumi elipsoidal yang digunakan dalam sistem kartografi modern dan GPS, yang diimplementasikan oleh *Skyfield* untuk mendefinisikan posisi spasial pengamat.
- **Wujudul Hilal:** Kriteria hisab (tradisional Muhammadiyah pra-KHGT) yang menetapkan masuknya bulan baru asalkan bulan telah wujud (lebih besar dari 0 derajat) saat matahari terbenam pasca-Ijtimak, tanpa batas minimal ketinggian.

X

- **X, Sumbu (X-Axis):** Representasi garis horizontal dalam sistem koordinat Kartesian yang dikonversi Python dalam rendering UI maupun *Matplotlib* (misalnya untuk merepresentasikan parameter Tanggal/Waktu dalam grafik *Equation of Time*).

Y

- **Y, Sumbu (Y-Axis):** Representasi garis vertikal dalam sistem koordinat Kartesian grafik data (misalnya untuk merepresentasikan tinggi fluktuasi Deklinasi atau menit *Perata Waktu* dalam *Matplotlib*).

Z

- **Zawal:** Waktu sesaat setelah matahari melewati titik puncaknya (kulminasi atas) dan mulai condong bergerak/tergelincir ke arah ufuk barat, menandai masuknya waktu Zuhur.
- **Zenith:** Titik paling tinggi di langit yang tepat berada tegak lurus (90 derajat) di atas kepala pengamat secara lokal.
- **Zīj:** Tabel astronomi numerik yang disusun oleh para sarjana falak Islam klasik (seperti Al-Khawarizmi atau Ulugh Beg) untuk memprediksi posisi benda langit sebelum era integrasi numerik modern.

LAMPIRAN 1

DOKUMEN FORMULA ASTRONOMI PENTING

Dokumen ini berisi kumpulan formula dan algoritma fundamental yang digunakan dalam buku *Silabus Perkuliahan: Python untuk Astronomi Islam*. Formula disusun menggunakan representasi teks standar dan operator matematika umum yang kompatibel dengan penyunting teks (seperti MS Word) dan bahasa pemrograman Python.

1. KONTANTA DAN KONVERSI DASAR

1.1 Nilai Pi (π) Standar

Digunakan untuk presisi tinggi dalam fungsi trigonometri.

- $\text{Pi} = 3.141592653589793$

1.2 Konversi Derajat Desimal ke Radian

Python memproses fungsi `sin`, `cos`, `tan` dalam satuan Radian.

- $\text{Radian} = \text{Derajat_Desimal} * (\text{Pi} / 180)$

1.3 Konversi Radian ke Derajat Desimal

Untuk menampilkan hasil keluaran (output) agar mudah dipahami manusia.

- $\text{Derajat_Desimal} = \text{Radian} * (180 / \text{Pi})$

1.4 Konversi Jam Desimal ke Derajat (Waktu ke Spasial)

Bumi berotasi 15 derajat dalam 1 jam.

- $\text{Derajat} = \text{Jam_Desimal} * 15$

1.5 Konversi Derajat ke Jam Desimal (Spasial ke Waktu)

Digunakan untuk Sudut Jam (Hour Angle).

- $\text{Jam_Desimal} = \text{Derajat} / 15$

2. GEODESI SPASIAL DAN ARAH KIBLAT

2.1 LMT Offset (Koreksi Bujur ke Waktu)

Menghitung selisih waktu menengah lokal (LMT) terhadap UTC berdasarkan bujur (dalam detik).

- 1 Derajat Bujur = 4 Menit = 240 Detik.
- $\text{Offset_LMT_Detik} = \text{Bujur_Desimal} * 240$
- *Catatan: Gunakan Bujur Positif untuk Bujur Timur (BT), Negatif untuk Bujur Barat (BB).*

2.2 Koordinat Absolut Ka'bah (WGS84)

- $\text{Lintang_K} = 21.422487$ derajat (Utara)
- $\text{Bujur_K} = 39.826206$ derajat (Timur)

2.3 Algoritma Azimut Kiblat (Spherical Trigonometry)

Menghitung sudut dari Utara Sejati searah jarum jam (UTS).

- Input: Lintang_Lokal (ϕ), Bujur_Lokal (λ).
- Variabel Pembantu:
 - $A = \sin(\text{Bujur_K_Radian} - \text{Bujur_Lokal_Radian})$
 - $B = \cos(\text{Lintang_Lokal_Radian}) * \tan(\text{Lintang_K_Radian})$
 - $C = \sin(\text{Lintang_Lokal_Radian}) * \cos(\text{Bujur_K_Radian} - \text{Bujur_Lokal_Radian})$
- Rumus Inti (Python `atan2`):
 - $\text{Azimut_Radian} = \text{atan2}(A, B - C)$
- Normalisasi Kompas (Derajat):
 - JIKA $\text{Azimut_Derajat} < 0$ MAKA: $\text{Azimut_Derajat} = \text{Azimut_Derajat} + 360$

3. WAKTU SALAT SIANG (ZUHUR DAN ASAR)

3.1 Waktu Zuhur (Solar Transit/Zawal)

- $\text{Zuhur_UTC} = t_{\text{meridian_transit}}$ (Dari Ephemeris JPL)
- $\text{Zuhur_Lokal} = \text{Zuhur_UTC} + \text{Zona_Waktu_Lokal} + \text{Ikhtiyat_2_Menit}$

3.2 Jarak Zenith Matahari Siang Hari (Z_d)

Selisih sudut antara Zenith lokal dan posisi Matahari saat kulminasi.

- $Z_d_{\text{Derajat}} = \text{abs}(\text{Lintang_Lokal} - \text{Deklinasi_Matahari})$

3.3 Tinggi Matahari Awal Waktu Asar (h)

Menghitung sudut ketinggian matahari berdasarkan rasio panjang bayangan (Jumhur = 1, Hanafi = 2).

- $\text{Target_Tangen_h} = 1 / (\text{Rasio_Faktor} + \tan(\text{Zd_Radian}))$
- $\text{Tinggi_Asar_Radian} = \text{atan}(\text{Target_Tangen_h})$

4. WAKTU SALAT MALAM (MAGHRIB, ISYA', SUBUH)

4.1 Persamaan Sudut Jam Matahari (Hour Angle / H)

Digunakan untuk mencari rentang waktu dari kulminasi menuju ketinggian target.

- Input: Ketinggian_Matahari_Target (h), Lintang_Lokal (phi), Deklinasi_Matahari (delta).
- Rumus Inti (cos H):
 - $\cos_H = (\sin(h_Radian) - \sin(phi_Radian) * \sin(delta_Radian)) / (\cos(phi_Radian) * \cos(delta_Radian))$
- $H_Radian = \text{acos}(\cos_H)$

4.2 Ketinggian Matahari Target (h) untuk Salat Gelap

- **Maghrib (Piringan Atas di Ufuk tampak):**
 - $h_Maghrib = -0.8333$ derajat (Termasuk refraksi standar 34' dan semi-diameter 16').
- **Isya' (Akhir Syafaq Astronomis - Kriteria Tarjih):**
 - $h_Isya = -18.0$ derajat (Sudut depresi pusat matahari).
- **Subuh (Awal Fajar Shadiq - Kriteria Tarjih Munas ke-31):**
 - $h_Subuh = -18.0$ derajat (Sudut depresi pusat matahari).

4.3 Waktu Salat Malam (Kalkulasi)

- $\text{Sudut_Jam_Waktu_Jam_Desimal} = H_Derajat / 15$
- $\text{Waktu_Salat_Astronomis} = \text{Zuhur_Astronomis} +/- \text{Sudut_Jam_Waktu_Jam_Desimal}$
 - (Gunakan '+' untuk Maghrib/Isya, '-' untuk Subuh).
- $\text{Waktu_Salat_Lokal} = \text{Waktu_Salat_Astronomis} + \text{Ikhtiyat_2_Menit}$

5. BULAN KAMARIAH DAN VISIBILITAS HILAL

5.1 Definisi Ijtimak (Konjungsi Geosentris)

Momen absolut global.

- Syarat: $\text{Bujur_Ekliptika_Geosentris_Bulan} - \text{Bujur_Ekliptika_Geosentris_Matahari} = 0$

5.2 Umur Bulan (Age of Moon)

Dihitung saat Matahari terbenam (Sunset) di lokasi pengamat.

- $\text{Umur_Jam} = (\text{t_Sunset_JulianDay} - \text{t_Ijtimak_JulianDay}) * 24$

5.3 Parameter Visibilitas Hilal Toposentrik (Sari)

Diekstraksi tepat pada detik Matahari Terbenam lokal (`Apparent AltAz`).

- **Tinggi Hilal (h_B):** `alt_bulan.degrees` (Toposentrik).
- **Elongasi (Separasi Sferis Bulan-Matahari):**
 - Gunakan fungsi `separation_from` antara objek `apparent` Bulan dan Matahari (hasil dalam Derajat).

6. LOGIKA POHON KEPUTUSAN KRITERIA

6.1 Kriteria Neo MABIMS (AND Logic)

Menentukan apakah bulan baru sudah masuk.

- Syarat 1 (Tinggi): `Tinggi_Hilal_Toposentrik` ≥ 3.0 Derajat.
- Syarat 2 (Elongasi): `Elongasi_Geosentris` ≥ 6.4 Derajat.
- Logika: JIKA (Syarat 1 terpenuhi) DAN (Syarat 2 terpenuhi) MAKA Tanggal 1 Mulai Besok.

Berikut adalah perbaikan format penulisan kriteria KHGT agar terlihat lebih rapi, profesional, dan sesuai dengan standar dokumen resmi atau naskah referensi:

6.2 KRITERIA KHGT (KALENDER HIJRIAH GLOBAL TUNGGAL — RESOLUSI ISTANBUL)

Implementasi Kalender Hijriah Global Tunggal (KHGT) didasarkan pada prinsip satu hari satu tanggal untuk seluruh dunia. Berikut adalah penjelasan Parameter Kalender Global (PKG) yang digunakan dalam penentuan awal bulan Hijriah berdasarkan Resolusi Istanbul 2016:

1. Parameter Kalender Global (PKG) 1

Awal bulan Hijriah ditetapkan apabila di belahan bumi mana pun, sebelum pukul 00:00 UTC (*Universal Time Coordinated*), telah terpenuhi dua syarat astronomis sebagai berikut:

- **Ketinggian Bulan (Tinggi Hilal):** Minimal 5° di atas ufuk pada saat matahari terbenam.
- **Elongasi Bulan (Jarak Sudut Bulan-Matahari):** Minimal 8° .

