

SOLUNA:

Software Astronomi Matahari dan Bulan

KHGT Times versi online
berbasis data ephemeris DE440s

Landasan ilmiah, panduan penggunaan,
algoritma kalender, dan validasi versi

Naskah Ilmiah • 5 Oktober 2026

Dilengkapi tabel validasi 1448 H, katalog lokasi,
kamus data, glosarium, dan referensi



KASMUI

SOLUNA: Software Astronomi Matahari dan Bulan

KHGT Times versi online
berbasis data ephemeris DE440s

<https://hisabmu.com/khgttimes/>
<https://kasmui.cloud/soluna/>

Landasan ilmiah, panduan penggunaan,
algoritma kalender, dan validasi versi

Naskah Ilmiah • 5 Oktober 2026

Dilengkapi tabel validasi 1448 H, katalog lokasi,
kamus data, glosarium, dan referensi

KATA PENGANTAR

SOLUNA hadir pada pertemuan antara astronomi komputasional, kebutuhan kalender, dan pembelajaran ilmu falak. Posisi Matahari dan Bulan dapat dihitung dengan perangkat lunak modern, tetapi angka yang tampil di layar baru bermakna setelah pembaca mengetahui titik acuannya, definisi waktunya, model atmosfernya, dan aturan yang dipakai untuk mengambil keputusan. Sebuah nilai tinggi hilal, misalnya, tidak dapat dibandingkan secara bertanggung jawab dengan nilai lain jika yang satu geosentris dan yang lain toposentris. Demikian pula, tanggal awal bulan tidak ditentukan oleh waktu ijtima saja. Ada hubungan antara geometri hilal, saat terbenam Matahari, batas hari UTC, dan ketentuan kalender yang harus dibaca sebagai satu rangkaian.

Buku ini disusun untuk menjelaskan SOLUNA: Software Astronomi Matahari dan Bulan, dalam konteks KHGT Times versi daring berbasis ephemeris DE440s. Tujuan utamanya ialah menyediakan dasar ilmiah dan panduan kerja yang cukup bagi pembaca untuk menggunakan keluaran aplikasi secara cermat, memahami cara keluaran itu dihasilkan, dan menilai batas kesimpulan yang dapat ditarik darinya. Buku ini tidak hanya menjadi petunjuk menekan tombol. Ia juga menguraikan geometri langit, koordinat, skala waktu, pengolahan kernel ephemeris, pencarian peristiwa, serta hubungan antara hasil astronomi dan parameter Kalender Hijriah Global Tunggal atau KHGT.

Pembaca yang dituju mencakup mahasiswa dan dosen ilmu falak, peminat astronomi, pengelola kegiatan pengamatan, pengembang perangkat lunak pendidikan, serta peneliti yang memerlukan alat awal untuk menelaah kalender. Pengetahuan trigonometri dasar membantu dalam mengikuti penurunan rumus, tetapi pembaca dapat lebih dahulu mempelajari panduan penggunaan dan contoh hasil. Istilah teknis dijelaskan ketika pertama kali dipakai dan dihimpun kembali dalam glosarium. Besaran sudut, jarak, waktu, dan satuan dipertahankan secara konsisten agar pembaca dapat menelusuri hubungan antara teori, nilai numerik, dan tampilan aplikasi.

Ruang lingkup buku mencakup perhitungan posisi Matahari dan Bulan, fase dan iluminasi, peristiwa terbit dan terbenam, transit, twilight, jadwal berbasis ketinggian Matahari, fenomena orbital, ringkasan gerhana, serta algoritma awal bulan global. Penjelasan tentang KHGT berpusat pada syarat tinggi geosentris sekurang-kurangnya 5° dan elongasi geosentris sekurang-kurangnya 8° , yang disertai mekanisme PKG 1 dan PKG 2. Pemisahan antara syarat angka dan syarat waktu sangat diperlukan. Hilal yang melampaui dua ambang sudut tersebut belum otomatis diterima sebagai saksi kalender apabila kedudukan waktunya tidak memenuhi ketentuan yang relevan.

Penyusunan didasarkan pada pembacaan kode sumber SOLUNA yang disediakan, dokumentasi resmi Skyfield, publikasi ephemeris JPL, dokumentasi SPICE/NAIF, sumber astronomi dari US Naval Observatory dan NASA, serta literatur penelitian visibilitas hilal. Untuk ketentuan KHGT, buku memakai sumber resmi Majelis Tarjih dan Tajdid PP Muhammadiyah. Literatur internasional digunakan untuk menjelaskan landasan astronomi dan metode komputasi; sumber kelembagaan dipakai untuk menjelaskan aturan kalender yang memang menjadi domain lembaga tersebut. Kedua jenis sumber ini mempunyai fungsi berbeda dan tidak dipertukarkan.

Versi program yang dibahas memiliki identitas berkas yang dicatat dalam lampiran reproduksibilitas. Kode yang dilampirkan untuk penyusunan buku identik dengan versi yang telah diuji terhadap draf kalender KHGT 1448 H. Karena itu, buku dapat menyertakan hasil pengujian bulanan yang nyata. Seluruh dua belas tanggal awal bulan pada himpunan tersebut sesuai dengan tanggal dalam PDF pembandingan. Namun, kesesuaian tanggal itu tidak berarti bahwa seluruh lokasi saksi, koordinat, atau waktu pada kedua sumber identik. Perbedaan tersebut dibahas secara terbuka agar pembaca tidak mencampurkan kecocokan keluaran kalender dengan pembuktian menyeluruh atas seluruh kemampuan perangkat lunak.

Sistematika buku bergerak dari tujuan dan struktur SOLUNA menuju geometri sistem Matahari–Bumi–Bulan, koordinat dan waktu, mesin Skyfield, serta berkas DE440s. Setelah fondasi itu, pembaca mengikuti penggunaan antarmuka, interpretasi hilal lokal, keputusan KHGT, peristiwa harian dan jadwal, serta fenomena dan gerhana. Bagian akhir menjelaskan validitas, reliabilitas, dan tata kerja akademik yang dapat diterapkan tanpa mengubah aplikasi. Lampiran menyediakan kamus data, katalog lokasi pemindaian, petunjuk penyiapan berkas, dan identitas pengujian. Urutan ini membantu pembaca memahami alasan suatu hasil sebelum menjadikannya bahan laporan.

Manfaat yang diharapkan ialah tumbuhnya kebiasaan membaca hasil astronomi dengan definisi yang jelas. Pengguna dapat membedakan jam perangkat dari waktu simulasi, posisi geosentris dari posisi pengamat, tinggi geometris dari tinggi yang dibias atmosfer, serta keputusan kalender dari kemungkinan pengamatan nyata. Peneliti dapat mencatat versi perangkat lunak dan ephemeris, menyimpan hasil, lalu melakukan perbandingan dengan sumber yang setara. Pendidik dapat memakai perubahan parameter untuk memperlihatkan hubungan sebab akibat, tanpa menjadikan angka yang terlihat presisi sebagai pengganti penalaran ilmiah.

Beberapa keterbatasan perlu ditempatkan pada konteksnya. Pemindaian kota pada SOLUNA merupakan pengujian atas katalog titik, bukan pencarian kontinu di setiap daratan bumi. Cuaca, profil horizon, ekstingsi atmosfer, dan kemampuan

alat pengamatan belum dimodelkan secara rinci. Sejumlah pencarian fenomena memakai langkah waktu terbatas. Ringkasan gerhana Matahari menggunakan pendekatan geometri bayangan dan belum menggantikan perhitungan kontak lokal berbasis elemen Besselian. Batas seperti ini tidak menghilangkan nilai pendidikan aplikasi; justru memberi arah tentang pengujian tambahan yang diperlukan sebelum hasil digunakan untuk tuntutan presisi yang lebih tinggi.

Penghargaan layak diberikan kepada para pengembang dan lembaga yang menyediakan ephemeris, dokumentasi terbuka, serta literatur yang memungkinkan metode astronomi diperiksa. Buku ini menyebut sumber tersebut secara spesifik dalam daftar pustaka, tanpa mengandaikan adanya kerja sama atau pengesahan terhadap SOLUNA. Harapannya, naskah ini mendukung penggunaan perangkat lunak yang dapat dijelaskan, diperiksa, dan direproduksi. Pembaruan aplikasi pada masa mendatang hendaknya diikuti pembaruan dokumentasi, karena pengetahuan tentang versi merupakan bagian dari mutu ilmiah suatu hasil komputasi.

Semarang, 5 Oktober 2026

Penyusun,

A handwritten signature in black ink, appearing to be 'Kasmui', with a stylized, flowing script.

Kasmui

PANDUAN MEMBACA DAN KONVENSI

Buku ini merupakan naskah ilmiah dan panduan penggunaan untuk versi SOLUNA yang diaudit, bukan pernyataan bahwa setiap implementasi dengan nama sama mempunyai perilaku identik. Uraian yang menyebut fungsi, tombol, ambang, jendela pencarian, atau pemilihan berkas berasal dari kode sumber yang disediakan. Keterangan umum astronomi memakai referensi bernomor dalam kurung siku. Daftar pustaka memuat judul, penulis atau lembaga, identitas publikasi, dan tautan yang dapat ditelusuri.

Pembaca yang ingin langsung menggunakan aplikasi dapat mulai dari Bab 6, lalu membaca Bab 7 dan Bab 8 sebelum menafsirkan keputusan hilal. Pembaca yang mempelajari teori dapat mengikuti Bab 1 sampai Bab 5 secara berurutan. Pembaca yang melakukan audit dapat mengutamakan Bab 8, Bab 10, Bab 11, Bab 12, dan lampiran reproduksibilitas. Jalur tersebut merupakan pilihan membaca, bukan pemisahan materi yang berdiri sendiri.

Notasi ϕ menyatakan lintang geografis, λ menyatakan bujur geografis ketika digunakan dalam persamaan waktu sidereal, α menyatakan asensio rekta, δ menyatakan deklinasi, H menyatakan sudut jam, h menyatakan tinggi, A menyatakan azimut, dan E menyatakan elongasi. Huruf λ juga lazim dipakai untuk bujur ekliptika; konteks dan subskrip membedakannya. UTC selalu ditulis bersama tanggalnya apabila suatu peristiwa dapat melintasi tengah malam. Waktu lokal memakai identitas zona IANA, seperti Asia/Jakarta atau Pacific/Auckland.

Tanggal tampilan menggunakan format Indonesia, misalnya 12-10-2026 atau 12 Oktober 2026. Bilangan desimal dalam penjelasan menggunakan koma, sedangkan nama kolom program dan data mesin dapat memakai titik desimal. Ini adalah perbedaan format penyajian, bukan perubahan nilai. Jam ditulis 18.30.00 untuk tampilan Indonesia; label UTC atau nama zona harus tetap menyertai waktu yang dibandingkan lintas wilayah.

Istilah “memenuhi” pada filter lokal berarti melewati ambang numerik yang diprogram. Istilah “diterima” pada keputusan global berarti melewati filter numerik dan syarat PKG. Istilah “teramati” membutuhkan bukti pengamatan. Ketiga istilah tersebut tidak digunakan sebagai sinonim. “Kota pertama” dalam hasil SOLUNA berarti saksi yang dipilih menurut urutan waktu dan prosedur pada sampel yang benar-benar dievaluasi.

DAFTAR ISI

KATA PENGANTAR	3
PANDUAN MEMBACA DAN KONVENSI	6
DAFTAR ISI	7
BAB 1 SOLUNA DALAM ASTRONOMI DAN KALENDER.....	12
1.1 Tujuan perangkat lunak.....	12
1.2 Empat lapisan yang harus dibedakan	12
1.3 Daring dengan perhitungan di peramban	13
1.4 Ruang lingkup dan batas produk.....	14
1.5 Cara membaca satu hasil secara bertanggung jawab	14
BAB 2 GEOMETRI MATAHARI BUMI DAN BULAN	15
2.1 Gerak yang membentuk gejala langit.....	15
2.2 Fase sebagai geometri pencahayaan.....	15
2.3 Siklus bujur dan nama fase	16
2.4 Ijtimak dan separasi minimum	16
2.5 Jarak diameter dan paralaks	16
2.6 Umur Bulan dan lag	17
2.7 Implikasi untuk pembacaan hilal	17
BAB 3 KOORDINAT SKALA WAKTU DAN HORIZON.....	18
3.1 Lokasi pengamat dan tanda koordinat.....	18
3.2 Ekuatorial ekliptika dan horizontal	18
3.3 Dari RA dan deklinasi ke tinggi.....	19
3.4 Geosentris dan toposentris	19
3.5 Apparent dan refraksi atmosfer.....	20
3.6 UTC UT1 TT dan TDB.....	20
3.7 Zona IANA dan tanggal yang melintasi batas hari	21
3.8 Jam perangkat dan waktu simulasi.....	21
BAB 4 SKYFIELD DAN PROSES KOMPUTASI	22
4.1 Kedudukan Skyfield.....	22
4.2 Rantai vektor dan pengamatan	22

4.3 Pencarian peristiwa diskret	22
4.4 Pencarian minimum dan maksimum	23
4.5 Perkiraan magnitudo	23
4.6 Kegagalan peristiwa dan data kosong	24
4.7 Reproduksiabilitas lingkungan.....	24
BAB 5 DE440S DAN PENGELOLAAN BERKAS BSP	25
5.1 Ephemeris Development Ephemeris	25
5.2 Struktur SPK dan representasi Chebyshev.....	25
5.3 Berkas yang dipilih SOLUNA	26
5.4 Data penyangga pada berkas tahunan	26
5.5 Fallback yang terbatas.....	27
5.6 Pemeriksaan format dan integritas	27
5.7 Cache antrean dan pemeliharaan.....	27
BAB 6 PANDUAN ANTARMUKA DAN PENGGUNAAN	29
6.1 Susunan ruang kerja	29
6.2 Penyiapan pertama	29
6.3 Memilih dan memeriksa lokasi	29
6.4 Jam manual dan maghrib lokal	30
6.5 Memilih mode koordinat.....	30
6.6 Membaca ringkasan dan tab.....	30
6.7 Hasil global dan label kalender	31
6.8 Menyalin dan mengunduh hasil	32
6.9 Prosedur dokumentasi yang disarankan	32
BAB 7 HILAL LOKAL DAN INTERPRETASI VISIBILITAS	33
7.1 Kedudukan filter lokal.....	33
7.2 Empat parameter pada maghrib	33
7.3 Pemindaian kandidat bulanan	34
7.4 Mengapa visibilitas memerlukan lebih banyak informasi	34
7.5 Geosentris dalam kalender dan toposentris dalam observasi.....	34
7.6 Ambang pembulatan dan nilai dekat batas.....	35
7.7 Menghubungkan hasil dengan catatan rukyat	35
BAB 8 KHGT PKG 1 DAN PKG 2.....	36

8.1 Dasar penyatuan kalender	36
8.2 Definisi operasional dalam SOLUNA	36
8.3 Filter numerik geosentris	36
8.4 PKG 1 sebelum tengah malam UTC	37
8.5 PKG 2 pada atau setelah tengah malam UTC	37
8.6 Bila koreksi tidak diterima	38
8.7 Prioritas sampel dan pemilihan saksi	38
8.8 Jumadilawal 1448 H.....	39
8.9 Jumadilakhir 1448 H.....	40
8.10 Syawal 1448 H sebagai PKG 2	40
8.11 Penamaan bulan dan audit tahun lain.....	40
BAB 9 PERISTIWA HARIAN DAN JADWAL.....	42
9.1 Terbit dan terbenam sebagai definisi model	42
9.2 Transit dan kulminasi.....	42
9.3 Twilight dan keadaan langit	42
9.4 Persamaan waktu dan panjang siang.....	43
9.5 Parameter jadwal yang diprogram	43
9.6 Model bayangan Ashar	44
9.7 Tengah malam dan sepertiga terakhir	44
9.8 Pemakaian dalam pendidikan dan observasi.....	45
BAB 10 FENOMENA ORBITAL DAN GERHANA.....	46
10.1 Jendela pencarian fenomena	46
10.2 Node perigee dan apogee	46
10.3 Mengapa gerhana tidak terjadi setiap bulan.....	47
10.4 Gerhana Bulan pada versi SOLUNA	47
10.5 Pendekatan gerhana Matahari	47
10.6 Uji lokal pada maksimum global	48
10.7 Referensi pembanding yang tepat	48
10.8 Keselamatan pengamatan Matahari	48
BAB 11 VALIDITAS DAN RELIABILITAS VERSI YANG DIUJI	50
11.1 Pertanyaan dan bahan pengujian.....	50
11.2 Desain pengulangan	50

11.3 Hasil tanggal bulanan.....	51
11.4 Perbandingan lokasi saksi	52
11.5 Pemeriksaan ulang titik referensi	52
11.6 Reliabilitas yang berhasil diperiksa	53
11.7 Akses BSP dan waktu eksekusi.....	53
11.8 Matriks batas kesimpulan.....	54
BAB 12 TATA KERJA AKADEMIK DAN PENGEMBANGAN	55
12.1 Memulai dari pertanyaan penelitian.....	55
12.2 Rekaman minimal yang dapat direplikasi	55
12.3 Perbandingan dengan sumber independen	55
12.4 Ketidakpastian dan sensitivitas	56
12.5 Cakupan geografis dan pencarian kontinu	56
12.6 Uji yang memberi nilai ilmiah	56
12.7 Pembelajaran tanpa menyederhanakan berlebihan	57
12.8 Dokumentasi versi dan pemeliharaan	57
LAMPIRAN A KAMUS DATA DAN SATUAN	58
A.1 Posisi Matahari dan Bulan	58
A.2 Peristiwa dan durasi	59
A.3 Audit keputusan global.....	60
LAMPIRAN B KATALOG TITIK PEMINDAIAN	62
B.1 Seratus titik utama	62
B.2 Tiga puluh titik tambahan Amerika.....	68
B.3 Pembacaan geografis dan pemeliharaan.....	70
LAMPIRAN C PENYIAPAN DAN PEMECAHAN MASALAH	72
C.1 Menyediakan aset yang sesuai.....	72
C.2 Setup lambat atau berkas tidak ditemukan	72
C.3 Tabel kosong atau peristiwa hilang	72
C.4 Ketidaksesuaian angka atau tanggal.....	73
LAMPIRAN D IDENTITAS DAN PROTOKOL REPRODUKSI.....	74
D.1 Identitas sumber dan lingkungan.....	74
D.2 Protokol perbandingan bulanan.....	74
D.3 Uji batas dan akses data	75

LAMPIRAN E SOURCE CODE SOLUNA	76
GLOSARIUM.....	216
DAFTAR PUSTAKA	220

BAB 1 SOLUNA DALAM ASTRONOMI DAN KALENDER

1.1 Tujuan perangkat lunak

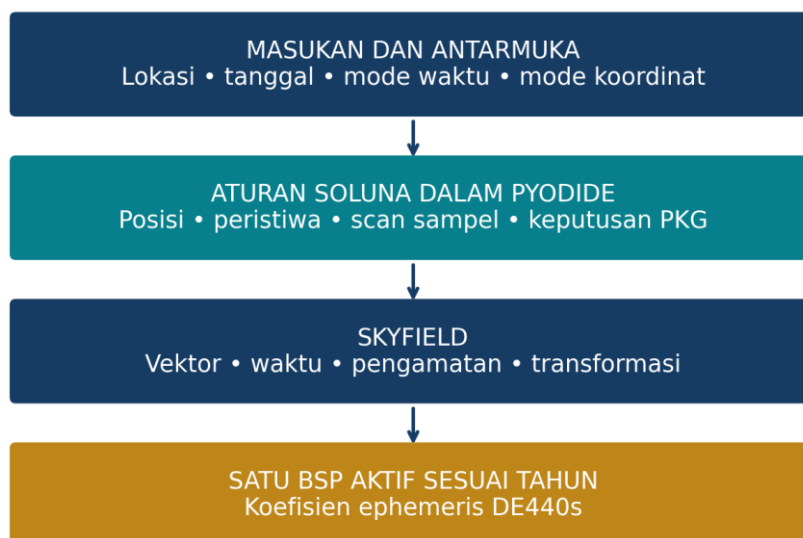
SOLUNA menyatukan perhitungan Matahari dan Bulan dalam antarmuka yang dapat dijalankan melalui peramban. Nama program, judul KHGT Times, dan penggunaan DE440s menunjukkan fokus pada hubungan antara astronomi posisi dan kalender Hijriah global. Nilai pendidikan aplikasi terletak pada keterhubungan keluaran: pengguna dapat melihat posisi benda langit pada suatu saat, peristiwa harian di lokasi tertentu, dan hasil pemindaian global yang memakai aturan kalender berbeda dari sekadar kondisi hilal lokal.

Untuk kegiatan akademik, hubungan ini lebih berharga daripada satu angka tanggal. Ketika tanggal atau lokasi diubah, pengguna dapat mengamati perubahan sudut jam, tinggi, azimut, waktu terbenam, dan elongasi. Pengamatan terhadap keluaran tersebut membuka kesempatan untuk menjelaskan rotasi Bumi, gerak Bulan, serta perbedaan sudut pandang. Program mendukung pemeriksaan numerik atas konsep yang sebelumnya sering disampaikan hanya dengan gambar dua dimensi.

Tujuan operasionalnya ialah menghasilkan informasi yang dapat dibaca dan disimpan. Tujuan metodologisnya ialah membuat sebagian aturan perhitungan dapat dilihat melalui halaman Metodologi dan indikator keputusan PKG. Tujuan pendidikan memerlukan keduanya: tampilan yang mudah digunakan dan penjelasan mengenai definisi besaran. Antarmuka yang ringkas tanpa definisi berisiko mendorong pembacaan angka di luar maknanya; dokumentasi tanpa contoh keluaran menyulitkan pengguna menghubungkan teori dengan proses kerja.

1.2 Empat lapisan yang harus dibedakan

Lapisan pertama adalah data ephemeris DE440s, yang menyediakan representasi posisi benda-benda sistem Tata Surya pada rentang waktu tertentu. Lapisan kedua adalah Skyfield, yang mengolah vektor, waktu, posisi pengamat, koreksi apparent, dan pencarian peristiwa. Lapisan ketiga adalah aturan SOLUNA: cara masukan dibaca, besaran dibentuk, peristiwa dicari, berkas dipilih, dan keluaran disajikan. Lapisan keempat adalah ketentuan kalender KHGT yang diimplementasikan melalui filter PKG. Publikasi JPL mendukung mutu model ephemeris; ia tidak dengan sendirinya mengesahkan logika kalender dalam suatu aplikasi. [R01–R04]



Gambar 1. Lapisan perhitungan SOLUNA. Panah menunjukkan aliran kerja; setiap lapisan memiliki fungsi dan lingkup validasi yang berbeda.

Pemisahan ini membantu melacak penyebab perbedaan. Jika dua aplikasi memakai DE440s tetapi berbeda tinggi hilal, penyebabnya dapat terletak pada kerangka geosentris atau toposentris, koreksi refraksi, definisi waktu sunset, atau koordinat. Jika posisi sama tetapi tanggal awal bulan berbeda, aturan batas UTC, fajar, dan pemilihan saksi perlu diperiksa. Mengganti ephemeris tanpa menganalisis lapisan lain tidak selalu menyelesaikan perbedaan.

1.3 Daring dengan perhitungan di peramban

Berkas utama bernama `index.php`, tetapi versi yang dibahas mengandung antarmuka HTML, gaya CSS, JavaScript, serta mesin Python yang dijalankan melalui Pyodide. Nama ekstensi tidak membuktikan bahwa perhitungan astronomi dikerjakan oleh server PHP. Pada implementasi ini, peramban menyiapkan lingkungan Python, memuat dependensi, memperoleh satu berkas BSP yang sesuai, lalu menghitung dan menampilkan hasil. Pyodide menjalankan Python dalam lingkungan WebAssembly. [R08]

Konsekuensinya, waktu penyiapan dipengaruhi jaringan, cache, ukuran dependensi, dan kemampuan perangkat. Sesudah kernel aktif tersedia, pemindaian banyak kota memakai objek ephemeris yang sama. Perhitungan seratus lokasi tidak berarti seratus unduhan ephemeris. Server tetap berperan menyediakan halaman dan aset; browser pengguna menjalankan beban hitung pada versi ini. Pembaca tidak boleh menganggap hasil benchmark Python pada komputer penguji sebagai waktu yang pasti dialami telepon genggam.

1.4 Ruang lingkup dan batas produk

Keluaran utama mencakup ringkasan, tabel Matahari, tabel Bulan, peristiwa, fenomena, jadwal, gerhana, metodologi, dan istilah. Halaman edukasi Beranda, Skyfield, dan DE440s memberi konteks di ruang kanan. Perhitungan lokal dapat memakai posisi pengamat atau pusat Bumi. Perhitungan kalender global selalu memakai parameter geosentris pada sunset lokal setiap titik yang dipindai. Perbedaan mode tersebut harus diperhatikan sebelum suatu tangkapan layar dibandingkan dengan tabel referensi.

SOLUNA bukan sistem pelaporan cuaca, basis data pengamatan hilal langsung, atau layanan pengesahan kalender. Versi yang diaudit tidak menerapkan semua model visibilitas hilal yang tersedia dalam literatur. Ia juga belum menyediakan pencarian daratan kontinu atau seluruh kontak gerhana lokal. Ruang lingkup yang jelas membantu pengguna memilih pertanyaan yang dapat dijawab oleh aplikasi dan mengenali kapan diperlukan alat lain.

1.5 Cara membaca satu hasil secara bertanggung jawab

Satu hasil astronomi sebaiknya dibaca sebagai rekaman yang terdiri atas masukan, metode, dan keluaran. Masukan meliputi lokasi, elevasi, zona waktu, tanggal, jam atau mode maghrib, serta pilihan koordinat. Metode meliputi ephemeris aktif, definisi horizon, refraksi, dan aturan PKG. Keluaran meliputi posisi dan keputusan beserta alasan. Mengutip hanya tanggal awal bulan tanpa identitas masukan dan versi program mengurangi kemampuan pembaca lain untuk mengulang pemeriksaan.

Dalam pelaporan, kalimat yang layak ialah: “Pada versi SOLUNA yang diuji, dengan kernel tahunan 2027 dan prosedur PKG yang didokumentasikan, awal Syawal 1448 H diperoleh 9 Maret 2027.” Kalimat tersebut menyatakan hasil sekaligus ruang berlakunya. Untuk menyatakan bahwa suatu titik merupakan titik pertama di seluruh bumi, diperlukan prosedur pencarian dan bukti cakupan yang lebih luas daripada katalog kota yang digunakan aplikasi.

BAB 2 GEOMETRI MATAHARI BUMI DAN BULAN

2.1 Gerak yang membentuk gejala langit

Perubahan posisi harian benda langit terutama disebabkan oleh rotasi Bumi. Perubahan musiman posisi Matahari berkaitan dengan revolusi Bumi dan kemiringan sumbu rotasinya. Perubahan posisi Bulan terhadap bintang dan Matahari mengikuti orbit Bulan yang miring terhadap ekliptika dan memiliki jarak yang berubah. Ketiganya hadir bersamaan dalam keluaran SOLUNA, sehingga perubahan tinggi dari satu jam ke jam berikutnya tidak boleh dianggap seluruhnya sebagai gerak orbital Bulan.

Model pengajaran berupa orbit lingkaran berguna untuk mengenalkan arah gerak. Perhitungan program memakai ephemeris numerik yang merepresentasikan dinamika lebih lengkap. Buku ini memakai gambar skematis untuk menjelaskan hubungan geometri, bukan untuk menyatakan bahwa orbit nyata berbentuk lingkaran sempurna atau berada pada satu bidang. Skala jarak dan ukuran pada gambar tidak dibuat sebanding kecuali disebutkan secara khusus.

2.2 Fase sebagai geometri pencahayaan

Matahari menerangi kurang lebih separuh permukaan Bulan, tetapi pengamat di Bumi melihat bagian yang berbeda dari sisi terang itu. Fase baru, kuartal pertama, purnama, dan kuartal terakhir berkaitan dengan susunan arah Matahari dan Bulan dilihat dari Bumi. Bagian terang yang terlihat bukan bayangan Bumi; bayangan Bumi baru terlibat dalam gerhana Bulan. Penjelasan ini penting karena bentuk sabit sering salah dihubungkan dengan tertutupnya Bulan oleh Bumi.

Dua sudut perlu dibedakan. Elongasi E adalah sudut antara arah ke Bulan dan arah ke Matahari dari pengamat yang ditetapkan. Sudut fase i adalah sudut Matahari–Bulan–pengamat, dengan titik sudut pada Bulan. Dalam model bola yang diterangi sumber jauh, fraksi iluminasi k berhubungan dengan i . Nilai i mendekati 180° ketika sisi yang menghadap pengamat hampir gelap dan mendekati 0° ketika hampir penuh. Sudut E dan i tidak identik karena titik sudutnya berbeda.

$$k = (1 + \cos i) / 2; \text{iluminasi (\%)} = 100 k$$

Persamaan tersebut menjelaskan fraksi cakram yang terang secara geometris. Ia tidak menghitung terang langit, ekstingsi atmosfer, atau ambang mata. Karena itu, persentase iluminasi yang positif tidak menjamin hilal dapat dilihat. Dalam program, iluminasi berasal dari perhitungan geometri Skyfield; persamaan di atas berfungsi sebagai penjelasan konsep, bukan pengganti seluruh koreksi yang dipakai mesin.

2.3 Siklus bujur dan nama fase

SOLUNA juga menghitung selisih bujur ekliptika Bulan dan Matahari, yang dinormalisasi menjadi satu putaran. Besaran ini disebut sudut siklus fase dalam uraian buku. Nilainya berbeda dari sudut fase fisik i . Selisih bujur 0° berhubungan dengan fase baru, 90° dengan kuartal pertama, 180° dengan purnama, dan 270° dengan kuartal terakhir. Program membagi siklus tersebut menjadi delapan kategori untuk memberi nama fase dan kecenderungan membesar atau mengecil.

$$\theta = (\lambda_{\text{Bulan}} - \lambda_{\text{Matahari}}) \bmod 360^\circ$$

Nama fase pada suatu saat merupakan kategori yang memiliki rentang sudut. Tulisan “Bulan Baru” dalam kategori delapan fase tidak berarti waktu yang dipilih persis saat ijtimak. Peristiwa fase baru membutuhkan pencarian saat perubahan fungsi mencapai batas yang didefinisikan. Perbedaan antara kategori dan saat peristiwa perlu dijelaskan ketika pengguna membandingkan kartu fase dengan baris waktu konjungsi.

2.4 Ijtimak dan separasi minimum

Konjungsi geosentris yang dipakai untuk kalender mengikuti kesamaan bujur ekliptika geosentris Matahari dan Bulan. Karena Bulan dapat memiliki lintang ekliptika yang tidak nol, elongasi pada saat itu tidak harus nol. Dua arah dapat memiliki bujur yang sama tetapi berada di lintang berbeda. Dengan alasan yang sama, saat kesamaan bujur dan saat separasi sudut minimum tidak wajib identik. Definisi konjungsi perlu disebutkan agar selisih beberapa menit tidak keliru dianggap kesalahan ephemeris. [R03, R11]

SOLUNA menyediakan pencarian yang diberi label konjungsi toposentris. Pada kode yang diaudit, besaran itu dicari sebagai minimum separasi apparent Matahari–Bulan dari lokasi pengamat di sekitar konjungsi geosentris. Ia bukan sekadar kesamaan bujur ekliptika toposentris. Perbandingan ilmiah harus menggunakan definisi operasional tersebut, bukan hanya kesamaan nama “konjungsi”.

2.5 Jarak diameter dan paralaks

Jarak Bulan berubah sepanjang orbit. Perubahan jarak menyebabkan perubahan diameter sudut dan paralaks. Untuk benda berbentuk bola dengan jari-jari R pada jarak d dari pengamat, diameter sudut geometris dapat dituliskan sebagai berikut, dengan satuan R dan d yang sama.

$$\text{diameter sudut} = 2 \arcsin(R / d)$$

Program menggunakan jari-jari Matahari 695.700 km, jari-jari Bulan 1.737,4 km, dan jari-jari ekuatorial Bumi 6.378,137 km pada perhitungan yang relevan. Angka

ini merupakan konstanta model yang dicatat dari kode. Diameter sudut memakai jarak pengamat sesuai mode, sedangkan paralaks horizontal memakai jarak geosentris. Pemisahan tersebut menjelaskan mengapa dua kolom jarak dapat sedikit berbeda pada tabel Bulan.

Paralaks horizontal adalah perubahan arah maksimum yang berkaitan dengan panjang dasar seukuran jari-jari Bumi untuk konfigurasi horizon ideal. Besarnya lebih nyata untuk Bulan daripada Matahari karena Bulan jauh lebih dekat. Nilai itu membantu memahami mengapa tinggi hilal toposentris dekat horizon dapat berbeda cukup berarti dari tinggi geosentris. Rumus pendekatan yang digunakan untuk menjelaskan model adalah $\pi = \arcsin(R_E / d_{\text{geosentris}})$.

2.6 Umur Bulan dan lag

Umur Bulan pada hasil program adalah selisih waktu terhadap konjungsi yang dipilih oleh prosedur peristiwa. Umur merupakan besaran waktu, sedangkan tinggi dan elongasi merupakan besaran sudut. Hubungan di antara ketiganya bergantung pada geometri dan lokasi. Dua tempat pada umur yang sama dapat memiliki tinggi hilal berbeda. Dua lunasi dengan umur serupa juga dapat memiliki elongasi dan kedudukan hilal yang berbeda.

Lag merupakan selisih waktu moonset terhadap sunset menurut pasangan peristiwa yang dipilih. Dalam kode ini, moonset yang paling dekat dengan sunset dicari dalam jendela sekitar ± 12 jam. Lag dapat bernilai negatif apabila peristiwa Bulan yang dipilih terjadi lebih dahulu. Nilai tersebut tidak boleh selalu diterjemahkan sebagai “berapa menit Bulan akan terbenam setelah Matahari” tanpa memeriksa tanda dan tanggal peristiwanya.

2.7 Implikasi untuk pembacaan hilal

Tinggi, elongasi, iluminasi, umur, dan lag menyampaikan aspek yang berbeda. Tinggi menyatakan kedudukan terhadap horizon model. Elongasi menyatakan separasi arah. Iluminasi menyatakan fraksi geometris sisi terang. Umur menyatakan jarak waktu dari konjungsi. Lag menyatakan hubungan dua peristiwa horizon. Menambah satu besaran tidak menghilangkan kebutuhan untuk mendefinisikan yang lain. Dalam penelitian visibilitas, geometri harus dihubungkan dengan kondisi pengamatan dan indikator keberhasilan rukyat yang dicatat secara independen. [R12, R13]

BAB 3 KOORDINAT SKALA WAKTU DAN HORIZON

3.1 Lokasi pengamat dan tanda koordinat

Lintang geografis positif berada di utara ekuator dan negatif di selatan. Bujur positif berada di timur Greenwich dan negatif di barat. Elevasi dimasukkan dalam meter. Koordinat aplikasi berupa derajat desimal; untuk mengubah derajat, menit, dan detik, jumlahkan $D + M/60 + S/3600$, lalu terapkan tanda sesuai hemisfer. Tanda negatif harus diberikan kepada keseluruhan besaran, bukan hanya komponen derajat secara terpisah.

Identitas kota merupakan label praktis untuk satu titik koordinat, bukan representasi seluruh wilayah administratif. Kota yang sama dapat diwakili pusat kota, bandar udara, observatorium, atau titik pantai yang berbeda. Lampiran katalog mencantumkan koordinat persis yang digunakan program agar pembaca dapat membedakan kesamaan nama dan kesamaan titik. Selisih titik dapat mengubah saat sunset serta parameter hilal, terutama ketika nilai mendekati ambang.

Skyfield membentuk pengamat toposentris memakai model WGS84. Model lokasi tersebut memperhitungkan bentuk Bumi dan kedudukan pengamat dalam pembentukan vektor. Akan tetapi, adanya elevasi pada masukan tidak berarti aplikasi telah memodelkan seluruh depresi horizon laut, relief pegunungan, dan obstruksi setempat. Definisi peristiwa horizon dan geometri posisi pengamat adalah dua komponen yang harus diperiksa sendiri-sendiri. [R05]

3.2 Ekuatorial ekliptika dan horizontal

Koordinat ekuatorial memakai asensio rekta α dan deklinasi δ . Asensio rekta lazim dinyatakan dalam jam, sehingga satu jam setara 15° . Deklinasi memakai derajat utara atau selatan ekuator langit. Koordinat ekliptika memakai bujur dan lintang terhadap bidang ekliptika. Koordinat horizontal memakai tinggi h dan azimut A terhadap horizon suatu lokasi. Ketiga pasangan tersebut menggambarkan arah yang sama melalui bidang acuan berbeda.

Program memakai RA/Dec apparent pada epoch waktu pengamatan, lalu menghubungkannya dengan waktu sidereal apparent. Bujur dan lintang ekliptika pada tabel dihitung dari geosenter, termasuk ketika mode utama toposentris dipilih. Dengan demikian, tabel tertentu dapat memuat besaran dengan konteks asal berbeda. Kamus data dalam lampiran membantu pembaca mengenali kolom yang mengikuti mode pengamat dan kolom yang selalu geosentris.

Azimut dinormalisasi menjadi 0° sampai kurang dari 360° , dengan utara 0° , timur 90° , selatan 180° , dan barat 270° . Tinggi positif berada di atas horizon geometris

model; tinggi negatif berada di bawahnya. Sudut jam berubah dengan rotasi Bumi. Untuk membandingkan tabel, satuan jam pada sudut jam atau RA harus dikonversi jika referensi menyajikannya dalam derajat.

3.3 Dari RA dan deklinasi ke tinggi

Untuk proyeksi geosentris ke horizon lokasi, program menghitung waktu sidereal lokal dari Greenwich Apparent Sidereal Time atau GAST dan bujur geografis. Sudut jam dibentuk dari selisih waktu sidereal lokal dan RA, lalu dinormalisasi. Persamaan berikut menggunakan λ_{timur} dalam derajat dan GAST serta α dalam jam.

$$\text{LST} = (\text{GAST} + \lambda_{\text{timur}} / 15) \bmod 24 \text{ jam}$$

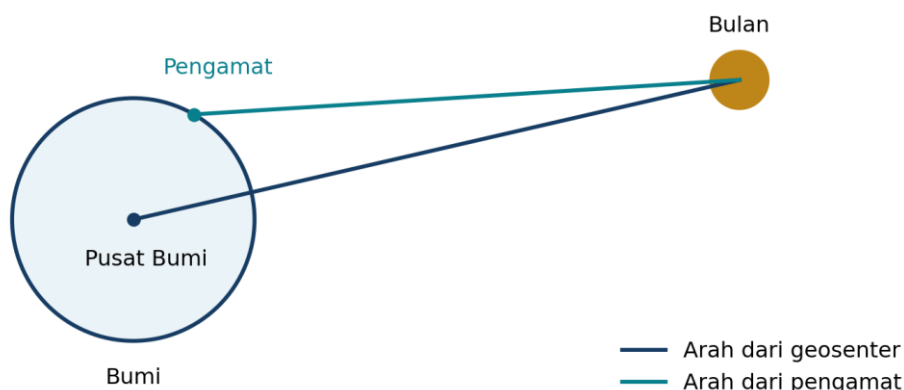
$$\text{Hjam} = ((\text{LST} - \alpha + 12) \bmod 24) - 12$$

$$\sin h = \sin \varphi \sin \delta + \cos \varphi \cos \delta \cos H$$

Dalam persamaan tinggi, H harus dipakai sebagai sudut dengan konversi 15° per jam. Fungsi trigonometri di perangkat lunak memerlukan radian, sedangkan tampilan mengembalikan derajat. Kesalahan konversi satuan dapat menimbulkan hasil yang tampak teratur tetapi salah besar. Menggunakan RA J2000 bersama GAST tanggal pengamatan juga mencampurkan acuan dan perlu dihindari. Definisi koordinat dan transformasi dijelaskan lebih lanjut dalam sumber astronomi standar. [R05, R18] Buku Montenbruck dan Pfleger dapat dipakai sebagai pendalaman astronomi komputasional setelah pembaca memahami satuan dan kerangka dasar. [R22]

3.4 Geosentris dan toposentris

Geosentris berarti arah dihitung dari pusat Bumi. Toposentris berarti arah dihitung dari titik pengamat di permukaan atau pada elevasi tertentu. Pada mode geosentris SOLUNA, RA/Dec dari pusat Bumi diproyeksikan ke horizon lokasi. Istilah tinggi geosentris karena itu tetap memerlukan lokasi untuk menentukan horizon, tetapi arah benda langitnya belum diberi paralaks lokal. Pada mode toposentris, posisi pengamat masuk ke pembentukan vektor sebelum arah horizontal dihitung.



Gambar 2. Perbedaan arah geosentris dan toposentris secara skematis. Sudut diperbesar untuk kejelasan dan tidak menunjukkan skala jarak sebenarnya.

Perbandingan yang benar mengunci lokasi, saat, serta definisi refraksi, lalu mengubah hanya kerangka pengamat. Jika parameter lain ikut berubah, pengaruh paralaks tidak dapat dipisahkan. Untuk KHGT, kode mempertahankan tinggi dan elongasi global dalam kerangka geosentris walaupun pengguna mengganti mode tabel lokal menjadi toposentris. Ini adalah pilihan aturan kalender dalam program, bukan bukti bahwa salah satu kerangka lebih benar untuk semua tujuan.

3.5 Apparent dan refraksi atmosfer

Posisi apparent dalam Skyfield memasukkan koreksi astronomis terhadap arah, seperti waktu rambat cahaya, aberasi, dan pembelokan gravitasi yang relevan. Refraksi atmosfer adalah koreksi yang berbeda. Karena itu, kata apparent pada RA/Dec tidak berarti tinggi tersebut sudah otomatis dibias oleh atmosfer. SOLUNA menampilkan tinggi geometris dan, pada kolom tertentu, tinggi dengan model refraksi standar. [R04]

Dekat horizon, atmosfer dapat menggeser kedudukan tampak cukup besar dan bergantung pada suhu, tekanan, serta struktur lapisan udara. Model standar berguna untuk almanak, tetapi tidak menggambarkan setiap keadaan lapangan. Tinggi hilal yang digunakan oleh filter global pada versi ini adalah tinggi geosentris geometris. Memasukkan tinggi yang telah direfraksi ke ambang yang dimaksudkan untuk tinggi geometris akan mengubah dasar kriteria.

3.6 UTC UT1 TT dan TDB

UTC dipakai untuk pelabelan waktu sipil dan perbandingan batas kalender global. UT1 berkaitan dengan rotasi Bumi dan diperlukan untuk orientasi Bumi terhadap langit. TT adalah skala waktu dinamis yang seragam dalam kerangka geosentris, sedangkan TDB digunakan dalam hubungan dengan ephemeris barysentris. Program menyerahkan banyak konversi internal tersebut kepada objek waktu

Skyfield. Pengguna memasukkan waktu sipil, tetapi mesin tidak menganggap semua skala waktu astronomis identik. [R06, R17]

Selisih TT dan UT1 dikenal sebagai ΔT . Ketidakpastian rotasi Bumi, terutama untuk masa jauh dari data pengamatan modern, dapat memengaruhi konversi arah langit menjadi lokasi atau waktu peristiwa di Bumi. Ephemeris yang memiliki rentang panjang tidak otomatis memberikan ketepatan waktu sipil yang sama di semua abad. Hal ini sangat penting untuk lintasan gerhana historis. [R15]

Program memuat data dan model waktu yang tersedia pada lingkungan Skyfield yang digunakan. Pengujian reproduksibilitas harus mencatat versi lingkungan tersebut, karena masukan tanggal yang sama dapat diproses dengan informasi waktu yang diperbarui. Dalam buku ini, identitas pengujian sebelumnya dicatat secara eksplisit; kesesuaian dua belas bulan bukan pengujian tersendiri terhadap setiap model ΔT atau UT1.

3.7 Zona IANA dan tanggal yang melintasi batas hari

Asia/Jakarta adalah identitas zona waktu, bukan sekadar tulisan UTC+7. Beberapa zona memiliki aturan perubahan jam musiman atau sejarah offset yang berbeda. Pacific/Auckland dapat menggunakan waktu musim panas pada bagian tahun tertentu. Penggunaan nama zona IANA memungkinkan lingkungan waktu menerapkan aturan tanggal yang bersangkutan, selama basis datanya tersedia dan benar. [R09]

Ketika sunset di Amerika terjadi pada 8 Maret waktu setempat, UTC-nya dapat sudah 9 Maret. Karena itu, perbandingan PKG harus memakai timestamp lengkap dengan tanggal dan zona, bukan hanya membandingkan angka jam. Urutan 23.50 dan 01.05 dapat berarti peristiwa kedua lebih kemudian, jika kedua tanggal berbeda. Menghilangkan tanggal adalah salah satu sumber kesalahan yang paling mudah terjadi dalam audit kalender global.

3.8 Jam perangkat dan waktu simulasi

Jam pada badge kanan atas memakai tanggal dan waktu sistem perangkat pengguna. Ia memberi konteks waktu saat aplikasi dibaca. Jam itu berbeda dari tanggal dan jam masukan astronomi. Memilih tahun 2027 tidak mengubah jam perangkat menjadi 2027. Sebaliknya, tanggal sistem perangkat yang keliru dapat menyebabkan penyiapan awal memilih kernel tahun yang keliru, walaupun perhitungan sesudah pengguna memperbaiki pilihan tahun dapat memakai berkas yang benar.

Untuk penelitian, catat waktu simulasi lokal beserta UTC, identitas data, dan mode. Jam perangkat pada badge dapat dicatat sebagai waktu dokumentasi tangkapan layar.

BAB 4 SKYFIELD DAN PROSES KOMPUTASI

4.1 Kedudukan Skyfield

Skyfield adalah pustaka Python untuk menghitung posisi astronomis dengan dukungan ephemeris JPL dan transformasi waktu serta koordinat. Perangkat ini memiliki dokumentasi teknis terbuka dan tercatat dalam Astrophysics Source Code Library. Ketersediaan dokumentasi dan kode mendukung pemeriksaan metode, tetapi mutu aplikasi yang memakai Skyfield tetap bergantung pada bagaimana pustaka itu dipanggil. [R02–R07]

SOLUNA memilih Skyfield karena perhitungan Matahari dan Bulan, lokasi WGS84, fase, peristiwa, dan pembacaan SPK dapat dikelola dalam satu lingkungan Python. Pendekatan ini mengurangi kebutuhan menulis ulang semua transformasi astronomi dari awal. Keuntungan tersebut tidak menghapus tanggung jawab pengembang dalam menentukan definisi hilal, jendela pencarian, dan kebijakan penanganan kegagalan.

4.2 Rantai vektor dan pengamatan

Ephemeris SPK menyimpan hubungan antara target dan pusat tertentu. Arah Bulan dari Bumi diperoleh melalui hubungan vektor yang sesuai, bukan membaca satu angka tinggi langsung dari berkas. Skyfield membentuk posisi benda pada waktu tertentu, menghitung pengamatan dari pengamat, dan menghasilkan arah apparent. Pada mode lokal, titik pengamat ditambahkan ke posisi Bumi. Pada mode geosentris, pengamatnya pusat Bumi.

Urutan yang tampak dalam kode adalah pembentukan waktu, pemilihan pengamat, pemanggilan observe, penerapan apparent, lalu konversi ke RA/Dec atau altaz. Besaran turunan seperti elongasi memakai separasi dua arah pada kerangka yang sama. Konstanta jari-jari dan jarak digunakan untuk diameter dan paralaks. Kategori fase, perkiraan magnitudo, dan filter 5° – 8° merupakan lapisan tambahan yang dibentuk SOLUNA.

4.3 Pencarian peristiwa diskret

Peristiwa seperti terbit dan terbenam dapat dipandang sebagai perubahan status dari di bawah horizon menjadi di atas, atau sebaliknya. Pencarian diskret memeriksa fungsi status dalam suatu jendela waktu dan memperhalus saat terjadinya perubahan. Konsep ini berbeda dari menghitung posisi pada satu timestamp. Fase utama juga dapat ditemukan melalui perubahan kategori fase yang mempunyai batas sudut tertentu. [R07]

Pemilihan jendela menentukan peristiwa yang mungkin ditemukan. Jika jendela tidak mencakup peristiwa, tabel dapat kosong meskipun peristiwa terjadi di luar batas itu. Langkah awal yang terlalu besar berisiko melewatkan perubahan singkat pada konfigurasi tertentu. Dokumentasi algoritma perlu mencatat fungsi pencarian, jendela, langkah, dan penanganan keadaan tanpa peristiwa. Presisi digit waktu pada tampilan tidak menjelaskan seluruh kualitas pencarian.

4.4 Pencarian minimum dan maksimum

SOLUNA menggunakan pencarian sampel untuk beberapa fenomena. Konjungsi toposentris dicari melalui separasi minimum di sekitar konjungsi geosentris: sampel awal sekitar lima menit di dalam jendela ± 12 jam, lalu penyempitan sekitar hasil terbaik dengan langkah lima detik. Perigee dan apogee Bulan menggunakan sampel jarak dengan penyempitan berikutnya. Ringkasan gerhana Matahari juga memperkirakan saat sumbu bayangan paling dekat ke Bumi melalui grid waktu.

Pencarian seperti ini menghasilkan minimum di antara sampel yang diperiksa. Minimum itu bisa sangat berguna untuk pendidikan dan penyaringan awal, tetapi bukan bukti bahwa fungsi kontinu telah dioptimalkan sampai ketelitian satu detik. Bila hasil ditampilkan hingga detik, pembaca perlu tetap memperhatikan ukuran langkah dan bentuk fungsi. Cara meningkatkan ketelitian adalah menambahkan pencarian akar atau optimasi terkontrol serta perbandingan dengan referensi independen, bukan sekadar menambah angka desimal.

4.5 Perkiraan magnitudo

Nilai magnitudo pada tabel merupakan hasil model perkiraan yang ditanam dalam kode. Untuk Matahari, variasi jarak dimasukkan melalui koreksi logaritmik terhadap nilai acuan. Untuk Bulan, program memakai pendekatan yang bergantung pada sudut fase dan jarak. Rumus tersebut tidak memasukkan seluruh karakteristik permukaan, hamburan, dan atmosfer pada pengamatan nyata.

$$m_{\text{Matahari}} = -26,86 + 5 \log_{10}(d_{\text{AU}})$$

$$m_{\text{Bulan}} \approx -12,73 + 0,026 i + 4 \times 10^{-9} i^4 + 5 \log_{10}(d / 384.400 \text{ km})$$

Pada rumus Bulan dalam kode, i dinyatakan dalam derajat dan d adalah jarak yang dipakai untuk koreksi. Rumus ini dicatat untuk transparansi implementasi. Buku tidak menjadikannya model visibilitas hilal atau ukuran fotometri hasil pengamatan. Apabila magnitudo menjadi objek penelitian utama, dasar kalibrasi empiris rumus dan kesesuaiannya pada sabit sangat tipis perlu diverifikasi lebih lanjut.

4.6 Kegagalan peristiwa dan data kosong

Sebagian fungsi pencarian menangkap exception dan mengembalikan nilai kosong atau melewati hasil. Perilaku ini menjaga antarmuka tetap berjalan, tetapi keluaran kosong dapat memiliki lebih dari satu penyebab: peristiwa tidak terjadi dalam jendela, objek berada di luar cakupan ephemeris, kondisi lokal tidak mempunyai crossing, atau proses pencarian gagal. Pengguna akademik tidak boleh langsung mengubah semua sel kosong menjadi kesimpulan “tidak ada fenomena”.

Untuk audit, hasil yang meragukan perlu ditelusuri dengan masukan identik dan pencarian independen. Idealnya, pengembangan berikutnya membedakan status tidak terjadi, tidak dicari, di luar cakupan, dan gagal hitung. Ini merupakan rekomendasi dokumentasi dan pengembangan; buku tidak menyatakan bahwa pemisahan status tersebut telah tersedia lengkap pada versi yang diaudit.

4.7 Reproduksiabilitas lingkungan

Versi pengujian yang dipakai dalam studi kasus buku menggunakan Python 3.12.14, Skyfield 1.55, jplephem 2.24, NumPy 2.5.3, serta basis data zona waktu 2025b. Lingkungan uji native tersebut digunakan untuk menjalankan mesin yang diekstrak dari sumber. Lingkungan browser memakai Pyodide dan dapat memiliki susunan paket yang berbeda. Persamaan keluaran kalender yang diuji tidak boleh diperluas menjadi jaminan bahwa seluruh aspek kinerja kedua lingkungan selalu identik.

Reproduksiabilitas memerlukan identitas kode, identitas kernel, versi paket, masukan, dan aturan pembandingan. Bila salah satu berubah, hasil baru sebaiknya dicatat sebagai uji versi lain. Kebiasaan tersebut memudahkan peneliti menilai apakah selisih timbul karena data ephemeris, basis waktu, perubahan kode, atau konfigurasi pengguna.

BAB 5 DE440S DAN PENGELOLAAN BERKAS BSP

5.1 Ephemeris Development Ephemeris

DE adalah keluarga Development Ephemeris yang dikembangkan JPL. Ephemeris modern menggabungkan model dinamika sistem Tata Surya dengan pengamatan untuk membentuk solusi numerik posisi dan gerak. Publikasi Park dan rekan-rekan menjelaskan DE440 dan DE441, termasuk pembaruan data dan pemodelan yang melandasinya. DE440 ditujukan pada rentang 1550–2650, sedangkan DE441 menyediakan rentang jauh lebih panjang dengan pilihan model yang mendukung integrasi jangka panjang. Keduanya tidak dapat diperlakukan sebagai berkas yang identik hanya karena berasal dari keluarga JPL. [R01]

DE440s adalah distribusi pendek dari solusi DE440. Dokumentasi Skyfield mencantumkan cakupannya sekitar 1849–2150 dan ukuran sekitar 32 MB. Huruf s merujuk pada versi pendek, bukan pada penggantian perhitungan dengan rumus fase sederhana. Cakupan tersebut merupakan informasi produk asal. SOLUNA yang dibahas menyediakan pilihan berkas nominal 1850–2149 serta lima berkas tahunan 2026–2030; cakupan aplikasi harus dinilai dari berkas yang benar-benar tersedia. [R03]

Tidak tepat menjanjikan bahwa DE440s memadai untuk semua tahun sebelum dan sesudah tahun acuan. Tahun di luar cakupan fisik kernel membutuhkan data lain dan pengujian. Di dalam cakupan pun, kualitas waktu rotasi Bumi dan definisi horizon tetap dapat membatasi hasil terapan. Ketelitian ephemeris merupakan satu bagian dari keseluruhan ketelitian aplikasi.

5.2 Struktur SPK dan representasi Chebyshev

BSP yang dipakai merupakan kernel SPK dalam struktur Double Precision Array File atau DAF. Segmen memuat hubungan target dan pusat, kerangka acuan, rentang waktu, dan data representasi numerik. Pada segmen SPK tipe 2, posisi direpresentasikan melalui koefisien polinomial Chebyshev dalam interval tertentu; kecepatan diperoleh dari turunan representasi itu. Pembaca berkas memerlukan metadata interval dan normalisasi waktu, bukan hanya koefisien mentah. [R10]

$$\tau = (t - t_m) / r; \text{posisi}(t) = \sum c_k T_k(\tau)$$

Pada persamaan konseptual tersebut, t_m adalah titik tengah interval, r adalah setengah lebar interval, T_k adalah polinomial Chebyshev orde k , dan c_k adalah koefisien untuk satu komponen posisi. Notasi ini menjelaskan prinsip pembacaan, bukan seluruh spesifikasi biner SPK. Kernel tidak berisi tabel tinggi hilal setiap kota. Tinggi dan elongasi dihasilkan sesudah posisi benda dikombinasikan dengan waktu dan kerangka pengamat.

Memotong ephemeris secara benar berarti mempertahankan rekaman dan hubungan segmen yang diperlukan pada jendela waktu baru. Sekadar membagi byte berkas menjadi potongan berukuran sama akan merusak struktur. Pengujian berkas turunan perlu memeriksa format, cakupan tiap segmen, target, dan kesesuaian nilai terhadap sumber pada tanggal yang sama. Dokumentasi NAIF dan utilitas jplephem menyediakan dasar untuk memahami proses ini. [R03, R10]

5.3 Berkas yang dipilih SOLUNA

Versi aplikasi mendahulukan satu berkas tahunan untuk 2026, 2027, 2028, 2029, atau 2030. Tahun lain dalam rentang konfigurasi memakai satu berkas seratus tahunan yang mencakup tahun pilihan. Pada penyiapan awal, tahun diambil dari tanggal perangkat. Ketika pengguna mengubah tahun masukan, pemilih mencari berkas dengan rentang nominal paling sempit yang sesuai.

Tabel 1. Berkas ephemeris dan rentang pemilihan nominal

Nama berkas	Tahun nominal	Byte	MiB
de440s_2026.bsp	2026	677.888	0,646
de440s_2027.bsp	2027	679.936	0,648
de440s_2028.bsp	2028	678.912	0,647
de440s_2029.bsp	2029	677.888	0,646
de440s_2030.bsp	2030	677.888	0,646
de440s_1850_1949.bsp	1850–1949	11.563.008	11,027
de440s_1950_2049.bsp	1950–2049	11.669.504	11,129
de440s_2050_2149.bsp	2050–2149	11.022.336	10,512

Ukuran berkas tahunan sekitar 0,65 MiB membantu mengurangi transfer dibandingkan berkas puluhan megabyte. Akan tetapi, pengurangan ukuran transfer tidak menjamin percepatan dengan faktor yang sama pada seluruh proses hitung. Waktu penyiapan paket Python, decoding berkas, pembangunan objek, komputasi posisi, dan render tabel memiliki komponen biaya sendiri. Klaim kinerja sebaiknya memisahkan download, setup, dan scan.

