

JavaScript-Based Digital Universal Astrolabe and Its Accuracy Test in Qibla Direction Calculation

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

^{1,2,3}UIN Siber Syekh Nurjati Cirebon

(Jl. Perjuangan, Sunyaragi, Kec. Kesambi, Kota Cirebon, Jawa Barat 45132)

*Email: ms_nahwandi@syekhnurjati.ac.id

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 ('Ilm al-Falak).

Article Info

Received:

03 Agustus 2025

Revised:

21 Oktober 2025

Accepted:

30 Desember 2025

Published:

31 Desember 2025

Keywords: *Astrolabe, Digitization, JavaScript, Qibla, Accuracy.*

A. Introduction

The astrolabe is an astronomical instrument that has been used since ancient times to measure the positions of stars and planets, as well as to determine time. The astrolabe has been used in many cultures and can be considered one of the most important navigational and astronomical tools of its time.[1] Although no longer widely used in modern astronomical observations, the Astrolabe is still valued as a historic tool and has educational value in understanding the development of science and navigation.[2] The use of the Astrolabe as a learning tool allows the integration of various scientific concepts and practical skills. Therefore, these tools not only have historical and astronomical value, but can also make a great contribution to the development of the knowledge and skills of learners or their users.

The astrolabe has significant educational value because it covers various aspects of knowledge, including astronomy, mathematics, and navigation.[3] So in the current era of technological development, Astrolabe, which represents the achievements of technology and knowledge in the past, needs to be developed into software or applications to continue to exist in the present. This is not new because many researchers in the field of Falak science have developed classical Falak calculation software such as the Digital Falak application by Ahmad Tholhah which can be used to calculate prayer times, qibla directions and istiwa' clocks with various classical and modern falak calculation methods. However, most of the Falak science software that has been designed, has a main function to help Falak practitioners in the field in providing data practically so that it lacks educational value. However, if Astrolabe, which has high educational value, is designed into a software, then the Astrolabe digitization product is expected to be able to educate its users about the treasures of Falak science in the past which have been combined with technological advances.

B. Method

This research is a research using the Research and Development method . Research and Development methods are research methods used to produce a particular product and test the effectiveness of that product. [4] Research and Development (R&D) is a process or steps to develop a new product or improve an existing product, which can be accounted for.[5] In this study, researchers developed the Astrolabe software with the JavaScript programming language. The data collection technique used in this study is a documentation method to obtain theories, how to make, and use Astrolabe from primary sources. The results of the primary

data documentation are used as the basis for obtaining product designs that will be developed into JavaScript-based software.

The data analysis technique used in this study is descriptive analytical to describe a descriptive understanding of the construction and application of the Astrolabe. Then a development process was carried out from the interpretation of the concept and needs in making the Astrolabe software. The researcher also uses software functionality tests and results accuracy tests to ensure that all features in the designed software or applications can run normally and have accuracy in accordance with the scale used in making the Astrolabe design.

C. Results and Discussion

1. Universal Definition of *Astrolabe*

The Universal Astrolabe is a type of *Astrolabe* that can be used for all locations on Earth, for latitudes ranging from 0° to 90° . *The Universal Astrolabe contains curves* resulting from horizontal coordinate projections that can be positioned and adjusted to the Equatorial coordinate curves so that the projection of the coordinates of the celestial sphere at all latitudes can be depicted on the two-dimensional plane of the *Universal Astrolabe plate*. 2]

The Universal Astrolabe was first designed by Abu Ishaq Ibrahim Ibn Yahya al-Zarqali, a Muslim mathematician and astronomer from Toledo, Andalusia (Spain). Al-Zarqali used stereographic projection to project the coordinates of the celestial sphere with the celestial poles, the *solstice* and *the equinox*. [6] The result of the projection is a plate containing the projection curves, declination circles and clockwise circles of celestial bodies with the position of the celestial poles just above and below. The projected slab designed by al-Zarqali was later called *Safihah al-Zarqali (The Saphea Arzachelis)*.