Ketentuan: Apabila kedua syarat tersebut terpenuhi di lokasi mana pun di dunia sebelum batas waktu 00:00 UTC, maka keesokan harinya secara otomatis ditetapkan sebagai awal bulan baru untuk seluruh dunia tanpa terkecuali.

2. Parameter Kalender Global (PKG) 2

Apabila kriteria ketinggian minimal 5° dan elongasi minimal 8° baru terpenuhi setelah pukul 00:00 UTC, maka awal bulan Hijriah tetap dimulai pada keesokan harinya, dengan catatan harus memenuhi dua syarat tambahan:

- **Terjadinya Ijtimak (Konjungsi):** Fase bulan baru astronomis (ijtimak) wajib sudah terjadi sebelum waktu fajar menyingsing di Selandia Baru (*New Zealand*).
- **Lokasi Visibilitas:** Kriteria ketinggian minimal 5° dan elongasi minimal 8° tersebut harus terpenuhi di wilayah daratan Benua Amerika.

Kesimpulan

Kriteria KHGT ini dirancang untuk menciptakan satu acuan tunggal yang bersifat ilmiah sekaligus syar'i bagi umat Islam di seluruh dunia. Melalui standarisasi parameter 5-8 ini, kepastian penanggalan Hijriah dapat diprediksi dengan akurat dalam jangka waktu yang sangat panjang, sekaligus mengeliminasi perbedaan hari raya antarwilayah geografis.

LAMPIRAN 2

LINI MASA (TIMELINE) SEJARAH KALENDER MASEHI (JULIAN & GREGORIAN)

Pemahaman terhadap sejarah kalender Masehi (Sipil) sangat krusial dalam komputasi astronomi (Ilmu Falak). Dalam pemrograman Python menggunakan pustaka seperti `skyfield` atau `PyEphem`, mesin harus mengkonversi tanggal sipil yang dimasukkan pengguna menjadi hitungan waktu absolut (*Julian Day*). Ketidaktahuan akan sejarah transisi kalender dapat menyebabkan kesalahan fatal saat menghitung peristiwa astronomis historis (seperti gerhana di masa Nabi Muhammad ﷺ) maupun prediksi masa depan.

Berikut adalah lini masa kronologis evolusi Kalender Masehi dari era Romawi hingga era komputasi modern:

1. Era Romawi Kuno (Sebelum 45 SM)

- **Sistem Lunisolar Kalut:** Kalender Romawi awal berbasis pada siklus bulan dan musim pertanian, hanya terdiri dari 10 bulan (304 hari) dalam setahun, dimulai dari bulan Martius (Maret) dan berakhir pada December. Musim dingin sering kali tidak dimasukkan ke dalam kalender.
- **Kekacauan Musim:** Para pendeta Romawi (*Pontifices*) sering menyisipkan bulan kabisat (*Intercalaris*) secara sewenang-wenang untuk tujuan politik (memperpanjang masa jabatan pejabat tertentu), sehingga kalender sangat melenceng dari musim matahari (equinoks dan solstis).

2. Reformasi Julian (46 SM - 45 SM)

- **Keterlibatan Astronom Aleksandria:** Julius Caesar, dibantu oleh astronom Yunani dari Aleksandria bernama Sosigenes, mereformasi sistem penanggalan secara radikal. Ia meninggalkan kalender lunar dan beralih sepenuhnya ke kalender matahari (Solar).
- **Ketetapan Matematis:** Satu tahun ditetapkan memiliki panjang rata-rata **365,25 hari**.
- **Aturan Kabisat:** Untuk mengakomodasi kelebihan 0,25 hari, setiap 4 tahun sekali ditambahkan 1 hari ekstra pada bulan Februari (Tahun Kabisat berumur 366 hari).
- **Tahun Kekacauan (Year of Confusion):** Tahun 46 SM diperpanjang menjadi 445 hari untuk menyelaraskan kembali kalender dengan musim. Mulai 1 Januari 45 SM, Kalender Julian resmi berlaku.

3. Konsili Nicea (325 M)

- **Penetapan Equinoks:** Gereja menetapkan bahwa peristiwa *Vernal Equinox* (Titik Musim Semi) jatuh pada tanggal **21 Maret**. Hal ini sangat penting untuk penentuan hari raya Paskah. Tanggal ini kemudian menjadi patokan pergeseran astronomis di abad-abad berikutnya.

4. Pengenalan Era Anno Domini / Masehi (525 M)

- **Dionysius Exiguus:** Seorang biarawan bernama Dionysius Exiguus menciptakan sistem perhitungan tahun *Anno Domini* (AD) atau Tahun Tuhan kita (di Indonesia dikenal sebagai Masehi / M). Ia menghitung mundur untuk menentukan tahun kelahiran Yesus (Nabi Isa) dan menetapkannya sebagai Tahun 1.
- **Tidak Ada Tahun Nol:** Dalam sistem ini, tahun 1 SM (Sebelum Masehi / BC) langsung dilanjutkan oleh tahun 1 Masehi (AD). Tidak ada Tahun 0. (*Catatan Falak: Dalam astronomi komputasi seperti Python, tahun 1 SM ditulis sebagai Tahun 0, dan tahun 2 SM ditulis sebagai Tahun -1 untuk menjaga kontinuitas kalkulus matematika*).

5. Reformasi Gregorian (1582 M) - TITIK KRUSIAL KOMPUTASI

- **Masalah Kalender Julian:** Panjang tahun tropis sebenarnya adalah 365,24219 hari, sedangkan Julian menetapkan 365,25 hari. Selisih 11 menit 14 detik ini terakumulasi menjadi kesalahan 1 hari setiap 128 tahun. Pada tahun 1582, *Vernal Equinox* telah bergeser dari 21 Maret menjadi 11 Maret.
- **Paus Gregorius XIII & Christopher Clavius:** Memperkenalkan Kalender Gregorian untuk mengoreksi kesalahan tersebut.
- **Lompatan Waktu (Potong 10 Hari):** Tanggal **4 Oktober 1582 (Kamis)** langsung diikuti oleh tanggal **15 Oktober 1582 (Jumat)**. Tanggal 5 hingga 14 Oktober 1582 tidak pernah ada dalam sejarah.
- **Aturan Kabisat Baru:** Tahun habis dibagi 4 adalah kabisat. NAMUN, tahun abad (berakhiran 00) *bukan* kabisat, KECUALI habis dibagi 400. (Contoh: Tahun 1700, 1800, 1900 bukan kabisat. Tahun 1600 dan 2000 adalah kabisat).

6. Adopsi Bertahap Kalender Gregorian (1582 - Abad ke-20)

- Transisi tidak terjadi serentak di seluruh dunia, yang menyebabkan kebingungan pencatatan sejarah:
 - **1582:** Italia, Spanyol, Portugal, Polandia, Prancis.
 - **1752:** Kekaisaran Inggris dan koloni Amerikanya (Memotong 11 hari: 2 September 1752 langsung diikuti 14 September 1752).
 - **1873:** Jepang mengadopsi Gregorian.
 - **1918:** Rusia (Soviet) beralih ke Gregorian setelah Revolusi Oktober (yang ironisnya terjadi pada bulan November di Kalender Gregorian).
 - **1923:** Yunani dan Turki menjadi negara terakhir yang mengadopsi kalender sipil ini.

7. Era Komputasi Astronomi & Skala Waktu Dinamis (Abad ke-20 - Sekarang)

- **Standar ISO 8601:** Sistem komputer menetapkan format YYYY-MM-DD untuk komunikasi data global yang bebas dari kebingungan regional.
- **Proliferasi Gregorian Proleptik:** Dalam beberapa aplikasi perangkat lunak (termasuk Python standar `datetime`), aturan Gregorian (kabisat baru) ditarik mundur hingga ke masa sebelum 1582 untuk mempermudah kalkulasi. Ini disebut *Proleptic Gregorian Calendar*.

- **Julian Day (JD) sebagai Solusi Kosmik:** Untuk menghindari kekacauan tanggal akibat hilangnya 10 hingga 13 hari dalam sejarah sipil, para astronom Islam maupun Barat modern mengonversi semua input penanggalan manusia ke dalam *Julian Day*, yakni penghitungan linear hari yang telah berlalu sejak 1 Januari 4713 SM tanpa terputus. Mesin falak (seperti Pustaka `Skyfield` dalam Python) menggunakan JD ini sebagai dasar pergerakan ephemeris JPL NASA.

LAMPIRAN 3

LINI MASA (TIMELINE) SEJARAH KALENDER HIJRIAH (ISLAM)

Memahami sejarah dan evolusi Kalender Hijriah sama pentingnya dengan memahami algoritma pergerakan bulan itu sendiri. Dalam pemrograman Ilmu Falak (seperti pembuatan fungsi Bidi-Converter Masehi-Hijriah), seorang pemrogram harus menyadari bahwa sistem penanggalan Islam telah melewati fase penyempurnaan dari observasi murni, formulasi tabel klasik (*Zij*), hingga menjadi arsitektur komputasi global di era modern.

Berikut adalah lini masa sejarah dan evolusi Kalender Hijriah:

1. Era Pra-Islam (Sebelum 622 M): Sistem Lunisolar dan Tradisi Nasi'

- **Lunisolar yang Kalut:** Masyarakat Arab pra-Islam menggunakan kalender berbasis fase bulan (lunar), namun mereka berusaha menyejarkannya dengan musim matahari (solar) agar waktu pelaksanaan ibadah haji dan pameran dagang (seperti pasar Ukaz) selalu jatuh pada musim panen yang sejuk.
- **Praktik Nasi' (Bulan Sisipan):** Untuk menyelaraskan bulan dan musim, para elite Arab melakukan *Nasi'*, yaitu menyisipkan bulan ke-13 secara sewenang-wenang. Praktik ini sering dimanipulasi untuk kepentingan politik dan ekonomi, membuat urutan bulan-bulan suci menjadi kacau balau.

2. Tahun 1 H (622 M): Peristiwa Hijrah

- **Tonggak Peradaban:** Nabi Muhammad ﷺ beserta para sahabat melakukan migrasi (hijrah) dari Makkah ke Yatsrib (Madinah). Pada saat peristiwa ini terjadi, tahun tersebut belum disebut sebagai "Tahun 1". Orang Arab saat itu menamai tahun berdasarkan peristiwa besar yang terjadi pada tahun tersebut (misalnya, Tahun Gajah).

