PERANCANGAN PROGRAM PERHITUNGAN DATA POSISI MATAHARI DAN BULAN BERBASIS MIKROKONTROLER

SKRIPSI

Diajukan untuk Memenuhi Tugas Dan Melengkapi Syarat Guna

Memperoleh

Gelar Sarjana Program Strata 1 (S1)

Dalam Ilmu Syari'ah dan Hukum

Oleh:

MUHAMMAD SAGISE AINUN NAIM

NIM: 1802046117

PRODI ILMU FALAK

FAKULTAS SYARI'AH DAN HUKUM

UNIVERSITAS ISLAM NEGERI WALISONGO

2022

MOTTO

ٱلشَّمۡسُ وَٱلۡقَمَرُ بِحُسۡبَانٖ ٥

"Matahari dan Bulan (beredar) menurut perhitungan"

(Q.S. Ar-Rahman : 5)

Dr. K.H. Akhmad Izzudin, M.Ag.
Jl. Raya Bukit Beringin BaratKav. C.131
Perumas Bukit Beringin Lestari, Ngaliyan, Semarang

PERSETUJUAN PEMBIMBING

Lamp. : 4 (empat) eks.
Hal : Naskah Skripsi
An. Sdr. Muhammad Sagise Ainun Naim

Kepada Yth.
Dekan Fakultas Syari'ah dan Hukum
UIN Walisongo Semarang
di Semarang

Assalamu'alaikum Wr. Wb.

Setelah saya mengoreksi dan mengadakan perbaikan seperlunya, bersama ini saya kirim naskah skripsi Saudari:

Nama : Muhammad Sagise Ainun Naim
NIM : 1802046117
Jurusan : Ilmu Falak
Judul Skripsi : **Perancangan Program Berbasis Mikrokontroler Untuk Perhitungan Data Posisi Matahari Dan Bulan**

Dengan ini saya mohon kiranya skripsi Saudari tersebut dapat segera dimunaqasyahkan. Demikian, harap menjadikan maklum.

Wassalamu'alaikum Wr. Wb.

Semarang, 26 Mei 2022
Pembimbing I,

Dr. K.H. Akhmad Izzudin, M.Ag.
NIP. 197205121999031003

Siti Rofiah, M.H
Bukit Beringin Lestari BVIII No. 205
Ngaliyan Semarang

PERSETUJUAN PEMBIMBING

Lamp. : 4 (empat) eks.
Hal : Naskah Skripsi
An. Sdr. Muhammad Sagise Ainun Naim

Kepada Yth.
Dekan Fakultas Syari'ah dan Hukum
UIN Walisongo Semarang
di Semarang

Assalamu'alaikum Wr. Wb.

Setelah saya mengoreksi dan mengadakan perbaikan seperlunya, bersama ini saya kirim naskah skripsi Saudari:

Nama : Muhammad Sagise Ainun Naim
NIM : 1802046117
Jurusan : Ilmu Falak
Judul Skripsi : **Perancangan Program Berbasis Mikrokontroler Untuk Perhitungan Data Posisi Matahari Dan Bulan**

Dengan ini saya mohon kiranya skripsi Saudari tersebut dapat segera dimunaqasyahkan. Demikian, harap menjadikan maklum.

Wassalamu'alaikum Wr. Wb.

Semarang, 26 Mei 2022
Pembimbing II,

Siti Rofiah, M.H
NIP. 198601062015032003

PERSEMBAHAN

***

“Saya persembahkan tulisan sederhana ini untuk

Ayah Bambang Sugiyarto S.Pd.I

Ibu Sahiroh

Adik Ahmad Apif Mahfud

Dan semua keluarga tercinta

Serta teman-teman yang terbaik”

***

KEMENTERIAN AGAMA
UNIVERSITAS ISLAM NEGERI (UIN) WALISONGO
FAKULTAS SYARI'AH DAN HUKUM
Alamat: Jl.Prof. Dr. HAMKA Kampus III Ngaliyan Telp. (024) 7601291 Semarang 50185

PENGESAHAN

Naskah skripsi Saudara,

Nama	:	Muhammad Sagise Ainun Naim
NIM	:	1802046117
Fakultas/Jurusan	:	Syari'ah dan Hukum/Ilmu Falak
Judul	:	**Perancangan Program Data Posisi Matahari Dan Bulan Berbasis Mikrokontroler**

Telah diujikan dalam sidang munaqosyah oleh Dewan Penguji Fakultas Syari'ah dan Hukum UIN Walisongo Semarang dan dinyatakan lulus dengan predikat **CUMLAUDE,** pada tanggal:

15 Juni 2022

dan dapat diterima sebagai syarat guna memperoleh gelar Sarjana Strata 1 tahun akademik 2021/2022.

Semarang, 19 Juni 2022

DEWAN PENGUJI

Ketua Sidang/Penguji I,

Hj. Nur Hidayati Setyani, SH.,MH.
NIP. 19670320199303 2 001

Sekretaris/Penguji II,

Dr. H. Ahmad Izzuddin, M.Ag
NIP. 19720512199903 1 003

Penguji III,

Dr. H. Agus Nurhadi, MA.
NIP. 19660407199103 1 004

Penguji IV,

Ahmad Syifaul Anam, S.H.I., M.H.
NIP. 198001202003121 001

Pembimbing I,

Dr. H. Ahmad Izzuddin, M.Ag
NIP. 19720512199903 1 003

Pembimbing II,

Siti Rofi'ah, M.H.
NIP. 19860106201503 2 003

DEKLARASI

Dengan penuh kejujuran dan tanggung jawab, penulis menyatakan bahwa skripsi ini tidak berisi materi yang telah pernah ditulis oleh orang lain atau diterbitkan. Demikian juga skripsi ini tidak berisi satu pun pikiran-pikiran orang lain, kecuali informasi yang terdapat dalam referensi yang dijadikan bahan rujukan dalam skripsi ini.

Semarang, 9 Juni 2022

Penulis

Muhammad Sagise Ainun Naim

NIM: 1802046117

Abstrak

Data posisi Matahari dan Bulan merupakan referensi dasar dalam pengkajian maupun untuk penerapan Ilmu Falak. Seiring dengan perkembangan teknologi, Ilmu Falak juga berkembang dengan pesat, terbukti ada banyak *software* / program penghitung data perhitungan data astronomis Matahari dan Bulan. Dari program tersebut masih dibutuhkan pengembangan program yang tidak hanya digunakan pada tahapan desain, namun sampai produk jadi. Berdasarkan permasalahan di atas penulis berinisiatif mengembangkan sebuah program perhitungan data posisi Matahari dan Bulan yang berbasis mikrokontroler yang dapat dikembangkan untuk instrumen falak berbasis elektro dan komputer dengan bantuan mikrokontroler yang bisa digunakan dalam aplikasi Ilmu Falak.

Penelitian ini dilakukan dengan menggunakan pendekatan R&D (*Research and Developmen*) atau penelitian pengembangan yang menghasilkan suatu produk. Penelitian ini menghasilkan dua hal: Pertama, penulis berhasil mengembangkan program berbasis mikrokontroler yang diberi nama *Orbita* yang dapat dijalankan pada mikrokontroler Arduino DUE. Kedua, dengan melakukan beberapa uji coba, diketahui bahwa program *Orbita* dapat memproses perhitungan dengan hasil baik dari tahun 2005 sampai tahun 3000. Selain itu, hasil dari komparasi program tersebut dengan program Daff Moon dan Stellarium hanya berselisih kurang dari 1 derajat, serta terdapat beberapa data pada Stellarium selisihnya tidak lebih dari 2 derajat. Sehingga, hasil perhitungan program ini dinilai dapat digunakan sebagai sumber data dalam perhitungan falak metode hisab untuk pengembangan instrumen falak berbasis elektro dan komputer.

Kata kunci: Data Matahari, Data Bulan, Mikrokontroler, Program Arduino

Abstract

The position of the Sun and Moon is a fundamental reference in both the study and practice of Astronomy. Along with technology advancements, astronomy is fast expanding; it has been demonstrated that there are numerous software programs for calculating astronomical data from the sun and moon. Software development is still required from the software, which is employed not only during the design stage but also until the finished product. Based on the issues raised above, the author decided to create a microcontroller-based program to calculate the position data of the Sun and Moon. The Arduino C++ programming language was used to create the application, which may be used to create electrical and computer-based astronomy instrumentation. Using a microcontroller that can be employed in Astronomy applications.