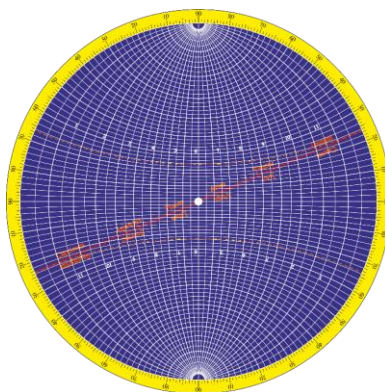


Figure 1. Design of the front of the *Universal Astrolabe* Al-Zarqali

The Universal Astrolabe designed by Al-Zarqali and his work in the field of astronomy helped in the development of astronomical tools and more precise time measurements of the time. His innovations not only had an impact on the Islamic world but also influenced the development of science and technology around the world. Some of the improvements given by Al-Zarqali to the *Astrolabe* include:

a. Additional Scale and Information:

Al-Zarqali improved *the Astrolabe* by adding additional scales and information that allowed for more accurate measurements and various astronomical calculations. [7]

b. Design Simplification:

Al-Zarqali designed *the Astrolabe* with a simpler design, making it easier to use and interpret, especially for scientists and navigators of his time. [7]

c. Time Measurement Precision:

Al-Zarqali also improved the instruments in the *Astrolabe* to improve the accuracy of time measurement and ease of navigation.[7]

d. Improvements in Astronomical Tables:

In addition to improvements to *the Astrolabe*, Al-Zarqali made more accurate astronomical tables, such as *al-Zij al-Mumtahan*, which helped in more precise astronomical calculations. (Sec. 8)

2. Universal Parts of the Astrolabe

a. *Mater* or *Front* (front)

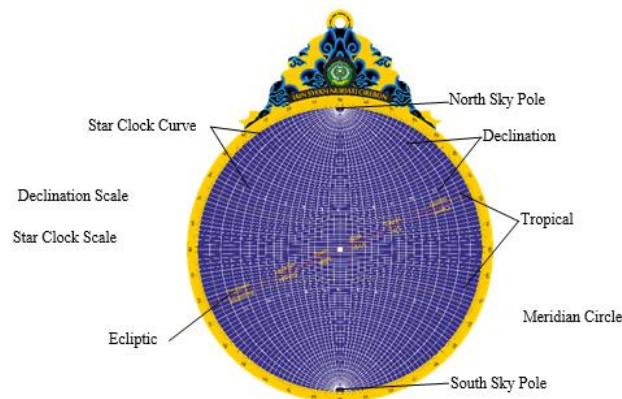


Figure 2. Components of the front of the *Universal Astrolabe* Al-Zarqali

1) North and South Sky Poles

The North and South celestial poles are projections of the Earth's North Poles and South Poles on the celestial sphere. In *the Al-Zarqali Universal Astrolabe*, the North pole is at the top while the South sky pole is at the bottom. [3]

2) Declination Curve

Declination is the distance of a celestial body calculated from the celestial equator. If the position of the celestial body is north of the celestial equator, then the celestial body has a north declination and has a positive value. Meanwhile, if the position of the celestial body is south of the celestial equator, then the celestial body has a southern declination and has a negative value.

In the *Universal Astrolabe* designed by Al-Zarqali, declination curves are horizontal transverse curves used to illustrate the daily movement of celestial bodies at each declination value.

3) Declination Scale

The declination scale is used to calculate the declination value of celestial bodies projected on *the Universal Astrolabe* designed by Al-Zarqali. The north and south celestial poles each have a declination scale value of 90° while the celestial equator has a declination scale value of 0° [6].

4) Equator Sky

The celestial equator is the Earth's Equator projected on the celestial sphere. In *the Universal Astrolabe* designed by al-Zarqali, the celestial equator is projected as a horizontal line dividing *the Mater* into two equally large parts. The celestial equator has a declination value scale of 0° .

5) Meridian Circle

The meridian circle is a special circle on the celestial sphere that connects the North-South Celestial Pole, the celestial equator, the north-south point, and the Zenith-Nadir point. In Al-Zarqali's *Universal Astrolabe*, the meridian circle is projected as an edge circle on *the Mater*.

6) Star Clock Curve

The Hour *Angle* is the distance of a celestial body from the meridian circle. It is called a star clock because by calculating the distance of a celestial body from the meridian circle, it can be known the angular value of time and the exact clock of the celestial body's position. In Al-Zarqali's Universal Astrolabe, the curves of the star clock are projected as vertical curves connecting the North and South celestial poles. 3]

7) Star Clock Scale

The star clock scale is used to calculate the *hour angle* of a celestial body projected on the *Universal Astrolabe* designed by Al-Zarqali. The star clock scale is expressed in hours to make it easier to find the exact time based on the position of a celestial body.

8) Tropical Curve

The Tropic Circle is the Sun's daily apparent circle of motion when the Sun is at the solstice or the north and south solstice. There are two types of Tropical Circles, namely the Cancer Tropic Circle and the Capricorn Tropic Circle. The Tropical Circle of Cancer is also referred to as the Sun's daily apparent circle of motion when the Sun is at the *point of the Summer Solstice*. Meanwhile, the Capricorn Tropical Circle is also called the Sun's daily apparent motion circle when the Sun is at the *Winter Solstice*.