3. Tahun 10 H (632 M): Penghapusan Praktik Nasi' (Haji Wada')

- **Kembali ke Lunar Murni:** Saat Khutbah Perpisahan (Haji Wada'), turun wahyu Al-Qur'an (Surat At-Taubah ayat 36-37) yang secara tegas melarang dan mengharamkan praktik *Nasi'*.
- **Ketetapan 12 Bulan:** Kalender Islam secara resmi dideklarasikan sebagai kalender lunar murni (*Pure Lunar Calendar*) yang berotasi bebas dari musim matahari. 1 tahun Hijriah ditetapkan berjumlah 12 bulan (sekitar 354 atau 355 hari), lebih pendek sekitar 11 hari dari kalender Masehi.

4. Tahun 17 H (638 M): Penetapan Resmi Kalender Hijriah

- **Krisis Administrasi:** Kekhalifahan Islam di bawah Umar bin Khattab meluas pesat. Terdapat kebingungan dalam administrasi surat-menyurat negara (contoh: surat tagihan utang tertulis "bulan Syaban", namun tidak jelas Syaban tahun ini atau tahun lalu).

- **Sidang Sahabat:** Khalifah Umar mengumpulkan para sahabat untuk menetapkan kalender resmi negara. Terdapat usulan untuk memulai kalender dari tahun kelahiran Nabi, tahun penobatan kenabian, atau tahun wafatnya.
- **Usulan Ali bin Abi Thalib:** Atas usulan Ali bin Abi Thalib, tahun terjadinya peristiwa Hijrah disepakati sebagai Tahun 1. Bulan Muharram dipertahankan sebagai bulan pertama dalam susunan kalender.
- **Titik Nol Astronomis:** Secara hisab astronomi mundur (ekstrapolasi), tanggal 1 Muharram Tahun 1 Hijriah bertepatan dengan tanggal **15 atau 16 Juli 622 M** (dalam Kalender Julian).

5. Abad ke-8 hingga ke-15 M: Era Keemasan Falak dan Hisab Klasik

- **Tabel Zij:** Astronom muslim seperti Al-Khawarizmi, Al-Battani, Nasir al-Din al-Tusi, hingga Ulugh Beg menyusun *Zij* (tabel astronomi matematis). Mereka menerjemahkan pergerakan bulan yang kompleks ke dalam algoritma trigonometri.
- **Peralihan Epistemologi:** Hisab (perhitungan) mulai digunakan secara luas untuk mendampingi rukyat (observasi visual), meletakkan fondasi bagi astronomi presisi tinggi di masa depan.

6. Abad ke-20 M: Era Komputasi Digital dan Kriteria Regional

- **Transisi Teknologi:** Perhitungan manual beralih menggunakan kalkulator dan komputer. Hisab berkembang dari *Hisab Taqribi* (pendekatan/rata-rata) menjadi *Hisab Hakiki* (posisi aktual dengan mempertimbangkan anomali orbit).
- **Kriteria Regional:** Muncul berbagai kriteria lokal untuk menetapkan awal bulan. Di Indonesia, Muhammadiyah memelopori penggunaan *Wujudul Hilal* (pokoknya bulan di atas ufuk saat Maghrib). Sementara itu, pemerintah merumuskan kriteria *Imkanur Rukyat* (kemungkinan bulan bisa dilihat) yang kemudian disepakati bersama negara serumpun melalui forum MABIMS (Menteri Agama Brunei, Indonesia, Malaysia, dan Singapura).

7. Mei 2016 M: Kongres Internasional Istanbul Turki

- **Konsensus Global:** Para ahli astronomi dan ulama dari seluruh dunia berkumpul di Istanbul untuk merumuskan satu sistem penanggalan yang menyatukan umat Islam secara global.
- **Lahirnya Parameter KHGT:** Kongres ini melahirkan Resolusi Istanbul yang menyepakati "Kalender Hijriah Global Tunggal (KHGT)". Resolusi ini menggunakan prinsip *Ittihad al-Matali'* (satu hari satu tanggal untuk seluruh dunia) dengan parameter fisis yang ketat (Tinggi 5 derajat dan Elongasi 8 derajat di mana saja di belahan bumi pertama kali terbenam matahari).

8. Tahun 1447 H (2025 M): Penerapan Penuh KHGT (Muhammadiyah)

- **Implementasi Nyata:** Sebagai tindak lanjut dari Resolusi Istanbul dan keputusan Mukhtar, Majelis Tarjih dan Tajdid PP Muhammadiyah secara resmi meninggalkan

kriteria Wujudul Hilal dan beralih sepenuhnya menerapkan Kalender Hijriah Global Tunggal (KHGT) mulai 1 Muharram 1447 H.

- **Era Pemrograman Astronomi Global:** Peralihan ini menuntut pengembangan perangkat lunak (seperti yang diajarkan menggunakan Python dalam buku ini) yang mampu menyapu koordinat seluruh dunia secara bersamaan untuk menemukan titik visibilitas hilal pertama, menghapus sekat-sekat batas negara dalam kalender Islam.

LAMPIRAN 4

ASAL-USUL DAN ETIMOLOGI NAMA BULAN HIJRIAH

Dalam studi Astronomi Islam (Ilmu Falak), pemahaman mengenai penamaan bulan bukan sekadar hafalan urutan waktu, melainkan pintu gerbang untuk memahami bagaimana masyarakat Arab kuno melakukan sinkronisasi antara aktivitas sosial-ekonomi dengan fenomena alam (musim). Meskipun saat ini kalender Hijriah bersifat lunar murni (lepas dari musim), nama-namanya tetap membawa jejak sejarah iklim dan sosiologis masa lalu.

Berikut adalah penjelasan etimologis dan asal-usul 12 nama bulan Hijriah:

Tabel Ringkasan Makna Bulan

No	Nama Bulan	Makna Dasar (Etimologi)	Konteks Sejarah / Musim
1	Muharram	Diharamkan / Suci	Larangan berperang demi keamanan haji.
2	Safar	Kosong / Kuning	Rumah yang kosong ditinggal penghuninya.
3	Rabi'ul Awal	Musim Semi Pertama	Awal musim mekarnya bunga dan rumput.
4	Rabi'ul Akhir	Musim Semi Terakhir	Berakhirnya masa pertumbuhan tanaman.
5	Jumadil Awal	Beku / Keras (I)	Air yang mulai membeku atau tanah yang mengeras.
6	Jumadil Akhir	Beku / Keras (II)	Puncak musim dingin atau kekeringan ekstrem.
7	Rajab	Mulia / Hormat	Bulan gencatan senjata yang sangat dihormati.
8	Sya'ban	Mempencar / Cabang	Orang-orang berpencar mencari sumber air.
9	Ramadan	Panas Membakar	Teriknya matahari yang membakar permukaan tanah.
10	Syawal	Mengangkat / Membawa	Unta betina yang mengangkat ekornya.
11	Zulkaidah	Duduk / Istirahat	Masa berhenti berperang dan menetap di rumah.
12	Zulhijah	Pemilik Haji	Waktu pelaksanaan ibadah haji ke Ka'bah.

Penjelasan Mendalam Per Bulan

1. Muharram (مُحَرَّم)

Berasal dari kata *haram* yang berarti suci atau terlarang. Dinamakan demikian karena pada bulan ini masyarakat Arab mengharamkan peperangan. Hal ini dilakukan untuk menjamin keamanan para peziarah yang kembali ke rumah masing-masing setelah menunaikan ibadah haji di bulan Zulhijah.

2. Safar (صَفَر)

Secara harfiah berarti "kosong" atau "nol". Ada dua teori populer: pertama, karena rumah-rumah di Makkah menjadi kosong ditinggal penghuninya yang pergi berperang atau berdagang. Kedua, berasal dari kata *shufrah* (kuning), merujuk pada tanaman yang menguning karena musim gugur.

3 & 4. Rabi'ul Awal & Akhir (رَبِيعُ الْأَوَّلِ / الْأَخِر)

Kata *Rabi'* berarti musim semi atau masa pertumbuhan. Penamaan ini muncul ketika bulan tersebut bertepatan dengan musim di mana bumi menghihijau, bunga bermekaran, dan ternak mendapatkan pakan melimpah.

5 & 6. Jumadil Awal & Akhir (جُمَادَى الْأُولَى / الْآخِرَة)

Berasal dari kata *Jamad* yang berarti membeku atau menjadi keras. Merujuk pada periode di mana air membeku karena puncak musim dingin (di beberapa wilayah gurun yang dingin), atau secara metaforis tanah yang mengeras karena kekeringan panjang di wilayah lainnya.

7. Rajab (رَجَب)

Secara etimologi berarti "mengagungkan". Rajab adalah bulan *Haram* yang berdiri sendiri (tidak berurutan dengan 3 bulan haram lainnya). Masyarakat Arab melepas mata tombak dan busur panah mereka pada bulan ini sebagai bentuk penghormatan.

8. Sya'ban (شَعْبَان)

Berasal dari kata *tasy'aba* yang berarti berpencar. Setelah masa istirahat di bulan Rajab, orang-orang mulai berpencar untuk mencari sumber air (*al-miyah*) atau berpencar untuk melakukan ekspedisi dagang dan perang.

9. Ramadan (رَمَضَانَ)

Berasal dari akar kata *ar-ramad* yang berarti panas yang menyengat atau membakar. Nama ini diberikan karena pada masa pembentukannya, bulan ini jatuh pada puncak musim panas di mana tanah menjadi sangat panas hingga membakar kaki unta. Secara maknawi, Ramadan juga diartikan sebagai "pembakar dosa".

10. Syawal (شَوَّال)

Berasal dari kata *syala* yang berarti mengangkat atau membawa. Hal ini merujuk pada unta betina yang mengangkat ekornya (*tasyawwala*) karena susunya yang mengering atau tanda siap kawin. Musim ini sering kali menjadi masa yang menantang bagi peternak.

11. Zulkaidah (ذُو الْقَعْدَة)

Berasal dari kata *qa'ada* yang berarti duduk. Dinamakan demikian karena orang-orang "duduk" (berdiam diri) di rumah dan berhenti melakukan perjalanan jauh atau peperangan guna menyambut musim haji yang akan datang.

12. Zulhijah (ذُو الْحِجَّة)

Secara harfiah berarti "Pemilik Haji" atau "Waktu Haji". Sejak zaman pra-Islam hingga syariat Islam ditegakkan, bulan ini merupakan waktu utama bagi masyarakat Arab dan umat Islam di seluruh dunia untuk berkumpul di Makkah menjalankan ritual ibadah haji.

Catatan Teknis untuk Pemrogram Falak: > Karena Kalender Hijriah saat ini adalah *Lunar Murni* (Tanpa sistem Nasi'), posisi bulan-bulan ini akan bergeser sekitar 11 hari setiap tahunnya terhadap kalender Masehi. Oleh karena itu, Anda mungkin akan menemui bulan Ramadan di musim dingin atau bulan Jumada di musim panas. Hal ini merupakan bukti fisis dari gerak revolusi bulan yang lebih cepat daripada matahari dalam siklus satu tahun sipil.