This research was conducted using an R&D (*Research and Development*) or development research that produces a product. This research resulted in two things: First, the author succeeded in developing microcontroller-based program called *Orbita* that can be run on Arduino DUE microcontroller. Second, by doing some trials, it is known that the *Orbita* program can process calculations with good results from 2005 until the year 3000. In addition, the results of the comparison of the program with the program Daff Moon and Stellarium are only less than 1 degree apart, and there are several data on Stellarium the difference is no more than 2 degrees. So, the calculation result This program is considered to be able to be used as a data source in astronomy calculations calculation method for the development of electro- and computer-based astronomical instruments

Keywords: Solar Data, Moon Data, Microcontroller, Arduino Program

KATA PENGANTAR

Alhamdulillah, puji syukur atas segala kenikmatan yang diberikan Allah swt kepada seluruh makhluknya terutama manusia yang bernaung di muka Bumi ini. Karena berkat Rahmat dan Karunia-Nyalah sehingga penulis dapat menyelesaikan skripsi yang berjudul **"Perancangan Program Berbasis Mikrokontroler Untuk Perhitungan Data Posisi Matahari Dan Bulan"** sebagai salah satu syarat menyelesaikan studi strata satu. *Allahumma Sholi'ala Sayyidina Muhammad,* shalawat dan salam semoga tetap tercurahkan kepada baginda Rasulullah Muhammad saw. Nabi yang telah diberikan wahyu dan mukjizat oleh Allah swt berupa al-Qur'an yang akan tetap terjaga hingga akhir zaman. Dialah teladan bagi seluruh umatnya serta Dialah pembawa risalah kebenaran dalam menuntun umatnya ke jalan keselamatan.

Penulis menyadari bahwa penelitian ini dapat diselesaikan tak luput dari berbagai pihak. Oleh karena itu, penulis menyampaikan banyak terima kasih kepada:

1. Dr. H. Ahmad Izuuddin, M.Ag. selaku Pembimbing I dan Siti Rofi'ah, M.H. selaku Pembimbing II yang telah meluangkan waktu, tenaga dan pikiran dengan sabar dan tulus ikhlas untuk memberikan bimbingan dan pengarahan dalam penyusunan skripsi ini.
2. Kedua orang tua dan segenap keluarga penulis, atas segala doa, perhatian, dukungan, dan curahan kasih sayangnya

yang sangat besar sekali, sehingga penulis mempunyai semangat untuk menyelesaikan skripsi ini.

3. Dekan Fakultas Syari'ah dan Hukum UIN Walisongo Semarang dan Wakil-wakil Dekan yang telah memberikan izin kepada penulis untuk menulis skripsi tersebut dan memberikan fasilitas untuk belajar dari awal hingga akhir.
4. Dosen-dosen Ilmu Falak Fakultas Syari'ah dan Hukum semoga ilmu yang diajarkan berkah dan bermanfaat bagi penulis.
5. Keluarga besar Pondok Pesantren Raudlatuth Thalibin Tugurejo. Terima kasih atas ilmu dan bimbingannya.
6. Seluruh guru penulis yang telah banyak memberikan ilmu dan pengetahuan serta didikan yang tak ternilai harganya.
7. Naviatul Hasanah, yang selalu memberikan semangat serta menjadi motivasi bagi penulis. Anda sepesial.
8. Teman-teman Signature, Lana, Ahsan, Imaddudin, Ade, Natiq, Faiq, Daryono, Ulum, Yasin, Ridwan, Reza, Fajar, Rian, Fahmi, Aam. Terima kasih atas kebersamaanya selama ini. Kalian istimewa.
9. Dan semua pihak yang tidak dapat penulis sebutkan satu persatu, yang secara langsung maupun tidak langsung selalu memberi bantuan, dorongan dan do'a kepada penulis selama melaksanakan studi di S1 Ilmu Falak UIN Walisongo Semarang.

Pada akhirnya penulis menyadari bahwa penulisan skripsi ini belum mencapai kesempurnaan dalam arti sebenarnya, untuk itu penulis mengharap saran dan kritik konstruktif dari pembaca demi kesempurnaan skripsi ini. Penulis berharap semoga skripsi ini dapat bermanfaat bagi penulis dan para pembaca.

PEDOMAN TRANSLITERASI

Pedoman transliterasi Arab-Latin yang digunakan merupakan hasil Surat Keputusan Bersama (SKB) Menteri Agama

No. 158 Tahun 1987 dan Menteri Pendidikan dan Kebudayaan R. I. No. 0543b/U/1987.

A. Konsonan

Daftar huruf bahasa Arab dan transliterasinya ke dalam huruf Latin dapat dilihat dalam tabel berikut:

Huruf Arab	Nama	Huruf Latin	Nama
ا	*Alif*	Tidak dilambangkan	Tidak dilambangkan
ب	*Ba*	B	Be
ت	*Ta*	T	Te
ث	*Sa*	Ṡ	Es (dengan titik di atas)
ج	*Jim*	J	Je
ح	*Ha*	Ḥ	Ha (dengan titik di bawah)
خ	*Kha*	Kh	Ka dan ha
د	*Da*	D	De
ذ	*Za*	Ż	Zet (dengan titik di atas)
ر	*Ra*	R	Er
ز	*Zal*	Z	Zet
س	*Sin*	S	Es
ش	*Syin*	Sy	Es dan ye

ص	*Sad*	Ṣ	Es (dengan titik di bawah)
ض	*Dad*	Ḍ	De(dengan titik di bawah)
ط	*Ta*	Ṭ	T(dengan titik di bawah)
ظ	*Za*	Ẓ	Z(dengan titik di bawah)
ع	*Ain*	‘_	Apostrof terbalik
غ	*Gain*	G	Ge
ف	*Fa*	F	Ef
ق	*Qaf*	Q	Qi
ك	*Kaf*	Kh	Ka Dan Ha
ل	*Lam*	L	El
م	*Mim*	M	Em
ن	*Nun*	N	En
و	*Wau*	W	We
ه	*Ha*	H	Ha
ء	*Hamzah*	_’	Apostrof
ي	*Ya*	Y	Ye

Hamzah (ء)yang terletak di awal kata mengikuti vokalnya tanpa diberi tanda apapun. Jika ia terletak di tengah atau di akhir, maka ditulis dengan tanda (’).

B. Vokal

Vokal bahasa Arab, seperti vokal dalam bahasa Indonesia, terdiri atas vokal tunggal dan vokal rangkap. Vokal tunggal bahasa Arab yang lambangnya berupa tanda harakat, transliterasinya sebagai berikut:

Tanda	Nama	Huruf Latin	Nama
◌َ	*Fathah*	A	A
◌ِ	*Kasrah*	I	I
◌ُ	*Dammah*	U	U

Vokal rangkap bahasa Arab yang lambangnya berupa gabungan antara harakat dan huruf, transliterasinya berupa gabungan huruf, yaitu:

Tanda	Nama	Huruf Latin	Nama
◌َيْ	*Fathah* dan *ya*	Ai	A dan I
◌َوْ	*Fathah* dan *wau*	Au	A dan U

C. *Maddah*

Maddah atau vokal panjang yang lambangnya berupa harakat dan huruf, transliterasinya berupa huruf dan tanda, yaitu:

Harakat dan Huruf	Nama	Huruf dan Tanda	Nama
ا.... ◌َ	*Fathah* dan *alif*	Ā	A dan garis di atas

ـِـي	*Kasrah* dan *ya*	Ī	I dan garis di atas
ـُـو	*Dammah* dan *wau*	Ū	U dan garis di atas

D. *Ta Marbūṭah*

Transliterasi untuk ta marbūṭah ada dua, yaitu: ta marbūṭah yang hidup atau memiliki harakat faṭhah, kasrah, atau ḍammah menggunakan transliterasi [t], sedangkan ta marbūṭah yang mati atau berharakat sukun menggunakan transliterasi [h].

E. *Syahaddah*

Syaddah atau tasydīd yang dalam penulisan Arab dilambangkan dengan tanda tasydīd (ّ), dalam transliterasi ini dilambangkan dengan pengulangan huruf (konsonan ganda) yang diberi tanda tasydīd. Jika huruf ya (ي)ber-tasydīd di akhir sebuah kata dan didahului harakat kasrah (ِ), maka ia ditransliterasi seperti huruf maddah (ī).