5.4 Data penyangga pada berkas tahunan

Nama tahunan menunjukkan prioritas pemilihan menurut tahun masukan, bukan janji bahwa berkas hanya memuat tepat 1 Januari sampai 31 Desember. Berkas turunan memakai data penyangga karena fungsi aplikasi mencari sejumlah fenomena sebelum atau sesudah waktu yang dipilih, termasuk pencarian gerhana

Bulan hingga sekitar lima tahun. Jika kernel hanya memiliki satu tahun kalender tanpa penyangga, fungsi lintas batas dapat gagal pada akhir tahun.

Perbedaan antara rentang nominal dan rentang fisik harus dicatat dalam manifest pemeliharaan. Rentang nominal mengatur pemilihan aplikasi. Rentang fisik berasal dari interval yang didukung segmen SPK. Untuk menjamin seluruh fenomena, rentang fisik segmen yang paling membatasi harus tetap mencakup semua timestamp pencarian. Ketika data asal berakhir sekitar 2150, penyangga tidak dapat menyediakan tahun yang memang tidak tersedia dalam sumber.

5.5 Fallback yang terbatas

Fallback ke berkas rentang hanya dijalankan ketika berkas tahunan tidak ditemukan, yaitu HTTP 404 atau 410 pada kode yang diaudit. Kesalahan server HTTP 500, data berukuran salah, atau format yang rusak tidak secara otomatis dianggap sebagai alasan membaca seluruh berkas cadangan. Perilaku ini mencegah aplikasi menyembunyikan kerusakan distribusi dengan unduhan besar yang tidak terkontrol.

Berkas asli dengan nama de440s.bsp dan variasi de440s.bsp dilarang diakses oleh pemilih JavaScript dan mesin Python pada versi ini. Semua kota dalam satu proses scan memakai kernel aktif yang sama. Ketika tahun berubah, aplikasi menyiapkan kernel yang sesuai dan menutup atau mengganti kernel sebelumnya, alih-alih membuka semua tahunan dan semua rentang sekaligus.

5.6 Pemeriksaan format dan integritas

Program memeriksa panjang byte terhadap konfigurasi dan pengenalan DAF/SPK di awal berkas. Pemeriksaan ini dapat menemukan berkas yang terpotong atau tanggapan HTML yang salah dikirim sebagai BSP. Namun, pemeriksaan ukuran dan magic bytes belum merupakan pemeriksaan integritas kriptografis. Dua berkas berbeda dapat memiliki ukuran dan pengenalan yang sama.

Dalam audit buku, hash SHA-256 digunakan untuk mengidentifikasi berkas yang diuji. Hash tersebut merupakan bagian dari prosedur pengujian, bukan fitur verifikasi hash yang otomatis dilakukan aplikasi pada setiap unduhan. Pemelihara distribusi dapat menambahkan manifest hash pada proses rilis tanpa mengubah cara pengguna membaca hasil. Jika perubahan itu diterapkan kelak, dokumentasi versi harus menyatakan secara jelas bahwa perilaku integritas telah berubah.

5.7 Cache antrean dan pemeliharaan

Permintaan berkas menggunakan kebijakan cache peramban yang membantu pemakaian ulang data. Antrean pemilihan ephemeris dan penanda revisi pilihan mencegah hasil penyiapan lama mengambil alih masukan tahun yang lebih baru.

Ini relevan ketika pengguna mengganti tahun beberapa kali saat proses masih berjalan. Versi aktif harus sesuai pilihan terakhir sebelum hasil ditafsirkan.

Pemeliharaan yang tertib menempatkan semua nama berkas sesuai konfigurasi, memeriksa tanggapan server, dan menghindari penggantian isi dengan nama identik tanpa strategi pembaruan cache. Satu BSP yang sehat dan sesuai tahun lebih bermanfaat daripada banyak BSP yang diunduh tanpa kebutuhan. Pengujian performa perlu memakai kondisi cache dingin dan cache hangat secara terpisah, dengan spesifikasi perangkat dan jaringan yang dicatat.

BAB 6 PANDUAN ANTARMUKA DAN PENGGUNAAN

6.1 Susunan ruang kerja

Sidebar menyediakan navigasi edukasi di atas bagian Lokasi Pengamat. Beranda menjelaskan SOLUNA dan metode umum; Skyfield menjelaskan mesin perhitungan; DE440s menjelaskan sumber ephemeris. Ruang kanan kemudian dipakai untuk informasi atau hasil hitung. Pergantian halaman edukasi merupakan navigasi konten dan tidak harus menyiapkan kernel baru. Pengguna dapat kembali ke hasil melalui kontrol yang tersedia dalam antarmuka.

Di bagian pengamatan, pengguna memilih kota atau memasukkan koordinat, elevasi, zona, tanggal, dan mode waktu. Ruang hasil menampilkan indikator utama serta tab yang membagi informasi secara tematis. Pada perangkat sempit, sidebar dapat dilipat. Isi tabel panjang tetap memerlukan scroll pada ruang kanan; dokumentasi tampilan tidak boleh menganggap semua informasi selalu muat dalam satu layar.

6.2 Penyiapan pertama

Pada kunjungan pertama, browser perlu memuat antarmuka, Pyodide, dependensi, dan satu BSP sesuai tahun berjalan. Status penyiapan harus selesai sebelum hasil dianggap tersedia. Koneksi lambat dapat memperpanjang tahap ini walaupun kernel tahunan berukuran kecil. Jika perangkat menunjukkan tahun yang tidak sesuai, pengguna sebaiknya memeriksa tanggal sistem sebelum menilai pilihan kernel awal.

Sesudah siap, gunakan tanggal masukan yang benar. Tombol Gunakan Tanggal & Jam Sekarang mengambil waktu perangkat sebagai masukan, sedangkan tanggal manual memungkinkan perhitungan masa lalu atau mendatang dalam cakupan data. Tanggal sekarang pada badge tetap mengikuti perangkat walaupun tanggal simulasi berbeda. Pemisahan ini membantu pengguna melihat konteks sesi tanpa kehilangan kemampuan melakukan simulasi.

6.3 Memilih dan memeriksa lokasi

Pencarian kota membantu mengisi koordinat dan zona. Setelah memilih, baca kembali lintang, bujur, elevasi, dan nama zona. Kota dengan nama mirip atau label wilayah berbeda tidak selalu memakai titik yang sama. Untuk lokasi observatorium, sebaiknya masukkan koordinat titik pengamatan sendiri dan simpan catatan sumber koordinat tersebut. Elevasi perlu dijelaskan apakah berasal dari peta, GNSS, atau data lokal jika digunakan dalam laporan.

Zona waktu harus sesuai lokasi dan tanggal yang diteliti. Mengganti bujur tanpa mengganti zona dapat menghasilkan waktu lokal yang tidak mewakili jam sipil tempat itu. Arah dan posisi pada timestamp UTC mungkin tetap dapat dihitung, tetapi pembacaan tanggal lokal menjadi tidak tepat. Karena itu, koordinat dan zona harus diperlakukan sebagai pasangan masukan yang diperiksa bersama.

6.4 Jam manual dan maghrib lokal

Mode Jam Manual menghitung posisi pada jam masukan. Ia cocok untuk menelaah perubahan tinggi selama pengamatan atau mengulang timestamp dari sumber lain. Mode Maghrib Lokal mencari waktu terbenam Matahari pada tanggal dan lokasi yang dipilih, kemudian menghitung posisi pada saat tersebut. Nilai jam input bukan penentu akhir ketika mode maghrib digunakan.

Untuk membandingkan dua aplikasi pada maghrib, pastikan definisi sunset setara. Perbedaan standar pusat Matahari, refraksi, semidiameter, dan horizon dapat mengubah waktu evaluasi. Jika tujuan hanya membandingkan posisi ephemeris, gunakan UTC yang sama pada mode manual. Jika tujuan membandingkan prosedur kalender, biarkan masing-masing menghitung sunset, tetapi catat selisih timestamp dan definisinya sebagai bagian analisis.

6.5 Memilih mode koordinat

Mode Toposentris berguna untuk menjelaskan arah dari pengamat, khususnya efek paralaks Bulan. Mode Geosentris dipakai untuk parameter tertentu pada kajian kalender. Sesudah mengganti mode, jalankan Hitung Astronomi dan pastikan penanda mode hasil sesuai. Perubahan mode tidak boleh disimpulkan hanya dari warna tombol atau masukan yang belum dihitung ulang.

Pada hasil global, perubahan mode lokal tidak mengubah aturan bahwa tinggi dan elongasi global selalu geosentris. Ini memungkinkan pembelajaran dua kerangka sambil menjaga parameter kalender tetap konsisten. Jika tanggal global berubah hanya karena pilihan koordinat lokal pada masukan identik, keadaan itu perlu ditelusuri sebagai perubahan perilaku atau cacat, karena tidak sesuai rancangan versi yang dijelaskan buku.

6.6 Membaca ringkasan dan tab

Tabel 2. Tab hasil dan fungsi utamanya

Tab	Isi yang dibaca	Catatan penggunaan
Ringkasan	Parameter utama Matahari dan Bulan	Gambaran cepat; periksa satuan pada tabel rinci.

Tab	Isi yang dibaca	Catatan penggunaan
Matahari	Arah, jarak, ukuran, dan besaran terkait Matahari	Parameter memiliki kerangka yang perlu dicatat.
Bulan	Arah, fase, iluminasi, orientasi, dan jarak	Nama fase kategori berbeda dari timestamp fase.
Peristiwa	Rise, set, transit, twilight, konjungsi, umur, lag	Baca tanggal dan zona peristiwa.
Fenomena	Fase, node, jarak ekstrem, musim, dan kandidat gerhana	Jendela pencarian berbeda menurut fenomena.
Jadwal	Shubuh, Dhuha, Dhuhur, Ashar, Maghrib, Isya, malam	Parameter aplikasi; tanpa ihtiyat otomatis.
Gerhana	Ringkasan tahun/bulan terpilih	Status lokal bukan semua kontak gerhana.
Metodologi	Definisi perhitungan dan batas model	Cocokkan dengan versi sumber.
Istilah	Arti parameter dan istilah antarmuka	Pendamping interpretasi, bukan pengujian.

Ringkasan menyediakan gambaran cepat, tetapi keputusan akademik biasanya memerlukan tabel rinci. Kolom Matahari dan Bulan harus dibaca bersama satuan serta catatan asalnya. Pada tab Peristiwa, tanggal peristiwa yang berbeda dari tanggal masukan perlu diperhatikan. Pada Fenomena, hasil dapat memuat pencarian masa depan dengan jendela berbeda. Pada Jadwal, parameter sudut dan model bayangan merupakan definisi operasional yang harus dicatat.

Halaman Metodologi dan Istilah adalah bagian dari dokumentasi produk. Jika teks metodologi berubah pada rilis berikutnya tanpa perubahan kode, pembaca harus mengaudit konsistensinya. Penjelasan yang tampak rapi belum menjamin bahwa implementasi memakai definisi yang ditulis. Buku ini menempatkan pembacaan kode sebagai dasar uraian fitur untuk versi yang dibahas.

6.7 Hasil global dan label kalender

Kotak global mencantumkan konjungsi, hari H yang diterima, jumlah lokasi, saksi, tinggi, elongasi, serta audit PKG. Label awal bulan diperluas menjadi bentuk seperti “Tanggal 1 Jumadilawal 1448 H = 12-10-2026”. Nama bulan dan tahun menambah konteks yang sebelumnya tidak terlihat pada tanggal Gregorian saja. Label tersebut harus dibaca bersama alasan keputusan, bukan sebagai pengganti audit astronomi.

Dalam program, penamaan bulan Hijriah memakai titik acuan 1 Jumadilawal 1448 H pada 12 Oktober 2026 dan perkiraan panjang lunasi untuk menelusuri urutan bulan. Tanggal Gregorian hasil keputusan tetap berasal dari scan PKG. Pemisahan antara pemberian nama dan penentuan tanggal ini penting untuk audit tahun yang jauh dari acuan. Perkiraan lunasi tidak boleh dianggap sebagai perhitungan semua panjang bulan Hijriah secara otomatis dan pasti.

6.8 Menyalin dan mengunduh hasil

Tombol Salin Hasil menyediakan teks keluaran yang sudah dihitung. Tombol Unduh CSV pada versi ini menghasilkan identitas lokasi, waktu UTC, mode, parameter ringkasan Matahari–Bulan, serta baris peristiwa. CSV tidak otomatis memuat seluruh tabel scan global, seluruh daftar fenomena, atau semua metadata yang diperlukan untuk replikasi. Pengguna yang membutuhkan audit global lengkap harus menyimpan informasi keputusan dan tabel terkait secara terpisah.

CSV memakai pemisah koma, pengutipan nilai, dan penanda UTF-8 pada berkas. Aplikasi spreadsheet dengan pengaturan regional tertentu mungkin memerlukan impor teks untuk memilih pemisah yang benar. Periksa bahwa angka, tanda negatif, dan waktu tidak berubah akibat konversi otomatis spreadsheet. Simpan pula versi sumber dan nama BSP jika hasil akan digunakan sebagai data penelitian.

6.9 Prosedur dokumentasi yang disarankan

Sebelum hitung, tetapkan tujuan: posisi sesaat, peristiwa harian, hilal lokal, atau kalender global. Kunci masukan yang dibutuhkan dan catat identitasnya. Sesudah hitung, periksa status siap, waktu lokal dan UTC, mode, serta kelengkapan tabel. Simpan hasil yang relevan berikut definisi dan versi. Saat menulis laporan, sebutkan keluaran mana yang langsung dihitung dan mana yang merupakan interpretasi penulis.

Prosedur ini tidak menambah pekerjaan administratif tanpa manfaat. Ia memungkinkan seseorang mengulang perhitungan, mengetahui apakah jam yang dipakai adalah jam manual atau sunset, dan memeriksa apakah nilai merupakan geosentris. Tanpa rekaman seperti itu, perbedaan hasil dapat sulit dibedakan dari perbedaan masukan sederhana.

BAB 7 HILAL LOKAL DAN INTERPRETASI VISIBILITAS

7.1 Kedudukan filter lokal

Status Hilal Lokal memakai ambang tinggi sekurang-kurangnya 5° dan elongasi sekurang-kurangnya 8° pada besaran hasil lokal. Pada jam manual, hasil itu adalah evaluasi saat yang dipilih; pada maghrib lokal, ia adalah evaluasi pada sunset. Status tersebut merupakan filter numerik. Ia tidak memasukkan seluruh syarat PKG global dan tidak menyatakan keberhasilan rukyat aktual.

$$\text{memenuhi lokal} = (h\text{Bulan} \geq 5^\circ) \text{ dan } (E \geq 8^\circ)$$

Jika mode koordinat lokal berubah, nilai tinggi dan elongasi lokal dapat berubah. Filter lokal kemudian mengikuti nilai tersebut. Hal ini berbeda dari filter global yang selalu geosentris. Laporan sebaiknya menyatakan “melewati filter lokal pada mode toposentris” atau “pada mode geosentris”, bukan menulis “hilal pasti terlihat” berdasarkan warna status.

7.2 Empat parameter pada maghrib

Tinggi Bulan membantu menilai kedudukannya terhadap horizon. Elongasi menggambarkan kedekatan arah dengan Matahari. Tinggi Matahari mencatat kondisi geometris saat evaluasi. Iluminasi menyatakan fraksi terang Bulan. Keempatnya memberi konteks lebih kaya daripada satu status ya atau tidak. Akan tetapi, tidak semua parameter menjadi komponen langsung predikat 5° – 8° dalam kode.

Tabel 3. Parameter hilal dan arti operasionalnya

Parameter	Satuan	Arti	Batas interpretasi
Tinggi Bulan	Derajat	Arah terhadap horizon model	Kerangka dan refraksi harus disebut.
Elongasi	Derajat	Separasi arah Bulan–Matahari	Bukan otomatis lebar sabit atau peluang terlihat.
Tinggi Matahari	Derajat	Keadaan geometris saat evaluasi	Kerangka dapat berbeda dari definisi sunset.
Iluminasi	Persen	Fraksi cakram terang secara geometris	Tidak memasukkan kontras atmosfer.

Tinggi Matahari geosentris pada sunset tidak harus tepat $-0,8333^\circ$ karena sunset ditentukan melalui peristiwa pengamat dan posisi Matahari kemudian dapat dilaporkan pada kerangka lain. Ini menjelaskan perbedaan kecil yang wajar antara angka pusat Matahari geosentris dan konvensi horizon. Membulatkan semua nilai Matahari menjadi satu konstanta akan menghilangkan informasi tentang metode yang dipakai.

7.3 Pemindaian kandidat bulanan

Bagian Kandidat Hilal Bulanan memeriksa sunset pada tanggal-tanggal bulan Gregorian yang dipilih, dengan data sebelum awal bulan untuk menetapkan konteks konjungsi. Ia menyajikan progres dari H0 dan mempertahankan tanggal yang lolos filter. Nilai H dalam bagian lokal didasarkan pada tanggal lokal konjungsi yang relevan, sedangkan H0 scan global didasarkan pada tanggal UTC konjungsi. Dua label H0 ini dapat berbeda satu tanggal pada wilayah tertentu.

Karena itu, kandidat lokal tidak boleh dijadikan pengganti keputusan awal bulan global. Satu tanggal yang lolos di kota pilihan menunjukkan keadaan kota tersebut, bukan keadaan seluruh sampel. Sebaliknya, keputusan global dapat diterima ketika lokasi pengguna sendiri masih belum memenuhi, karena aturan kalender mentransfer kriteria dari saksi yang diterima.

7.4 Mengapa visibilitas memerlukan lebih banyak informasi

Hilal adalah sumber cahaya yang tipis dengan kontras terhadap langit senja. Keterlihatan dipengaruhi ukuran dan bentuk sabit, kedudukan relatif Matahari–Bulan, atmosfer, kemampuan mata atau alat, dan kondisi horizon. Literatur visibilitas membangun model serta membandingkannya dengan data pengamatan. Doggett dan Schaefer membahas pengujian visibilitas, sedangkan Odeh menyusun kriteria dari kumpulan pengamatan dengan variabel geometri toposentris dan lebar sabit. [R11–R13]

Kriteria kalender 5° – 8° dan model prediksi pengamatan mempunyai fungsi yang berhubungan tetapi berbeda. Yang pertama menjadi aturan penyusunan kalender pada implementasi KHGT. Yang kedua berusaha menghubungkan kondisi dengan peluang atau kelas keterlihatan menurut basis data dan definisi observasi tertentu. Buku tidak menempatkan hasil satu model sebagai pengesahan otomatis terhadap model lain.

7.5 Geosentris dalam kalender dan toposentris dalam observasi

Pengamat di lapangan melihat dari satu titik permukaan, sehingga posisi toposentris relevan untuk arah instrumen dan kedudukan hilal terhadap horizon setempat. Kalender dapat memilih geosentris sebagai parameter aturan agar sistemnya konsisten. Ketika dua kerangka dipakai untuk dua tujuan, hasil harus

diberi label. Tidak ada alasan ilmiah untuk menghapus salah satunya hanya karena hasilnya berbeda.

Untuk menghubungkan kalender dengan pengamatan, pengguna dapat menyimpan kedua hasil pada timestamp sama. Selisihnya memperlihatkan kontribusi paralaks dan geometri pengamat. Namun, keputusan global tetap dievaluasi dengan kerangka yang diminta aturan, sementara rencana pointing teleskop memakai kerangka pengamatan yang sesuai. Ini merupakan contoh bagaimana pengetahuan tentang definisi mencegah kesalahan praktis.

7.6 Ambang pembulatan dan nilai dekat batas

Filter memakai nilai numerik internal, bukan angka tiga desimal yang sudah ditampilkan. Nilai $4,9996^\circ$ dapat terlihat sebagai $5,000^\circ$ setelah pembulatan, tetapi tetap lebih kecil dari 5° . Nilai $7,9996^\circ$ juga dapat terlihat $8,000^\circ$ tanpa melewati ambang internal. Karena itu, status dan angka tampilan yang tampak bertentangan di dekat batas dapat merupakan akibat pembulatan, bukan kesalahan logika.

Untuk audit dekat ambang, simpan lebih banyak digit dan catat margin terhadap kriteria: $\Delta h = h - 5^\circ$ serta $\Delta E = E - 8^\circ$. Margin positif kecil memerlukan perhatian terhadap koordinat, timestamp, dan definisi model. Margin tersebut bukan simpangan baku ketidakpastian; ia hanya jarak hasil deterministik terhadap ambang keputusan.

7.7 Menghubungkan hasil dengan catatan rukyat

Catatan rukyat yang berguna mencantumkan lokasi, waktu mulai dan berakhir, koordinat, horizon, keadaan atmosfer, alat, metode pencarian, serta hasil positif atau negatif. Tidak berhasil melihat dapat terjadi karena awan atau salah arah, sehingga tidak selalu menjadi bukti bahwa geometri mustahil. Berhasil melihat juga perlu dokumentasi yang cukup agar benda yang diidentifikasi benar-benar hilal.

SOLUNA dapat menyumbang prediksi arah, posisi, serta waktu, tetapi data pengamatan harus dikumpulkan secara terpisah. Untuk penelitian, perbandingan sebaiknya dilakukan setelah definisi keberhasilan observasi dan kriteria kualitas rekaman ditetapkan. Prosedur tersebut dapat digunakan sebagai pedoman metodologis untuk kegiatan observasi.

BAB 8 KHGT PKG 1 DAN PKG 2

8.1 Dasar penyatuan kalender

KHGT bertujuan menyediakan satu penanggalan Hijriah global. Dalam uraian kelembagaan, kesatuan matlak dan transfer imkan rukyat memungkinkan satu keadaan yang memenuhi syarat dipakai untuk keputusan kalender dunia. Prinsip itu harus diterjemahkan menjadi aturan waktu agar wilayah timur tidak memasuki awal bulan sebelum kondisi yang ditentukan terpenuhi. Buku saku resmi Majelis Tarjih menjelaskan kriteria Istanbul 2016 beserta koreksi setelah tengah malam UTC. [R14]

Kesamaan tanggal kalender global tidak berarti semua zona waktu mengalami tengah malam pada satu detik UTC yang sama. Pergantian hari sipil lokal tetap mengikuti zona masing-masing. Yang diseragamkan ialah label tanggal awal bulan menurut aturan kalender. Ungkapan “serentak” perlu dibaca dalam konteks keterpaduan tanggal, bukan penghapusan perbedaan waktu sipil di seluruh bumi.

8.2 Definisi operasional dalam SOLUNA

Tuliskan C sebagai timestamp konjungsi geosentris. H0 adalah tanggal UTC dari C. Untuk hari uji ke-j, $D = H0 + j$ dengan j bernilai 0, 1, atau 2. Setiap lokasi dievaluasi pada sunset lokal dengan tanggal sipil D. Batas B adalah 00.00 UTC pada tanggal D + 1. Calon tanggal awal bulan global adalah D + 1. F adalah fajar Gisborne pada tanggal lokal calon awal bulan tersebut.

Pada versi ini, fajar Gisborne didefinisikan sebagai saat pusat Matahari toposentris naik melewati -18° . Koordinatnya $-38,6533333^\circ$ lintang, $178,0041667^\circ$ bujur, elevasi 15 m, dengan zona Pacific/Auckland. Definisi ini adalah pilihan implementasi yang konkret. Ia perlu disebutkan karena kata “fajar Selandia Baru” tanpa titik dan sudut tidak memberi timestamp tunggal yang dapat direplikasi.

8.3 Filter numerik geosentris

Untuk lokasi x pada sunset sx, predikat numerik menggunakan tinggi Bulan geosentris hx dan elongasi geosentris Ex. Keduanya harus melewati ambang secara bersama. Sunset yang mendahului konjungsi atau berada di luar jendela waktu yang ditetapkan tidak menjadi saksi yang diterima. Jika peristiwa sunset atau data yang dibutuhkan tidak tersedia, lokasi itu tidak otomatis dianggap memenuhi.

$$Qx = (hx \geq 5^\circ) \text{ dan } (Ex \geq 8^\circ)$$

Predikat Qx belum membedakan PKG 1 dan PKG 2. Ia menjawab hanya apakah syarat sudut dipenuhi. Penetapan saksi kalender memerlukan pembandingan sx dengan B dan, untuk kondisi koreksi, pembandingan C dengan F. Seluruh pembandingan waktu harus dilakukan terhadap timestamp yang sama-sama dapat diurutkan dalam UTC.

8.4 PKG 1 sebelum tengah malam UTC

PKG 1 menerima lokasi daratan dalam sampel yang memenuhi Qx dan sunset-nya terjadi sebelum B. Saksi tersebut mendukung awal bulan pada $D + 1$. Tidak ada keharusan bahwa saksi berada di timur atau tengah bumi secara eksklusif; lokasi Amerika juga dapat memenuhi PKG 1 apabila timestamp UTC-nya masih sebelum batas. Menyebut PKG 1 hanya sebagai “wilayah timur” akan mengaburkan syarat waktu yang sebenarnya digunakan.

PKG 1: Qx dan $sx < B$, dengan sx tidak mendahului C

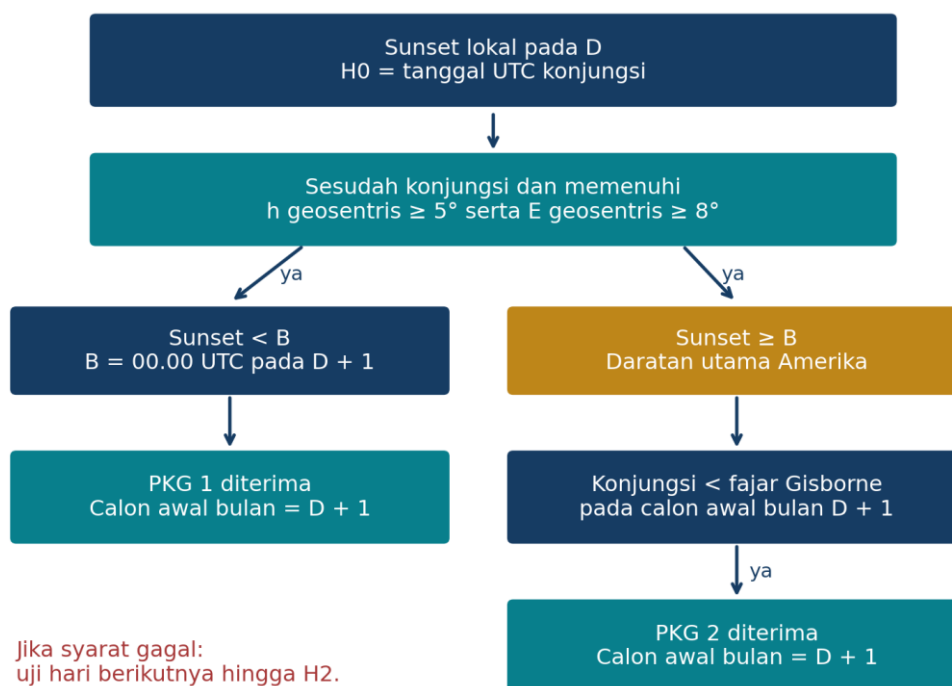
Pulau pada katalog dapat menjadi saksi PKG 1. Persyaratan daratan utama Amerika merupakan filter khusus untuk cabang PKG 2 pada implementasi ini. Untuk audit, tuliskan tanggal dan jam sunset UTC; nama benua saja belum menentukan cabang keputusan.

8.5 PKG 2 pada atau setelah tengah malam UTC

Jika syarat sudut baru diterima pada atau setelah B, program menerapkan PKG 2 kepada saksi di daratan utama Amerika. Penerimaan membutuhkan $C < F$. Konjungsi tepat sama dengan fajar tidak memenuhi syarat “sebelum” dalam kode. Sunset tepat pada B masuk cabang PKG 2, bukan PKG 1. Perlakuan pada titik batas ini perlu diuji karena perubahan operator dapat mengubah hasil kasus kritis.

PKG 2: Qx dan $sx \geq B$ dan AmerikaDaratan(x) dan $C < F$

Kalimat “hilal memenuhi sebelum fajar Gisborne” tidak boleh menggantikan syarat tersebut. Yang dibandingkan terhadap fajar adalah waktu konjungsi C. Timestamp sunset Amerika dapat terjadi kemudian daripada timestamp fajar Gisborne; keadaan itu tetap dapat diterima jika aturan PKG 2 dan syarat konjungsi terpenuhi. Perbedaan objek yang dibandingkan sangat menentukan ketepatan algoritma.



Gambar 3. Alur keputusan per lokasi pada hari uji D. Diagram menyederhanakan penanganan data tidak tersedia; keputusan akhir juga mengikuti prioritas dan cakupan sampel.

8.6 Bila koreksi tidak diterima

Jika syarat PKG 2 gagal, hari itu tidak diterima melalui cabang tersebut. SOLUNA menguji hari berikutnya dalam jendela H0–H2. Oleh karena itu, ungkapan “ditunda lusa” dapat dipakai sebagai uraian kalender untuk suatu keadaan, tetapi bukan perintah program untuk selalu menambah dua hari secara buta. Keadaan sunset dan fajar berikutnya tetap dihitung dan diperiksa.

Jika tidak ditemukan saksi hingga akhir jendela, program menyatakan tidak adanya keputusan yang diterima pada prosedur itu. Hasil tersebut tidak membuktikan bahwa tidak ada satu pun titik daratan yang memenuhi di seluruh dunia, sebab pencarian memakai katalog terbatas. Ekspansi jendela atau pencarian kontinu adalah pekerjaan pengembangan yang memerlukan pengujian tambahan.

8.7 Prioritas sampel dan pemilihan saksi

Daftar utama berisi 100 titik. Jika PKG 1 belum ditemukan dalam daftar utama pada suatu hari uji, program mengevaluasi 30 titik tambahan di daratan Amerika. Setelah himpunan yang relevan dihitung, PKG 1 diprioritaskan apabila tersedia; PKG 2 dipakai ketika tidak ada saksi PKG 1 yang diterima. Saksi yang

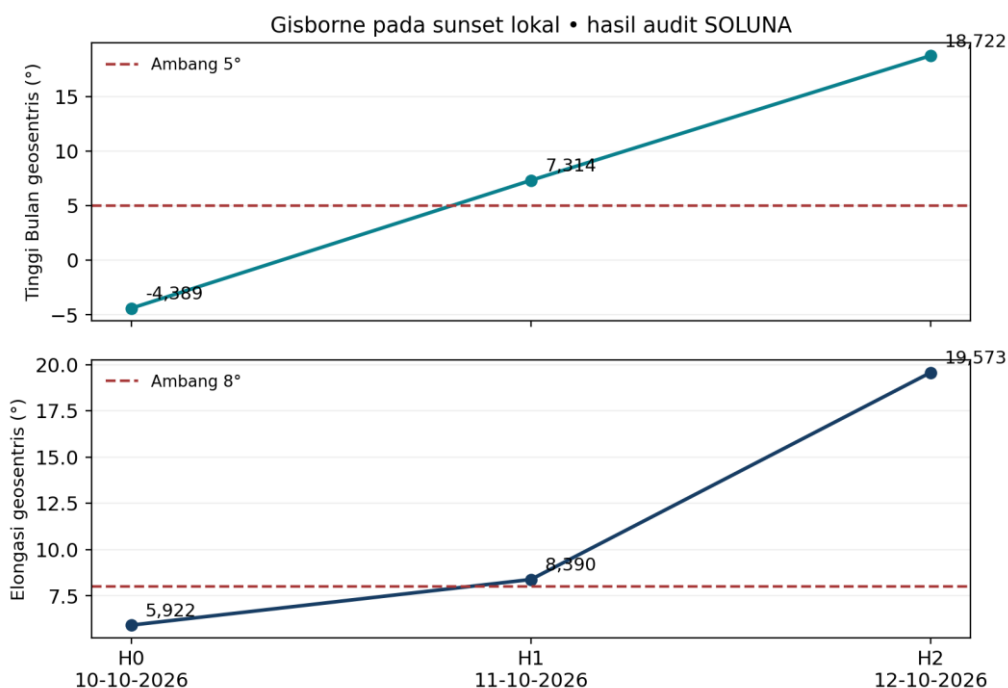
ditampilkan dipilih berdasarkan waktu sunset UTC paling awal di dalam kandidat yang diterima menurut prosedur tersebut.

Titik tambahan tidak selalu dievaluasi. Dengan demikian, “pertama” berkaitan dengan himpunan yang benar-benar dihitung dan kebijakan prioritas, bukan seluruh 130 titik dalam setiap keadaan. Cakupan juga belum mencakup daratan di antara titik-titik katalog. Resolusi geografis dan kepadatan kota yang tidak seragam dapat memengaruhi lokasi yang ditampilkan, walaupun tanggal awal bulan masih sesuai referensi.

Klasifikasi daratan utama Amerika pada kode bertumpu pada identitas zona America/ dan pengecualian nama pulau. Ini merupakan aturan katalog, bukan uji topologi geospasial memakai poligon daratan. Perubahan katalog atau penambahan kota perlu memeriksa klasifikasi itu secara eksplisit. Lampiran lokasi mempertahankan nilai sumber agar audit dapat diulang.

8.8 Jumadilawal 1448 H

Untuk kasus Oktober 2026, tidak ada saksi yang diterima pada H0 dalam prosedur yang diuji. Pada H1, Gisborne memenuhi dengan tinggi sekitar $7,314^\circ$ dan elongasi $8,390^\circ$. Sunset terjadi 11 Oktober 2026 pukul 06.21.47 UTC, jauh sebelum batas 12 Oktober pukul 00.00 UTC. Cabang yang diterima adalah PKG 1 dan awal bulan diperoleh 12 Oktober 2026.



Gambar 4. Tinggi dan elongasi pada sunset Gisborne untuk H0–H2 Oktober 2026. H0 di titik ini mendahului konjungsi; keputusan dipilih dari scan global, bukan dari grafik satu kota.

Label yang ditampilkan ialah “Tanggal 1 Jumadilawal 1448 H = 12-10-2026”. Label ini cocok dengan titik acuan penamaan bulan dalam antarmuka. Namun, alasan astronomis penetapan tanggal pada kasus tersebut tetap berasal dari evaluasi H1 dan PKG 1. Jangan membalik hubungan itu dengan memaksa scan menghasilkan tanggal acuan hanya agar label terlihat konsisten.

8.9 Jumadilakhir 1448 H

Untuk November 2026, hasil validasi menerima Arica, Chile, pada H0. Sunset 9 November sekitar 22.52.21 UTC memenuhi tinggi $6,906^\circ$ dan elongasi $8,988^\circ$. Karena timestamp itu mendahului 10 November pukul 00.00 UTC, kasus yang dipilih memenuhi PKG 1. Awal bulan ialah 10 November 2026. Draf kalender pembeding mencantumkan saksi Porto Alegre yang juga memenuhi sebelum batas UTC.

Kasus ini memperbaiki penafsiran bahwa tanggal 10 November harus dijelaskan sebagai PKG 2. Tanggal yang benar tidak mengharuskan alasan yang pertama kali diduga juga benar. Dalam buku ilmiah, alasan keputusan mengikuti timestamp dan predikat yang diuji. Pernyataan ini terbatas pada versi program, data, dan prosedur yang dicatat dalam studi validasi.

8.10 Syawal 1448 H sebagai PKG 2

Konjungsi terjadi 8 Maret 2027 sekitar 09.29.29 UTC. Manzanillo, Meksiko, memenuhi pada sunset lokal 8 Maret, yang dalam UTC jatuh 9 Maret sekitar 01.05.06. Tinggi sekitar $6,484^\circ$ dan elongasi $8,129^\circ$ melewati ambang, tetapi timestamp sunset sudah setelah B, yaitu 9 Maret pukul 00.00 UTC. Titik tersebut diklasifikasikan sebagai daratan utama Amerika.

Fajar Gisborne untuk calon tanggal 9 Maret dihitung oleh program pada 8 Maret sekitar 16.28.08 UTC. Konjungsi mendahului fajar tersebut, sehingga PKG 2 diterima dan awal Syawal diperoleh 9 Maret 2027. Buku saku resmi KHGT juga memakai contoh Syawal 1448 H, tetapi mencantumkan fajar Selandia Baru sekitar 16.35.16 UTC. Selisih sekitar tujuh menit menunjukkan perlunya menyamakan titik dan definisi fajar; pada kasus ini kedua nilai tetap menempatkan konjungsi sebelum fajar dan tidak mengubah cabang keputusan. [R14]

8.11 Penamaan bulan dan audit tahun lain

Tanggal awal bulan dan nomor lunasi harus dipisahkan dalam model data. SOLUNA menghitung tanggal Gregorian melalui scan, kemudian membentuk nama bulan dan tahun menggunakan acuan yang tersedia. Pendekatan berbasis perkiraan rata-rata lunasi dapat dipakai untuk menentukan urutan di sekitar acuan, tetapi jarak waktu jauh perlu diperiksa terhadap rangkaian konjungsi sebenarnya dan kalender pembeding.

Pengujian dua belas bulan 1448 H menunjukkan nama dan tahun yang benar pada himpunan tersebut. Ia belum membuktikan bahwa semua tahun nominal 1850–2149 telah diberi label Hijriah tanpa kesalahan. Audit masa jauh perlu memasukkan urutan konjungsi, pergantian Muharam, tahun kabisat kalender yang dibandingkan bila relevan, dan bulan Gregorian yang memuat dua konjungsi. Fungsi global versi ini memilih konjungsi pertama di dalam bulan Gregorian UTC yang diteliti, sehingga bulan dengan dua konjungsi merupakan kasus tersendiri yang perlu diuji.

BAB 9 PERISTIWA HARIAN DAN JADWAL

9.1 Terbit dan terbenam sebagai definisi model

Terbit dan terbenam tidak ditentukan hanya oleh tinggi pusat benda sama dengan 0° . Almanak perlu menentukan bagian cakram yang menjadi acuan dan model refraksi yang digunakan. Untuk Matahari, konvensi yang umum memakai tepi atas tampak di horizon, setara pusat geometris sekitar 50 menit busur di bawah horizon dengan asumsi standar tertentu. Nilai itu menggabungkan semidiameter sekitar 16 menit busur dan refraksi horizon sekitar 34 menit busur. [R16]

SOLUNA memakai rutin rise dan set Skyfield untuk Matahari dan Bulan. Hasil tersebut merupakan peristiwa toposentris, walaupun mode tabel posisi dipilih geosentris. Peristiwa horizon sebaiknya tidak dibandingkan dengan crossing tinggi geosentris 0° tanpa menjelaskan perubahan definisi. Pada Bulan, ukuran tampak dan paralaks lebih bervariasi sehingga penggunaan konstanta Matahari untuk moonrise atau moonset tidak tepat.

Horizon nyata dapat terangkat oleh pegunungan atau bangunan, atau berbeda karena pengamat berada di ketinggian menghadap laut. Model horizon standar merupakan titik awal almanak. Untuk kegiatan pengamatan dengan tuntutan menit atau lebih baik, profil horizon setempat perlu dicatat. Kepresisian timestamp komputer tidak meniadakan ketidakpastian pengamatan horizon.

9.2 Transit dan kulminasi

Transit atas adalah saat benda melewati meridian lokal bagian atas. Pada Matahari, peristiwa ini menjadi dasar Dhuhur dalam program. Transit tidak wajib berada tepat pada pukul 12.00 jam sipil, karena bujur dalam zona, persamaan waktu, dan aturan waktu sipil memengaruhi hubungan itu. Transit juga tidak selalu persis tengah antara terbit dan terbenam. [R16]

Tinggi saat transit membantu menghubungkan lintang dan deklinasi. Dalam geometri sederhana, tinggi kulminasi atas berkaitan dengan $90^\circ - |\varphi - \delta|$ ketika kondisi meridian sesuai. Perhitungan nyata memakai posisi pada timestamp transit. Program tidak menambahkan ihtiyat otomatis pada Dhuhur, sehingga keluaran itu perlu disebut sebagai dasar astronomis, bukan jadwal institusi yang telah memasukkan seluruh kebijakan penyesuaian.

9.3 Twilight dan keadaan langit

Twilight sipil, nautika, dan astronomis masing-masing berhubungan dengan pusat Matahari sekitar 6° , 12° , dan 18° di bawah horizon. Fajar adalah cabang naik, sedangkan senja adalah cabang turun. Ambang tersebut merupakan konvensi

astronomis untuk mengelompokkan pencahayaan, bukan pengukuran langsung luminansi langit pada saat itu. Awan, aerosol, dan lingkungan terang setempat dapat mengubah pengalaman pengamat. [R16]

SOLUNA mengklasifikasikan keadaan langit pada saat hitung dan menampilkan peristiwa dawn serta dusk. Nilai -18° juga digunakan untuk fajar Gisborne pada PKG 2 dan salah satu pilihan Shubuh. Penggunaan angka yang sama tidak menyamakan semua tujuan: twilight menggambarkan konvensi astronomi, jadwal memakai parameter yang dipilih, dan PKG memakai syarat kalender terhadap timestamp tersebut.

9.4 Persamaan waktu dan panjang siang

Persamaan waktu menyatakan selisih waktu Matahari tampak dan waktu Matahari rata-rata dengan konvensi tanda tertentu. Pada kode, ia dihitung dari timestamp transit, bujur, dan posisi transit terhadap tengah hari waktu Matahari rata-rata. Nilai positif berarti transit apparent terjadi sebelum pukul 12.00 waktu Matahari rata-rata pada konvensi ini. Referensi yang mendefinisikan tanda sebaliknya akan menampilkan angka berlawanan tanpa harus berbeda secara fisik.

Panjang siang dibentuk dari selisih sunset dan sunrise yang ditemukan. Jika salah satunya tidak tersedia, panjang siang dapat kosong. Pada lintang tinggi, keadaan tanpa terbit atau tanpa terbenam membutuhkan interpretasi khusus. Kode tidak menyediakan seluruh prosedur pengganti untuk hari atau malam kutub. Membaca sel kosong sebagai panjang siang nol akan menghasilkan kesimpulan yang salah.

9.5 Parameter jadwal yang diprogram

Tabel 4. Definisi jadwal pada kode SOLUNA

Baris	Dasar hitung	Penyesuaian dan konteks
Shubuh -20° / -18°	Pusat Matahari toposentris naik melewati dua ambang	Dua parameter ditampilkan berdampingan.
Sunrise	Rutin terbit Matahari Skyfield	Horizon standar rutin.
Dhuha	Pusat Matahari naik melewati $+4,5^\circ$	Parameter yang dipilih aplikasi.
Dhuhur	Transit atas Matahari	Tidak ada tambahan menit ihtiyat.
Ashar	Faktor bayangan tambahan 1	Target tinggi dihitung dari lintang dan deklinasi transit.
Maghrib	Rutin sunset Skyfield	Dasar mode Maghrib Lokal.

Baris	Dasar hitung	Penyesuaian dan konteks
Isya	Pusat Matahari turun melewati -18°	Crossing geometris toposentris.
Midnight	Tengah interval Maghrib–Shubuh -18° berikutnya	Berbeda dari tengah malam sipil.
1/3 Malam	Dua pertiga interval malam yang sama	Menunjukkan awal sepertiga terakhir.

Jadwal tersebut merupakan hasil parameter astronomis versi SOLUNA. Buku tidak menganggap angka sudut sebagai ketentuan universal semua organisasi atau mazhab. Dua nilai Shubuh ditampilkan agar perbedaan parameter dapat dilihat. Ketika jadwal dipakai dalam kegiatan lembaga, pengguna perlu memeriksa parameter resmi yang berlaku, kebijakan ihtiyat, serta perlakuan lintang tinggi secara tersendiri.

Perbedaan Shubuh -20° dan -18° berasal dari saat crossing ketinggian Matahari yang berbeda. Selisih menitnya tidak selalu tetap sepanjang tahun atau untuk semua lintang. Mengubah hasil -18° menjadi -20° dengan menambah suatu jumlah menit konstan akan mengabaikan laju perubahan tinggi Matahari yang bergantung pada geometri. Untuk pembelajaran, perbandingan kedua nilai pada beberapa musim memberi penjelasan yang lebih baik daripada aturan menit tetap.

9.6 Model bayangan Ashar

Program memakai faktor bayangan tambahan satu kali tinggi benda. Jika tinggi benda b dan bayangan pada kulminasi adalah $b \tan|\varphi - \delta|$ dalam model sederhana, panjang bayangan target menjadi $b(1 + \tan|\varphi - \delta|)$. Sudut tinggi target kemudian dihitung dari hubungan kotangen antara tinggi dan panjang bayangan.

$$h_{\text{Ashar}} = \arctan[1 / (1 + \tan|\varphi - \delta_{\text{transit}}|)]$$

Persamaan ini menjelaskan model yang diprogram. Program tidak menyediakan pilihan faktor dua pada versi yang dibahas. Pada kondisi ekstrem, seperti Matahari tidak mencapai konfigurasi harian biasa atau selisih zenit yang keluar dari ranah model bayangan sederhana, hasil perlu diuji tersendiri. Menuliskan persamaan tidak cukup untuk memastikan semua lintang dan semua tanggal telah ditangani secara benar.

9.7 Tengah malam dan sepertiga terakhir

Midnight pada tabel bukan selalu pukul 00.00 sipil. Ia adalah titik tengah interval dari Maghrib hingga Shubuh -18° hari berikutnya. Awal sepertiga malam terakhir berada pada dua pertiga interval yang sama dari Maghrib. Definisi interval harus

dibaca sebelum membandingkannya dengan jadwal yang memakai sunset–sunrise atau Isya–Shubuh.

$$t_{\text{Tengah}} = t_{\text{Maghrib}} + (t_{\text{ShubuhBerikut}} - t_{\text{Maghrib}}) / 2$$

$$t_{\text{SepertigaTerakhir}} = t_{\text{Maghrib}} + 2(t_{\text{ShubuhBerikut}} - t_{\text{Maghrib}}) / 3$$

Apabila salah satu batas interval tidak tersedia, keluaran terkait juga dapat kosong. Perubahan jam musiman pada malam yang melintasi transisi zona waktu merupakan kasus pengujian khusus; studi validasi kalender 1448 H tidak merupakan validasi menyeluruh terhadap seluruh operasi durasi malam di semua zona. Untuk kebutuhan presisi, audit durasi sebaiknya membandingkan selisih timestamp dalam UTC dan selisih waktu lokal.

9.8 Pemakaian dalam pendidikan dan observasi

Peristiwa harian dapat digunakan untuk merencanakan kapan mulai mencari hilal, kapan Matahari turun ke tingkat twilight tertentu, dan berapa lama Bulan berada dekat horizon. Pengguna tetap perlu memperhitungkan waktu penyiapan alat dan kondisi lapangan. Tabel peristiwa merupakan bantuan perencanaan, bukan pengganti pemeriksaan langsung cuaca dan horizon.

Dalam pengajaran, satu lokasi dapat dipertahankan sambil tanggal diubah untuk melihat musim, atau satu tanggal dipertahankan sambil lintang diubah untuk melihat perbedaan panjang siang. Perbandingan keluaran mendukung penjelasan hubungan antara geometri langit dan lokasi pengamat. Interpretasi harus tetap membedakan perubahan model, perubahan masukan, dan perubahan keadaan astronomis.

BAB 10 FENOMENA ORBITAL DAN GERHANA

10.1 Jendela pencarian fenomena

Tab Fenomena memuat kondisi lokal, fase saat perhitungan, fase utama berikutnya, node Bulan, jarak ekstrem, serta sejumlah calon peristiwa gerhana. Jendela pencarian berbeda menurut fenomena. Kode mencari fase dalam sekitar 40 hari dan node dalam sekitar 35 hari dari waktu masukan. Pencarian perigee dan apogee menggunakan jendela sekitar 40 hari dengan grid jarak dan pemurnian sampel.

Jendela adalah bagian dari definisi hasil. “Perigee berikutnya” dalam tabel berarti perigee yang berhasil ditemukan oleh prosedur pada jendela tersebut. Apabila proses gagal atau data tidak menjangkau interval, ketiadaan baris tidak boleh langsung dinilai sebagai ketiadaan perigee. Penyangga pada BSP membantu pencarian lintas tahun, tetapi tetap dibatasi cakupan fisik data asal.

SOLUNA juga mencari ekuinoks dan solstis dalam jendela sekitar 370 hari. Ekuinoks berkaitan dengan lintasan Matahari melintasi ekuator langit, sedangkan solstis berkaitan dengan ekstrem deklinasi tahunan. Panjang siang dan malam pada ekuinoks tidak harus tepat sama menurut waktu sunrise–sunset standar, karena semidiameter, refraksi, lintang, dan definisi horizon memengaruhi durasinya.

Perihelion dan aphelion Bumi dicari dari jarak Bumi–Matahari dalam jendela sekitar 380 hari dengan sampel awal enam jam dan pemurnian berikutnya. Kedua peristiwa jarak itu berbeda dari solstis; musim tidak disebabkan terutama oleh dekat atau jauhnya Bumi terhadap Matahari. Pemisahan ini membantu membaca tab Fenomena tanpa mencampurkan kategori orbit dan orientasi.

10.2 Node perigee dan apogee

Node adalah perpotongan lintasan Bulan dengan bidang ekliptika. Node naik berarti lintang ekliptika berubah dari selatan ke utara; node turun dari utara ke selatan. Kedekatan fase baru atau purnama dengan node menjelaskan mengapa gerhana tidak terjadi pada setiap lunasi. Fase mengatur susunan Matahari–Bumi–Bulan, sedangkan node mengatur apakah penyimpangan dari bidang memungkinkan bayangan berpotongan.

Perigee adalah jarak minimum Bulan terhadap Bumi dalam siklus terkait dan apogee jarak maksimum. Saat kedua peristiwa itu tidak sama dengan waktu fase utama secara umum. Purnama yang dekat perigee dapat mempunyai diameter sudut lebih besar daripada purnama yang dekat apogee, tetapi istilah populer

untuk keadaan itu tidak menjadi jenis gerhana. Buku memakai nama peristiwa geometris agar pembaca tidak mencampurkan kategori.

10.3 Mengapa gerhana tidak terjadi setiap bulan

Pada fase baru, Bulan berada pada arah Matahari, tetapi lintang ekliptikanya dapat membuat bayangan lewat di atas atau di bawah Bumi. Pada purnama, Bulan berlawanan arah dengan Matahari, tetapi dapat lewat di luar bayangan Bumi. Gerhana memerlukan kedekatan arah dan posisi terhadap bidang atau sumbu bayangan yang cukup. Kondisi tersebut merupakan hubungan tiga dimensi, bukan sekadar kesamaan tanggal fase.

Ukuran sudut Matahari dan Bulan menentukan jenis gerhana Matahari bagi titik yang berada pada jalur sentral. Bila cakram Bulan menutup seluruh cakram Matahari, pengamat mengalami totalitas. Bila Bulan tampak lebih kecil dan meninggalkan cincin Matahari, pengamat mengalami anularitas. Pada konfigurasi tertentu, jenis sentral dapat berubah sepanjang lintasan, sehingga disebut hibrida. Untuk gerhana Bulan, klasifikasi mengikuti hubungan cakram Bulan dengan penumbra dan umbra Bumi.

10.4 Gerhana Bulan pada versi SOLUNA

SOLUNA menggunakan fasilitas gerhana Bulan Skyfield untuk memperoleh kandidat dan jenis penumbral, parsial, atau total. Ringkasan pada tab Gerhana berfokus pada tahun dan bulan yang dipilih. Bagian fenomena dapat mencari gerhana Bulan berikutnya dalam jendela hingga sekitar lima tahun. Dua tampilan dengan rentang berbeda tidak wajib memiliki isi yang sama.

Penilaian lokal menggunakan tinggi Bulan pada saat maksimum global yang diperoleh. Jika Bulan berada di atas horizon pada saat itu, program dapat menandai keadaan yang mendukung pengamatan maksimum. Namun, hal ini bukan perhitungan lengkap seluruh kontak lokal dan durasi keterlihatan. Bulan di bawah horizon pada maksimum tidak selalu berarti seluruh fase awal atau akhir gerhana juga tidak terlihat dari lokasi tersebut.

10.5 Pendekatan gerhana Matahari

Untuk gerhana Matahari, versi yang dibahas membentuk pendekatan kerucut bayangan dari vektor Matahari, Bulan, dan Bumi. Kandidat dikumpulkan di sekitar fase baru, lalu jarak sumbu bayangan ke Bumi diperiksa. Penumbra yang berpotongan dengan Bumi mendukung kandidat parsial; hubungan sumbu dan geometri umbra atau antumbra dipakai untuk perkiraan jenis sentral. Bumi diperlakukan secara sederhana dalam bagian geometri bayangan tersebut.

Pencarian saat terbaik memakai sampel awal sekitar lima menit dan pemurnian sekitar dua puluh detik pada jendela yang lebih sempit. Karena itu, detik pada timestamp maksimum tidak menunjukkan bahwa pencarian kontinu telah mencapai ketelitian satu detik. Posisi titik bayangan juga merupakan keluaran pendekatan, yang perlu dibandingkan dengan katalog dan metode gerhana standar sebelum dipakai untuk perencanaan lintasan presisi.

10.6 Uji lokal pada maksimum global

Untuk kandidat Matahari, aplikasi memeriksa tinggi Matahari dan tumpang tindih cakram dari lokasi pada perkiraan maksimum global. Pemeriksaan pada satu saat ini tidak sama dengan pencarian maksimum lokal. Maksimum lokal dapat mendahului atau mengikuti maksimum global, dan gerhana parsial lokal dapat berlangsung pada interval ketika syarat tidak terpenuhi tepat pada saat global tersebut.

Dengan demikian, status lokal pada ringkasan harus dibaca sesuai waktu pemeriksaannya. Program belum menghitung semua kontak C1, C2, C3, dan C4 bagi setiap lokasi melalui metode lengkap. Pernyataan “tidak memenuhi pada maksimum global yang diperiksa” lebih tepat daripada “gerhana tidak mungkin terlihat sepanjang peristiwa” apabila hanya satu timestamp yang diuji.

10.7 Referensi pembanding yang tepat

NASA menyediakan kanon dan katalog gerhana dengan peta lintasan, klasifikasi, serta pembahasan pengaruh ΔT . Referensi seperti itu sesuai untuk menguji jenis, tanggal, saat maksimum, dan geografi gerhana. Pembandingan harus memisahkan koordinat pusat lintasan, batas penumbra, kontak lokal, serta waktu maksimum. Menyamakan tanggal saja belum cukup untuk menilai ketepatan model gerhana. [R15]

Metode elemen Besselian dan transformasi yang teliti dapat dipakai untuk verifikasi lebih lanjut. Buku tidak mengklaim bahwa validasi KHGT 1448 H sekaligus memvalidasi modul gerhana. Keduanya mempunyai fungsi dan data pembanding yang berbeda. Audit modul gerhana perlu disusun sebagai himpunan uji tersendiri, termasuk kasus parsial, total, cincin, hibrida, dan lokasi di dekat batas keterlihatan.

10.8 Keselamatan pengamatan Matahari

Pengamatan Matahari membutuhkan penyaring yang memang dirancang untuk pengamatan Matahari serta pemasangan yang benar pada alat. Kacamata hitam biasa tidak memadai. Melihat melalui teleskop, binokular, atau kamera tanpa penyaring yang sesuai dapat menimbulkan bahaya pada mata dan alat. Keluaran

posisi dari SOLUNA tidak memberi perlindungan fisik apa pun. Panduan keselamatan NASA menjadi rujukan untuk kegiatan pengamatan. [R19]

Untuk kegiatan akademik, penanggung jawab observasi perlu menentukan metode aman sebelum pengamatan dimulai, misalnya sistem proyeksi yang dirancang dengan benar atau penyaring yang tepat. Buku ini menempatkan keselamatan sebagai bagian dari penggunaan data arah Matahari, bukan sebagai fitur yang dilakukan perangkat lunak.

BAB 11 VALIDITAS DAN RELIABILITAS VERSI YANG DIUJI

11.1 Pertanyaan dan bahan pengujian

Studi yang dirangkum dalam bab ini memeriksa dua hal: apakah tanggal awal bulan hasil SOLUNA sesuai dengan PDF draf kalender KHGT 1448 H yang disediakan, dan apakah lokasi yang ditampilkan identik dengan saksi yang dicantumkan PDF. Pengujian menilai keluaran terhadap satu himpunan pembandingan tetap, bukan terhadap sampel acak seluruh tahun. Identitas kode dan PDF dicatat melalui hash agar versi yang dibandingkan jelas. [R20, R21]

Bahan pembandingan adalah draf enam halaman yang menyajikan dua belas bulan 1448 H, dari Muharam pada Juni 2026 sampai Zulhijah pada Mei 2027. Elevasi titik PDF tidak tersedia lengkap, sehingga uji ulang saksi referensi menggunakan asumsi elevasi 0 m. Asumsi tersebut dinyatakan dalam data diagnostik dan membatasi penafsiran selisih detik. Tahun 1448 H tidak seluruhnya berada dalam satu tahun Gregorian, sehingga kernel 2026 dan 2027 dibutuhkan pada bagian yang berbeda.