LAMPIRAN 5

WAKTU SHALAT MENURUT HPT 3

BAB III

WAKTU-WAKTU SALAT

A. Pendahuluan

Salat merupakan rukun Islam kedua, karena itu merupakan salah satu bentuk ibadah yang amat penting. As-Sarakhsi (w. 490/1096) menegaskan bahwa salat merupakan unsur agama terkuat sesudah iman kepada Allah swt.¹ Para ulama mengatakan bahwa salat merupakan tiang agama. Barang siapa menegakkannya berarti menegakkan agama dan barang siapa meruntuhkannya berarti meruntuhkan agama.² Mengingat pentingnya salat dalam agama Islam, maka ibadah ini tentu harus menjadi perhatian sungguh-sungguh umat Islam, termasuk memperhatikan waktu-waktu pelaksanaannya. Menyangkut waktu pelaksanaan salat, di dalam al-Quran ditegaskan,

إِنَّ الصَّلَاةَ كَانَتْ عَلَى الْمُؤْمِنِينَ كِتَابًا مَّوْقُوتًا [النساء (٤): ١٠٣].

Sesungguhnya salat itu adalah kewajiban yang ditentukan waktunya atas orang-orang yang beriman [QS an-Nisā' (4): 103].

Salat dalam Islam dibedakan menjadi dua kelompok: (1) salat-salat fardu (*aṣ-ṣalawāt al-makṭūbah*) dan salat-salat sunat (*ṣalawāt at-taṭawwu'*). Salat fardu meliputi Magrib, Isya, Subuh, Zuhur dan Asar. Waktu-waktu mengerjakannya telah ditentukan sebagaimana akan dikemukakan kemudian. Salat-salat sunat dari segi waktu mengerjakannya dibedakan menjadi dua: *pertama*, salat sunat rawatib, yaitu salat-salat sunat yang menyertai salat fardu lima waktu, dan waktunya mengikuti waktu-waktu salat fardu tersebut. *Kedua*, salat-salat sunat selain salat rawatib yang waktu-waktunya telah ditentukan pula.

B. Waktu-waktu Salat Fardu

Waktu-waktu salat fardu telah ditentukan dalam sumber-sumber syariah, yaitu al-Quran dan Sunnah Nabi saw. Hanya saja di dalam al-Quran tidak disebutkan secara rinci waktu mengerjakan masing-masing salat wajib, melainkan hanya terdapat isyarat-isyarat umum. Perincian waktu-waktu salat secara tegas disebutkan dalam sejumlah hadis Nabi saw.

¹ As-Sarakhsī, *al-Mabsūṭ* (Beirut: Dār al-Ma'rifah, t.t.), I: 4.

² Mengenai salat sebagai tiang agama ini terdapat beberapa riwayat hadis, namun hadis-hadis itu daif dan sebagian mursal meskipun rijalnya reliabel (terpercaya). Lihat as-Sakhāwī, *al-Maqāṣid al-Ḥasanah fī Bayāni Kaṣīrin min al-Aḥādīs al-Musytahirah 'alā al-Aṣṣināh* (Beirut: Dār al-Kitāb al-'Arabī, 1405/1985), I: 427.

1. Isyarat Waktu Salat dalam al-Quran

Di dalam al-Quran terdapat beberapa ayat yang menyebutkan waktu salat secara umum, yaitu beberapa ayat berikut:

- 1) Firman Allah dalam surat an-Nisā' ayat 103,

إِنَّ الصَّلَاةَ كَانَتْ عَلَى الْمُؤْمِنِينَ كِتَابًا مَوْقُوتًا [النساء (٤): ١٠٣].

Artinya: *Sesungguhnya salat itu adalah kewajiban yang ditentukan waktunya atas orang-orang yang beriman [QS an-Nisā' (4): 103].*

- 2) Firman Allah dalam surat Hūd ayat 114,

وَأَقِمِ الصَّلَاةَ طَرَفِي النَّهَارِ وَرُفُلًا مِنَ اللَّيْلِ إِنَّ الْحَسَنَاتِ يُذْهِبْنَ السَّيِّئَاتِ ذَلِكَ ذِكْرَى لِلذَّاكِرِينَ [هود (١١): ١١٤].

Dan dirikanlah salat itu pada kedua tepi siang (pagi dan petang) dan pada bahagian permulaan daripada malam. Sesungguhnya perbuatan-perbuatan yang baik itu menghapuskan (dosa) perbuatan-perbuatan yang buruk. Itulah peringatan bagi orang-orang yang ingat [QS Hūd (11): 114].

- 3) Firman Allah dalam surat al-Isrā' ayat 78,

أَقِمِ الصَّلَاةَ لِدُلُوكِ الشَّمْسِ إِلَى غَسَقِ اللَّيْلِ وَقُرْءَانَ الْفَجْرِ إِنَّ قُرْءَانَ الْفَجْرِ كَانَ مَشْهُودًا [الإسراء (١٧): ٧٨].

Dirikanlah salat dari sesudah matahari tergelincir sampai gelap malam dan (dirikanlah pula salat) subuh. Sesungguhnya salat subuh itu disaksikan (oleh malaikat) [QS al-Isrā' (17): 78].

- 4) Firman Allah dalam surat Ṭāhā ayat 130,

فَاصْبِرْ عَلَىٰ مَا يَقُولُونَ وَسَبِّحْ بِحَمْدِ رَبِّكَ قَبْلَ طُلُوعِ الشَّمْسِ وَقَبْلَ غُرُوبِهَا وَمِنْ آنَاءِ اللَّيْلِ فَسَبِّحْ وَأَطْرَافَ النَّهَارِ لَعَلَّكَ تَرْضَىٰ [طه (٢٠): ١٣٠].

Maka sabarlah kamu atas apa yang mereka katakan, dan bertasbihlah dengan memuji Tuhanmu sebelum terbit matahari dan sebelum terbenamnya dan bertasbih pulalah pada waktu-waktu di malam hari dan pada waktu-waktu di siang hari, supaya kamu merasa senang [QS Ṭāhā (20): 130].

2. Waktu Salat Fardu dalam Hadis

Terdapat beberapa hadis Nabi saw yang menyebutkan rincian waktu-waktu salat fardu.

- 1) Hadis 'Abdullah Ibn 'Amr,

عَنْ عَبْدِ اللَّهِ بْنِ عَمْرٍو أَنَّ رَسُولَ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ قَالَ وَقْتُ الظُّهْرِ إِذَا زَالَتِ الشَّمْسُ وَكَانَ ظِلُّ الرَّجُلِ كَطَوِيلِهِ مَا لَمْ يَحْضُرِ العَصْرُ وَوَقْتُ العَصْرِ مَا لَمْ تَصْفَرَ الشَّمْسُ وَوَقْتُ صَلَاةِ المَغْرِبِ مَا لَمْ يَغِبِ الشَّفَقُ وَوَقْتُ صَلَاةِ العِشَاءِ إِلَى نِصْفِ اللَّيْلِ الأَوْسَطِ وَوَقْتُ صَلَاةِ الصُّبْحِ مِنْ طُلُوعِ الفَجْرِ مَا لَمْ تَطْلُعِ الشَّمْسُ ... [رواه مسلم].

Dari 'Abdullah Ibn 'Amr (diriwayatkan) bahwa Rasulullah saw bersabda: Waktu zuhur adalah ketika matahari tergelincir (dan berlangsung hingga) bayangan orang sama dengan badannya selama belum masuk waktu asar. Waktu asar adalah berlangsung sampai matahari belum menguning. Waktu salat magrib berlangsung sampai hilangnya syafak. Waktu salat Isya berlangsung hingga pertengahan malam. Dan waktu salat subuh adalah dari terbit fajar sampai sebelum matahari terbit ... [HR Muslim].

2) Hadis Jabir,

عَنْ جَابِرٍ قَالَ جَاءَ جِبْرِيلُ إِلَى النَّبِيِّ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ حِينَ زَالَتِ الشَّمْسُ فَقَالَ قُمْ يَا مُحَمَّدُ فَصَلِّ الظُّهْرَ فَقَامَ فَصَلَّى الظُّهْرَ ثُمَّ جَاءَهُ حِينَ كَانَ ظِلُّ كُلِّ شَيْءٍ مِثْلَهُ فَقَالَ قُمْ فَصَلِّ العَصْرَ فَقَامَ فَصَلَّى العَصْرَ ثُمَّ جَاءَهُ حِينَ غَابَتِ الشَّمْسُ فَقَالَ قُمْ فَصَلِّ المَغْرِبَ فَقَامَ فَصَلَّى المَغْرِبَ ثُمَّ مَكَثَ حَتَّى ذَهَبَ الشَّفَقُ فَجَاءَهُ فَقَالَ قُمْ فَصَلِّ العِشَاءَ فَقَامَ فَصَلَّاهَا ثُمَّ جَاءَهُ حِينَ سَطَعَ الفَجْرُ بِالصُّبْحِ فَقَالَ قُمْ يَا مُحَمَّدُ فَصَلِّ فَصَلَّى الصُّبْحَ وَجَاءَهُ مِنَ العَدِ حِينَ صَارَ ظِلُّ كُلِّ شَيْءٍ مِثْلَهُ فَقَالَ قُمْ فَصَلِّ الظُّهْرَ فَقَامَ فَصَلَّى الظُّهْرَ ثُمَّ جَاءَهُ حِينَ كَانَ ظِلُّ كُلِّ شَيْءٍ مِثْلِيهِ فَقَالَ قُمْ فَصَلِّ العَصْرَ فَقَامَ فَصَلَّى العَصْرَ ثُمَّ جَاءَهُ حِينَ غَابَتِ الشَّمْسُ وَقَتًا وَاحِدًا لَمْ يَزَلْ عَنْهُ فَقَالَ قُمْ فَصَلِّ المَغْرِبَ فَقَامَ فَصَلَّى المَغْرِبَ ثُمَّ جَاءَهُ العِشَاءُ حِينَ ذَهَبَ ثُلُثُ اللَّيْلِ فَقَالَ قُمْ فَصَلِّ العِشَاءَ فَقَامَ فَصَلَّى العِشَاءَ ثُمَّ جَاءَهُ الصُّبْحُ حِينَ أُسْفَرَ جِدًّا فَقَالَ قُمْ فَصَلِّ الصُّبْحَ فَقَامَ فَصَلَّى الصُّبْحَ فَقَالَ مَا بَيْنَ هَذَيْنِ وَقْتُ كُلُّهُ [رواه ابن حبان في صحيحه].