F. Kata sandang

Kata sandang dalam sistem tulisan Arab dilambangkan dengan huruf alif lam ma'arifah(ال)Dalam pedoman transliterasi ini, kata sandang ditransliterasi seperti biasa [al-], baik ketika diikuti oleh huruf syamsiah maupun huruf qamariah. Kata sandang ditulis terpisah dari kata yang mengikutinya dan dihubungkan dengan garis mendatar (-).

G. *Hamzah*

Hamzah Aturan transliterasi huruf hamzah menjadi apostrof (’) hanya berlaku bagi hamzah yang terletak di tengah dan akhir kata. Namun, bila hamzah terletak di awal kata, maka ia tidak dilambangkan, karena dalam tulisan Arab ia berupa alif.

H. Penulisan Kata Arab yang Lazim digunakan dalam Bahasa Indonesia

Penulisan Kata Arab yang Lazim digunakan dalam Bahasa Indonesia Kata, istilah, atau kalimat Arab yang ditransliterasi merupakan kata, istilah, atau kalimat yang belum dibakukan dalam bahasa Indonesia. Kata, istilah, atau kalimat yang sudah lazim dan menjadi bagian dari pembendaharaan bahasa Indonesia atau sudah sering ditulis dalam bahasa Indonesia tidak lagi ditulis menurut cara transliterasi ini. Namun, apabila kata, istilah, atau kalimat tersebut menjadi bagian dari satu rangkaian teks Arab, maka harus ditransliterasi secara utuh.

I. *Lafẓ al-Jalālah*

Lafẓ al-Jalālah (الله) Kata “Allah” yang didahului parikel seperti huruf jarr atau huruf lainnya atau berkedudukan sebagai muḍāf ilaih (frasa nominal), ditransliterasi tanpa huruf hamzah. Adapun ta marbūṭah di

akhir kata yang disandarkan pada lafẓ al-jalālah ditransliterasi dengan huruf [t].

J. Huruf Kapital

Huruf Kapital Walau sistem tulisan Arab tidak mengenal huruf kapital, dalam transliterasinya huruf-huruf tersebut dikenai ketentuan tentang penggunaan huruf kapital berdasarkan pedomaan ejaan bahasa Indonesia yang berlaku (EYD). Huruf kapital digunakan untuk menuliskan huruf awal nama, dan huruf pertama pada permulaan kalimat. Apabila kata nama tersebut diawali oleh kata sandang (al-), maka yang ditulis kapital adalah huruf awal nama tersebut, kata sandang ditulis kapital (Al-) apabila berada di awal kalimat.

DAFTAR ISI

HALAMAN JUDUL i
MOTTO ii
HALAMAN PERSEMBAHAN..... v

HALAMAN PENGESAHAN ..... **vi**
DEKLARASI ..... **vii**
ABSTRAK ..... **viii**
KATA PENGANTAR ..... **x**
PEDOMAN TRANSLITERASI ..... **xiii**
DAFTAR ISI ..... **xix**
BAB I PENDAHULUAN **1**
A. Latar Belakang 1
B. Rumusan Masalah 4
C. Tujuan Dan Manfaat Penelitian 5
D. Telaah Pustaka 6
E. Metode Penelitian 11
F. Sistematika Penulisan 17
BAB II TINJAUAN UMUM DATA POSISI MATAHARI DAN BULAN SERTA PENGERTIAN MIKROKONTROLER **22**
A. Tinjauan Data Posisi Matahari Dan Bulan 22
B. Mikrokontroler 45
BAB III PERANCANGAN BERBASIS MIKROKONTROLER UNTUK PERHITUNGAN POSISI MATAHARI DAN BULAN BERUPA PROGRAM *ORBITA* **53**
A. Proses Instalasi Mikrokontroler Arduino ke dalam Laptop 53

B. Penyusunan Program *Orbita* 59
BAB IV ANALISIS TERHADAP *ORBITA* SEBAGAI PROGRAM BERBASIS MIKROKONTROLER..... 64
A. Pengujian Program 64
B. Hasil Perancangan Program 77
BAB V PENUTUP 79
A. Kesimpulan 79
B. Saran-saran 82
DAFTAR PUSTAKA 83
LAMPIRAN-LAMPIRAN 89
DAFTAR RIWAYAT HIDUP 132

BAB I

PENDAHULUAN

A. Latar Belakang

Program pada mikrokontroler merupakan program yang tidak hanya menampilkan sebatas desain, namun dapat dikembangkan menjadi sebuah produk jadi.

Mikrokontroler adalah komputer *mini* yang menggunakan *chip* sebagai pengontrol rangkaian elektronik yang tersusun dari CPU (*Central Processing Unit*), memori, I/O (*input/output*), bahkan sudah dilengkapi dengan ADC(*Analog-to-Digital Converter*) yang sudah terintegrasi di dalamnya.[1] Mikrokontroler dapat dimanfaatkan sebagai penerima input dari beberapa sensor serta dapat mengontrol atau memberi output keperangkat lainnya.[2] Dari sekian banyaknya vendor mikrokontroler, Arduino menjadi yang paling digemari oleh pengguna mikrokontroler.[3] Dari penjelasan tersebut,

---

[1]Fajrullah, "Konstruksi Instrumen Moon Verificator Dan Tingkat Akurasinya Dalam Rukyatulhilal", *Skripsi,* (Semarang: Fakultas Syari'ah Dan Hukum Universitas Islam Negeri Walisongo, 2020), 6.

[2] Fajrullah, "Konstruksi Instrumen Moon Verificator Dan Tingkat Akurasinya Dalam Rukyatulhilal"...8.

[3] Fajrullah, "Konstruksi Instrumen Moon Verificator Dan Tingkat Akurasinya Dalam Rukyatulhilal"...7.

dapat disimpulkan bahwa sebuah program bisa disusun pada mikrokontroler, sehingga sebuah mikrokontroler dapat menggantikan posisi *Smartphone* ataupun PC sebagai perangkat yang digunakan dalam menyusunan sebuah program.

Mikrokontroler sebagai instrumen elektro dan komputer salah satu fungsi utamanya adalah digunakan untuk membuat sebuah program, seperti program yang akan disusun oleh penulis yaitu menghitung data.

Perancangan program data Matahari dan Bulan pada mikrokontroler dapat dikembangkan secara khusus untuk pengimplementasian Ilmu Falak. Oleh karenanya, penulis tertarik untuk menyajikan data tersebut dalam sebuah program yang penulis namakan *Orbita.* Nama tersebut diambil dari kosakata bahasa Rusia yaitu *орбита*(*Orbita)*[4] yang mempunyai arti Orbit.[5] Kata tersebut dipilih karena mempunyai objek yang sama

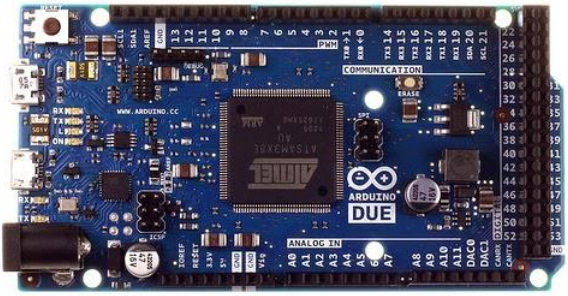
---

[4] (Орбита) atau orbita yang berarti orbit. Lihat dalam, Google Terjemahan, https://translate.google.com/?sl=ru&tl=id&text=%D0%BE%D1%80%D0%B1%D0%B8%D1%82%D0% B0&op=translate, diakses 7 Januari 2022.

[5] Orbit adalah jalur yang dilewati oleh benda langit dalam peredarannya mengitari benda langit lain yang lebih besar gaya gravitasinya. Lihat dalam, Kamus Besar Bahasa Indonesia (KBBI), https://www.kbbi.web.id/orbit, diakses 7 Januari 2022.

dengan nilai *output* dari instrument di atas yaitu data pergerakan benda langit.

Program *Orbita* pada penelitian ini dirancang untuk mikrokontroler Arduino Due[6] , Arduino Due dipilih karena terdapat fitur-fitur yang cocok digunakan dalam perhitungan data ephemeris. Penyusunan program dilakukan pada aplikasi IDE (*Integrated Development Environment*).[7] Fungsi IDE adalah untuk membuka, membuat dan mengedit program yang akan diperintahkan ke dalam arduino.