In the *Universal Astrolabe* designed by Al-Zarqali, the Tropical Circle is projected as two curves with a declination scale value of $23^{\circ}26'12''$ which is equal to the *obliquity* or angle of ecliptic tilt of the celestial equator.

9) The Ecliptic

The ecliptic is the plane of the Earth's orbit around the Sun. In the Earth-oriented (geocentric) celestial sphere, the ecliptic is projected as the Sun's annual apparent circle of motion. During the annual apparent movement (with reference to the Earth), the Sun is seen in the constellations. The constellations that are in the ecliptic circle are called the Zodiac.

In the *Al-Zarqali Design Universal Astrolabe*, the Ecliptic is projected as a horizontal line that has a slope of $23^{\circ}26'12''$ from the celestial equator. On the ecliptic line, there is a zodiac scale that can be used to determine the position of the Sun based on the value of *the al-Shams* or the longitude of the Sun.

b. Rent

On the *Universal Astrolabe* designed by Al-Zarqali, the *Rete* is a rotating front *Astrolabe* plate. Rete contains curves resulting from the projection of the horizontal coordinate system of the celestial sphere which contains *the curves of the almucantras* and *Azimuth*.

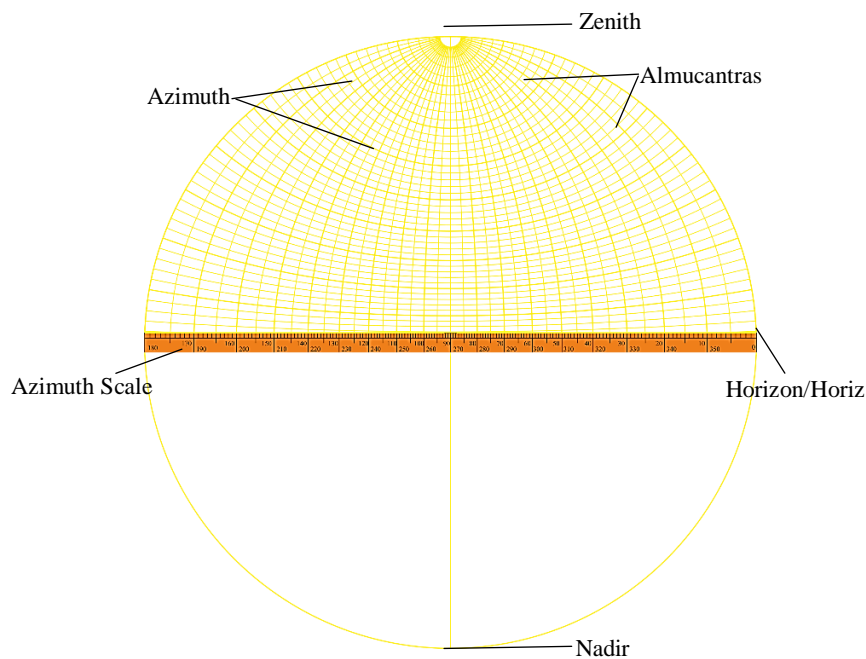


Figure 3. Components of the *Al-Zarqali Astrolabe Universal Rete*

1) Zenith

Zenith is the peak point or overhead of the observer. In the *Universal Astrolabe* designed by Al-Zarqali, Zenith is the peak point or the top point of *Rete*. The positioning of Zenith on the declination scale must be precise according to the latitude value of the place. If the *al-Zarqali Universal Astrolabe* is used for 6° LU, then Zenith is positioned to be at a declination scale value of 6° north (upper) of the celestial equator. Whereas if the *Al-Zarqali Universal Astrolabe* is used for 6° LS, then Zenith is positioned at the value of the declination scale of 6° South (below) the celestial equator. 6]

2) Nadir

Nadir of the point below the observer's foot. The Nadir point is always 180° from the Zenith point and -90° from the Horizon.

3) Horizon/Horizon Line

The horizon is the boundary between the sky and the sea (land) that the observer sees. In the *Universal Astrolabe* designed by Al-Zarqali, the Horizon is projected as the center line of *the Rete* which divides the *Rete* into two equally large parts (top and bottom). On the Horizon/Horizontal line, there is usually a ruler containing the *Azimuth scale*[6].

4) *Almucantras*

Almucantras are horizontal transverse curves on the *Rete Universal Astrolabe* that are used to project the position of celestial bodies based on their height from the horizon. Or it can also be said that *Almucantras* are horizontal curves on the *Rete Universal Astrolabe* that are used to calculate the height of celestial bodies from the horizon.

