

RANCANG BANGUN SOFTWARE *UNIVERSAL ASTROLABE* BERBASIS JAVASCRIPT DAN UJI AKURASINYA DALAM HISAB ARAH KIBLAT

M. Syaoyi Nahwandi^{1*}, Edy Setyawan² Akhmad Nadirin³

IAIN Syekh Nurjati Cirebon

Email: ¹*ms_nahwandi@syekhnurjati.ac.id*, ²*edysetyawan@syekhnurjati.ac.id*

³*akhmadnadirin@syekhnurjati.ac.id*

*Korespondensi

Abstract:

The astrolabe, as a classical astronomical instrument, holds significant historical and educational value, yet its use has become limited due to technological advancements. This study aims to develop a JavaScript-based Universal Astrolabe *software* to digitize the functions of a physical astrolabe, with a focus on testing the accuracy of Qibla direction calculations. The Research and Development (R&D) method was implemented through stages of needs analysis, design, implementation, and accuracy testing. Test results from three locations (Cirebon, Kuala Lumpur, Istanbul) showed minor discrepancies ($0^{\circ}1'56''-0^{\circ}4'20''$) between the *software's* calculations and those derived from the spherical triangle formula using a calculator. These differences were attributed to the smallest rotation angle rounding of the Rete ($0^{\circ}15'$) and the 1° interval scale for altitude and azimuth. The *software* offers advantages in flexibility, automated Rete positioning, and ease of presentation compared to physical astrolabes, though it does not replace the function of direct observation. The interactive simulation of the Universal Astrolabe *software* facilitates its use for educational purposes in Islamic astronomy (Ilmu Falak).

Keywords: Astrolabe, Digitization, JavaScript, Qibla Direction, Islamic Astronomy, Accuracy.

Abstrak:

Astrolabe sebagai instrumen astronomi klasik memiliki nilai historis dan edukatif yang tinggi, namun penggunaannya terbatas akibat perkembangan teknologi. Penelitian ini bertujuan mengembangkan *software Universal Astrolabe* berbasis JavaScript untuk mendigitalisasi fungsi astrolabe fisik, dengan fokus pengujian pada akurasi perhitungan arah Kiblat. Metode *Research and Development* (R&D) diterapkan melalui tahapan analisis kebutuhan, desain, implementasi, dan uji akurasi. Hasil pengujian di tiga lokasi (Cirebon, Kuala Lumpur, Istanbul) menunjukkan selisih yang kecil ($0^{\circ}1'56''-0^{\circ}4'20''$) antara hasil perhitungan *software* dan hasil perhitungan rumus segitiga bola menggunakan kalkulator, yang disebabkan oleh pembulatan sudut terkecil perputaran *Rete* $0^{\circ}15'$ dan skala interval 1° pada *altitude* dan *azimuth*. *Software* ini menawarkan keunggulan fleksibilitas, otomatisasi pemosisian *Rete*, dan kemudahan presentasi dibanding astrolabe fisik, meski tidak menggantikan fungsi observasi langsung. Simulasi interaktif *software Universal Astrolabe* memudahkan penggunaan untuk tujuan edukasi ilmu falak.

Kata Kunci: Astrolabe, Universal, Digitalisasi, JavaScript, Arah Kiblat, Akurasi.

PENDAHULUAN

Astrolabe adalah sebuah instrumen astronomi yang telah digunakan sejak zaman kuno untuk mengukur posisi bintang dan planet, serta untuk menentukan waktu. Astrolabe telah digunakan dalam berbagai budaya dan dapat dianggap sebagai salah satu alat navigasi dan astronomi paling penting pada masanya.¹ Meskipun sudah tidak banyak digunakan dalam pengamatan astronomi modern, astrolabe masih dihargai sebagai alat bersejarah dan memiliki nilai edukatif dalam memahami perkembangan ilmu pengetahuan dan navigasi.² Penggunaan Astrolabe sebagai alat pembelajaran memungkinkan pengintegrasian berbagai konsep ilmiah dan keterampilan praktis. Oleh karena itu, alat ini tidak hanya memiliki nilai historis dan astronomi, tetapi juga dapat memberikan kontribusi besar dalam pengembangan pengetahuan dan keterampilan para pelajar atau penggunanya.

Astrolabe memiliki nilai edukatif yang signifikan karena mencakup berbagai aspek pengetahuan, termasuk astronomi, matematika, dan navigasi.³ Sehingga di era perkembangan teknologi saat ini, Astrolabe yang mewakili pencapaian teknologi dan pengetahuan pada masa lampau perlu dikembangkan menjadi *software* atau aplikasi agar tetap eksis di masa sekarang. Hal ini bukanlah hal baru karena banyak peneliti di bidang ilmu Falak yang mengembangkan *software* perhitungan klasik ilmu Falak seperti aplikasi Digital Falak karya Ahmad Tholhah yang dapat digunakan untuk perhitungan waktu salat, arah kiblat dan jam istiwa' dengan berbagai metode perhitungan falak klasik dan modern. Namun kebanyakan *software* ilmu Falak yang telah dirancang, memiliki fungsi utama untuk membantu praktisi Falak di lapangan dalam memberikan data secara praktis sehingga kurang memiliki nilai edukasi. Namun jika Astrolabe yang memiliki nilai edukasi tinggi ini dirancang menjadi sebuah *software*, maka produk digitalisasi Astrolabe diharapkan mampu mengedukasi penggunanya mengenai khazanah ilmu Falak di masa lalu yang telah dipadukan dengan kemajuan teknologi.

Penelitian ini merupakan penelitian yang menggunakan metode *Research and Development* (penelitian dan pengembangan). Metode Penelitian dan Pengembangan adalah metode penelitian yang digunakan untuk menghasilkan produk tertentu dan menguji keefektifan produk tersebut.⁴ Penelitian dan Pengembangan atau *Research and Development* (R&D) adalah suatu proses atau langkah-langkah untuk mengembangkan suatu produk baru atau menyempurnakan produk yang telah ada, yang dapat dipertanggungjawabkan⁵. Dalam penelitian ini, peneliti mengembangkan *software* Astrolabe dengan bahasa pemrograman JavaScript. Teknik pengumpulan data yang digunakan dalam penelitian ini adalah dengan metode dokumentasi untuk mendapatkan teori, cara pembuatan, dan penggunaan Astrolabe dari sumber-sumber primer. Hasil dari dokumentasi data primer tersebut dijadikan dasar untuk mendapatkan desain produk yang akan dikembangkan menjadi *software* berbasis JavaScript.

¹ John North, *The Norton History of Astronomy and Cosmology* (New York: W.W. Norton & Company, 1995), 45-47.

² David A. King, *In Synchrony with the Heavens: Studies in Astronomical Timekeeping and Instrumentation in Medieval Islamic Civilization* (Leiden: Brill, 2005), 2: 563-570.

³ David A. King, *Astronomy in the Service of Islam* (Aldershot: Variorum, 1993), 312-315

⁴ Borg & Gall, *Educational Research: An Introduction, 7th ed.* (New York: Longman, 2003), 569-571.

⁵ Sugiyono, *Metode Penelitian Kuantitatif, Kualitatif, dan R&D*, Bandung: Alfabeta, 2013, Cet. 19, hal.