[6] Ar lari segala sesuatu nel SAM3X8E asis ARM inti 32 di antaranya dapat digunakan sebagai output PWM), 12 input analog, 4 UART (port serial perangkat keras), jam 84 MHz, koneksi USB OTG yang mampu, 2 DAC (digital to analog) , 2 TWI, colokan listrik, header SPI, header JTAG, tombol reset dan tombol hapus. Lihat dalam Beetrona, Tutorial Menggunakan Arduino Due, https://beetrona.com/tutorial-arduino-Due/, diakses 25 Januari 2022.

[7] *Integrated development environment* atau IDE adalah software yang digunakan untuk menulis atau menyusun program yang seterusnya di *upload* dari computer ke-board arduino. Lihat dalam Aldyrazor.com, "Software Arduino IDE: Cara Download, Instal, dan Fungsinya", https://www.aldyrazor.com/2020/05/software-arduino-ide.html, diakses 31 Desember 2021.

Gambar 1.1 Arduino Due

Dari penjelasan rancangan program diatas, penulis berinspirasi mengangkat penelitian dengan judul "*Perancangan Program Perhitungan Data Posisi Matahari Dan Bulan Berbasis Mikrokontroler"*. Dengan harapan penuh perancangan program tersebut dapat digunakan untuk kepentingan keilmuan falak sebagai hisab maupun pengamatan benda langit secara khusus.

B. Rumusan Masalah

Dari serangkaian penjelasan dilatar belakang, penulis merumuskan pokok-pokok permasalahan yang akan dikaji sebagai berikut:

1. Apakah program perhitungan data posisi Matahari dan Bulan dapat dilakukan pada program berbasis mikrokontroler ?
2. Apakah program perhitungan data posisi Matahari dan Bulan berbasis mikrokontroler efektif pada penggunaanya serta akurat pada hasil nilainya ?

C. Tujuan Dan Kegunaan Penelitian

Adapun yang menjadi tujuan penelitian ini adalah:

1. Menghasilkan perancangan program perhitungan data posisi Matahari dan Bulan berbasis mikrokontroler yaitu *Orbita*.
2. Mengetahui penggunaan serta tingkat akurasi perancangan program perhitungan data posisi Matahari dan Bulan berbasis mikrokontroler yaitu *Orbita*.

Sedangkan kegunaan dari penelitian ini adalah:

1. Memberikan kontribusi keilmuan falak khususnya dalam dunia *digital*.
2. Memberikan wawasan terkait mikrokontroler sebagai instrumen untuk pengimplementasian keilmuan falak.

Sebagai motivasi bagi pegiat falak lainnya agar lebih bersemangat serta berinovasi untuk menghasilkan produk-produk Ilmu Falak yang dapat berguna bagi masyarakat banyak.

D. Telaah Pustaka

Penelitian ini secara umum berfokus pada pembahasan perhitungan data ephemeris menggunakan

program berbasis mikrokontroler untuk pengamatan serta hisab dalam keilmuan falak. Sejauh penelusuran yang dilakukan penulis, terdapat karya ilmiah maupun penelitian yang mengkaji dan berkaitan dengan penelitian penulis, diantaranya:

Penelitian yang dilakukan oleh Afif Aulia Rahman yang berjudul *"Sistem Otomatisasi Pelacakan Objek Astronomi Menggunakan Teleskop Berdasarkan Stellarium".* Dalam penelitian tersebut peneliti merancang sistem yang dibangun dengan sebuah Arduino sebagai mikrokontroler serta perangkat pendukung lainya seperti kamera, sensor potensiometer, motor servo dan motor dc. Data *input* diambil dari *software Stellarium* yang berupa koordinat suatu tempat, kemudian data tersebut diolah oleh komputer yang selanjutnya digunakan untuk memerintahkan motor dc mengarahkan ke objek pengamatan dan selanjutnya dilakukan pemotretan oleh kamera. Berdasarkan pengujian, kesalahan hasil konversi sensor derajat azimuth rata-rata sebesar 0.6 derajat, sedangkan untuk altitude rata-rata sebesar 0.6 derajat.[8]

[8] Afif Aulia Rahman, "Sistem Otomatisasi Pelacakan Objek Astronomi Menggunakan Teleskop Berdasarkan Stellarium", *Skripsi*, (Surabaya: Fakultas Teknologi Industri Institut Teknologi Sepuluh Nopember, 2016).

Sistem yang dirancang menggunakan mikrokontroler Arduino sebagai otak kendali instrumennya. Penggunaan mikrontroler Arduino yang populer serta dinilai mudah dipahami oleh pemula menjadi inisiatif penulis untuk menggunakannya pada perancangan program *Orbita.*

M. Burhanuddin Latief dalam karya ilmiahnya yang berjudul *"Sistem Pelacak Otomatis Gerakan Benda Langit Pada Teleskop Refraktor Berbasis Mikrokontroler"* menjelaskan bagaimana penyusunan system pelacak benda-benda langit dengan menggunakan motor stepper yang dikendalikan oleh mikrokontroler jenis ATMega32. Pemrogramanya dibuat untuk *Tracking* benda langit agar memudahkan dalam pengamatan. Kinerja instrumen diuji dengan membandingkan hasil pengukuran pergeseran sudut setiap jam dari teleskop refraktor terhadap nilai teoritiknya untuk setiap benda langit yang diamati.[9] Pemrograman yang dibuat dalam penelitian tersebut memerintahkan instrumen untuk dapat *Tracking* benda langit. Dari program tersebut dapat juga digunakan dalam keperluan Ilmu Falak, sehingga penulis mencoba

[9] M. Burhanuddin Latief, "Sistem Pelacak Otomatis Gerakan Benda Langit Pada Teleskop Refraktor Berbasis Mikrokontroler", Jurnal Fisika Indonesia, Vol 18, No: 54, (Desember 2014), 82.

untuk menghasilkan program tersebut dalam bentuk data pada program *Orbita.*

Penelitian lain juga dilakukan oleh Zul Amri Fathinul Inshafi dalam skripsinya yang berjudul *"Aplikasi Data Ephemeris Matahari Dan Bulan Berdasarkan Perhitungan Jean Meeus Pada Smartphone Android".* Penelitian tersebut menjelaskan rancangan sebuah program aplikasi android yang disusun untuk mencari data ephemeris Matahari dan Bulan. Aplikasi tersebut dirancang tanpa harus terkoneksi dengan internet dengan menggunakan algoritma Jean Meeus dalam bahasa pemrograman Java dan diberi nama Zephemeris. Dari hasil uji coba aplikasi tersebut, disimpulkan bahwa aplikasi ini dapat dijalankan di semua perangkat pintar atau smartphone baik yang berspesifikasi tinggi maupun rendah.[10] Pengoprasian aplikasi Zephemeris tersebut dapat digunakan tanpa harus tersambung dengan internet. Kelebihan dari aplikasi tersebut juga mendorong penulis untuk merancang program *Orbita* yang dapat menghitung azimuth dan ketinggian benda langit dengan

[10] Zul Amri Fathinul Inshafi, "Aplikasi Data Ephemeris Matahari Dan Bulan Berdasarkan Perhitungan Jean Meeus Pada Smartphone Android", Skripsi, (Semarang: Fakultas Syariah Dan Hukum Universitas Islam Negeri Walisongo, 2016).

menggunakan sensor yang dapat menangkap sinyal satellit untuk diteruskan dan dibaca oleh mikrokontroler sehingga menghasilkan data berupa koordinat tempat, waktu terkini hingga ketinggian tempat.

M. Yakub Mubarok dalam penelitian di skripsinya yang berjudul *"Pemrograman Data Ephemeris Matahari Dan Bulan Berdasarkan Perhitungan Jean Meeus Menggunakan Bahasa Program Php (Personal Homepage Hypertext Preprocessor) Dan Mysql (My Structure Query Language)"* menjelaskan sebuah rancangan aplikasi yang disusun dengan sebuah aplikasi perhitungan data ephemeris Matahari dan Bulan yang berjenis web program. Dirancang menggunakan menggunakan algoritma Jean Meeus dan bahasa pemrograman PHP dan MySQL. Berdasarkan hasil dari uji coba aplikasi tersebut disimpulkan bahwa aplikasi ini dapat dijalankan di semua perangkat pintar baik perangkat komputer maupun perangkat mobile. Untuk uji verifikasi hasil perhitungan aplikasi ini yang dikomparasikan dengan Program Posisi Matahari dan Bulan Algoritma Meeus karya Rinto Anugraha, disimpulkan bahwa selisih hasil terjadi pada data Delta T, bujur Bulan, ascensio rekta Bulan, deklinasi Bulan dan jarak Bumi-Bulan untuk perhitungan pada

tahun sebelum epoch, dengan nilai selisih tidak lebih dari dua angka di belakang koma pada orde detik busur.[11] Pemrograman data ephemeris pada penelitian tersebut menggunakan algoritma Jean Meeus. Untuk keperluan praktis, algoritma tersebut sudah cukup akurat. Dalam rancangan program *Orbita*, penulis juga berinisiatif menggunakan algoritma Jean Meeus dalam perhitungan data ephemerisnya.