5) *Azimuth*

The azimuth is the arc on the horizon circle calculated from the North point clockwise to the position of the celestial body. In *al-Zarqali's Universal Astrolabe*, *the Azimuth* is projected as the vertical curves of the *Rete* that are used to project the position of celestial bodies based on their Azimuth. Or it can also be said that *Azimuth* is the vertical curves on the *Rete Universal Astrolabe* that are used to calculate the Azimuth of celestial bodies.

6) *Azimuth Scale*

The *Azimuth scale* is on the Horizon/Horizontal line which is used to determine the *Azimuth value* of a celestial body. The *Azimuth scale* is on the Horizon/Horizontal line. The value of 0° *Azimuth scale* is on the meridian circle.

7) *Stuart O'Neill*

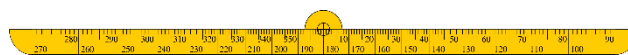


Figure 4. *Universal Ruler Astrolabe Al-Zarqali*

A *ruler* is a ruler located in the *Mater* or front of the *Universal Astrolabe* that can be rotated and used to calculate the value of the celestial body's time angle.

c. *Back*

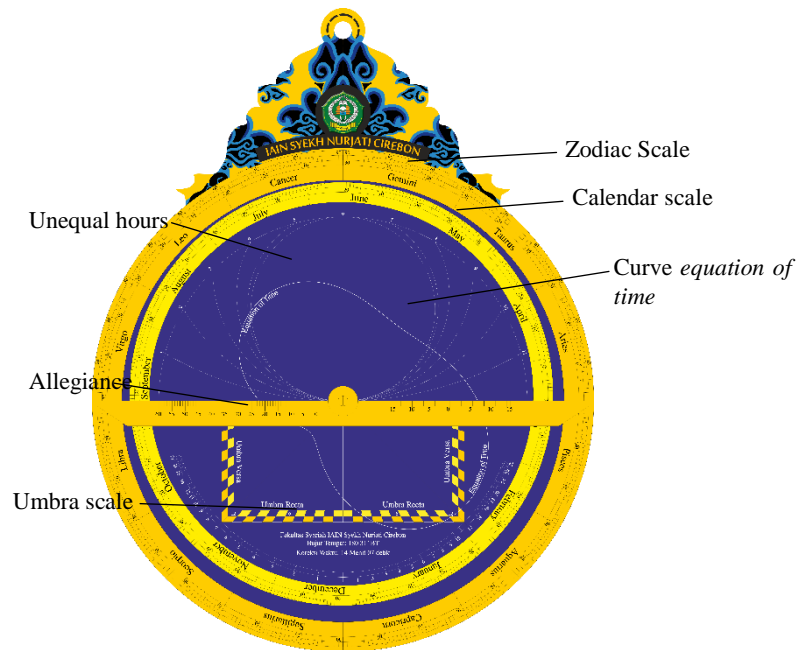


Figure 5. Rear components of *the Universal Astrolabe Al-Zarqali*

1) Zodiac Scale

The Zodiac Scale is a scale used to determine the position of the Sun on the ecliptic. There are 12 Zodiac signs that are on the zodiac scale. Each zodiac on the zodiac scale has a large angle of 30° .

2) Calendar Scale

The Calendar Scale contains the dates and months in the Christian calendar. By positioning *Alidade* on the desired date, it can be known the position of the Sun on that date based on the zodiac on the ecliptic circle.

3) Curve Equation of Time

The Equation of Time is the difference between the actual time (the Sun's time) and the time system used today. In *the Universal Astrolabe*, the *Equation of Time* curve is a curve used to determine the value of *the Equation of Time* on each AD date. By positioning *Alidade* on the desired date and looking at the position of *Alidade* on the *Equation of Time* curve, the value of *the Equation of Time* on that date can be known directly.

4) Unequal Hours

Unequal Hours is a time system used in medieval Rome. At present, *Unequal Hours* is no longer generally used. However, some still use it for the calculation of the worship time of Orthodox Christians.

5) Umbra Scale

The *Umbra* scale or shadow scale is a scale used to calculate the distance or height of an object on Earth. Generally, *the Umbra* scale uses the base number 12.

6) Allegiance

Alidade is the ruler behind *the Astrolabe*. *Alidade* is used to get the Sun's longitude and *Equation of Time* values based on dates. In *Alidade* there is an *Equation of Time* scale which is used to read the value of *the Equation of Time* and the *Haueter* scale or the maximum altitude of the Sun to get time in the *Unequal Hours* system.