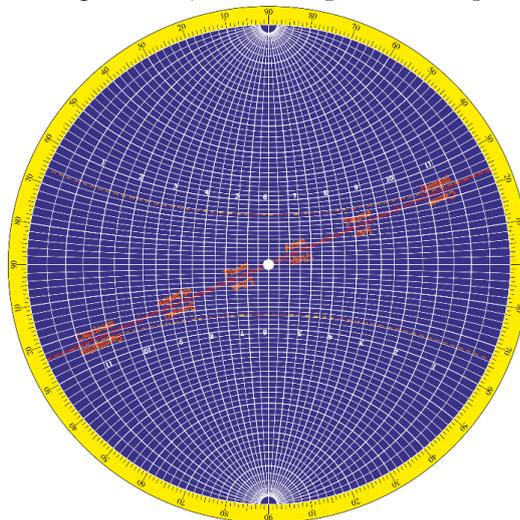
Teknik analisis data yang digunakan dalam penelitian ini adalah deskriptif analitis untuk menggambarkan sebuah pemahaman secara deskriptif mengenai konstruksi dan pengaplikasian Astrolabe. Kemudian dilakukan proses pengembangan dari interpretasi terhadap konsep dan kebutuhan dalam pembuatan *software* Astrolabe. Peneliti juga menggunakan uji fungsionalitas *software* dan uji akurasi hasil untuk memastikan semua fitur yang ada pada *software* atau aplikasi yang dirancang dapat berjalan dengan normal serta memiliki akurasi yang sesuai dengan skala yang digunakan dalam pembuatan desain Astrolabe.

PEMBAHASAN

1. Pengertian *Universal Astrolabe*

Universal Astrolabe merupakan jenis *Astrolabe* yang dapat digunakan untuk semua lokasi di Bumi, untuk lintang tempat 0° sampai dengan 90° . *Universal Astrolabe* berisi kurva-kurva hasil proyeksi koordinat Horizontal yang dapat diposisikan dan disesuaikan dengan kurva-kurva koordinat Equatorial sehingga proyeksi koordinat bola langit di semua lintang tempat dapat tergambarkan pada bidang dua dimensi lempengan *Universal Astrolabe*.⁶

Universal Astrolabe pertama kali dirancang oleh Abū Ishāq Ibrāhīm Ibn Yahyā al-Zarqālī, seorang ahli matematika sekaligus astronom Muslim dari Toledo, Andalusia (Spanyol). Al-Zarqālī menggunakan proyeksi stereografik untuk memproyeksikan koordinat bola langit dengan acuan kutub langit, lingkaran *solstice* dan *equinox*.⁷ Hasil dari proyeksi tersebut merupakan sebuah lempengan yang berisi kurva-kurva proyeksi lingkaran deklinasi dan lingkaran jam benda langit dengan posisi titik kutub langit yang berada tepat di atas dan di bawah. Lempengan hasil proyeksi yang dirancang oleh al-Zarqālī kemudian disebut dengan *Safīḥah al-Zarqālī (The Saphea Arzachelis)*.



Gambar 1. Desain bagian depan *Universal Astrolabe* al-Zarqālī

Universal Astrolabe yang dirancang oleh al-Zarqālī dan karyanya dalam bidang astronomi membantu dalam pengembangan alat-alat astronomi dan pengukuran waktu

⁶ David A. King, *In Synchrony with the Heavens: Studies in Astronomical Timekeeping and Instrumentation in Medieval Islamic Civilization* (Leiden: Brill, 2004), 145–148.

⁷ James E. Morrison, *The Astrolabe* (London: Reichelt & Sons, 2007), 112–115.

yang lebih presisi pada masa itu. Inovasinya tidak hanya berdampak di dunia Islam tetapi juga mempengaruhi perkembangan ilmu pengetahuan dan teknologi di seluruh dunia. Beberapa perbaikan yang diberikan oleh al-Zarqālī pada astrolabe di antaranya:

a. Tambahannya Skala dan Informasi:

Al-Zarqālī meningkatkan astrolabe dengan menambahkan skala-skala tambahan dan informasi yang memungkinkan untuk pengukuran yang lebih akurat dan berbagai perhitungan astronomi.⁸

b. Penyederhanaan Desain:

Al-Zarqālī merancang astrolabe dengan desain yang lebih sederhana, memudahkan penggunaan dan interpretasi, terutama bagi ilmuwan dan navigator pada masanya.⁹

c. Presisi Pengukuran Waktu:

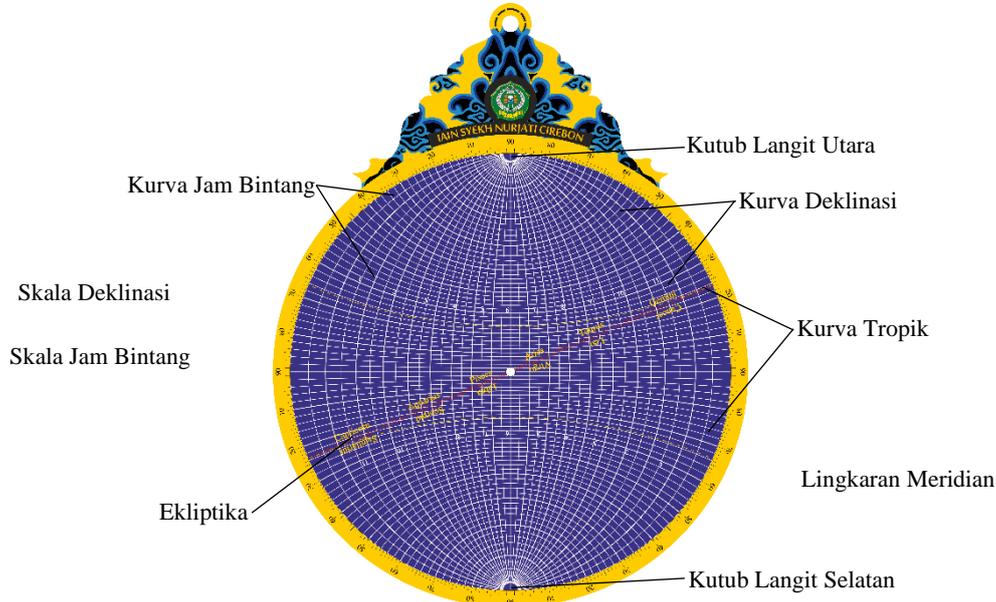
Al-Zarqālī juga memperbaiki instrumen-instrumen di dalam astrolabe untuk meningkatkan presisi pengukuran waktu dan memudahkan navigasi.¹⁰

d. Peningkatan dalam Tabel Astronomi:

Selain perbaikan pada astrolabe, al-Zarqālī membuat tabel astronomi yang lebih akurat, seperti *al-Zij al-Mumtahan*, yang membantu dalam perhitungan astronomi yang lebih tepat.¹¹

2. Bagian-Bagian *Universal Astrolabe*

a. *Mater* atau *Front* (bagian depan)



Gambar 2. Komponen bagian depan *Universal Astrolabe* al-Zarqālī

1) Kutub Langit Utara dan Selatan

⁸ Julio Samsó, *Islamic Astronomy and Medieval Spain* (Aldershot: Variorum, 1994), 78

⁹ Julio Samsó, *Islamic...*, 78

¹⁰ Julio Samsó, *Islamic...*, 80

¹¹ José Chabás dan Bernard R. Goldstein, *The Alfonsine Tables of Toledo* (Dordrecht: Springer, 2003), 45–47.

Kutub langit Utara dan Selatan merupakan proyeksi kutub Utara dan kutub Selatan Bumi pada bola langit. Pada *Universal Astrolabe* desain Al-Zarqālī, kutub langit Utara berada pada bagian atas sedangkan kutub langit Selatan berada pada bagian bawah.¹²

2) Kurva Deklinasi

Deklinasi adalah jarak suatu benda langit dihitung dari equator langit. Jika posisi benda langit berada di sebelah utara equator langit, maka benda langit tersebut memiliki deklinasi utara dan bernilai positif. Sedangkan Jika posisi benda langit berada di sebelah selatan equator langit, maka benda langit tersebut memiliki deklinasi selatan dan bernilai negatif.