11.2 Desain pengulangan

Mesin Python diekstrak dari sumber tanpa mengubah algoritma. Untuk setiap bulan dilakukan satu scan dasar, satu pengulangan dengan masukan identik, dan satu varian yang mengubah mode lokal menjadi toposentris, zona acuan menjadi Pacific/Auckland, serta tanggal menjadi tanggal 20 dalam bulan Gregorian yang sama. Totalnya 36 scan. Varian tersebut menguji bahwa keputusan global tetap memakai geosentris dan konjungsi bulan yang sama pada himpunan terpilih.

Pengulangan identik dibandingkan terhadap seluruh keluaran global. Untuk varian, metadata yang memang berubah akibat masukan atau format zona dinormalisasi sebelum membandingkan keputusan dan nilai astronomi terkait. Ada pula sebelas uji batas predikat PKG serta pemeriksaan penamaan bulan dan interval awal bulan. Desain tersebut menguji beberapa sifat penting, tetapi belum mencakup semua kombinasi masukan atau semua kejadian kesalahan jaringan.

11.3 Hasil tanggal bulanan

Tabel 5. Awal bulan 1448 H dan lokasi saksi pada dua sumber

Bulan	Tanggal sama	Saksi SOLUNA	Saksi PDF	PKG / H
Muharam	16-06-2026	Karachi, Pakistan	Islamabad, Pakistan	PKG 1 / H0
Safar	15-07-2026	Lima, Peru	Caracas, Venezuela	PKG 1 / H0
Rabiulawal	14-08-2026	Port Moresby, Papua New Guinea	Jayapura, Indonesia	PKG 1 / H1
Rabiulakhir	12-09-2026	Recife, Brazil	Salvador, Brazil	PKG 1 / H0
Jumadilawal	12-10-2026	Gisborne, New Zealand	Christchurch, New Zealand	PKG 1 / H1
Jumadilakhir	10-11-2026	Arica, Chile	Porto Alegre, Brazil	PKG 1 / H0
Rajab	10-12-2026	Dakar, Senegal	Sofala, Mozambik	PKG 1 / H0
Syakban	09-01-2027	Muscat, Oman	Muscat, Oman	PKG 1 / H1
Ramadan	08-02-2027	Ambon, Indonesia	Darwin, Australia	PKG 1 / H1
Syawal	09-03-2027	Manzanillo, Mexico	Guadalajara, Mexico	PKG 2 / H0
Zulkaidah	08-04-2027	Karachi, Pakistan	Madras, India	PKG 1 / H1
Zulhijah	07-05-2027	New York, USA	St. Johns, Canada	PKG 1 / H0

Seluruh dua belas tanggal awal bulan cocok dengan PDF pembandingan, yaitu 12/12 pada himpunan ini. Interval antara sebelas pasangan awal bulan yang tersedia adalah 29 atau 30 hari dan konsisten antara kedua sumber. Panjang akhir Zulhijah menuju Muharam berikutnya belum dinilai karena penutupan berikutnya tidak termasuk dua belas awal bulan pembandingan.

Kesimpulan yang didukung ialah kesesuaian tanggal untuk 1448 H pada versi dan prosedur ini. Tidak tepat mengubah angka 12/12 menjadi jaminan akurasi 100% untuk semua tahun, semua kota, atau seluruh modul aplikasi. Persentase pada himpunan deterministik kecil merupakan ringkasan kesesuaian, bukan estimasi tingkat keberhasilan populasi yang telah diuji secara statistik.

11.4 Perbandingan lokasi saksi

Hanya satu dari dua belas nama kota cocok, yaitu Muscat pada Syakban. Bahkan pada Muscat, titik koordinat program dan PDF tidak identik. Tidak ada dari dua belas koordinat referensi yang identik persis dengan titik yang dipakai untuk saksi program. Karena itu, jawaban atas pertanyaan “apakah tanggal dan tempat pertama sama” harus dipisahkan: tanggal sama pada dua belas bulan, lokasi tidak identik pada keseluruhan himpunan.

PDF memakai rumusan bahwa PKG terpenuhi “antara lain” pada suatu lokasi. Rumusan ini menunjukkan contoh saksi dan tidak selalu mendefinisikan titik pertama global. Program memilih saksi awal dalam sampel yang dievaluasi. Pada sembilan kasus, sunset yang dihitung ulang pada titik PDF mendahului saksi program dan tetap memenuhi. Fakta itu memperlihatkan keterbatasan klaim “pertama”, tetapi tidak otomatis menggugurkan tanggal kalender yang didukung oleh saksi lain.

11.5 Pemeriksaan ulang titik referensi

Semua dua belas titik PDF memenuhi tinggi dan elongasi serta cabang PKG yang relevan apabila dihitung ulang pada sunset model dengan koordinat referensi. Ketika posisi dihitung pada jam yang tercetak dalam PDF, hanya sepuluh dari dua belas yang melewati kedua ambang. Perbedaan ini memperlihatkan bahwa konsistensi timestamp sama pentingnya dengan nilai sudut yang dicantumkan.

Tabel 6. Empat selisih besar waktu sunset pada titik PDF

Bulan	Titik referensi	Tercetak UTC	Hitung ulang UTC	Selisih menit
Jumadilawal	Christchurch, New Zealand	11-10-2026 05:48:15	11-10-2026 06:48:15	+60,01
Ramadan	Darwin, Australia	07-02-2027 09:17:59	07-02-2027 09:47:58	+29,99
Zulkaidah	Madras, India	07-04-2027 12:21:15	07-04-2027 12:51:15	+30,00
Zulhijah	St. Johns, Canada	06-05-2027 23:19:43	06-05-2027 22:49:45	-29,97

Empat selisih besar muncul pada Christchurch, Darwin, Madras, dan St. Johns. Penyebab pasti tidak dapat ditetapkan hanya dari PDF dan perhitungan ini, karena dasar zona, koordinat, horizon, serta proses editorial sumber tidak sepenuhnya dijelaskan. Buku tidak menyatakan semua perbedaan itu sebagai salah ketik. Ia mencatat ketidaksesuaian waktu yang memerlukan klarifikasi sumber.

Pada jam tercetak untuk Madras, elongasi yang dihitung sekitar $7,921^\circ$, di bawah 8° . Pada jam tercetak untuk St. Johns, tinggi sekitar $3,164^\circ$, di bawah 5° . Pada sunset yang dihitung ulang, keduanya melewati ambang. Madras memiliki tinggi sekitar $5,031^\circ$ pada saat itu, cukup dekat dengan batas sehingga perubahan definisi atau titik tetap penting. Selisih konjungsi antara mesin dan PDF kurang dari satu detik pada seluruh bulan, dengan maksimum absolut sekitar 0,736 detik.

11.6 Reliabilitas yang berhasil diperiksa

Dua belas pasangan pengulangan identik memberikan keluaran global yang identik. Dua belas pasangan varian mempertahankan keputusan dan nilai terkait setelah normalisasi metadata yang memang berbeda. Sebelas uji batas PKG lulus. Hal ini mendukung repeatability dan beberapa sifat invariansi rancangan pada versi yang diuji.

Istilah reliabilitas di sini tidak merujuk pada konsistensi kuesioner psikometrik, sehingga Cronbach alpha tidak relevan. Yang diperiksa adalah stabilitas keluaran deterministik, ketepatan operator batas, dan ketidakbergantungan keputusan global terhadap mode lokal yang seharusnya tidak memengaruhinya. Reliabilitas antarperangkat browser, kondisi memori, perubahan cache, dan versi paket perlu diuji terpisah.

11.7 Akses BSP dan waktu eksekusi

Catatan pembukaan berkas hanya menunjukkan BSP tahunan 2026 dan 2027 pada penyiapan atau perpindahan tahun. Tidak ada pembukaan BSP tambahan selama masing-masing scan berjalan. Berkas asli de440s.bsp dan berkas rentang tidak dibuka dalam himpunan uji tersebut. Temuan ini mendukung bahwa pemindaian menggunakan kernel aktif yang sama, sesuai rancangan pemilih berkas.

Waktu scan pada lingkungan native penguji rata-rata sekitar 5,553 detik, dengan minimum 4,863 dan maksimum 6,686 detik. Angka ini tidak memasukkan pengunduhan awal, penyiapan Pyodide, dan render antarmuka pada HP atau laptop pengguna. Laporan kinerja browser perlu memisahkan tahap tersebut dan mencatat perangkat, kondisi jaringan, serta cache.

11.8 Matriks batas kesimpulan

Tabel 7. Kesimpulan yang didukung dan batas pengujiannya

Aspek	Bukti pada himpunan uji	Batas
Tanggal kalender	12/12 sesuai PDF 1448 H	Tidak mencakup semua tahun.
Nama saksi	1/12 sama	Sumber PDF memberi contoh saksi.
Koordinat saksi	0/12 identik persis	Nama kota bukan identitas titik.
Pengulangan	12 pasangan identik	Repeatability pada satu lingkungan.
Varian	12 pasangan konsisten setelah normalisasi	Tidak semua varian masukan diuji.
Batas PKG	11 uji predikat lulus	Tidak seluruh skenario geografi/jaringan.
Akses BSP	Tahunan aktif dipakai ulang	Benchmark browser belum menyeluruh.
Gerhana / rukyat	Tidak divalidasi oleh uji kalender ini	Memerlukan himpunan pembandingan lain.

Hasil yang baik perlu dipertahankan bersama batas buktinya. Kesesuaian tanggal memberikan dasar penggunaan pada kasus yang sudah diuji. Perbedaan lokasi, beberapa ketidaksesuaian waktu PDF, dan cakupan geografis terbatas memberi arah pengembangan. Menyampaikan keduanya secara bersamaan lebih informatif daripada menyebut aplikasi sepenuhnya benar atau sepenuhnya salah berdasarkan satu jenis keluaran.

BAB 12 TATA KERJA AKADEMIK DAN PENGEMBANGAN

12.1 Memulai dari pertanyaan penelitian

Pertanyaan penelitian menentukan data yang harus direkam. Studi perbandingan tinggi hilal memerlukan koordinat, timestamp, kerangka, dan refraksi. Studi kalender memerlukan rangkaian konjungsi, sunset saksi, batas UTC, fajar, dan keputusan. Studi performa memerlukan tahap setup, ukuran berkas, perangkat, cache, serta lama scan. Satu CSV ringkasan tidak otomatis memadai untuk seluruh jenis penelitian tersebut.

Sebelum menghitung, tetapkan satuan, definisi, dan kondisi yang akan dikunci. Jika pertanyaan tentang pengaruh paralaks, waktu harus sama. Jika pertanyaan tentang pengaruh definisi sunset, timestamp yang berbeda justru menjadi bagian hasil yang dianalisis. Menetapkan variabel dengan jelas mencegah kesimpulan yang menisbahkan seluruh selisih kepada satu faktor padahal beberapa parameter berubah bersamaan.

12.2 Rekaman minimal yang dapat direplikasi

Rekaman yang layak memuat versi sumber, hash kernel, versi lingkungan, lokasi, elevasi, zona IANA, tanggal lokal, UTC, mode waktu, mode koordinat, hasil mentah, dan catatan penanganan data kosong. Untuk keputusan KHGT, tambahkan H0, hari uji, batas UTC, parameter fajar Gisborne, daftar yang dievaluasi, status PKG, dan identitas saksi yang dipilih. Format mesin dapat dipakai untuk angka, sedangkan laporan manusia menggunakan format Indonesia yang jelas.

Simpan hasil mentah sebelum membuat pembulatan dan tabel publikasi. Pembulatan dapat mengubah pembacaan dekat batas, sehingga keputusan sebaiknya ditelusuri ke nilai asli. Jika hasil diedit untuk tata letak, perubahan penyajian harus dapat dibedakan dari perubahan nilai. Nama kota yang disingkat tidak boleh menyebabkan titik berbeda dianggap identik.

12.3 Perbandingan dengan sumber independen

Sumber independen sebaiknya memakai metode dan data yang dapat ditelusuri. Perbandingan dua antarmuka yang sebenarnya menjalankan algoritma yang sama hanya menguji penyajian atau konsistensi implementasi, bukan independensi model. Untuk astronomi posisi, cocokkan kerangka, epoch, waktu, dan koreksi. Untuk gerhana, gunakan katalog dengan definisi maksimum dan kontak yang jelas. Untuk kalender, cocokkan pula aturan keputusan dan cakupan lokasi.

Laporkan selisih sebagai besaran yang relevan: detik waktu, menit busur, kilometer koordinat, hari kalender, atau perbedaan cabang PKG. Jangan menggabungkan semua menjadi satu skor tanpa alasan. Dua tanggal yang sama dengan kota berbeda menunjukkan kesesuaian keputusan, tetapi perbedaan pemilihan saksi. Dua waktu yang selisih satu jam dengan tinggi yang mirip dapat menandakan masalah pelabelan zona, tetapi penyebab itu perlu diuji dan tidak cukup ditebak.

12.4 Ketidakpastian dan sensitivitas

Ketidakpastian masukan mencakup koordinat, elevasi, dan waktu. Ketidakpastian model mencakup refraksi, horizon, rotasi Bumi, dan pendekatan pencarian. Ketidakpastian pengamatan mencakup atmosfer, alat, serta identifikasi objek. Semua itu berbeda dari error pembulatan keluaran. Laporan yang menyebut enam angka desimal harus tetap menjelaskan kualitas data masukan dan metode.

Analisis sensitivitas dapat dilakukan dengan mengubah satu parameter dalam rentang yang beralasan, lalu melihat perubahan hasil. Contohnya, uji koordinat titik di sekitar kota ketika margin hilal kecil; atau bandingkan definisi fajar dengan sudut yang dinyatakan sumber. Hasil sensitivitas bukan ukuran ketidakpastian otomatis jika rentang dan distribusi tidak didasarkan pada data. Ia menunjukkan pengaruh perubahan parameter yang telah ditetapkan.

12.5 Cakupan geografis dan pencarian kontinu

Untuk menjadikan klaim saksi awal lebih kuat, pengembangan dapat memakai grid daratan yang terkontrol, penapisan poligon, dan pemurnian di sekitar batas memenuhi. Sampling harus mempertimbangkan daerah yang sunset-nya berada dekat tengah malam UTC dan lokasi dengan perubahan tinggi atau elongasi yang cepat terhadap posisi. Daftar kota tetap berguna sebagai panel ilustrasi, tetapi pencarian matematis tidak sebaiknya bergantung hanya pada kepadatan nama kota.

Algoritma kontinu juga memerlukan definisi “pertama”: pertama menurut sunset UTC, pertama saat kriteria tercapai sepanjang senja, atau titik yang memberi tanggal kalender paling awal. Ketiganya tidak selalu setara. Versi SOLUNA yang dibahas mengevaluasi pada sunset lokal dan memilih waktu saksi dalam sampel. Pengembangan baru perlu mempertahankan definisi itu atau menyatakan perubahan secara eksplisit.

12.6 Uji yang memberi nilai ilmiah

Himpunan uji masa mendatang dapat mencakup bulan dengan dua konjungsi Gregorian, pergantian tahun, nilai dekat 5° atau 8° , konjungsi tepat di sekitar fajar, sunset sekitar 00.00 UTC, lintang tinggi tanpa sunset, transisi waktu musim panas,

batas cakupan kernel, dan gangguan unduhan tahunan. Uji tersebut dipilih karena masing-masing dapat mengubah keputusan atau mengungkap asumsi yang berbeda, bukan sekadar memperbanyak jumlah eksekusi.

Uji ephemeris turunan perlu membandingkan nilai terhadap sumber pada interval acak dan batas rekaman, serta mengaudit cakupan seluruh segmen. Uji gerhana perlu membandingkan tanggal, jenis, saat maksimum, lokasi, dan kontak. Uji UI perlu memeriksa keterbacaan, scroll, penutupan sidebar, serta hasil ketika tahun diubah saat setup masih aktif. Himpunan uji seperti ini memberi bukti yang lebih lengkap daripada satu persentase kecocokan kalender.

12.7 Pembelajaran tanpa menyederhanakan berlebihan

Pendidik dapat menyusun demonstrasi berurutan dari satu konsep ke konsep berikutnya. Mulailah dengan posisi pada satu saat, lalu ubah lokasi untuk memperlihatkan horizon dan zona. Pertahankan UTC sambil mengubah mode untuk memperlihatkan paralaks. Setelah itu, bandingkan timestamp sunset dan aturan PKG. Urutan tersebut membuat hubungan antara teori dan antarmuka lebih mudah ditelusuri.

Aplikasi tidak perlu menampilkan seluruh rincian internal kepada setiap pengguna pada setiap langkah. Namun, definisi dan satuan harus tersedia ketika diperlukan. Buku ini menyediakan kedalaman metodologis sebagai pendamping, sehingga tampilan utama dapat tetap ringkas. Informasi akademik yang baik mengarahkan pembaca pada alasan perhitungan, bukan hanya nama pustaka atau jumlah digit.

12.8 Dokumentasi versi dan pemeliharaan

Setiap perubahan aturan global, definisi fajar, katalog lokasi, ephemeris, atau prosedur gerhana perlu dicatat sebagai perubahan perilaku yang dapat memengaruhi hasil. Perubahan warna dan tata letak umumnya tidak mengubah angka, tetapi tetap memerlukan pemeriksaan agar informasi tidak tersembunyi atau tabel kehilangan scroll. Pemeliharaan sebaiknya menghubungkan perubahan dengan uji yang relevan dan hasil perbandingan versi.

Buku ini menjelaskan versi yang tersedia pada 5 Oktober 2026. Setelah kode berkembang, dokumentasi dapat diperbarui menggunakan struktur dan identitas data yang sama. Prinsip utamanya ialah menjaga hubungan yang dapat ditelusuri antara masukan, metode, hasil, dan batas bukti.

LAMPIRAN A KAMUS DATA DAN SATUAN

A.1 Posisi Matahari dan Bulan

Kamus berikut mengikuti nama field dalam mesin sumber. Nama field dipertahankan agar pembaca dapat menelusuri data mesin. Tabel menjelaskan kerangka yang dominan, tetapi pengguna tetap perlu mencatat mode hasil dan timestamp. Kolom ekliptika dan sejumlah sudut orientasi selalu geosentris, sedangkan arah horizontal dan jarak pengamat mengikuti mode yang diterapkan pada perhitungan lokal.

Tabel 8. Field BodyResult dan satuannya

Field	Satuan	Definisi operasional
name	Teks	Identitas benda
altitude_deg	°	Tinggi geometris sesuai mode
refracted_altitude_deg	° / kosong	Tinggi dengan model refraksi standar jika tersedia
azimuth_deg	°	Azimut; utara 0°, timur 90°
hour_angle_hours	Jam	Sudut jam, normalisasi sekitar -12 sampai +12
ra_hours	Jam	RA apparent pada epoch saat hitung, sesuai pengamat
dec_deg	°	Deklinasi apparent pada epoch saat hitung
ecliptic_longitude_deg	°	Bujur ekliptika dari geosenter
ecliptic_latitude_deg	°	Lintang ekliptika dari geosenter
constellation	Teks	Identitas rasi menurut pemetaan yang dipakai
elongation_deg	°	Separasi apparent terhadap Matahari sesuai pengamat
illumination_pct	%	Fraksi terang geometris
phase_angle_deg	°	Sudut fase fisik pada benda
phase_cycle_deg	° / kosong	Selisih bujur ekliptika geosentris Bulan–Matahari

Field	Satuan	Definisi operasional
phase_name	Teks	Kategori fase; bukan waktu tepat peristiwa
waxing_waning	Teks	Kecenderungan membesar atau mengecil
bright_limb_position_angle_deg	° / kosong	Sudut posisi sisi terang dari geometri geosentris
magnitude	Magnitudo	Perkiraan model empiris pada kode
distance_km	km	Jarak geosentris
distance_au	AU	Jarak geosentris dalam satuan astronomi
observer_distance_km	km	Jarak dari pengamat yang dipakai
angular_diameter_arcmin	Menit busur	Diameter dari jari-jari model dan jarak pengamat
semidiameter_arcmin	Menit busur	Setengah diameter sudut
horizontal_parallax_deg	°	Paralaks horizontal dari jarak geosentris

A.2 Peristiwa dan durasi

Tabel 9. Field peristiwa dan durasi

Field atau kelompok	Satuan	Arti
sun_rise / sun_set	Waktu lokal	Terbit dan terbenam Matahari
moon_rise / moon_set	Waktu lokal	Terbit dan terbenam Bulan
sun_transit / moon_transit	Waktu lokal	Transit atas pada meridian
..._azimuth_deg	°	Azimut pada peristiwa rise atau set
..._transit_altitude_deg	°	Tinggi pada transit
civil_dawn / civil_dusk	Waktu lokal	Twilight sipil -6°

Field atau kelompok	Satuan	Arti
nautical_dawn / nautical_dusk	Waktu lokal	Twilight nautika -12°
astronomical_dawn / astronomical_dusk	Waktu lokal	Twilight astronomis -18°
day_length_hours	Jam	Durasi sunrise–sunset
equation_of_time_minutes	Menit	Persamaan waktu dengan tanda model kode
local_sidereal_time_hours	Jam	Waktu sidereal lokal saat hitung
twilight_state	Teks	Kategori keadaan langit saat hitung
conjunction_geocentric	Waktu lokal	Konjungsi geosentris terdekat yang ditemukan
conjunction_topocentric	Waktu lokal	Minimum separasi lokal dalam jendela pencarian
conjunction_topocentric_min_elong_deg	°	Separasi minimum pada grid yang diperiksa
moon_age_days	Hari	Selisih terhadap konjungsi sebelumnya
lag_time_minutes	Menit bertanda	Moonset terdekat dikurangi sunset

Nilai string waktu perlu dibaca dengan tanggal dan zona. Nilai kosong dapat berasal dari tidak tersedianya peristiwa atau kegagalan pencarian, bergantung pada fungsi. Durasi berbentuk angka tidak boleh diberi satuan detik jika field menyatakan jam atau menit. Untuk replikasi yang teliti, data mentah harus dipertahankan sebelum diubah menjadi tampilan pendek.

A.3 Audit keputusan global

Tabel 10. Field audit global dan saksi

Field atau kelompok	Jenis	Arti
hijri_start_gregorian_date	Tanggal	Calon awal bulan yang diterima
h_day	Bilangan / kosong	Hari uji diterima sejak H0 UTC

Field atau kelompok	Jenis	Arti
utc_reference_date	Tanggal UTC	Tanggal konjungsi yang menjadi H0
utc_midnight	Timestam p	Batas B pada D + 1
conjunction_local_datetime	Timestam p	Konjungsi geosentris dalam zona acuan
gisborne_dawn_utc_datetime	Timestam p	Fajar Gisborne -18° pada calon awal bulan
conjunction_before_gisborne_dawn	Boolean / kosong	Hasil pembandingan $C < F$
decision_pkg / decision_reason	Teks	Cabang PKG dan alasan
first_match	Record / kosong	Saksi yang dipilih dalam sampel dievaluasi
cities	Daftar record	100 kota untuk masing-masing H0–H2
validation_sites	Daftar record	Titik tambahan yang dievaluasi bersyarat
daily_decisions	Daftar audit	Jumlah dan status per hari uji
matched_count / total_count	Bilangan	Ringkasan pada himpunan yang dilaporkan
pkg_status / pkg_accepted	Teks / boolean	Status serta penerimaan saksi
america_mainland	Boolean	Klasifikasi katalog bagi PKG 2

Jumlah lokasi yang melewati kriteria numerik tidak selalu sama dengan jumlah saksi diterima PKG. Lokasi dapat melewati 5° – 8° tetapi berada pada cabang waktu yang tidak diterima. Daftar tambahan dipisahkan dari tabel utama dan dievaluasi bersyarat. Laporan sebaiknya menyebut himpunan yang digunakan ketika menyajikan jumlah atau saksi paling awal.

LAMPIRAN B KATALOG TITIK PEMINDAIAN

B.1 Seratus titik utama

Katalog diambil langsung dari daftar GLOBAL_HIJRI_SCAN_CITIES pada mesin yang diaudit. Urutan adalah urutan sumber, bukan urutan hasil menurut timestamp. Lintang dan bujur dinyatakan dalam derajat desimal; tanda negatif masing-masing berarti selatan dan barat. Elevasi dinyatakan dalam meter dan identitas zona mengikuti sumber. Angka dalam tabel dibulatkan untuk keterbacaan; audit presisi menggunakan nilai asli pada kode sumber.

Tabel 11. Seratus titik utama sesuai urutan sumber

No.	Titik	Lintang Bujur (°)	Elev. (m)	Zona IANA
1	Gisborne, New Zealand	-38,653333 178,004167	15,0	Pacific/Auckland
2	Wellington, New Zealand	-41,286643 174,775572	13,0	Pacific/Auckland
3	Auckland, New Zealand	-36,848530 174,763490	43,0	Pacific/Auckland
4	Lord Howe Island, Australia	-31,550000 159,080000	5,0	Australia/Lord_Howe
5	Byron Bay, Australia	-28,649890 153,612463	6,0	Australia/Sydney
6	Sydney, Australia	-33,867850 151,207323	58,0	Australia/Sydney
7	Hobart, Australia	-42,879361 147,329407	20,0	Australia/Hobart
8	Port Moresby, Papua New Guinea	-9,443800 147,180300	35,0	Pacific/Port_Moresby
9	Jayapura, Indonesia	-2,562626 140,692633	95,0	Asia/Jayapura
10	Ambon, Indonesia	-3,694324 128,181303	11,0	Asia/Jayapura
11	Makassar, Indonesia	-5,133216 119,407837	6,0	Asia/Makassar

No.	Titik	Lintang Bujur (°)	Elev. (m)	Zona IANA
12	Denpasar, Indonesia	-8,654459 115,216533	32,0	Asia/Makassar
13	Pontianak, Indonesia	-0,022086 109,338104	5,0	Asia/Jakarta
14	Jakarta, Indonesia	-6,177802 106,828444	8,0	Asia/Jakarta
15	Singapore	1,352100 103,819800	15,0	Asia/Singapore
16	Kuala Lumpur, Malaysia	3,139000 101,686900	22,0	Asia/Kuala_Lumpur
17	Bangkok, Thailand	13,756300 100,501800	5,0	Asia/Bangkok
18	Banda Aceh, Indonesia	5,550050 95,317827	11,0	Asia/Jakarta
19	Dhaka, Bangladesh	23,810300 90,412500	4,0	Asia/Dhaka
20	Karachi, Pakistan	24,860700 67,001100	8,0	Asia/Karachi
21	Muscat, Oman	23,588000 58,382900	15,0	Asia/Muscat
22	Riyadh, Saudi Arabia	24,687732 46,721849	612,0	Asia/Riyadh
23	Makkah, Saudi Arabia	21,426640 39,825632	277,0	Asia/Riyadh
24	Madinah, Saudi Arabia	24,468611 39,614167	625,0	Asia/Riyadh
25	Cairo, Egypt	30,062631 31,249666	23,0	Africa/Cairo
26	Istanbul, Turkey	41,013843 28,949661	40,0	Europe/Istanbul
27	Rome, Italy	41,902800 12,496400	21,0	Europe/Rome

No.	Titik	Lintang Bujur (°)	Elev. (m)	Zona IANA
28	Lisbon, Portugal	38,722300 -9,139300	2,0	Europe/Lisbon
29	Dakar, Senegal	14,716700 -17,467700	22,0	Africa/Dakar
30	Recife, Brazil	-8,047600 -34,877000	10,0	America/Recife
31	Paramaribo, Suriname	5,852000 -55,203800	3,0	America/Paramaribo
32	New York, USA	40,712800 -74,006000	10,0	America/New_York
33	Chicago, USA	41,878100 -87,629800	181,0	America/Chicago
34	Los Angeles, USA	34,052200 -118,243700	71,0	America/Los_Angeles
35	San Francisco, USA	37,774900 -122,419400	16,0	America/Los_Angeles
36	Ketchikan, Alaska, USA	55,357145 -131,674552	24,0	America/Sitka
37	Juneau, Alaska, USA	58,301944 -134,419722	17,0	America/Juneau
38	Skagway, Alaska, USA	59,464431 -135,275720	9,0	America/Juneau
39	Sitka, Alaska, USA	57,193204 -135,367396	8,0	America/Sitka
40	Haines, Alaska, USA	59,235800 -135,445000	11,0	America/Juneau
41	Yakutat, Alaska, USA	59,563588 -139,606191	9,0	America/Yakutat
42	Kaktovik, Alaska, USA	70,121452 -143,629435	5,0	America/Anchorage
43	Fort Yukon, Alaska, USA	66,588406 -145,227642	132,0	America/Anchorage

No.	Titik	Lintang Bujur (°)	Elev. (m)	Zona IANA
44	Delta Junction, Alaska, USA	64,037800 -145,732200	354,0	America/Anchorage
45	Cordova, Alaska, USA	60,542400 -145,752500	18,0	America/Anchorage
46	Valdez, Alaska, USA	61,083621 -146,317179	30,0	America/Anchorage
47	Fairbanks, Alaska, USA	64,836531 -147,651745	136,0	America/Anchorage
48	Whittier, Alaska, USA	60,777318 -148,663713	13,0	America/Anchorage
49	Palmer, Alaska, USA	61,597166 -149,114683	71,0	America/Anchorage
50	Seward, Alaska, USA	60,124786 -149,391581	27,0	America/Anchorage
51	Wasilla, Alaska, USA	61,581136 -149,460790	104,0	America/Anchorage
52	Anchorage, Alaska, USA	61,218056 -149,900278	31,0	America/Anchorage
53	Soldotna, Alaska, USA	60,485318 -151,069014	32,0	America/Anchorage
54	Kenai, Alaska, USA	60,562240 -151,208060	22,0	America/Anchorage
55	Homer, Alaska, USA	59,642500 -151,548300	29,0	America/Anchorage
56	Seldovia, Alaska, USA	59,439480 -151,710387	9,0	America/Anchorage
57	Kodiak, Alaska, USA	57,793001 -152,397481	15,0	America/Anchorage
58	Ouzinkie, Alaska, USA	57,922749 -152,469678	4,0	America/Anchorage
59	Port Lions, Alaska, USA	57,781940 -152,843864	6,0	America/Anchorage

No.	Titik	Lintang Bujur (°)	Elev. (m)	Zona IANA
60	Old Harbor, Alaska, USA	57,211542 -153,326621	4,0	America/Anchorage
61	King Salmon, Alaska, USA	58,735227 -156,528544	17,0	America/Anchorage
62	Utqiagvik, Alaska, USA	71,253861 -156,798246	3,0	America/Anchorage
63	South Naknek, Alaska, USA	58,663408 -156,991671	5,0	America/Anchorage
64	Naknek, Alaska, USA	58,728300 -157,013300	10,0	America/Anchorage
65	Egegik, Alaska, USA	58,205148 -157,521978	8,0	America/Anchorage
66	Pilot Point, Alaska, USA	57,579918 -157,765733	5,0	America/Anchorage
67	Chignik, Alaska, USA	56,295180 -158,391784	15,0	America/Anchorage
68	Chignik Lagoon, Alaska, USA	56,304267 -158,500259	15,0	America/Anchorage
69	Dillingham, Alaska, USA	59,060467 -158,532251	29,0	America/Anchorage
70	Port Heiden, Alaska, USA	56,942514 -158,604854	29,0	America/Anchorage
71	Aleknagik, Alaska, USA	59,288961 -158,664343	11,0	America/Anchorage
72	Chignik Lake, Alaska, USA	56,273221 -158,793376	15,0	America/Anchorage
73	Perryville, Alaska, USA	55,932900 -159,155803	3,0	America/Anchorage
74	Ivanof Bay, Alaska, USA	55,918199 -159,493298	5,0	America/Anchorage
75	Wainwright, Alaska, USA	70,658232 -159,985214	12,0	America/Anchorage

No.	Titik	Lintang Bujur (°)	Elev. (m)	Zona IANA
76	Togiak, Alaska, USA	59,061900 -160,376100	5,0	America/Anchorage
77	Sand Point, Alaska, USA	55,313470 -160,484344	7,0	America/Anchorage
78	Unalakleet, Alaska, USA	63,894206 -160,796775	6,0	America/Nome
79	Koyuk, Alaska, USA	64,939700 -161,154900	10,0	America/Nome
80	Shaktoolik, Alaska, USA	64,354481 -161,193514	7,0	America/Nome
81	Bethel, Alaska, USA	60,793905 -161,796322	3,0	America/Anchorage
82	Elim, Alaska, USA	64,618226 -162,276560	11,0	America/Nome
83	King Cove, Alaska, USA	55,084993 -162,316806	27,0	America/Anchorage
84	Kotzebue, Alaska, USA	66,898300 -162,596700	6,0	America/Nome
85	Cold Bay, Alaska, USA	55,206568 -162,714987	30,0	America/Anchorage
86	False Pass, Alaska, USA	54,872441 -163,334268	6,0	America/Anchorage
87	White Mountain, Alaska, USA	64,682359 -163,412753	17,0	America/Nome
88	Nome, Alaska, USA	64,501111 -165,406389	6,0	America/Nome
89	Akutan, Alaska, USA	54,139979 -165,763026	20,0	America/Anchorage
90	Brevig Mission, Alaska, USA	65,343196 -166,508946	10,0	America/Nome
91	Unalaska, Alaska, USA	53,910255 -166,529029	4,0	America/Anchorage

No.	Titik	Lintang Bujur (°)	Elev. (m)	Zona IANA
92	Point Hope, Alaska, USA	68,344554 -166,674485	5,0	America/Anchorage
93	Wales, Alaska, USA	65,613232 -168,068313	8,0	America/Nome
94	Diomedede, Alaska, USA	65,753924 -168,922577	28,0	America/Nome
95	Saint George, Alaska, USA	56,589923 -169,628111	10,0	America/Nome
96	Saint Paul, Alaska, USA	57,181744 -170,275129	9,0	America/Nome
97	Savoonga, Alaska, USA	63,679706 -170,483324	14,0	America/Nome
98	Gambell, Alaska, USA	63,779700 -171,733300	8,0	America/Nome
99	Atka, Alaska, USA	52,211546 -174,213452	18,0	America/Adak
100	Adak, Alaska, USA	51,909889 -176,598067	7,0	America/Adak

B.2 Tiga puluh titik tambahan Amerika

Titik berikut dievaluasi ketika PKG 1 belum ditemukan pada daftar utama di hari uji yang relevan. Kehadiran titik pada lampiran tidak berarti titik itu selalu dihitung pada setiap keputusan. Katalog ini merupakan sampel daratan yang ditetapkan program, bukan inventaris seluruh kota atau seluruh daratan Amerika.

Tabel 12. Tiga puluh titik tambahan Amerika

No.	Titik	Lintang Bujur (°)	Elev. (m)	Zona IANA
1	Tijuana, Mexico	32,514900 -117,038200	20,0	America/Tijuana
2	Hermosillo, Mexico	29,072967 -110,955919	210,0	America/Hermosillo

No.	Titik	Lintang Bujur (°)	Elev. (m)	Zona IANA
3	Mazatlan, Mexico	23,249400 -106,411100	10,0	America/Mazatlan
4	Manzanillo, Mexico	19,113800 -104,338500	10,0	America/Mexico_City
5	Acapulco, Mexico	16,853100 -99,823700	20,0	America/Mexico_City
6	San Salvador, El Salvador	13,692900 -89,218200	658,0	America/El_Salvador
7	Liberia, Costa Rica	10,634600 -85,440700	144,0	America/Costa_Rica
8	Panama City, Panama	8,982400 -79,519900	10,0	America/Panama
9	Esmeraldas, Ecuador	0,959200 -79,653970	15,0	America/Guayaquil
10	Manta, Ecuador	-0,967650 -80,708900	20,0	America/Guayaquil
11	Guayaquil, Ecuador	-2,170998 -79,922359	4,0	America/Guayaquil
12	Quito, Ecuador	-0,180700 -78,467800	2850,0	America/Guayaquil
13	Trujillo, Peru	-8,111600 -79,028800	34,0	America/Lima
14	Lima, Peru	-12,046400 -77,042800	154,0	America/Lima
15	Arequipa, Peru	-16,409000 -71,537500	2335,0	America/Lima
16	Arica, Chile	-18,478300 -70,312600	20,0	America/Santiago
17	Iquique, Chile	-20,230700 -70,135700	20,0	America/Santiago
18	Antofagasta, Chile	-23,650900 -70,397500	40,0	America/Santiago

No.	Titik	Lintang Bujur (°)	Elev. (m)	Zona IANA
19	La Serena, Chile	-29,902700 -71,251900	28,0	America/Santiago
20	Valparaiso, Chile	-33,047200 -71,612700	20,0	America/Santiago
21	Santiago, Chile	-33,448900 -70,669300	520,0	America/Santiago
22	Concepcion, Chile	-36,820100 -73,044400	12,0	America/Santiago
23	Valdivia, Chile	-39,814200 -73,245900	5,0	America/Santiago
24	Puerto Montt, Chile	-41,469300 -72,942400	14,0	America/Santiago
25	Coyhaique, Chile	-45,571200 -72,068300	300,0	America/Santiago
26	Punta Arenas, Chile	-53,163800 -70,917100	30,0	America/Punta_Arenas
27	Neuquen, Argentina	-38,951600 -68,059100	260,0	America/Argentina/Salta
28	Bariloche, Argentina	-41,133500 -71,310300	893,0	America/Argentina/Salta
29	Comodoro Rivadavia, Argentina	-45,864100 -67,496600	60,0	America/Argentina/Catamarca
30	Rio Gallegos, Argentina	-51,623000 -69,216800	20,0	America/Argentina/Rio_Gallegos

B.3 Pembacaan geografis dan pemeliharaan

Nama lokasi beserta zona waktu tidak menjamin klasifikasi pulau atau daratan utama tanpa pemeriksaan. Kode memakai daftar pengecualian untuk beberapa titik Amerika yang berada di pulau. Penambahan lokasi perlu memeriksa koordinat, elevasi, zona, dan klasifikasi PKG 2. Titik yang berada dekat garis pantai atau batas zona membutuhkan dokumentasi asal koordinat agar perubahan dapat dilacak.

Pembulatan koordinat pada publikasi tidak boleh dipakai mengganti nilai sumber tanpa pengujian. Lampiran dimaksudkan untuk keterbacaan manusia. Identitas sumber pada Lampiran D memungkinkan pembaca memperoleh definisi persis versi yang dibahas dan memeriksa apakah katalog telah berubah pada versi berikutnya.

LAMPIRAN C PENYIAPAN DAN PEMECAHAN MASALAH

C.1 Menyediakan aset yang sesuai

Distribusi memerlukan halaman `index.php` beserta berkas BSP yang namanya sesuai konfigurasi dan aset runtime yang dapat dimuat. Berkas tahunan 2026–2030 sebaiknya tersedia pada lokasi relatif yang diminta pemilih. Tiga berkas rentang menyediakan cakupan nominal bagi tahun lain atau fallback berkas tahunan yang memang tidak ditemukan. Berkas asli `de440s.bsp` tidak menjadi jalur akses pada versi ini.

Server perlu mengirim byte BSP yang sebenarnya, bukan halaman galat dengan status berhasil. Ukuran dan pengenalan format akan diperiksa oleh aplikasi. Aset harus dapat diakses sesuai aturan origin dan jaringan browser. Karena komputasi dilakukan di browser, menyediakan server PHP saja belum memastikan paket Python dan kernel telah siap pada perangkat pengguna.

C.2 Setup lambat atau berkas tidak ditemukan

Pisahkan waktu unduhan paket, unduhan BSP, dan perhitungan. Pada kunjungan pertama, dependensi bisa menjadi bagian terbesar penyiapan meskipun BSP tahunan kecil. Periksa tahun perangkat dan tahun masukan sebelum menyimpulkan pemilih memakai berkas yang salah. Untuk tahun di luar 2026–2030, penggunaan BSP rentang memang sesuai konfigurasi.

Jika tahunan mengembalikan HTTP 404 atau 410, aplikasi mencoba satu rentang yang sesuai. Jika terjadi HTTP 500, ukuran salah, atau format tidak valid, cari penyebab distribusi atau server; jangan menganggap fallback besar pasti akan dijalankan. Mengunggah BSP baru dengan nama yang sama dapat menyisakan cache lama, sehingga prosedur pembaruan dan pemeriksaan ukuran perlu diselaraskan.

C.3 Tabel kosong atau peristiwa hilang

Periksa apakah hitung telah selesai dan apakah ruang kanan dapat discroll. Informasi yang berada di bawah layar berbeda dari informasi yang tidak dihitung. Setelah itu, periksa cakupan tahun, mode waktu, koordinat, serta kemungkinan tidak adanya crossing pada lintang tinggi. Untuk fenomena masa depan, pastikan seluruh jendela pencarian berada dalam cakupan kernel.

Apabila tabel gerhana atau fenomena kosong pada kasus yang diketahui dari referensi, lakukan perbandingan timestamp dan jendela, lalu audit fungsi yang menangani exception. Sel kosong belum cukup untuk menentukan penyebab.

Dokumentasi laporan harus menyatakan bahwa hasil tidak tersedia, bukan mengubahnya menjadi pernyataan pasti bahwa peristiwa tidak terjadi.

C.4 Ketidaksesuaian angka atau tanggal

Mulai dari identitas versi dan masukan, kemudian periksa kerangka, refraksi, definisi sunset, dan UTC. Untuk kalender global, lanjutkan ke H0, B, F, cabang PKG, serta katalog saksi. Untuk perbedaan nama bulan, periksa mekanisme penamaan dan titik acuan secara terpisah dari tanggal Gregorian hasil scan. Urutan ini membantu mempersempit penyebab tanpa mengubah kode lain yang belum terbukti bermasalah.

Ekspor CSV dapat membantu membandingkan ringkasan posisi dan peristiwa, tetapi audit global membutuhkan rekaman tambahan. Simpan sumber sebelum perubahan jika pengembangan dilakukan. Buku ini tidak mengubah program atau memberikan patch; petunjuk merupakan penjelasan operasional atas versi yang diaudit.

LAMPIRAN D IDENTITAS DAN PROTOKOL REPRODUKSI

D.1 Identitas sumber dan lingkungan

Tabel 13. Identitas bahan dan lingkungan pengujian

Komponen	Nilai
Tanggal audit	5 Oktober 2026
Sumber utama	https://kasmui.cloud/soluna/
SHA-256	756e075c86fe1ed32157d8778378be71fd6089622a42082b665bfca57445113d
SHA-256 mesin	36313889c728437ecb3d972d085d7cfc3e40f2ec78f069a989b711f7b39da97e
SHA-256 PDF	0c02c438e532e91ce558aa5e01051cb69a4ef8730d54f4e4d68b0b1162fd1feb
Python / platform	3.12.14 / Linux pada pengujian native
Skyfield / jplephem	1.55 / 2.24
NumPy / tzdata	2.5.3 / 2025b
BSP uji	de440s_2026.bsp dan de440s_2027.bsp
Jumlah scan	36; dua belas dasar, dua belas ulang, dua belas varian
SHA-256 de440s_2026.bsp	36768a269ec18563a005d017a40da09bdfef1f4689e279a3006625110f621a934
SHA-256 de440s_2027.bsp	992a695355ed1de811eaac4fb59e49382be84859a2729402a26bff79b582823c

Hash mengidentifikasi byte sumber, bukan mengukur kualitas astronomi. Sumber yang identik memungkinkan hasil uji sebelumnya digunakan untuk versi yang sama, dengan catatan bahwa berkas data dan lingkungan juga harus dicatat. Jika satu karakter kode berubah, identitas baru perlu dicatat meskipun perubahan itu hanya tata letak.

D.2 Protokol pembandingan bulanan

Gunakan dua belas bulan Gregorian yang memuat konjungsi referensi untuk awal Muharam sampai Zulhijah 1448 H. Jalankan satu scan dasar, ulangi dengan masukan identik, lalu jalankan varian yang ditetapkan pada studi. Simpan seluruh

keluaran global dan bandingkan tanggal, hari H, cabang PKG, saksi, serta nilai astronomi. Pisahkan metadata yang sengaja berubah pada varian sebelum menguji invariansi.

Untuk titik PDF, masukkan koordinat referensi dan asumsi elevasi yang dinyatakan. Hitung sunset ulang dan posisi geosentris pada timestamp itu. Hitung pula posisi pada timestamp tercetak untuk memeriksa konsistensi angka sumber. Laporkan selisih waktu, sudut, dan status secara terpisah. Jangan memakai waktu yang telah dikoreksi diam-diam tanpa mencatat perubahan terhadap sumber pembandingan.

D.3 Uji batas dan akses data

Uji operator pada tinggi tepat 5° , elongasi tepat 8° , sunset tepat B, konjungsi tepat F, serta nilai di masing-masing sisi batas. Uji klasifikasi daratan utama Amerika dan penolakan pulau pada PKG 2. Catat pembukaan kernel pada setup atau pergantian tahun dan pastikan scan kota memakai objek aktif yang sama. Pengujian jaringan dan fallback harus dilakukan secara tersendiri apabila distribusi web turut dinilai.

Waktu eksekusi dicatat untuk tahap yang jelas. Benchmark native pada bab validasi mengukur scan, sedangkan benchmark browser harus memisahkan setup runtime, transfer kernel, hitung, dan render. Melaporkan satu waktu gabungan tanpa kondisi cache dapat menyulitkan pembandingan atau diagnosis performa.

LAMPIRAN E SOURCE CODE SOLUNA

```
<!doctype html>
<html lang="id">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-
width,initial-scale=1,viewport-fit=cover">
<meta name="description" content="SOLUNA - Astronomi
Matahari & Bulan, web statis berbasis Skyfield/JPL
DE440s yang berjalan di browser.">
<title>SOLUNA - Astronomi Matahari & Bulan</title>
<link rel="icon" type="image/png"
href="https://kasmui.cloud/kasmui-penulis.png">
<link rel="preconnect" href="https://cdn.jsdelivr.net"
crossorigin>
<script
src="https://cdn.jsdelivr.net/pyodide/v0.27.7/full/pyod
ide.js"></script>
<style>
:root{--bg:#07111f;--side:#0c1729;--panel:#101e34;--
panel2:#0d1829;--card:#14253e;--line:#253c5d;--
text:#eef6ff;--muted:#9fb2ca;--blue:#2f80ed;--
cyan:#16b8c4;--purple:#8b5cf6;--orange:#f59e0b;--
green:#16a34a;--red:#dc2626;--shadow:0 18px 55px
rgba(0,0,0,.24);--radius:16px}
*{box-sizing:border-box}html{scroll-
behavior:smooth}body{margin:0;background:linear-
gradient(180deg,#07111f,#091424
55%,#07111f);color:var(--text);font-family:"Segoe
UI",Inter,Arial,sans-serif;font-size:15px;line-
height:1.48}button,input,select{font:inherit}.app{min-
height:100vh;display:grid;grid-template-columns:340px
minmax(0,1fr);transition:grid-template-columns .22s
ease}.app.sidebar-hidden{grid-template-columns:0
minmax(0,1fr)}.sidebar{position:sticky;top:0;height:100
vh;width:340px;max-
width:100%;overflow:auto;background:linear-
gradient(180deg,#0c1729,#091322);border-right:1px solid
#18304e;padding:18px 16px;z-
index:20;transition:transform .22s ease,opacity .22s
ease;will-change:transform}.app.sidebar-hidden
.sidebar{transform:translateX(-103%);opacity:0:pointer-
events:none}.brand{display:flex;gap:12px;align-
items:center;margin-
bottom:12px}.brandmark{width:52px;height:52px;border-
radius:16px;background:linear-
```

```

gradient(135deg,#2f80ed,#8b5cf6);display:grid;place-
items:center;font-size:28px;font-weight:900;box-
shadow:0 12px 30px rgba(47,128,237,.28)}.brand h1{font-
size:18px;margin:0;line-height:1.12}.brand
small{color:var(--muted)}.side-title{font-
size:12px;letter-spacing:.12em;color:#7fa0c5;font-
weight:800;margin:17px 2px 7px}.group{margin:9px
0}.group label{display:block;font-size:12px;font-
weight:700;color:#b9c9db;margin:0 0
5px}.input,.select{width:100%;background:#0a1423;color:
#fff;border:1px solid #294564;border-
radius:10px;padding:9px
10px;outline:none}.input:focus,.select:focus{border-
color:#4d98f5;box-shadow:0 0 0 3px
rgba(47,128,237,.15)}.grid2{display:grid;grid-template-
columns:1fr 1fr;gap:8px}.seg{display:grid;grid-
template-columns:1fr 1fr;background:#08111e;border:1px
solid #294564;border-
radius:11px;padding:3px;gap:3px}.seg
button{border:0;border-
radius:8px;background:transparent;color:#a8bbd1;padding
:8px 6px;cursor:pointer;font-weight:700;font-
size:12px}.seg
button.active{background:#2f80ed;color:white}.btn{width
:100%;border:0;border-radius:11px;padding:11px
13px;color:#fff;background:#2f80ed;font-
weight:800;cursor:pointer;margin-
top:8px}.btn:hover{filter:brightness(1.06)}.btn.seconda
ry{background:#1d334f}.btn.green{background:#15803d}.bt
n.orange{background:#b45309}.hint{font-
size:11px;color:#839ab5;margin-top:5px}.main{min-
width:0}.topbar{position:sticky;top:0;z-
index:15;height:72px;padding:0 22px 0
76px;display:flex;align-
items:center;gap:12px;background:rgba(7,17,31,.88);back-
drop-filter:blur(14px);border-bottom:1px solid
#17304e}.menubtn{display:block;position:fixed;top:16px;
left:352px;z-
index:60;border:0;background:#17304e;color:#fff;width:4
2px;height:40px;border-radius:10px;font-
size:20px;cursor:pointer;box-shadow:0 8px 24px
rgba(0,0,0,.28);transition:left .22s ease,filter .15s
ease}.menubtn:hover{filter:brightness(1.12)}.app.sideba
r-hidden .menubtn{left:12px}.top-title{font-
weight:900;font-size:21px}.engine-badge{margin-
left:auto;max-width:60%;overflow:hidden;text-
overflow:ellipsis;white-space:nowrap;border-
radius:999px;padding:7px
12px;background:#10233c;border:1px solid

```

```

#284a70;color:#a9c9ec;font-size:12px}.content{max-
width:1500px;margin:0 auto;padding:20px 22px
42px}.hero{background:radial-gradient(circle at 84%
18%,rgba(139,92,246,.35),transparent 26%),linear-
gradient(130deg,#0f2a4c,#12213a 55%,#0f1830);border:1px
solid #274a70;border-radius:22px;padding:26px 28px;box-
shadow:var(--shadow)}.hero h2{font-
size:clamp(25px,3vw,39px);margin:0 0 5px}.hero
p{margin:0;color:#c0cfe0;max-
width:1000px}.statusline{margin-top:14px;padding:10px
12px;border-radius:11px;background:#0a1728;border:1px
solid #263f5f;color:#a9bfd7;font-
size:13px}.metrics{display:grid;grid-template-
columns:repeat(4,1fr);gap:12px;margin:14px
0}.metric{background:var(--card);border:1px solid var(--
line);border-radius:15px;padding:16px;text-
align:center}.metric strong{display:block;font-
size:28px;line-height:1.12}.metric span{font-
size:12px;color:var(--muted)}.metric:nth-child(1)
strong{color:#62a7ff}.metric:nth-child(2)
strong{color:#c084fc}.metric:nth-child(3)
strong{color:#43d6df}.metric:nth-child(4)
strong{color:#fbbf24}.cards{display:grid;grid-template-
columns:repeat(3,1fr);gap:12px}.card{background:var(--
panel);border:1px solid var(--line);border-
radius:16px;padding:16px;box-shadow:0 10px 30px
rgba(0,0,0,.12)}.card h3{margin:0 0 8px}.card
p{color:#b8c7d8}.tabs{display:flex;flex-
wrap:wrap;gap:6px;margin:16px 0 10px}.tabbtn{border:1px
solid #294867;background:#10213a;color:#b9c9db;border-
radius:10px;padding:8px 12px;font-
weight:800;cursor:pointer}.tabbtn.active{background:#2f
80ed;color:#fff;border-
color:#2f80ed}.tabpane{display:none}.tabpane.active{dis-
play:block}.table-wrap{overflow:auto;border:1px solid
var(--line);border-radius:13px;background:#091421;max-
height:680px}.table-wrap table{border-
collapse:collapse;width:100%;min-width:760px}.table-
wrap th,.table-wrap td{padding:9px 11px;border-
bottom:1px solid #1d314b;border-right:1px solid
#142a45;text-align:left;vertical-align:top}.table-wrap
th{position:sticky;top:0;background:#10233b;color:#d9ea
ff;z-index:1}.table-wrap tbody tr:nth-
child(even){background:#0b192a}.muted{color:var(--
muted)}.ok{color:#7ce59a}.warn{color:#ffd27a}.err{color
:#ff8585}.doc h3{color:#78b7ff;margin-top:18px}.doc
p,.doc li{color:#c3d1df}.doc
code{background:#07111f;padding:2px 5px;border-
radius:5px}.downloadbar{max-width:1500px;margin:12px

```

```

auto 0;padding:0 22px}.downloadbar a{display:inline-
flex;align-
items:center;gap:8px;color:#fff;background:#15803d;bord-
er:1px solid #22a34a;border-radius:10px;padding:9px
14px;font-weight:800;text-decoration:none}.downloadbar
a:hover{filter:brightness(1.08)}.footer{text-
align:center;color:#7189a4;padding:28px 8px 3px;font-
size:12px}.loading{position:fixed;inset:0;background:rg-
ba(3,9,17,.72);backdrop-filter:blur(5px);z-
index:100;display:none;align-items:center;justify-
content:center}.loading.show{display:flex}.loadbox{widt-
h:min(560px,92vw);background:#0d1b2e;border:1px solid
#2d4e73;border-radius:18px;padding:22px;box-shadow:0
30px 90px rgba(0,0,0,.42)}.loadbox h3{margin-
top:0}.calc-
spinner{display:none;width:52px;height:52px;margin:4px
auto 18px;border:5px solid
rgba(255,255,255,.16);border-top-color:#62a7ff;border-
right-color:#8b5cf6;border-radius:50%;animation:soluna-
spin .8s linear infinite;will-
change:transform}.loading.calculating .calc-
spinner{display:block}@keyframes soluna-
spin{to{transform:rotate(360deg)}}.progress{height:9px;
border-
radius:99px;background:#07111f;overflow:hidden}.bar{hei-
ght:100%;width:10%;background:linear-
gradient(90deg,#2f80ed,#8b5cf6);transition:.25s}.toast{
position:fixed;left:50%;bottom:20px;transform:translate
(-50%,18px);opacity:0:pointer-
events:none;background:#07111f;border:1px solid
#355577;padding:10px 15px;border-
radius:10px;transition:.2s;z-
index:120}.toast.show{opacity:1;transform:translate(-
50%,0)}
.global-hijri{margin-top:14px}.global-hijri h3{margin:0
0 6px;font-size:14px}.global-route{margin-
top:5px;color:#7f9ab8;font-size:11px}.global-hijri
.table-wrap{margin-top:10px;max-height:330px}.global-
status-ok{color:#7ce59a;font-weight:800}.global-status-
no{color:#ffd27a;font-weight:800}.global-status-
off{color:#9fb2ca;font-weight:800}.terms-pre{white-
space:pre-wrap;word-
break:normal;margin:0;color:#c3d1df;font:inherit;line-
height:1.62}.time-disabled{opacity:.72}
.global-hijri .global-hijri-date{margin-
top:12px;padding:12px 14px;border:2px solid
#fbbbf24;border-left:6px solid #b45309;border-
radius:12px;background:#fef3c7;color:#422006;font-

```

```

size:clamp(16px,2vw,21px);font-weight:900;line-
height:1.45;overflow-wrap:anywhere}
@media(max-width:1100px){.app,.app.sidebar-hidden{grid-
template-
columns:1fr}.sidebar{position:fixed;left:0;top:0;width:
340px;max-
width:90vw;transform:translateX(0);opacity:1;pointer-
events:auto;transition:transform .22s ease,opacity .22s
ease;box-shadow:18px 0 48px
rgba(0,0,0,.35)}.app.sidebar-hidden
.sidebar{transform:translateX(-103%);opacity:0;pointer-
events:none}.menubtn{display:block;left:calc(min(340px,
90vw) + 6px)}.app.sidebar-hidden
.menubtn{left:10px}.metrics{grid-template-
columns:repeat(2,1fr)}.cards{grid-template-
columns:1fr}.engine-badge{max-width:48%}}
@media(max-width:650px){.content{padding:14px 11px
32px}.topbar{padding:0 10px 0
58px;height:64px}.menubtn{top:13px;left:calc(min(318px,
90vw) + 2px);width:38px;height:38px;font-
size:19px}.app.sidebar-hidden .menubtn{left:10px}.top-
title{font-size:16px}.engine-
badge{display:none}.hero{padding:21px
18px}.metrics{grid-template-columns:1fr 1fr}.metric
strong{font-size:22px}.grid2{grid-template-
columns:1fr}.sidebar{width:318px}}
.global-hijri .global-hijri-date {
    text-align: center;
}
</style>
<style id="soluna-education-styles">
/* SOLUNA: tambahan navigasi dan dokumen pembelajaran.
*/
.soluna-education-shell{min-
height:0;height:100vh;height:100dvh;overflow-
x:hidden;overflow-y:auto;-webkit-overflow-
scrolling:touch}
.soluna-edu-nav{display:grid;grid-template-
columns:1fr;gap:7px;margin:16px 0 20px}
.soluna-edu-nav button{display:flex;align-
items:center;gap:11px;width:100%;padding:11px
12px;border:1px solid #35506f;border-
radius:11px;background:#12233a;color:#eaf3ff;text-
align:left;cursor:pointer;font-size:13px;font-
weight:800;letter-spacing:.035em}
.soluna-edu-nav button:hover{background:#203a59;border-
color:#88add8}
.soluna-edu-nav button[aria-
current="page"]{background:linear-