3. How to Use *the Universal Astrolabe* in the Qibla Direction Hisab

The steps for using *the Universal Astrolabe* Al-Zarqali in the direction of the Qibla are as follows:

- a. Use the front of the *Universal Astrolabe* and make sure the position of the zenith point on the *Rete* is correct according to the latitude of the place.
- b. Calculate the value of the difference between the longitude of the place and the longitude of the Kaaba (SB)
- c. Convert the value of SB in units of degrees to units of hours
- d. Mark the intersection between the altitude curve equal to the latitude of the Kaaba and the hour curve of the SB value
- e. See the position of the mark based on the azimuth curve. So that is the azimuth of the Qibla for the location.

Example of calculating the direction of the Qibla for Cirebon with Latitude of Place: 6^{043} , S and Longitude of 108^{034} E:

- a) Use the front of the *Universal Astrolabe* and make sure the position of the zenith point on the *Rete* is correct according to the latitude of place 6^{043} , LS

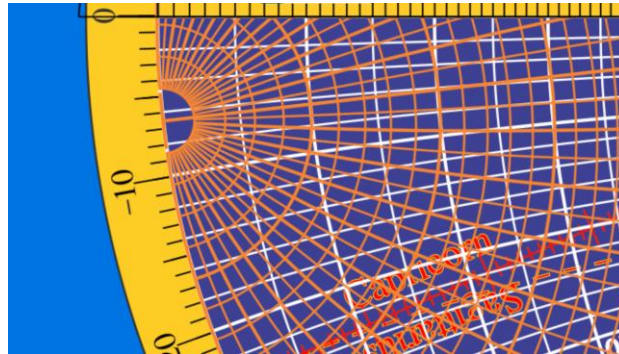


Figure 6. The position of the zenith point on *the Rete* is exactly according to the latitude of place $6^{043' LS}$

- b) Calculate the value of the difference between the longitude of the place and the longitude of the Kaaba (SB):

$$\begin{aligned} SB &= 108^{\circ}34' - 39^{\circ}49'34.36'' \\ &= 68^{\circ}44'25'' \end{aligned}$$

- c) Convert the value of SB in units of degrees to units of hours

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

- d) Mark the intersection between the altitude curve equal to the latitude of the Kaaba ($21^{025'21'' N}$) and the hour curve of the SB value

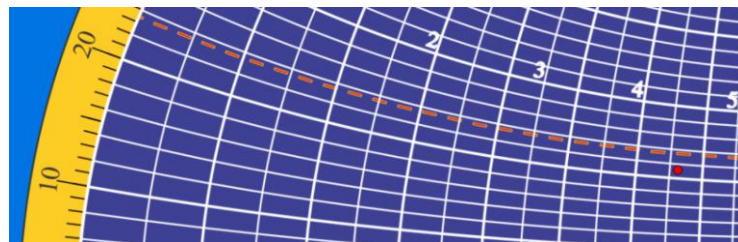


Figure 7. *The pointer* is positioned at the point of intersection between the altitude curve that is equal to the latitude of the Kaaba

- e) See the position of the mark based on the azimuth curve. So that's the azimuth of the Qibla for Cirebon, which is $294^{050'}$

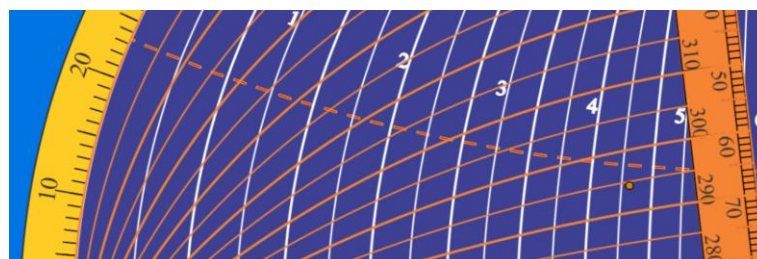


Figure 8. Reading of *pointer* position based on azimuth curve with the help of *azimuth scale ruler*

4. Overview of JavaScript-based programming

JavaScript (JS) is one of the main programming languages in web development, working alongside HTML and CSS. [8] In 2022, nearly 98% of websites are using client-side JS to monitor user interactions, with many relying on external *libraries*[9]. Every modern browser is equipped with a dedicated engine to run JS code directly on the user's device. As a high-level language that follows the ECMAScript standard, JS uses *just-in-time* compilation and has features such as *dynamic typing*, *prototype-based object-oriented programming*, and *first-class functions*. [10] The language is *multi-paradigm*, supporting *event-based*, *functional*, and *imperative programming*. [11]

Developers can leverage the JavaScript API to access a variety of features, including text manipulation, dates, regular expressions, data structures, and DOMs. While the ECMAScript standard does not include *built-in I/O* such as network or graphics operations, an execution environment such as a browser provides additional APIs for those purposes. Initially, JS only ran in the browser, but now it is an important part of the server-side through runtimes like Node.js. Despite their similar names, JavaScript and Java have fundamental differences in syntax and architecture, although they are often compared.