Dari Jabir [diriwayakan bahwa] ia berkata: Jibril pernah datang kepada Nabi saw ketika matahari tergelincir dan berkata: Wahai Muhammad, berdirilah dan kerjakan salat zuhur! Maka Nabi saw berdiri dan mengerjakan salat zuhur. Kemudian Jibril datang lagi kepada Nabi saw ketika bayangan benda sama panjangnya dengan bendanya dan berkata: Berdirilah dan kerjakan salat asar. Maka Nabi saw berdiri dan mengerjakan salat Asar. Kemudian Jibril datang lagi kepada Nabi saw ketika matahari terbenam dan berkata: Berdirilah dan kerjakan salat Magrib. Maka Nabi saw berdiri dan mengerjakan salat Magrib. Kemudian Jibril tinggal di situ sampai hilangnya syafak, kemudian datang kepada Nabi saw dan berkata: Berdirilah dan kerjakan salat Isya. Maka Nabi saw berdiri dan mengerjakan salat Isya. Kemudian Jibril datang lagi kepada Nabi saw ketika fajar menyingsing memulai subuh dan berkata: Berdirilah wahai Muhammad dan kerjakan salat Subuh. Maka Nabi saw berdiri dan mengerjakan salat Subuh. Pada keesokan hari Jibril datang lagi kepada Nabi saw ketika bayangan benda sama panjang dengan bendanya dan berkata: Berdirilah dan kerjakan salat Zuhur. Maka Nabi saw berdiri dan mengerjakan salat Zuhur. Kemudian Jibril datang lagi kepada Nabi saw ketika bayangan benda dua kali panjang bendanya dan berkata: Berdirilah dan kerjakan salat Asar. Kemudian Jibril datang lagi kepada Nabi saw ketika matahari terbenam dan beliau terus di situ dan tidak beranjak, kemudian berkata: Berdirilah dan kerjakan salat Magrib. Maka Nabi saw berdiri dan mengerjakan salat magrib. Kemudian Jibril datang lagi kepada Nabi saw ketika Isya saat sepertiga malam telah berlalu dan berkata: Berdirilah dan kerjakan salat Isya. Maka Nabi saw berdiri dan mengerjakan salat Isya. Kemudian Jibril datang lagi kepada Nabi saw pada waktu subuh ketika subuh itu sudah sangat terang dan berkata: Berdirilah dan kerjakan salat Subuh. Maka Nabi saw berdiri dan mengerjakan salat Subuh. Kemudian ia berkata: Waktu di antara kedua waktu itu seluruhnya adalah waktu salat [HR Ibn Hibbān dalam *Ṣaḥīḥ*-nya. Hadis senada juga diriwayakan oleh an-Nasā'ī dan beberapa ahli hadis lainnya].

3) Hadis Ibn 'Abbās,

عَنِ ابْنِ عَبَّاسٍ قَالَ قَالَ رَسُولُ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ أَمِنِي جِبْرِيلُ عَلَيْهِ السَّلَامُ عِنْدَ الْبَيْتِ مَرَّتَيْنِ فَصَلَّى فِي الظُّهْرِ حِينَ زَالَتْ الشَّمْسُ وَكَانَتْ قَدَرُ النَّيْرَاكِ وَصَلَّى فِي الْعَصْرِ حِينَ كَانَ ظِلُّهُ مِثْلَهُ وَصَلَّى فِي يَغْنِي الْمَغْرِبِ حِينَ أَفْطَرَ الصَّائِمُ وَصَلَّى فِي الْعِشَاءِ حِينَ غَابَ الشَّفَقُ وَصَلَّى فِي الْفَجْرِ حِينَ حَرَّمَ الطَّعَامَ وَالشَّرَابَ عَلَى الصَّائِمِ فَلَمَّا كَانَ الْغَدُ صَلَّى فِي الظُّهْرِ حِينَ كَانَ ظِلُّهُ مِثْلَهُ وَصَلَّى فِي الْعَصْرِ حِينَ كَانَ ظِلُّهُ مِثْلِيهِ وَصَلَّى فِي الْمَغْرِبِ حِينَ أَفْطَرَ الصَّائِمُ وَصَلَّى فِي الْعِشَاءِ إِلَى ثُلُثِ اللَّيْلِ وَصَلَّى فِي الْفَجْرِ فَأَسْفَرَ ثُمَّ التَّفَّتْ إِلَى فَقَالَ يَا مُحَمَّدُ هَذَا وَقْتُ لَأَنْبِيَاءٍ مِنْ قَبْلِكَ وَالْوَقْتُ مَا بَيْنَ هَذَيْنِ الْوَقْتَيْنِ [رواه أبو داود واللفظ له والترمذی وابن ماجه].

Dari Ibn 'Abbās [diriwayatkan bahwa] ia berkata: Rasulullah saw telah bersabda: Jibril a.s. pernah mengimami saya untuk salat di Baitullah dua kali. Ia salat Zuhur mengimami saya ketika matahari tergelincir dan membentuk bayang-bayang sepanjang tali sepatu, dan salat Asar mengimami saya pada saat bayang-bayang sama panjangnya dengan bendanya. (Selanjutnya) ia salat mengimami saya – maksudnya salat Magrib– ketika orang berpuasa berbuka. Ia salat Isya mengimami saya ketika syafak menghilang. Ia salat Fajar (Subuh) mengimami saya ketika makanan dan minuman tidak lagi boleh disantap oleh orang berpuasa. Kemudian pada keesokan harinya ia salat Zuhur mengimami saya ketika bayang-bayang sama panjang dengan bendanya; ia salat Asar mengimami saya ketika bayang-bayang dua kali panjang bendanya; ia salat magrib mengimami saya ketika orang berpuasa berbuka; ia salat isya mengimami saya ketika menjelang berakhir sepertiga malam; dan ia salat Fajar (Subuh) mengimami saya ketika subuh sudah sangat terang. Kemudian beliau berpaling kepada saya dan berkata: Wahai Muhammad, ini adalah waktu salat para Nabi sebelum engkau. Waktu salat itu adalah antara kedua waktu ini [HR Abū Dāwūd, dan ini lafalnya, dan juga diriwayatkan oleh at-Turmuḏī dan dan Ibn Mājah].

- 4) Hadis amat panjang dari Abū Qatādah, yang potongannya adalah sebagai berikut,

ثُمَّ قَالَ أَمَا إِنَّهُ لَيْسَ فِي النَّوْمِ تَفْرِيطٌ إِنَّمَا التَّفْرِيطُ عَلَى مَنْ لَمْ يُصَلِّ الصَّلَاةَ حَتَّى يَجِيءَ وَقْتُ الصَّلَاةِ الْآخَرَى ... [رواه مسلم عن أبي قتادة].

Rasulullah saw bersabda: Ingatlah ketiduran (hingga tidak sempat salat pada waktunya) bukanlah kelalaian dalam salat. Kelalaian salat adalah tidak mengerjakan salat hingga masuk waktu salat berikutnya ... [HR Muslim dari Abū Qatādah].

C. Rincian Waktu-waktu Salat Fardu

Berdasarkan ayat-ayat al-Quran dan hadis-hadis yang dikemukakan di atas disimpulkan awal dan akhir waktu-waktu salat fardu sebagai berikut:

- 1) Waktu Salat Zuhur
Waktu salat Zuhur dimulai sejak matahari tergelincir (*zawal*), yaitu sesaat setelah matahari mencapai titik kulminasi (*culmination*) dalam peredaran hariannya, sampai tiba waktu salat Asar.
- 2) Waktu Salat Asar
Waktu salat Asar dimulai pada saat bayang-bayang suatu benda sama panjangnya dengan bendanya sendiri ditambah dengan bayang-bayang *zawal*, sampai tibanya waktu salat Magrib.
- 3) Waktu Salat Magrib
Waktu salat Magrib dimulai sejak matahari terbenam sampai tibanya waktu salat Isya.
- 4) Waktu Salat Isya

Waktu salat Isya dimulai sejak hilang mega (syafak) merah sampai masuknya waktu salat Subuh.

5) Waktu Salat Subuh

Waktu salat Subuh dimulai sejak terbit fajar sampai terbit matahari.

Perlu dicatat bahwa untuk waktu salat Isya digunakan penafsiran yang bersifat kompromistik (*al-jam'*), yaitu mengkompromikan antara hadis-hadis 'Abdullāh Ibn 'Amr, Jābir dan Ibn 'Abbās di satu pihak dengan hadis Abū Qatādah di lain pihak yang menyatakan waktu salat yang satu berakhir ketika masuknya waktu salat berikutnya. Dengan pemahaman seperti ini tidak ada satu hadis yang ditinggalkan. Dari kesemuanya dipahami bahwa waktu dari suatu salat berlangsung hingga masuk waktu salat berikutnya kecuali waktu salat Subuh hanya berlangsung sampai terbit matahari karena adanya hadis yang melarang salat setelah matahari terbit dan karena para ulama telah ijmak bahwa waktu salat Subuh hanya hingga terbit matahari dan tidak ada salat Subuh sesudah matahari terbit kecuali bagi orang yang ketiduran. Dengan demikian waktu salat Isya adalah dari sejak hilangnya syafak (mega) merah hingga masuknya waktu salat Subuh ketika terbit fajar sadik. Hadis-hadis yang menyatakan bahwa waktu salat Isya adalah hingga sepertiga malam atau hingga pertengahan malam ditafsirkan sebagai menyatakan waktu yang afdal.

D. Posisi Matahari Terkait Waktu Salat Fardu

Dari ketentuan yang termuat dalam al-Quran dan hadis sebagaimana dikemukakan di atas dapat dipahami bahwa ketentuan salat tersebut berkaitan dengan posisi matahari pada bola langit. Oleh karena itu dalam penentuan awal waktu salat, data astronomis (*zij*) terpenting adalah posisi matahari, terutama tinggi [*al-irtifā'* (*h*)], atau jarak zenit [*bu'd as-sumt* (*z*)] = $90^\circ - h$. Fenomena awal fajar (*morning twilight*), matahari terbit (*sunrise*), matahari melintasi meridian (*culmination*), matahari terbenam (*sunset*), dan akhir senja (*evening twilight*) berkaitan dengan jarak zenit matahari.³

Waktu Zuhur diawali sejak matahari berkulminasi atas, namun untuk kehati-hatian dan kepastian waktu Zuhur dirumuskan sejak seluruh bundaran matahari meninggalkan meridian, biasanya diambil sekitar 2 menit setelah kulminasi.⁴ Saat berkulminasi atas pusat bundaran matahari berada di meridian.