Berdasarkan penelusuran penulis dari beberapa referensi di atas, belum terdapat penelitian atau tulisan yang membahas secara spesifik perhitungan data ephemeris dengan program yang berbasis mikrokontroler, sehingga penulis menilai bahwa penelitian tentang metode ini perlu untuk dikembangkan serta didalami lebih lanjut.

E. Metode Penelitian

Metode penelitian merupakan ilmu tentang proses ilmiah serta bertanggung jawab yang harus

[11] M. Yakub Mubarok, "Pemrograman Data Ephemeris Matahari Dan Bulan Berdasarkan Perhitungan Jean Meeus Menggunakan Bahasa Program Php (Personal Homepage Hypertext Preprocessor) Dan Mysql (My Structure Query Language)", *Skripsi*, (Semarang: Institut Agama Islam Negeri Walisongo, 2013).

dilalui untuk tercapainya suatu pemahaman dengan harus melalui syarat ketelitian.[12] Dalam penelitian ini, penulis menggunakan metode penelitian sebagai berikut:

1. Jenis Penelitian

   Penelitian ini menggunakan desain *Research & Development* (R&D). Desain dari *reseach and development* merupakan penelitian yang dilakukan dalam konteks pengembangan produk atau program untuk tujuan meningkatkan baik dalam hal yang sedang dikembangkan atau pengembangan.[13] Metode *Research & Development* juga dapat mencakup pengembangan alat baru, produk, atau proses.[14]Sumber Data

   a) Data Primer

      Data yang diperoleh langsung dari sebuah pengukuran, pengamatan

[12] Priyono, *Metode Penelitian Kuantitatif*, (Sidoarjo: Zifatama Publishing, 2016), 3.

[13] Sugiyono, *Metode Penelitian Kuantitatif, Kualitatif Dan R & D*, (Bandung: Alfabeta, 2013), 297.

[14] Sugiyono, *Metode Penelitian Kuantitatif, Kualitatif Dan R & D..*, 297.

atau observasi.[15] Dalam penelitian ini, data primer didapatkan dari buku *Mekanika Benda Langit* karya Rinto Anugraha, sebagai rujukan dasar perhitungan data ephemeris, dan buku *Panduan Dasar Mikrokontroler Arduino* sebagai rujukan pemrograman serta perancangan program *Orbita*.

b) Data Sekunder

Data yang diperoleh secara tidak langsung dari sebuah laporan, buku, ataupun hasil penelitian lain.[16] Untuk data sekunder diambil penulis dari buku-buku, makalah-makalah hingga tulisan-tulisan yang berkaitan dengan keilmuan falak, data ephemeris dan pemrograman yang berbasis mikrokontroler.

2. Metode Pengumpulan Data

---

[15] Hardani dkk, *Metode Penelitian Kualitatif & Kuantitatif*, (Yogyakarta: CV. Pustaka Ilmu, 2020), 247.

[16] Hardani dkk, *Metode Penelitian Kualitatif & Kuantitatif*,…247.

Metode pengumpulan data adalah cara-cara yang perlu ditempuh dalam sebuah penelitian untuk mengumpulkan data yang dapat menjelaskan suatu permasalahan dalam penelitian secara objektif.[17] Penulis menggunakan metode *Library research* dengan memulai menelusuri literatur-literatur yang berkaitan dengan pembahasan mengenai data-data astronomis Matahari dan Bulan yang digunakan dalam perhitungan falak dan selanjutnya penelusuran mengenai tata cara pemrograman mikrokontroler dan perancangannya berikut kelemahan dan kelebihannya.

Berdasarkan hasil penelusuran di atas, penulis kemudian menemukan bahwa perhitungan data ephemeris Matahari dan Bulan yang menggunakan algoritma Jean Meeus termasuk dalam perhitungan dengan akurasi tinggi, di mana hasil perhitungannya dapat digunakan untuk data perhitungan falak

[17] Syahrum, Salim, *Metodologi Penelitian Kuantitatif*, (Bandung: Citapustaka Media, 2012), 131.

kontemporer. Selanjutnya penulis mendapatkan bahwa perhitungan data posisi Matahari dan Bulan tersebut dapat implementasikan pada pemrograman yang berbasis mikrokontroler.

3. Metode Pemrograman

Secara garis besar metode pemrograman yang digunakan penulis dibagi menjadi dua tahap, yakni: tahap desain dan perancangan program, dan tahap implementasi rancangan program aplikasi.

a. Perancangan Program

Pada tahap rancangan ini, penulis merancang skema umum program yang meliputi proses: data masukan dan penyusunan program perhitungan data posisi Matahari dan Bulan.

b. Tahap Implementasi Rancangan Program

Pada tahap ini, program perhitungan data posisi Matahari dan Bulan diimplementasikan pada mikrokontroler. Penyusunan program dilakukan pada aplikasi IDE (*Integrated Development Environment*) yang selanjutnya di*upload* pada mikrokontroler.

4. Uji Coba dan Evaluasi

Secara umum uji coba program dilaksanakan untuk mengetahui apakah program yang telah dibuat dapat berjalan dengan baik atau tidak. Pada bagian ini penulis merencanakan 2 (dua) tahap pengujian terhadap program tersebut. Kedua tahap pengujian adalah sebagai berikut:

a. Eksaminasi I: Uji coba penggunaan program.

Pada uji coba tahap ini selain dilakukan uji data input program dengan mengoperasikan pada berbagai variasi lokasi dan waktu. Adapun Uji coba tahap ini dilakukan untuk mengetahui kemampuan kompatibilitas program dan batas validitas data yang dihasilkan.

b. Eksaminasi II: Uji akurasi program.

Proses uji coba tahap ini menggunakan metode analisis komparatif yakni dengan mengkomparasikan hasil program dengan hasil dari aplikasi Daff Moon

dan Stellarium. Uji coba tahap ini bertujuan untuk mengetahui selisih antara data yang dihasilkan program *Orbita* dengan data yang dihasilkan program pembanding.

Dengan 2 tahap pengujian tersebut, diharapkan dapat diketahui apakah program *Orbita* layak digunakan atau tidak dalam perhitungan falak. Disamping itu, dengan pengujian-pengujian tersebut penulis juga dapat mempertanggungjawabkan hasil penelitian penulis secara ilmiah.

F. Sistematika Penulisan

Secara garis besar, penelitian ini penulis susun dengan lima bab pembahasan, serta setiap bab terdapat beberapa sub bab pembahasan sebagaimana berikut.

BAB I: PENDAHULUAN

Bab pertama berisi deskripsi umum tentang penelitian yang meliputi latar belakang, rumusan masalah, tujuan penulisan, kegunaan penulisan, telaah pustaka, metode penelitian, dan sistematika penulisan.

BAB II: TINJAUAN UMUM DATA POSISI MATAHARI DAN BULAN SERTA PENGERTIAN

MIKROKONTROLER

Bab kedua berisi tentang landasan teori serta konsep umum dari judul penelitian penulis, diantaranya: tentang tinjauan umum dari data posisi Matahari dan Bulan dan pengimplementasinya terhadap Ilmu Falak serta tinjauan umum dari mikrokontroler.

BAB III: PERANCANGAN BERBASIS MIKROKONTROLER UNTUK PERHITUNGAN POSISI MATAHARI DAN BULAN BERUPA PROGRAM *ORBITA*

Bab ketiga berisi gambaran umum dari perancangan, serta pemrograman data posisi Matahari dan Bulan berbasis mikrokontroler arduino.