As a scripting language, JS runs in HTML documents and became a pioneer in the category of web scripting languages. It extends the capabilities of HTML by executing client-side commands, without the need for server-side processing. The execution of JS code depends on the browser processing the scripts embedded in HTML. Unlike Java, which requires a separate compiler, JS can be run immediately because *the engine* is already integrated into the browser[12].

There are two ways to write JavaScript code so that it can be displayed on an HTML page:

- a. JavaScript written on the same file

For writing in this way, the command used is

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

The command is usually placed between the `<BODY>... </BODY>`

Writing Example:

```
<HTML>
<HEAD><TITLE>.....</TITLE>
</HEAD>
<BODY>
<SCRIPT LANGUAGE="Javascript">
JavaScript code here
</SCRIPT>
HTML code here
</BODY>
</HTML>
```

b. JavaScript written in a separate file

We can also create Javascript code in a separate file with the aim that the HTML document is not too long. The attributes used are

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

Between the `<SCRIPT... >` and `<SCRIPT>` no more JavaScript code is required because they are already created in separate files. Files containing Javascript code with the extension *.js.

Broadly speaking, the architecture and components that build JavaScript-based scripting code are as follows:

a. Data Structure

All programming languages have built-in data structures, but the data structure model on one programming language with another is often different. The following is the data structure available in JavaScript:

1) Boolean

Boolean *data types* represent logical entities and have two values: `true` and `false`. Boolean values are commonly used for conditional operations, including logic operators: `if... else`, `while`, and more.

2) Number

Number type is a type of data in number format in both decimal numbers and integers. JavaScript is capable of storing *Number types* with numbers ranging from a minimum value of 2^{-1074} (`Number.MIN_VALUE`) to a maximum of 2^{1024} (`Number.MAX_VALUE`). While the number has a negative value between -2^{-1074} and -2^{1024} . However, *Numbers* in JavaScript can only

securely store integers and decimals in the range of numerical values valued between $-(2^{53} - 1)$ (Number.MIN_SAFE_INTEGER) to $2^{53} - 1$ (Number.MAX_SAFE_INTEGER).

3) BigInt

BigInt is a number-formatted data type in JavaScript that is rounding of decimal numbers with attention to decimal values after commas. Example:

```
const x = BigInt (2.6); x value = 3
const y = BigInt (2.3); Y value = 2
```

4) String

String types are textual data that cannot be processed with arithmetic operators. So that when data with *the String type* is written, it cannot be changed. The String data type that contains the letters must be written between two quotation marks. Examples of "this is String data" and "12".

5) Dates

Dates types are data that contains date, month, and year information that can be retrieved using the Date (function).

6) Array

An array is a type of structured data that is useful for storing a large amount of data of the same type. Usually an array is a data structure consisting of many variables with the same data type, where each variable element has an index value.

b. Variable

A variable is a symbolic name for a value in an application. Variable names, called value or data identifiers, are written using certain rules. Variable writing in JavaScript can also use most ISO 8859-1 or Unicode letters such as å and ü. Some of the variable writing in JavaScript are Number_hits, temp99, \$credit, and _name. As for declaring JavaScript variables, it can be done in two ways:

1. With the keyword var. Example: var x = 42. This syntax can be used to declare both local and global variables, depending on the context of the execution.
2. With the keywords const or let. Example: let y = 13. This syntax can be used to declare block-scoped local variables.

Variables should always be declared before use. Sometimes *Variable* writing in JavaScript is not declared directly by its value. This allows a *programmer* to collect various *variables* that will be used to store the results of the coding process in the application.

Example:

```
var name, age, city, province;
Name = "Syaoqi";
Age = 30;
City = "Cirebon";
Province = "Jawa_Barat";
```

5. Implementation of *Astrolabe's Universal Software Design Based on JavaScript*

The *Astrolabe instrument* is a treasure that needs to be preserved. In this *era of industry 4.0*, the classic instrument of astronomy needs to be designed into *software* that can be run on devices that are commonly used today, such as laptops. In this study, researchers developed a *JavaScript-based Universal Astrolabe software* that makes it easier for users to learn and operate *Universal Astrolabe* through a laptop with Windows OS. *The Universal Astrolabe software* is designed *to be easy to use*, where users can operate *the Universal Astrolabe* using a laptop keyboard. The following is the implementation of *the JavaScript-based Universal Astrolabe software design*:

a. IDE selection

In making this *software*, the researcher used *the Integrated Development Environment (IDE)* or development environment called SwishMax, which is a software that is specialized in creating animations and interactive multimedia applications. Using SwishMax, users can create Flash animations with a variety of visual and sound effects.