```

```

gradient(110deg,#174c85,#343576);border-
color:#9abaff;box-shadow:inset 3px 0 #7cf0dc}
.soluna-edu-nav-icon{display:grid;place-
items:center;width:29px;height:29px;border-
radius:8px;background:#e0efff;color:#164978;flex-
shrink:0;font-size:16px}
.soluna-edu-nav button:nth-child(2) .soluna-edu-nav-
icon{background:#d8f7ef;color:#075f54}
.soluna-edu-nav button:nth-child(3) .soluna-edu-nav-
icon{background:#fff0ce;color:#804e0c}
#solunaEducationView[hidden],#solunaCalculationView[hid-
den],.soluna-edu-page[hidden]{display:none!important}
.soluna-edu-page{--edu-accent:#245eae;--edu-
tint:#edf4ff;--edu-
border:#c8daf2;color:#263c53;background:#f8fafc;border:
1px solid #c7d9ed;border-
radius:19px;overflow:hidden;box-shadow:0 15px 44px
#0003;line-height:1.75;overflow-wrap:anywhere}
.soluna-edu-page[data-soluna-page="skyfield">{--edu-
accent:#087168;--edu-tint:#e9f7f4;--edu-border:#b5dfd5}
.soluna-edu-page[data-soluna-page="de440s">{--edu-
accent:#885213;--edu-tint:#fff6e7;--edu-border:#eed2a3}
.soluna-edu-cover{position:relative;padding:30px
34px;background:linear-gradient(125deg,#14365c,#245e91
65%,#425583);color:#fff;border-bottom:5px solid
#71cfdd}
[data-soluna-page="skyfield"] .soluna-edu-
cover{background:linear-gradient(125deg,#073c46,#126a69
65%,#255c71);border-color:#84e2c3}
[data-soluna-page="de440s"] .soluna-edu-
cover{background:linear-gradient(125deg,#372c55,#5e4276
65%,#805367);border-color:#f2c27d}
.soluna-edu-kicker{font-size:11px;letter-
spacing:.16em;font-weight:800;text-
transform:uppercase;color:#d5eafa;margin-bottom:9px}
.soluna-edu-cover h2{margin:0 0 8px;font-
size:clamp(27px,3.2vw,41px);line-
height:1.2;color:#fff;letter-spacing:-.025em}
.soluna-edu-cover p{margin:0;max-
width:75ch;color:#edf5ff;font-size:16px;line-
height:1.7}
.soluna-edu-tags{display:flex;flex-
wrap:wrap;gap:7px;margin-top:19px}
.soluna-edu-tags span{padding:4px 10px;border:1px solid
#d4e5ff6b;border-
radius:30px;color:#fff;background:#ffffff15;font-
size:11px;font-weight:700}
.soluna-edu-toolbar{display:flex;align-
items:center;justify-content:space-

```

```

between;gap:15px;padding:15px
28px;background:#eaf0f7;border-bottom:1px solid
#d4dfef}
.soluna-edu-toolbar span{font-size:12px;color:#405975}
.soluna-edu-action{padding:10px 15px;border:1px solid
#174c80;border-
radius:9px;background:#174c80;color:#fff;font-
size:13px;font-weight:750;cursor:pointer;white-
space:normal}
.soluna-edu-action:hover{background:#0e365b}
.soluna-edu-body{padding:26px 32px 34px;max-
width:1120px;margin:auto}
.soluna-edu-abstract{padding:19px 21px;background:var(--
edu-tint);border:1px solid var(--edu-border);border-
left:5px solid var(--edu-accent);border-
radius:10px;margin-bottom:23px}
.soluna-edu-abstract strong{display:block;color:var(--
edu-accent);font-size:12px;letter-spacing:.08em;text-
transform:uppercase;margin-bottom:5px}
.soluna-edu-page p{margin:0 0 13px}
.soluna-edu-abstract p:last-child{margin-bottom:0}
.soluna-edu-toc{padding:17px
20px;background:white;border:1px solid #d5dfef;border-
radius:12px;margin:0 0 27px}
.soluna-edu-toc strong{display:block;font-
size:13px;color:#173d66;margin-bottom:8px}
.soluna-edu-toc div{display:flex;flex-wrap:wrap;gap:7px
16px}
.soluna-edu-page a{color:#155b9e;text-
decoration:underline;text-underline-offset:3px}
.soluna-edu-toc a{font-size:12px;font-weight:650}
.soluna-edu-section{margin:29px 0 0;scroll-margin-
top:92px}
.soluna-edu-section h3{display:flex;align-
items:center;gap:10px;font-
size:20px;color:#133753;line-height:1.35;margin:0 0
16px;padding-bottom:10px;border-bottom:1px solid
#d2deea}
.soluna-edu-number{display:inline-grid;place-
items:center;min-width:32px;height:32px;border-
radius:9px;background:var(--edu-
accent);color:#fff;font-size:12px;font-weight:800}
.soluna-edu-page h4{font-
size:15px;color:#123e65;margin:0 0 7px;line-height:1.4}
.soluna-edu-grid{display:grid;grid-template-
columns:repeat(2,minmax(0,1fr));gap:13px;margin:17px 0}
.soluna-edu-card{border:1px solid #d2deea;border-
top:4px solid #397cbb;border-radius:11px;padding:17px
18px;background:#fff;min-width:0}

```

```

.soluna-edu-card:nth-child(4n+2){border-top-
color:#169180;background:#f4fbf9}
.soluna-edu-card:nth-child(4n+3){border-top-
color:#9b68bc;background:#fbf8ff}
.soluna-edu-card:nth-child(4n+4){border-top-
color:#cc9333;background:#fffaf0}
.soluna-edu-card p:last-child{margin-bottom:0}
.soluna-edu-callout{margin:17px 0;padding:17px
20px;border:1px solid #d6c2ef;border-left:5px solid
#8555af;border-
radius:10px;background:#f4eefb;color:#44325d}
.soluna-edu-callout strong{color:#523575}
.soluna-edu-callout.soluna-edu-note{border-
color:#eed3a5;border-left-
color:#bb771e;background:#fff7e9;color:#654815}
.soluna-edu-callout p:last-child{margin-bottom:0}
.soluna-edu-table-wrap{max-width:100%;overflow-
x:auto;border:1px solid #ccd9e7;border-
radius:10px;background:#fff;margin:15px 0;-webkit-
overflow-scrolling:touch}
.soluna-edu-table-wrap table{border-
collapse:collapse;width:100%;min-width:580px;font-
size:13px;line-height:1.6}
.soluna-edu-table-wrap th{padding:12px
14px;background:#e4edf6;color:#153a5b;text-
align:left;vertical-align:top;border-bottom:1px solid
#c6d7e8;font-size:12px}
.soluna-edu-table-wrap td{padding:12px 14px;vertical-
align:top;border-bottom:1px solid
#e0e7f0;color:#294259}
.soluna-edu-table-wrap tbody tr:nth-
child(even){background:#f5f8fc}
.soluna-edu-table-wrap tbody tr:last-child td{border-
bottom:0}
.soluna-edu-table-wrap caption{padding:10px
14px;background:#f7fafc;text-align:left;font-
size:12px;color:#476079;font-weight:700}
.soluna-edu-page ol,.soluna-edu-page ul{padding-
left:23px;margin:12px 0 17px}
.soluna-edu-page li{margin:8px 0;padding-left:3px}
.soluna-edu-steps{list-style:none;counter-reset:soluna-
edu-step;padding:0!important}
.soluna-edu-steps>li{counter-increment:soluna-edu-
step;position:relative;padding:14px 17px 14px
53px;margin:9px 0;background:#fff;border:1px solid
#d5dfef;border-radius:11px}
.soluna-edu-steps>li:before{content:counter(soluna-edu-
step,decimal-leading-

```

```

zero);position:absolute;left:15px;top:15px;font-
size:13px;font-weight:800;color:var(--edu-accent) }
.soluna-edu-equation{font-family:Georgia,"Times New
Roman",serif;font-size:18px;padding:17px;text-
align:center;background:var(--edu-tint);border:1px
solid var(--edu-border);border-radius:10px;margin:15px
0;color:#153a5b}
.soluna-edu-equation small{display:block;font-
family:"Segoe UI",Arial,sans-serif;font-
size:12px;color:#41566d;margin-top:7px}
.soluna-edu-page code{font-
size:.9em;color:#184568;background:#e9f0f7;border-
radius:4px;padding:2px 5px;overflow-wrap:anywhere}
.soluna-edu-ref{font-size:11px;vertical-
align:super;white-space:nowrap;margin-left:2px}
.soluna-edu-bibliography{font-size:13px;line-
height:1.7}
.soluna-edu-end{display:flex;flex-wrap:wrap;align-
items:center;justify-content:space-
between;gap:13px;margin-top:28px;padding:19px
20px;background:#eaf1f8;border:1px solid
#cbdbe9;border-radius:12px}
.soluna-edu-end p{margin:0;font-size:13px;max-
width:65ch}
.soluna-edu-nav button:focus-visible,.soluna-edu-
action:focus-visible,.soluna-edu-page a:focus-
visible{outline:3px solid #f6c65c;outline-offset:4px}
.soluna-edu-cover h2:focus{outline:none}
@media(max-width:800px){.soluna-edu-
body{padding:22px}.soluna-edu-
cover{padding:26px}.soluna-edu-toolbar{padding:13px
22px}.soluna-edu-grid{grid-template-columns:1fr}}
@media(max-width:650px){.soluna-edu-page{border-
radius:13px;font-size:14px}.soluna-edu-
cover{padding:23px 19px}.soluna-edu-cover p{font-
size:14px}.soluna-edu-body{padding:19px 15px
25px}.soluna-edu-toolbar{padding:12px 15px;align-
items:flex-start;flex-direction:column;gap:9px}.soluna-
edu-abstract{padding:16px}.soluna-edu-section h3{font-
size:18px}.soluna-edu-toc{padding:15px}.soluna-edu-
table-wrap table{min-width:540px}.soluna-edu-
equation{font-size:16px}.soluna-edu-
tags{gap:5px}.soluna-edu-kicker{font-size:10px}}
@media(prefers-reduced-motion:reduce){.soluna-
education-shell{scroll-behavior:auto}}
</style>
<style id="soluna-global-pkg-styles">
/* Tambahan tampilan audit penentuan global PKG; CSS
lainnya dipertahankan. */

```

```

.global-pkg-badge{display:inline-block;margin:6px 0
9px;padding:7px 12px;border:1px solid #89c8ff;border-
radius:9px;background:#173e68;color:#eaf6ff;font-
weight:800}
.global-pkg-badge.global-pkg-two{border-
color:#d2adff;background:#493469;color:#f4edff}
.global-pkg-audit{display:grid;grid-template-
columns:repeat(3,minmax(0,1fr));gap:11px;margin:15px 0}
.global-pkg-audit>div{display:flex;flex-
direction:column;gap:6px;padding:14px;border:1px solid
#52799d;border-radius:12px;background:#0d2139;overflow-
wrap:anywhere}
.global-pkg-audit small{color:#b8d7f3;font-
weight:800;letter-spacing:.02em}
.global-pkg-audit b{color:#fff}.global-pkg-audit
span{font-size:12px;color:#cad9e9}.global-pkg-audit
strong{font-size:12px}
.global-pkg-extra{margin:12px 0;padding:11px;border:1px
solid #6e5898;border-radius:10px;background:#1b2039}
.global-pkg-extra>summary{cursor:pointer;font-
weight:800;color:#e7d4ff}
.global-pkg-day{margin:7px 0;padding:8px 11px;border-
left:3px solid #71cbd8;background:#122c46;border-
radius:5px;color:#e3f3ff;font-size:13px}
.global-pkg-note{font-size:12px;color:#c3d2e5;line-
height:1.7;margin:14px 0 0}
@media(max-width:850px){.global-pkg-audit{grid-
template-columns:1fr}}
</style>
</head>
<body>
<div class="app">
<aside id="sidebar" class="sidebar">
  <div class="brand"><div
class="brandmark">☉</div><div><h1>SOLUNA<br>Astronomi
Matahari & Bulan</h1><small>WEB STATIC • JPL
DE440s</small></div></div>
  <nav class="soluna-edu-nav" aria-label="Informasi
SOLUNA">
    <button type="button" data-soluna-open="beranda"
aria-controls="solunaEducationView" aria-
current="page"><span class="soluna-edu-nav-icon" aria-
hidden="true">△</span>BERANDA</button>
    <button type="button" data-soluna-open="skyfield"
aria-controls="solunaEducationView"><span
class="soluna-edu-nav-icon" aria-
hidden="true">◆</span>SKYFIELD</button>

```

```

        <button type="button" data-soluna-open="de440s"
aria-controls="solunaEducationView"><span
class="soluna-edu-nav-icon" aria-
hidden="true"></span>DE440s</button>
    </nav>
    <div class="side-title">LOKASI PENGAMAT</div>
    <div class="group"><label>Cari Kota</label><input
id="citySearch" class="input" placeholder="Ketik nama
kota / negara..."></div>
    <div class="group"><label>Lokasi
Pengamatan</label><select id="citySelect"
class="select"></select><div id="cityCount"
class="hint"></div></div>
    <div class="grid2"><div
class="group"><label>Latitude</label><input id="lat"
class="input" type="number" step="any"></div><div
class="group"><label>Longitude</label><input id="lon"
class="input" type="number" step="any"></div></div>
    <div class="grid2"><div class="group"><label>Elevasi
(m)</label><input id="elev" class="input" type="number"
step="any"></div><div class="group"><label>Zona waktu
IANA</label><input id="tz" class="input"></div></div>
    <div class="side-title">WAKTU PERHITUNGAN</div>
    <div class="grid2"><div
class="group"><label>Tanggal</label><input id="date"
class="input" type="date"></div><div
class="group"><label>Jam</label><input id="time"
class="input" type="time" step="1"></div></div>
    <div class="group"><label>Mode waktu</label><div
id="timeMode" class="seg"><button data-v="Jam
Manual">Jam Manual</button><button data-v="Maghrib
Lokal" class="active">Maghrib Lokal</button></div><div
id="timeModeNote" class="hint">Maghrib Lokal = waktu
sunset pada kota dan tanggal terpilih.</div></div>
    <div class="group"><label>Kerangka
koordinat</label><div id="coordMode"
class="seg"><button data-
v="Toposentris">Toposentris</button><button data-
v="Geosentris"
class="active">Geosentris</button></div></div>
    <button class="btn secondary" id="nowBtn">Gunakan
Tanggal & Jam Sekarang</button>
    <button class="btn" id="calcBtn">HITUNG
ASTRONOMI</button>
    <button class="btn green" id="copyBtn">Salin
Hasil</button>
    <button class="btn orange" id="csvBtn">Unduh
CSV</button>
    <div class="side-title">ENGINE</div>

```

```

<div class="hint">Tidak memakai Python/PHP server.
Browser memuat Pyodide + Skyfield dan membaca <b>BSP
per 100 tahun</b> sesuai tahun pilihan, langsung dari
folder SOLUNA.</div>
</aside>
<button class="menubtn" id="menuBtn" type="button"
aria-label="Tampilkan/sembunyikan sidebar" aria-
expanded="true" title="Sembunyikan sidebar">≡</button>
<main class="main soluna-education-shell">
<header class="topbar"><div class="top-title">SOLUNA -
Astronomi Matahari & Bulan</div><div id="engineBadge"
class="engine-badge">Engine belum dimuat</div></header>

<div class="content">
<section id="solunaEducationView" aria-
label="Dokumentasi dan pembelajaran SOLUNA">
<article class="soluna-edu-page" data-soluna-
page="beranda" aria-labelledby="soluna-edu-home-title">
  <header class="soluna-edu-cover">
    <div class="soluna-edu-kicker">Beranda • Ilmu falak
& astronomi</div>
    <h2 id="soluna-edu-home-title" tabindex="-
1">SOLUNA</h2>
    <p>Ruang kajian Matahari dan Bulan untuk memahami
langit, menelusuri perhitungan, dan menghubungkan teori
dengan pengamatan.</p>
    <div class="soluna-edu-tags"><span>Pendidikan
akademik</span><span>Eksplorasi &
riset</span><span>Skyfield + JPL
DE440s</span><span>Aplikasi web • Kasmui</span></div>
  </header>
  <div class="soluna-edu-toolbar"><span>Dokumentasi
aplikasi • Pengantar, metode, dan panduan
praktik</span><button type="button" class="soluna-edu-
action" data-soluna-open="calculator">Buka Perhitungan
Astronomi →</button></div>
  <div class="soluna-edu-body">
    <div class="soluna-edu-
abstract"><strong>Abstrak</strong><p>SOLUNA adalah
aplikasi astronomi Matahari dan Bulan yang menyatukan
posisi benda langit, peristiwa harian, fase Bulan,
kajian hilal, pemindaian awal bulan Hijriah, jadwal
astronomis, dan informasi gerhana. Aplikasi ini
membantu mahasiswa, dosen, peneliti, serta peminat ilmu
falak mempelajari hubungan antara lokasi, waktu,
kerangka koordinat, dan hasil hitungan. Perhitungan
dijalankan di browser melalui Skyfield dengan data
ephemeris JPL DE440s; dokumentasi ini menjelaskan cara

```

memakai dan menafsirkan keluaran versi aplikasi yang sedang digunakan.

```
<nav class="soluna-edu-toc" aria-label="Daftar isi Beranda"><strong>Daftar isi</strong><div><a href="#soluna-edu-home-1">01 Pengertian & tujuan</a><a href="#soluna-edu-home-2">02 Fitur</a><a href="#soluna-edu-home-3">03 Panduan penggunaan</a><a href="#soluna-edu-home-4">04 Membaca hasil</a><a href="#soluna-edu-home-5">05 Awal bulan Hijriah</a><a href="#soluna-edu-home-6">06 Praktikum & riset</a><a href="#soluna-edu-home-7">07 Catatan metodologis</a><a href="#soluna-edu-home-8">08 Rujukan</a></div></nav>
```

```
<section class="soluna-edu-section" id="soluna-edu-home-1"><h3><span class="soluna-edu-number">01</span>Pengertian, tujuan, dan kegunaan</h3>
```

<p>Dalam SOLUNA, satu pilihan tempat dan waktu menjadi titik awal untuk memeriksa berbagai besaran astronomis. Pengguna dapat mengubah variabel secara terkontrol, membandingkan hasil, lalu menjelaskan penyebab perbedaannya. Cara kerja ini cocok untuk pembelajaran berbasis penyelidikan: pertanyaan dirumuskan terlebih dahulu, hasil dihitung, dan kesimpulan ditulis berdasarkan data.</p>

```
<div class="soluna-edu-grid">
```

```
<div class="soluna-edu-card"><h4>Memahami konsep</h4><p>Pelajari tinggi dan azimut, asensio rekta dan deklinasi, fase serta iluminasi Bulan, paralaks, konjungsi, dan perbedaan geosentris-toposentris melalui contoh yang dapat diulang.</p></div>
```

```
<div class="soluna-edu-card"><h4>Mendukung kajian falak</h4><p>Telusuri keadaan Matahari dan Bulan pada Maghrib, hubungan konjungsi dengan umur Bulan, perkembangan parameter hilal, serta pengaruh parameter astronomis terhadap jadwal.</p></div>
```

```
<div class="soluna-edu-card"><h4>Merencanakan pengamatan</h4><p>Gunakan arah benda langit, waktu terbit-terbenam, transit, dan senja sebagai bahan menyusun rencana observasi. Cocokkan prediksi dengan keadaan ufuk, cuaca, dan catatan pengamatan.</p></div>
```

```
<div class="soluna-edu-card"><h4>Menyusun analisis yang dapat diulang</h4><p>Simpan keluaran, catat masukan dan mode, lalu bandingkan antarwaktu atau antarlokasi. Data dapat dipakai untuk tabel laporan, diskusi kelas, dan pengujian hipotesis sederhana.</p></div>
```

```
</div>
```

<p>Tiga unsur memiliki peran berbeda:
SOLUNA menyediakan antarmuka dan aturan analisis; Skyfield menjalankan

fungsi astronomi; DE440s memasok data ephemeris. Penjelasan kedua komponen terakhir tersedia melalui menu di sidebar.</p>

</section>

<section class="soluna-edu-section" id="soluna-edu-home-2"><h3>02Fitur yang tersedia</h3>

<div class="soluna-edu-table-wrap" role="region" aria-label="Tabel fitur SOLUNA, dapat digeser mendatar" tabindex="0"><table><caption>Tabel 1. Peta fitur dan penggunaan akademik</caption><thead><tr><th>Ruang / tab</th><th>Isi

utama</th><th>Pemanfaatan</th></tr></thead><tbody>

<tr><td>Lokasi & waktu</td><td>Pencarian kota, lintang, bujur, elevasi, zona waktu IANA, tanggal, Jam Manual atau Maghrib Lokal.</td><td>Menetapkan kondisi eksperimen dan mencatat variabel masukan.</td></tr>

<tr><td>Ringkasan, Matahari, Bulan</td><td>Koordinat, arah, jarak, ukuran sudut, elongasi, iluminasi, serta parameter posisi dan fase.</td><td>Menghubungkan sistem koordinat dengan geometri benda langit.</td></tr>

<tr><td>Peristiwa</td><td>Terbit-terbenam, transit, senja, konjungsi, umur Bulan, dan selang waktu terkait.</td><td>Mengkaji perubahan harian dan merencanakan observasi.</td></tr>

<tr><td>Fenomena</td><td>Peristiwa astronomi dan perkembangan parameter hilal pada bulan terpilih.</td><td>Mempelajari evolusi kondisi Bulan serta perbandingan hari pengamatan.</td></tr>

<tr><td>Penentuan 1 Hijriah global</td><td>Pemindaian 100 kota pada H0-H2, verifikasi daratan Amerika, keputusan PKG 1/PKG 2 geosentris, dan tanggal awal bulan.</td><td>Mengaudoit penerapan ambang numerik dan pengaruh lokasi pada hasil.</td></tr>

<tr><td>Jadwal</td><td>Waktu astronomis Shubuh, Dhuha, Dhuhur, Ashar, Maghrib, Isya, tengah malam, dan sepertiga malam.</td><td>Membandingkan konvensi sudut Matahari dan panjang malam.</td></tr>

<tr><td>Gerhana</td><td>Informasi pencarian gerhana Bulan dan geometri gerhana Matahari.</td><td>Mempelajari susunan Matahari-Bumi-Bulan dan membaca batas model.</td></tr>

<tr><td>Metodologi & Istilah</td><td>Uraian aturan perhitungan dan penjelasan istilah pada panel hasil.</td><td>Menelusuri definisi sebelum menarik kesimpulan.</td></tr>

```
|  |  |  |
| --- | --- | --- |
| Salin Hasil & Unduh CSV | Penyimpanan keluaran perhitungan yang sudah dijalankan. | Membuat arsip, tabel perbandingan, dan lampiran laporan. |

```

```


```

```


```

```

<section class="soluna-edu-section" id="soluna-edu-home-3">
<h3><span class="soluna-edu-number">03</span>Panduan penggunaan</h3>

```

```

<ol class="soluna-edu-steps">

```

```

<li><strong>Kenali halaman informasi.</strong>
BERANDA menjelaskan aplikasi; SKYFIELD menjelaskan
pustaka perhitungan; DE440s menjelaskan data ephemeris.
Tombol <em>Buka Perhitungan Astronomi</em> menampilkan
panel hitung. Pergantian halaman tetap mempertahankan
masukan dan hasil yang sudah ada.</li>

```

```

<li><strong>Siapkan lokasi pengamat.</strong>
Cari dan pilih kota, lalu periksa lintang, bujur,
elevasi dalam meter, dan zona waktu. Untuk lokasi
sendiri, isikan koordinat yang benar: lintang utara dan
bujur timur positif, lintang selatan dan bujur barat
negatif. Zona waktu mengikuti lokasi, misalnya
<code>Asia/Jakarta</code>.</li>

```

```

<li><strong>Tentukan tanggal dan mode
waktu.</strong> <em>Jam Manual</em> memakai jam yang
diisikan. <em>Maghrib Lokal</em> menghitung waktu
terbenam Matahari untuk kota dan tanggal terpilih.
Tombol <em>Gunakan Tanggal & Jam Sekarang</em> membantu
mengisi waktu kini sesuai perilaku aplikasi.</li>

```

```

<li><strong>Pilih kerangka koordinat.</strong>
Gunakan Toposentris untuk kajian dari tempat pengamat,
atau Geosentris untuk kajian dengan pusat Bumi sebagai
acuan posisi benda. Saat membandingkan keduanya,
pertahankan lokasi dan waktu yang sama. Panel penentuan
awal bulan global selalu memakai geosentris untuk PKG
1/PKG 2.</li>

```

```

<li><strong>Jalankan HITUNG ASTRONOMI.</strong>
Tunggu mesin siap dan proses selesai. Aplikasi memilih
satu BSP yang sesuai tahun, kemudian menyajikan hasil
lokal, kajian bulanan, dan scan 100 kota. Scan global
selalu memakai sunset lokal masing-masing kota dan
kerangka geosentris, termasuk saat panel lokal memakai
Jam Manual atau Toposentris. H0 beracuan tanggal UTC
ijtimak; PKG 2 memeriksa ijtimak terhadap fajar
Gisborne.</li>

```

```

<li><strong>Telusuri hasil dan simpan.</strong>
Baca kartu waktu, lokasi, dan status, lalu pilih tab
yang diperlukan. Ruang kanan dapat digulir ke bawah;
tabel yang lebar dapat digeser mendatar. Gunakan

```

Salin Hasil atau Unduh CSV setelah perhitungan. Tombol  membuka kembali sidebar apabila tertutup.

<div class="soluna-edu-callout"><p>Dua waktu yang berbeda. Jam pada pojok kanan atas mengikuti jam sistem laptop atau HP. Tanggal dan waktu pada kartu hasil mengikuti masukan perhitungan serta zona waktu lokasi. Mengubah tanggal kajian tidak mengubah jam sistem, dan jam sistem bukan penanda waktu dari hasil yang tersimpan.</p></div>

</section>

<section class="soluna-edu-section" id="soluna-edu-home-4"><h3>04Cara membaca besaran astronomis</h3>

<div class="soluna-edu-grid">

<div class="soluna-edu-card"><h4>Tinggi & azimut</h4><p>Tinggi menunjukkan sudut terhadap ufuk; nilai negatif berarti posisi geometris di bawah ufuk. Azimut menunjukkan arah horizontal: 0° utara, 90° timur, 180° selatan, dan 270° barat. Jangan menyamakan tinggi sudut dengan elevasi tempat dalam meter.</p></div>

<div class="soluna-edu-card"><h4>Elongasi & iluminasi</h4><p>Elongasi adalah jarak sudut Bulan-Matahari pada kerangka yang dipakai. Iluminasi adalah persentase cakram Bulan yang diterangi dalam model. Dua Bulan dengan iluminasi hampir sama dapat memiliki tinggi atau keadaan ufuk yang berbeda.</p></div>

<div class="soluna-edu-card"><h4>Konjungsi & umur Bulan</h4><p>Konjungsi geosentris untuk fase Bulan berkaitan dengan kesamaan bujur ekliptika Bulan dan Matahari. Umur Bulan mengukur selang sejak konjungsi yang dipakai. Periksa acuan dan tanggalnya; umur saja belum menjelaskan seluruh keadaan hilal.</p></div>

<div class="soluna-edu-card"><h4>Geosentris & toposentris</h4><p>Perubahan titik pengamat mengubah garis pandang, terutama untuk Bulan yang relatif dekat. Karena itu, perbedaan tinggi atau elongasi antarmode perlu dijelaskan melalui geometri dan paralaks, bukan langsung dianggap sebagai kesalahan hitung.</p></div>

</div>

<p>Asensio rekta umumnya dinyatakan dalam jam sudut, sedangkan deklinasi dalam derajat. Hubungan satuan sudutnya adalah 24 jam = 360°, sehingga 1 jam = 15°. Tinggi geometris dan tinggi yang dikoreksi refraksi juga perlu dibedakan: pembiasan atmosfer memengaruhi arah tampak, terutama dekat ufuk.</p>

```
<div class="soluna-edu-equation">sin h = sin φ
sin δ + cos φ cos δ cos H<small>Hubungan dasar
koordinat bola: h = tinggi; φ = lintang; δ = deklinasi;
H = sudut jam. Gunakan satuan sudut yang konsisten.
Persamaan ini membantu pembelajaran geometri; pipeline
aplikasi juga memakai transformasi dan koreksi
astronomis.</small></div>
```

```
</section>
```

```
<section class="soluna-edu-section" id="soluna-edu-
home-5"><h3><span class="soluna-edu-
number">05</span>KHGT: penentuan awal bulan melalui PKG
1 dan PKG 2</h3>
```

```
<p><strong>Kalender Hijriah Global Tunggal
(KHGT)</strong> mengupayakan satu hari satu tanggal
untuk seluruh dunia. Parameternya berhubungan dengan
kesatuan matlak, transfer keterpenuhan kriteria, serta
keselarasan hari. Angka tinggi 5° dan elongasi 8°
merupakan syarat astronomis; penentuan awal bulan juga
harus memeriksa batas waktu UTC dan koreksi PKG 2. <a
class="soluna-edu-ref"
href="https://muhammadiyah.or.id/2023/07/syamsul-anwar-
terangkan-validitas-dan-parameter-kalender-islam-
global/" target="_blank" rel="noopener noreferrer"
aria-label="Rujukan parameter KHGT">[3]</a></p>
```

```
<div class="soluna-edu-grid">
```

```
<div class="soluna-edu-card"><h4>Kesatuan
matlak</h4><p>Seluruh Bumi dipandang sebagai satu
wilayah kalender. Hasil global tidak dibuat berbeda
hanya karena pengamat memilih Sabang, Jakarta, atau
kota lain.</p></div>
```

```
<div class="soluna-edu-card"><h4>Transfer
keterpenuhan kriteria</h4><p>Satu saksi daratan yang
memenuhi parameter dan ketentuan waktunya dapat menjadi
dasar awal bulan secara global. Semua kota tidak harus
memenuhi pada saat yang sama.</p></div>
```

```
<div class="soluna-edu-card"><h4>Keterpaduan
hari</h4><p>Tanggal awal bulan berlaku sebagai tanggal
kalender yang sama di seluruh dunia. Pergantian hari
tetap mengikuti sistem tanggal internasional; bukan
satu saat tengah malam lokal yang bersamaan di semua
negara.</p></div>
```

```
<div class="soluna-edu-card"><h4>Parameter
geosentris 5° + 8°</h4><p>Pada sunset lokal, tinggi
Bulan geosentris harus ≥ 5° dan elongasi geosentris ≥
8°. Panel global selalu memakai geosentris, walaupun
perhitungan posisi lokal dipilih Toposentris.</p></div>
```

```
</div>
```

```
<p>Dalam algoritme SOLUNA, <strong>H0 adalah
tanggal UTC ijtimak geosentris</strong>. H1 dan H2
```

adalah satu dan dua hari kalender setelahnya. Untuk setiap hari evaluasi D, program menghitung sunset lokal bertanggal D pada daftar lokasi; batas PKG adalah **pukul 00.00 UTC tanggal D+1**.

Perbandingan dilakukan memakai waktu lengkap beserta tanggalnya, bukan membandingkan angka jam saja.

`<div class="soluna-edu-table-wrap" role="region" aria-label="Tabel keputusan PKG, dapat digeser mendatar" tabindex="0"><table><caption>Tabel 2. Filter keputusan awal bulan KHGT dalam SOLUNA</caption><thead><tr><th>Jalur</th><th>Syarat tambahan setelah 5° + 8°</th><th>Keputusan untuk hari D</th></tr></thead><tbody>`

`<tr><td><strong>PKG 1</strong></td><td>Sunset saksi terjadi setelah ijtimak dan sebelum batas D+1 pukul 00.00 UTC.</td><td>Awal bulan global ditetapkan pada tanggal D+1.</td></tr>`

`<tr><td><strong>PKG 2</strong></td><td>Kriteria tercapai pada/setelah batas tengah malam UTC di daratan utama Amerika; ijtimak terjadi sebelum fajar Gisborne pada tanggal D+1.</td><td>Awal bulan tetap tanggal D+1. Sunset di Amerika dapat berlabel UTC D+1 tanpa menambah satu hari lagi.</td></tr>`

`<tr><td><strong>PKG 2 ditunda</strong></td><td>Ijtimak tidak terjadi sebelum fajar Gisborne tanggal D+1, atau syarat daratan Amerika tidak terpenuhi.</td><td>Tanggal D+1 tidak disahkan melalui PKG 2; program memeriksa hari D+1 berikutnya. Jika lolos, awal bulan menjadi D+2.</td></tr>`

`</tbody></table></div>`

`<p>PKG 1 diperiksa lebih dahulu. PKG 2 dipakai ketika PKG 1 belum ditemukan dan syarat koreksinya terpenuhi. Yang dibandingkan dengan fajar adalah saat ijtimak, bukan saat sunset atau saat 5° + 8° tercapai. Pulau-pulau pada daftar Amerika tetap ditampilkan, tetapi tidak menjadi saksi daratan utama Amerika untuk PKG 2.</p>`

`<div class="soluna-edu-callout"><p>Definisi fajar yang dapat diaudit. Versi ini menghitung fajar Gisborne dari naiknya pusat Matahari toposentris melewati -18°, pada koordinat Gisborne yang sudah ada dalam aplikasi, dengan zona waktu <code>Pacific/Auckland</code>. Tanggal fajar adalah tanggal calon awal bulan D+1. Memakai "fajar pertama setelah ijtimak" akan selalu membuat ijtimak lebih dahulu dan dapat meloloskan PKG 2 secara keliru. Sudut -18° merupakan konvensi komputasi yang dinyatakan di layar, bukan klaim bahwa semua rujukan fajar memakai definisi identik.</p></div>`

<h4>Contoh kajian: Jumadilakhir 1448 H</h4>

<p>Hasil yang dituju adalah 1 Jumadilakhir 1448 H = 10-11-2026, sesuai kalender KHGT Muhammadiyah. Dalam pengujian DE440s pada aplikasi ini, lokasi Santiago memenuhi pada 9 November sekitar 23.17 UTC dengan tinggi geosentris sekitar $8,05^{\circ}$ dan elongasi sekitar $9,15^{\circ}$. Saksi tersebut termasuk PKG 1 karena mendahului batas 10 November pukul 00.00 UTC. Jika saksi yang diperiksa baru memenuhi sesudah batas itu, keputusan harus ditelusuri melalui PKG 2 dan perbandingan ijtimak-fajar. [4]</p>

<p>Daftar utama 100 kota lama dipertahankan. Bila PKG 1 belum ditemukan pada daftar tersebut, algoritme memeriksa sampel tambahan daratan Amerika, termasuk Amerika Selatan yang sebelumnya kurang terwakili. Saksi dan hasil tambahan ditampilkan terpisah. Adanya satu saksi yang sah mendukung keputusan; ketiadaan saksi pada sampel terbatas belum membuktikan bahwa setiap titik daratan di seluruh Bumi tidak memenuhi.</p>

<div class="soluna-edu-callout"><p>Patokan penamaan bulan dan tahun tetap. Label mengikuti 1 Jumadilawal 1448 H = 12-10-2026, lalu indeks bulan bergerak maju atau mundur menurut lunasi. Penamaan bulan/tahun mengikuti acuan ini; tanggal Gregorian awal bulan berasal dari keputusan PKG 1/PKG 2, tanpa penguncian tanggal khusus November 2026.</p></div>

<p>Untuk menelusuri hasil, baca label PKG, tanggal H0 dalam UTC, batas tengah malam, waktu ijtimak, fajar Gisborne dalam waktu lokal dan UTC, kota saksi, serta tabel H0-H2. Kolom Filter $5^{\circ} + 8^{\circ}$ dan Keputusan PKG dipisahkan agar lolos ambang numerik tidak disamakan dengan lolos seluruh syarat kalender. Keluaran program tetap merupakan hasil perhitungan, bukan catatan keberhasilan rukyat lapangan.</p>

</section>

<section class="soluna-edu-section" id="soluna-edu-home-6"><h3>06Praktikum dan penggunaan untuk riset</h3>

<div class="soluna-edu-grid">

<div class="soluna-edu-card"><h4>Praktikum A • Paralaks Bulan</h4><p>Pilih satu tempat, tanggal, dan jam. Hitung pada mode Geosentris, lalu Toposentris.</p>

Catat tinggi, elongasi, dan jarak; hitung selisihnya. Jelaskan mengapa perubahan titik pengamat memengaruhi posisi Bulan.

Praktikum B • Pengaruh lokasi
Bandingkan dua kota pada saat UTC yang sama dengan mengonversi jam lokalnya. Amati tinggi, azimut, dan waktu terbenam. Pisahkan pengaruh letak geografis dari perbedaan waktu pengamatan.

Praktikum C • Perkembangan hilal
Pilih bulan kajian dan mode Maghrib Lokal. Baca perkembangan tinggi, elongasi, dan iluminasi setelah konjungsi. Bandingkan dengan tabel H0-H2 global, lalu telusuri apakah keputusan melalui PKG 1 atau PKG 2. Jelaskan pengaruh batas UTC serta perbandingan ijtima-fajar Gisborne.

Praktikum D • Konvensi jadwal
Bandingkan Shubuh dengan sudut Matahari -20° dan -18° untuk tempat dan hari yang sama. Periksa perbedaan waktu dan perubahan panjang malam ketika tanggal atau lintang diubah.

Untuk laporan akademik, susun pertanyaan, landasan teori, data dan metode, hasil, pembahasan, serta kesimpulan. Catat nama lokasi, koordinat, elevasi, zona waktu, tanggal dan jam, mode waktu, kerangka koordinat, sumber ephemeris, dan aturan seleksi. Cantumkan satuan serta jumlah angka yang relevan. Pisahkan hasil perhitungan dari catatan observasi agar keduanya dapat dibandingkan dengan jelas.

Validasi dilakukan terhadap sumber pembanding dengan kondisi yang setara: waktu UTC sama, titik pengamat sama, definisi geometris atau tampak sama, serta definisi peristiwa sama. Jika hasil berbeda, telusuri satu variabel pada satu waktu sebelum menilai penyebabnya. Arsip CSV membantu mengulang dan memeriksa analisis.

Catatan metodologis

Jadwal dalam versi ini memakai Shubuh -20° dan -18° , Dhuha $+4,5^\circ$, Dhuhur saat transit Matahari, Ashar faktor bayangan 1, Maghrib saat sunset, serta Isya -18° . Tengah malam dan awal sepertiga malam terakhir memakai selang Maghrib hingga Shubuh -18° keesokan harinya. Hasilnya belum ditambah menit ihtiyat.

Konvensi ini perlu disebutkan ketika angka digunakan dalam laporan atau dibandingkan dengan jadwal lain.

Gerhana Matahari pada aplikasi ditelusuri melalui geometri bayangan dan pendekatan pencarian numerik. Penafsiran wilayah, jenis, serta waktu pengamatan perlu mengikuti rincian model pada tab Metodologi. Untuk analisis lintasan gerhana yang menuntut ketelitian khusus, gunakan data pembanding dengan model dan definisi yang sesuai.

Ketelitian penyajian dan batas penerapan. Banyak angka desimal menunjukkan resolusi keluaran, bukan jaminan akurasi pengamatan. Keadaan atmosfer, profil ufuk, ketepatan koordinat, data waktu, dan definisi peristiwa memengaruhi perbandingan dengan langit nyata. Pada lintang tinggi, sebagian peristiwa terbit-terbenam atau ambang senja dapat tidak terjadi pada tanggal tertentu.

Saat pertama dibuka, aplikasi menyiapkan runtime dan BSP sesuai tahun berjalan. Berkas tahunan 2026–2030 didahulukan; untuk tahun lain digunakan satu berkas rentang yang sesuai. Navigasi halaman informasi ini tidak menjalankan perhitungan baru dan tidak meminta berkas BSP tambahan.

Rujukan dan tindak lanjut

Uraian fitur dan aturan di halaman ini mengikuti kode SOLUNA yang menyertai aplikasi. Landasan pustaka dan data dapat ditelusuri pada dokumentasi resmi berikut. Untuk penulisan akademik, sertakan tanggal akses dan identitas versi aplikasi yang dipakai.

Rhodes, B. <https://rhodesmill.org/skyfield/> Skyfield: dokumentasi resmi. Penjelasan lebih rinci tersedia pada menu SKYFIELD.

Park, R. S., Folkner, W. M., Williams, J. G., & Boggs, D. H. (2021). <https://doi.org/10.3847/1538-3881/abd414> The JPL Planetary and Lunar Ephemerides DE440 and DE441. *The Astronomical Journal*, 161, 105. Penjelasan data tersedia pada menu DE440s.

Muhammadiyah (2023). <https://muhammadiyah.or.id/2023/07/syamsul-anwar-terangkan-validitas-dan-parameter-kalender-islam-global/>

```

noreferrer">Penjelasan prinsip dan parameter kalender
Islam global</a>. Lihat pula <a
href="https://tarjih.or.id/wp-
content/uploads/2025/06/Buku-Saku-KHGT.pdf"
target="_blank" rel="noopener noreferrer">Buku Saku
KHGT, Majelis Tarjih dan Tajdid PP Muhammadiyah
(2025)</a>.</li><li>Majelis Tarjih dan Tajdid PP
Muhammadiyah. <a
href="https://khgt.muhammadiyah.or.id/kalendar-hijriah"
target="_blank" rel="noopener noreferrer">Kalender
KHGT</a>: pembandingan tanggal awal bulan.</li></ol>
</section>
<div class="soluna-edu-end"><p><strong>Mulai dari
satu pertanyaan.</strong> Tentukan tempat, waktu, dan
parameter yang ingin dibandingkan; hitung, simpan, lalu
jelaskan hasilnya.</p><button type="button"
class="soluna-edu-action" data-soluna-
open="calculator">Buka Perhitungan Astronomi
→</button></div>
</div>
</article>

<article class="soluna-edu-page" data-soluna-
page="skyfield" aria-labelledby="soluna-edu-sky-title"
hidden>
<header class="soluna-edu-cover"><div class="soluna-
edu-kicker">Skyfield • Pustaka perhitungan
astronomi</div><h2 id="soluna-edu-sky-title"
tabindex="-1">SKYFIELD</h2><p>Dari data ephemeris ke
posisi dan peristiwa astronomis yang dapat dianalisis,
dibandingkan, dan dipelajari.</p><div class="soluna-
edu-tags"><span>Python</span><span>Koordinat &
waktu</span><span>Geometri
pengamat</span><span>Dokumentasi
resmi</span></div></header>
<div class="soluna-edu-toolbar"><span>Landasan
komputasi • Peran Skyfield dalam SOLUNA</span><button
type="button" class="soluna-edu-action" data-soluna-
open="calculator">Buka Perhitungan Astronomi
→</button></div>
<div class="soluna-edu-body">
<div class="soluna-edu-
abstract"><strong>Abstrak</strong><p>Skyfield adalah
pustaka Python untuk menghitung posisi astronomis dan
berbagai peristiwa langit. Dalam SOLUNA, Skyfield
dipakai untuk membaca ephemeris JPL, membentuk geometri
Bumi-pengamat-benda langit, mengelola skala waktu, dan
menjalankan pencarian peristiwa. Aturan kajian hilal,
pemindaian kota, penyajian tabel, dan pilihan antarmuka

```

tetap merupakan bagian dari aplikasi SOLUNA. Pemahaman peran ini membantu pengguna menelusuri asal suatu hasil. [1]</p></div>

<nav class="soluna-edu-toc" aria-label="Daftar isi Skyfield">Daftar isi<div>01 Pengertian02 Mengapa digunakan03 Alur perhitungan04 Koordinat & waktu05 Peristiwa & interpretasi06 Reprodusibilitas07 Daftar pustaka</div></nav>

<section class="soluna-edu-section" id="soluna-edu-sky-1"><h3>01Apa itu Skyfield?</h3><p>Skyfield dikembangkan oleh Brandon Rhodes sebagai perangkat komputasi astronomi berbasis Python. Pustaka ini mendukung posisi geosentris dan posisi pengamat di permukaan Bumi, termasuk koordinat geografis dengan WGS84. Skyfield juga tercatat dalam Astrophysics Source Code Library dengan pengenal ascl:1907.024, sehingga perangkat lunaknya dapat dirujuk dalam karya ilmiah. [1][2]</p><p>Skyfield menyediakan cara menghitung dan mentransformasikan besaran; hasil tertentu tetap bergantung pada ephemeris yang dibaca, saat yang dihitung, titik pengamat, dan jenis koreksi yang diterapkan. Karena itu, nama pustaka perlu disertai penjelasan konfigurasi ketika suatu hasil dikutip.</p></section>

<section class="soluna-edu-section" id="soluna-edu-sky-2"><h3>02Mengapa Skyfield digunakan dalam SOLUNA?</h3>

<div class="soluna-edu-grid"><div class="soluna-edu-card"><h4>Satu alur untuk posisi dan peristiwa</h4><p>Program memakai objek waktu dan ephemeris yang sama untuk posisi Matahari-Bulan, fase, terbit-terbenam, transit, dan kajian lanjutan. Susunan ini membantu menjaga kesesuaian acuan antarbagian keluaran.</p></div><div class="soluna-edu-card"><h4>Perbandingan kerangka pengamat</h4><p>Objek

Bumi dan lokasi WGS84 memungkinkan SOLUNA menyediakan pilihan geosentris serta toposentris. Pengguna dapat mempelajari pengaruh titik pengamat tanpa mengganti sumber ephemeris.

Metode dapat ditelusuri

Dokumentasi menjelaskan fungsi, definisi koordinat, dan peristiwa. Kode aplikasi dapat dikaitkan dengan rujukan tersebut untuk mengaudit bagaimana angka dihasilkan dan mengapa konvensi tertentu dipakai.

Sesuai rancangan aplikasi web

SOLUNA menjalankan Python melalui Pyodide di browser. Kode yang tersedia memuat NumPy, jplephem, dan Skyfield, lalu membaca BSP lokal yang sesuai. Hosting tidak perlu mengeksekusi Python atau PHP untuk hitungan astronominya.

Pemakaian Skyfield juga membantu pembelajaran bertahap: mulai dari posisi pada satu saat, lanjut ke pencarian peristiwa dalam interval, kemudian ke perbandingan lokasi atau lunasi. Aturan aplikasi dapat dibahas terpisah dari operasi dasar pustaka.

03 Alur perhitungan dalam SOLUNA

- Masukan.** Aplikasi membaca lokasi, lintang, bujur, elevasi, zona waktu, tanggal, mode waktu, dan kerangka koordinat.
- Penyiapan waktu dan data.** Tanggal lokal dikaitkan dengan zona waktu, dikonversi menjadi saat yang sesuai, lalu ephemeris untuk tahun pilihan disiapkan.
- Pembentukan pengamat.** Kode membuat lokasi WGS84 dan menggabungkannya dengan posisi Bumi untuk membentuk pengamat toposentris; mode geosentris memakai acuan pusat Bumi untuk posisi benda.
- Posisi dan pencarian peristiwa.** Matahari dan Bulan dihitung pada saat terpilih. Fungsi almanak serta pencarian tambahan dalam SOLUNA menelusuri terbit-terbenam, fase, ambang sudut, dan fenomena yang diperlukan.
- Analisis dan penyajian.** SOLUNA menerapkan aturan hilal, scan 100 kota, dan konvensi jadwal, lalu menampilkan tabel, kartu, penjelasan, serta ekspor hasil.

Contoh pemisahan tanggung jawab. Pencarian fase Bulan memakai fasilitas Skyfield. Pemilihan tanggal H0-H2, daftar 100 kota, ambang $5^{\circ}/8^{\circ}$, dan penamaan bulan

Hijriah ditentukan kode SOLUNA. Skyfield tidak menetapkan kebijakan kalender Hijriah aplikasi.

Koordinat, koreksi, dan skala waktu

Dokumentasi Skyfield membedakan posisi barysentris, astrometris, dan tampak. Posisi astrometris memperhitungkan waktu tempuh cahaya dari benda; posisi tampak selanjutnya mencakup koreksi seperti aberasi dan defleksi gravitasi. Transformasi koordinat kemudian memungkinkan pembacaan dalam asensio rekta-deklinasi atau tinggi-azimut. Istilah

tampak pada pipeline ini tidak otomatis berarti pembiasan atmosfer sudah diterapkan. [\[3\]](https://rhodesmill.org/skyfield/positions.html)

Tabel konsep koordinat dan waktu, dapat digeser mendatar

Konsep	Makna	Perhatian dalam kajian
Geosentris	Acuan posisi benda dari pusat Bumi.	Nyatakan mode ketika membandingkan data
Toposentris	Acuan dari tempat pengamat di permukaan Bumi.	Catat lintang, bujur, dan elevasi; perhatikan paralaks.
Geometris / koreksi refraksi	Arah geometris dan perubahan arah akibat pembiasan atmosfer merupakan besaran yang berbeda.	Periksa kolom tinggi yang digunakan dalam kriteria.
Waktu lokal / UTC	Penulisan waktu menurut zona lokasi atau standar UTC.	Dua penulisan dapat mewakili saat yang sama; tanggal kalendernya dapat berbeda.
Skala astronomis	Skyfield mendukung UTC, UT1, TAI, TT, dan TDB.	Skala waktu diperlukan sesuai operasi, bukan sekadar pergantian label jam.

UT1 berkaitan dengan rotasi Bumi; TT digunakan sebagai skala waktu terrestrial, sedangkan TDB berkaitan dengan perhitungan dinamika barysentris dan ephemeris JPL. Skyfield menyediakan objek waktu untuk

menghubungkan skala tersebut. Nilai waktu lokal harus terlebih dahulu diartikan dengan zona yang benar sebelum dibandingkan dalam skala lain. [4]</p>

<p>Latihan yang berguna adalah memilih satu saat UTC, menuliskannya dalam dua zona waktu, lalu memastikan bahwa hasil yang dibandingkan memang dihitung pada saat fisik yang sama. Latihan ini menghindari kekeliruan yang sering muncul ketika hari kalender lokal berubah saat melintasi zona waktu.</p>

</section>

<section class="soluna-edu-section" id="soluna-edu-sky-5"><h3>05Peristiwa astronomis dan interpretasinya</h3>

<p>Fungsi almanak Skyfield dapat mencari fase Bulan, terbit-terbenam, dan transit dalam suatu interval. Peristiwa ditemukan melalui evaluasi posisi berulang, sehingga pencarian dapat lebih berat daripada menghitung satu posisi. Dokumentasinya juga menjelaskan bahwa definisi terbit-terbenam melibatkan ukuran cakram dan konvensi refraksi; di kawasan kutub, peristiwa tersebut dapat tidak terjadi setiap hari. [5]</p>

<p>Dalam praktik SOLUNA, pencarian peristiwa dipadukan dengan aturan aplikasi. Jadwal Shubuh dan Isya, misalnya, bergantung pada sudut Matahari yang dipilih kode. Pemindaian global mengulangi perhitungan untuk banyak kota dan hari. Berkas BSP yang kecil mengurangi beban pemuatan data, tetapi banyaknya evaluasi astronomis tetap berpengaruh pada lama proses.</p>

<div class="soluna-edu-callout soluna-edu-note"><p>Baca definisi sebelum membandingkan. Tinggi pusat cakram, tinggi tepi cakram, tinggi geometris, dan tinggi yang dikoreksi refraksi tidak selalu menghasilkan saat yang sama. Demikian pula, waktu transit dan waktu tinggi maksimum tidak harus persis sama. Catat definisi sumber pembanding agar selisih dapat dijelaskan.</p></div>

<p>Untuk pembelajaran, pilih satu peristiwa seperti Maghrib. Uraikan masukan lokasi dan tanggal, definisi sunset yang dipakai, hasil waktu lokal, dan kondisi ufuk saat observasi. Dengan langkah ini,

mahasiswa belajar menilai hubungan antara model numerik dan pengukuran.</p>

</section>

<section class="soluna-edu-section" id="soluna-edu-sky-6"><h3>06Reproduksibilitas dan penggunaan akademik</h3><p>Hasil yang dapat diulang memerlukan rekaman konfigurasi, bukan hanya satu angka. Cantumkan aplikasi SOLUNA, versi Skyfield yang digunakan jika diketahui dari instalasi, ephemeris, koordinat, zona waktu, saat perhitungan, mode, dan definisi peristiwa. Jangan menebak nomor versi pustaka: kode web ini memasang paket saat penyiapan sehingga identitas instalasi perlu dicatat dari lingkungan yang benar-benar digunakan.</p><p>Ketika menyusun laporan, gunakan dokumentasi untuk menjelaskan fungsi dan ASCL untuk identitas perangkat lunak. Hindari menyamakan ketelitian numerik pustaka dengan ketelitian seluruh hasil observasi; kesalahan masukan, data waktu, asumsi atmosfer, dan model analisis juga perlu dinilai.</p><div class="soluna-edu-callout"><p>Latihan audit metode. Pilih satu baris hasil Bulan. Tulis besaran dan satuannya, titik pengamat, saat perhitungan, ephemeris, serta koreksi yang relevan. Jelaskan bagian yang dihitung Skyfield dan bagian yang ditentukan aturan SOLUNA.</p></div></section>

<section class="soluna-edu-section" id="soluna-edu-sky-7"><h3>07Daftar pustaka dan bacaan resmi</h3><ol class="soluna-edu-bibliography">Rhodes, B. Skyfield: situs dan dokumentasi resmi.Rhodes, B. (2019). Skyfield: High precision research-grade positions for planets and Earth satellites generator. Astrophysics Source Code Library, ascl:1907.024.Dokumentasi Skyfield. Positions and Coordinates.Dokumentasi Skyfield. Dates and Time.Dokumentasi Skyfield. <a href="https://rhodesmill.org/skyfield/almanac.html"

```

target="_blank" rel="noopener noreferrer"><em>Almanac
Computation</em></a>.</li></ol></section>
  <div class="soluna-edu-end"><p><strong>Lanjutkan ke
sumber datanya.</strong> Menu DE440s menjelaskan
ephemeris yang dibaca oleh Skyfield dalam
SOLUNA.</p><button type="button" class="soluna-edu-
action" data-soluna-open="de440s">Pelajari DE440s
-></button></div>
  </div>
</article>

<article class="soluna-edu-page" data-soluna-
page="de440s" aria-labelledby="soluna-edu-de-title"
hidden>
  <header class="soluna-edu-cover"><div class="soluna-
edu-kicker">DE440s • Ephemeris planet dan Bulan
JPL</div><h2 id="soluna-edu-de-title" tabindex="-
1">DE440s</h2><p>Landasan data posisi dan gerak benda
langit untuk perhitungan Matahari-Bulan dalam
SOLUNA.</p><div class="soluna-edu-tags"><span>NASA /
JPL</span><span>Development
Ephemeris</span><span>Kernel SPK / BSP</span><span>Data
sesuai tahun pilihan</span></div></header>
  <div class="soluna-edu-toolbar"><span>Sumber data •
Cakupan, penggunaan, dan penelitian</span><button
type="button" class="soluna-edu-action" data-soluna-
open="calculator">Buka Perhitungan Astronomi
-></button></div>
  <div class="soluna-edu-body">
    <div class="soluna-edu-
abstract"><strong>Abstrak</strong><p>DE440 adalah
ephemeris planet dan Bulan yang dikembangkan di Jet
Propulsion Laboratory. Data dibangun dari integrasi
numerik yang dicocokkan dengan pengamatan. DE440s
menyediakan cakupan waktu yang lebih pendek dalam
keluarga data tersebut. SOLUNA menggunakan salinan
bagian ephemeris yang sesuai tahun pilihan untuk
memasok posisi Bumi, Matahari, dan Bulan kepada
Skyfield. Pemotongan rentang data membantu mengurangi
beban pemuatan tanpa mengganti model sumber. <a
class="soluna-edu-ref"
href="https://ssd.jpl.nasa.gov/doc/de440_de441.html"
target="_blank" rel="noopener noreferrer">[1]</a><a
class="soluna-edu-ref"
href="https://rhodesmill.org/skyfield/planets.html"
target="_blank" rel="noopener
noreferrer">[2]</a></p></div>
    <nav class="soluna-edu-toc" aria-label="Daftar isi
DE440s"><strong>Daftar isi</strong><div><a

```

[01 Pengertian](#soluna-edu-de-1) 02 Mengapa digunakan 03 BSP & cara baca 04 Pemilihan berkas 05 Cakupan & batas 06 Kajian akademik 07 Daftar pustaka </div> </nav>

<section class="soluna-edu-section" id="soluna-edu-de-1"><h3>01 Apa itu DE440s?</h3><p>Ephemeris adalah representasi posisi dan gerak benda langit sebagai fungsi waktu. Singkatan DE merujuk pada Development Ephemeris, sedangkan 440 menunjukkan seri model. Berkas DE440s merupakan pilihan dengan rentang lebih pendek; dokumentasi Skyfield mencantumkan cakupan sekitar 1849-2150 dan ukuran sekitar 32 MB untuk berkas distribusinya. [2]</p><p>Publikasi Park dan rekan-rekan menjelaskan DE440 dan DE441 sebagai hasil pencocokan orbit yang diintegrasikan secara numerik terhadap pengamatan di darat dan antariksa. Dibanding DE430, pengembangannya menambahkan data pengamatan serta perbaikan model dinamika dan kalibrasi. Ephemeris ini bukan sekadar rumus orbit lingkaran seragam. [1]</p><p>Untuk pengguna SOLUNA, arti praktisnya adalah semua kajian Matahari-Bulan dibangun dari sumber data gerak yang dapat ditelusuri. Nama model perlu dicantumkan dalam laporan agar hasil dapat dibandingkan dengan penelitian yang memakai ephemeris lain.</p></section>

<section class="soluna-edu-section" id="soluna-edu-de-2"><h3>02 Mengapa DE440s digunakan dalam SOLUNA?</h3>

<div class="soluna-edu-grid"><div class="soluna-edu-card"><h4>Data gerak yang terdokumentasi</h4><p>Model sumber memiliki publikasi ilmiah dan dokumentasi JPL. Hal ini memberi landasan yang jelas untuk menjelaskan asal data, mengevaluasi pilihan ephemeris, dan merujuk hasil dalam kegiatan akademik.</p></div><div class="soluna-edu-card"><h4>Kesesuaian dengan kajian Matahari-

Bulan

Posisi Bumi, Matahari, dan Bulan mendasari koordinat, elongasi, fase, konjungsi, serta geometri bayangan. Sumber yang sama digunakan untuk rangkaian kajian sehingga perbandingan tidak mencampur ephemeris tanpa penjelasan.