Pada *Universal Astrolabe* desain Al-Zarqālī, kurva deklinasi merupakan kurva-kurva yang melintang secara horizontal yang digunakan untuk mengilustrasikan pergerakan harian benda langit pada masing-masing nilai deklinasi.

3) Skala Deklinasi

Skala deklinasi digunakan untuk menghitung nilai deklinasi benda langit yang diproyeksikan pada *Universal Astrolabe* desain Al-Zarqālī. Kutub langit utara dan selatan masing-masing memiliki nilai skala deklinasi 90° sedangkan equator langit memiliki nilai skala deklinasi 0° .¹³

4) Equator Langit

Equator langit adalah Equator Bumi yang diproyeksikan pada bola langit. Pada *Universal Astrolabe* desain al-Zarqālī, Equator langit diproyeksikan sebagai garis horizontal yang membagi *Mater* menjadi dua bagian yang sama besarnya. Equator langit memiliki skala nilai deklinasi 0° .

5) Lingkaran Meridian

Lingkaran meridian merupakan lingkaran istimewa pada bola langit yang menghubungkan Kutub Langit Utara-Selatan, Equator langit, titik arah utara-selatan, dan titik Zenith-Nadir. Pada *Universal Astrolabe* desain Al-Zarqālī, lingkaran meridian diproyeksikan sebagai lingkaran tepi pada *Mater*.

6) Kurva Jam Bintang

Jam bintang atau *Hour Angle* adalah jarak suatu benda langit dari lingkaran meridian. Disebut jam bintang karena dengan menghitung jarak suatu benda langit dari lingkaran meridian, dapat diketahui nilai sudut waktu dan jam hakiki dari posisi benda langit tersebut. Pada *Universal Astrolabe* desain Al-Zarqālī, kurva jam bintang diproyeksikan sebagai kurva-kurva vertical yang menghubungkan kutub langit Utara dan Selatan.¹⁴

7) Skala Jam Bintang

Skala jam bintang digunakan untuk menghitung sudut waktu atau *hour angle* suatu benda langit yang diproyeksikan pada *Universal Astrolabe* desain Al-Zarqālī.

¹² David A. King, In *Synchrony ...*, 150–152.

¹³ James E. Morrison, *The Astrolabe* (London: Reichelt & Sons, 2007), 118.

¹⁴ David A. King, *Astronomy in the Service of Islam* (Aldershot: Variorum, 1993), 230.

Skala jam bintang dinyatakan dalam satuan jam untuk mempermudah pencarian waktu hakiki berdasarkan posisi suatu benda langit.

8) Kurva Tropik

Lingkar Tropik merupakan lingkaran gerak semu harian Matahari saat Matahari berada di titik solstice atau titik balik utara dan selatan. Lingkaran Tropik ada dua macam yaitu lingkaran Tropik Cancer dan Lingkaran Tropik Capricorn. Lingkaran Tropik Cancer disebut juga sebagai lingkaran gerak semu harian Matahari saat Matahari berada di titik *Summer Solstice*. Sedangkan lingkaran Tropik Capricorn disebut juga sebagai lingkaran gerak semu harian Matahari saat Matahari berada di titik *Winter Solstice*.

Pada *Universal Astrolabe* desain Al-Zarqālī, Lingkaran Tropik diproyeksikan sebagai dua kurva dengan nilai skala deklinasi $23^{\circ}26'12''$ yang merupakan senilai dengan *obliquity* atau sudut kemiringan ekliptika dari equator langit.

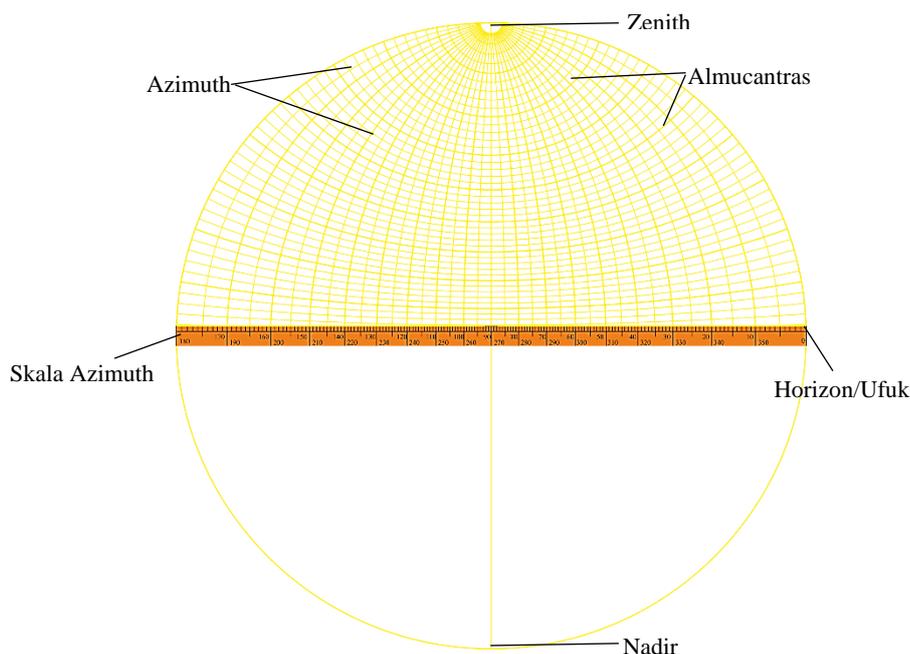
9) Ekliptika

Ekliptika merupakan bidang orbit Bumi mengelilingi Matahari. Pada bola langit dengan acuan Bumi (geosentris), ekliptika diproyeksikan sebagai lingkaran pergerakan semu tahunan Matahari. Pada saat melakukan pergerakan semu tahunan (dengan acuan Bumi), Matahari terlihat berada pada rasi-rasi bintang. Rasi-rasi bintang yang berada pada lingkaran ekliptika disebut Zodiak.

Pada *Universal Astrolabe* desain Al-Zarqālī, Ekliptika diproyeksikan sebagai garis horizontal yang memiliki kemiringan sebesar $23^{\circ}26'12''$ dari equator langit. Pada garis ekliptika, terdapat skala zodiac yang dapat digunakan untuk menentukan posisi Matahari berdasarkan nilai *darajah al-Syams* atau bujur Matahari.

b. Rete

Pada *Universal Astrolabe* desain Al-Zarqālī, *Rete* adalah lempengan *Astrolabe* bagian depan yang dapat diputar. Rete berisi kurva-kurva hasil proyeksi tata koordinat horizontal bola langit yang berisi kurva-kurva *almucantras* dan *Azimuth*.



Gambar 3. Komponen *Rete Universal Astrolabe* al-Zarqālī

1) Zenith

Zenith adalah titik puncak atau atas kepala pengamat. Pada *Universal Astrolabe* desain Al-Zarqālī, Zenith merupakan titik puncak atau titik atas *Rete*. Pemosisian Zenith pada skala deklinasi harus tepat sesuai dengan nilai lintang tempat. Jika *Universal Astrolabe* al-Zarqālī digunakan untuk 6° LU, maka Zenith diposisikan berada pada nilai skala deklinasi 6° di Utara (atas) garis Equator langit. Sedangkan jika *Universal Astrolabe* Al-Zarqālī digunakan untuk 6° LS, maka Zenith diposisikan berada pada nilai skala deklinasi 6° di Selatan (bawah) garis Equator langit.¹⁵

2) Nadir

Nadir titik bawah kaki pengamat. Titik Nadir selalu berada 180° dari titik Zenith dan -90° dari Horizon.