Awal waktu Asar ketika matahari membuat bayang-bayang sama panjang dengan bendanya ditambah bayang-bayangnya pada saat matahari berkulminasi. Ada pula yang mengambil cara mudah, yaitu mengambil tengah antara kulminasi dan gurub. Ini disimpulkan dari Ibn 'Abbās yang dikaitkan dengan bahwa hadis itu disabdakan saat matahari berkulminasi tepat di atas Kakbah sehingga tidak menciptakan bayang-

³ Moedji Raharto. "Posisi Matahari untuk Penentuan Awal Waktu Shalat dan Bayangan Arah Kiblat" makalah disampaikan dalam *Workshop Nasional Mengkaji Ulang Penentuan Awal Waktu Shalat & Arah Kiblat*, Yogyakarta Auditorium UII, 7 April 2001, h. 8.

⁴ *Ibid.* Lihat juga Mohammad Ilyas, *A Modern Guide to Astronomical Calculations of Islamic Calendar, Times & Qibla*, (Kuala Lumpur : Berita Publishing, 1984).

bayang dan waktu asar adalah ketika bayang-bayang adalah sepanjang bendanya, maka waktu Asar disimpulkan adalah pertengahan antara kulminasi dan gurub.

Waktu Magrib dalam ilmu falak berarti saat terbenam matahari (gurub) di mana seluruh piringan matahari tidak kelihatan oleh pengamat. Piringan matahari rata-rata berdiameter 32 menit busur, setengahnya berarti 16 menit busur, selain itu di dekat horison terdapat refraksi yang menyebabkan kedudukan matahari lebih tinggi dari kenyataan sebenarnya yang diasumsikan 34 menit busur. Koreksi semidiameter piringan matahari dan refraksi terhadap jarak zenit matahari saat matahari terbit atau terbenam adalah sebesar semidiameter ditambah refraksi atau sering dibulatkan sebesar 50 menit busur. Oleh karena itu terbit dan terbenam matahari secara *falak ilmi* didefinisikan bila jarak zenit matahari mencapai $z = 90^\circ 50'$.⁵ Definisi itu untuk tempat pada ketinggian di permukaan air laut atau jarak zenit matahari $z = 91^\circ$ bila memasukkan koreksi kerendahan ufuk (Dip) akibat tinggi posisi pengamat 30 meter dari permukaan laut. Secara lebih detail kedudukan matahari waktu Magrib adalah $z = 90^\circ + (s.d. + R' + Dip)$ atau $h = - (s.d. + R' + Dip)$.

Waktu Isya ditandai dengan mulai memudarnya cahaya merah (*asy-syafaq al-ahmar*) di bagian langit sebelah barat, yaitu tanda masuknya gelap malam (Q.S. Al-Isra' ayat 78). Peristiwa ini dalam *falak ilmi* dikenal sebagai akhir senja astronomi (*astronomical twilight*). Pada saat itu matahari berkedudukan 18° di bawah ufuk (*horizon*)⁶ sebelah barat atau bila jarak zenit matahari = 108° .

Waktu Subuh adalah sejak terbit fajar sadik sampai waktu terbit matahari. Cahaya fajar dalam *falak ilmi* difahami sebagai *astronomical twilight* (fajar astronomi). Dalam Ilmu Falak yang berkembang di Indonesia fajar sadik mulai muncul di ufuk timur menjelang terbit matahari pada saat matahari berada sekitar 20° di bawah ufuk (atau jarak zenit matahari = 110°).⁷

E. Langkah-langkah Hisab Awal Waktu Salat

Secara umum perhitungan awal waktu salat itu melalui empat langkah, yaitu (1) penyediaan data, (2) penyediaan rumus-rumus, (3) pemrosesan data melalui rumus, dan (4) penarikan kesimpulan. Keempat langkah ini berlaku untuk semua perhitungan awal waktu salat, bahkan untuk perhitungan waktu terbit matahari (*syurūq*). Secara rinci, langkah dan proses perhitungan awal waktu salat itu dapat dijelaskan sebagai berikut:

⁵ Sumber *The Astronomical Almanac* sebagaimana dikutip Moedji Raharto dalam "Posisi Matahari untuk Penentuan Awal Waktu Shalat dan Bayangan Arah Kiblat".

⁶ Baca Saadod'ddin Jambek, *Shalat dan Puasa di Daerah Kutub*, h. 11. Sementara itu, para ahli falak Mesir biasa menggunakan jarak zenit matahari waktu Isya = $107^\circ 30'$. Selengkapnya baca 'Abd al Qawī Zakī Ayyād. "Mawāqītu aṣ-Ṣalāh fī Khuṭūṭ al-'Arḍ al-Kabīrah", dimuat dalam *Jurnal Al-Islām al-Yaum*, Vol. 6, Th. VI/1988/1408, h. 96.

⁷ Di Indonesia pendapat ini mula pertama dikembangkan oleh Syeikh Tahir Djalaluddin Azhari. Pendapat ini kemudian diteruskan oleh Saadod'ddin Djambek dan H. Abdurrahim. Kini pendapat ini dijadikan rujukan Badan Hisab Rukyat Departemen Agama RI.

1. Penyediaan data

Data berikut ini adalah data yang dipergunakan dalam perhitungan awal waktu salat, meskipun tidak setiap data itu harus ada dalam setiap awal waktu salat. Penyediaan data ini sangat tergantung pada awal waktu salat mana yang akan dihitung.

a. Lintang Tempat atau 'Urdul-Balad (φ)

Lintang tempat ialah jarak sepanjang meridian bumi yang diukur dari ekuator bumi (khatulistiwa) sampai suatu tempat yang bersangkutan. Harga lintang tempat adalah 0° sampai 90° . Lintang tempat bagi tempat-tempat yang berada di belahan bumi utara bertanda positif (+) dan tempat-tempat yang berada di belahan bumi selatan bertanda negatif (-). Dalam astronomi disebut *latitude* yang biasanya digunakan lambang φ (*phi*).

b. Bujur Tempat atau *Tūlul-Balad* (λ)

Bujur tempat ialah jarak sepanjang ekuator bumi dihitung dari meridian yang melewati kota Greenwich sampai meridian yang melewati tempat bersangkutan. Dalam astronomi dikenal dengan nama *longitude*, biasanya digunakan lambang λ (*lamda*). Harga bujur tempat mulai 0° sampai 180° . Bagi tempat-tempat yang berada di sebelah barat Greenwich disebut "Bujur Barat (BB)" dan diberi tanda negatif (-). Sedangkan bagi tempat-tempat yang berada di sebelah timur Greenwich disebut "Bujur Timur (BT)" dan diberi tanda positif (+).

c. Bujur Tolok Waktu Daerah

Berdasarkan KEPRES No. 41 tahun 1987 Negara Republik Indonesia dibagi menjadi tiga wilayah waktu, yaitu: Waktu Indonesia Barat (WIB) dengan bujur tolok 105° BT, Waktu Indonesia Tengah (WITA) dengan bujur tolok 120° BT dan Waktu Indonesia Timur (WIT) dengan bujur tolok 135° BT.

d. Kerendahan Ufuk (*Ikhtilāf al-Ufuq / Inkhifāq* (Dip))

Kerendahan ufuk ialah perbedaan kedudukan antara ufuk yang sebenarnya (hakiki) dengan ufuk terlihat (mar'i) oleh seorang pengamat. Dalam astronomi disebut DIP dan dalam dunia falak diistilahkan dengan *ikhtilāf al-ufuq*.

Dip terjadi karena ketinggian tempat pengamatan mempengaruhi ufuk (horizon). Horizon yang teramati pada ketinggian mata sama dengan ketinggian permukaan laut disebut horizon benar (*true horizon*) atau *ufuk hissi*. Ufuk ini sejajar dengan *ufuk hakiki* yang bidangnya melalui titik pusat bumi. Horizon yang teramati oleh mata pada ketinggian tertentu di atas permukaan laut disebut horizon semu atau *ufuk mar'i*. Rumus pendekatan untuk menghitung sudut Dip adalah sebagai berikut:

$$\text{Dip} = 1,76' \sqrt{m}$$

m = Ketinggian tempat dari permukaan laut atau dari daerah sekitar (markaz).

Data ini diperlukan untuk menghitung ketinggian matahari pada saat terbenam dan terbit.

e. Semi Diameter Matahari (*Nisfu Quṭr asy-Syams*) (s.d.)

Semi Diameter Matahari ialah jarak antara titik pusat matahari dengan piringan luarnya atau seperdua garis tengah piringan matahari (jari-jari).

Data ini diperlukan untuk menghitung ketinggian matahari pada saat terbenam dan terbit.

f. Refraksi Matahari (*Daqā'iq al-Ikhtilāf* (R'))

Refraksi matahari ialah pembiasan sinar matahari. Pembiasan cahaya benda langit terjadi di dalam atmosfer bumi, menyebabkan posisi benda langit yang terlihat di permukaan bumi berbeda dengan yang sebenarnya. Refraksi membuat ketinggian posisi benda langit bertambah besar. Refraksi (R') menyatakan selisih antara ketinggian benda langit menurut penglihatan dengan ketinggian sebenarnya. R' berubah harganya menurut ketinggian benda langit. Data ini diperlukan untuk menghitung ketinggian matahari pada saat terbenam dan terbit. Besar refraksi matahari di horizon adalah 34' 30".

g. Deklinasi Matahari (*Mail asy-Syams*) (δ)

Deklinasi matahari ialah jarak matahari dari lingkaran equator diukur sepanjang lingkaran waktu yang melalui matahari itu hingga ke titik pusat matahari tersebut.

h. Perata Waktu (*Equation of time / Ta'dīl al-Waqt* (e))

Perata waktu ialah selisih waktu antara waktu matahari hakiki dengan matahari rata-rata (pertengahan). Peredaran semu harian matahari dari arah Timur ke barat itu tidaklah konstan, kadang-kadang cepat kadang-kadang lambat. Keadaan ini diakibatkan oleh percepatan bumi mengelilingi matahari tidak konstan dan bidang edarnya berbentuk ellips.

Waktu matahari hakiki ialah waktu peredaran semu matahari senyatanya, sedangkan waktu matahari pertengahan adalah waktu peredaran semu matahari diandaikan ia beredar dengan konstan sebagaimana terlihat pada jarum jam.

i. Ihtiyat (i)

Ihtiyat ialah suatu langkah kehati-hatian sebagai suatu langkah pengamanan dalam perhitungan awal waktu salat dengan cara menambah atau mengurangi sebesar 1-2 menit waktu dari hasil perhitungan yang sebenarnya. Ihtiyat ini bertujuan antara lain:

- 1) Agar hasil perhitungan dapat mencakup daerah-daerah sekitarnya, terutama yang berada di sebelah baratnya. Dengan menambah 1 (satu) menit berarti telah mencakup $\pm 27,5$ km ke sebelah barat.
- 2) Menjadikan pembulatan hasil hitungan pada satuan terkecil dalam menit waktu, sehingga penggunaannya lebih mudah.
- 3) Memberikan koreksi atas reduksi dalam perhitungan.
- 4) Memberi keyakinan bahwa waktu salat benar-benar sudah masuk, sehingga ibadah salat itu benar-benar dilaksanakan dalam waktunya.