Dapat dibaca oleh Skyfield

Format BSP/SPK memungkinkan Skyfield memakai data vektor ephemeris dalam operasi astronomi. SOLUNA tidak perlu memuat tabel posisi harian buatan sendiri untuk setiap kota.

Dapat dipilih berdasarkan rentang waktu

Aplikasi mendahulukan berkas tahunan yang kecil, kemudian satu berkas rentang bila diperlukan. Pilihan ini cocok untuk perangkat pengguna yang memiliki kapasitas jaringan atau memori terbatas.

DE440s menyediakan data gerak; Skyfield mengolah data dan kerangka pengamat; SOLUNA menerapkan aturan analisis serta penyajian. Data ephemeris sendiri tidak berisi keputusan kalender Hijriah, tabel visibilitas hilal untuk semua tempat, atau konvensi waktu ibadah.

03 Apa yang ada di dalam BSP?

SPK adalah format kernel untuk data posisi dan gerak benda. Berkas biner SPK sering memakai ekstensi `.bsp`. Data tersusun dalam segmen yang memiliki benda sasaran, pusat acuan, kerangka, jenis representasi, dan batas waktu. Pada SPK tipe 2, posisi direpresentasikan dengan koefisien polinomial Chebyshev per interval; kecepatan dapat diperoleh dari turunannya. https://naif.jpl.nasa.gov/pub/naif/toolkit_docs/C/req/spk.html

[3]

Dengan representasi tersebut, program dapat mengevaluasi posisi pada waktu di antara titik kalender. Sebuah BSP tidak perlu berisi satu baris untuk setiap detik atau satu tabel terpisah untuk setiap kota. Lokasi pengamat ditambahkan pada tahap perhitungan, sehingga data gerak yang sama dapat digunakan bagi Sabang, Jakarta, atau kota lain.

Mempertahankan rantai vektor. Pemotongan BSP harus mempertahankan segmen yang diperlukan untuk menghubungkan Bumi, Bulan, dan Matahari dengan pusat acuannya. Mengambil satu segmen hanya karena namanya "Bulan" belum tentu menyediakan seluruh hubungan yang dibutuhkan. Cakupan waktu semua segmen yang digunakan

juga harus mencakup waktu evaluasi.

Untuk pembelajaran, bayangkan posisi sebagai vektor berantai. Data ephemeris memberi gerak benda terhadap acuannya; lokasi di permukaan Bumi menambah vektor pengamat; perbedaan vektor membentuk garis pandang. Transformasi berikutnya menghasilkan besaran yang dapat dibaca sebagai arah di langit.

<section class="soluna-edu-section" id="soluna-edu-de-4"><h3>04Pemilihan berkas pada versi SOLUNA ini</h3><p>Pemilihan mengikuti tahun tanggal perhitungan. Pada penyiapan pertama, aplikasi memilih menurut tahun berjalan dalam zona waktu lokasi. Ketika tanggal kajian berubah, loader menyiapkan satu kernel yang sesuai. Berkas yang sama digunakan kembali selama pilihan tidak memerlukan kernel lain.</p>

<div class="soluna-edu-table-wrap" role="region" aria-label="Tabel pemilihan BSP, dapat digeser mendatar" tabindex="0"><table><caption>Tabel 3. Berkas yang digunakan menurut tahun pilihan</caption><thead><tr><th>Tahun pilihan</th><th>Pilihan utama</th><th>Cadangan rentang yang sesuai</th></tr></thead><tbody><tr><td>2026</td><td><code>de440s_2026.bsp</code></td><td rowspan="5"><code>de440s_1950_2049.bsp</code>, hanya jika berkas tahunan yang bersangkutan tidak ditemukan.</td></tr><tr><td>2027</td><td><code>de440s_2027.bsp</code></td></tr><tr><td>2028</td><td><code>de440s_2028.bsp</code></td></tr><tr><td>2029</td><td><code>de440s_2029.bsp</code></td></tr><tr><td>2030</td><td><code>de440s_2030.bsp</code></td></tr><tr><td>1850-1949</td><td><code>de440s_1850_1949.bsp</code></td><td>Satu berkas untuk rentang ini.</td></tr><tr><td>1950-2049 selain 2026-2030</td><td><code>de440s_1950_2049.bsp</code></td><td>Satu berkas untuk rentang ini.</td></tr><tr><td>2050-2149</td><td><code>de440s_2050_2149.bsp</code></td><td>Satu berkas untuk rentang ini.</td></tr></tbody></table></div>

<p>Nama tahunan menunjukkan tahun yang dilayani loader. Di dalam berkas juga terdapat penyangga waktu agar pencarian peristiwa di sekitar batas tanggal dan pencarian lanjutan tetap memiliki data. Jadi, "berkas tahunan" tidak berarti semua koefisien dipotong tepat pada pukul 00.00 tanggal 1 Januari dan 31 Desember.</p><p>Berkas sumber besar <code>de440s.bsp</code> tidak dibuka oleh loader versi ini. Scan 100 kota memakai kernel yang dipilih, bukan

memuat satu BSP per kota atau membuka seluruh berkas rentang. Penambahan halaman BERANDA, SKYFIELD, dan DE440s tidak mengubah mekanisme tersebut.

Kecepatan pemuatan dan kecepatan hitungan berbeda. Ukuran berkas yang lebih kecil mengurangi transfer dan pembacaan data. Lama pencarian fase, terbit-terbenam, atau scan banyak kota tetap dipengaruhi jumlah evaluasi, kemampuan perangkat, dan runtime browser. Mengganti halaman bacaan tidak menjalankan scan baru.

05Cakupan waktu dan batas penerapan

Rentang distribusi DE440s dan rentang pilihan loader bukan hal yang identik. Dokumentasi sumber menyebut cakupan sekitar 1849–2150; daftar berkas SOLUNA ini melayani tahun pilihan **1850–2149**. Pengguna perlu memeriksa cakupan data dan pencarian yang diminta, terutama saat peristiwa membutuhkan evaluasi sebelum atau sesudah tanggal pilihan.

Memotong data berarti menyalin bagian rentang dan segmen yang diperlukan dari model sumber. Jika koefisien dipertahankan dan waktu evaluasi berada dalam cakupan yang sama, pemotongan tidak membangun model gerak baru. Pemeriksaan yang tepat adalah membandingkan posisi sumber dan berkas potongan pada saat uji, serta memastikan rantai vektor dan batas waktu lengkap.

Ephemeris tidak memodelkan awan, kecerlangan langit lokal, profil pegunungan pada ufuk, atau kemampuan mata dan instrumen. Faktor tersebut memengaruhi observasi hilal. Penulisan banyak angka di layar juga tidak menetapkan ketelitian nyata suatu klaim; ketelitian harus dinilai terhadap besaran, model, masukan, dan pembanding yang digunakan.

Nama yang mirip dapat memiliki cakupan berbeda. DE440, DE440s, dan DE441 tidak boleh dianggap saling menggantikan hanya dari nama. Pilihan ephemeris harus disesuaikan dengan rentang waktu kajian dan dokumentasi model. Untuk versi SOLUNA ini, gunakan daftar berkas dan aturan loader yang tercantum di atas.

06Penggunaan dalam kajian akademik

Bagian metode laporan dapat menyatakan: *"Posisi Matahari dan Bulan dihitung dengan SOLUNA menggunakan Skyfield dan ephemeris JPL DE440s. Berkas*

dipilih sesuai tahun kajian; lokasi, waktu, serta kerangka koordinat dicatat untuk setiap perhitungan." Lengkapi kalimat tersebut dengan identitas berkas aktual, koordinat, zona waktu, mode, definisi peristiwa, dan aturan analisis yang digunakan.

Untuk praktikum validasi data, pilih beberapa saat pada awal, tengah, dan akhir rentang berkas. Bandingkan posisi menggunakan kondisi yang sama; pisahkan pemeriksaan data ephemeris dari pemeriksaan konvensi seperti refraksi, ambang hilal, dan penamaan tanggal. Jika membandingkan dengan layanan astronomi lain, catat model serta pusat pengamat layanan itu.

Untuk penelusuran ilmiah, rujuk publikasi DE440/DE441 bagi model sumber, dokumentasi SPK bagi format data, dan dokumentasi Skyfield bagi cara pembacaan serta penggunaan dalam kode. Nama file saja belum menjelaskan seluruh metode penelitian.

```
<section class="soluna-edu-section" id="soluna-edu-de-7"><h3><span class="soluna-edu-number">07</span>Daftar pustaka dan sumber resmi</h3><ol class="soluna-edu-bibliography"><li>Park, R. S., Folkner, W. M., Williams, J. G., & Boggs, D. H. (2021). <a href="https://doi.org/10.3847/1538-3881/abd414" target="_blank" rel="noopener noreferrer"><em>The JPL Planetary and Lunar Ephemerides DE440 and DE441</em></a>. <em>The Astronomical Journal</em>, 161</em>, 105. DOI: 10.3847/1538-3881/abd414. <a href="https://ssd.jpl.nasa.gov/doc/de440_de441.html" target="_blank" rel="noopener noreferrer">Ringkasan dan dokumen resmi JPL</a>.</li><li>Dokumentasi Skyfield. <a href="https://rhodesmill.org/skyfield/planets.html" target="_blank" rel="noopener noreferrer"><em>Planetary ephemerides</em></a>: pilihan berkas, cakupan waktu, dan penggunaan data.</li><li>NASA/JPL, Navigation and Ancillary Information Facility. <a href="https://naif.jpl.nasa.gov/pub/naif/toolkit_docs/C/req/spk.html" target="_blank" rel="noopener noreferrer"><em>SPK Required Reading</em></a>: struktur kernel dan representasi data SPK.</li><li>NASA/JPL, Solar System Dynamics. <a href="https://ssd.jpl.nasa.gov/planets/eph_export.html" target="_blank" rel="noopener noreferrer"><em>Planetary and Lunar Ephemerides</em></a>: dokumentasi distribusi ephemeris.</li></ol></section>
```

```
<div class="soluna-edu-end"><p><strong>Kembali ke praktik.</strong> Gunakan sumber data, masukan, dan definisi yang jelas untuk menafsirkan setiap hasil.</p><button type="button" class="soluna-edu-
```

```

action" data-soluna-open="calculator">Buka Perhitungan
Astronomi →</button></div>
</div>
</article>
</section>

<div id="solunaCalculationView" hidden>
<section class="hero"><h2>Posisi, Fenomena & Jadwal
Matahari-Bulan</h2>
<p style="color: white;">
    Web statis berbasis Skyfield di browser. Kernel
    ephemeris JPL
    <b>DE440s</b> dibaca dari direktori yang sama
    sehingga tidak membutuhkan
    eksekusi Python/PHP pada shared hosting. <br/><br/>
    <a href="./soluna.exe"
        target="_blank"
        style="color: white; text-decoration:
underline;">
        Download SOLUNA exe (Windows)</a> & <a
href="./soluna.apk"
        target="_blank"
        style="color: white; text-decoration:
underline;">
        Download SOLUNA apk (Android)
    </a>
</p>
<div id="statusLine" class="statusline">Memulai
aplikasi...</div></section>
<div class="metrics"><div class="metric"><strong
id="mMoonAlt">—</strong><span>Tinggi
Bulan</span></div><div class="metric"><strong
id="mSunAlt">—</strong><span>Tinggi
Matahari</span></div><div class="metric"><strong
id="mElong">—</strong><span>Elongasi
Bulan</span></div><div class="metric"><strong
id="mIllum">—</strong><span>Iluminasi
Bulan</span></div></div>
<div class="cards"><div class="card"><h3>Waktu</h3><div
id="timeCard" class="muted">Belum
dihitung.</div></div><div
class="card"><h3>Lokasi</h3><div id="locCard"
class="muted">Belum dihitung.</div></div><div
class="card"><h3>Status Hilal Lokal</h3><div
id="hilalCard" class="muted">Kriteria layar: tinggi
Bulan  $\geq 5^\circ$  dan elongasi  $\geq 8^\circ$ .</div></div></div>

<section id="globalHijriBox" class="global-hijri card">

```

```

<h3>PENENTUAN TANGGAL 1 HIJRIAH GLOBAL - SCAN 100
KOTA BERDASARKAN KONJUNGSI BULAN TERPILIH</h3>
<div id="globalHijriSummary" class="muted">Belum
dihitung • Sistem akan mencari tanggal konjungsi (H0)
pada bulan terpilih, lalu memindai H0, H1, dan H2 •
Geosentris 5° + 8° • PKG 1: sebelum batas 00.00 UTC •
PKG 2: daratan Amerika dan ijtimak sebelum fajar
Gisborne.</div>
<div class="global-route">Gisborne → Wellington →
Pasifik/Asia → Timur Tengah/Eropa → Amerika → Alaska
Peninsula/Aleutians • total 100 kota</div>
<div id="globalScanTable"></div>
</section>
<div class="tabs" id="tabs"></div>
<section id="pane-summary" class="tabpane
active"></section><section id="pane-sun"
class="tabpane"></section><section id="pane-moon"
class="tabpane"></section><section id="pane-events"
class="tabpane"></section><section id="pane-phenomena"
class="tabpane"></section><section id="pane-schedule"
class="tabpane"></section><section id="pane-eclipse"
class="tabpane"></section><section id="pane-method"
class="tabpane"></section><section id="pane-terms"
class="tabpane"></section>
</div>
<div class="footer">Copyright &copy; Kasmui
(0818294312), 2026. All Rights Reserved</div>
</div></main></div>
<div id="loading" class="loading"><div
class="loadbox"><h3 id="loadTitle">Menyiapkan engine
astronomi</h3><div id="calcSpinner" class="calc-
spinner" aria-hidden="true"></div><div id="loadText"
class="muted">Memuat...</div><div class="progress"
style="margin-top:14px"><div id="loadBar"
class="bar"></div></div></div></div>
<div id="toast" class="toast"></div>

<script type="text/plain" id="py-engine">
from __future__ import annotations

from dataclasses import dataclass
from datetime import datetime, timedelta, time as
dt_time, timezone
from zoneinfo import ZoneInfo
import calendar
import math
import json

import numpy as np

```

```

# Browser/Pyodide compatibility: SOLUNA tidak memakai
satelit/TLE.
import sys, types
_sgp4lib_stub = types.ModuleType('skyfield.sgp4lib')
_sgp4lib_stub.EarthSatellite = None
sys.modules.setdefault('skyfield.sgp4lib',
_sgp4lib_stub)

from skyfield import almanac, eclipselib
from skyfield.api import Loader, load_file,
load_constellation_map, load_constellation_names, wgs84
from skyfield.framelib import ecliptic_frame

from pathlib import Path

def user_data_dir() -> Path:
    path = Path('/home/pyodide')
    path.mkdir(parents=True, exist_ok=True)
    return path

# Daftar nama/rentang saja; tidak membuka atau memuat
semua BSP.
EPHEMERIS_KERNELS =
json.loads(globals().get('SOLUNA_EPHEMERIS_CONFIG',
'[]'))

def ensure_selected_ephemeris(path_text=None):
    selected_path = path_text if path_text is not None
    else globals().get('SOLUNA_EPHEMERIS_PATH')
    if not selected_path:
        raise RuntimeError('BSP tahun pilihan belum
ditenentukan. Tidak ada kernel cadangan.')
    path = Path(selected_path)
    if path.name.lower() in ('de440s.bsp',
'de440s,bsp'):
        raise RuntimeError('BSP asli tidak boleh
diakses. Gunakan BSP tahunan atau rentang tahun.')
    if not any(kernel['file'] == path.name for kernel
in EPHEMERIS_KERNELS):
        raise RuntimeError('BSP harus berasal dari
pemetaan tahun pilihan.')
    if not path.is_file():
        raise RuntimeError('BSP tahun pilihan belum
tersedia di filesystem browser.')
    return path, 'browser-fetch'

def require_ephemeris_for_date(date_text):
    year = datetime.strptime(date_text, '%Y-%m-
%d').year

```

```

# Tahunan menjadi prioritas di browser. Rentang
hanya sah jika mencakup tahun pilihan.
    kernels = [k for k in EPHEMERIS_KERNELS if
k['year_min'] <= year <= k['year_max']]
    if not any(_ENGINE.ephemeris_path.name ==
kernel['file'] for kernel in kernels):
        raise RuntimeError('Kernel aktif tidak sesuai
tahun pilihan. Pilih BSP yang sesuai sebelum
menghitung.')

AU_KM = 149_597_870.700
MEAN_MOON_DISTANCE_KM = 384_400.0
SUN_RADIUS_KM = 695_700.0
MOON_RADIUS_KM = 1_737.4
EARTH_EQUATORIAL_RADIUS_KM = 6_378.137

# Scan global KHGT: 100 kota/lokasi observasi dari
Gisborne/Wellington sampai Alaska barat.
# Waktu evaluasi memakai sunset lokal H0-H2; H0
beracuan tanggal UTC ijtimak.
# Penentuan global selalu geosentris dan mengikuti
filter PKG 1/PKG 2.
GLOBAL_HIJRI_SCAN_CITIES = [
    ('Gisborne, New Zealand', -38.6533333, 178.0041667,
15.0, 'Pacific/Auckland'),
    ('Wellington, New Zealand', -41.2866428,
174.7755718, 13.0, 'Pacific/Auckland'),
    ('Auckland, New Zealand', -36.8485300, 174.7634900,
43.0, 'Pacific/Auckland'),
    ('Lord Howe Island, Australia', -31.5500000,
159.0800000, 5.0, 'Australia/Lord_Howe'),
    ('Byron Bay, Australia', -28.6498900, 153.6124630,
6.0, 'Australia/Sydney'),
    ('Sydney, Australia', -33.8678500, 151.2073230,
58.0, 'Australia/Sydney'),
    ('Hobart, Australia', -42.8793606, 147.3294067,
20.0, 'Australia/Hobart'),
    ('Port Moresby, Papua New Guinea', -9.4438000,
147.1803000, 35.0, 'Pacific/Port_Moresby'),
    ('Jayapura, Indonesia', -2.5626262, 140.6926331,
95.0, 'Asia/Jayapura'),
    ('Ambon, Indonesia', -3.6943237, 128.1813034, 11.0,
'Asia/Jayapura'),
    ('Makassar, Indonesia', -5.1332156, 119.4078369,
6.0, 'Asia/Makassar'),
    ('Denpasar, Indonesia', -8.6544589, 115.2165333,
32.0, 'Asia/Makassar'),
    ('Pontianak, Indonesia', -0.0220864, 109.3381039,
5.0, 'Asia/Jakarta'),

```

```

('Jakarta, Indonesia', -6.1778016, 106.8284444,
8.0, 'Asia/Jakarta'),
('Singapore', 1.3521000, 103.8198000, 15.0,
'Asia/Singapore'),
('Kuala Lumpur, Malaysia', 3.1390000, 101.6869000,
22.0, 'Asia/Kuala_Lumpur'),
('Bangkok, Thailand', 13.7563000, 100.5018000, 5.0,
'Asia/Bangkok'),
('Banda Aceh, Indonesia', 5.5500499, 95.3178267,
11.0, 'Asia/Jakarta'),
('Dhaka, Bangladesh', 23.8103000, 90.4125000, 4.0,
'Asia/Dhaka'),
('Karachi, Pakistan', 24.8607000, 67.0011000, 8.0,
'Asia/Karachi'),
('Muscat, Oman', 23.5880000, 58.3829000, 15.0,
'Asia/Muscat'),
('Riyadh, Saudi Arabia', 24.6877323, 46.7218494,
612.0, 'Asia/Riyadh'),
('Makkah, Saudi Arabia', 21.4266400, 39.8256320,
277.0, 'Asia/Riyadh'),
('Madinah, Saudi Arabia', 24.4686111, 39.6141667,
625.0, 'Asia/Riyadh'),
('Cairo, Egypt', 30.0626314, 31.2496662, 23.0,
'Africa/Cairo'),
('Istanbul, Turkey', 41.0138430, 28.9496613, 40.0,
'Europe/Istanbul'),
('Rome, Italy', 41.9028000, 12.4964000, 21.0,
'Europe/Rome'),
('Lisbon, Portugal', 38.7223000, -9.1393000, 2.0,
'Europe/Lisbon'),
('Dakar, Senegal', 14.7167000, -17.4677000, 22.0,
'Africa/Dakar'),
('Recife, Brazil', -8.0476000, -34.8770000, 10.0,
'America/Recife'),
('Paramaribo, Suriname', 5.8520000, -55.2038000,
3.0, 'America/Paramaribo'),
('New York, USA', 40.7128000, -74.0060000, 10.0,
'America/New_York'),
('Chicago, USA', 41.8781000, -87.6298000, 181.0,
'America/Chicago'),
('Los Angeles, USA', 34.0522000, -118.2437000,
71.0, 'America/Los_Angeles'),
('San Francisco, USA', 37.7749000, -122.4194000,
16.0, 'America/Los_Angeles'),
('Ketchikan, Alaska, USA', 55.3571450, -
131.6745520, 24.0, 'America/Sitka'),
('Juneau, Alaska, USA', 58.3019444, -134.4197222,
17.0, 'America/Juneau'),

```

```

    ('Skagway, Alaska, USA', 59.4644310, -135.2757200,
9.0, 'America/Juneau'),
    ('Sitka, Alaska, USA', 57.1932040, -135.3673960,
8.0, 'America/Sitka'),
    ('Haines, Alaska, USA', 59.2358000, -135.4450000,
11.0, 'America/Juneau'),
    ('Yakutat, Alaska, USA', 59.5635880, -139.6061910,
9.0, 'America/Yakutat'),
    ('Kaktovik, Alaska, USA', 70.1214520, -143.6294350,
5.0, 'America/Anchorage'),
    ('Fort Yukon, Alaska, USA', 66.5884060, -
145.2276420, 132.0, 'America/Anchorage'),
    ('Delta Junction, Alaska, USA', 64.0378000, -
145.7322000, 354.0, 'America/Anchorage'),
    ('Cordova, Alaska, USA', 60.5424000, -145.7525000,
18.0, 'America/Anchorage'),
    ('Valdez, Alaska, USA', 61.0836210, -146.3171790,
30.0, 'America/Anchorage'),
    ('Fairbanks, Alaska, USA', 64.8365310, -
147.6517450, 136.0, 'America/Anchorage'),
    ('Whittier, Alaska, USA', 60.7773180, -148.6637130,
13.0, 'America/Anchorage'),
    ('Palmer, Alaska, USA', 61.5971658, -149.1146826,
71.0, 'America/Anchorage'),
    ('Seward, Alaska, USA', 60.1247860, -149.3915810,
27.0, 'America/Anchorage'),
    ('Wasilla, Alaska, USA', 61.5811360, -149.4607900,
104.0, 'America/Anchorage'),
    ('Anchorage, Alaska, USA', 61.2180556, -
149.9002778, 31.0, 'America/Anchorage'),
    ('Soldotna, Alaska, USA', 60.4853180, -151.0690140,
32.0, 'America/Anchorage'),
    ('Kenai, Alaska, USA', 60.5622400, -151.2080600,
22.0, 'America/Anchorage'),
    ('Homer, Alaska, USA', 59.6425000, -151.5483000,
29.0, 'America/Anchorage'),
    ('Seldovia, Alaska, USA', 59.4394800, -151.7103870,
9.0, 'America/Anchorage'),
    ('Kodiak, Alaska, USA', 57.7930014, -152.3974812,
15.0, 'America/Anchorage'),
    ('Ouzinkie, Alaska, USA', 57.9227490, -152.4696780,
4.0, 'America/Anchorage'),
    ('Port Lions, Alaska, USA', 57.7819396, -
152.8438639, 6.0, 'America/Anchorage'),
    ('Old Harbor, Alaska, USA', 57.2115416, -
153.3266210, 4.0, 'America/Anchorage'),
    ('King Salmon, Alaska, USA', 58.7352270, -
156.5285440, 17.0, 'America/Anchorage'),

```

('Utqiagvik, Alaska, USA', 71.2538610, -
 156.7982460, 3.0, 'America/Anchorage'),
 ('South Naknek, Alaska, USA', 58.6634084, -
 156.9916713, 5.0, 'America/Anchorage'),
 ('Naknek, Alaska, USA', 58.7283000, -157.0133000,
 10.0, 'America/Anchorage'),
 ('Egegik, Alaska, USA', 58.2051480, -157.5219780,
 8.0, 'America/Anchorage'),
 ('Pilot Point, Alaska, USA', 57.5799176, -
 157.7657334, 5.0, 'America/Anchorage'),
 ('Chignik, Alaska, USA', 56.2951800, -158.3917840,
 15.0, 'America/Anchorage'),
 ('Chignik Lagoon, Alaska, USA', 56.3042670, -
 158.5002590, 15.0, 'America/Anchorage'),
 ('Dillingham, Alaska, USA', 59.0604670, -
 158.5322510, 29.0, 'America/Anchorage'),
 ('Port Heiden, Alaska, USA', 56.9425144, -
 158.6048543, 29.0, 'America/Anchorage'),
 ('Aleknagik, Alaska, USA', 59.2889610, -
 158.6643430, 11.0, 'America/Anchorage'),
 ('Chignik Lake, Alaska, USA', 56.2732210, -
 158.7933760, 15.0, 'America/Anchorage'),
 ('Perryville, Alaska, USA', 55.9329003, -
 159.1558029, 3.0, 'America/Anchorage'),
 ('Ivanof Bay, Alaska, USA', 55.9181990, -
 159.4932980, 5.0, 'America/Anchorage'),
 ('Wainwright, Alaska, USA', 70.6582320, -
 159.9852140, 12.0, 'America/Anchorage'),
 ('Togiak, Alaska, USA', 59.0619000, -160.3761000,
 5.0, 'America/Anchorage'),
 ('Sand Point, Alaska, USA', 55.3134700, -
 160.4843440, 7.0, 'America/Anchorage'),
 ('Unalakleet, Alaska, USA', 63.8942060, -
 160.7967750, 6.0, 'America/Nome'),
 ('Koyuk, Alaska, USA', 64.9397000, -161.1549000,
 10.0, 'America/Nome'),
 ('Shaktoolik, Alaska, USA', 64.3544810, -
 161.1935140, 7.0, 'America/Nome'),
 ('Bethel, Alaska, USA', 60.7939050, -161.7963220,
 3.0, 'America/Anchorage'),
 ('Elim, Alaska, USA', 64.6182260, -162.2765600,
 11.0, 'America/Nome'),
 ('King Cove, Alaska, USA', 55.0849930, -
 162.3168060, 27.0, 'America/Anchorage'),
 ('Kotzebue, Alaska, USA', 66.8983000, -162.5967000,
 6.0, 'America/Nome'),
 ('Cold Bay, Alaska, USA', 55.2065680, -162.7149874,
 30.0, 'America/Anchorage'),

```

        ('False Pass, Alaska, USA', 54.8724410, -
163.3342680, 6.0, 'America/Anchorage'),
        ('White Mountain, Alaska, USA', 64.6823590, -
163.4127530, 17.0, 'America/Nome'),
        ('Nome, Alaska, USA', 64.5011110, -165.4063890,
6.0, 'America/Nome'),
        ('Akutan, Alaska, USA', 54.1399790, -165.7630260,
20.0, 'America/Anchorage'),
        ('Brevig Mission, Alaska, USA', 65.3431960, -
166.5089460, 10.0, 'America/Nome'),
        ('Unalaska, Alaska, USA', 53.9102550, -166.5290290,
4.0, 'America/Anchorage'),
        ('Point Hope, Alaska, USA', 68.3445542, -
166.6744850, 5.0, 'America/Anchorage'),
        ('Wales, Alaska, USA', 65.6132320, -168.0683130,
8.0, 'America/Nome'),
        ('Diomedede, Alaska, USA', 65.7539240, -168.9225770,
28.0, 'America/Nome'),
        ('Saint George, Alaska, USA', 56.5899226, -
169.6281114, 10.0, 'America/Nome'),
        ('Saint Paul, Alaska, USA', 57.1817441, -
170.2751293, 9.0, 'America/Nome'),
        ('Savoonga, Alaska, USA', 63.6797060, -170.4833240,
14.0, 'America/Nome'),
        ('Gambell, Alaska, USA', 63.7797000, -171.7333000,
8.0, 'America/Nome'),
        ('Atka, Alaska, USA', 52.2115460, -174.2134520,
18.0, 'America/Adak'),
        ('Adak, Alaska, USA', 51.9098890, -176.5980670,
7.0, 'America/Adak'),
]

```

```

# PKG global memakai geosentris, terlepas dari mode
hasil astronomi lokal.

```

```

# Fajar untuk tanggal awal bulan yang diuji: pusat
Matahari toposentris -18°.

```

```

KHGT_GISBORNE_DAWN_ALTITUDE_DEG = -18.0

```

```

# Daftar utama 100 kota dipertahankan. Titik tambahan
hanya diverifikasi jika

```

```

# PKG 1 belum ditemukan pada daftar utama; semuanya
berada di daratan Amerika.

```

```

# Sampel ini tidak mengklaim memetakan setiap titik
daratan di seluruh Bumi.

```

```

KHGT_AMERICA_VALIDATION_SITES = [

```

```

    ('Tijuana, Mexico', 32.5149, -117.0382, 20.0,
'America/Tijuana'),

```

```

('Hermosillo, Mexico', 29.072967, -110.955919,
210.0, 'America/Hermosillo'),
('Mazatlan, Mexico', 23.2494, -106.4111, 10.0,
'America/Mazatlan'),
('Manzanillo, Mexico', 19.1138, -104.3385, 10.0,
'America/Mexico_City'),
('Acapulco, Mexico', 16.8531, -99.8237, 20.0,
'America/Mexico_City'),
('San Salvador, El Salvador', 13.6929, -89.2182,
658.0, 'America/El_Salvador'),
('Liberia, Costa Rica', 10.6346, -85.4407, 144.0,
'America/Costa_Rica'),
('Panama City, Panama', 8.9824, -79.5199, 10.0,
'America/Panama'),
('Esmeraldas, Ecuador', 0.9592, -79.65397, 15.0,
'America/Guayaquil'),
('Manta, Ecuador', -0.96765, -80.7089, 20.0,
'America/Guayaquil'),
('Guayaquil, Ecuador', -2.170998, -79.922359, 4.0,
'America/Guayaquil'),
('Quito, Ecuador', -0.1807, -78.4678, 2850.0,
'America/Guayaquil'),
('Trujillo, Peru', -8.1116, -79.0288, 34.0,
'America/Lima'),
('Lima, Peru', -12.0464, -77.0428, 154.0,
'America/Lima'),
('Arequipa, Peru', -16.409, -71.5375, 2335.0,
'America/Lima'),
('Arica, Chile', -18.4783, -70.3126, 20.0,
'America/Santiago'),
('Iquique, Chile', -20.2307, -70.1357, 20.0,
'America/Santiago'),
('Antofagasta, Chile', -23.6509, -70.3975, 40.0,
'America/Santiago'),
('La Serena, Chile', -29.9027, -71.2519, 28.0,
'America/Santiago'),
('Valparaiso, Chile', -33.0472, -71.6127, 20.0,
'America/Santiago'),
('Santiago, Chile', -33.4489, -70.6693, 520.0,
'America/Santiago'),
('Concepcion, Chile', -36.8201, -73.0444, 12.0,
'America/Santiago'),
('Valdivia, Chile', -39.8142, -73.2459, 5.0,
'America/Santiago'),
('Puerto Montt, Chile', -41.4693, -72.9424, 14.0,
'America/Santiago'),
('Coyhaique, Chile', -45.5712, -72.0683, 300.0,
'America/Santiago'),

```

```

        ('Punta Arenas, Chile', -53.1638, -70.9171, 30.0,
 'America/Punta_Arenas'),
        ('Neuquen, Argentina', -38.9516, -68.0591, 260.0,
 'America/Argentina/Salta'),
        ('Bariloche, Argentina', -41.1335, -71.3103, 893.0,
 'America/Argentina/Salta'),
        ('Comodoro Rivadavia, Argentina', -45.8641, -
 67.4966, 60.0, 'America/Argentina/Catamarca'),
        ('Rio Gallegos, Argentina', -51.6230, -69.2168,
 20.0, 'America/Argentina/Rio_Gallegos'),
    ]

```

```

# Pulau-pulau pada daftar lama tetap boleh menjadi
saksi PKG 1, tetapi bukan
# saksi daratan utama Benua Amerika untuk PKG 2. NYC
berada di Manhattan.
KHGT_AMERICAN_ISLAND_SITES = frozenset({
    'New York, USA', 'Ketchikan, Alaska, USA', 'Sitka,
Alaska, USA',
    'Kodiak, Alaska, USA', 'Ouzinkie, Alaska, USA',
    'Port Lions, Alaska, USA',
    'Old Harbor, Alaska, USA', 'Sand Point, Alaska,
USA', 'False Pass, Alaska, USA',
    'Akutan, Alaska, USA', 'Unalaska, Alaska, USA',
    'Diomedede, Alaska, USA',
    'Saint George, Alaska, USA', 'Saint Paul, Alaska,
USA',
    'Savoonga, Alaska, USA', 'Gambell, Alaska, USA',
    'Atka, Alaska, USA', 'Adak, Alaska, USA',
})

```

```

def khgt_pkg_status(*, conjunction_utc, sunset_utc,
 utc_midnight,
                    gisborne_dawn_utc,
numeric_qualifies, america_mainland):
    """Keputusan pada satu sunset untuk hari D, dengan
batas D+1 00:00 UTC.

```

```

    Perbandingan memakai datetime sadar-zona, bukan
string jam lokal. Fajar
    adalah fajar Gisborne pada tanggal D+1, bukan fajar
pertama sesudah ijtimak.
    """

```

```

    if sunset_utc is None:
        return 'NO SUNSET', False
    if not (utc_midnight - timedelta(days=1) <=
sunset_utc < utc_midnight + timedelta(days=1)):
        return 'DI LUAR JENDELA HARI PKG', False

```

```

if sunset_utc < conjunction_utc:
    return 'PRA-IJTIMAK', False
if not numeric_qualifies:
    return 'BELUM 5° + 8°', False
if sunset_utc < utc_midnight:
    return 'PKG 1', True
if not america_mainland:
    return 'DI LUAR DARATAN AMERIKA (PKG 2)', False
if gisborne_dawn_utc is None:
    return 'PKG 2: FAJAR TIDAK TERSEDIA', False
if conjunction_utc < gisborne_dawn_utc:
    return 'PKG 2', True
return 'PKG 2 DITUNDA: IJTIMAK TIDAK SEBELUM
FAJAR', False

```

```

@dataclass
class BodyResult:
    name: str
    altitude_deg: float
    refracted_altitude_deg: float | None
    azimuth_deg: float
    hour_angle_hours: float
    ra_hours: float
    dec_deg: float
    ecliptic_longitude_deg: float
    ecliptic_latitude_deg: float
    constellation: str
    elongation_deg: float
    illumination_pct: float
    phase_angle_deg: float
    phase_cycle_deg: float | None
    phase_name: str
    waxing_waning: str
    bright_limb_position_angle_deg: float | None
    magnitude: float
    distance_km: float
    distance_au: float
    observer_distance_km: float
    angular_diameter_arcmin: float
    semidiameter_arcmin: float
    horizontal_parallax_deg: float

```

```

@dataclass
class EventResult:
    sun_rise: str
    sun_rise_azimuth_deg: float | None
    sun_set: str

```

```

sun_set_azimuth_deg: float | None
moon_rise: str
moon_rise_azimuth_deg: float | None
moon_set: str
moon_set_azimuth_deg: float | None
sun_transit: str
sun_transit_altitude_deg: float | None
moon_transit: str
moon_transit_altitude_deg: float | None
civil_dawn: str
nautical_dawn: str
astronomical_dawn: str
civil_dusk: str
nautical_dusk: str
astronomical_dusk: str
day_length_hours: float | None
equation_of_time_minutes: float | None
local_sidereal_time_hours: float
twilight_state: str
conjunction_geocentric: str
conjunction_topocentric: str
conjunction_topocentric_min_elong_deg: float | None
moon_age_days: float
lag_time_minutes: float | None

```

```

@dataclass
class PhenomenonItem:
    category: str
    name: str
    value: str
    description: str

```

```

@dataclass
class ScheduleItem:
    name: str
    value: str
    basis: str
    note: str

```

```

@dataclass
class GlobalScanCityResult:
    name: str
    timezone_name: str
    sunset_local: datetime | None
    moon_altitude_deg: float | None
    sun_altitude_deg: float | None

```

```

elongation_deg: float | None
illumination_pct: float | None
qualifies: bool
h_day: int
scan_date: str
sunset_utc: datetime | None = None
pkg_status: str = "BELUM MEMENUHI"
pkg_accepted: bool = False
america_mainland: bool = False

```

```

@dataclass
class GlobalHijriResult:
    selected_date: str
    mode: str
    qualifies: bool
    matched_count: int
    total_count: int
    hijri_start_gregorian_date: str | None
    first_match: GlobalScanCityResult | None
    cities: list[GlobalScanCityResult]
    window_eligible: bool
    h_day: int | None
    conjunction_local_datetime: datetime | None
    reference_timezone_name: str
    decision_pkg: str | None = None
    decision_reason: str = ""
    utc_reference_date: str | None = None
    utc_midnight: datetime | None = None
    gisborne_dawn_local_datetime: datetime | None =
None
    gisborne_dawn_utc_datetime: datetime | None = None
    gisborne_dawn_altitude_deg: float = -18.0
    conjunction_before_gisborne_dawn: bool | None =
None
    validation_sites: list[GlobalScanCityResult] | None
= None
    daily_decisions: list[dict] | None = None

```

```

@dataclass
class CalculationResult:
    local_datetime: datetime
    utc_datetime: datetime
    time_mode: str
    mode: str
    location_name: str
    latitude: float
    longitude: float

```

```

        timezone_name: str
        sun: BodyResult
        moon: BodyResult
        relative_altitude_deg: float
        relative_azimuth_deg: float
        events: EventResult
        phenomena: list[PhenomenonItem]
        schedule: list[ScheduleItem]
        global_hijri: GlobalHijriResult
        eclipses: dict

class AstronomyEngine:
    def __init__(self):
        # Timescale memakai data bawaan. Hanya BSP
        # pilihan dari browser yang dibuka;
        # tidak ada unduhan otomatis atau pencarian
        # kernel cadangan.
        self.loader = Loader(str(user_data_dir() /
"skyfield"))
        self.ts = self.loader.timescale(builtin=True)
        self.ephemeris_path, self.ephemeris_source =
ensure_selected_ephemeris()
        self.eph = load_file(str(self.ephemeris_path))
        self.earth = self.eph["earth"]
        self.sun = self.eph["sun"]
        self.moon = self.eph["moon"]

        try:
            self.constellation_at =
load_constellation_map()
            self.constellation_names =
dict(load_constellation_names())
        except Exception:
            self.constellation_at = None
            self.constellation_names = {}

    @staticmethod
    def parse_local_datetime(date_text: str, time_text:
str, timezone_name: str) -> datetime:
        date_value =
datetime.strptime(date_text.strip(), "%Y-%m-%d").date()
        raw_time = time_text.strip()
        parsed = None
        for fmt in ("%H:%M:%S", "%H:%M"):
            try:
                parsed = datetime.strptime(raw_time,
fmt).time()
            except:
                break

```

```

        except ValueError:
            continue
    if parsed is None:
        raise ValueError("Format waktu harus HH:MM
atau HH:MM:SS.")
    tz = ZoneInfo(timezone_name)
    return datetime.combine(date_value, parsed,
tzinfo=tz)

def calculate(
    self,
    *,
    location_name: str,
    latitude: float,
    longitude: float,
    elevation_m: float,
    timezone_name: str,
    date_text: str,
    time_text: str,
    time_mode: str = "Jam Manual",
    mode: str = "Toposentris",
) -> CalculationResult:
    topos = wgs84.latlon(latitude, longitude,
elevation_m=elevation_m)
    observer = self.earth + topos

    if time_mode == "Maghrib Lokal":
        local_dt = self._local_sunset_datetime(
            date_text=date_text,
            timezone_name=timezone_name,
            observer=observer,
        )
    else:
        local_dt =
self.parse_local_datetime(date_text, time_text,
timezone_name)
        time_mode = "Jam Manual"

    utc_dt = local_dt.astimezone(timezone.utc)
    t = self.ts.from_datetime(utc_dt)

    # Compute one authoritative topocentric
reference for the selected instant.
    # This is reused by the main result when
mode=Toposentris and by the
    # monthly hilal audit when the selected instant
is Maghrib Lokal.
    topo_sun_reference = self._body_result(

```

```

        "Matahari", self.sun, t, observer, topos,
latitude, longitude, "Toposentris"
    )
    topo_moon_reference = self._body_result(
        "Bulan", self.moon, t, observer, topos,
latitude, longitude, "Toposentris"
    )

    if mode == "Toposentris":
        sun_result = topo_sun_reference
        moon_result = topo_moon_reference
    else:
        sun_result = self._body_result(
            "Matahari", self.sun, t, observer,
topos, latitude, longitude, mode
        )
        moon_result = self._body_result(
            "Bulan", self.moon, t, observer, topos,
latitude, longitude, mode
        )

    events = self._events(
        local_dt=local_dt,
        observer=observer,
        topos=topos,
        longitude=longitude,
        timezone_name=timezone_name,
    )

    rel_alt = moon_result.altitude_deg -
sun_result.altitude_deg
    rel_az =
self._signed_angle_diff(moon_result.azimuth_deg,
sun_result.azimuth_deg)

    phenomena = self._phenomena(
        current_utc=utc_dt,
        local_dt=local_dt,
        observer=observer,
        topos=topos,
        timezone_name=timezone_name,
        sun_result=sun_result,
        moon_result=moon_result,
        twilight_state=events.twilight_state,
    )

    # Monthly hilal screening follows the SAME
coordinate mode selected

```

```

        # by the user. The sunset instant itself
        remains a local/topocentric
        # event, but the displayed Moon altitude, Sun
        altitude, elongation,
        # and illumination are evaluated in either
        Toposentris or Geosentris
        # according to `mode`. This is essential when
        the monthly filter is
        # used to test a criterion whose numerical
        values depend on the chosen
        # coordinate frame.
        if time_mode == "Maghrib Lokal":
            phenomena.append(PhenomenonItem(
                "Konsistensi Data",
                "Referensi kartu vs Kandidat Hilal",
                f"SAMA: {mode} @ Maghrib Lokal",
                "Tanggal/waktu terpilih memakai
kerangka koordinat dan sumber nilai yang sama. Baris
tanggal terpilih pada Kandidat Hilal wajib identik
dalam toleransi numerik.",
            ))
        else:
            phenomena.append(PhenomenonItem(
                "Konsistensi Data",
                "Referensi kartu vs Kandidat Hilal",
                f"SAMA KERANGKA ({mode}), BERBEDA
WAKTU",
                "Kartu utama memakai Jam Manual,
sedangkan Kandidat Hilal memakai Maghrib lokal.
Kerangka koordinatnya sama, tetapi waktu perhitungannya
berbeda sehingga angka tidak boleh dibandingkan
langsung.",
            ))

        phenomena.extend(
            self._monthly_hilal_candidates(
                selected_local_dt=local_dt,
                observer=observer,
                latitude=latitude,
                longitude=longitude,
                timezone_name=timezone_name,
                selected_time_mode=time_mode,
                selected_display_mode=mode,
                selected_sun=sun_result,
                selected_moon=moon_result,
            )
        )

        global_hijri = self._global_hijri_scan(

```

```

        date_text=date_text,
        mode=mode,
        reference_timezone_name=timezone_name,
        min_moon_altitude_deg=5.0,
        min_elongation_deg=8.0,
    )

    schedule = self._prayer_schedule(
        local_dt=local_dt,
        observer=observer,
        latitude=latitude,
        timezone_name=timezone_name,
    )

    # Laporan gerhana untuk bulan dan tahun dari
    tanggal yang dipilih.
    # Perhitungan tetap memakai ephemeris DE440s
    yang sama dengan modul utama.
    eclipses = self._eclipse_report(
        selected_local_dt=local_dt,
        observer=observer,
        timezone_name=timezone_name,
    )

    self._validate_core_result(
        sun=sun_result,
        moon=moon_result,
        mode=mode,
        time_mode=time_mode,
        relative_altitude_deg=rel_alt,
    )

    return CalculationResult(
        local_datetime=local_dt,
        utc_datetime=utc_dt,
        time_mode=time_mode,
        mode=mode,
        location_name=location_name,
        latitude=latitude,
        longitude=longitude,
        timezone_name=timezone_name,
        sun=sun_result,
        moon=moon_result,
        relative_altitude_deg=rel_alt,
        relative_azimuth_deg=rel_az,
        events=events,
        phenomena=phenomena,
        schedule=schedule,
        global_hijri=global_hijri,
    )

```

```

        eclipses=eclipses,
    )

def _validate_core_result(
    self,
    *,
    sun: BodyResult,
    moon: BodyResult,
    mode: str,
    time_mode: str,
    relative_altitude_deg: float,
) -> None:
    """Fail fast if a core Sun/Moon result is
internally inconsistent."""
    problems: list[str] = []

    for body in (sun, moon):
        if not (-90.000001 <= body.altitude_deg <=
90.000001):
            problems.append(f"{body.name} altitude
di luar rentang: {body.altitude_deg}")
        if not (0.0 <= body.azimuth_deg <
360.000001):
            problems.append(f"{body.name} azimuth di
luar rentang: {body.azimuth_deg}")
        if not (-90.000001 <= body.dec_deg <=
90.000001):
            problems.append(f"{body.name} deklinasi
di luar rentang: {body.dec_deg}")
        if not (0.0 <= body.ra_hours < 24.000001):
            problems.append(f"{body.name} RA di
luar rentang: {body.ra_hours}")
        if not (0.0 <= body.illumination_pct <=
100.000001):
            problems.append(f"{body.name} iluminasi
di luar rentang: {body.illumination_pct}")
        if body.distance_km <= 0 or
body.observer_distance_km <= 0:
            problems.append(f"{body.name} jarak
tidak valid")
        if body.angular_diameter_arcmin <= 0 or
body.semidiаметer_arcmin <= 0:
            problems.append(f"{body.name} diameter
sudut tidak valid")

        if not (0.0 <= moon.elongation_deg <=
180.000001):
            problems.append(f"Elongasi Bulan di luar
rentang: {moon.elongation_deg}")

```

```

        if not (0.0 <= moon.phase_angle_deg <=
180.000001):
            problems.append(f"Sudut fase Bulan di luar
rentang: {moon.phase_angle_deg}")

        # The illuminated fraction and phase angle are
two representations of
        # the same geometry. They should agree to much
better than 0.1%.
        expected_illum = 50.0 * (1.0 +
math.cos(math.radians(moon.phase_angle_deg)))
        if abs(expected_illum - moon.illumination_pct)
> 0.1:
            problems.append(
                "Iluminasi Bulan tidak konsisten dengan
sudut fase "
                f"(calc={moon.illumination_pct:.6f}%,
phase={expected_illum:.6f}%) "
            )

        if abs((moon.altitude_deg - sun.altitude_deg) -
relative_altitude_deg) > 1e-9:
            problems.append("Beda tinggi Bulan-Matahari
tidak konsisten")

        # Skyfield's default solar setting uses the
USNO sunset definition,
        # where the Sun's geometric center is about -
0.8333 degree. Only
        # enforce this for a real Toposentris Maghrib
calculation.
        if mode == "Toposentris" and time_mode ==
"Maghrib Lokal":
            if abs(sun.altitude_deg - (-0.8333)) >
0.03:
                problems.append(
                    "Tinggi Matahari pada Maghrib tidak
sesuai event sunset "
                    f"(didapat {sun.altitude_deg:.6f}°,
harapan sekitar -0.8333°) "
                )

        if problems:
            raise RuntimeError(
                "Audit konsistensi astronomi gagal: " +
" | ".join(problems)
            )

    def local_sunset_datetime(

```

```

        self,
        *,
        latitude: float,
        longitude: float,
        elevation_m: float,
        timezone_name: str,
        date_text: str,
    ) -> datetime:
        """Public helper for UI preview of the selected
        city's local sunset."""
        topos = wgs84.latlon(latitude, longitude,
        elevation_m=elevation_m)
        observer = self.earth + topos
        return self._local_sunset_datetime(
            date_text=date_text,
            timezone_name=timezone_name,
            observer=observer,
        )

    def _local_sunset_datetime(self, *, date_text: str,
        timezone_name: str, observer) -> datetime:
        date_value =
        datetime.strptime(date_text.strip(), "%Y-%m-%d").date()
        tz = ZoneInfo(timezone_name)
        day_start_local = datetime.combine(date_value,
        dt_time.min, tzinfo=tz)
        day_end_local = day_start_local +
        timedelta(days=1)
        t0 =
        self.ts.from_datetime(day_start_local.astimezone(timezo
        ne.utc))
        t1 =
        self.ts.from_datetime(day_end_local.astimezone(timezone
        .utc))

        times, flags = almanac.find_settings(observer,
        self.sun, t0, t1)
        sunset = self._first_real_event(times, flags,
        tz)

        if sunset is None:
            raise ValueError(
                "Waktu Maghrib/sunset tidak ditemukan
        untuk kota dan tanggal ini. "
                "Gunakan Jam Manual atau pilih tanggal
        lain."
            )
        return sunset

```

```

    def _apparent_position(self, body, t, observer,
mode: str):
        if mode == "Toposentris":
            return
        observer.at(t).observe(body).apparent()
            return
        self.earth.at(t).observe(body).apparent()

    def _body_result(self, name, body, t, observer,
topos, latitude, longitude, mode):
        pos = self._apparent_position(body, t,
observer, mode)
        sun_pos = self._apparent_position(self.sun, t,
observer, mode)

        ra, dec, _ = pos.radec(epoch=t)
        ra_hours = float(ra.hours)
        dec_deg = float(dec.degrees)

        if mode == "Toposentris":
            alt, az, _ = pos.altaz()
            altitude_deg = float(alt.degrees)
            azimuth_deg = float(az.degrees)
            try:
                refr_alt, _, _ =
pos.altaz(temperature_C="standard")
                refracted_altitude_deg =
float(refr_alt.degrees)
            except Exception:
                refracted_altitude_deg = None
        else:
            altitude_deg, azimuth_deg =
self._geocentric_horizon_projection(
                t=t,
                ra_hours=ra_hours,
                dec_deg=dec_deg,
                latitude=latitude,
                longitude=longitude,
            )
            try:
                refracted_altitude_deg = float(
                    topos.refract(altitude_deg,
"standard", "standard").degrees
                )
            except Exception:
                refracted_altitude_deg = None

        lst_hours = (float(t.gast) + longitude / 15.0)
% 24.0

```

```

        hour_angle_hours = ((lst_hours - ra_hours +
12.0) % 24.0) - 12.0

        earth_center_pos =
self.earth.at(t).observe(body).apparent()
        ecl_lat, ecl_lon, _ =
earth_center_pos.frame_latlon(ecliptic_frame)
        ecliptic_latitude_deg = float(ecl_lat.degrees)
        ecliptic_longitude_deg = float(ecl_lon.degrees)
% 360.0

        if self.constellation_at is not None:
            try:
                abbr =
self.constellation_at(earth_center_pos)
                constellation =
self.constellation_names.get(abbr, abbr)
            except Exception:
                constellation = "-"
        else:
            constellation = "-"

        if name == "Matahari":
            elongation_deg = 0.0
            phase_angle_deg = 0.0
            illumination_pct = 100.0
            phase_cycle_deg = None
            phase_name = "-"
            waxing_waning = "-"
            bright_limb_pa = None
            radius_km = SUN_RADIUS_KM
        else:
            elongation_deg =
float(pos.separation_from(sun_pos).degrees)
            phase_angle_deg =
float(pos.phase_angle(self.sun).degrees)
            illumination_pct =
float(pos.fraction_illuminated(self.sun) * 100.0)
            phase_cycle_deg =
float(almanac.moon_phase(self.eph, t).degrees) % 360.0
            phase_name =
self._moon_phase_name(phase_cycle_deg)
            waxing_waning =
self._waxing_waning(phase_cycle_deg)
            bright_limb_pa =
self._bright_limb_position_angle(t)
            radius_km = MOON_RADIUS_KM

```

```

        distance_km =
float(earth_center_pos.distance().km)
        distance_au = distance_km / AU_KM
        observer_distance_km = float(pos.distance().km)

        angular_diameter_deg = math.degrees(
            2.0 * math.asin(min(1.0, radius_km /
max(observer_distance_km, radius_km + 1.0)))
        )
        angular_diameter_arcmin = angular_diameter_deg
* 60.0
        semidiameter_arcmin = angular_diameter_arcmin /
2.0
        horizontal_parallax_deg = math.degrees(
            math.asin(min(1.0,
EARTH_EQUATORIAL_RADIUS_KM / max(distance_km,
EARTH_EQUATORIAL_RADIUS_KM + 1.0)))
        )

        if name == "Matahari":
            magnitude =
self._sun_magnitude(distance_au)
        else:
            magnitude =
self._moon_magnitude(phase_angle_deg, distance_km)

        return BodyResult(
            name=name,
            altitude_deg=altitude_deg,

refracted_altitude_deg=refracted_altitude_deg,
            azimuth_deg=azimuth_deg,
            hour_angle_hours=hour_angle_hours,
            ra_hours=ra_hours,
            dec_deg=dec_deg,

ecliptic_longitude_deg=ecliptic_longitude_deg,

ecliptic_latitude_deg=ecliptic_latitude_deg,
            constellation=constellation,
            elongation_deg=elongation_deg,
            illumination_pct=illumination_pct,
            phase_angle_deg=phase_angle_deg,
            phase_cycle_deg=phase_cycle_deg,
            phase_name=phase_name,
            waxing_waning=waxing_waning,

bright_limb_position_angle_deg=bright_limb_pa,
            magnitude=magnitude,

```

```

        distance_km=distance_km,
        distance_au=distance_au,
        observer_distance_km=observer_distance_km,

angular_diameter_arcmin=angular_diameter_arcmin,
        semidiameter_arcmin=semidiameter_arcmin,

horizontal_parallax_deg=horizontal_parallax_deg,
    )

    def _bright_limb_position_angle(self, t) -> float |
None:
        try:
            e = self.earth.at(t)
            m = e.observe(self.moon).apparent()
            s = e.observe(self.sun).apparent()
            mra, mdec, _ = m.radec(epoch=t)
            sra, sdec, _ = s.radec(epoch=t)
            am = math.radians(float(mra.hours) * 15.0)
            dm = math.radians(float(mdec.degrees))
            a_s = math.radians(float(sra.hours) * 15.0)
            d_s = math.radians(float(sdec.degrees))
            da = a_s - am
            y = math.cos(d_s) * math.sin(da)
            x = (
                math.sin(d_s) * math.cos(dm)
                - math.cos(d_s) * math.sin(dm) *
math.cos(da)
            )
            return math.degrees(math.atan2(y, x)) %
360.0
        except Exception:
            return None

    @staticmethod
    def _moon_phase_name(phase_deg: float) -> str:
        p = phase_deg % 360.0
        if p < 22.5 or p >= 337.5:
            return "Bulan Baru (New Moon)"
        if p < 67.5:
            return "Sabit Muda (Waxing Crescent)"
        if p < 112.5:
            return "Perbani Awal (First Quarter)"
        if p < 157.5:
            return "Cembung Membesar (Waxing Gibbous)"
        if p < 202.5:
            return "Purnama (Full Moon)"
        if p < 247.5:
            return "Cembung Mengecil (Waning Gibbous)"

```

```

        if p < 292.5:
            return "Perbani Akhir (Last Quarter)"
        return "Sabit Tua (Waning Crescent)"

    @staticmethod
    def _waxing_waning(phase_deg: float) -> str:
        p = phase_deg % 360.0
        if abs(p) < 1e-9:
            return "Bulan Baru"
        if p < 180.0:
            return "Membesar / waxing"
        if abs(p - 180.0) < 1e-9:
            return "Purnama"
        return "Mengecil / waning"

    @staticmethod
    def _signed_angle_diff(a_deg: float, b_deg: float)
-> float:
        return ((a_deg - b_deg + 180.0) % 360.0) -
180.0

    @staticmethod
    def _geocentric_horizon_projection(t, ra_hours,
dec_deg, latitude, longitude):
        lat = math.radians(latitude)
        dec = math.radians(dec_deg)
        lst_hours = (float(t.gast) + longitude / 15.0)
% 24.0
        hour_angle = math.radians(((lst_hours -
ra_hours) * 15.0 + 180.0) % 360.0 - 180.0)

        sin_alt = (
            math.sin(lat) * math.sin(dec)
            + math.cos(lat) * math.cos(dec) *
math.cos(hour_angle)
        )
        sin_alt = max(-1.0, min(1.0, sin_alt))
        alt = math.asin(sin_alt)

        az = math.atan2(
            math.sin(hour_angle),
            math.cos(hour_angle) * math.sin(lat) -
math.tan(dec) * math.cos(lat),
        )
        az = (az + math.pi) % (2.0 * math.pi)

        return math.degrees(alt), math.degrees(az)

```

```

    @staticmethod
    def _sun_magnitude(distance_au: float) -> float:
        return -26.86 + 5.0 *
math.log10(max(distance_au, 1e-12))

    @staticmethod
    def _moon_magnitude(phase_angle_deg: float,
distance_km: float) -> float:
        a = abs(float(phase_angle_deg))
        phase_mag = -12.73 + 0.026 * a + 4.0e-9 * (a **
4)

        distance_correction = 5.0 *
math.log10(max(distance_km, 1.0) /
MEAN_MOON_DISTANCE_KM)
        return phase_mag + distance_correction

    def _prayer_schedule(self, *, local_dt, observer,
latitude, timezone_name):
        """Astronomical prayer-time schedule for the
selected local date.

        Conventions used in this application:
        - Shubuh: two parallel solar-center altitude
criteria, -20° and -18°.
        - Sunrise/Maghrib: Skyfield sunrise/sunset
(standard apparent horizon).
        - Dhuha: solar-center altitude +4.5° while
rising.
        - Dhuhur: local meridian transit of the Sun.
        - Ashar: standard shadow factor 1; target
altitude is derived from
            latitude and solar declination at transit.
        - Isya: solar-center altitude -18° while
setting.
        - Midnight: midpoint from Maghrib to next-day
Shubuh -18°.
        - 1/3 Malam: start of the last third of the
same night interval.

        These are astronomical outputs without
ihtiyat/additional minute offsets.
        """
        tz = ZoneInfo(timezone_name)
        local_date = local_dt.date()
        day_start = datetime.combine(local_date,
dt_time.min, tzinfo=tz)
        day_end = day_start + timedelta(days=1)

```

```

        t0 =
self.ts.from_datetime(day_start.astimezone(timezone.utc
))
        t1 =
self.ts.from_datetime(day_end.astimezone(timezone.utc))

        sunrise = self._first_real_event(
            *almanac.find_risings(observer, self.sun,
t0, t1), tz=tz
        )
        maghrib = self._first_real_event(
            *almanac.find_settings(observer, self.sun,
t0, t1), tz=tz
        )
        dhuhur = self._first_time(
            almanac.find_transits(observer, self.sun,
t0, t1), tz
        )

        shubuh_20 = self._solar_altitude_crossing(
            observer=observer, local_date=local_date,
tz=tz,
            altitude_deg=-20.0, direction="rising"
        )
        shubuh_18 = self._solar_altitude_crossing(
            observer=observer, local_date=local_date,
tz=tz,
            altitude_deg=-18.0, direction="rising"
        )
        dhuha = self._solar_altitude_crossing(
            observer=observer, local_date=local_date,
tz=tz,
            altitude_deg=4.5, direction="rising"
        )
        isya = self._solar_altitude_crossing(
            observer=observer, local_date=local_date,
tz=tz,
            altitude_deg=-18.0, direction="setting"
        )

        ashar_target = None
        ashar = None
        if dhuhur is not None:
            try:
                tt =
self.ts.from_datetime(dhuhur.astimezone(timezone.utc))
                _, dec, _ =
self.earth.at(tt).observe(self.sun).apparent().radec(ep
och=tt)

```

```

        dec_deg = float(dec.degrees)
        zenith_diff = abs(float(latitude) -
dec_deg)
        denom = 1.0 +
math.tan(math.radians(zenith_diff))
        ashar_target =
math.degrees(math.atan2(1.0, denom))
        ashar = self._solar_altitude_crossing(
            observer=observer,
local_date=local_date, tz=tz,
            altitude_deg=ashar_target,
direction="setting"
        )
    except Exception:
        ashar_target = None
        ashar = None

        next_date = local_date + timedelta(days=1)
        next_shubuh_18 = self._solar_altitude_crossing(
            observer=observer, local_date=next_date,
tz=tz,
            altitude_deg=-18.0, direction="rising"
        )
        midnight = None
        last_third = None
        if maghrib is not None and next_shubuh_18 is
not None and next_shubuh_18 > maghrib:
            night = next_shubuh_18 - maghrib
            midnight = maghrib + night / 2
            last_third = maghrib + night * (2.0 / 3.0)

        shubuh_pair = (
            f"{self._fmt_clock(shubuh_20, local_date)}
/ {self._fmt_clock(shubuh_18, local_date)}"
            if shubuh_20 is not None or shubuh_18 is
not None else "- / -"
        )
        ashar_basis = (
            f"Faktor bayangan 1; tinggi Matahari
{ashar_target:.3f}°"
            if ashar_target is not None else "Faktor
bayangan 1"
        )

        return [
            ScheduleItem(
                "Shubuh (-20/-18)", shubuh_pair,
                "Tinggi pusat Matahari -20° / -18° di
ufuk timur",

```

```

        "Dua nilai ditampilkan berdampingan; -
18° adalah kriteria Subuh Muhammadiyah saat ini."
    ),
    ScheduleItem(
        "Sunrise", self._fmt_clock(sunrise,
local_date),
        "Terbit Matahari menurut rutin rise
Skyfield",
        "Waktu lokal; memperhitungkan model
horizon standar pada rutin rise/set."
    ),
    ScheduleItem(
        "Dhuha", self._fmt_clock(dhuha,
local_date),
        "Tinggi pusat Matahari +4,5° saat
naik",
        "Sesuai parameter +4,5° yang dipilih
untuk aplikasi ini."
    ),
    ScheduleItem(
        "Dhuhur", self._fmt_clock(dhuhur,
local_date),
        "Transit / kulminasi atas Matahari pada
meridian lokal",
        "Waktu zawal astronomis; tanpa tambahan
ihtiyat menit."
    ),
    ScheduleItem(
        "Ashar", self._fmt_clock(ashar,
local_date),
        ashar_basis,
        "Model bayangan standar: panjang
bayangan tambahan = 1 × tinggi benda."
    ),
    ScheduleItem(
        "Maghrib", self._fmt_clock(maghrib,
local_date),
        "Terbenam Matahari menurut rutin set
Skyfield",
        "Waktu lokal; sama dengan dasar Maghrib
Lokal pada aplikasi."
    ),
    ScheduleItem(
        "Isya", self._fmt_clock(isya,
local_date),
        "Tinggi pusat Matahari -18° di ufuk
barat",
        "Kriteria astronomis yang digunakan
aplikasi untuk awal Isya."