3) Garis Horizon/Ufuk

Horizon/ufuk adalah batas antara langit dan laut (tanah) yang dilihat oleh pengamat. Pada *Universal Astrolabe* desain Al-Zarqālī, Horizon/Ufuk diproyeksikan sebagai garis tengah *Rete* yang membelah *Rete* menjadi dua bagian (atas dan bawah) yang sama besar. Pada garis Horizon/Ufuk, biasanya diletakkan ruler (penggaris) yang berisi skala *Azimuth*.¹⁶

4) *Almucantras*

Almucantras adalah kurva-kurva yang melintang secara horizontal pada *Rete Universal Astrolabe* yang digunakan untuk memproyeksikan posisi benda langit berdasarkan ketinggiannya dari ufuk. Atau dapat dikatakan pula bahwa *Almucantras* adalah kurva-kurva horizontal pada *Rete Universal Astrolabe* yang digunakan untuk menghitung tinggi benda langit dari ufuk.

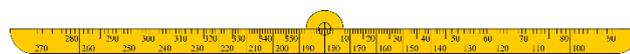
5) *Azimuth*

Azimuth adalah busur pada lingkaran ufuk yang dihitung dari titik Utara searah jarum jam hingga ke posisi benda langit. Pada *Universal Astrolabe* desain al-Zarqālī, *Azimuth* diproyeksikan sebagai kurva-kurva vertical pada *Rete* yang digunakan untuk memproyeksikan posisi benda langit berdasarkan *Azimuth*nya. Atau dapat dikatakan pula bahwa *Azimuth* adalah kurva-kurva vertical pada *Rete Universal Astrolabe* yang digunakan untuk menghitung *Azimuth* benda langit.

6) Skala *Azimuth*

Skala *Azimuth* berada pada garis Horizon/Ufuk yang digunakan untuk mengetahui nilai *Azimuth* benda langit. Skala *Azimuth* berada di garis Horizon/Ufuk. Nilai 0° skala *Azimuth* berada pada lingkaran meridian.

7) *Ruler*



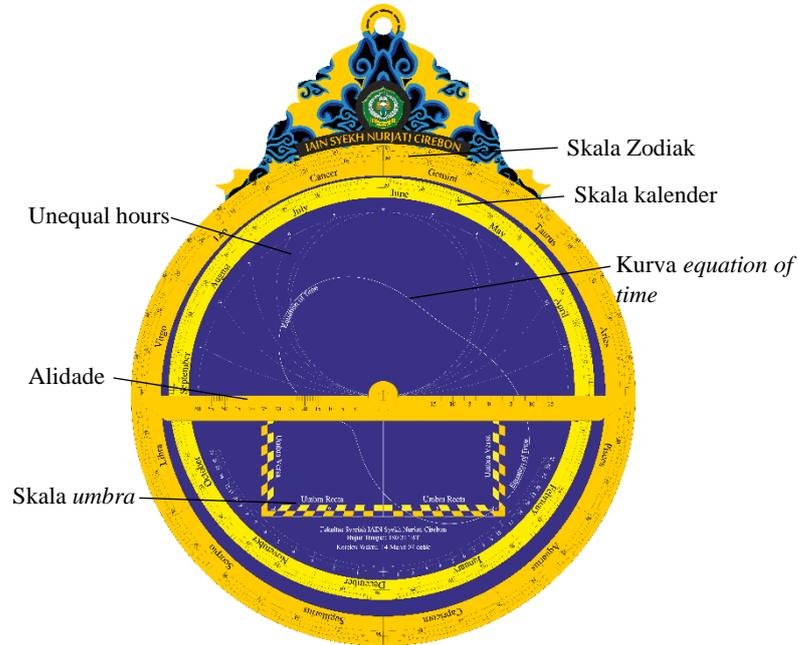
Gambar 4. *Ruler Universal Astrolabe* al-Zarqālī

¹⁵ James E. Morrison, *The Astrolabe* (London: Reichelt & Sons, 2007), 119.

¹⁶ James E. Morrison, *The Astrolabe* (London: Reichelt & Sons, 2007), 119.

Ruler adalah penggaris yang berada di bagian *Mater* atau depan *Universal Astrolabe* yang dapat diputar dan digunakan untuk menghitung nilai sudut waktu benda langit.

c. *Back* (Bagian Belakang)



Gambar 5. Komponen bagian belakang *Universal Astrolabe* al-Zarqālī

1) Skala Zodiak

Skala Zodiak adalah skala yang digunakan untuk mengetahui posisi Matahari pada ekliptika. Terdapat 12 Zodiak yang berada pada skala zodiac. Masing-masing zodiac pada skala zodiac memiliki besar sudut 30° .

2) Skala Kalender

Skala Kalender berisi tanggal dan bulan dalam kalender Masehi. Dengan memposisikan *Alidade* pada tanggal yang diinginkan, dapat diketahui posisi Matahari pada tanggal tersebut berdasarkan zodiac pada lingkaran ekliptika.

3) Kurva *Equation of Time*

Equation of Time atau perata waktu adalah selisih antara waktu hakiki (waktu Matahari) dengan sistem waktu yang digunakan pada masa sekarang. Pada *Universal Astrolabe*, kurva *Equation of Time* adalah kurva yang digunakan untuk mengetahui nilai *Equation of Time* pada setiap tanggal Masehi. Dengan memposisikan *Alidade* pada tanggal yang diinginkan dan melihat posisi *Alidade* pada kurva *Equation of Time*, nilai *Equation of Time* pada tanggal tersebut dapat diketahui secara langsung.

4) *Unequal Hours*

Unequal Hours adalah sistem waktu yang digunakan pada abad pertengahan di Romawi. Pada masa sekarang, *Unequal Hours* tidak lagi digunakan secara umum. Namun beberapa masih menggunakannya untuk perhitungan waktu ibadah umat Kristen Orthodox.

5) Skala *Umbra*

Skala *Umbra* atau skala bayangan adalah skala yang digunakan untuk menghitung jarak atau ketinggian suatu objek di Bumi. Umumnya skala *Umbra* menggunakan basis bilangan 12.

6) *Alidade*

Alidade adalah penggaris yang berada di belakang *Astrolabe*. *Alidade* digunakan untuk mendapatkan nilai bujur Matahari dan *Equation of Time* berdasarkan tanggal. Pada *Alidade* terdapat skala *Equation of Time* yang digunakan untuk membaca nilai *Equation of Time* dan skala *Haueter* atau ketinggian maksimum Matahari untuk mendapatkan waktu dalam sistem *Unequal Hours*.

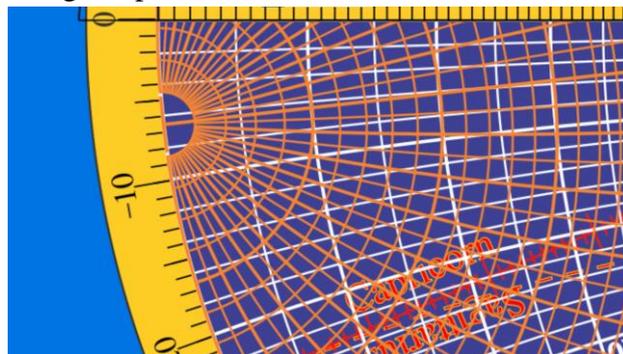
3. Cara Penggunaan *Universal Astrolabe* dalam Hisab Arah Kiblat

Langkah-langkah penggunaan *Universal Astrolabe* Al-Zarqālī dalam hisab arah Kiblat adalah sebagai berikut:

- Gunakan bagian depan *Universal Astrolabe* dan pastikan posisi titik zenith pada *Rete* telah tepat sesuai lintang tempat.
- Hitung nilai selisih bujur tempat dengan bujur Ka'bah (SB)
- Rubah nilai SB yang bersatuan derajat menjadi satuan jam
- Tandai posisi perpotongan antara kurva altitude yang senilai dengan lintang Ka'bah dengan kurva jam nilai SB
- Lihat posisi tanda tersebut berdasarkan kurva azimuth. Maka itulah azimuth Kiblat untuk lokasi tersebut.