BAB IV: ANALISIS TERHADAP *ORBITA* SEBAGAI PROGRAM BERBASIS MIKROKONTROLER

Pembahasan bab keempat lebih terfokuskan pada uji coba, verifkasi serta evaluasi. Bab ini membahas beberapa tahap pengujian sebagaimana telah ditetapkan sebelumnya, yang dilakukan untuk mengetahui kelayakan program. Selanjutnya pada bab ini juga disertakan evaluasi hal-hal penting yang diketahui setelah pelaksanaan berbagai macam pengujian terhadap program tersebut.

BAB V: PENUTUP

Pada bagian penutup dijelaskan mengenai kesimpulan, saran/rekomendasi terkait dengan hasil penelitian penulis, berupa program berikut algoritma pemrogramannya, dan penutup.

BAB II

TINJAUAN UMUM DATA POSISI MATAHARI DAN BULAN SERTA PENGERTIAN MIKROKONTROLER

A. Tinjauan Data Posisi Matahari Dan Bulan

Secara umum ilmu astronomi mempelajari tentang jarak, posisi dan pergerakan benda-benda langit. Jarak, posisi dan pergerakan benda langit tentunya tidak dapat diketahui secara langsung, namun dapat diketahui dengan pendekatan perhitungan dan pengamatan yang selanjutnya dapat dirangkum serta disajikan dalam sebuah tabel yang bernama data ephemeris.[18]

Setiap waktu, terjadi pergerakan benda langit di tata surya, namun pergerakan tersebut bukan tanpa tujuan seperti yang sudah Allah SWT singgung dalam Al-qur'an Surat Ad-Dukhan ayat 38-39.

وَمَا خَلَقْنَا ٱلسَّمَٰوَٰتِ وَٱلْأَرْضَ وَمَا بَيْنَهُمَا لَٰعِبِينَ ٣٨ مَا خَلَقْنَٰهُمَآ إِلَّا بِٱلْحَقِّ وَلَٰكِنَّ أَكْثَرَهُمْ لَا يَعْلَمُونَ ٣٩

[18] Asrini, "Studi Komparatif Hisab Kontemporer Ephemeris Dan Algoritma", *Skripsi*, (Makasar: Fakultas Syariah Dan Hukum Uin Alauddin, 2020), 29.

Artinya:"(38) dan Kami tidak menciptakan langit dan Bumi dan apa yang ada antara keduanya dengan bermain-main. (39) Kami tidak menciptakan keduanya melainkan dengan haq, tetapi kebanyakan mereka tidak mengetahui.*"*[19]

Allah SWT menciptakan alam semesta ini beserta isinya dengan penuh keserasian, keharmonisan dan keteraturan. Semua itu bukan tanpa kebetulan, melainkan dengan haqq, di mana terdapat tujuan dan manfaat dalam penciptanya. Tujuan dan manfaat untuk kehidupan makhluk di Bumi, serta untuk menuntun manusia menuju keimanan dan penghambaan kepada-Nya.[20]

Benda-benda langit yang dapat diamati dari Bumi, baik yang nampak pada siang maupun malam hari, seluruhnya bergerak secara teratur dari arah timur ke barat. Setiap benda langit tersebut terbit dan tenggelam pada posisi tertentu di Bumi, di mana posisi terbit dan tenggelamnya kemudian berubah ke arah tertentu secara gradual dan kembali lagi ke posisi semula pada waktu tertentu. Seolah-olah semua benda langit tersebut,

[19] QS. Ad-Dukhan: 38-39, lihat: Tim Penerjemah, Al-Qur'an dan Terjemahannya (Terjemahan Departermen Agama RI), Bandung: CV Diponegoro, tt., 397.

[20] Lihat: Quraisy Shihab, Tafsir al-Misbah, vol. 13, Jakarta: Lentera Hati, 2001, hlm. 21- 22 Lihat juga: Tantowi Jauhari, Jawahir fi Tafsir al-Qur'an al-Karim, juz 21, Mesir: Musthofa al-Baaby al-Khaaly wa Awladuhu, tt, 16.

termasuk Matahari dan Bulan, beredar mengelilingi Bumi.21 Pada awalnya manusia menganggap bahwa peristiwa pergerakan benda langit tersebut merupakan sesuatu yang magis. Meski demikian, manusia telah lama memanfaatkan peristiwa tersebut untuk urusan hidup mereka khususnya sebagai penanda waktu untuk memulai pekerjaanpekerjaan tertentu. Seiring dengan perkembangan peradaban dan keilmuan manusia, berbagai macam teori pergerakan benda langit pun dikemukakan. Dalam sejarah keilmuan astronomi, terdapat 3 teori pergerakan benda langit yang pernah dikemukakan oleh para astronom terdahulu, yakni teori egosentris, geosentris dan heliosentris.[22]

1. Teori Pergerakan Benda Langit

   Teori egosentris adalah teori yang menganggap bahwa manusia merupakan pusat alam semesta. Teori ini telah diyakini oleh manusia sejak zaman purbakala. Berdasarkan teori ini seluruh benda langit berputar mengikuti kemanapun manusia bergerak. Berangkat dari teori ini, Thales astronom Yunani pada

[21] Moedji Raharto, *Sistem Penanggalan Syamsiah/Masehi*, (Bandung: Penerbit ITB, 2001), 1-2

[22] Slamet Hambali, *Pengantar Ilmu Falak*, (Banyuwangi: Bismillah Publisher, 2012), 175-186

sekitar abad ke-6 SM berpendapat bahwa Bumi berbentuk dataran yang sangat luas, di mana benda langit bergerak di atasnya.[23] Teori geosentris adalah teori yang mengatakan bahwa Bumi merupakan pusat tata surya. Teori ini pertama kali dikemukakan oleh Aristoteles pada abad ke-3 SM. Menurutnya Bumi merupakan benda langit yang yang berbentuk bulat sebagaimana yang dikemukakan oleh Pythagoras pada abad ke-5 SM, sedangkan Matahari, Bulan, planet-planet dan bintang-bintang, seluruhnya bergerak mengitari Bumi. Teori ini kemudian disempurnakan oleh Claudius Ptolomeus dengan bentuk lintasan orbit yang lebih rumit dimana beberapa planet, seperti Mars, Yupiter dan Saturnus bergerak mengelilingi Matahari sekaligus mengelilingi Bumi bersama Matahari. Teori ini bertahan cukup lama bahkan menjadi ajaran resmi gereja beberapa ratus tahun kemudian.[24]

Teori heliosentris adalah teori yang menyatakan bahwa Matahari merupakan pusat tata surya. Bumi beserta planet-planet lainnya secara bersamaan beredar mengelilingi Matahari pada orbit-orbit

[23] Slamet Hambali, *Pengantar Ilmu Falak,…*178.
[24] Slamet Hambali, *Pengantar Ilmu Falak,…*179-182.

tertentu yang berbentuk epicycle (bulat). Teori ini dikemukakan oleh Nicolas Copernicus pada abad ke-14 M untuk membantah teori Geosentris yang selama ini dianut oleh gereja. Teori ini sebenarnya bukanlah murni pemikiran Copernicus, melainkan pengembangan dari teori heliosentris yang pernah dikemukakan oleh Aristarchus dari Samos pada abad ke-4 SM namun tidak mendapat dukungan pada masa itu.[25]

Adapun mengenai teori heliosentris tersebut, Al-Qur'an telah lama menyinggungnya di dalam surat Yasin ayat 38-40.

وَٱلشَّمۡسُ تَجۡرِي لِمُسۡتَقَرّٖ لَّهَاۚ ذَٰلِكَ تَقۡدِيرُ ٱلۡعَزِيزِ ٱلۡعَلِيمِ ٣٨ وَٱلۡقَمَرَ قَدَّرۡنَٰهُ مَنَازِلَ حَتَّىٰ عَادَ كَٱلۡعُرۡجُونِ ٱلۡقَدِيمِ ٣٩ لَا ٱلشَّمۡسُ يَنۢبَغِي لَهَآ أَن تُدۡرِكَ ٱلۡقَمَرَ وَلَا ٱلَّيۡلُ سَابِقُ ٱلنَّهَارِۚ وَكُلّٞ فِي فَلَكٖ يَسۡبَحُونَ ٤٠

Artinya:"(38) dan Matahari berjalan ditempat peredarannya. Demikianlah ketetapan yang Maha Perkasa lagi Maha mengetahui. (39) dan telah Kami tetapkan bagi Bulan manzilah-manzilah, sehingga (setelah Dia sampai ke manzilah yang terakhir) Kembalilah Dia sebagai bentuk tandan yang tua. (40) tidaklah mungkin bagi Matahari

[25] Slamet Hambali, *Pengantar Ilmu Falak,…*182-184.