2. Penyediaan rumus-rumus

a. Rumus Ketinggian Matahari (h)

1) Awal Waktu Salat Zuhur

Awal waktu salat Zuhur tidak memerlukan rumus ketinggian matahari, sebab secara langsung data untuk awal waktu salat Zuhur dapat dilihat pada almanak-almanak astronomis yaitu saat matahari berkulminasi.

2) Awal Waktu Salat Asar

$$\cotan h = \frac{1}{\tan(\Phi - \delta)} (z_m + 1)$$

Keterangan

h : Ketinggian matahari

Φ : Lintang tempat

δ : Deklinasi

z_m : Jarak zenit pada saat matahari berkulminasi

3) Awal Waktu Salat Magrib dan Terbit Matahari.

$$h = - (s.d. + R' + Dip)$$

$$Dip = 1,76' \sqrt{m}$$

s.d. = Semi Diameter Matahari

R' = Refraksi

Dip = Kerendahan ufuk

m = Ketinggian tempat dari permukaan laut atau dari daerah sekitar (markaz) diukur dengan ukuran meter.

4) Awal Waktu Salat Isya: ($h = -18^\circ$)

5) Awal Waktu Salat Subuh: ($h = -20^\circ$)

b. Rumus Sudut Waktu Matahari (t)

$$t = \cos^{-1} (-\tan \varphi \tan \delta + \sin h \sec \varphi \sec \delta)$$

t = Sudut waktu matahari pada awal waktu salat

φ = Lintang tempat

δ = Deklinasi matahari

h = Ketinggian matahari

c. Rumus Ephemeris Transit (e.t.)

$$e.t. = 12^j - e$$

e = Equation of Time (perata waktu)

d. Rumus Selisih Waktu Bujur

$$\text{Selisih waktu bujur (sw}\lambda) = \lambda_{tp} - \lambda_{dh}; 15$$

λ_{tp} = Bujur tempat

λ_{dh} = Bujur tolok waktu daerah

e. Rumus Penyimpulan

- 1) Awal waktu salat Zuhur = e.t.-(atau+) selisih waktu bujur+i
- 2) Awal waktu salat Asar, Magrib dan Isya=(e.t.+t)-(atau+) selisih waktu bujur+i
- 3) Awal waktu salat Subuh = (e.t.-t) – (atau +) selisih waktu bujur + i
- 4) Waktu terbit (syuruq) = (e.t.-t) – (atau +) selisih waktu bujur - i

F. Contoh Hisab Waktu Salat

1. Awal waktu salat Zuhur untuk kota Yogyakarta dan sekitarnya menurut WIB pada tanggal 6 Desember 2008.

1) Data:

- a. Lintang tempat (φ) = $-07^{\circ} 48'$
- b. Bujur tempat (λ) = $110^{\circ} 21'$ BT.
- c. Equation of Time (e) = $+08^m 54^d$

2) Rumus awal waktu Zuhur

$$\text{Zuhur} = \text{e.t.} -/+ (\text{sw}\lambda) + i$$

$$\text{e.t.} = 12^j - e$$

3) Proses Perhitungan

Ephemeris Transit (e.t.)	= $12^j - e$
	= $12^j - 08^m 54^d$
	= $11^j 51^m 06^d$ WST
Selisih waktu bujur (sw λ)	= $/110^{\circ} 21' - 105^{\circ} / : 15$
	= $05^{\circ} 21' : 15$
	= $21^m 24^d$
e.t. - sw λ	= $11^j 51^m 06^d - 21^m 24^d$
	= $11^j 29^m 42^d$ WIB
Ihtiyat	= $\frac{01^m 18^d}{+}$
Zuhur	= $11^j 31^m$ WIB

Kesimpulan: Awal waktu salat Zuhur di Yogyakarta tanggal 6 Desember 2008 pukul 11:31 WIB.

2. Awal waktu salat Asar untuk kota Yogyakarta dan sekitarnya menurut WIB pada tanggal 6 Desember 2008.

1) Data:

- a. Lintang tempat (φ) = $-07^{\circ} 48'$
- b. Bujur tempat (λ) = $110^{\circ} 21'$ BT.
- c. Equation of Time (e) = $+08^m 51^d$
- d. Deklinasi matahari (δ) = $-22^{\circ} 32' 53''$

2) Rumus awal waktu Asar

$$\text{Asar} = [(e.t.+t) -/+ sw\lambda] + i$$

$$e.t. = 12^j - e$$

$$t = \cos^{-1} (-\tan \varphi \tan \delta +) \frac{\sin h}{\cos \varphi \cos \delta}$$

$$h = \cotan^{-1}(\tan z_m + 1)$$

$$z_m = /\varphi - \delta/$$

$$sw\lambda = / \lambda_{tp} - \lambda_{dh} / : 15$$

3) Proses Perhitungan

$$\text{Tinggi matahari (h): } \cotan h = \tan z_m + 1$$

$$z_m = / \varphi - \delta /$$

$$z_m = / -07^\circ 48' - (-22^\circ 32' 53'') /$$

$$= 14^\circ 44' 52''$$

$$\cotan h = \tan z_m + 1$$

$$= \tan 14^\circ 44' 52'' + 1$$

$$= 0,263236555 + 1$$

$$= 1,263236555$$

$$h = 38^\circ 21' 56,28''$$

Sudut waktu matahari (t):

$$\cos t = -\tan \varphi \tan \delta + \frac{\sin h}{\cos \varphi \cos \delta}$$

$$= -\tan -07^\circ 48' \tan -22^\circ 32' 53'' + \frac{\sin 38^\circ 21' 56,28''}{\cos -07^\circ 48' \cos -22^\circ 32' 53''}$$

$$= 0,13698296 \times (-0,41519085) + \frac{0,620679196}{0,99074784 \times 0,923560099}$$

$$= -0,05687407 + \frac{0,620679196}{0,915015174}$$

$$= -0,05687407 + 0,67832667$$

$$= 0,621452598$$

$$t = 51^\circ 34' 40,07'' : 15 = 03^j 26^m 18,67^d$$

$$e.t. = 12^j - e = 12^j - 08^m 54^d = 11^j 51^m 9^d +$$

$$e.t. + t = 15^j 17^m 27,67^d \text{ WST}$$

$$\text{Selisih waktu bujur (sw}\lambda) = /110^\circ 21' - 105^\circ / : 15$$

$$= 05^\circ 21' : 15$$

$$\begin{aligned}
 &= 21^m 24^d \\
 (e.t. + t) - sw\lambda &= 15^j 17^m 27,67^d - 21^m 24^d \\
 &= 14^j 56^m 03,67^d \text{ WIB} \\
 Ihtiyat &= 01^m 56,33^d + \\
 \hline
 Asar &= 14^j 58^m \text{ WIB}
 \end{aligned}$$

Kesimpulan: Awal waktu salat Asar di Yogyakarta tanggal 6 Desember 2008 pukul 14:58 WIB.

3. Awal waktu salat Magrib untuk kota Yogyakarta dan sekitarnya menurut WIB pada tanggal 6 Desember 2008.

1) Data:

- Lintang tempat (φ) = $-07^\circ 48'$
- Bujur tempat (λ) = $110^\circ 21' \text{ BT}$.
- Tinggi tempat (m) = 90 meter
- Equation of Time (e) = $08^m 48^d$
- Deklinasi matahari (δ) = $-22^\circ 33' 45''$
- Semi diameter matahari (s.d.) = $16' 14,03''$
- Refraksi matahari (R') = $34' 30''$

2) Rumus awal waktu Magrib

$$\text{Magrib} = [(e.t.+t) -/+ sw\lambda] + i$$

$$e.t. = 12^j - e$$

$$t = \cos^{-1} \left(-\tan \varphi \tan \delta + \frac{\sin h}{\cos \varphi \cos \delta} \right)$$

$$h = -(s.d. + R' + \text{Dip})$$

$$\text{Dip} = 1,76' \sqrt{m}$$

$$sw\lambda = /\lambda tp - \lambda dh / : 15$$

3) Proses Perhitungan

$$\text{Dip} = 1,76' \sqrt{m}$$

$$= 1,76' \sqrt{90}$$

$$= 16' 41,81''$$

$$\text{Tinggi matahari } (h_m): h = -(s.d. + R' + \text{Dip})$$

$$= -(16' 14,03'' + 34' 30'' + 16' 41,81'')$$

$$= -01^\circ 07' 25,84''$$

Sudut waktu matahari (t):

$$\cos t = -\tan \varphi \tan \delta + \frac{\sin h}{\cos \varphi \cos \delta}$$

$$\begin{aligned}
&= -\tan -07^\circ 48' \tan -22^\circ 23' 45'' + \frac{\sin -01^\circ 07' 25,84''}{\cos -07^\circ 48' \cos -22^\circ 33' 45''} \\
&= 0,13698296 \times (-0,41205115) + \frac{-0,01961347}{0,99074784 \times 0,9234727} \\
&= -0,05644398 + \frac{-0,01961347}{0,914928584} \\
&= -0,05644398 + (-0,02143717) \\
&= -0,07788115 \\
t &= 94^\circ 29' 38,01'' : 15 &= 06^j 17^m 58,53^d \\
e.t. &= 12^j - e = 12^j - 08^m 54^d &= 11^j 51^m 12^d +
\end{aligned}$$

$$\begin{aligned}
e.t. + t &= 18^j 9^m 10,53^d \text{ WST} \\
\text{Selisih waktu bujur (sw}\lambda) &= /110^\circ 21' - 105^\circ / : 15 \\
&= 05^\circ 21' : 15 \\
&= 21^m 24^d \\
(e.t. + t) - \text{sw}\lambda &= 18^j 9^m 10,53^d - 21^m 24^d \\
&= 17^j 47^m 46,53^d \text{ WIB} \\
\text{Ihtiyat} &= 01^m 13,47^d + \\
\text{Magrib} &= 17^j 49^m \text{ WIB}
\end{aligned}$$

Kesimpulan: Awal waktu salat Magrib di Yogyakarta tanggal 6 Desember 2008 pukul 17:49 WIB.