Contoh menghitung arah kiblat untuk Cirebon dengan Lintang Tempat: 6°43' LS dan Bujur Tempat 108°34' BT:

- Gunakan bagian depan *Universal Astrolabe* dan pastikan posisi titik zenith pada *Rete* telah tepat sesuai lintang tempat 6°43' LS



Gambar 6. posisi titik zenith pada *Rete* telah tepat sesuai lintang tempat 6°43' LS

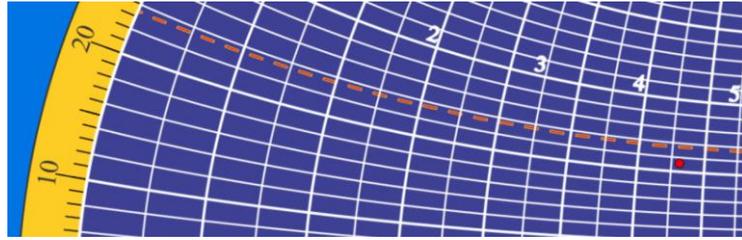
- Hitung nilai selisih bujur tempat dengan bujur Ka'bah (SB):

$$\begin{aligned} \text{SB} &= 108^{\circ}34' - 39^{\circ}49'34,36'' \\ &= 68^{\circ}44'25'' \end{aligned}$$

- Rubah nilai SB yang bersatuan derajat menjadi satuan jam

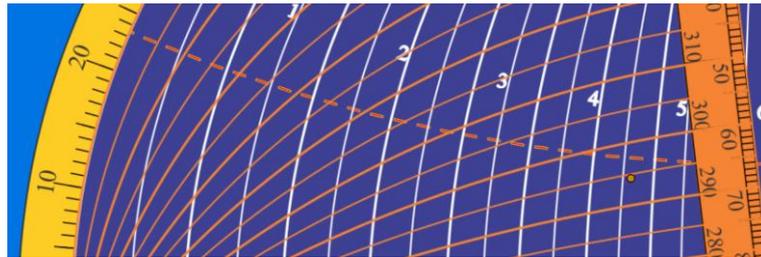
$$68^{\circ}44'25'' / 15 = 4:34:58$$

- Tandai posisi perpotongan antara kurva altitude yang senilai dengan lintang Ka'bah (21°25'21" LU) dengan kurva jam nilai SB



Gambar 7. *Pointer* diposisikan pada titik perpotongan antara kurva altitude yang senilai dengan lintang Ka'bah

e) Lihat posisi tanda tersebut berdasarkan kurva azimuth. Maka itulah azimuth Kiblat untuk Cirebon, yaitu $294^{\circ}50'$



Gambar 8. Pembacaan posisi *pointer* berdasarkan kurva azimuth dengan bantuan *ruler* skala azimuth

4. Gambaran Umum Pemrograman berbasis JavaScript

JavaScript (JS) merupakan salah satu bahasa pemrograman utama dalam pengembangan web, bekerja bersama HTML dan CSS.¹⁷ Pada 2022, hampir 98% website memanfaatkan JS di sisi klien guna memantau interaksi pengguna, dengan banyak mengandalkan *library* eksternal.¹⁸ Setiap browser modern dilengkapi *engine* khusus untuk menjalankan kode JS langsung di perangkat pengguna. Sebagai bahasa tingkat tinggi yang mengikuti standar ECMAScript, JS menggunakan kompilasi *just-in-time* dan memiliki fitur seperti *dynamic typing*, *prototype-based object-oriented programming*, serta *first-class functions*.¹⁹ Bahasa ini bersifat *multi-paradigm*, mendukung pemrograman berbasis *event*, *functional*, maupun *imperative*.²⁰

Developer dapat memanfaatkan JavaScript API untuk mengakses berbagai fitur, termasuk manipulasi teks, tanggal, regular expressions, struktur data, dan DOM. Meski standar ECMAScript tidak mencakup *built-in I/O* seperti operasi jaringan atau grafis, lingkungan eksekusi seperti browser menyediakan API tambahan untuk keperluan tersebut. Awalnya, JS hanya berjalan di browser, namun kini menjadi bagian penting di sisi server melalui runtime seperti Node.js. Walaupun namanya mirip, JavaScript dan Java memiliki perbedaan mendasar dalam sintaksis dan arsitektur, meski keduanya sering dibandingkan.

¹⁷ MDN Web Docs, "JavaScript", Mozilla Developer Network, 2023, <https://developer.mozilla.org/en-US/docs/Web/JavaScript> (akses 13 April 2024).

¹⁸ Douglas Crockford, *JavaScript: The Good Parts* (Sebastopol: O'Reilly Media, 2008), 2-5.

¹⁹ ECMA International, *ECMAScript® 2023 Language Specification*, 14th Edition (June 2023), <https://262.ecma-international.org/14.0/>.

²⁰ David Flanagan, *JavaScript: The Definitive Guide, 7th ed.* (Sebastopol: O'Reilly Media, 2020), 45-48.

Sebagai bahasa scripting, JS dijalankan dalam dokumen HTML dan menjadi pionir dalam kategori bahasa skrip web. Fungsinya memperluas kemampuan HTML dengan mengeksekusi perintah di sisi klien (*client-side*), tanpa perlu pengolahan di server. Eksekusi kode JS bergantung pada browser yang memproses skrip tertanam dalam HTML. Berbeda dengan Java yang membutuhkan kompiler terpisah, JS dapat langsung dijalankan karena *engine*-nya sudah terintegrasi dalam browser.²¹

Terdapat dua cara untuk menuliskan kode-kode JavaScript agar dapat ditampilkan pada halaman HTML, yaitu:

a. JavaScript ditulis pada file yang sama

Untuk penulisan dengan cara ini, perintah yang digunakan adalah

```
<SCRIPT LANGUAGE = "JavaScript" >
program java script disini
</SCRIPT>.
```

Perintah tersebut biasanya diletakkan diantara Tag <BODY>...</BODY>

Contoh Penulisan :

```
<HTML>
<HEAD><TITLE>.....</TITLE>
</HEAD>
<BODY>
<SCRIPT LANGUAGE="Javascript">
kode javascript disini
</SCRIPT>
kode HTML disini
</BODY>
</HTML>
```

b. JavaScript ditulis pada file terpisah

Kode Javascript bisa juga kita buat dalam file terpisah dengan tujuan agar dokumen HTML isinya tidak terlalu panjang. Atribut yang digunakan adalah

```
<SCRIPT SRC="namafile.js">...</SCRIPT>
```

Diantara tag <SCRIPT...> dan <SCRIPT> tidak diperlukan lagi kode JavaScript karena sudah dibuat dalam file terpisah. File yang mengandung kode Javascript berekstensi *.js.

Secara garis besar, arsitektur dan komponen yang membangun kode skrip berbasis JavaScript adalah sebagai berikut:

a. *Data Structure* (Struktur Data)

Semua bahasa pemrograman memiliki struktur data bawaan, tetapi model struktur data pada satu bahasa pemrograman dengan yang lainnya sering berbeda. Berikut ini struktur data yang tersedia pada JavaScript:

1) *Boolean*

Tipe data *Boolean* mewakili entitas logis dan memiliki dua nilai, yaitu: *true* dan *false*. Nilai Boolean biasanya digunakan untuk operasi bersyarat, termasuk operator logika: *if...else*, *while*, dan lainnya.