```

```

        ),
        ScheduleItem(
            "Midnight", self._fmt_clock(midnight,
local_date),
            "Titik tengah Maghrib → Shubuh -18°
hari berikutnya",
            "Midnight syar'i/astronomis pada
aplikasi, bukan selalu pukul 00:00."
        ),
        ScheduleItem(
            "1/3 Malam",
self._fmt_clock(last_third, local_date),
            "Awal sepertiga malam terakhir: 2/3
interval Maghrib → Shubuh -18°",
            "Ditampilkan sebagai awal 1/3 malam
terakhir."
        ),
    ]

    def _solar_altitude_crossing(self, *, observer,
local_date, tz, altitude_deg, direction):
        """Return one local datetime when the Sun
crosses a requested altitude."""
        start_local = datetime.combine(local_date,
dt_time.min, tzinfo=tz)
        end_local = start_local + timedelta(days=1)
        t0 =
self.ts.from_datetime(start_local.astimezone(timezone.u
tc))
        t1 =
self.ts.from_datetime(end_local.astimezone(timezone.utc
))

        threshold = float(altitude_deg)

        def above(t):
            alt, _, _ =
observer.at(t).observe(self.sun).apparent().altaz()
            return np.asarray(alt.degrees >= threshold,
dtype=np.int8)

        # A 5-minute coarse step is safely below the
duration of normal solar
        # altitude crossings; find_discrete refines the
transition time.
        above.step_days = 5.0 / 1440.0
        try:
            prev =
bool(int(np.asarray(above(t0)).reshape(-1)[0]))

```

```

        times, states = almanac.find_discrete(t0,
t1, above)
        for ti, state in zip(times, states):
            cur = bool(int(state))
            match = (
                direction == "rising" and (not
prev) and cur
            ) or (
                direction == "setting" and prev and
(not cur)
            )
            if match:
                return
ti.utc_datetime().replace(tzinfo=timezone.utc).astimezo
ne(tz)
            prev = cur
        except Exception:
            return None
        return None

    @staticmethod
    def _fmt_clock(dt, base_date):
        if dt is None:
            return "-"
        day_delta = (dt.date() - base_date).days
        suffix = ""
        if day_delta > 0:
            suffix = f" ({day_delta} hari)"
        elif day_delta < 0:
            suffix = f" ({day_delta} hari)"
        return dt.strftime("%H:%M:%S %Z") + suffix

    def _events(self, local_dt, observer, topos,
longitude, timezone_name):
        tz = ZoneInfo(timezone_name)
        local_date = local_dt.date()
        day_start_local = datetime.combine(local_date,
dt_time.min, tzinfo=tz)
        day_end_local = day_start_local +
timedelta(days=1)

        t0 =
self.ts.from_datetime(day_start_local.astimezone(timezo
ne.utc))
        t1 =
self.ts.from_datetime(day_end_local.astimezone(timezone
.utc))

        sun_rise_dt = self._first_real_event(

```

```

        *almanac.find_risings(observer, self.sun,
t0, t1), tz=tz
    )
    sun_set_dt = self._first_real_event(
        *almanac.find_settings(observer, self.sun,
t0, t1), tz=tz
    )
    moon_rise_dt = self._first_real_event(
        *almanac.find_risings(observer, self.moon,
t0, t1), tz=tz
    )
    moon_set_dt = self._first_real_event(
        *almanac.find_settings(observer, self.moon,
t0, t1), tz=tz
    )

    sun_transit_dt =
self._first_time(almanac.find_transits(observer,
self.sun, t0, t1), tz)
    moon_transit_dt =
self._first_time(almanac.find_transits(observer,
self.moon, t0, t1), tz)

    twilight = self._twilight_events(t0, t1, topos,
tz)
    t_current =
self.ts.from_datetime(local_dt.astimezone(timezone.utc)
)
    twilight_state = self._twilight_state(topos,
t_current)

    nearest_new, previous_new =
self._new_moon_times(local_dt.astimezone(timezone.utc))
    conj_geo_text =
self._fmt_dt(nearest_new.astimezone(tz)) if nearest_new
else "-"

    conj_topo_dt = None
    conj_topo_min = None
    if nearest_new is not None:
        try:
            conj_topo_dt, conj_topo_min =
self._topocentric_conjunction(observer, nearest_new)
        except Exception:
            conj_topo_dt, conj_topo_min = None,
None

    moon_age_days = 0.0
    if previous_new is not None:

```

```

        moon_age_days = (
            local_dt.astimezone(timezone.utc) -
previous_new
        ).total_seconds() / 86400.0

        lag = self._lag_minutes(observer, sun_set_dt,
tz)

        day_length_hours = None
        if sun_rise_dt is not None and sun_set_dt is
not None:
            seconds = (sun_set_dt -
sun_rise_dt).total_seconds()
            if seconds < 0:
                seconds += 86400.0
            if 0.0 <= seconds <= 86400.0:
                day_length_hours = seconds / 3600.0

        equation_of_time =
self._equation_of_time_minutes(sun_transit_dt,
longitude)
        lst_hours = (float(t_current.gast) + longitude
/ 15.0) % 24.0

        return EventResult(
            sun_rise=self._fmt_dt(sun_rise_dt),

sun_rise_azimuth_deg=self._event_azimuth(observer,
self.sun, sun_rise_dt),
            sun_set=self._fmt_dt(sun_set_dt),

sun_set_azimuth_deg=self._event_azimuth(observer,
self.sun, sun_set_dt),
            moon_rise=self._fmt_dt(moon_rise_dt),

moon_rise_azimuth_deg=self._event_azimuth(observer,
self.moon, moon_rise_dt),
            moon_set=self._fmt_dt(moon_set_dt),

moon_set_azimuth_deg=self._event_azimuth(observer,
self.moon, moon_set_dt),
            sun_transit=self._fmt_dt(sun_transit_dt),

sun_transit_altitude_deg=self._event_altitude(observer,
self.sun, sun_transit_dt),
            moon_transit=self._fmt_dt(moon_transit_dt),

moon_transit_altitude_deg=self._event_altitude(observer
, self.moon, moon_transit_dt),

```

```

civil_dawn=self._fmt_dt(twilight.get("civil_dawn")),

nautical_dawn=self._fmt_dt(twilight.get("nautical_dawn"
)),

astronomical_dawn=self._fmt_dt(twilight.get("astronomic
al_dawn")),

civil_dusk=self._fmt_dt(twilight.get("civil_dusk")),

nautical_dusk=self._fmt_dt(twilight.get("nautical_dusk"
)),

astronomical_dusk=self._fmt_dt(twilight.get("astronomic
al_dusk")),
        day_length_hours=day_length_hours,
        equation_of_time_minutes=equation_of_time,
        local_sidereal_time_hours=lst_hours,
        twilight_state=twilight_state,
        conjunction_geocentric=conj_geo_text,

conjunction_topocentric=self._fmt_dt(conj_topo_dt.astim
ezone(tz)) if conj_topo_dt else "-",

conjunction_topocentric_min_elong_deg=conj_topo_min,
        moon_age_days=moon_age_days,
        lag_time_minutes=lag,
    )

    def _twilight_events(self, t0, t1, topos, tz):
        f = almanac.dark_twilight_day(self.eph, topos)
        try:
            prev = int(np.asarray(f(t0)).reshape(-
1)[0])
        except Exception:
            prev = 0
        times, states = almanac.find_discrete(t0, t1,
f)

        out = {
            "astronomical_dawn": None,
            "nautical_dawn": None,
            "civil_dawn": None,
            "civil_dusk": None,
            "nautical_dusk": None,
            "astronomical_dusk": None,
        }
        for ti, state in zip(times, states):
            cur = int(state)

```

```

        dt =
ti.utc_datetime().replace(tzinfo=timezone.utc).astimezo
ne(tz)
        transition = (prev, cur)
        if transition == (0, 1):
            out["astronomical_dawn"] = dt
        elif transition == (1, 2):
            out["nautical_dawn"] = dt
        elif transition == (2, 3):
            out["civil_dawn"] = dt
        elif transition == (3, 2):
            out["civil_dusk"] = dt
        elif transition == (2, 1):
            out["nautical_dusk"] = dt
        elif transition == (1, 0):
            out["astronomical_dusk"] = dt
        prev = cur
    return out

def _twilight_state(self, topos, t) -> str:
    names = {
        0: "Malam / gelap astronomis",
        1: "Senja/fajar astronomis",
        2: "Senja/fajar nautika",
        3: "Senja/fajar sipil",
        4: "Siang / Matahari di atas ufuk",
    }
    try:
        code =
int(np.asarray(almanac.dark_twilight_day(self.eph,
topos)(t)).reshape(-1)[0])
        return names.get(code, "Tidak diketahui")
    except Exception:
        return "Tidak diketahui"

    def _equation_of_time_minutes(self,
sun_transit_local, longitude):
        if sun_transit_local is None:
            return None
        utc =
sun_transit_local.astimezone(timezone.utc)
        utc_minutes = utc.hour * 60.0 + utc.minute +
utc.second / 60.0 + utc.microsecond / 60_000_000.0
        local_mean_solar_minutes = (utc_minutes +
longitude * 4.0) % 1440.0
        eot = 720.0 - local_mean_solar_minutes
        while eot > 720.0:
            eot -= 1440.0
        while eot <= -720.0:

```

```

        eot += 1440.0
    return eot

    def _event_azimuth(self, observer, body, dt):
        if dt is None:
            return None
        try:
            t =
self.ts.from_datetime(dt.astimezone(timezone.utc))
            _, az, _ =
observer.at(t).observe(body).apparent().altaz()
            return float(az.degrees)
        except Exception:
            return None

    def _event_altitude(self, observer, body, dt):
        if dt is None:
            return None
        try:
            t =
self.ts.from_datetime(dt.astimezone(timezone.utc))
            alt, _, _ =
observer.at(t).observe(body).apparent().altaz()
            return float(alt.degrees)
        except Exception:
            return None

    def _new_moon_times(self, current_utc: datetime):
        t0 = self.ts.from_datetime(current_utc -
timedelta(days=40))
        t1 = self.ts.from_datetime(current_utc +
timedelta(days=40))
        times, phases = almanac.find_discrete(t0, t1,
almanac.moon_phases(self.eph))

        new_moons: list[datetime] = []
        for ti, phase in zip(times, phases):
            if int(phase) == 0:

new_moons.append(ti.utc_datetime().replace(tzinfo=timez
one.utc))

        if not new_moons:
            return None, None

        nearest = min(new_moons, key=lambda dt: abs((dt
- current_utc).total_seconds()))
        previous_candidates = [dt for dt in new_moons
if dt <= current_utc]

```

```

        previous = max(previous_candidates) if
previous_candidates else None
        return nearest, previous

    def _topocentric_conjunction(self, observer,
geo_conjunction_utc: datetime):
        coarse = [
            geo_conjunction_utc + timedelta(minutes=m)
            for m in range(-720, 721, 5)
        ]
        tc = self.ts.from_datetimes(coarse)
        obs = observer.at(tc)
        moon_pos = obs.observe(self.moon).apparent()
        sun_pos = obs.observe(self.sun).apparent()
        sep =
np.asarray(moon_pos.separation_from(sun_pos).degrees)
        idx = int(np.nanargmin(sep))
        coarse_best = coarse[idx]

        fine = [
            coarse_best + timedelta(seconds=s)
            for s in range(-300, 301, 5)
        ]
        tf = self.ts.from_datetimes(fine)
        obs2 = observer.at(tf)
        moon2 = obs2.observe(self.moon).apparent()
        sun2 = obs2.observe(self.sun).apparent()
        sep2 =
np.asarray(moon2.separation_from(sun2).degrees)
        idx2 = int(np.nanargmin(sep2))

        return fine[idx2], float(sep2[idx2])

    def _lag_minutes(self, observer, sunset_local, tz):
        if sunset_local is None:
            return None

        start_local = sunset_local -
timedelta(hours=12)
        end_local = sunset_local + timedelta(hours=12)
        t0 =
self.ts.from_datetime(start_local.astimezone(timezone.u
tc))
        t1 =
self.ts.from_datetime(end_local.astimezone(timezone.utc
))
        times, flags = almanac.find_settings(observer,
self.moon, t0, t1)

```

```

        moonsets = []
        for ti, ok in zip(times, flags):
            if bool(ok):

moonsets.append(ti.utctime().replace(tzinfo=timezo
ne.utc).astimezone(tz))

        if not moonsets:
            return None

        nearest = min(moonsets, key=lambda dt: abs((dt
- sunset_local).total_seconds()))
        return (nearest - sunset_local).total_seconds()
/ 60.0

def _global_hijri_scan(
    self,
    *,
    date_text: str,
    mode: str,
    reference_timezone_name: str,
    min_moon_altitude_deg: float = 5.0,
    min_elongation_deg: float = 8.0,
) -> GlobalHijriResult:
    """Scan geosentris KHGT: PKG 1 / PKG 2 dengan
audit H0-H2.

        H0 memakai tanggal UTC ijtimak, sehingga
tanggal global tidak bergeser
        ketika lokasi/zona waktu atau mode hasil lokal
diganti. Pada setiap hari
        D, sunset lokal bertanggal D dipisahkan menurut
batas D+1 00:00 UTC.
        PKG 2 perlu saksi daratan utama Amerika dan
ijtimak sebelum fajar
        Gisborne bertanggal D+1. Jika gagal, hari
berikutnya diuji kembali.
        """
        selected_date = datetime.strptime(date_text,
"%Y-%m-%d").date()
        reference_tz =
ZoneInfo(reference_timezone_name)
        global_mode = "Geosentris"
        month_start = datetime(selected_date.year,
selected_date.month, 1,
                                tzinfo=timezone.utc)
        if selected_date.month == 12:
            next_month = datetime(selected_date.year +
1, 1, 1, tzinfo=timezone.utc)

```

```

        else:
            next_month = datetime(selected_date.year,
selected_date.month + 1, 1,
                                tzinfo=timezone.utc)
            phase_times, phase_codes =
almanac.find_discrete(
                self.ts.from_datetime(month_start),
self.ts.from_datetime(next_month),
                almanac.moon_phases(self.eph),
            )
            conjunctions =
[ti.utc_datetime().astimezone(timezone.utc)
                for ti, code in
zip(phase_times, phase_codes) if int(code) == 0]
            conjunction_utc = conjunctions[0] if
conjunctions else None
            if conjunction_utc is None:
                return GlobalHijriResult(

selected_date=selected_date.isoformat(),
mode=global_mode,
                qualifies=False, matched_count=0,
total_count=len(GLOBAL_HIJRI_SCAN_CITIES),
                hijri_start_gregorian_date=None,
first_match=None, cities=[],
                window_eligible=False, h_day=None,
conjunction_local_datetime=None,

reference_timezone_name=reference_timezone_name,
                decision_reason="Konjungsi tidak
ditemukan pada bulan Gregorian UTC pilihan.",
            )

            h0_date = conjunction_utc.date()
            gisborne = GLOBAL_HIJRI_SCAN_CITIES[0]
            _, gisborne_lat, gisborne_lon, gisborne_elev,
gisborne_zone = gisborne
            gisborne_tz = ZoneInfo(gisborne_zone)
            gisborne_observer = self.earth + wgs84.latlon(
                gisborne_lat, gisborne_lon,
elevation_m=gisborne_elev)
            dawns = {}
            for h_day in range(3):
                candidate_start = h0_date +
timedelta(days=h_day + 1)
                dawns[h_day] =
self._solar_altitude_crossing(
                    observer=gisborne_observer,
local_date=candidate_start, tz=gisborne_tz,

```

```

altitude_deg=KHGT_GISBORNE_DAWN_ALTITUDE_DEG,
direction="rising",
)

def scan_site(site, h_day):
    name, lat, lon, elevation_m, tz_name = site
    scan_date = h0_date + timedelta(days=h_day)
    midnight = datetime.combine(scan_date +
timedelta(days=1), dt_time.min,

tzinfo=timezone.utc)
    dawn = dawns[h_day]
    dawn_utc = dawn.astimezone(timezone.utc) if
dawn is not None else None
    america_mainland =
(tz_name.startswith('America/')
                                and name not in
KHGT_AMERICAN_ISLAND_SITES)
    topos = wgs84.latlon(lat, lon,
elevation_m=elevation_m)
    observer = self.earth + topos
    try:
        sunset = self._local_sunset_datetime(
            date_text=scan_date.isoformat(),
timezone_name=tz_name, observer=observer)
    except ValueError as error:
        # Tidak menutupi error
kernel/ephemeris; hanya sunset yang tidak ada.
        if not str(error).startswith('Waktu
Maghrib/sunset tidak ditemukan'):
            raise
        sunset = None
    if sunset is None:
        sun_result = moon_result = None
        numeric_qualifies = False
    else:
        t =
self.ts.from_datetime(sunset.astimezone(timezone.utc))
        sun_result = self._body_result(
            "Matahari", self.sun, t, observer,
topos, lat, lon, global_mode)
        moon_result = self._body_result(
            "Bulan", self.moon, t, observer,
topos, lat, lon, global_mode)
        numeric_qualifies =
(moon_result.altitude_deg >= min_moon_altitude_deg
                                and
moon_result.elongation_deg >= min_elongation_deg)

```

```

        sunset_utc =
sunset.astimezone(timezone.utc) if sunset is not None
else None
        status, accepted = khgt_pkg_status(
            conjunction_utc=conjunction_utc,
sunset_utc=sunset_utc,
            utc_midnight=midnight,
gisborne_dawn_utc=dawn_utc,
            numeric_qualifies=numeric_qualifies,
america_mainland=america_mainland,
        )
        return GlobalScanCityResult(
            name=name, timezone_name=tz_name,
sunset_local=sunset,

moon_altitude_deg=moon_result.altitude_deg if
moon_result else None,

sun_altitude_deg=sun_result.altitude_deg if sun_result
else None,

elongation_deg=moon_result.elongation_deg if
moon_result else None,

illumination_pct=moon_result.illumination_pct if
moon_result else None,
            qualifies=numeric_qualifies,
h_day=h_day, scan_date=scan_date.isoformat(),
            sunset_utc=sunset_utc,
pkg_status=status, pkg_accepted=accepted,
            america_mainland=america_mainland,
        )

        # Tabel utama tetap 100 kota per hari (300
baris audit H0-H2).
        rows = [scan_site(site, h) for h in range(3)]
for site in GLOBAL_HIJRI_SCAN_CITIES]
        validation_rows = []
        decisions = []
        first_match = None
        decision_pkg = None
        selected_h = None
        for h_day in range(3):
            primary = [row for row in rows if row.h_day
== h_day]
            witnesses = primary[:]
            if not any(row.pkg_status == 'PKG 1' for
row in primary):

```

```

        # PKG 1 harus diperiksa sebelum PKG 2;
daftar lama terlalu padat
        # Alaska dan dapat melewati daratan
Amerika Selatan.
        extra = [scan_site(site, h_day) for
site in KHGT_AMERICA_VALIDATION_SITES]
        validation_rows.extend(extra)
        witnesses.extend(extra)
        pkg1 = [row for row in witnesses if
row.pkg_status == 'PKG 1']
        pkg2 = [row for row in witnesses if
row.pkg_status == 'PKG 2']
        postponed = [row for row in witnesses
if
row.pkg_status.startswith('PKG 2 DITUNDA')]
        dawn = dawns[h_day]
        decision = {
            'h_day': h_day, 'scan_date': (h0_date +
timedelta(days=h_day)).isoformat(),
            'utc_midnight':
datetime.combine(h0_date + timedelta(days=h_day + 1),
dt_time.min, tzinfo=timezone.utc),
            'gisborne_dawn_local_datetime': dawn,
            'gisborne_dawn_utc_datetime':
dawn.astimezone(timezone.utc) if dawn else None,
            'conjunction_before_dawn':
conjunction_utc < dawn if dawn else None,
            'pkg1_count': len(pkg1), 'pkg2_count':
len(pkg2),
            'pkg2_postponed_count': len(postponed),
            'primary_accepted_count':
sum(row.pkg_accepted for row in primary),
            'validation_count': len(witnesses) -
len(primary),
        }
        decisions.append(decision)
        accepted = pkg1 if pkg1 else pkg2
        if accepted:
            selected_h = h_day
            decision_pkg = 'PKG 1' if pkg1 else
'PKG 2'
            first_match = min(accepted, key=lambda
row: row.sunset_utc)
            break

        selected_decision = decisions[-1]
        qualifies = first_match is not None

```

```

        hijri_start = ((h0_date +
timedelta(days=selected_h + 1)).isoformat()
                        if selected_h is not None else
None)
        if decision_pkg == 'PKG 1':
            reason = 'Kriteria geosentris 5° + 8°
terpenuhi di daratan sebelum batas 00:00 UTC.'
        elif decision_pkg == 'PKG 2':
            reason = ('Kriteria geosentris 5° + 8°
terpenuhi di daratan utama Amerika '
                    'setelah batas 00:00 UTC dan
ijtimak sebelum fajar Gisborne.')
        else:
            reason = 'Belum ditemukan saksi PKG 1/PKG 2
yang sah pada sampel H0-H2.'
            if any(item['pkg2_postponed_count'] for item in
decisions[:-1]):
                reason += ' PKG 2 pada hari sebelumnya
ditunda karena ijtimak tidak sebelum fajar Gisborne.'

        return GlobalHijriResult(
            selected_date=selected_date.isoformat(),
mode=global_mode, qualifies=qualifies,

matched_count=selected_decision['primary_accepted_count
'],
            total_count=len(GLOBAL_HIJRI_SCAN_CITIES),
hijri_start_gregorian_date=hijri_start,
            first_match=first_match, cities=rows,
window_eligible=True, h_day=selected_h,

conjunction_local_datetime=conjunction_utc.astimezone(r
eference_tz),

reference_timezone_name=reference_timezone_name,
decision_pkg=decision_pkg,
            decision_reason=reason,
utc_reference_date=h0_date.isoformat(),

utc_midnight=selected_decision['utc_midnight'],

gisborne_dawn_local_datetime=selected_decision['gisborn
e_dawn_local_datetime'],

gisborne_dawn_utc_datetime=selected_decision['gisborne_
dawn_utc_datetime'],

gisborne_dawn_altitude_deg=KHGT_GISBORNE_DAWN_ALTITUDE_
DEG,

```

```

conjunction_before_gisborne_dawn=selected_decision['con
junction_before_dawn'],
        validation_sites=validation_rows,
daily_decisions=decisions,
    )

```

```

def _monthly_hilal_candidates(
    self,
    *,
    selected_local_dt: datetime,
    observer,
    latitude: float,
    longitude: float,
    timezone_name: str,
    selected_time_mode: str = "Jam Manual",
    selected_display_mode: str = "Toposentris",
    selected_sun: BodyResult | None = None,
    selected_moon: BodyResult | None = None,
    min_moon_altitude_deg: float = 5.0,
    min_elongation_deg: float = 8.0,
) -> list[PhenomenonItem]:
    """

```

Scan the selected calendar month at each local sunset (Maghrib).

The hilal-day labels are anchored to the local calendar date of the geocentric New Moon (conjunction):

```

    H0 = Maghrib on the local calendar date of
conjunction
    H1 = Maghrib one calendar day after H0
    H2 = Maghrib two calendar days after H0,
and so on.

```

```

Candidate dates satisfy BOTH numerical filters:
    Moon geometric altitude in the selected
mode >= 5 degrees
    Moon-Sun elongation in the selected mode
>= 8 degrees

```

To make the threshold development easy to audit, the output includes

a continuous progression from H0 through the first qualifying date.

Later qualifying dates in the selected month are also shown. Each

dated row intentionally contains ONLY the four requested parameters:

- Moon/crescent altitude
- Elongation
- Sun altitude
- Moon illumination

These thresholds are screening filters, not a formal crescent-visibility criterion.

```

"""
    tz = ZoneInfo(timezone_name)
    year = selected_local_dt.year
    month = selected_local_dt.month
    days_in_month = calendar.monthrange(year,
month)[1]
    month_start_local = datetime(year, month, 1, 0,
0, 0, tzinfo=tz)
    if month == 12:
        next_month_local = datetime(year + 1, 1, 1,
0, 0, 0, tzinfo=tz)
    else:
        next_month_local = datetime(year, month +
1, 1, 0, 0, 0, tzinfo=tz)

    # Extend backward enough to retain H0
progression when a qualifying
    # date falls in the first days of the selected
month.
    scan_start_local = month_start_local -
timedelta(days=10)

    items: list[PhenomenonItem] = []
    criteria = (
        f"Filter {year:04d}-{month:02d}: tinggi
hilal ≥ {min_moon_altitude_deg:.1f}° "
        f"dan elongasi ≥ {min_elongation_deg:.1f}°
pada Maghrib lokal. "
        f"Semua nilai Kandidat Hilal menggunakan
kerangka {selected_display_mode} pada waktu Maghrib
lokal yang sama dari Skyfield."
    )

    try:
        t0 =
self.ts.from_datetime(scan_start_local.astimezone(timez
one.utc))
        t1 =
self.ts.from_datetime(next_month_local.astimezone(timez
one.utc))

```

```

        sunset_times, sunset_flags =
almanac.find_settings(observer, self.sun, t0, t1)

        local_sunsets: list[datetime] = []
        for ti, ok in zip(sunset_times,
sunset_flags):
            if not bool(ok):
                continue
            dt_utc =
ti.utc_datetime().replace(tzinfo=timezone.utc)
            dt_local = dt_utc.astimezone(tz)
            if scan_start_local.date() <=
dt_local.date() < next_month_local.date():
                local_sunsets.append(dt_local)

        if not local_sunsets:
            items.append(PhenomenonItem(
                "Kandidat Hilal Bulanan",
                f"Ringkasan {year:04d}-
{month:02d}",
                "0 tanggal",
                criteria + " Tidak ditemukan sunset
nyata pada bulan/lokasi ini.",
            ))
            return items

        utc_sunsets = [dt.astimezone(timezone.utc)
for dt in local_sunsets]
        tv = self.ts.from_datetimes(utc_sunsets)

        # The Maghrib instant itself is always a
local/topocentric sunset
        # event. Keep a dedicated topocentric Sun
altitude solely for
        # auditing that event against the
Skyfield/USNO sunset convention.
        topo_obs = observer.at(tv)
        topo_sun_pos =
topo_obs.observe(self.sun).apparent()
        topo_sun_alt = np.atleast_1d(
np.asarray(topo_sun_pos.altaz()[0].degrees,
dtype=float)
        )

        if selected_display_mode == "Toposentris":
            sun_pos = topo_sun_pos
            moon_pos =
topo_obs.observe(self.moon).apparent()

```

```

        sun_alt = topo_sun_alt.copy()
        moon_alt = np.atleast_1d(

np.asarray(moon_pos.altaz()[0].degrees, dtype=float)
        )
        elong = np.atleast_1d(

np.asarray(moon_pos.separation_from(sun_pos).degrees,
dtype=float)
        )
        illum = np.atleast_1d(

np.asarray(moon_pos.fraction_illuminated(self.sun) *
100.0, dtype=float)
        )
        elif selected_display_mode == "Geosentris":
            # Use the exact same scalar computation
path as the main
            # Geosentris cards. A month contains
at most 31 sunsets, so
            # this small loop prioritizes numerical
identity and robustness
            # over vectorization.
            sun_alt_values = []
            moon_alt_values = []
            elong_values = []
            illum_values = []
            for i in range(len(local_sunsets)):
                ti = tv[i]
                earth_center = self.earth.at(ti)
                sun_geo =
earth_center.observe(self.sun).apparent()
                moon_geo =
earth_center.observe(self.moon).apparent()

                sun_ra, sun_dec, _ =
sun_geo.radec(epoch=ti)
                moon_ra, moon_dec, _ =
moon_geo.radec(epoch=ti)
                sun_h, _ =
self._geocentric_horizon_projection(
                    t=ti,
                    ra_hours=float(sun_ra.hours),
                    dec_deg=float(sun_dec.degrees),
                    latitude=latitude,
                    longitude=longitude,
                )
                moon_h, _ =
self._geocentric_horizon_projection(

```

```

        t=ti,
        ra_hours=float(moon_ra.hours),

dec_deg=float(moon_dec.degrees),
        latitude=latitude,
        longitude=longitude,
    )
    sun_alt_values.append(sun_h)
    moon_alt_values.append(moon_h)

elong_values.append(float(moon_geo.separation_from(sun_
geo).degrees))

illum_values.append(float(moon_geo.fraction_illuminated
(self.sun) * 100.0))

        sun_alt = np.asarray(sun_alt_values,
dtype=float)
        moon_alt = np.asarray(moon_alt_values,
dtype=float)
        elong = np.asarray(elong_values,
dtype=float)
        illum = np.asarray(illum_values,
dtype=float)
    else:
        raise ValueError(
            f"Mode Kandidat Hilal tidak
dikenal: {selected_display_mode}"
        )

        # Every real sunset returned by
find_settings() should place the
        # topocentric geometric center of the Sun
near -0.8333 degree.
        # This audit concerns the EVENT TIME, not
the user's selected
        # display frame. In Geosentris mode the
displayed Sun altitude is
        # therefore allowed to differ slightly from
-0.8333 degree.
        bad_sunset_alt = [
            (local_sunsets[i],
float(topo_sun_alt[i]))
            for i in range(len(local_sunsets))
            if abs(float(topo_sun_alt[i]) - (-
0.8333)) > 0.03
        ]
        if bad_sunset_alt:
            dt_bad, alt_bad = bad_sunset_alt[0]

```

```

        raise RuntimeError(
            "Audit Maghrib bulanan gagal:
tinggi Matahari topocentris pada "
            f"{dt_bad.strftime('%Y-%m-%d
%H:%M:%S %Z')}" = {alt_bad:.6f}°, "
            "bukan sekitar -0.8333°."
        )

        # If the main calculation itself is at
        Maghrib Lokal, the selected
        # date must use the exact same coordinate
        frame and authoritative
        # values as the cards above, whether
        Toposentris or Geosentris.
        selected_reference_date = None
        if (
            selected_time_mode == "Maghrib Lokal"
            and selected_sun is not None
            and selected_moon is not None
        ):
            selected_reference_date =
selected_local_dt.date()
            ref_index = next(
                (i for i, dt_local in
enumerate(local_sunsets)
                 if dt_local.date() ==
selected_reference_date),
                None,
            )
            if ref_index is not None:
                diffs = {
                    "tinggi Matahari":
abs(float(sun_alt[ref_index]) -
selected_sun.altitude_deg),
                    "tinggi Bulan":
abs(float(moon_alt[ref_index]) -
selected_moon.altitude_deg),
                    "elongasi":
abs(float(elong[ref_index]) -
selected_moon.elongation_deg),
                    "iluminasi":
abs(float(illum[ref_index]) -
selected_moon.illumination_pct),
                }
                bad = {k: v for k, v in
diffs.items() if v > 0.002}
                if bad:
                    detail = ", ".join(f"{k}
Δ={v:.6f}" for k, v in bad.items())

```

```

        raise RuntimeError(
            "Pemeriksaan konsistensi
Kandidat Hilal gagal "

f"({selected_display_mode}): " + detail
        )

        # One source of truth for the
displayed selected date.
        sun_alt[ref_index] =
selected_sun.altitude_deg
        moon_alt[ref_index] =
selected_moon.altitude_deg
        elong[ref_index] =
selected_moon.elongation_deg
        illum[ref_index] =
selected_moon.illumination_pct

        # Keep one sunset index per local calendar
date.
        by_date: dict = {}
        for i, dt_local in
enumerate(local_sunsets):
            by_date.setdefault(dt_local.date(), i)

        selected_indices = [
            i for i, dt_local in
enumerate(local_sunsets)
            if dt_local.year == year and
dt_local.month == month
        ]

        match_indices = [
            i for i in selected_indices
            if float(moon_alt[i]) >=
min_moon_altitude_deg
            and float(elong[i]) >=
min_elongation_deg
        ]
        matches = len(match_indices)

        # Find New Moon events around the month. H0
is defined by the LOCAL
        # CALENDAR DATE of the geocentric
conjunction, regardless of whether
        # the exact conjunction instant occurs
before or after that day's sunset.
        phase_t0 = self.ts.from_datetime(

```

```

        (scan_start_local -
timedelta(days=35)).astimezone(timezone.utc)
    )
    phase_t1 = self.ts.from_datetime(
        (next_month_local +
timedelta(days=3)).astimezone(timezone.utc)
    )
    phase_times, phase_codes =
almanac.find_discrete(
        phase_t0, phase_t1,
almanac.moon_phases(self.eph)
    )
    conjunctions_local: list[datetime] = []
    for ti, code in zip(phase_times,
phase_codes):
        if int(code) != 0:
            continue
        conjunctions_local.append(

ti.utc_datetime().replace(tzinfo=timezone.utc).astimezo
ne(tz)

        )

    def conjunction_for_date(local_date):
        eligible = [dt for dt in
conjunctions_local if dt.date() <= local_date]
        return max(eligible, key=lambda dt:
dt.date()) if eligible else None

    def h_label_for_date(local_date):
        conj = conjunction_for_date(local_date)
        if conj is None:
            return None, None
        h = (local_date - conj.date()).days
        return f"H{h}", conj

    progression_indices: list[int] = []
    anchor_conjunction = None
    if match_indices:
        first_match_date =
local_sunsets[match_indices[0]].date()
        anchor_conjunction =
conjunction_for_date(first_match_date)
        if anchor_conjunction is not None:
            d = anchor_conjunction.date()
            while d <= first_match_date:
                idx = by_date.get(d)
                if idx is not None:

```

```

progression_indices.append(idx)
    d += timedelta(days=1)

    if anchor_conjunction is not None:
        h0_note = (
            f" H0 = Maghrib tanggal konjungsi
geosentris "
            f"{anchor_conjunction.strftime('%Y-
%m-%d')} waktu lokal; "
            "H1 satu hari sesudahnya, lalu H2,
H3, dan seterusnya."
        )
    else:
        h0_note = (
            " H0 didefinisikan sebagai Maghrib
pada tanggal lokal konjungsi "
            "geosentris; konjungsi acuan tidak
tersedia pada rentang pencarian."
        )

    progression_note = (
        f" Urutan H0 sampai kandidat pertama
menampilkan {len(progression_indices)} hari."
        if progression_indices else ""
    )
    items.append(PhenomenonItem(
        "Kandidat Hilal Bulanan",
        f"Ringkasan {year:04d}-{month:02d}",
        f"{matches} dari {days_in_month}
tanggal memenuhi filter",
        criteria + h0_note + progression_note
        + " Filter ini bukan keputusan
visibilitas hilal formal.",
    ))

    def make_row(i: int) -> PhenomenonItem:
        dt_local = local_sunsets[i]
        label, _conj =
h_label_for_date(dt_local.date())
        name = dt_local.strftime("%Y-%m-%d")
        if label:
            name += f" ({label})"
        value = (
            f"Tinggi hilal
{float(moon_alt[i]):.3f}° | "
            f"Elongasi {float(elong[i]):.3f}°"
        )
        desc = (

```

```

        f"Tinggi Matahari
{float(sun_alt[i]):.3f}° | "
        f"Iluminasi {float(illum[i]):.3f}%"
    )
    return PhenomenonItem(
        "Kandidat Hilal Bulanan",
        name,
        value,
        desc,
    )

    # Start from H0 and continue day-by-day
until the first qualifying
    # date. This replaces the older negative H
labels entirely.
    emitted: set[int] = set()
    for idx in progression_indices:
        items.append(make_row(idx))
        emitted.add(idx)

    # Keep every qualifying date in the
selected month. Its H number is
    # always calculated from the most recent
local-date conjunction.
    for idx in match_indices:
        if idx in emitted:
            continue
        items.append(make_row(idx))
        emitted.add(idx)

    if matches == 0:
        items.append(PhenomenonItem(
            "Kandidat Hilal Bulanan",
            "Tidak ada tanggal lolos",
            "_",
            "Tidak ada Maghrib pada bulan
terpilih yang sekaligus memenuhi kedua ambang
tersebut.",
        ))
    return items
except Exception as exc:
    return [PhenomenonItem(
        "Kandidat Hilal Bulanan",
        f"Ringkasan {year:04d}-{month:02d}",
        "Gagal dihitung",
        f"Pemindaian bulanan tidak tersedia:
{exc}",
    )]

```

```

def _phenomena(
    self,
    *,
    current_utc: datetime,
    local_dt: datetime,
    observer,
    topos,
    timezone_name: str,
    sun_result: BodyResult,
    moon_result: BodyResult,
    twilight_state: str,
) -> list[PhenomenonItem]:
    tz = ZoneInfo(timezone_name)
    items: list[PhenomenonItem] = []

    items.append(PhenomenonItem(
        "Kondisi lokal",
        "Keadaan cahaya langit",
        twilight_state,
        "Klasifikasi Skyfield: malam, twilight
astronomis/nautika/sipil, atau siang berdasarkan posisi
Matahari.",
    ))
    items.append(PhenomenonItem(
        "Bulan",
        "Fase saat perhitungan",
        f"{moon_result.phase_name} •
{moon_result.phase_cycle_deg:.3f}°" if
moon_result.phase_cycle_deg is not None else
moon_result.phase_name,
        f"{moon_result.waxing_waning}; iluminasi
{moon_result.illumination_pct:.3f}%",
    ))

    # Fase Bulan berikutnya.
    try:
        t0 = self.ts.from_datetime(current_utc)
        t1 = self.ts.from_datetime(current_utc +
timedelta(days=40))
        times, phases = almanac.find_discrete(t0,
t1, almanac.moon_phases(self.eph))
        names = {
            0: ("Bulan Baru berikutnya", "Konjungsi
geosentris Matahari-Bulan; awal siklus fase
astronomis."),
            1: ("Perbani Awal berikutnya", "Fase
sekitar 90°; setengah piringan Bulan tampak terang."),

```

```

                2: ("Purnama berikutnya", "Oposisi
geosentris Matahari-Bulan; iluminasi mendekati
penuh."),
                3: ("Perbani Akhir berikutnya", "Fase
sekitar 270°; setengah piringan Bulan tampak terang."),
            }
            found = set()
            for ti, phase in zip(times, phases):
                code = int(phase)
                if code in found:
                    continue
                found.add(code)
                dt =
ti.utc_datetime().replace(tzinfo=timezone.utc).astimezo
ne(tz)

                title, desc = names[code]
                items.append(PhenomenonItem("Fase
Bulan", title, self._fmt_dt(dt), desc))
            except Exception:
                pass

            # Node Bulan berikutnya.
            try:
                t0 = self.ts.from_datetime(current_utc)
                t1 = self.ts.from_datetime(current_utc +
timedelta(days=35))
                times, states = almanac.find_discrete(t0,
t1, almanac.moon_nodes(self.eph))
                found = set()
                for ti, state in zip(times, states):
                    code = int(state)
                    if code in found:
                        continue
                    found.add(code)
                    dt =
ti.utc_datetime().replace(tzinfo=timezone.utc).astimezo
ne(tz)

                    if code == 1:
                        title = "Node naik Bulan
berikutnya"
                        desc = "Bulan melintasi ekliptika
dari selatan ke utara; kedekatan fase dengan node
berkaitan dengan musim gerhana."
                    else:
                        title = "Node turun Bulan
berikutnya"
                        desc = "Bulan melintasi ekliptika
dari utara ke selatan; kedekatan fase dengan node
berkaitan dengan musim gerhana."

```

```

            items.append(PhenomenonItem("Orbit
Bulan", title, self._fmt_dt(dt), desc))
        except Exception:
            pass

        # Perigee/apogee Bulan berikutnya.
        try:
            perigee =
self._next_distance_extremum(self.moon, current_utc,
40, "min", 1.0)
            apogee =
self._next_distance_extremum(self.moon, current_utc,
40, "max", 1.0)
            if perigee:
                dt, dist = perigee
                items.append(PhenomenonItem(
                    "Orbit Bulan", "Perigee Bulan
berikutnya",
                    f"{self._fmt_dt(dt.astimezone(tz))}
• {dist:,.0f} km",
                    "Jarak pusat Bumi-Bulan mencapai
minimum lokal; diameter sudut Bulan cenderung lebih
besar.",
                    ))
            if apogee:
                dt, dist = apogee
                items.append(PhenomenonItem(
                    "Orbit Bulan", "Apogee Bulan
berikutnya",
                    f"{self._fmt_dt(dt.astimezone(tz))}
• {dist:,.0f} km",
                    "Jarak pusat Bumi-Bulan mencapai
maksimum lokal; diameter sudut Bulan cenderung lebih
kecil.",
                    ))
        except Exception:
            pass

        # Ekuinoks/solstis berikutnya.
        try:
            t0 = self.ts.from_datetime(current_utc)
            t1 = self.ts.from_datetime(current_utc +
timedelta(days=370))
            times, seasons = almanac.find_discrete(t0,
t1, almanac.seasons(self.eph))
            season_names = {
                0: ("Ekuinoks Maret", "Matahari
melintasi ekuator langit ke utara; panjang siang dan
malam mendekati sama secara global."),

```

```

        1: ("Solstis Juni", "Deklinasi Matahari
mencapai ekstrem utara tahunan."),
        2: ("Ekuinoks September", "Matahari
melintasi ekuator langit ke selatan; panjang siang dan
malam mendekati sama secara global."),
        3: ("Solstis Desember", "Deklinasi
Matahari mencapai ekstrem selatan tahunan."),
    }
    for ti, code in zip(times, seasons):
        dt =
ti.utc_datetime().replace(tzinfo=timezone.utc).astimezo
ne(tz)
        title, desc = season_names[int(code)]
        items.append(PhenomenonItem("Matahari",
title, self._fmt_dt(dt), desc))
    except Exception:
        pass

    # Perihelion/aphelion Bumi (jarak Matahari
tampak berubah sedikit).
    try:
        peri =
self._next_distance_extremum(self.sun, current_utc,
380, "min", 6.0)
        aph =
self._next_distance_extremum(self.sun, current_utc,
380, "max", 6.0)
        if peri:
            dt, dist = peri
            items.append(PhenomenonItem(
                "Matahari", "Perihelion Bumi
berikutnya",
                f"{self._fmt_dt(dt.astimezone(tz))}
• {dist / AU_KM:.6f} AU",
                "Bumi berada pada minimum lokal
jarak ke Matahari; diameter sudut Matahari sedikit
lebih besar.",
            ))
        if aph:
            dt, dist = aph
            items.append(PhenomenonItem(
                "Matahari", "Aphelion Bumi
berikutnya",
                f"{self._fmt_dt(dt.astimezone(tz))}
• {dist / AU_KM:.6f} AU",
                "Bumi berada pada maksimum lokal
jarak ke Matahari; diameter sudut Matahari sedikit
lebih kecil.",
            ))

```

```

        except Exception:
            pass

        # Gerhana Bulan: Skyfield memiliki rutin high-
        level resmi.
        try:
            t0 = self.ts.from_datetime(current_utc)
            t1 = self.ts.from_datetime(current_utc +
timedelta(days=5 * 366))
            times, kinds, _details =
eclipselib.lunar_eclipses(t0, t1, self.eph)
            if len(times):
                ti = times[0]
                kind = int(kinds[0])
                type_map = {0: "Penumbral", 1:
"Sebagian", 2: "Total"}
                dt_utc =
ti.utc_datetime().replace(tzinfo=timezone.utc)
                dt_local = dt_utc.astimezone(tz)
                try:
                    alt, _, _ =
observer.at(ti).observe(self.moon).apparent().altaz()
                    moon_alt = float(alt.degrees)
                    visibility = "Bulan di atas ufuk
lokasi" if moon_alt > 0 else "Bulan di bawah ufuk
lokasi pada maksimum"
                    extra = f"{visibility}; tinggi
Bulan {moon_alt:.2f}°."
                except Exception:
                    extra = "Visibilitas lokal tidak
dapat dihitung."
                items.append(PhenomenonItem(
                    "Gerhana", "Gerhana Bulan
terdekat",
                    f"{type_map.get(kind, 'Gerhana
Bulan')}" • {self._fmt_dt(dt_local)}",
                    "Momen maksimum menurut algoritma
gerhana Bulan Skyfield. " + extra,
                    ))
            except Exception:
                pass

        # Kandidat gerhana Matahari global berdasarkan
        geometri New Moon.
        # Ini sengaja diberi label kandidat, bukan
        klasifikasi resmi jalur gerhana.
        try:
            candidate =
self._next_solar_eclipse_candidate(current_utc, 800)

```

```

        if candidate:
            dt, sep, threshold = candidate
            items.append(PhenomenonItem(
                "Gerhana", "Kandidat gerhana
Matahari global",
                self._fmt_dt(dt.astimezone(tz)),
                f"New Moon dekat node: separasi
geosentris {sep:.3f}°, ambang geometrik kasar
{threshold:.3f}°."
                "Ini indikator global, bukan
prediksi jalur/jenis resmi; verifikasi dengan katalog
gerhana NASA/JPL bila diperlukan.",
            ))
        except Exception:
            pass

    return items

    def _eclipse_report(self, *, selected_local_dt,
observer, timezone_name):
        """Gerhana Matahari/Bulan pada bulan dan tahun
tanggal terpilih.

        Gerhana Bulan memakai routine resmi Skyfield
`eclipselib.lunar_eclipses`.
        Gerhana Matahari dideteksi dari geometri
kerucut bayangan Bulan-Matahari
terhadap Bumi menggunakan posisi DE440s.
Wilayah yang ditampilkan adalah
ringkasan geometrik global dan status
visibilitas lokasi terpilih, bukan
peta batas kontak rinci seperti katalog
Besselian.
        """
        tz = ZoneInfo(timezone_name)
        year = selected_local_dt.year
        month = selected_local_dt.month

        year_start_local = datetime(year, 1, 1,
tzinfo=tz)
        year_end_local = datetime(year + 1, 1, 1,
tzinfo=tz)
        start_utc =
year_start_local.astimezone(timezone.utc)
        end_utc =
year_end_local.astimezone(timezone.utc)
        t0 = self.ts.from_datetime(start_utc)
        t1 = self.ts.from_datetime(end_utc)

```

```

events = []

# ---- Gerhana Bulan ----
try:
    times, kinds, _details =
eclipselib.lunar_eclipses(t0, t1, self.eph)
    lunar_type = {0: "Penumbral", 1:
"Sebagian", 2: "Total"}
    for ti, kind in zip(times, kinds):
        dt_utc =
ti.utc_datetime().replace(tzinfo=timezone.utc)
        dt_local = dt_utc.astimezone(tz)
        if dt_local.year != year:
            continue

        try:
            alt, _, _ =
observer.at(ti).observe(self.moon).apparent().altaz()
            moon_alt = float(alt.degrees)
            local_status = (
                f"Lokasi terpilih berpotensi
melihat maksimum (tinggi Bulan {moon_alt:.2f}°). "
                if moon_alt > 0.0
                else f"Lokasi terpilih tidak
melihat maksimum karena Bulan di bawah ufuk
({moon_alt:.2f}°). "
            )
        except Exception:
            local_status = "Status visibilitas
lokasi terpilih tidak dapat dihitung."

        center = self._subpoint_text(self.moon,
ti)
        events.append({
            "object": "Bulan",
            "local_datetime":
dt_local.isoformat(),
            "utc_datetime": dt_utc.isoformat(),
            "type": lunar_type.get(int(kind),
"Gerhana Bulan"),
            "region": (
                "Dapat diamati dari sisi malam
Bumi yang pada saat maksimum memiliki Bulan di atas
ufuk. "
                f"Pusat hemisfer visibilitas
sekitar {center}. {local_status}"
            ),
            "month": dt_local.month,
        })

```

```

except Exception:
    pass

    # ---- Gerhana Matahari ----
    # Semua New Moon dalam tahun dipindai, lalu
    geometri penumbra/umbra
    # diperiksa terhadap bola Bumi. Momen
    diperhalus di sekitar New Moon.
    try:
        phase_times, phases =
almanac.find_discrete(t0, t1,
almanac.moon_phases(self.eph))
        for ti, phase in zip(phase_times, phases):
            if int(phase) != 0:
                continue
            solar = self._solar_eclipse_event(ti,
observer, tz)
            if solar is not None and
solar["local_dt"].year == year:
                events.append({
                    "object": "Matahari",
                    "local_datetime":
solar["local_dt"].isoformat(),
                    "utc_datetime":
solar["utc_dt"].isoformat(),
                    "type": solar["type"],
                    "region": solar["region"],
                    "month":
solar["local_dt"].month,
                })
    except Exception:
        pass

    events.sort(key=lambda x: x["utc_datetime"])
    month_events = [e for e in events if
int(e.get("month", 0)) == month]
    return {
        "selected_year": year,
        "selected_month": month,
        "timezone_name": timezone_name,
        "month_events": month_events,
        "year_events": events,
        "note": (
            "Gerhana Bulan diklasifikasikan dengan
Skyfield eclipselib. "
            "Gerhana Matahari dihitung dari
geometri kerucut bayangan DE440s; "
            "wilayah adalah ringkasan astronomis,
bukan peta jalur kontak rinci."
        )
    }

```

```

        ),
    }

    def _solar_eclipse_event(self, new_moon_time,
observer, tz):
        """Kembalikan event gerhana Matahari bila
penumbra Bulan memotong Bumi."""
        center_utc =
new_moon_time.utcnow().replace(tzinfo=timezone.ut
c)

        # Penyaring cepat pada momen New Moon agar
bulan-bulan yang jauh dari
        # musim gerhana tidak menjalankan pencarian
halus yang tidak diperlukan.
        initial_geom =
self._solar_shadow_geometry(new_moon_time)
        if initial_geom is None:
            return None
        initial_limit = (
            EARTH_EQUATORIAL_RADIUS_KM
            + initial_geom["penumbra_radius_km"]
            + 5000.0
        )
        if initial_geom["axis_distance_km"] >
initial_limit:
            return None

        # Cari pendekatan minimum sumbu bayangan ke
pusat Bumi ±8 jam.
        coarse_dts = [
            center_utc + timedelta(minutes=m)
            for m in range(-8 * 60, 8 * 60 + 1, 5)
        ]
        coarse_t = self.ts.from_datetimes(coarse_dts)
        coarse_dist =
self._shadow_axis_distance_km(coarse_t)
        if len(coarse_dist) == 0:
            return None
        idx = int(np.nanargmin(coarse_dist))
        rough = coarse_dts[idx]

        fine_dts = [
            rough + timedelta(seconds=s)
            for s in range(-8 * 60, 8 * 60 + 1, 20)
        ]
        fine_t = self.ts.from_datetimes(fine_dts)
        fine_dist =
self._shadow_axis_distance_km(fine_t)