b. Software initial view implementation

The initial view *of the software* contains a list of options for the main menu *of the classic Astronomical Instrument Digitization software* consisting of *the Universal Astrolabe* and *Rubu' Mujayyab* menus. To create an initial appearance on the *software*, the researcher wrote the *source code* on the file with the .xml extension. The following *is the source code* of the software's initial menu display:

```
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._vis
ible =1;
Background_Aplikasi_png._visible
= 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=1;
}
}

Background_Aplikasi_wide_png._visible
=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);
}
    
```

Based on the *source code* above, the initial view of the software consists of 5 buttons that will lead to the *Universal Astrolabe software display*, the *Rubu' Mujayyab software display*, the user manual, the *fullscreen display*, and the button to close the software. The results of the implementation of the initial display of the software are as follows:



Figure 9. Display of software for Digitization of Classical Instruments of Astronomy (*Universal Astrolabe and Rubu' Mujayyab*)

c. Implementation of the *Universal Astrolabe software main display*

The main view of the *Universal Astrolabe software* contains a picture of the *Universal Astrolabe's* design and all its components that can be moved like a physical shape instrument. In addition, some buttons are added to show the curves or hide them with the aim of making the operation of the *Universal Astrolabe* easier and the reading of the scale or the results of its calculations is not too complex as some curves can be hidden. To create the main view of the *Universal Astrolabe software*, researchers write the *source code* on a file with the extension .xml. The following is the source code of the main view of the 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;
}

```

Based on the *source code* above, the main display of the software containing the design of the *Universal Astrolabe* components can be rotated with the smallest movement of 0.25° or $0^{015'}$. In addition, the *Ruler* on the *Universal Astrolabe* can be placed automatically on the horizon, sky equator, and ecliptic by clicking the *Ruler placement* button on the display menu. There are also buttons used to display the front and back of the *Universal Astrolabe*. The results of the implementation of the main display of the *Universal Astrolabe* software are as follows:

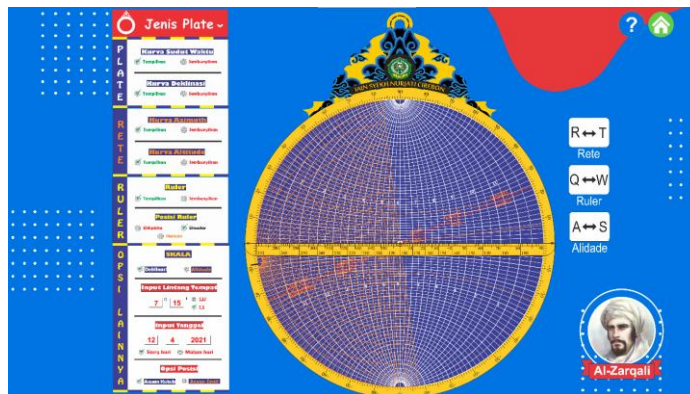


Figure 10. The main view of the *Universal Astrolabe* software

The process of implementing the *Universal Astrolabe* software design by writing *source code* produces *Universal Astrolabe* software that can be run on PC/Laptop with Windows OS. The following are various comparisons of the features of the *JavaScript-based Universal Astrolabe* software with its physical instruments:

Table 1. Comparison of the features of the *JavaScript-based Universal Astrolabe* software with its physical instruments

Yes	Features	<i>Software Universal Astrolabe</i>	<i>Astrolabe Universal Instrument</i>
1	Can be displayed via	☑	☒

	projector/LCD/Infocus		
2	Each instrument part can be separated/disassembled for ease of use	✓	✗
3	Operable for computing	✓	✓
4	Operable for observation	✗	✓
5	Automation of <i>Rete</i> positioning based on spot latitude	✓	✗

Based on table 1 presented above, there are several differences between the *JavaScript-based Universal Astrolabe software* and its physical instruments. *The JavaScript-based Universal Astrolabe software* offers advantages in terms of flexibility, automation, and ease of presentation compared to its physical instruments. These advantages make the *software* more suitable for educational, presentation, and simulation purposes. Meanwhile, physical instruments are more valuable as tools for direct observation experiences, although they have limited functionality in terms of presentation, namely due to the complexity of the curve display on the front of the *Universal Astrolabe*.

6. Test of the Accuracy of the Use of *Universal Astrolabe software* in Qibla Direction Hisab

The test was carried out by comparing the results of the calculation of the direction of the Qibla using *the Universal Astrolabe software* and scientific calculators with the formula of the spherical triangle for the calculation of the direction of the qiblah. Test locations were selected at 3 different points (Cirebon, Kuala Lumpur, and Istanbul) to validate the consistency of the results. The following is the result of a comparison between the azimuth value of the Qibla calculated using *the Universal Astrolabe software* and the calculator:

Table 2. Comparison between the Qibla azimuth value calculated using *the Universal Astrolabe software* and the calculator

Cirebon (LT: 6o43' LS BT: 108o34' E)		
<i>Universal Astrolabe</i>	Calculator	Differences
294o50'	294o52'23"	0o2'23"
Kuala Lumpur (LT: 03°08'52" N E: 101°41'43" E)		
<i>Universal Astrolabe</i>	Calculator	Differences
292o30'	292o31'56"	0o1'56"
Istanbul (LT: 41°00'49" N E.T.: 28°57'18"E)		
<i>Universal Astrolabe</i>	Calculator	Differences
151o30'	151o34'20"	0o4'20"

The results of the Qibla azimuth calculation test using *Universal Astrolabe software* and scientific calculators with the spherical triangle formula show that *this software* has a high level of accuracy. This can be seen from the difference in calculation results between the *Universal Astrolabe software* and the scientific calculator which is very small, which only ranges from $0^{\circ}1'56''$ to $0^{\circ}4'20''$. The difference occurs due to two main factors:

- a. The smallest Rete rotation angle in *software* is $0^{\circ}15'$, so the azimuth value reading is limited to multiples of that angle.
- b. The altitude and azimuth scales in the *software* use 1° intervals, which require estimating the value to the nearest multiple.

Both of these factors cause the *software* calculation to be an approximation of the true value, although the difference remains within the acceptable range for most practical applications.

D. Conclusion

Based on the research and development of *JavaScript-based Universal Astrolabe software*, the following can be concluded:

1. *The Universal Astrolabe software* has been successfully developed as a digital tool that combines the treasures of classical astronomy with modern technology. The software is able to mimic the functions of *the physical Astrolabe*, including the calculation of the direction of the Qibla, with an interactive and easy-to-use interface.
2. *Universal Astrolabe software* has an advantage over physical *Universal Astrolabe* instruments in terms of flexibility because *Universal Astrolabe* components can be virtually separated (*show* and *hide*) for easy learning and scale reading. In addition, *the Universal Astrolabe software* also has Rete's *positioning automation feature* based on the latitude of the place and good accessibility because it can be displayed through projectors or digital devices, which support education and presentation. However, it has the disadvantage of not being able to be used as a tool for direct observation of celestial bodies.
3. The results of the accuracy test of the Qibla azimuth calculation showed a very small difference ($0^{\circ}1'56''$ to $0^{\circ}4'20''$) compared to the manual calculation using the spherical triangle formula. This difference is due to the rounding of the smallest angle of *the Rete rotation* of $0^{\circ}15'$ and the interval of 1° on the altitude and azimuth scales, but it remains within acceptable tolerance limits for practical applications.

References

- [1] J. North, *The Norton History of Astronomy and Cosmology*. New York: W.W. Norton & Company, 1995.
- [2] D. A. King, *In Synchrony with the Heavens: Studies in Astronomical Timekeeping and*

Instrumentation in Medieval Islamic Civilization. Leiden: Brill, 2004.

- [3] D. A. King, *Astronomy in the Service of Islam*. Aldershot: Variorum, 1993.
- [4] W. R. Borg and M. D. Gall, *Educational Research: An Introduction*, 7th ed. New York: Longman, 2003.
- [5] Sugiyono, *Metode Penelitian Kuantitatif, Kualitatif, dan R&D*. Bandung: Alfabeta, 2013.
- [6] J. E. Morrison, *The Astrolabe*. London: Reichelt & Sons, 2007.
- [7] J. Samsó, *Islamic Astronomy and Medieval Spain*. Aldershot: Variorum, 1994.
- [8] J. Chabás and B. R. Goldstein, *The Alfonsine Tables of Toledo*. Dordrecht: Springer, 2003.
- [9] MDN Web Docs, "JavaScript," Mozilla Developer Network, 2023. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>. [Accessed: Apr. 13, 2024].
- [10] ECMA International, "ECMAScript® 2023 Language Specification," 14th ed., Jun. 2023. [Online]. Available: <https://262.ecma-international.org/14.0/>
- [11] D. Flanagan, *JavaScript: The Definitive Guide*, 7th ed. Sebastopol: O'Reilly Media, 2020.
- [12] W3Schools, "JavaScript Introduction," 2023. [Online]. Available: https://www.w3schools.com/js/js_intro.asp. [Accessed: Apr. 13, 2024]. Claude is AI and can make mistakes. Please double-check responses.