2) *Number*

Tipe *Number* adalah tipe data berformat angka baik berupa bilangan decimal maupun bilangan bulat. JavaScript mampu menyimpan tipe *Number* dengan angka

²¹ W3Schools, "JavaScript Introduction", 2023, https://www.w3schools.com/js/js_intro.asp (akses 13 April 2024).

antara nilai minimum 2^{-1074} (`Number.MIN_VALUE`) hingga maksimal 2^{1024} (`Number.MAX_VALUE`). Sedangkan bilangan bernilai negatif antara -2^{-1074} dan -2^{1024} . Namun *Number* pada JavaScript hanya dapat menyimpan bilangan bulat dan desimal dengan aman dalam rentang nilai bilangan yang bernilai antara $-(2^{53} - 1)$ (`Number.MIN_SAFE_INTEGER`) hingga $2^{53} - 1$ (`Number.MAX_SAFE_INTEGER`).

3) *BigInt*

Tipe *BigInt* adalah tipe data berformat angka dalam JavaScript yang merupakan pembulatan dari angka decimal dengan memperhatikan nilai decimal setelah koma.

Contoh:

```
const x = BigInt (2.6); // nilai x = 3
const y = BigInt (2.3); // nilai y = 2
```

4) *String*

Tipe *String* adalah data tekstual yang tidak dapat diproses dengan operator aritmatika. Sehingga saat data dengan tipe *String* ditulis, maka tidak dapat dirubah. Tipe data *String* yang berisi huruf harus ditulis di antara dua tanda petik. Contoh "ini data String" dan "12".

5) *Dates* (tanggal)

Tipe *Dates* adalah data yang berisi informasi tanggal, bulan, dan tahun yang dapat diambil dengan menggunakan fungsi `Date ()`.

6) *Array*

Array merupakan tipe data terstruktur yang berguna untuk menyimpan sejumlah data yang bertipe sama. Biasanya *Array* berupa sebuah struktur data yang terdiri atas banyak *variable* dengan tipe data sama, di mana masing-masing elemen *variable* memiliki nilai indeks.

b. *Variable* (Variabel)

Variable adalah nama simbolis untuk nilai dalam aplikasi. Nama-nama *variable* yang disebut pengidentifikasi (*identifier*) nilai atau data, ditulis dengan menggunakan aturan tertentu. Penulisan *variable* pada JavaScript juga dapat menggunakan sebagian besar huruf ISO 8859-1 atau Unicode seperti å dan ü. Beberapa penulisan *variable* pada JavaScript adalah `Number_hits`, `temp99`, `$credit`, dan `_name`. Adapun mendeklarasikan *variable* JavaScript dapat dilakukan dengan dua cara:

- 1) Dengan kata kunci `var`. Contoh: `var x = 42`. Sintaks ini dapat digunakan untuk mendeklarasikan variabel lokal dan global, tergantung pada konteks eksekusi.
- 2) Dengan kata kunci `const` atau `let`. Contoh: `let y = 13`. Sintaks ini dapat digunakan untuk mendeklarasikan variabel lokal lingkup-blok.

Variable harus selalu dideklarasikan sebelum digunakan. Terkadang penulisan *Variable* pada JavaScript tidak dideklarasikan secara langsung dengan nilainya. Hal ini memungkinkan bagi seorang *programmer* untuk menghimpun berbagai *variable* yang akan digunakan untuk menyimpan hasil dari proses coding pada aplikasi. Contoh:

```
Var Nama, Umur, Kota, Provinsi;
Nama = "Syaoqi";
Umur = 30;
Kota = "Cirebon";
Provinsi = "Jawa_Barat";
```

5. Implementasi Rancangan *Software Universal Astrolabe* Berbasis JavaScript

Instrumen Astrolabe merupakan khazanah yang perlu untuk dilestarikan. Di era *industry 4.0* ini, instrument klasik ilmu falak perlu dirancang menjadi *software* yang dapat dijalankan di perangkat yang umum digunakan di masa sekarang, seperti laptop. Dalam penelitian ini, peneliti mengembangkan sebuah *software Universal Astrolabe* berbasis JavaScript yang memudahkan pengguna untuk mempelajari dan mengoperasikan *Universal Astrolabe* melalui laptop dengan OS Windows. *software Universal Astrolabe* ini didesain *easy to use*, di mana pengguna dapat mengoperasikan *Universal Astrolabe* dengan menggunakan keyboard laptop. Berikut ini Implementasi rancangan *software Universal Astrolabe* berbasis JavaScript:

a. Pemilihan IDE

Dalam pembuatan *software* ini peneliti menggunakan *Integrated Development Environment* (IDE) atau lingkungan pengembangan bernama SwishMax, yaitu sebuah perangkat lunak yang dikhususkan untuk membuat animasi dan aplikasi multimedia interaktif. Dengan menggunakan SwishMax, pengguna dapat membuat animasi Flash dengan berbagai efek visual dan suara.

b. Implementasi tampilan awal *software*

Tampilan awal *software* berisi daftar pilihan menu utama *software* Digitalisasi Instrumen Falak klasik yang terdiri dari menu *Universal Astrolabe* dan *Rubu' Mujayyab*. Untuk membuat tampilan awal pada *software*, peneliti menuliskan *Source code* pada file berekstensi .xml. Berikut ini *source code* tampilan menu awal *software*:

```
onSelfEvent (load) {
    fscommand("AllowScale",true);
    fscommand("FullScreen",true);
}
onSelfEvent (load) {
    klk_hm._visible=0;
}
on (release){
    tentang_software_png._visible
=0;
Background_Aplikasi_wide_png._v
isible =1;
Background_Aplikasi_png._visibl
e = 1;
    icon_png._visible=1;
    home_png._visible=0;
    klk_hm._visible=0;
    exit_btn._visible=1;
    full_btn._visible=1;
}
onSelfEvent (load){
    this._visible=0;
}
on (release){
    tentang_software_png._visible
=0;
Background_Aplikasi_wide_png._v
isible =0;
    Background_Aplikasi_png._visible
= 0;
    icon_png._visible=0;
    home_png._visible=1;
    exit_btn._visible=0;
    full_btn._visible=0;
    klk_hm._visible=1;
}
on (press){
    fscommand("FullScreen",false);
}
on (release){
    fscommand("FullScreen",true);
}
on (release){
    fscommand("Quit",true);
}
```

Berdasarkan *source code* di atas, tampilan awal *software* terdiri dari 5 tombol yang akan mengarahkan ke tampilan *software Universal Astrolabe*, tampilan *software Rubu' Mujayyab*, panduan pengguna, tampilan *fullscreen*, dan tombol untuk menutup *software*. Adapun hasil implementasi tampilan awal *software* adalah sebagai berikut:



Gambar 9. Tampilan *software* Digitalisasi Instrumen Klasik Ilmu Falak (*Universal Astrolabe* dan *Rubu' Mujayyab*)

c. Implementasi tampilan utama *software Universal Astrolabe*

Tampilan utama *software Universal Astrolabe* berisi gambar desain *Universal Astrolabe* beserta seluruh komponennya yang dapat digerakkan seperti instrument bentuk fisiknya. Selain itu, beberapa tombol ditambahkan untuk menampilkan kurva atau menyembunyikannya dengan tujuan agar pengoperasian *Universal Astrolabe* lebih mudah dan pembacaan skala atau hasil perhitungannya tidak terlalu kompleks karena beberapa kurva dapat disembunyikan. Untuk membuat tampilan utama *software Universal Astrolabe*, peneliti menuliskan *Source code* pada file berekstensi .xml.