> mendapatkan Bulan dan malampun tidak dapat mendahului siang. dan masing-masing beredar pada garis edarnya.”[26]

Pada ayat di atas dijelaskan bahwa Matahari dan Bulan beredar pada garis edarnya masing-masing sebagaimana yang telah ditentukan oleh Allah SWT. Matahari tidak mungkin bergerak mendahului Bulan karena garis edar dan pusat peredaran keduanya berbeda, Bulan bergerak mengitari Bumi sedangkan Matahari bergerak mengitari pusat galaksi Bimasakti.[27] Meski teori egosentris dan geosentris telah terbantahkan oleh teori heliosentris, namun hingga saat ini kedua teori tersebut masih dipakai dalam keilmuan astronomi dan falak sebagai dasar teori pembahasan sistem tata koordinat bola langit.

2. Gerakan Matahari, Bumi dan Bulan

Menurut teori heliosentris, Matahari merupakan pusat peredaran benda-benda langit di dalam tata surya kita. Planet Bumi selain berputar pada porosnya, bersama dengan Bulan bergerak mengitari

---

[26] QS. Yasin: 38-40, lihat: Tim Penerjemah, Al-Qur‟an dan Terjemahannya (Terjemahan Departermen Agama RI), Bandung: CV Diponegoro, tt., 353.

[27] Quraisy Shihab, Tafsir al-Misbah, vol. 11, Jakarta: Lentera Hati, 2001, 540-544.

Matahari melalui lintasan khayal berbentuk ellips, sebagaimana yang dijelaskan dalam hukum Kepler. Sedangkan Bulan pada saat yang bersamaan berputar pada porosnya sembari mengitari Bumi. Pergerakan-pergerakan tersebut ketika diamati dari Bumi terlihat sebagai pergerakan yang bersifat semu. Gerak semu inilah yang sejak lama telah banyak dimanfaatkan oleh manusia khususnya dalam perhitungan waktu.[28] Dalam keilmuan falak pergerakan-pergerakan tersebut sangat penting, karena beberapa perintah ibadah dalam Islam, waktu pelaksanaannya sangat terkait dengan posisi dan pergerakan Matahari, Bumi dan Bulan tersebut.[29]

1) Gerak Bumi

Bumi merupakan planet ketiga dari Matahari, setelah Merkurius dan Venus. Diameter Bumi sekitar 12.769 km. Adapun jarak rata-rata Bumi-Matahari sekitar 149.597.871 km. Jarak tersebut dalam astronomi disebut dengan Astronomical Unit (AU), di mana 1 AU =

[28] Moedji Raharto, *Sistem Penanggalan Syamsiah/Masehi*, (Bandung: Penerbit ITB, 2001), 4-5.

[29] Ahmad Izzuddin, *Ilmu Falak Praktis*, (Semarang: Pustaka Rizki Putra, 2012), 2-4.

149.597.871 km.[30] Jarak Bumi-Matahari tidak selalu sama melainkan kadang jauh kadang dekat, sesuai dengan posisi Bumi di ekliptika. Bumi mengelilingi Matahari melalui lintasan semu berbentuk ellips mendekati lingkaran. Selisih antara titik perihelium dengan titik aphelium-nya adalah sekitar 5.000.000 km.[31] Jarak terdekat (perihelium) BumiMatahari adalah sekitar 147 juta km, dan jarak terjauhnya (aphelium) sekitar 152 juta km.[32] Selama beredar mengelilingi Matahari, Bumi tidak hanya melakukan satu gerakan saja, bahkan lima gerakan sekaligus. Kelima gerakan Bumi tersebut yakni:

a) Rotasi

Rotasi Bumi adalah gerakan Bumi pada porosnya. Bumi berotasi dari arah barat ke timur. Dalam sekali rotasi Bumi, dibutuhkan waktu rata-rata 23^{j} 56^{m} 4^{d} , bisa kurang dari itu dan bisa lebih, tergantung posisi Bumi

[30] Lihat dalam http://neo.jpl.nasa.gov/glossary/au.html diakses pada hari Sabtu, diakses pada 5 Maret 2022 pukul 16:07 WIB.

[31] Slamet Hambali, *Pengantar Ilmu Falak*, (Banyuwangi: Bismillah Publisher, 2012), 131.

[32] Muhyiddin Khazin, *Ilmu Falak Dalam Teori dan Praktik*, (Yogyakarta: Buana Pustaka, 2008), 125.

terhadap Matahari ketika itu. [33] Arah rotasi Bumi yang berlawanan dengan jarum jam (retrograde) menyebabkan pergerakan semu harian benda langit, sehingga semua benda langit jika dilihat dari Bumi nampak bergerak dari arah timur ke barat.[34]

b) Revolusi

Bumi merupakan salah satu planet yang bergerak mengitari Matahari. Pergerakan Bumi mengitari Matahari disebut dengan revolusi Bumi. Dalam revolusinya, posisi Bumi miring sekitar 66,5° terhadap bidang ekliptika, sehingga bidang ekliptika Bumi tidak sejajar dengan ekuator Bumi, melainkan miring membentuk sudut sebesar 23,5°. Arah kemiringan tersebut meski kadang berubah namun relatif tetap.[35] Ekliptika Bumi berbentuk ellips mendekati lingkaran dengan selisih antara titik perihelium dengan titik aphelium-nya sekitar 5.000.000 km.[36] Periode

---

[33] Slamet Hambali, *Pengantar Ilmu Falak*, (Banyuwangi: Bismillah Publisher, 2012), 197.

[34] Slamet Hambali, *Pengantar Ilmu Falak*,..197.

[35] Slamet Hambali, *Pengantar Ilmu Falak*,..202-203.

[36] Bayong Tjasyono, *Ilmu Kebumian dan Antariksa*, (Bandung: Pascasarjana UPI, 2009), 33-34.

rata-rata revolusi Bumi yakni 365^{h} 5^{j} 48^{m} $45{,}2^{d}$. Periode tersebut kemudian disebut dengan 1 tahun sideris. Arah revolusi Bumi sama dengan arah rotasinya yakni dari barat ke timur, berlawanan dengan arah jarum jam.[37] Jika Matahari berada di kulminasi atas di suatu tempat, maka setelah Bumi berotasi selama 23^{j} 56^{m} , sebenarnya Matahari belum mencapai titik kulminasi kembali melainkan masih kurang sekitar 1°. Namun akibat dari gerak revolusi Bumi terhadap Matahari rupanya pada saat yang bersamaan Matahari juga telah bergerak semu sekitar 1°.[38]

c) Presesi

Kemiringan sumbu Bumi terhadap ekliptika tidak selalu tetap, melainkan terus berubah-ubah mirip perubahan sumbu gasing. Perubahan tersebut mengakibatkan gerak goyang pada Bumi sebesar 50.24”/tahun. Gerak tersebut disebut dengan gerak presesi atau *dahriyah* atau *mubadarat al-I’tidalain*.

[37] Slamet Hambali, *Pengantar Ilmu Falak*, (Banyuwangi: Bismillah Publisher, 2012), 202-203.

[38] Slamet Hambali, *Pengantar Ilmu Falak*, …197.

Arah gerak presesi berlawan dengan gerak rotasi, yakni dari arah timur ke barat jika dilihat dari kutub utara langit, dan akan kembali ke posisi semula dalam jangka waktu sekitar 25.796 tahun.[39]

d) Nutasi

Gerak nutasi adalah gerak gelombang dalam gerak presesi. Gerak presesi tidak lurus, melainkan bergelombang membentuk lingkaran-lingkaran kecil. Gerak nutasi untuk membentuk 1 lingkaran penuh (360°) memerlukan waktu sekitar 18,66 tahun, sehingga besar gerak nutasi adalah 0o 03’10,05”/hari.[40]

e) Apsiden

Gerak apsiden merupakan gerak pergeseran titik aphelium dan perihelium dari timur ke barat. Untuk menempuh 1 kali putaran gerak apsiden memerlukan waktu sekitar 21.000 tahun, sehingga besar gerak apsiden adalah 0,17”/hari.[41]

---

[39] Muhyiddin Khazin, *Ilmu Falak Dalam Teori dan Praktik*, (Yogyakarta: Buana Pustaka, 2008), 130-131.
[40] Muhyiddin Khazin, *Ilmu Falak Dalam Teori dan Praktik*,… 130-131.
[41] Muhyiddin Khazin, *Ilmu Falak Dalam Teori dan Praktik*,… 130-131.