```

```

idx = int(np.nanargmin(fine_dist))
dt_utc = fine_dts[idx]
ti = self.ts.from_datetime(dt_utc)

geom = self._solar_shadow_geometry(ti)
if geom is None:
    return None

    # Penumbra harus menyentuh bola Bumi agar
    terjadi gerhana Matahari.
    if geom["axis_distance_km"] >
EARTH_EQUATORIAL_RADIUS_KM +
geom["penumbra_radius_km"]:
    return None

    # Klasifikasi central/partial berdasarkan
    apakah sumbu bayangan
    # sendiri menembus bola Bumi. Untuk central,
    posisi vertex umbra
    # terhadap sisi dekat/jauh Bumi membedakan
    total/cincin/hibrida.
    axis_distance = geom["axis_distance_km"]
    if axis_distance <= EARTH_EQUATORIAL_RADIUS_KM:
        chord_half = math.sqrt(max(
            0.0,
            EARTH_EQUATORIAL_RADIUS_KM ** 2 -
axis_distance ** 2
        ))
        near = geom["axis_parameter_km"] -
chord_half
        far = geom["axis_parameter_km"] +
chord_half
        vertex = geom["umbra_vertex_km"]
        if vertex > far:
            eclipse_type = "Total"
        elif vertex < near:
            eclipse_type = "Cincin (Annular)"
        else:
            eclipse_type = "Hibrida"
    else:
        eclipse_type = "Sebagian"

    dt_local = dt_utc.astimezone(tz)
    ti = self.ts.from_datetime(dt_utc)

    try:
        sun_alt, _, _ =
observer.at(ti).observe(self.sun).apparent().altaz()

```

```

        moon_app =
observer.at(ti).observe(self.moon).apparent()
        sun_app =
observer.at(ti).observe(self.sun).apparent()
        local_sep =
float(moon_app.separation_from(sun_app).degrees)
        moon_dist = float(moon_app.distance().km)
        sun_dist = float(sun_app.distance().km)
        moon_sd = math.degrees(math.asin(min(1.0,
MOON_RADIUS_KM / moon_dist)))
        sun_sd = math.degrees(math.asin(min(1.0,
SUN_RADIUS_KM / sun_dist)))
        overlaps = local_sep <= (moon_sd + sun_sd)
and float(sun_alt.degrees) > 0.0
        local_status = (
            f"Lokasi terpilih berada pada area
gerhana saat sekitar maksimum; "
            f"tinggi Matahari
{float(sun_alt.degrees):.2f}°, separasi piringan
{local_sep:.3f}°."
            if overlaps
            else f"Lokasi terpilih tidak berada
pada area gerhana pada momen maksimum global "
            f"(tinggi Matahari
{float(sun_alt.degrees):.2f}°, separasi
{local_sep:.3f}°)."
        )
        except Exception:
            local_status = "Status visibilitas lokasi
terpilih tidak dapat dihitung."

        region_center =
self._shadow_axis_subpoint_text(ti, geom)
        region = (
            f"Penumbra mengenai Bumi; pusat geometri
sekitar {region_center}. "
            f"{local_status}"
        )
        return {
            "utc_dt": dt_utc,
            "local_dt": dt_local,
            "type": eclipse_type,
            "region": region,
        }

def _shadow_axis_distance_km(self, t):
    e = self.earth.at(t)

```

```

        moon_vec =
np.asarray(e.observe(self.moon).position.km,
dtype=float)
        sun_vec =
np.asarray(e.observe(self.sun).position.km,
dtype=float)
        if moon_vec.ndim == 1:
            moon_vec = moon_vec.reshape(3, 1)
            sun_vec = sun_vec.reshape(3, 1)
        axis = moon_vec - sun_vec
        axis /= np.linalg.norm(axis, axis=0)
        s = -np.sum(moon_vec * axis, axis=0)
        closest = moon_vec + axis * s
        return np.linalg.norm(closest, axis=0)

def _solar_shadow_geometry(self, t):
    try:
        e = self.earth.at(t)
        moon_vec =
np.asarray(e.observe(self.moon).position.km,
dtype=float).reshape(3)
        sun_vec =
np.asarray(e.observe(self.sun).position.km,
dtype=float).reshape(3)
        axis = moon_vec - sun_vec
        sun_moon_distance =
float(np.linalg.norm(axis))
        if sun_moon_distance <= 0.0:
            return None
        axis /= sun_moon_distance

        s = float(-np.dot(moon_vec, axis))
        if s <= 0.0:
            return None
        closest = moon_vec + axis * s
        axis_distance =
float(np.linalg.norm(closest))

        penumbra_radius = MOON_RADIUS_KM + s * (
            (SUN_RADIUS_KM + MOON_RADIUS_KM) /
sun_moon_distance
        )
        umbra_vertex = MOON_RADIUS_KM *
sun_moon_distance / (
            SUN_RADIUS_KM - MOON_RADIUS_KM
        )
        return {
            "axis": axis,
            "moon_vec": moon_vec,

```

```

        "closest_vec": closest,
        "axis_parameter_km": s,
        "axis_distance_km": axis_distance,
        "penumbra_radius_km":
float(penumbra_radius),
        "umbra_vertex_km": float(umbra_vertex),
    }
    except Exception:
        return None

    def _shadow_axis_subpoint_text(self, t, geom):
        """Perkiraan titik pusat lintasan pada bola
        Bumi (bukan geodesi Besselian)."""
        try:
            axis = np.asarray(geom["axis"],
dtype=float).reshape(3)
            moon_vec = np.asarray(geom["moon_vec"],
dtype=float).reshape(3)
            s0 = float(geom["axis_parameter_km"])
            d = float(geom["axis_distance_km"])
            if d <= EARTH_EQUATORIAL_RADIUS_KM:
                half = math.sqrt(max(0.0,
EARTH_EQUATORIAL_RADIUS_KM ** 2 - d ** 2))
                s_surface = s0 - half
                point = moon_vec + axis * s_surface
            else:
                point = np.asarray(geom["closest_vec"],
dtype=float).reshape(3)
                norm = float(np.linalg.norm(point))
                if norm <= 0.0:
                    return "koordinat tidak tersedia"
                point = point *
(EARTH_EQUATORIAL_RADIUS_KM / norm)
            return self._vector_subpoint_text(point, t)
        except Exception:
            return "koordinat tidak tersedia"

    def _subpoint_text(self, body, t):
        try:
            vec =
np.asarray(self.earth.at(t).observe(body).apparent().po
sition.km, dtype=float).reshape(3)
            return self._vector_subpoint_text(vec, t)
        except Exception:
            return "koordinat tidak tersedia"

    @staticmethod
    def _vector_subpoint_text(vec, t):
        x, y, z = [float(v) for v in vec]

```

```

        r = math.sqrt(x*x + y*y + z*z)
        if r <= 0.0:
            return "koordinat tidak tersedia"
        lat = math.degrees(math.asin(max(-1.0, min(1.0,
z / r))))
        ra_deg = math.degrees(math.atan2(y, x)) % 360.0
        lon = ((ra_deg - float(t.gast) * 15.0 + 180.0)
% 360.0) - 180.0
        ns = "LU" if lat >= 0 else "LS"
        ew = "BT" if lon >= 0 else "BB"
        return f"{abs(lat):.1f}° {ns}, {abs(lon):.1f}°
{ew}"

    def _next_distance_extremum(self, body, start_utc,
days, kind, step_hours):
        steps = int(days * 24 / step_hours) + 1
        dts = [start_utc + timedelta(hours=i *
step_hours) for i in range(steps)]
        t = self.ts.from_datetimes(dts)
        dist =
np.asarray(self.earth.at(t).observe(body).distance().km
, dtype=float)
        if len(dist) < 3:
            return None
        for i in range(1, len(dist) - 1):
            is_ext = (dist[i] < dist[i - 1] and dist[i]
< dist[i + 1]) if kind == "min" else (dist[i] > dist[i
- 1] and dist[i] > dist[i + 1])
            if is_ext:
                return
self._refine_distance_extremum(body, dts[i], kind,
step_hours)
        return None

    def _refine_distance_extremum(self, body,
center_utc, kind, coarse_step_hours):
        half = max(3.0, coarse_step_hours * 2.0)
        fine_dts = [
            center_utc + timedelta(minutes=m)
            for m in range(int(-half * 60), int(half *
60) + 1, 2)
        ]
        t = self.ts.from_datetimes(fine_dts)
        dist =
np.asarray(self.earth.at(t).observe(body).distance().km
, dtype=float)
        idx = int(np.nanargmin(dist) if kind == "min"
else np.nanargmax(dist))
        return fine_dts[idx], float(dist[idx])

```

```

    def _next_solar_eclipse_candidate(self, start_utc,
days):
        t0 = self.ts.from_datetime(start_utc)
        t1 = self.ts.from_datetime(start_utc +
timedelta(days=days))
        times, phases = almanac.find_discrete(t0, t1,
almanac.moon_phases(self.eph))
        for ti, phase in zip(times, phases):
            if int(phase) != 0:
                continue
            e = self.earth.at(ti)
            moon = e.observe(self.moon).apparent()
            sun = e.observe(self.sun).apparent()
            sep =
float(moon.separation_from(sun).degrees)
            moon_dist = float(moon.distance().km)
            sun_dist = float(sun.distance().km)
            moon_sd = math.degrees(math.asin(min(1.0,
MOON_RADIUS_KM / moon_dist)))
            sun_sd = math.degrees(math.asin(min(1.0,
SUN_RADIUS_KM / sun_dist)))
            moon_hp = math.degrees(math.asin(min(1.0,
EARTH_EQUATORIAL_RADIUS_KM / moon_dist)))
            threshold = moon_hp + moon_sd + sun_sd
            if sep <= threshold:
                dt =
ti.utc_datetime().replace(tzinfo=timezone.utc)
                return dt, sep, threshold
        return None

    @staticmethod
    def _first_real_event(times, flags, tz):
        for ti, ok in zip(times, flags):
            if bool(ok):
                return
ti.utc_datetime().replace(tzinfo=timezone.utc).astimezo
ne(tz)
        return None

    @staticmethod
    def _first_time(times, tz):
        if len(times) == 0:
            return None
        return
times[0].utc_datetime().replace(tzinfo=timezone.utc).as
timezone(tz)

    @staticmethod

```

```

def _fmt_dt(dt):
    if dt is None:
        return "-"
    return dt.strftime("%Y-%m-%d %H:%M:%S %Z")

# ===== Bridge JSON untuk antarmuka Web =====
_ENGINE = AstronomyEngine()

def activate_ephemeris(path_text):
    """Ganti kernel aktif tanpa mengubah parameter atau
    algoritme astronomi."""
    path, source = ensure_selected_ephemeris(path_text)
    if path == _ENGINE.ephemeris_path:
        return
    new_eph = load_file(str(path))
    try:
        earth, sun, moon = new_eph['earth'],
new_eph['sun'], new_eph['moon']
    except Exception:
        new_eph.close()
        raise
    previous = _ENGINE.eph
    _ENGINE.eph = new_eph
    _ENGINE.earth, _ENGINE.sun, _ENGINE.moon = earth,
sun, moon
    _ENGINE.ephemeris_path = path
    _ENGINE.ephemeris_source = source
    globals()['SOLUNA_EPHEMERIS_PATH'] = str(path)
    previous.close()

def _jsonable(value):
    if hasattr(value, '__dataclass_fields__'):
        return {name: _jsonable(getattr(value, name))
        for name in value.__dataclass_fields__}
    if isinstance(value, datetime):
        return value.isoformat()
    if isinstance(value, (list, tuple)):
        return [_jsonable(v) for v in value]
    if isinstance(value, dict):
        return {str(k): _jsonable(v) for k, v in
value.items()}
    try:
        if isinstance(value, np.generic):
            return value.item()
    except Exception:
        pass
    return value

```

```

def calculate_json(payload_json):
    p = json.loads(payload_json)
    require_ephemeris_for_date(p['date'])
    result = _ENGINE.calculate(
        location_name=p.get('location_name', 'Lokasi
Manual'),
        latitude=float(p['lat']),
        longitude=float(p['lon']),
        elevation_m=float(p.get('elev', 0.0)),
        timezone_name=p.get('timezone_name') or
'Asia/Jakarta',
        date_text=p['date'],
        time_text=p.get('time') or '12:00:00',
        time_mode=p.get('time_mode') or 'Jam Manual',
        mode=p.get('mode') or 'Toposentris',
    )
    out = _jsonable(result)
    out['engine'] = 'Skyfield + JPL DE440s
(browser/Pyodide)'
    return json.dumps(out, ensure_ascii=False)

def sunset_preview_json(payload_json):
    p = json.loads(payload_json)
    require_ephemeris_for_date(p['date'])
    dt = _ENGINE.local_sunset_datetime(
        latitude=float(p['lat']),
        longitude=float(p['lon']),
        elevation_m=float(p.get('elev', 0.0)),
        timezone_name=p.get('timezone_name') or
'Asia/Jakarta',
        date_text=p['date'],
    )
    return json.dumps({
        'local_datetime': dt.isoformat(),
        'clock': dt.strftime('%H:%M:%S'),
        'display': dt.strftime('%Y-%m-%d %H:%M:%S %Z'),
    }, ensure_ascii=False)
</script>

<script>
const BASE_CITIES=[{"name":"Abha","country":"Saudi
Arabia","lat":18.216389,"lon":42.505278,"elev":2228.0,"
tz":"Asia/Riyadh","search":""},{ "name":"Adana","country
":"Turkey","lat":37.0,"lon":35.32133,"elev":23.0,"tz":"
Europe/Istanbul","search":""},{ "name":"Alexandria","cou
ntry":"Egypt","lat":31.20176,"lon":29.91582,"elev":9.0,
"tz":"Africa/Cairo","search":""},{ "name":"Ambon","count
ry":"Indonesia","lat":-

```

3.6943237,"lon":128.1813034,"elev":11.0,"tz":"Asia/Jaya
 pura","search":"Maluku"},{"name":"Anchorage","country":
 "Alaska, USA","lat":61.2180556,"lon":-
 149.9002778,"elev":31.0,"tz":"America/Anchorage","searc
 h":""}, {"name":"Ankara","country":"Turkey","lat":39.919
 8744,"lon":32.8542709,"elev":850.0,"tz":"Europe/Istanbu
 l","search":""}, {"name":"Antalya","country":"Turkey","l
 at":36.908118,"lon":30.695565,"elev":61.0,"tz":"Europe/
 Istanbul","search":""}, {"name":"Aswan","country":"Egypt
 ","lat":24.09082,"lon":32.89942,"elev":99.0,"tz":"Afric
 a/Cairo","search":""}, {"name":"Auckland","country":"New
 Zealand","lat":-
 36.84853,"lon":174.76349,"elev":43.0,"tz":"Pacific/Auck
 land","search":""}, {"name":"Banda
 Aceh","country":"Indonesia","lat":5.5500499,"lon":95.31
 78267,"elev":11.0,"tz":"Asia/Jakarta","search":"Aceh"},
 {"name":"Bandung","country":"Indonesia","lat":-
 6.9175,"lon":107.6191,"elev":768.0,"tz":"Asia/Jakarta",
 "search":"Jawa
 Barat"}, {"name":"Buraidah","country":"Saudi
 Arabia","lat":26.325986,"lon":43.974966,"elev":650.0,"t
 z":"Asia/Riyadh","search":""}, {"name":"Byron
 Bay","country":"Australia","lat":-
 28.64989,"lon":153.612463,"elev":6.0,"tz":"Australia/Sy
 dney","search":"Cape Byron easternmost mainland
 Australia
 sunrise"}, {"name":"Cairo","country":"Egypt","lat":30.06
 26314,"lon":31.2496662,"elev":23.0,"tz":"Africa/Cairo",
 "search":""}, {"name":"Cold Bay","country":"Alaska,
 USA","lat":55.20656795,"lon":-
 162.71498738,"elev":30.0,"tz":"America/Anchorage","searc
 h":""}, {"name":"Dammam","country":"Saudi
 Arabia","lat":26.434417,"lon":50.103263,"elev":10.0,"tz
 ":"Asia/Riyadh","search":""}, {"name":"Denpasar","countr
 y":"Indonesia","lat":-
 8.6544589,"lon":115.2165333,"elev":32.0,"tz":"Asia/Maka
 ssar","search":"Bali"}, {"name":"Diyarbakir","country":"
 Turkey","lat":37.913629,"lon":40.217213,"elev":675.0,"t
 z":"Europe/Istanbul","search":""}, {"name":"El
 Arish","country":"Egypt","lat":31.13159,"lon":33.79844,
 "elev":32.0,"tz":"Africa/Cairo","search":""}, {"name":"E
 rzurum","country":"Turkey","lat":39.90861,"lon":41.2769
 4,"elev":1900.0,"tz":"Europe/Istanbul","search":""}, {"n
 ame":"Fairbanks","country":"Alaska,
 USA","lat":64.8377778,"lon":-
 147.7163889,"elev":136.0,"tz":"America/Anchorage","searc
 h":""}, {"name":"False Pass","country":"Alaska,
 USA","lat":54.85503086,"lon":-
 163.41473878,"elev":6.0,"tz":"America/Anchorage","searc

```

h":"","},{ "name": "Gaziantep", "country": "Turkey", "lat": 37.05944, "lon": 37.3825, "elev": 850.0, "tz": "Europe/Istanbul", "search": ""}, {"name": "Gisborne", "country": "New Zealand", "lat": -38.65333333, "lon": 178.0041667, "elev": 8.0, "tz": "Pacific/Auckland", "search": ""}, {"name": "Giza", "country": "Egypt", "lat": 30.009444, "lon": 31.208611, "elev": 30.0, "tz": "Africa/Cairo", "search": ""}, {"name": "Gold Coast", "country": "Australia", "lat": -28.00029, "lon": 153.430878, "elev": 12.0, "tz": "Australia/Brisbane", "search": ""}, {"name": "Hail", "country": "Saudi Arabia", "lat": 27.521876, "lon": 41.690732, "elev": 992.0, "tz": "Asia/Riyadh", "search": ""}, {"name": "Hawtat Sudair", "country": "Saudi Arabia", "lat": 25.596725, "lon": 45.627436, "elev": 738.0, "tz": "Asia/Riyadh", "search": "Sudair Sudayr حوطة سدير hilal crescent sighting rukyat"}, {"name": "Hobart", "country": "Australia", "lat": -42.8793606, "lon": 147.3294067, "elev": 20.0, "tz": "Australia/Hobart", "search": ""}, {"name": "Hurghada", "country": "Egypt", "lat": 27.25738, "lon": 33.81291, "elev": 14.0, "tz": "Africa/Cairo", "search": ""}, {"name": "Ismailia", "country": "Egypt", "lat": 30.60427, "lon": 32.27225, "elev": 13.0, "tz": "Africa/Cairo", "search": ""}, {"name": "Istanbul", "country": "Turkey", "lat": 41.013843, "lon": 28.9496613, "elev": 40.0, "tz": "Europe/Istanbul", "search": ""}, {"name": "Izmir", "country": "Turkey", "lat": 38.412726, "lon": 27.138376, "elev": 122.0, "tz": "Europe/Istanbul", "search": ""}, {"name": "Jakarta", "country": "Indonesia", "lat": -6.1778016, "lon": 106.8284444, "elev": 8.0, "tz": "Asia/Jakarta", "search": "DKI Jakarta"}, {"name": "Jayapura", "country": "Indonesia", "lat": -2.5626262, "lon": 140.6926331, "elev": 95.0, "tz": "Asia/Jayapura", "search": "Papua"}, {"name": "Jazan", "country": "Saudi Arabia", "lat": 16.889167, "lon": 42.551111, "elev": 40.0, "tz": "Asia/Riyadh", "search": ""}, {"name": "Jeddah", "country": "Saudi Arabia", "lat": 21.49012, "lon": 39.18624, "elev": 12.0, "tz": "Asia/Riyadh", "search": ""}, {"name": "Juneau", "country": "Alaska, USA", "lat": 58.3019444, "lon": -134.4197222, "elev": 17.0, "tz": "America/Juneau", "search": ""}, {"name": "King Cove", "country": "Alaska, USA", "lat": 55.05950824, "lon": -162.31126632, "elev": 27.0, "tz": "America/Anchorage", "search": ""}, {"name": "King Salmon", "country": "Alaska, USA", "lat": 58.6883, "lon": -156.6614, "elev": 17.0, "tz": "America/Anchorage", "search": ""}

```

```

    }, {"name": "Kodiak", "country": "Alaska,
USA", "lat": 57.79300136, "lon": -
152.39748118, "elev": 15.0, "tz": "America/Anchorage", "search": ""}, {"name": "Konya", "country": "Turkey", "lat": 37.871
354, "lon": 32.484636, "elev": 1016.0, "tz": "Europe/Istanbul
", "search": ""}, {"name": "Lord Howe
Island", "country": "Australia", "lat": -
31.55, "lon": 159.08, "elev": 5.0, "tz": "Australia/Lord_Howe
", "search": "eastern Australia island early sunrise Lord
Howe"}, {"name": "Luxor", "country": "Egypt", "lat": 25.69892
7, "lon": 32.6420975, "elev": 89.0, "tz": "Africa/Cairo", "search": ""}, {"name": "Madinah", "country": "Saudi
Arabia", "lat": 24.4686111, "lon": 39.6141667, "elev": 625.0,
"tz": "Asia/Riyadh", "search": "Medina المدينة المنورة Al
Madinah hilal
rukyyat"}, {"name": "Makassar", "country": "Indonesia", "lat":
:-
5.1332156, "lon": 119.4078369, "elev": 6.0, "tz": "Asia/Makas
sar", "search": "Sulawesi
Selatan"}, {"name": "Makkah", "country": "Saudi
Arabia", "lat": 21.42664, "lon": 39.825632, "elev": 277.0, "tz":
"Asia/Riyadh", "search": "Mecca مكة Makkah al
Mukarramah hilal
rukyyat"}, {"name": "Mallacoota", "country": "Australia", "lat":
:-
37.5603, "lon": 149.7569, "elev": 22.0, "tz": "Australia/Melb
ourne", "search": "Cape Howe eastern Victoria southeast
Australia sunrise"}, {"name": "Marsa
Matruh", "country": "Egypt", "lat": 31.35254, "lon": 27.24528
, "elev": 8.0, "tz": "Africa/Cairo", "search": ""}, {"name": "N
ajran", "country": "Saudi
Arabia", "lat": 17.493259, "lon": 44.127659, "elev": 1293.0, "
tz": "Asia/Riyadh", "search": ""}, {"name": "Nome", "country":
"Alaska, USA", "lat": 64.501111, "lon": -
165.406389, "elev": 6.0, "tz": "America/Nome", "search": ""},
{"name": "Pontianak", "country": "Indonesia", "lat": -
0.0220864, "lon": 109.3381039, "elev": 5.0, "tz": "Asia/Jakar
ta", "search": "Kalimantan Barat"}, {"name": "Port
Heiden", "country": "Alaska,
USA", "lat": 56.93007699, "lon": -
158.61901559, "elev": 29.0, "tz": "America/Anchorage", "search": ""}, {"name": "Port
Said", "country": "Egypt", "lat": 31.25654, "lon": 32.28412, "
elev": 6.0, "tz": "Africa/Cairo", "search": ""}, {"name": "Riy
adh", "country": "Saudi
Arabia", "lat": 24.6877323, "lon": 46.7218494, "elev": 612.0,
"tz": "Asia/Riyadh", "search": ""}, {"name": "Sabang", "count
ry": "Indonesia", "lat": 5.889687448369128, "lon": 95.316438
67492676, "elev": 8.0, "tz": "Asia/Jakarta", "search": "Aceh

```

Kota Sabang Pulau Weh"}, {"name": "Sand Point", "country": "Alaska, USA", "lat": 55.34090243, "lon": -160.49809006, "elev": 7.0, "tz": "America/Anchorage", "search": ""}, {"name": "Sanliurfa", "country": "Turkey", "lat": 37.16708, "lon": 38.79392, "elev": 518.0, "tz": "Europe/Istanbul", "search": ""}, {"name": "Semarang", "country": "Indonesia", "lat": -6.966667, "lon": 110.416664, "elev": 8.0, "tz": "Asia/Jakarta", "search": "Kota Semarang Jawa Tengah"}, {"name": "Shagra", "country": "Saudi Arabia", "lat": 25.240187, "lon": 45.251045, "elev": 730.0, "tz": "Asia/Riyadh", "search": "Shagra Shagrah شقراء hilal crescent sighting rukyat"}, {"name": "Sharm el-Sheikh", "country": "Egypt", "lat": 27.91582, "lon": 34.32995, "elev": 11.0, "tz": "Africa/Cairo", "search": ""}, {"name": "Suez", "country": "Egypt", "lat": 29.973714, "lon": 32.526267, "elev": 5.0, "tz": "Africa/Cairo", "search": ""}, {"name": "Surabaya", "country": "Indonesia", "lat": -7.2575, "lon": 112.7521, "elev": 5.0, "tz": "Asia/Jakarta", "search": "Jawa Timur"}, {"name": "Sydney", "country": "Australia", "lat": -33.86785, "lon": 151.207323, "elev": 58.0, "tz": "Australia/Sydney", "search": ""}, {"name": "Tabuk", "country": "Saudi Arabia", "lat": 28.399804, "lon": 36.571505, "elev": 768.0, "tz": "Asia/Riyadh", "search": ""}, {"name": "Taif", "country": "Saudi Arabia", "lat": 21.270278, "lon": 40.415833, "elev": 1879.0, "tz": "Asia/Riyadh", "search": ""}, {"name": "Trabzon", "country": "Turkey", "lat": 41.005, "lon": 39.72694, "elev": 39.0, "tz": "Europe/Istanbul", "search": ""}, {"name": "Tumair", "country": "Saudi Arabia", "lat": 25.70393, "lon": 45.86796, "elev": 670.0, "tz": "Asia/Riyadh", "search": "Tumayr تمر hilal crescent sighting rukyat"}, {"name": "Unalaska", "country": "Alaska, USA", "lat": 53.87309246, "lon": -166.53768178, "elev": 4.0, "tz": "America/Anchorage", "search": "Dutch Harbor Aleutian Islands"}, {"name": "Van", "country": "Turkey", "lat": 38.494565, "lon": 43.383231, "elev": 1730.0, "tz": "Europe/Istanbul", "search": ""}, {"name": "Wales", "country": "Alaska, USA", "lat": 65.6052, "lon": -168.0862, "elev": 8.0, "tz": "America/Nome", "search": "Seward Peninsula western mainland America Alaska"}, {"name": "Wellington", "country": "New Zealand", "lat": -41.2866428425807, "lon": 174.77557182312, "elev": 13.0, "tz": "Pacific/Auckland", "search": ""}, {"name": "Wilsons Promontory", "country": "Australia", "lat": -

```

39.00531,"lon":146.37634,"elev":20.0,"tz":"Australia/Me
lbourne","search":"South Point southernmost mainland
Australia sunrise"},{"name":"Yanbu","country":"Saudi
Arabia","lat":24.08954,"lon":38.0618,"elev":8.0,"tz":"A
sia/Riyadh","search":""}, {"name":"Yogyakarta","country"
:"Indonesia","lat":-
7.7956,"lon":110.3695,"elev":113.0,"tz":"Asia/Jakarta",
"search":"DIY"}]];
let CITIES=[...BASE_CITIES], pyodide=null, pyCalc=null,
pySunset=null, lastResult=null, coordMode='Geosentris',
timeMode='Maghrib Lokal';
// Prioritas: BSP tahunan 2026-2030, lalu BSP rentang
yang mencakup tahun pilihan.
// Daftar ini hanya metadata; tidak memuat atau
memeriksa semua berkas.
const SOLUNA_EPHEMERIS_KERNELS=[
  {
    "year_min": 2026,
    "year_max": 2026,
    "kind": "annual",
    "file": "de440s_2026.bsp",
    "size_bytes": 677888
  },
  {
    "year_min": 2027,
    "year_max": 2027,
    "kind": "annual",
    "file": "de440s_2027.bsp",
    "size_bytes": 679936
  },
  {
    "year_min": 2028,
    "year_max": 2028,
    "kind": "annual",
    "file": "de440s_2028.bsp",
    "size_bytes": 678912
  },
  {
    "year_min": 2029,
    "year_max": 2029,
    "kind": "annual",
    "file": "de440s_2029.bsp",
    "size_bytes": 677888
  },
  {
    "year_min": 2030,
    "year_max": 2030,
    "kind": "annual",
    "file": "de440s_2030.bsp",

```

```

    "size_bytes": 677888
  },
  {
    "year_min": 1850,
    "year_max": 1949,
    "file": "de440s_1850_1949.bsp",
    "size_bytes": 11563008,
    "kind": "range"
  },
  {
    "year_min": 1950,
    "year_max": 2049,
    "file": "de440s_1950_2049.bsp",
    "size_bytes": 11669504,
    "kind": "range"
  },
  {
    "year_min": 2050,
    "year_max": 2149,
    "file": "de440s_2050_2149.bsp",
    "size_bytes": 11022336,
    "kind": "range"
  }
];
const unavailableAnnualKernels=new Set();
let activeEphemeris=null, pyActivate=null,
engineReady=false, ephemerisQueue=Promise.resolve(),
selectionRevision=0;
function ephemerisForDate(dateText){
  const match=/^(\d{4})-\d{2}-
\d{2}$/.exec(String(dateText||''));
  const year=match?Number(match[1]):NaN;
  const
kernel=SOLUNA_EPHEMERIS_KERNELS.filter(k=>year>=k.year_min&&year<=k.year_max&&!unavailableAnnualKernels.has(k.file)).sort((a,b)=>(a.year_max-a.year_min)-(b.year_max-b.year_min))[0];
  if(!kernel)throw new Error('Tahun pilihan harus berada pada rentang 1850-2149 sesuai data DE440s yang tersedia.');
```

```

  return kernel;
}
function rangeFallbackFor(kernel){
  const
fallback=SOLUNA_EPHEMERIS_KERNELS.filter(k=>k.kind==='range'&&k.year_min<=kernel.year_min&&k.year_max>=kernel.year_max).sort((a,b)=>(a.year_max-a.year_min)-(b.year_max-b.year_min))[0];

```

```

    if(!fallback)throw new Error(`BSP tahunan
    ${kernel.file} tidak tersedia dan tidak ada BSP rentang
    untuk tahun tersebut.`);
    return fallback;
}
function ephemerisYearLabel(kernel){return
kernel.year_min===kernel.year_max?String(kernel.year_mi
n):`${kernel.year_min}-${kernel.year_max}`}
async function
loadSelectedEphemeris(kernel,initial=false){

if(kernel.kind==='annual'&&unavailableAnnualKernels.has
(kernel.file))kernel=rangeFallbackFor(kernel);

if(['de440s.bsp','de440s,bsp'].includes(String(kernel.f
ile).toLowerCase()))throw new Error('BSP asli tidak
boleh diakses. Gunakan BSP tahunan atau rentang
tahun.');
```

```

if(!initial&&activeEphemeris&&activeEphemeris.file===ke
rnel.file)return kernel;
    const range=ephemerisYearLabel(kernel);
    setLoad(true,'Menyiapkan data astronomi','Memuat
    ${kernel.file} untuk tahun ${range}...',68);
    const response=await
    fetch('./'+kernel.file,{cache:'force-cache'});
    if(!response.ok){
        // Hanya berkas tahunan yang benar-benar tidak ada
        boleh memakai rentang cadangan.

if(kernel.kind==='annual'&&(response.status===404||resp
onse.status===410)){
    unavailableAnnualKernels.add(kernel.file);
    return
loadSelectedEphemeris(rangeFallbackFor(kernel),initial)
;
    }
    throw new Error(`${kernel.file} gagal dibaca (HTTP
    ${response.status}). Letakkan BSP tahun pilihan sejajar
    dengan index.php.`);
    }
    const buffer=await response.arrayBuffer(),bytes=new
    Uint8Array(buffer);
    if(buffer.byteLength!==kernel.size_bytes)throw new
    Error(`Ukuran ${kernel.file} tidak sesuai. Unggah
    kembali BSP lengkap dari paket ini.`);
    if(!new
    TextDecoder('ascii').decode(bytes.slice(0,8)).startsWith

```

```

h('DAF/SPK'))throw new Error(`${kernel.file} bukan SPK
valid.`);
const path='/home/pyodide/'+kernel.file;
const
previousPath=activeEphemeris?'/home/pyodide/'+activeEph
emeris.file:null;
pyodide.FS.writeFile(path,bytes);
try{
  if(initial){
    pyodide.globals.set('SOLUNA_EPHEMERIS_PATH',path);

pyodide.globals.set('SOLUNA_EPHEMERIS_CONFIG',JSON.stri
ngify(SOLUNA_EPHEMERIS_KERNELS));
    setLoad(true,'Menyiapkan SOLUNA','Mengaktifkan
Skyfield dan kernel JPL DE440s...',84);
    await pyodide.runPythonAsync(`${'py-
engine'}).textContent);

pyCalc=pyodide.globals.get('calculate_json');pySunset=p
yodide.globals.get('sunset_preview_json');pyActivate=py
odide.globals.get('activate_ephemeris');
  }else await pyActivate(path);
}catch(error){
  try{pyodide.FS.unlink(path)}catch(ignored){}
  throw error;
}
activeEphemeris=kernel;

if(previousPath&&previousPath!==path){try{pyodide.FS.un
link(previousPath)}catch(ignored){}}
startEngineBadgeClock();

if(!initial&&!$('loading').classList.contains('calculat
ing'))setLoad(false);
return kernel;
}
function
withSelectedEphemeris(dateText,action,isCurrent=()=>tru
e){
  let kernel;

try{kernel=ephemerisForDate(dateText)}catch(error){retu
rn Promise.reject(error)}
  const job=ephemerisQueue.catch(()=>{}).then(async()=>{
    // Abaikan pratinjau lama sebelum membaca BSP atau
menjalankan Python.
    if(!isCurrent())return null;
    if(!engineReady||!pyodide)throw new Error('Engine
belum siap.');
```

```

    const loadedKernel=await
loadSelectedEphemeris(kernel);
    if(!isCurrent())return null;
    return action?await action():loadedKernel;
});
ephemerisQueue=job.catch(()=>{});
return job;
}
function
ensureSelectedEphemeris(dateText,isCurrent){return
withSelectedEphemeris(dateText,null,isCurrent)}
function handleSelectedDateChange(reason='perubahan
tanggal'){
    if(timeMode==='Maghrib
Lokal'){refreshMaghribTime(reason);return}
    const
revision=++selectionRevision,date=$('#date').value;
    const
isCurrent=()=>revision===selectionRevision&&date===$('#d
ate').value;

if(engineReady)ensureSelectedEphemeris(date,isCurrent).
catch(error=>{if(!isCurrent())return;setLoad(false);set
Status('Gagal memilih data astronomi:
'+error.message,'err');toast(error.message)});
}
let engineBadgeClock=null;
function updateEngineBadgeClock(){
    const now=new Date();
    const date=now.toLocaleDateString('id-ID',{day:'2-
digit',month:'long',year:'numeric'});
    const
time=[now.getHours(),now.getMinutes(),now.getSeconds()]
.map(value=>String(value).padStart(2,'0')).join(':');
    $('#engineBadge').textContent=`SKYFIELD • DE440S •
${date} • ${time}`;
}
function startEngineBadgeClock(){
    updateEngineBadgeClock();

if(engineBadgeClock===null)engineBadgeClock=setInterval
(updateEngineBadgeClock,1000);
}
const $=id=>document.getElementById(id);
const esc=s=>String(s??'–
').replace(/[\&<>"']/g,m=>({'&':'&amp;','<':'&lt;','>':'&gt;','"':'&quot;','"':'&#39;'}[m]));
function
toast(s){$('#toast').textContent=fmtDisplayValue(s);$('#t

```

```

oast').classList.add('show');setTimeout(()=>$('toast').
classList.remove('show'),2200)}
function
setLoad(show,title='',text='',pct=10){$('loading').clas
sList.toggle('show',show);if(title){('loadTitle').textC
ontent=title;if(text){('loadText').textContent=text;$('
loadBar').style.width=pct+'%'}
function setStatus(s,kind='') {
$('statusLine').className='statusline '+kind;
$('statusLine').textContent=fmtDisplayValue(s); }
function formatNum(v,d=3){return
v==null||Number.isNaN(Number(v))?'-
':Number(v).toFixed(d)}
function raHms(h){if(h==null)return'-
';h=((+h)%24+24)%24;let H=Math.floor(h),mf=(h-
H)*60,M=Math.floor(mf),S=(mf-M)*60;return
`${String(H).padStart(2,'0')}${h}
${String(M).padStart(2,'0')}${m}
${S.toFixed(2).padStart(5,'0')}${s}`}
function haHms(h){if(h==null)return'-';let
sg=h>=0?'+':'-',x=Math.abs(+h),H=Math.floor(x),mf=(x-
H)*60,M=Math.floor(mf),S=(mf-M)*60;return
`${sg}${String(H).padStart(2,'0')}${h}
${String(M).padStart(2,'0')}${m}
${S.toFixed(2).padStart(5,'0')}${s}`}
function decDms(x){if(x==null)return'-';let
sg=x>=0?'+':'-',a=Math.abs(x),d=Math.floor(a),mf=(a-
d)*60,m=Math.floor(mf),s=(mf-m)*60;return
`${sg}${String(d).padStart(2,'0')}${°}
${String(m).padStart(2,'0')}${'}
${s.toFixed(2).padStart(5,'0')}${"}`}
function fmtDisplayValue(value){
  if(value==null)return '-';
  // Format Indonesia hanya untuk tampilan; data ISO
  asli tetap dipakai engine.
  return String(value)
    .replace(/\\b(\\d{4})-(\\d{2})-
(\\d{2})T(\\d{2}:\\d{2}:\\d{2})(?:\\.\\d+)?(Z|[-
+](\\d{2}:\\d{2}))?/g, (_,y,m,d,clock,offset)=>`${d}-${m}-${
y} ${clock}${offset?
'+(offset==='Z'||offset==='+00:00'||offset==='-
00:00'? 'UTC': 'UTC'+offset): ''}`)
    .replace(/\\b(\\d{4})-(\\d{2})-(\\d{2})\\b/g, '$3-$2-$1')
    .replace(/\\b(\\d{4})-(\\d{2})\\b/g, '$2-$1');
}
function fmtTime(iso,tz=($('tz').value){
  if(!iso)return '-';
  try{
    const date=new Date(iso);

```

```

    if(!Number.isFinite(date.getTime())) return
    fmtDisplayValue(iso);
    tz=tz||'Asia/Jakarta';
    const parts={};
    new Intl.DateTimeFormat('id-
ID',{timeZone:tz,year:'numeric',month:'2-digit',day:'2-
digit',hour:'2-digit',minute:'2-digit',second:'2-
digit',hourCycle:'h23',timeZoneName:'short'}).formatToP
arts(date).forEach(p=>{if(p.type!=='literal')parts[p.ty
pe]=p.value});
    const
    zone={'Asia/Jakarta':'WIB','Asia/Pontianak':'WIB','Asia
/Makassar':'WITA','Asia/Ujung_Pandang':'WITA','Asia/Jay
apura':'WIT','UTC':'UTC','Etc/UTC':'UTC','GMT':'UTC','E
tc/GMT':'UTC'}[tz]||(parts.timeZoneName||tz).replace(/^(
GMT/, 'UTC');
    return `${parts.day}-${parts.month}-${parts.year}
${parts.hour}:${parts.minute}:${parts.second} ${zone}`;
  }catch(e){return fmtDisplayValue(iso)}
}
function timezoneOffsetMinutes(dateText,timeText,tz){
  try{const base=new
Date(`${dateText}T${timeText}||'12:00:00'Z`);const
parts=new Intl.DateTimeFormat('en-
US',{timeZone:tz,year:'numeric',month:'2-digit',day:'2-
digit',hour:'2-digit',minute:'2-digit',second:'2-
digit',hour12:false}).formatToParts(base);let
o={};parts.forEach(p=>o[p.type]=p.value);let
h=+o.hour;if(h===24)h=0;const
asUTC=Date.UTC(+o.year,+o.month-
1,+o.day,h,+o.minute,+o.second);return
Math.round((asUTC-
base.getTime())/60000)}catch(e){return 0}
}
function nowInZone(tz){const d=new Date();let p={};new
Intl.DateTimeFormat('en-
CA',{timeZone:tz,year:'numeric',month:'2-digit',day:'2-
digit',hour:'2-digit',minute:'2-digit',second:'2-
digit',hour12:false}).formatToParts(d).forEach(x=>p[x.t
ype]=x.value);let h=p.hour==='24'?'00':p.hour;return
{date:`${p.year}-${p.month}-
${p.day}`,time:`${h}:${p.minute}:${p.second}`}}
function labelCity(c){return `${c.name}, ${c.country}`}
function renderCityOptions(filter='') {
  const q=filter.trim().toLocaleLowerCase('id');const
rows=CITIES.filter(c=>!q||(`${c.name} ${c.country}
${c.search}||'')`).toLocaleLowerCase('id').includes(q));
  const
  prev=$('#citySelect').value;$('#citySelect').innerHTML=ro

```

```

ws.map(c=><option
value="${esc(labelCity(c))}">${esc(labelCity(c))}</opti
on>`).join('');$('cityCount').textContent=`Database
browser: ${CITIES.length} lokasi • tampil
${rows.length}`;

if(rows.some(c=>labelCity(c)===prev))$('citySelect').va
lue=prev; else
if(rows.length){$('citySelect').value=labelCity(rows[0]
);applyCity(rows[0])}
}
function
applyCity(c){$('lat').value=c.lat;$('lon').value=c.lon;
$('elev').value=c.elev;$('tz').value=c.tz;if(!$('date')
.value)setNow();else
handleSelectedDateChange('perubahan kota')}
function selectedCity(){return
CITIES.find(c=>labelCity(c)===$('citySelect').value)}
function setNow(){let
z=($('tz').value||'Asia/Jakarta',n=nowInZone(z);$('date'
).value=n.date;$('time').value=n.time;handleSelectedDat
eChange('tanggal sekarang')}
async function mergeIndonesiaData(){try{let r=await
fetch('./data/indonesia_locations.json',{cache:'force-
cache'});if(!r.ok)return;let j=await
r.json(),rows=Array.isArray(j)?j:(j.locations||[]),seen
=new
Set(CITIES.map(c=>labelCity(c).toLowerCase()));for(const
x of rows){let
c={name:x.name||x.capital,country:'Indonesia',lat:+x.la
titude,lon:+x.longitude,elev:+(x.elevation_m||0),tz:x.t
imezone||'Asia/Jakarta',search:`${x.admin_area||''}
${x.province||''}
${x.search_terms||''}`};if(c.name&&!seen.has(labelCity(
c).toLowerCase())){CITIES.push(c);seen.add(labelCity(c)
.toLowerCase())}CITIES.sort((a,b)=>labelCity(a).locale
Compare(labelCity(b),'id'));renderCityOptions($('citySe
arch').value)}catch(e){}}
async function initEngine(){
  engineReady=false;
  try{
    // Setup pertama memakai tahun berjalan di zona waktu
    lokasi.
    const
    kernel=ephemerisForDate(nowInZone($('tz').value||'Asia/
    Jakarta').date);
    setLoad(true,'Menyiapkan SOLUNA','Memuat runtime
    Python WebAssembly

```

```

(Pyodide)...',12);$('engineBadge').textContent='Memuat
Pyodide...';
  pyodide=await
loadPyodide({indexURL:'https://cdn.jsdelivr.net/pyodide
/v0.27.7/full/'});
  setLoad(true,'Menyiapkan SOLUNA','Memuat NumPy dan
micropip...',30);await
pyodide.loadPackage(['numpy','micropip']);
  setLoad(true,'Menyiapkan SOLUNA','Memasang jplephem +
certifi + Skyfield (tanpa SGP4/TLE)...',48);
  await pyodide.runPythonAsync(`
import micropip
# Skyfield dipasang tanpa dependency resolver agar
Pyodide tidak mencoba
# memasang sgp4 (fitur satelit/TLE tidak digunakan
SOLUNA).
await micropip.install(['jplephem','certifi','tzdata'],
deps=False)
await micropip.install('skyfield', deps=False)
`);
  const loadedKernel=await
loadSelectedEphemeris(kernel,true);engineReady=true;
  setLoad(true,'SOLUNA siap','DE440s tahun
${ephemerisYearLabel(loadedKernel)} aktif`,100);
  setStatus(`Engine siap: ${loadedKernel.file}. BSP
tahunan didahulukan sesuai tahun
pilihan.`,'ok');setLoad(false);
  if(timeMode==='Maghrib
Lokal')refreshMaghribTime('engine siap');else
handleSelectedDateChange('engine siap');

}catch(e){console.error(e);$('engineBadge').textContent
='Engine gagal dimuat';setStatus('Gagal memuat engine:
'+e.message,'err');setLoad(true,'Engine
gagal',e.message,100);setTimeout(()=>setLoad(false),350
0)}
}
function table(headers,rows){return `

SOLUNA | 192


```

```

['Tinggi / altitude
geometrik',formatNum(r.sun.altitude_deg,6),formatNum(b.
altitude_deg,6),'derajat'],
['Tinggi dengan refraksi
standar',formatNum(r.sun.refracted_altitude_deg,6),form
atNum(b.refracted_altitude_deg,6),'derajat; estimasi
atmosfer standar'],

['Azimut',formatNum(r.sun.azimuth_deg,6),formatNum(b.az
imuth_deg,6),'0°=Utara, 90°=Timur'],
['Hour
angle',haHms(r.sun.hour_angle_hours),haHms(b.hour_angle
_hours),'lokal; negatif sebelum meridian'],
['Right
Ascension',raHms(r.sun.ra_hours),raHms(b.ra_hours),'app
arent, epoch tanggal'],

['Deklinasi',decDms(r.sun.dec_deg),decDms(b.dec_deg),'a
pparent, epoch tanggal'],
['Longitude
ekliptika',formatNum(r.sun.ecliptic_longitude_deg,6),fo
rmatNum(b.ecliptic_longitude_deg,6),'derajat;
geosentris apparent'],
['Latitude
ekliptika',formatNum(r.sun.ecliptic_latitude_deg,6),for
matNum(b.ecliptic_latitude_deg,6),'derajat; geosentris
apparent'],
['Konstelasi IAU',r.sun.constellation||'-
',b.constellation||'-','berdasarkan posisi
geosentris'],
['Elongasi dari
Matahari','0.000000',formatNum(b.elongation_deg,6),'der
ajat'],
['Fase Bulan 0-360°','- ',b.phase_cycle_deg==null?'-
':formatNum(b.phase_cycle_deg,6),'0° baru; 90° perbani
awal; 180° purnama'],
['Nama fase','- ',b.phase_name||'- ',b.waxing_waning||'-
'],

['Iluminasi',formatNum(r.sun.illumination_pct,4),format
Num(b.illumination_pct,4),'%'],
['Sudut
fase',formatNum(r.sun.phase_angle_deg,6),formatNum(b.ph
ase_angle_deg,6),'derajat'],
['Position angle sisi terang','-
',b.bright_limb_position_angle_deg==null?'-
':formatNum(b.bright_limb_position_angle_deg,6),'deraja
t; bright limb Bulan'],

```

```

    ['Diameter
sudut',formatNum(r.sun.angular_diameter_arcmin,6),forma
tNum(b.angular_diameter_arcmin,6),'arcmin'],

    ['Semidiameter',formatNum(r.sun.semidiameter_arcmin,6),
formatNum(b.semidiameter_arcmin,6),'arcmin'],
    ['Paralaks
horizontal',formatNum(r.sun.horizontal_parallax_deg,8),
formatNum(b.horizontal_parallax_deg,8),'derajat'],
    ['Magnitudo
visual',formatNum(r.sun.magnitude,3),formatNum(b.magnit
ude,3),'V, estimasi'],
    ['Jarak dari pusat
Bumi',Number(r.sun.distance_km).toLocaleString('en-
US',{minimumFractionDigits:3,maximumFractionDigits:3}),
Number(b.distance_km).toLocaleString('en-
US',{minimumFractionDigits:3,maximumFractionDigits:3}),
'km'],
    ['Jarak dari pusat
Bumi',formatNum(r.sun.distance_au,9),formatNum(b.distan
ce_au,9),'AU'],
    ['Jarak dari
pengamat',Number(r.sun.observer_distance_km).toLocaleSt
ring('en-
US',{minimumFractionDigits:3,maximumFractionDigits:3}),
Number(b.observer_distance_km).toLocaleString('en-
US',{minimumFractionDigits:3,maximumFractionDigits:3}),
'km; sesuai mode'],
    ['Beda tinggi Bulan - Matahari','-
',(r.relative_altitude_deg>=0?'+'+'')+formatNum(r.relati
ve_altitude_deg,6),'derajat'],
    ['Beda azimuth Bulan - Matahari','-
',(r.relative_azimuth_deg>=0?'+'+'')+formatNum(r.relati
ve_azimuth_deg,6),'derajat; rentang -180°..+180°'],
    ['Rise',e.sun_rise,e.moon_rise,'waktu
lokal'],['Set',e.sun_set,e.moon_set,'waktu
lokal'],['Transit
meridian',e.sun_transit,e.moon_transit,'waktu lokal'],
    ['Local Sidereal
Time',formatNum(e.local_sidereal_time_hours,8),formatNu
m(e.local_sidereal_time_hours,8),'jam'],
    ['Equation of
Time',e.equation_of_time_minutes==null?'-
':((e.equation_of_time_minutes>=0?'+'+'')+formatNum(e.e
quation_of_time_minutes,3)),'-','menit; apparent - mean
solar time'],
    ['Panjang siang',e.day_length_hours==null?'-
':formatNum(e.day_length_hours,6),'-','jam sunrise-
sunset'],

```

```

    ['Keadaan twilight',e.twilight_state||' ','-','pada
waktu perhitungan'],
    ['Age Bulan','-',formatNum(e.moon_age_days,6),'hari
sejak New Moon geosentris'],
    ['Lag time Bulan','-'
',lagText(e.lag_time_minutes),'moonset - sunset']
];}
function bodyRows(b,mode){let
altNote=mode==='Toposentris'?'toposentris':'proyeksi
horizon tanpa paralaks';return [
    ['Tinggi / altitude
geometrik',formatNum(b.altitude_deg,8),'derajat;
${altNote}`],[ 'Tinggi dengan refraksi
standar',formatNum(b.refracted_altitude_deg,8),'derajat
; estimasi atmosfer
standar'], ['Azimut',formatNum(b.azimuth_deg,8),'derajat
; 0=N, 90=E'], ['Hour
angle',haHms(b.hour_angle_hours),formatNum(b.hour_angle
_hours,10)+' jam'], ['Right
Ascension',raHms(b.ra_hours),formatNum(b.ra_hours,10)+'
jam'], ['Deklinasi',decDms(b.dec_deg),formatNum(b.dec_de
g,10)+'°'], ['Longitude
ekliptika',formatNum(b.ecliptic_longitude_deg,10),'dera
jat; true ecliptic of date'], ['Latitude
ekliptika',formatNum(b.ecliptic_latitude_deg,10),'deraja
t; true ecliptic of date'], ['Konstelasi
IAU',b.constellation||' ','-','posisi
geosentris'], ['Elongasi',formatNum(b.elongation_deg,8),
'derajat dari Matahari'], ['Fase 0-
360°',b.phase_cycle_deg==null?'-
':formatNum(b.phase_cycle_deg,8),'khusus Bulan'], ['Nama
fase',b.phase_name||' ','-','b.waxing_waning||'-'
'], ['Iluminasi',formatNum(b.illumination_pct,6),'%'], ['
Sudut
fase',formatNum(b.phase_angle_deg,8),'derajat'], ['Posit
ion angle sisi
terang',b.bright_limb_position_angle_deg==null?'-
':formatNum(b.bright_limb_position_angle_deg,8),'deraja
t; khusus Bulan'], ['Diameter
sudut',formatNum(b.angular_diameter_arcmin,8),'arcmin']
, ['Semidiameter',formatNum(b.semidiameter_arcmin,8),'ar
cmin'], ['Paralaks
horizontal',formatNum(b.horizontal_parallax_deg,10),'de
rajat'], ['Magnitudo
visual',formatNum(b.magnitude,4),'V; estimasi'], ['Jarak
geosentris',Number(b.distance_km).toLocaleString('en-
US',{minimumFractionDigits:3,maximumFractionDigits:3}),
'km dari pusat Bumi'], ['Jarak
geosentris',formatNum(b.distance_au,12),'AU'], ['Jarak

```

```

dari
pengamat',Number(b.observer_distance_km).toLocaleString
('en-
US',{minimumFractionDigits:3,maximumFractionDigits:3}),'
'km; sesuai mode']
];}
function eventRows(r){let
e=r.events,min=e.conjunction_topocentric_min_elong_deg=
=null?'-
':formatNum(e.conjunction_topocentric_min_elong_deg,8)+
'°';return [
['Fajar astronomis',e.astronomical_dawn,'awal twilight
astronomis; Matahari sekitar -18°'],['Fajar
nautika',e.nautical_dawn,'awal twilight nautika;
Matahari sekitar -12°'],['Fajar
sipil',e.civil_dawn,'awal twilight sipil; Matahari
sekitar -6°'],['Sunrise',e.sun_rise,'Azimut
${formatNum(e.sun_rise_azimuth_deg,3)}°;
topocentris`'],['Transit Matahari',e.sun_transit,'Tinggi
${formatNum(e.sun_transit_altitude_deg,3)}°; meridian
lokal`'],['Sunset',e.sun_set,'Azimut
${formatNum(e.sun_set_azimuth_deg,3)}°;
topocentris`'],['Senja sipil
berakhir',e.civil_dusk,'Matahari melewati sekitar
-6°'],['Senja nautika
berakhir',e.nautical_dusk,'Matahari melewati sekitar
-12°'],['Senja astronomis
berakhir',e.astronomical_dusk,'Matahari melewati
sekitar -18°'],['Panjang
siang',e.day_length_hours==null?'-
':formatNum(e.day_length_hours,6)+' jam','sunrise →
sunset'],['Equation of
Time',e.equation_of_time_minutes==null?'-
':((e.equation_of_time_minutes>=0?'+' ':'')+formatNum(e.e
quation_of_time_minutes,3)+' menit'),'apparent solar
time - mean solar time'],['Local Sidereal
Time',raHms(e.local_sidereal_time_hours),formatNum(e.lo
cal_sidereal_time_hours,8)+' jam'],['Keadaan cahaya
langit',e.twilight_state||'-','pada waktu
perhitungan'],['Moonrise',e.moon_rise,'Azimut
${formatNum(e.moon_rise_azimuth_deg,3)}°;
topocentris`'],['Transit Bulan',e.moon_transit,'Tinggi
${formatNum(e.moon_transit_altitude_deg,3)}°; meridian
lokal`'],['Moonset',e.moon_set,'Azimut
${formatNum(e.moon_set_azimuth_deg,3)}°;
topocentris`'],['Konjungsi / New Moon geosentris
terdekat',e.conjunction_geocentric,'fase New
Moon'],['Konjungsi topocentris
terdekat',e.conjunction_topocentric,'minimum separasi

```

```

    lokal'], ['Elongasi minimum topocentris', min, 'pada
    konjungsi lokal'], ['Age
    Bulan', formatNum(e.moon_age_days, 8) + ' hari', 'sejak New
    Moon geosentris terakhir'], ['Lag
    time', lagText(e.lag_time_minutes), 'moonset terdekat -
    sunset']
  ];
}
function globalHijriStartLabel(gh) {
  // Patokan penamaan: 1 Jumadilawal 1448 H = 12-10-
  2026.
  const
  monthNames = ['Muharam', 'Safar', 'Rabiulawal', 'Rabiulakhir',
    'Jumadilawal', 'Jumadilakhir', 'Rajab', 'Syakban', 'Ramad
    an', 'Syawal', 'Zulkaidah', 'Zulhijah'];
  const dateParts = /^(\d{4})-(\d{2})-(
    \d{2})$/ .exec(String(gh.hijri_start_gregorian_date || ''
    ));
  const
  conjunctionMs = Date.parse(gh.conjunction_local_datetime |
    '');
  if (!dateParts || !Number.isFinite(conjunctionMs)) return
    'Tanggal 1 Hijriah belum dapat ditentukan.';
  // Rerata sinodik hanya menomori lunasi konjungsi;
  tanggal awal bulan tetap hasil scan asli.
  const lunationOffset = Math.round((conjunctionMs -
    Date.UTC(2026, 9, 12)) / (29.530588853 * 86400000));
  const monthSerial = 1448 * 12 + 4 + lunationOffset;
  const hijriYear = Math.floor(monthSerial / 12);
  const monthIndex = (monthSerial % 12 + 12) % 12;
  return `Tanggal 1 ${monthNames[monthIndex]}
    ${hijriYear} H = ${dateParts[3]}-${dateParts[2]}-
    ${dateParts[1]}`;
}
function renderGlobalHijri(gh) {
  if (!gh) { $('globalHijriSummary').textContent = 'Data scan
    global PKG belum
    tersedia.'; $('globalScanTable').innerHTML = ''; return }
  const
  localZone = gh.reference_timezone_name || $('tz').value;
  const
  conjunction = gh.conjunction_local_datetime ? fmtTime(gh.co
    njunction_local_datetime, localZone) : 'tidak ditemukan';
  const
  conjunctionUTC = gh.conjunction_local_datetime ? fmtTime(gh
    .conjunction_local_datetime, 'UTC') : 'tidak ditemukan';

  if (!gh.window_eligible) { $('globalHijriSummary').innerHT
    ML = `<span class="global-status-off">KONJUNGSI TIDAK
    DITEMUKAN</span> • Bulan Gregorian UTC pilihan:

```

```

    ${esc(fmtDisplayValue(gh.selected_date))} • Scan PKG
    belum dapat
    dijalankan.`;$('#globalScanTable').innerHTML='';return}
    const rows=gh.cities||[],
    validation=gh.validation_sites||[];
    const
    groups=[0,1,2].map(h=>({h,rows:rows.filter(row=>Number(
    row.h_day)===h)}));
    const
    dawnLocal=gh.gisborne_dawn_local_datetime?fmtTime(gh.gi
    sborne_dawn_local_datetime,'Pacific/Auckland'):'Tidak
    tersedia';
    const
    dawnUTC=gh.gisborne_dawn_utc_datetime?fmtTime(gh.gisbor
    ne_dawn_utc_datetime,'UTC'):'Tidak tersedia';
    const beforeDawn=gh.conjunction_before_gisborne_dawn;
    const dawnStatus=beforeDawn===true?'IJTIMAK SEBELUM
    FAJAR':beforeDawn===false?'IJTIMAK TIDAK SEBELUM
    FAJAR':'FAJAR BELUM TERSEDIA';
    const audit=`<div class="global-pkg-
    audit"><div><small>Ijtimak
    geosentris</small><b>${esc(conjunctionUTC)}</b><span>Lo
    kasi utama:
    ${esc(conjunction)}</span></div><div><small>Batas PKG
    pada hari
    keputusan</small><b>${esc(fmtTime(gh.utc_midnight,'UTC'
    ))}</b><span>PKG 1: sebelum batas • PKG 2: pada/setelah
    batas</span></div><div><small>Fajar Gisborne • Matahari
    ${formatNum(gh.gisborne_dawn_altitude_deg,0)}°</small><
    b>${esc(dawnLocal)}</b><span>${esc(dawnUTC)}</span><str
    ong
    class="${beforeDawn===true?'ok':'warn'}">${esc(dawnStat
    us)}</strong></div></div>`;
    let decision;
    if(gh.qualifies&&gh.first_match){
    const fm=gh.first_match,h=Number(gh.h_day);
    const
    mainAccepted=groups[h].rows.filter(row=>row.pkg_accepte
    d).length;
    const
    extra=validation.filter(row=>Number(row.h_day)===h);
    const
    extraAccepted=extra.filter(row=>row.pkg_accepted).lengt
    h;
    decision=`<span class="global-pkg-badge
    ${gh.decision_pkg==='PKG 2'? 'global-pkg-
    two':''}">${esc(gh.decision_pkg)} TERPENUHI •
    H${h}</span> • H0 (tanggal ijtimak UTC):
    ${esc(fmtDisplayValue(gh.utc_reference_date))}<br><b>${

```

```

esc(gh.decision_reason)}</b><br>Saksi paling awal UTC
dalam sampel ${esc(gh.decision_pkg)}:
<b>${esc(fm.name)}</b> • Maghrib
${esc(fmtTime(fm.sunset_utc||fm.sunset_local,'UTC'))} •
Tinggi geosentris ${formatNum(fm.moon_altitude_deg,3)}°
• Elongasi geosentris
${formatNum(fm.elongation_deg,3)}°<br>Daftar utama:
${mainAccepted}/${esc(gh.total_count)} kota lolos PKG
pada H${h}${extra.length?` • Validasi daratan Amerika:
${extraAccepted}/${extra.length} lokasi lolos PKG`:''}
• Mode global: <b>Geosentris</b>${audit}<div
class="global-hijri-date" role="status" aria-
live="polite">${esc(globalHijriStartLabel(gh))}</div>`;
    }else{
        decision=`<span class="global-status-no">BELUM
DITEMUKAN SAKSI PKG YANG SAH</span> • H0:
${esc(fmtDisplayValue(gh.utc_reference_date))}<br>${esc
(gh.decision_reason)} • Mode global:
Geosentris.${audit}`;
    }
    $('globalHijriSummary').innerHTML=decision;
    function renderRows(items){
        return `<div class="table-
wrap"><table><thead><tr><th>Kota</th><th>Maghrib
Lokal</th><th>Maghrib UTC</th><th>Tinggi Bulan
(G)</th><th>Tinggi Matahari (G)</th><th>Elongasi
(G)</th><th>Iluminasi</th><th>Daratan Utama
Amerika</th><th>Filter 5° + 8°</th><th>Keputusan
PKG</th></tr></thead><tbody>${items.map(row=>`<tr><td>${
esc(row.name)}</td><td>${row.sunset_local?esc(fmtTime(
row.sunset_local,row.timezone_name)): 'Tidak ada
sunset'}</td><td>${row.sunset_utc?esc(fmtTime(row.sunse
t_utc,'UTC'))}: '-
'}</td><td>${formatNum(row.moon_altitude_deg,3)}°</td><
td>${formatNum(row.sun_altitude_deg,3)}°</td><td>${form
atNum(row.elongation_deg,3)}°</td><td>${formatNum(row.i
llumination_pct,3)}%</td><td>${row.america_mainland?'Ya
': 'Bukan'}</td><td>${row.qualifies?'MEMENUHI': '-
'}</td><td>
class="${row.pkg_accepted?'ok':row.qualifies?'warn': 'mu
ted'}">${esc(row.pkg_status|| '-
')}</td></tr>`).join('')}</tbody></table></div>`;
    }
    const
auditDays=(gh.daily_decisions||[]).map(day=>`<div
class="global-pkg-day">H${Number(day.h_day)} •
${esc(fmtDisplayValue(day.scan_date))} • PKG 1:
${Number(day.pkg1_count)} saksi • PKG 2:
${Number(day.pkg2_count)} saksi • Fajar Gisborne:

```

```

    ${esc(day.gisborne_dawn_utc_datetime?fmtTime(day.gisborne_dawn_utc_datetime,'UTC')):'Tidak tersedia'}} •
    ${day.conjunction_before_dawn===true?'Ijtimak sebelum fajar':day.conjunction_before_dawn===false?'Ijtimak tidak sebelum fajar':'Fajar belum tersedia'}}${Number(day.pkg2_postponed_count)?` • PKG 2 ditunda: ${Number(day.pkg2_postponed_count)} lokasi karena ijtimak tidak sebelum fajar`:''}</div>`).join('');

    $('globalScanTable').innerHTML=auditDays+groups.map(group=>{
      const
      date=group.rows.length?group.rows[0].scan_date:'-';
      const
      numeric=group.rows.filter(row=>row.qualifies).length;
      const
      accepted=group.rows.filter(row=>row.pkg_accepted).length;
      const
      extra=validation.filter(row=>Number(row.h_day)===group.h);
      return `<details
    ${gh.qualifies&&Number(gh.h_day)===group.h?'open':''}><summary style="cursor:pointer;margin-top:8px;font-weight:800">H${group.h} • ${esc(fmtDisplayValue(date))} • Daftar utama ${numeric}/${group.rows.length} memenuhi 5° + 8°; ${accepted} lolos PKG</summary>${renderRows(group.rows)}${extra.length?`<details class="global-pkg-extra" open><summary>Validasi tambahan daratan Amerika • ${extra.length} lokasi • ${extra.filter(row=>row.pkg_accepted).length} lolos PKG</summary>${renderRows(extra)}</details>`:''}</details>`;
    }).join('')+`<p class="global-pkg-note">G = geosentris. Tabel utama 100 kota dipertahankan; lokasi validasi tambahan ditampilkan terpisah. PKG 1 diperiksa lebih dahulu. PKG 2 membutuhkan daratan utama Amerika dan ijtimak sebelum fajar Gisborne pada tanggal awal bulan yang diuji. Scan ini berbasis sampel lokasi, bukan pemetaan seluruh titik daratan.</p>`;
  }
  function renderEclipses(report){
    if(!report){$('pane-eclipse').innerHTML='<div class="card"><div class="muted">Data gerhana belum tersedia.</div></div>';return}
    const
    monthNames=['Januari','Februari','Maret','April','Mei',

```

```

'Juni','Juli','Agustus','September','Oktober','November',
', 'Desember'];
const
mname=monthNames[Math.max(0,Math.min(11,Number(report.s
elected_month||1)-1))];
const rowsFor=events=>(events||[]).map(x=>[
  x.object||'-',

fmtTime(x.local_datetime,report.timezone_name||$('tz').
value),
  fmtTime(x.utc_datetime,'UTC'),
  x.type||'-',
  x.region||'-
]);
const monthRows=rowsFor(report.month_events);
const yearRows=rowsFor(report.year_events);
const monthHtml=monthRows.length
  ? table(['Objek','Tanggal & Waktu Lokal','Waktu
UTC','Jenis Gerhana','Wilayah Terdampak /
Visibilitas'],monthRows)
  : `

SOLUNA | 201


```

```

textContent=formatNum(r.moon.elongation_deg,3)+'°';$('m
Illum').textContent=formatNum(r.moon.illumination_pct,3
)+'%';$('timeCard').innerHTML=`Lokal:
<b>${esc(fmtTime(r.local_datetime,r.timezone_name))}</b>
<br>UTC:
${esc(fmtTime(r.utc_datetime,'UTC'))}<br>Mode:
${esc(r.time_mode)} •
${esc(r.mode)}`;$('locCard').innerHTML=`<b>${esc(r.loca
tion_name)}</b><br>${formatNum(r.latitude,6)},
${formatNum(r.longitude,6)} •
${formatNum(r.elevation_m??$('elev').value,1)}
m<br>${esc(r.timezone_name)}`;let
ok=r.moon.altitude_deg>=5&r.moon.elongation_deg>=8;$('
hilalCard').innerHTML=`<b
class="${ok?'ok':'warn'}">${ok?'MEMENUHI FILTER':'BELUM
MEMENUHI'}</b><br>Tinggi
${formatNum(r.moon.altitude_deg,3)}° • Elongasi
${formatNum(r.moon.elongation_deg,3)}°<br><span
class="muted">Filter numerik, bukan keputusan
visibilitas formal.</span>`;
renderGlobalHijri(r.global_hijri);
$('pane-
summary').innerHTML=table(['Parameter','Matahari','Bula
n','Unit / Catatan'],summaryRows(r));$('pane-
sun').innerHTML=table(['Parameter','Nilai','Unit /
Catatan'],bodyRows(r.sun,r.mode));$('pane-
moon').innerHTML=table(['Parameter','Nilai','Unit /
Catatan'],bodyRows(r.moon,r.mode));$('pane-
events').innerHTML=table(['Peristiwa','Nilai /
Waktu','Catatan'],eventRows(r));$('pane-
phenomena').innerHTML=table(['Kelompok','Fenomena','Nil
ai /
Waktu','Keterangan'],(r.phenomena||[]).map(x=>[x.catego
ry,x.name,x.value,x.description]));$('pane-
schedule').innerHTML=table(['Jadwal','Waktu
Lokal','Dasar
Astronomi','Keterangan'],(r.schedule||[]).map(x=>[x.nam
e,x.value,x.basis,x.note]));renderEclipses(r.eclipses);
setStatus(`Selesai • ${r.engine} • ${r.location_name} •
${r.mode}`, 'ok'))
async function
calculate(){setSidebarHidden(true);if(!pyCalc){toast('E
ngine belum
siap.')}return}try{($('loading').classList.add('calculat
ing');setLoad(true,'Menghitung astronomi','Skyfield
memakai satu BSP sesuai tahun pilihan, mencari
konjungsi bulan terpilih, lalu memindai 100 kota pada
H0, H1, dan H2...',55);await new
Promise(resolve=>requestAnimationFrame(()=>requestAnima

```

```

tionFrame(resolve)));let
city=selectedCity(),tz=$('#tz').value.trim()||'Asia/Jakarta';let
payload={location_name:city?labelCity(city):'Lokasi Manual',lat:+$('#lat').value,lon:+$('#lon').value,elev:+$('#elev').value,timezone_name:tz,date:$('#date').value,time:$('#time').value||'12:00:00',tz_offset_min:timezoneOffsetMinutes($('#date').value,$('time').value||'12:00:00',tz),time_mode:timeMode,mode:coordMode};if(!payload.date||!Number.isFinite(payload.lat)||!Number.isFinite(payload.lon))throw new Error('Tanggal, latitude, dan longitude harus valid.');
```

```

let out=await
withSelectedEphemeris(payload.date,async()=>{let
value=pyCalc(JSON.stringify(payload));if(value&&typeof
value.then=== 'function')value=await value;return
value});let
r=JSON.parse(String(out));renderResult(r);$('#loading').classList.remove('calculating');setLoad(false)}catch(e){console.error(e);$('#loading').classList.remove('calculating');setLoad(false);setStatus('Error perhitungan: '+e.message,'err');toast('Perhitungan gagal: '+e.message)}}

```

const SOLUNA_METHOD_TEXT="METODOLOGI PERHITUNGAN\n\nA. Sumber Data dan Sistem Referensi\n\nEphemeris menggunakan JPL DE440s melalui Skyfield. Pada versi web statis, browser mendahulukan satu BSP tahunan: de440s_2026.bsp sampai de440s_2030.bsp. Tahun lain memakai satu BSP rentang yang mencakup tahun pilihan. Jika BSP tahunan tidak ditemukan (HTTP 404/410), hanya BSP rentang yang sesuai yang dipakai. BSP asli de440s.bsp tidak dibuka. Setiap BSP tahunan memiliki data penyangga untuk mempertahankan pencarian fenomena hingga lima tahun mendatang; seluruh scan 100 kota memakai kernel aktif yang sama.\n\nB. Kerangka Koordinat\n\nMode Toposentris menggunakan posisi pengamat WGS84 dan memasukkan paralaks lokal. Mode Geosentris memakai pusat Bumi untuk RA/Dec apparent dan memproyeksikannya ke horizon lokasi.\n\nC. Matahari, Bulan, Fase & Hilal\n\nElongasi adalah separasi apparent Bulan-Matahari. Iluminasi dihitung dari geometri fase. Filter hilal lokal memakai tinggi Bulan $\geq 5^\circ$ dan elongasi $\geq 8^\circ$ pada Maghrib lokal. Penentuan 1 Hijriah global selalu memakai parameter geosentris, terlepas dari mode hasil lokal. H0 adalah tanggal UTC ijtimak. PKG 1 memerlukan $5^\circ + 8^\circ$ sebelum 00.00 UTC tanggal berikutnya; PKG 2 memerlukan $5^\circ + 8^\circ$ di daratan utama Amerika pada/setelah batas tersebut serta ijtimak sebelum fajar Gisborne pada tanggal calon awal bulan.

Fajar dihitung saat pusat Matahari toposentris naik melewati -18° , zona Pacific/Auckland. Bila PKG 2 gagal, hari berikutnya diuji kembali. Daftar utama 100 kota dipertahankan; sampel verifikasi daratan Amerika ditambahkan jika PKG 1 belum ditemukan. Scan berbasis sampel lokasi, bukan pemetaan seluruh daratan.\n\nD. Peristiwa\n\nSunrise/sunset, moonrise/moonset, transit serta twilight dihitung dengan almanak Skyfield. Maghrib memakai rutin sunset Skyfield.\n\nE. Jadwal\n\nShubuh ditampilkan untuk -20° dan -18° , Dhuha $+4,5^\circ$, Duhur transit Matahari, Ashar faktor bayangan 1, Maghrib sunset, Isya -18° , Midnight, dan awal 1/3 malam terakhir. Tidak ada ihtiyat otomatis.\n\nCATATAN BATASAN MODEL\n\nCuaca, relief/horizon nyata, ekstingsi atmosfer lokal, tekanan dan suhu aktual, serta obstruksi bangunan tidak dimodelkan secara rinci.";

const SOLUNA_TERMS_TEXT="ISTILAH DAN SINGKATAN PENTING\n\nA. SISTEM WAKTU DAN KOORDINAT\n\nUTC – Coordinated Universal Time (Waktu Universal Terkoordinasi). Acuan waktu internasional.\n\nTT – Terrestrial Time (Waktu Terrestrial). Skala waktu astronomi seragam yang dipakai antara lain untuk definisi epoch J2000.0.\n\nWIB – Waktu Indonesia Barat, UTC+7.\n\nWITA – Waktu Indonesia Tengah, UTC+8.\n\nWIT – Waktu Indonesia Timur, UTC+9.\n\nUSA – United States of America (Amerika Serikat). Singkatan negara yang dapat muncul pada nama lokasi Amerika.\n\nYYYY – Year empat digit pada format tanggal; contoh 2026. MM – Month/bulan dua digit. DD – Day/tanggal dua digit.\n\nHH – Hour/jam dua digit. MM pada format waktu – Minute/menit dua digit. SS – Second/detik dua digit. Jadi YYYY-MM-DD = tahun-bulan-tanggal dan HH:MM:SS = jam:menit:detik.\n\nAl. SINGKATAN ZONA WAKTU INTERNASIONAL\n\nAKST – Alaska Standard Time. Waktu standar Alaska, UTC-9.\n\nAKDT – Alaska Daylight Time. Waktu musim panas Alaska, UTC-8.\n\nHST – Hawaii Standard Time / Hawaii-Aleutian Standard Time. Umumnya UTC-10; juga dipakai di Kepulauan Aleutian bagian barat pada waktu standar.\n\nHDT – Hawaii-Aleutian Daylight Time. Waktu musim panas Kepulauan Aleutian yang menerapkan DST, UTC-9.\n\nPST – Pacific Standard Time. Waktu standar zona Pasifik Amerika Utara, UTC-8.\n\nPDT – Pacific Daylight Time. Waktu musim panas zona Pasifik Amerika Utara, UTC-7.\n\nMST – Mountain Standard Time. Waktu standar zona Pegunungan Amerika Utara, UTC-7.\n\nMDT – Mountain Daylight Time. Waktu musim panas zona Pegunungan Amerika Utara, UTC-6.\n\nCST – Central Standard Time. Umumnya waktu standar zona Tengah Amerika Utara, UTC-6; singkatan ini dapat memiliki arti

lain di wilayah berbeda.\nCDT – Central Daylight Time. Waktu musim panas zona Tengah Amerika Utara, UTC-5.\nEST – Eastern Standard Time. Waktu standar zona Timur Amerika Utara, UTC-5.\nEDT – Eastern Daylight Time. Waktu musim panas zona Timur Amerika Utara, UTC-4.\nAST – Atlantic Standard Time. Waktu standar Atlantik di sebagian Kanada/Karibia, UTC-4. Singkatan AST pada sumber lain kadang juga berarti Arabia Standard Time; perhatikan wilayah/zona IANA yang dipilih.\nADT – Atlantic Daylight Time. Waktu musim panas Atlantik, UTC-3.\nNST – Newfoundland Standard Time. Waktu standar Newfoundland, UTC-3:30.\nNDT – Newfoundland Daylight Time. Waktu musim panas Newfoundland, UTC-2:30.\nGMT – Greenwich Mean Time. Waktu rata-rata Greenwich, UTC±0.\nBST – British Summer Time. Waktu musim panas Britania Raya, UTC+1; singkatan BST dapat memiliki arti lain di negara berbeda.\nWET – Western European Time. Waktu Eropa Barat, UTC±0.\nWEST – Western European Summer Time. Waktu musim panas Eropa Barat, UTC+1.\nCET – Central European Time. Waktu Eropa Tengah, UTC+1.\nCEST – Central European Summer Time. Waktu musim panas Eropa Tengah, UTC+2.\nEET – Eastern European Time. Waktu Eropa Timur, UTC+2.\nEEST – Eastern European Summer Time. Waktu musim panas Eropa Timur, UTC+3. Mesir dapat menampilkan EET/EEST sesuai tanggal.\nTRT – Turkey Time. Waktu Türkiye, UTC+3 ketika sistem menampilkan singkatan TRT; sebagian sistem IANA dapat menampilkan +03.\nMSK – Moscow Standard Time. Waktu Moskwa, UTC+3.\nIST – India Standard Time, UTC+5:30, jika zona yang dipilih adalah India. Singkatan IST juga dapat berarti Israel Standard Time atau Irish Standard Time; lihat lokasi/zona IANA.\nIDT – Israel Daylight Time. Waktu musim panas Israel, UTC+3.\nIRST – Iran Standard Time. Waktu standar Iran, UTC+3:30.\nIRDT – Iran Daylight Time. Istilah historis waktu musim panas Iran, UTC+4:30 pada periode ketika DST berlaku.\nGST – Gulf Standard Time. Waktu Teluk untuk Uni Emirat Arab dan Oman, UTC+4.\nPKT – Pakistan Standard Time. Waktu Pakistan, UTC+5.\nNPT – Nepal Time. Waktu Nepal, UTC+5:45.\nBDT – Bangladesh Standard Time. Waktu Bangladesh, UTC+6.\nMMT – Myanmar Time. Waktu Myanmar, UTC+6:30.\nICT – Indochina Time. Waktu Indochina, UTC+7.\nHKT – Hong Kong Time. Waktu Hong Kong, UTC+8.\nSGT – Singapore Time. Waktu Singapura, UTC+8.\nMYT – Malaysia Time. Waktu Malaysia, UTC+8.\nPHT – Philippine Time. Waktu Filipina, UTC+8.\nAWST – Australian Western Standard Time. Waktu standar Australia Barat, UTC+8.\nJST – Japan Standard Time. Waktu standar Jepang, UTC+9.\nKST – Korea

Standard Time. Waktu standar Korea Selatan, UTC+9.\nChST – Chamorro Standard Time. Waktu Guam dan Kepulauan Mariana Utara, UTC+10.\nAEST – Australian Eastern Standard Time. Waktu standar Australia bagian timur, UTC+10.\nAEDT – Australian Eastern Daylight Time. Waktu musim panas Australia bagian timur, UTC+11.\nACST – Australian Central Standard Time. Waktu standar Australia bagian tengah, UTC+9:30.\nACDT – Australian Central Daylight Time. Waktu musim panas Australia bagian tengah, UTC+10:30.\nLHST – Lord Howe Standard Time. Waktu standar Lord Howe Island, UTC+10:30.\nLHDT – Lord Howe Daylight Time. Waktu musim panas Lord Howe Island, UTC+11.\nNZST – New Zealand Standard Time. Waktu standar Selandia Baru, UTC+12.\nNZDT – New Zealand Daylight Time. Waktu musim panas Selandia Baru, UTC+13.\nSAST – South Africa Standard Time. Waktu Afrika Selatan, UTC+2.\nEAT – East Africa Time. Waktu Afrika Timur, UTC+3.\nCAT – Central Africa Time. Waktu Afrika Tengah, UTC+2.\nWAT – West Africa Time. Waktu Afrika Barat, UTC+1.\nART – Argentina Time. Waktu Argentina, UTC-3.\nBRT – Brasília Time. Waktu Brasília/Brasil timur, UTC-3.\nBRST – Brasília Summer Time. Istilah historis waktu musim panas Brasília, UTC-2 ketika DST berlaku.\nCLT – Chile Standard Time. Waktu standar Chile, umumnya UTC-4.\nCLST – Chile Summer Time. Waktu musim panas Chile, umumnya UTC-3.\nCOT – Colombia Time. Waktu Kolombia, UTC-5.\nPET – Peru Time. Waktu Peru, UTC-5.\nVET – Venezuela Time. Waktu Venezuela, UTC-4.\nBOT – Bolivia Time. Waktu Bolivia, UTC-4.\nUYT – Uruguay Standard Time. Waktu Uruguay, UTC-3.\nPYT – Paraguay Time. Singkatan waktu Paraguay; offset dapat mengikuti aturan resmi yang berlaku pada tanggal terkait.\nAZOT – Azores Standard Time. Waktu standar Azores, UTC-1.\nAZOST – Azores Summer Time. Waktu musim panas Azores, UTC±0.\nCVT – Cape Verde Time. Waktu Cabo Verde, UTC-1.\nCatatan zona waktu – Singkatan zona waktu dapat ambigu antarnegara. Aplikasi juga menyimpan nama zona IANA (contoh: America/Anchorage, Pacific/Auckland) sebagai identitas zona yang lebih pasti. Offset UTC mengikuti tanggal karena DST dapat berubah.\nLST – Local Sidereal Time (Waktu Sideris Lokal). Menyatakan orientasi lokal terhadap bola langit dan digunakan untuk menghitung hour angle.\nGAST – Greenwich Apparent Sidereal Time. Waktu sideris tampak di meridian Greenwich yang sudah memperhitungkan nutasi.\nRA – Right Ascension (Asensio Rekta). Koordinat ekuatorial benda langit yang analog dengan bujur; biasanya dinyatakan dalam jam, menit,

detik.\nDec – Declination (Deklinasi). Koordinat ekuatorial utara/selatan dari ekuator langit, dinyatakan dalam derajat.\nHA – Hour Angle (Sudut Jam). Selisih antara LST dan RA; menunjukkan posisi benda terhadap meridian lokal.\nWGS84 – World Geodetic System 1984. Model referensi bentuk dan koordinat Bumi yang digunakan untuk posisi pengamat.\nToposentris – Posisi benda langit sebagaimana dilihat dari lokasi pengamat tertentu di permukaan Bumi. Memasukkan efek paralaks lokal.\nGeosentris – Posisi benda langit sebagaimana dilihat dari pusat Bumi.\nApparent – Posisi tampak setelah koreksi astronomi seperti aberasi/nutasi yang relevan dalam model ephemeris.\nEpoch of Date / Equinox of Date – Sistem koordinat yang dirujuk ke tanggal pengamatan, bukan epoch tetap seperti J2000.\n\nB. EPHEMERIS DAN SATUAN\n\nJPL – Jet Propulsion Laboratory, lembaga riset NASA yang mengembangkan ephemeris numerik presisi tinggi tata surya.\nNASA – National Aeronautics and Space Administration. Badan antariksa sipil Amerika Serikat.\nIAU – International Astronomical Union. Organisasi astronomi internasional yang menetapkan standar, nomenklatur, dan batas resmi konstelasi.\nSPICE – Spacecraft, Planet, Instrument, C-matrix, Events. Sistem toolkit/kernel navigasi dan geometri antariksa yang dikembangkan NASA/JPL.\nJ2000 – Julian Date 2000.0 / epoch J2000.0. Epoch referensi astronomi yang berpusat sekitar 1 Januari 2000 pukul 12:00 TT.\nDE440s – Development Ephemeris 440 short. Berkas ephemeris JPL yang digunakan aplikasi untuk posisi Matahari, Bulan, dan Bumi.\nSPK – Spacecraft and Planet Kernel. Format kernel ephemeris yang digunakan dalam keluarga SPICE/JPL.\nBSP – Binary SPK / Binary Spacecraft and Planet Kernel. Ekstensi berkas kernel SPK biner seperti de440s.bsp.\nAU – Astronomical Unit (Satuan Astronomi), jarak rata-rata Bumi-Matahari, sekitar 149,6 juta km.\nkm – kilometer.\n° – derajat; ' – menit busur; " – detik busur.\narcmin – arcminute atau menit busur; 1° = 60 arcmin.\nh m s – jam, menit, detik; lazim digunakan untuk RA dan waktu.\n\nC. POSISI MATAHARI DAN BULAN\n\nAltitude / Tinggi – Sudut benda langit di atas atau di bawah ufuk. Nilai positif berarti di atas ufuk.\nAltitude geometrik – Tinggi yang dihitung murni dari geometri tanpa koreksi refraksi atmosfer.\nAltitude refraksi – Perkiraan tinggi tampak setelah koreksi pembiasan atmosfer standar.\nAzimut – Arah benda pada horizon, umumnya dihitung dari utara melalui timur, 0°-360°.\nEkliptika – Lingkaran besar jalur semu tahunan Matahari pada bola langit; menjadi acuan longitude/latitude ekliptika.\nLongitude

ekliptika – Posisi sudut sepanjang ekliptika.
 ekliptika – Jarak sudut benda ke utara/selatan bidang ekliptika.
 konstelasi IAU – Rasi resmi menurut pembagian International Astronomical Union tempat posisi benda berada.
 Jarak geosentris – Jarak pusat Bumi ke benda langit.
 Jarak pengamat – Jarak dari lokasi pengamat topocentris ke benda langit.
 Diameter sudut – Besar tampak seluruh piringan benda langit di langit.
 Semidiameter – Setengah diameter sudut; penting untuk peristiwa horizon dan kontak piringan.
 Paralaks horizontal – Pergeseran sudut maksimum antara posisi geosentris dan topocentris ketika benda berada dekat horizon.
 Magnitudo / V-band – $V = \text{Visual}$. Ukuran kecerlangan tampak dalam pita fotometri visual; semakin negatif berarti semakin terang.
 D. FASE DAN GEOMETRI BULAN
 Elongasi – Separasi sudut tampak antara Bulan dan Matahari.
 Iluminasi – Persentase piringan Bulan yang tampak diterangi dari arah pengamat.
 Sudut fase – Sudut Matahari-Bulan-pengamat; berkaitan langsung dengan bagian Bulan yang diterangi.
 Fase 0–360° – Sudut siklus fase Bulan: 0° Bulan Baru, 90° Perbani Awal, 180° Purnama, 270° Perbani Akhir.
 Waxing – Fase membesar; bagian terang Bulan bertambah menuju purnama.
 Waning – Fase mengecil; bagian terang Bulan berkurang setelah purnama.
 Bright Limb Position Angle – Sudut posisi sisi terang Bulan, yaitu orientasi garis menuju sisi piringan yang paling terang.
 Hilal – Sabit Bulan muda setelah konjungsi yang menjadi perhatian dalam observasi awal bulan Hijriah.
 Konjungsi / New Moon – Saat Matahari dan Bulan memiliki longitude ekliptika geosentris yang sama atau fase astronomis Bulan Baru.
 Konjungsi geosentris – Konjungsi yang dihitung dari pusat Bumi.
 Konjungsi topocentris – Minimum separasi tampak Matahari-Bulan dari lokasi pengamat tertentu.
 Age Bulan – Selang waktu sejak konjungsi geosentris/Bulan Baru terakhir sampai waktu perhitungan.
 Lag Time – Selisih waktu antara moonset dan sunset. Nilai positif berarti Bulan terbenam setelah Matahari.
 H0 – Maghrib pada tanggal lokal terjadinya konjungsi geosentris (New Moon). Ini adalah titik awal penomoran hari hilal pada tabel bulanan.
 H1 – Maghrib satu hari kalender setelah H0; H2 dua hari setelah H0; H3 tiga hari setelah H0; dan seterusnya.
 Kandidat Hilal Bulanan – Pemindaian seluruh Maghrib pada bulan terpilih. Filter aplikasi: tinggi hilal $\geq 5^\circ$ dan elongasi $\geq 8^\circ$. Filter ini bukan keputusan visibilitas formal.
 E. PERISTIWA HARIAN
 Rise / Terbit – Saat benda langit naik

melewati batas horizon menurut model peristiwa yang digunakan.

- \nSet / Terbenam – Saat benda langit turun melewati batas horizon.
- \nTransit meridian – Saat benda melintasi meridian lokal dan biasanya mencapai tinggi harian maksimum.
- \nSunrise – Matahari terbit.
- \nSunset / Maghrib lokal – Matahari terbenam di lokasi pengamat; aplikasi menggunakan sunset astronomis yang dihitung untuk lokasi/tanggal.
- \nMoonrise – Bulan terbit.
- \nMoonset – Bulan terbenam.
- \nTwilight / Syafak-Fajar – Masa transisi cahaya ketika Matahari berada di bawah ufuk.
- \nCivil Twilight – Senja/fajar sipil, batas umum Matahari sekitar 6° di bawah ufuk.
- \nNautical Twilight – Senja/fajar nautika, sekitar 12° di bawah ufuk.
- \nAstronomical Twilight – Senja/fajar astronomis, sekitar 18° di bawah ufuk.
- \nDay Length / Panjang siang – Selang waktu dari sunrise hingga sunset.
- \nEoT – Equation of Time (Persamaan Waktu). Selisih antara waktu Matahari semu dan waktu Matahari rata-rata.

\n\nF. JADWAL MATAHARI DAN WAKTU SHALAT

- \n\nShubuh / Fajr – Awal fajar; pada tab Jadwal aplikasi menampilkan dua kriteria tinggi pusat Matahari: -20° dan -18° .
- \nSunrise – Waktu Matahari terbit pada horizon lokal.
- \nDhuha – Pada tab Jadwal dihitung ketika tinggi pusat Matahari mencapai $+4,5^\circ$ setelah terbit.
- \nDhuhur / Zuhur – Waktu Matahari transit atau berkulminasi pada meridian lokal.
- \nAshar / Asar – Pada tab Jadwal menggunakan faktor bayangan 1: panjang bayangan tambahan sama dengan tinggi benda.
- \nMaghrib – Waktu Matahari terbenam pada horizon lokal.
- \nIsya – Pada tab Jadwal dihitung ketika pusat Matahari mencapai -18° di ufuk barat.
- \nMidnight – Titik tengah interval Maghrib sampai Shubuh -18° hari berikutnya; tidak selalu sama dengan pukul 00:00.
- \n1/3 Malam – Awal sepertiga malam terakhir, yaitu setelah 2/3 interval Maghrib sampai Shubuh -18° berlalu.

\n\nG. FENOMENA ORBITAL

- \n\nPerigee – Titik/jangka waktu ketika Bulan berada pada jarak minimum lokal dari Bumi.
- \nApogee – Titik/jangka waktu ketika Bulan berada pada jarak maksimum lokal dari Bumi.
- \nPerihelion – Saat Bumi berada pada jarak minimum lokal dari Matahari.
- \nAphelion – Saat Bumi berada pada jarak maksimum lokal dari Matahari.
- \nAscending Node / Node Naik – Saat Bulan melintasi ekliptika dari selatan ke utara.
- \nDescending Node / Node Turun – Saat Bulan melintasi ekliptika dari utara ke selatan.
- \nEkuinoks – Saat Matahari melintasi ekuator langit; terjadi sekitar Maret dan September.
- \nSolstis – Saat deklinasi Matahari mencapai ekstrem tahunan; terjadi sekitar Juni dan Desember.
- \nGerhana Bulan – Bulan memasuki bayangan Bumi saat konfigurasi dekat purnama dan node orbit.
- \nGerhana

Matahari – Bulan menutupi sebagian/seluruh piringan Matahari bagi pengamat di wilayah tertentu saat konfigurasi dekat Bulan Baru dan node orbit.

Kandidat gerhana Matahari global – Indikator geometrik awal di aplikasi; bukan klasifikasi resmi jalur atau jenis gerhana.

H. FORMAT DATA, ANTARMUKA, DAN PERANGKAT LUNAK

API – Application Programming Interface. Antarmuka yang memungkinkan perangkat lunak mengakses fungsi/data dari sistem lain.

UI – User Interface (Antarmuka Pengguna). Bagian visual yang digunakan pengguna untuk berinteraksi dengan aplikasi.

GUI – Graphical User Interface (Antarmuka Pengguna Grafis). UI berbasis jendela, tombol, tabel, dan kontrol visual.

EXE – Executable. Berkas program Windows yang dapat dijalankan langsung.

ID – Identifier / Identification. Pengenal unik suatu item, record, lokasi, atau objek.

DST – Daylight Saving Time. Penyesuaian musiman jam sipil; menyebabkan singkatan seperti AKST berubah menjadi AKDT, atau NZST menjadi NZDT.

IANA – Internet Assigned Numbers Authority. Pengelola basis data nama zona waktu IANA yang dipakai sistem seperti America/Anchorage atau Asia/Jakarta.

CSV – Comma-Separated Values. Format teks tabel yang digunakan untuk ekspor hasil.

JSON – JavaScript Object Notation. Format data terstruktur yang digunakan untuk data lokasi/pengaturan internal.

Cache – Penyimpanan lokal permanen agar resource seperti DE440s tidak perlu disiapkan ulang setiap aplikasi dijalankan.

Catatan: istilah pada tab ini menjelaskan arti parameter aplikasi. Untuk penggunaan ilmiah/publikasi, tetap perhatikan definisi metode, frame koordinat, model atmosfer, horizon, dan kriteria rukyat yang digunakan.

```
function applyTimeModeState(){let
mag=timeMode==='Maghrib
Lokal';$('time').disabled=mag;$('time').classList.toggle('time-
disabled',mag);if(!mag)$('timeModeNote').textContent='M
aghrib Lokal = waktu sunset pada kota dan tanggal
terpilih.'}
async function refreshMaghribTime(reason='') {
  const revision=++selectionRevision;
  if(timeMode!=='Maghrib
Lokal')return;applyTimeModeState();if(!engineReady||!py
Sunset){$('timeModeNote').textContent='Menunggu BSP
tahun berjalan/engine untuk menghitung Maghrib
Lokal...';return}
  let date=($('date').value;if(!date)return;let
tz=($('tz').value.trim()||'Asia/Jakarta';let
```

```

city=selectedCity();let
payload={location_name:city?labelCity(city):'Lokasi
Manual',lat:+$('lat').value,lon:+$('lon').value,elev:+$
('elev').value,timezone_name:tz,date};

if(!Number.isFinite(payload.lat)||!Number.isFinite(payload.lon))return;
const
isCurrent={()=>revision===selectionRevision&&timeMode===
'Maghrib Lokal'&&date===$('date').value;

try{$('time').value='';$('timeModeNote').textContent=`M
enghitung Maghrib Lokal${reason?' • '+reason:''}`;let
out=await
withSelectedEphemeris(payload.date,async()=>{let
value=pySunset(JSON.stringify(payload));if(value&&typeof
value.then==='function')value=await value;return
value},isCurrent);if(out===null||!isCurrent())return;let
d=JSON.parse(String(out));$('time').value=d.clock;$('ti
meModeNote').textContent=`Maghrib Lokal
${city?labelCity(city):'Lokasi Manual'}:
${fmtDisplayValue(d.display)}`;setStatus(`Maghrib Lokal
diperbarui: ${city?labelCity(city):'Lokasi Manual'} •
${fmtDisplayValue(d.display)}`, 'ok')}catch(e){if(!isCur
rent())return;console.error(e);setLoad(false);$('time')
.value='';$('timeModeNote').textContent='Maghrib Lokal
tidak ditemukan pada tanggal/lokasi
ini.';setStatus('Maghrib Lokal gagal dihitung:
'+e.message,'err')}
}
function buildDocs(){
  $('pane-method').innerHTML=`<div class="card doc"><pre
class="terms-
pre">${esc(SOLUNA_METHOD_TEXT)}</pre></div>`;
  $('pane-terms').innerHTML=`<div class="card doc"><pre
class="terms-
pre">${esc(SOLUNA_TERMS_TEXT)}</pre></div>`;
}
function downloadCSV(){if(!lastResult){toast('Hitung
data terlebih dahulu.');
```

```

L(new Blob(['\u0001'+csv], {type: 'text/csv; charset=utf-8'})); a.download = 'hasil_SOLUNA_astronomi.csv'; a.click();
URL.revokeObjectURL(a.href)}
async function
copyResult() {if(!lastResult) {toast('Hitung data
terlebih dahulu.')}; return} let t = ['SOLUNA - Astronomi
Matahari & Bulan', `Lokasi:
${lastResult.location_name}`, `UTC:
${fmtTime(lastResult.utc_datetime, 'UTC')}`, `Mode:
${lastResult.mode}`, `', 'Parameter\tMatahari\tBulan\tUni
t'].concat(summaryRows(lastResult).map(r=>r.map(fmtDisp
layValue).join('\t'))).join('\n'); await
navigator.clipboard.writeText(t); toast('Hasil disalin
ke clipboard.')}
function setSidebarHidden(hidden) {
  const app=document.querySelector('.app');
  if(!app) return;
  app.classList.toggle('sidebar-
hidden', Boolean(hidden));
  const btn=$('#menuBtn');
  if(btn) {
    btn.setAttribute('aria-
expanded', hidden?'false':'true');
    btn.title=hidden?'Tampilkan sidebar':'Sembunyikan
sidebar';
  }
}
function toggleSidebar() {
  const app=document.querySelector('.app');
  if(!app) return;
  setSidebarHidden(!app.classList.contains('sidebar-
hidden'));
}
function initUI() {
  const
tabDefs=[['summary', 'Ringkasan'], ['sun', 'Matahari'], ['m
oon', 'Bulan'], ['events', 'Peristiwa'], ['phenomena', 'Feno
mena'], ['schedule', 'Jadwal'], ['eclipse', 'Gerhana'], ['me
thod', 'Metodologi'], ['terms', 'Istilah']]; $('tabs').inne
rHTML=tabDefs.map((x,i)=>`<button class="tabbtn
${i===0?'active':''}" data-
tab="${x[0]}">${x[1]}</button>`).join(''); document.quer
ySelectorAll('.tabbtn').forEach(b=>b.onclick=()=>{docum
ent.querySelectorAll('.tabbtn').forEach(x=>x.classList.
remove('active')); document.querySelectorAll('.tabpane')
.forEach(x=>x.classList.remove('active')); b.classList.a
dd('active'); $('pane-
'+b.dataset.tab).classList.add('active')});

```

```

function
wireSeg(id,setter){document.querySelectorAll('#'+id+'
button').forEach(b=>b.onclick=()=>{document.querySelect
orAll('#'+id+'
button').forEach(x=>x.classList.remove('active'));b.cla
ssList.add('active');setter(b.dataset.v)}}}wireSeg('coo
rdMode',v=>coordMode=v);wireSeg('timeMode',v=>{timeMode
=v;applyTimeModeState();if(v==='Maghrib
Lokal')refreshMaghribTime('mode Maghrib Lokal
dipilih');else setStatus('Mode Jam Manual
aktif.','ok')));
renderCityOptions();let
sab=CITIES.find(c=>c.name==='Sabang')||CITIES[0];$('cit
ySelect').value=labelCity(sab);applyCity(sab);$('citySe
arch').oninput=e=>renderCityOptions(e.target.value);$('
citySelect').onchange=()=>{let
c=selectedCity();if(c)applyCity(c)};$('date').onchange=
()=>handleSelectedDateChange('perubahan
tanggal');['lat','lon','elev','tz'].forEach(id=>$(id).a
ddEventListener('change',()=>{if(timeMode==='Maghrib
Lokal')refreshMaghribTime('perubahan
lokasi')});$('nowBtn').onclick=setNow;$('calcBtn').onc
lick=calculate;$('copyBtn').onclick=copyResult;$('csvBt
n').onclick=downloadCSV;$('menuBtn').onclick=toggleSide
bar;setSidebarHidden(false);applyTimeModeState();buildD
ocs();mergeIndonesiaData();initEngine();
}
window.addEventListener('DOMContentLoaded',initUI);
</script>
<script id="soluna-education-navigation">
// Navigasi informasi saja; seluruh engine dan handler
perhitungan asli dipertahankan.
window.addEventListener('DOMContentLoaded',function
initSolunaEducation(){
const
education=document.getElementById('solunaEducationView'
);
const
calculation=document.getElementById('solunaCalculationV
iew');
const scroller=document.querySelector('.soluna-
education-shell');
const
pages=Array.from(education.querySelectorAll('[data-
soluna-page]'));
const
links=Array.from(document.querySelectorAll('.soluna-
edu-nav [data-soluna-open]'));

```

```

function
showPage(name, {focus=false, closeSidebar=false}={}) {

if(name!=='calculator'&&!pages.some(page=>page.dataset.
solunaPage===name)) return;
  const isCalculation=name==='calculator';
  education.hidden=isCalculation;
  calculation.hidden=!isCalculation;

pages.forEach(page=>page.hidden=page.dataset.solunaPage
!==name);
  links.forEach(link=>{

if(link.dataset.solunaOpen===name) link.setAttribute('ar
ia-current','page');
    else link.removeAttribute('aria-current');
  });
  scroller.scrollTop=0;
  if(closeSidebar&&window.matchMedia('(max-
width:1100px)').matches) setSidebarHidden(true);
  if(focus&&!isCalculation){
    const
heading=pages.find(page=>page.dataset.solunaPage===name
).querySelector('h2');
    heading.focus({preventScroll:true});
  }
  document.querySelectorAll('[data-soluna-
open]').forEach(button=>button.addEventListener('click'
, ()=>{

showPage(button.dataset.solunaOpen, {focus:true, closeSid
ebar:true});
  }));
  education.querySelectorAll('.soluna-edu-toc
a').forEach(link=>link.addEventListener('click', event=>
{
    const
target=document.getElementById(link.getAttribute('href'
).slice(1));
    if(!target) return;
    event.preventDefault();
    const top=target.getBoundingClientRect().top-
scroller.getBoundingClientRect().top+scroller.scrollTop
-88;

scroller.scrollTo({top, behavior:window.matchMedia('(pre
fers-reduced-
motion:reduce)').matches?'auto':'smooth'});

```

```
    ));  
    // Capture hanya menampilkan panel lama sebelum  
    onclick=calculate berjalan.  
  
    document.getElementById('calcBtn').addEventListener('click', ()=>showPage('calculator',{capture:true});  
        showPage('beranda');  
    ));  
</script>  
</body></html>
```

GLOSARIUM

A

Aphelion. Maksimum jarak Bumi–Matahari dalam siklus yang diperiksa.

Apogee. Maksimum jarak Bulan terhadap Bumi dalam siklus yang diperiksa.

Apparent. Posisi setelah koreksi astronomis seperti aberasi dan pembelokan cahaya yang relevan; berbeda dari refraksi atmosfer.

Asensio rekta. Koordinat ekuatorial sepanjang ekuator langit, lazim dalam jam.

Azimut. Arah mendatar pada horizon; pada aplikasi utara 0° dan timur 90° .

B

Barysentris. Mengacu pada pusat massa sistem yang ditetapkan.

BSP. Berkas biner kernel SPK yang memuat representasi ephemeris.

C

Cache. Penyimpanan untuk pemakaian ulang aset; bukan jaminan data selalu tersedia tanpa jaringan.

Chebyshev. Keluarga polinomial yang dipakai dalam representasi numerik sebagian segmen SPK.

CSV. Format tabel teks dengan pemisah koma; ekspor versi ini berisi ringkasan dan peristiwa.

D

DAF. Double Precision Array File, struktur berkas yang mendasari SPK.

DE440s. Distribusi pendek solusi Development Ephemeris 440 dari JPL.

Deklinasi. Koordinat ekuatorial utara atau selatan ekuator langit.

Delta T. Selisih skala waktu TT dan UT1.

DST. Daylight Saving Time; perubahan jam musiman pada zona yang menerapkannya.

E

Ekliptika. Bidang atau lintasan acuan terkait orbit Bumi dan gerak tahunan Matahari.

Elongasi. Sudut separasi arah Bulan–Matahari dari pengamat yang ditetapkan.

Ephemeris. Representasi posisi dan gerak benda langit sebagai fungsi waktu.

Epoch. Saat acuan koordinat atau elemen yang harus disebutkan dalam perbandingan.

F

Fajar Gisborne. Dalam versi ini, crossing naik pusat Matahari toposentris -18° pada titik Gisborne dan tanggal calon awal bulan.

G

GAST. Greenwich Apparent Sidereal Time, waktu sidereal apparent pada Greenwich.

Geosentris. Arah atau posisi dengan pusat Bumi sebagai pengamat.

H

H0. Pada scan global, tanggal UTC konjungsi; pada bagian lokal mengikuti definisi tanggal lokal yang dijelaskan.

Hilal. Sabit Bulan dekat awal lunasi yang menjadi objek pengamatan dan kajian kalender.

Horizon geometris. Bidang acuan horizontal model; berbeda dari profil halangan nyata.

I

IANA. Pengelola basis data identitas dan aturan zona waktu yang digunakan sistem.

Ijtimak. Konjungsi; definisi geosentris kalender pada buku mengikuti kesamaan bujur ekliptika.

Illuminasi. Fraksi geometris cakram yang diterangi dan terlihat dari pengamat.

J

JPL. Jet Propulsion Laboratory, penyedia solusi ephemeris yang menjadi dasar DE440s.

K

KHGT. Kalender Hijriah Global Tunggal, penanggalan yang menggunakan aturan global yang dijelaskan.

Konjungsi toposentris. Label program untuk minimum separasi apparent lokal di sekitar konjungsi geosentris.

L

Lag. Selisih moonset terpilih terhadap sunset, dalam menit dan dapat bertanda negatif.

LST. Local Sidereal Time, waktu sidereal lokal.

Lunasi. Siklus antara dua fase baru berturutan; durasinya tidak selalu tepat sama.

M

Magnitudo. Skala terang astronomis; kolom program memakai model perkiraan.

Matlak. Wilayah keberlakuan rukyat atau kriteria dalam pembahasan kalender.

N

Node. Perpotongan lintasan Bulan dengan bidang ekliptika.

P

Paralaks. Perubahan arah akibat perubahan posisi pengamat.

Perigee. Minimum jarak Bulan terhadap Bumi dalam siklus yang diperiksa.

Perihelion. Minimum jarak Bumi–Matahari dalam siklus yang diperiksa.

PKG 1. Cabang penerimaan kriteria sebelum batas tengah malam UTC.

PKG 2. Cabang koreksi setelah atau tepat pada batas UTC, dengan daratan utama Amerika dan konjungsi sebelum fajar.

Pyodide. Lingkungan Python berbasis WebAssembly yang dijalankan peramban.

R

Refraksi. Pembelokan cahaya oleh atmosfer yang mengubah arah tampak.

Reliabilitas. Dalam audit ini, kestabilan pengulangan dan sifat invariansi keluaran yang diuji.

S

Semidiameter. Setengah diameter sudut cakram benda.

Skyfield. Pustaka Python untuk posisi dan peristiwa astronomis.

SPK. Spacecraft and Planet Kernel, format ephemeris vektor dalam sistem SPICE.

Sudut fase. Sudut Matahari–benda–pengamat dengan titik sudut pada benda.

Sudut jam. Selisih waktu sidereal lokal dan asensio rekta, dinyatakan sebagai sudut atau jam.

T

TDB. Barycentric Dynamical Time, skala waktu yang terkait perhitungan ephemeris barysentris.

Toposentris. Arah atau posisi dari lokasi pengamat.

Transit. Perlintasan benda pada meridian lokal.

TT. Terrestrial Time, skala waktu dinamis geosentris.

Twilight. Keadaan transisi pencahayaan ketika Matahari berada di bawah horizon.

U

UT1. Skala waktu yang terkait rotasi Bumi.

UTC. Coordinated Universal Time, skala pelabelan waktu sipil global.

V

Validitas. Kesesuaian hasil atau klaim terhadap definisi dan bukti pembandingan yang relevan.

Visibilitas. Keterlihatan yang memerlukan definisi model atau bukti observasi.

W

WebAssembly. Format eksekusi untuk menjalankan komputasi dalam lingkungan web.

WGS84. Model geodetik yang dipakai untuk membentuk lokasi pengamat.

Z

Zona waktu. Aturan pengubahan waktu sipil lokal terhadap UTC untuk lokasi dan tanggal tertentu.

DAFTAR PUSTAKA

- [R01] Park, R. S., Folkner, W. M., Williams, J. G., & Boggs, D. H. (2021). The JPL Planetary and Lunar Ephemerides DE440 and DE441. *The Astronomical Journal*, 161, 105. DOI: 10.3847/1538-3881/abd414. [Sumber / tautan DOI](#)
- [R02] Rhodes, B. (2019). Skyfield: High precision research-grade positions for planets and Earth satellites generator. *Astrophysics Source Code Library*, ascl:1907.024. [Sumber / tautan DOI](#)
- [R03] Rhodes, B. Dokumentasi Skyfield: Planets and their moons — JPL ephemeris files. Dokumentasi pemilihan, cakupan, target, dan excerpt ephemeris. [Sumber / tautan DOI](#)
- [R04] Rhodes, B. Dokumentasi Skyfield: Positions. Posisi barycentric, astrometric, dan apparent serta pengamatan benda langit. [Sumber / tautan DOI](#)
- [R05] Rhodes, B. Dokumentasi Skyfield: Coordinates. Kerangka dan satuan koordinat astronomis. [Sumber / tautan DOI](#)
- [R06] Rhodes, B. Dokumentasi Skyfield: Time. Skala waktu dan konversi objek waktu. [Sumber / tautan DOI](#)
- [R07] Rhodes, B. Dokumentasi Skyfield: Searching for the dates of astronomical events. Almanac, pencarian peristiwa, fase, dan gerhana Bulan. [Sumber / tautan DOI](#)
- [R08] Pyodide Project. (Versi 0.27.7). Using Pyodide. Dokumentasi resmi penggunaan Python dalam lingkungan peramban. [Sumber / tautan DOI](#)
- [R09] Internet Assigned Numbers Authority. Time Zone Database. Sumber resmi identitas dan aturan zona waktu. [Sumber / tautan DOI](#)
- [R10] NASA/JPL Navigation and Ancillary Information Facility. (2021). SPK Required Reading. Dokumentasi toolkit SPICE; revisi 31 Agustus 2021. [Sumber / tautan DOI](#)
- [R11] U.S. Naval Observatory, Astronomical Applications Department. Crescent Moon Visibility. Penjelasan konjungsi dan faktor keterlihatan sabit. [Sumber / tautan DOI](#)
- [R12] Doggett, L. E., & Schaefer, B. E. (1994). Lunar crescent visibility. *Icarus*, 107(2), 388–403. DOI: 10.1006/icar.1994.1031. [Sumber / tautan DOI](#)

- [R13] Odeh, M. Sh. (2004). New Criterion for Lunar Crescent Visibility. *Experimental Astronomy*, 18, 39–64. DOI: 10.1007/s10686-005-9002-5. Tahun volume mengikuti sitasi penerbit; publikasi daring 29 Maret 2006. [Sumber / tautan DOI](#)
- [R14] Majelis Tarjih dan Tajdid PP Muhammadiyah. (2025). *Memahami Kalender Hijriah Global Tunggal Secara Ringkas*. Diringkas oleh Najmuddin Saifullah. Buku saku, Juni 2025. Bagian parameter dan koreksi kalender, khususnya hlm. 47–48 dan contoh Syawal 1448 H hlm. 52–53. [Sumber / tautan DOI](#)
- [R15] Espenak, F., & Meeus, J. (2006). *Five Millennium Canon of Solar Eclipses: –1999 to +3000*. NASA Technical Publication TP-2006-214141. [Sumber / tautan DOI](#)
- [R16] U.S. Naval Observatory, Astronomical Applications Department. *Rise, Set, and Twilight Definitions*. Definisi horizon, transit, dan twilight. [Sumber / tautan DOI](#)
- [R17] Petit, G., & Luzum, B. (Eds.). (2010). *IERS Conventions (2010)*. IERS Technical Note No. 36. Frankfurt am Main: Verlag des Bundesamts für Kartographie und Geodäsie. [Sumber / tautan DOI](#)
- [R18] Urban, S. E., & Seidelmann, P. K. (Eds.). (2013). *Explanatory Supplement to the Astronomical Almanac* (edisi ke-3). University Science Books. Metadata dan deskripsi dikonfirmasi melalui USNO; rujukan pendalaman. [Sumber / tautan DOI](#)
- [R19] NASA Eclipse Web Site. *Eye Safety During Solar Eclipses*. Panduan keselamatan pengamatan Matahari. [Sumber / tautan DOI](#)
- [R20] SOLUNA. (Versi sumber diperiksa 5 Oktober 2026). *index.php*, mesin Python tertanam, konfigurasi BSP, dan antarmuka. Bahan sumber yang diberikan pengguna. Identitas SHA-256 tercantum pada Lampiran D.
- [R21] Draft KHGT 1448__SM.pdf. Draf kalender enam halaman yang diberikan pengguna untuk perbandingan dua belas bulan 1448 H. Metadata tanggal terbit dan pengesahan resmi tidak ditetapkan dari berkas; identitas SHA-256 tercantum pada Lampiran D.
- [R22] Montenbruck, O., & Pfleger, T. (2000). *Astronomy on the Personal Computer* (edisi ke-4). Springer. DOI: 10.1007/978-3-642-03436-7. Rujukan pendalaman astronomi komputasional. [Sumber / tautan DOI](#)

PENYUSUN

- KASMUI -
Dosen Kimia, Komputasi, IT, AI UNNES

<https://hisabmu.com/>
<https://kasmui.cloud/>

Ketua PCM Gunungpati 2	Anggota Majelis Tabligh PDM Kota Semarang & PWM Jawa Tengah	Tim Pengembang Software KHGT MTT PP Muhammadiyah	Praktisi Ilmu Falak
---------------------------	---	--	------------------------

SOLUNA:

Software Astronomi Matahari dan Bulan

KHGT Times versi online
berbasis data ephemeris DE440s

Sinopsis

Buku ini menjelaskan landasan ilmiah dan panduan penggunaan SOLUNA, perangkat astronomi komputasional untuk menghitung posisi Matahari dan Bulan, fase, iluminasi, peristiwa harian, fenomena orbital, gerhana, serta analisis awal bulan Hijriah global.

Di dalamnya dibahas geometri Matahari–Bumi–Bulan, koordinat dan skala waktu, penggunaan Skyfield dan ephemeris DE440s, perbedaan geosentris–toposentris, serta cara membaca hasil perhitungan secara cermat dan bertanggung jawab.

Naskah ini juga memuat algoritma keputusan KHGT, validasi versi terhadap 1448 H, katalog lokasi pemindaian, kamus data, glosarium, serta protokol reproduksi. Buku ini bermanfaat bagi mahasiswa, dosen, peminat ilmu falak, pengembang, dan peneliti yang memerlukan rujukan ilmiah sekaligus panduan praktis.

Naskah Ilmiah • 5 Oktober 2026

KASMUI