Berikut ini *source code* tampilan utama *software*:

```

onSelfEvent (load){
    _parent.background._rotation
    =180;
    _parent.background2._visible=
    0;
    _parent.alidade._visible=0;
    "Univ".depth=12;
    depth++;
}
on (press,keyPress("p")) {
    all.plate._rotation
    +=0.25 ;
    isDragging=true;
}
on (press,keyPress("o")) {
    all.plate._rotation -
    =0.25 ;
    isDragging=true;
}
on (press,keyPress("t")) {
    all.rete._rotation +=0.25
;
    all.az_ruler._rotation
    +=0.25 ;
    isDragging=true;
}
on (press,keyPress("T")) {
    all.rete._rotation +=0.25
;
    all.az_ruler._rotation
    +=0.25 ;
    isDragging=true;
}
on (press,keyPress("r")) {
    all.rete._rotation -=0.25
;
    all.az_ruler._rotation -
    =0.25 ;
    isDragging=true;
}
on (press,keyPress("R")) {
    all.rete._rotation -=0.25
;
    all.az_ruler._rotation -
    =0.25 ;
    isDragging=true;
}
on (press,keyPress("l")) {
    all._rotation +=0.25 ;
    isDragging=true;
}
on (press,keyPress("L")) {
    all._rotation +=0.25 ;
    isDragging=true;
}
on (press,keyPress("k")) {
    all._rotation -=0.25 ;
    isDragging=true;
}
on (press,keyPress("K")) {
    all._rotation -=0.25 ;
    isDragging=true;
}
on (press,keyPress("m")) {
    all.ruler._rotation+=180;
    isDragging=true;
}

```

```

    }
    on (press, keyPress("M")) {
all.ruler._rotation+=180;

        isDragging=true;
    }
    on (press, keyPress("n")) {
        all.rete._rotation +=180 ;
        all.az_ruler._rotation +=180
;
        isDragging=true;
    }
    on (press, keyPress("N")) {
        all.rete._rotation +=180 ;
        all.az_ruler._rotation +=180
;
        isDragging=true;
    }
    on (press, keyPress("d")) {
        txtdek._visible =1;
    }
    on (press, keyPress("D")) {
        txtdek._visible =1;
    }
    on (press, keyPress("B")) {
        txtbjur._visible =1;
    }
    on (press, keyPress("b")) {
        txtbjur._visible =1;
    }
        on (press, keyPress("h")) {
        txtdek._visible =0;
        txtbjur._visible =0;
    }
        on (press, keyPress("H")) {
        txtdek._visible =0;
        txtbjur._visible =0;
    }
    on (press, keyPress("W")) {
        all.ruler._rotation +=0.25 ;
        isDragging=true;
    }
    on (press, keyPress("w")) {
        all.ruler._rotation +=0.25 ;
        isDragging=true;
    }
    on (press, keyPress("q")) {
        all.ruler._rotation -=0.25 ;
        isDragging=true;
    }
    on (press, keyPress("Q")) {
        all.ruler._rotation -=0.25 ;
        isDragging=true;
    }
}

    on (press, keyPress("a")) {
        alidade._rotation -=0.25 ;
        isDragging=true;
    }
    on (press, keyPress("A")) {
        alidade._rotation -=0.25 ;
        isDragging=true;
    }
    on (press, keyPress("s")) {
        alidade._rotation +=0.25 ;
        isDragging=true;
    }
    on (press, keyPress("S")) {
        alidade._rotation +=0.25 ;
        isDragging=true;
    }
    onSelfEvent (load) {

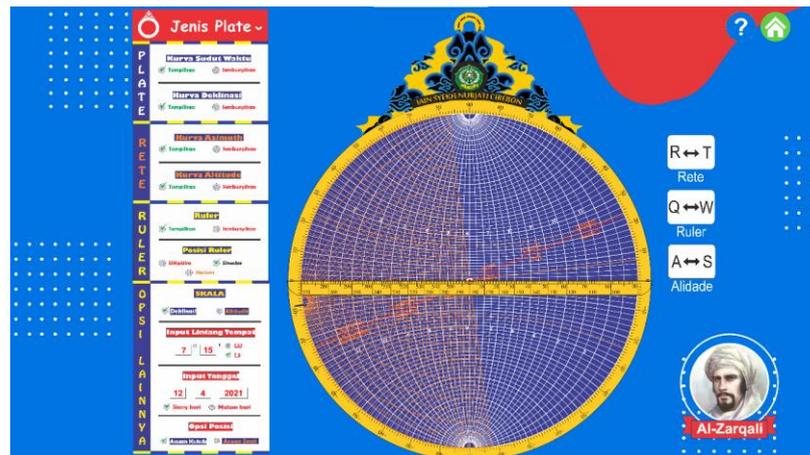
        _parent.all.az_ruler._visible
=0;

    }
    onSelfEvent (load){
        this._visible =0;

    }
    on (changed) {
        b = int(bln);
        t = int(tgl);
        if (b>12){b=b%12;}else{b=b;}
        if (b ==1){jmlh =0;}
        if (b ==2){jmlh =31;}
        if (b ==3){jmlh =59;}
        if (b ==4){jmlh =90;}
        if (b ==5){jmlh =120;}
        if (b ==6){jmlh =151;}
        if (b ==7){jmlh =181;}
        if (b ==8){jmlh =212;}
        if (b ==9){jmlh =243;}
        if (b ==10){jmlh =273;}
        if (b ==11){jmlh =304;}
        if (b ==12){jmlh =334;}
        if (b ==0){jmlh =334;}
        n = 6574.5 +jmlh+t;
        L =
(280.46+0.9856474*n)%360;
        g =
(357.528+0.9856003*n)%360;
        Long
=(L+1.915*Math.sin(g*Math.PI/
180)+0.02*Math.sin((2*g)*Math
.PI/180))%360;
        alidade._rotation = -Long;
    }
}

```

Berdasarkan *source code* di atas, tampilan utama *software* berisi desain komponen *Universal Astrolabe* dapat diputar dengan pergerakan terkecil $0,25^{\circ}$ atau $0^{\circ}15'$. Selain itu, *Ruler* pada *Universal Astrolabe* dapat ditempatkan secara otomatis pada horizon, equator langit, dan ekliptika dengan mengklik tombol penempatan *Ruler* pada menu tampilan. Ada pula tombol yang digunakan untuk menampilkan bagian depan dan belakang *Universal Astrolabe*. Adapun hasil implementasi tampilan utama *software Universal Astrolabe* adalah sebagai berikut:



Gambar 10. Tampilan utama *software Universal Astrolabe*

Proses implementasi rancangan *software Universal Astrolabe* dengan penulisan *source code* menghasilkan *software Universal Astrolabe* yang dapat dijalankan pada PC/Laptop ber-OS Windows. Berikut ini berbagai perbandingan fitur *software Universal Astrolabe* Berbasis JavaScript dengan instrument berwujud fisiknya:

Tabel 1. Perbandingan fitur *software Universal Astrolabe* Berbasis JavaScript dengan instrument berwujud fisiknya

No	Fitur	<i>Software Universal Astrolabe</i>	Instrumen <i>Universal Astrolabe</i>
1	Dapat ditampilkan melalui proyektor/LCD/Infocus	✓	✗
2	Masing-masing bagian instrumen dapat dipisah/dibongkar untuk mempermudah penggunaan	✓	✗
3	Dapat dioperasikan untuk komputasi	✓	✓
4	Dapat dioperasikan untuk observasi	✗	✓
5	Otomatisasi pemosisian <i>Rete</i> berdasarkan lintang tempat	✓	✗

Berdasarkan tabel 1 yang disajikan di atas, terdapat beberapa perbedaan antara *software Universal Astrolabe* berbasis JavaScript dengan instrument berwujud fisiknya. *Software Universal Astrolabe* berbasis JavaScript menawarkan keunggulan dalam hal fleksibilitas, otomatisasi, dan kemudahan presentasi dibandingkan dengan instrumen fisiknya. Kelebihan ini membuat *software* lebih cocok untuk tujuan edukasi, presentasi, dan simulasi. Sementara itu, instrument fisik lebih bernilai sebagai alat untuk pengalaman observasi secara langsung, meskipun memiliki keterbatasan fungsionalitas

dalam hal presentasi yaitu karena kerumitan tampilan kurva pada bagian depan *Universal Astrolabe*.