4. Awal waktu salat Isya untuk kota Yogyakarta dan sekitarnya menurut WIB pada tanggal 6 Desember 2008.

1) Data:

- Lintang tempat (φ) = $-07^\circ 48'$
- Bujur tempat (λ) = $110^\circ 21'$ BT.
- Equation of Time* (e) = $08^m 47^d$
- Deklinasi matahari (δ) = $-22^\circ 34' 02''$
- Tinggi matahari (h) = -18°

2) Rumus awal salat Isya

$$\text{Isya} = [(e.t.+t) -/+ \text{sw}\lambda] + i$$

$$e.t. = 12^j - e$$

$$t = \cos^{-1} \left(-\tan \varphi \tan \delta + \frac{\sin h}{\cos \varphi \cos \delta} \right)$$

$$\text{sw}\lambda = / \lambda_{tp} - \lambda_{dh} / : 15$$

3) Proses Perhitungan

Sudut waktu matahari (t):

$$\cos t = -\tan \varphi \tan \delta + \frac{\sin h}{\cos \varphi \cos \delta}$$

$$= -\tan -07^\circ 48' \tan -22^\circ 34' 02'' + \frac{\sin -18^\circ}{\cos -07^\circ 48' \cos -22^\circ 34' 02''}$$

$$= 0,13698296 \times (-0,41558877) + \frac{-0,30901699}{0,99074784 \times 0,923429913}$$

$$= -0,05692858 + \frac{-0,30901699}{0,914886193}$$

$$= -0,05692858 + (-0,3377655)$$

$$= -0,39469408$$

$$t = 113^\circ 14' 48,80'' : 15 = 07^j 32^m 59,26^d$$

$$\text{e.t.} = 12^j - e = 12^j - 08^m 47^d = 11^j 51^m 13^d +$$

$$\text{e.t.} + t = 19^j 24^m 12,26^d \text{ WST}$$

$$\text{Selisih waktu bujur (sw}\lambda) = /110^\circ 21' - 105^\circ / : 15$$

$$= 05^\circ 21' : 15$$

$$= 21^m 24^d$$

$$\bullet (\text{e.t.} + t) - \text{sw}\lambda = 19^j 24^m 12,26^d - 21^m 24^d$$

$$= 19^j 02^m 48,26^d \text{ WIB}$$

$$\text{Ihtiyat} = 01^m 11,74^d +$$

$$\text{Isya} = 19^j 04^m \text{ WIB}$$

Kesimpulan: Awal waktu salat Isya di Yogyakarta tanggal 6 Desember 2008 pukul 19:04 WIB.

5. Awal waktu salat Subuh untuk kota Yogyakarta dan sekitarnya menurut WIB pada tanggal 6 Desember 2008.

1) Data:

- a. Lintang tempat (φ) = $-07^\circ 48'$
- b. Bujur tempat (λ) = $110^\circ 21'$ BT.
- c. Equation of Time (e) = $+09^m 03^d$
- d. Deklinasi matahari (δ) = $-22^\circ 29' 39''$
- e. Tinggi matahari (h) = -20°

- 2) Rumus awal salat Subuh
 Subuh = [(e.t.-t) +/- swλ] + i
 e.t. = 12^j - e

$$t = \cos^{-1}(-\tan \varphi \tan \delta + \frac{\sin h}{\cos \varphi \cos \delta})$$

$$sw\lambda = /\lambda_{tp} - \lambda_{dh} / : 15$$

- 3) Proses Perhitungan
 Sudut waktu matahari (t):

$$\cos t = -\tan \varphi \tan \delta + \frac{\sin h}{\cos \varphi \cos \delta}$$

$$= -\tan -07^{\circ} 48' \tan -22^{\circ} 29' 39'' + \frac{\sin -20^{\circ}}{\cos -07^{\circ} 48' \cos -22^{\circ} 29' 39''}$$

$$= -0,13698296 \times (-0,4140886) + \frac{-0,34202014}{0,99074784 \times 0,923920344}$$

$$= 0,13698296 \times (-0,4140886) + \frac{-0,34202014}{0,915372085}$$

$$= -0,05672308 + \frac{-0,34202014}{0,915372085}$$

$$= -0,05672308 + (-0,37364056)$$

$$= -0,43036364$$

$$t = 115^{\circ} 29' 26,6'' : 15 = 07^j 41^m 57,78^d$$

$$e.t. = 12^j - e = 12^j - 08^m 54^d = 11^j 50^m 57^d$$

$$= 07^j 41^m 57,78^d -$$

$$e.t. - t = 04^j 08^m 59,22^d \text{ WST}$$

$$\text{Selisih waktu bujur (sw}\lambda) = /110^{\circ} 21' - 105^{\circ} / : 15$$

$$= 05^{\circ} 21' : 15$$

$$= 21^m 24^d$$

$$(e.t. - t) - sw\lambda = 04^j 08^m 59,22^d - 21^m 24^d$$

$$= 03^j 47^m 35,22^d \text{ WIB}$$

$$\text{Ihtiyat} = 01^m 26,78^d +$$

$$\text{Subuh} = 03^j 49^m \text{ WIB}$$

Kesimpulan: Awal waktu salat Subuh di Yogyakarta tanggal 6 Desember 2008 pukul 03:49 WIB.

6. Waktu terbit matahari (syuruq) untuk kota Yogyakarta dan sekitarnya menurut WIB pada tanggal 6 Desember 2008.

1) Data:

- a. Lintang tempat (φ) = $-07^{\circ} 48'$
- b. Bujur tempat (λ) = $110^{\circ} 21'$ BT.
- c. Tinggi tempat (m) = 90 meter
- d. Equation of Time (e) = $+09^m 02^d$
- e. Deklinasi matahari (δ) = $-22^{\circ} 29' 59''$
- f. Semi diameter matahari (s.d.) = $16' 13,95''$
- g. Refraksi matahari (R') = $34' 30''$

2) Rumus terbit Matahari

$$\text{Magrib} = [(e.t.-t) -/+ sw\lambda] - i$$

$$e.t. = 12^j - e$$

$$t = \cos^{-1} \left(-\tan \varphi \tan \delta + \frac{\sin h}{\cos \varphi \cos \delta} \right)$$

$$h = -(s.d. + R' + \text{Dip})$$

$$\text{Dip} = 1,76' \sqrt{m}$$

$$sw\lambda = / \lambda_{tp} - \lambda_{kb} / : 15$$

3) Proses Perhitungan

$$\text{Dip} = 1,76' \sqrt{m}$$

$$= 1,76' \sqrt{90}$$

$$= 16' 41,81''$$

Tinggi matahari (h):

$$h = -(s.d. + R' + \text{Dip})$$

$$= -(16' 13,95'' + 34' 30'' + 16' 41,81'')$$

$$= -01^{\circ} 07' 25,76''$$

Sudut waktu matahari (t):

$$\cos t = -\tan \varphi \tan \delta + \frac{\sin h}{\cos \varphi \cos \delta}$$

$$= -\tan -07^{\circ} 48' \tan -22^{\circ} 29' 59'' + \frac{\sin -01^{\circ} 07' 25,76''}{\cos -07^{\circ} 48' \cos -22^{\circ} 29' 59''}$$

$$= -0,13698296 \times (-0,41418516) + \frac{-0,01961314}{0,99074784 \times 0,923888808}$$

$$= 0,13698296 \times (-0,41418516) + \frac{-0,01961314}{0,99074784 \times 0,923888808}$$

$$\begin{aligned}
&= -0,0567363 + \frac{-0,01961314}{0,915340842} \\
&= -0,0567363 + (-0,02142714) \\
&= -0,07816345 \\
t &= 94^\circ 28' 59,51'' : 15 = 06^j 17^m 55,97^d \\
e.t. &= 12^j - e = 12^j - 9^m 02^d = 11^j 50^m 58^d \\
t &= 06^j 17^m 55,97^d - \\
\hline
e.t. - t &= 05^j 33^m 02,03^d \text{ WST} \\
\text{Selisih waktu bujur (sw}\lambda) &= /110^\circ 21' - 105^\circ / : 15 \\
&= 05^\circ 21' : 15 \\
&= 21^m 24^d \\
(e.t. - t) - \text{sw}\lambda &= 05^j 33^m 02,03^d - 21^m 24^d \\
&= 05^j 11^m 38,03^d \text{ WIB} \\
\text{Ihtiyat} &= 01^m 38,03^d - \\
\hline
\text{Terbit} &= 05^j 10^m \text{ WIB}
\end{aligned}$$

Kesimpulan: Waktu terbit matahari di Yogyakarta tanggal 16 Desember 2008 pukul 05:10 WIB.

The author profile for Kasmui is presented on a dark blue background with a golden circuit board pattern. At the top, the word "PENULIS" is written in large, bold, golden letters. Below it is a square portrait of Kasmui, a man wearing a white cap and a blue batik shirt. Underneath the portrait, the name "KASMUI" is written in bold, golden letters. To the left of the portrait are icons of a brain and a compass, and to the right are icons of an open book and a closed book. Below the name, a list of bullet points describes his professional and personal background. A small white star icon is located at the bottom right of the profile area.

PENULIS



KASMUI

- Dosen Kimia, Komputasi, IT, dan AI UNNES, serta Praktisi Ilmu Falak;
- Anggota Majelis Tabligh PDM Kota Semarang dan PWM Jawa Tengah;
- Anggota Tim Pengembang Software KHGT MTT PP Muhammadiyah;
- Website pribadi: <https://hisabmu.com/>, <https://kasmui.cloud/>;
- Minat & Hobi: Computer programming.

Silabus Perkuliahan: Python untuk Astronomi Islam

Buku ini menyajikan panduan komprehensif untuk menguasai Python dalam konteks Astronomi Islam.

Melalui pendekatan silabus yang terstruktur, pelajar akan dibimbing dari dasar-dasar pemrograman hingga aplikasi praktis untuk menghitung fase bulan, arah kiblat, waktu shalat, dan visualisasi Hilal. Dilengkapi dengan contoh kode nyata, analisis data, dan integrasi pustaka astronomi modern, buku ini menjadi jembatan antara sains data dan tradisi Falakiah, memberdayakan pelajar untuk mengeksplorasi kosmos dengan presisi modern.

```
python = 1
def rework_kabupaten():
    try:
        print("Berikut adalah kabupaten, "sbluat")
    else:
        print("Error: ")
    print("Berikut adalah kabupaten")
```

```
python -- 1
code = 123
```

```
python {
    baint +
    patofat
```

```
asf msincx
try:
    print("Berikut adalah kabupaten")
else:
    print("Error: ")
    Letter()
    nown()
    rasi: bintange')
```