6. Uji Akurasi Penggunaan *software Universal Astrolabe* dalam Hisab Arah Kiblat

Pengujian dilakukan dengan membandingkan hasil perhitungan arah kiblat menggunakan *software Universal Astrolabe* dan kalkulator saintifik dengan rumus segitiga bola untuk perhitungan arah qiblat. Lokasi pengujian dipilih di 3 titik berbeda (Cirebon, Kuala Lumpur, dan Istanbul) untuk memvalidasi konsistensi hasil. Berikut ini adalah hasil perbandingan antara nilai azimuth Kiblat yang dihitung menggunakan *software Universal Astrolabe* dan kalkulator:

Tabel 2. hasil perbandingan antara nilai azimuth Kiblat yang dihitung menggunakan *software Universal Astrolabe* dan kalkulator

Cirebon (LT: 6°43' LS BT: 108°34' BT)		
<i>Universal Astrolabe</i>	Kalkulator	Selisih
294°50'	294°52'23"	0°2'23"
Kuala Lumpur (LT: 03°08'52" LU BT: 101°41'43" BT)		
<i>Universal Astrolabe</i>	Kalkulator	Selisih
292°30'	292°31'56"	0°1'56"
Istanbul (LT: 41°00'49" LU BT: 28°57'18"BT)		
<i>Universal Astrolabe</i>	Kalkulator	Selisih
151°30'	151°34'20"	0°4'20"

Hasil pengujian perhitungan azimuth Kiblat menggunakan *software Universal Astrolabe* dan kalkulator saintifik dengan rumus segitiga bola menunjukkan bahwa *software* ini memiliki tingkat akurasi yang tinggi. Hal ini terlihat dari selisih hasil perhitungan antara *software Universal Astrolabe* dan kalkulator saintifik yang sangat kecil, yaitu hanya berkisar antara 0°1'56" hingga 0°4'20". Perbedaan tersebut terjadi karena dua faktor utama:

- Besar sudut perputaran Rete terkecil dalam *software* adalah 0°15', sehingga pembacaan nilai azimuth terbatas pada kelipatan sudut tersebut.
- Skala altitude dan azimuth pada *software* menggunakan interval 1°, yang mengharuskan adanya penaksiran nilai ke kelipatan terdekat.

Kedua faktor ini menyebabkan hasil perhitungan *software* merupakan pendekatan dari nilai sebenarnya, meskipun selisihnya tetap dalam rentang yang dapat diterima untuk kebanyakan aplikasi praktis.

KESIMPULAN

Berdasarkan penelitian dan pengembangan *software Universal Astrolabe* berbasis JavaScript, dapat disimpulkan hal-hal berikut:

- Software Universal Astrolabe* berhasil dikembangkan sebagai alat digital yang memadukan khazanah ilmu falak klasik dengan teknologi modern. *Software* ini mampu meniru fungsi

- astrolabe fisik, termasuk perhitungan arah Kiblat, dengan antarmuka yang interaktif dan mudah digunakan.
2. *Software Universal Astrolabe* memiliki kelebihan dibandingkan instrumen *Universal Astrolabe* fisik dalam hal fleksibilitas karena komponen *Universal Astrolabe* dapat dipisahkan secara virtual (*show* dan *hide*) untuk memudahkan pembelajaran dan pembacaan skala. Selain itu, *software Universal Astrolabe* juga memiliki fitur otomasi pemosisian *Rete* berdasarkan lintang tempat dan aksesibilitas yang baik karena dapat ditampilkan melalui proyektor atau perangkat digital, yang mendukung edukasi dan presentasi. Namun memiliki kelemahan karena tidak dapat digunakan sebagai alat observasi benda langit secara langsung.
 3. Hasil uji akurasi perhitungan azimuth Kiblat menunjukkan selisih sangat kecil ($0^{\circ}1'56''$ hingga $0^{\circ}4'20''$) dibandingkan perhitungan manual menggunakan rumus segitiga bola. Selisih ini disebabkan oleh pembulatan sudut terkecil perputaran *Rete* sebesar $0^{\circ}15'$ dan interval 1° pada skala altitude dan azimuth, namun tetap dalam batas toleransi yang dapat diterima untuk aplikasi praktis.

DAFTAR PUSTAKA

- Berggren, J.L. *Episodes in the Mathematics of Medieval Islam*. New York: Springer, 1986.
- Borg, Walter R., & Gall, Meredith D. *Educational Research: An Introduction, 7th ed.* New York: Longman, 2003.
- Chabás, José, dan Bernard R. Goldstein. *The Alfonsine Tables of Toledo*. Dordrecht: Springer, 2003.
- Comes, Mercè. "The Legacy of al-Zarqālī in the European Renaissance." Dalam *Science in the Medieval Hebrew and Arabic Traditions*, disunting oleh Gad Freudenthal, 237–260. Aldershot: Ashgate, 2005.
- Crockford, Douglas. *JavaScript: The Good Parts*. Sebastopol: O'Reilly Media, 2008.
- ECMA International. ECMAScript® 2023 Language Specification, 14th Edition. Juni 2023. <https://262.ecma-international.org/14.0/>.
- Flanagan, David. *JavaScript: The Definitive Guide, 7th ed.* Sebastopol: O'Reilly Media, 2020.
- Hill, Donald R. *Islamic Science and Engineering*. Edinburgh: Edinburgh University Press, 1993.
- King, David A. *Astronomy in the Service of Islam*. Aldershot: Variorum, 1993.
- King, David A. *In Synchrony with the Heavens: Studies in Astronomical Timekeeping and Instrumentation in Medieval Islamic Civilization*. Leiden: Brill, 2004.
- MDN Web Docs. "JavaScript." Mozilla Developer Network, 2023. <https://developer.mozilla.org/en-US/docs/Web/JavaScript> (diakses 13 April 2024).
- Morrison, James E. *The Astrolabe*. London: Reichelt & Sons, 2007.

- Neugebauer, Otto. *A History of Ancient Mathematical Astronomy*. Berlin: Springer, 1975.
- North, John. *The Norton History of Astronomy and Cosmology*. New York: W.W. Norton & Company, 1995.
- Pouille, Emmanuel. *Les Instruments Astronomiques du Moyen Âge*. Paris: Brieux, 1983.
- Saliba, George. *A History of Arabic Astronomy: Planetary Theories During the Golden Age of Islam*. New York: New York University Press, 1994.
- Samsó, Julio. *Islamic Astronomy and Medieval Spain*. Aldershot: Variorum, 1994.
- Sugiyono. *Metode Penelitian Kuantitatif, Kualitatif, dan R&D*. Bandung: Alfabeta, 2013.
- Turner, Gerard L'E. *Astrolabes and Astrolabe Related Instruments*. London: Sotheby's Publications, 1985.
- W3Schools. "JavaScript Introduction."
2023. https://www.w3schools.com/js/js_intro.asp (diakses 13 April 2024).