2) Gerak Matahari

Matahari merupakan pusat tata surya kita. Bumi, planet-planet dan benda langit yang berada di jangkauan gravitasi Matahari, bergerak bersamaan mengitari Matahari. Pada saat yang bersamaan Matahari juga juga terus bergerak di alam semesta ini bersamaan bintang-bintang lainnya. Dalam keilmuan astronomi gerak Matahari dibagi menjadi dua macam, yakni gerak hakiki dan gerak semu.

a) Gerak Matahari Hakiki

Gerak Matahari Hakiki adalah gerakan sebenarnya yang dimiliki oleh Matahari. Gerakan Matahari Hakiki ada dua, yakni:

i. Rotasi Matahari. Matahari berputar pada porosnya dengan waktu rotasi yang berbeda-beda pada tiap bagiannya, yakni sekitar 25,5 hari pada bidang ekuator dan 27 hari pada daerah kutubnya. Perbedaan tersebut disebabkan Matahari sebenarnya merupakan bola gas pijar raksasa yang berada di luar angkasa yang terus bergerak.[42]

[42] Slamet Hambali, *Pengantar Ilmu Falak*, (Banyuwangi: Bismillah Publisher, 2012), 212-213.

ii. Gerak Matahari di antara gugusan bintang. Matahari bersamaan dengan sistem tata surya-nya bergerak di alam semesta ini dari suatu tempat menuju tempat yang lainnya mengitari pusat galaksi Bimasakti dengan kecepatan sekitar 20 km/detik atau 72.000 km/jam atau 600 juta km/tahun. Daerah yang dituju oleh Matahari disebut dengan apeks dan daerah yang telah ditinggalkan oleh Matahari disebut anti-apeks.[43]

b) Gerak Semu Matahari

Jika diamati dari permukaan Bumi, Matahari terlihat seolaholah bergerak dari timur ke barat mengitari Bumi. Posisi terbit dan terbenam Matahari tidak selalu tetap, melainkan berubah secara gradual dari satu titik ke titik yang lain hingga akhirnya kembali ke titik awal lagi. Lintasan Matahari tersebut kemudian membentuk lingkaran besar yang disebut lingkaran ekliptika. Lingkaran ekliptika tidak berimpit dengan ekuator, namun membentuk

[43] Slamet Hambali, *Pengantar Ilmu Falak*,…212-213.

sudut sekitar 23º 27'. [44] Secara umum gerak semu Matahari dapat dibagi menjadi dua, yakni gerak semu harian dan gerak semu tahunan.

i. Gerak Semu Harian (Gerak Diurnal), terjadi akibat rotasi Bumi. Periode menengahnya yakni 24 jam. Arah pergerakannya adalah dari timur ke barat. Kemiringan lintasan gerak harian Matahari tergantung letak geografis pengamat. Lintasan pada bagian ekuator Bumi adalah berupa lingkaran tegak, di bagian kutub mendatar, di belahan Bumi selatan terlihat miring ke arah utara dan sebaliknya di belahan Bumi utara terlihat miring ke selatan. Besar kemiringan tersebut berbanding lurus dengan besar lintangnya.[45]

ii. Gerak Semu Tahunan (Gerak Annual), arah gerak semu tahunan Matahari yakni ke arah timur sekitar 0o 59'/hari. Periode gerak semu tahunan Matahari adalah sekitar 365,25 hari, akibatnya arah terbit dan

[44] Muhyiddin Khazin, *Ilmu Falak Dalam Teori dan Praktik*, (Yogyakarta: Buana Pustaka, 2008), 126.

[45] Abdur Rachim, *Ilmu Falak*, (Yogyakarta: Penerbit Liberty, 1983), 1.

tenggelam Matahari selalu berubah letaknya sepanjang tahun.[46]

Pada tanggal 21 Maret dan 23 September Matahari terbit tepat di titik timur dan tenggelam tepat di titik barat, pada tanggal 22 Juni Matahari terbit dan tenggelam sejauh 23,5o ke arah utara dari titik timur dan barat, sebaliknya pada tanggal 22 Desember Matahari berada 23,5 o ke arah selatan dari titik timur dan barat. Posisi Matahari ketika berada di dua titik terakhir disebut dengan soltitium, yang artinya pemberhentian Matahari. Hal tersebut karena pada saat itu perubahan deklinasi Matahari sangat lambat seolah-olah berhenti. Sebaliknya pada titik ekuinox, yakni ketika lintasan Matahari berada tepat pada titik timur dan barat, perubahan deklinasi berlangsung cepat.[47]

3) Gerak Bulan

Bulan merupakan satelit dari Bumi yang mempunyai jarak terdekat adalah adalah 363.000 dan

[46] Muhyiddin Khazin, *Ilmu Falak Dalam Teori dan Praktik*, (Yogyakarta: Buana Pustaka, 2008), 126.

[47] Slamet Hambali, *Pengantar Ilmu Falak*, (Banyuwangi: Bismillah Publisher, 2012), 214.

jarak terjauhnya adalah 406.000 km.[48] Meski jarak Bulan-Bumi cukup dekat bahkan masih dalam jangkauan gravitasi Bumi, Bulan tidak sepenuhnya tertarik gaya gravitasi Bumi, sebab Bulan memiliki gaya sentrifugal yang membuatnya tetap dapat bertahan pada lintasannya.[49] Namun akibat gaya sentrifugal Bulan yang sedikit lebih besar dibanding gaya gravitasi Bumi-Bulan, Bulan semakin menjauh sekitar 3,8 cm setiap tahunnya.[50]

Sebagaimana gerak Matahari, di dalam astronomi juga dikenal dua jenis gerak Bulan yakni gerak hakiki dan gerak semu.

a) Gerak Bulan Hakiki

Gerak Bulan hakiki adalah gerak yang sebenarnya dilakukan oleh ketika beredar di angkasa luar. Gerak hakiki Bulan terdiri dari tiga macam gerak, yakni rotasi, revolusi dan gerak Bulan bersama dengan Bumi mengitari Matahari.

---

[48] A. Gunawan Admiranto, *Eksplorasi Tata Surya*, (Bandung: PT Mizan Pustaka, 2017), 105.

[49] Slamet Hambali, *Pengantar Ilmu Falak*, (Banyuwangi: Bismillah Publisher, 2012), 135.

[50] Slamet Hambali, *Pengantar Ilmu Falak*,…136.

i. Rotasi Bulan. Bulan berputar pada porosnya dengan periode sekitar 27 hari lebih 7 jam dengan arah rotasi berlawanan dengan jarum jam. Lama rotasi Bulan adalah sama dengan lama revolusinya. Hal tersebut yang mengakibatkan permukaan Bulan yang menghadap ke Bumi selalu sama.[51]

ii. Revolusi Bulan. Bulan membutuhkan 27,3 hari untuk menyelesaikan orbitnya terhadap Bumi.[52] Sebagaimana rotasinya, arah revolusi Bulan juga berlawanan dengan arah jarum jam. Lama revolusi Bulan tersebut kemudian disebut dengan 1 periode sideris Bulan.[53]

iii. Gerak Bulan bersama Bumi mengelilingi Matahari. Bulan bergerak mengitari Bumi, maka secara otomatis Bulan juga bergerak mengitari Matahari bersama-sama dengan Bumi. Hal tersebut yang menyebabkan lintasan revolusi Bulan tidak berbentuk lingkaran sempurna melainkan lingkaran

---

[51] Slamet Hambali, *Pengantar Ilmu Falak*,…217.

[52] A. Gunawan Admiranto, *Eksplorasi Tata Surya*, (Bandung: PT Mizan Pustaka, 2017), 105.

[53] Muhyiddin Khazin, *Ilmu Falak Dalam Teori dan Praktik*, (Yogyakarta: Buana Pustaka, 2008), 132.

berpilin di mana titik awal revolusi Bulan tidak bertemu titik akhirnya. Satu lingkaran berpilin ini ditempuh Bulan dalam waktu 29,5 hari.[54] Adapun waktu yang diperlukan Bulan untuk mencapai titik awalnya yakni sekitar 365,5 hari atau setelah melewati 12 kali lingkaran spiral.[55]

b) Gerak Semu Bulan

Gerak rotasi Bumi mengakibatkan penampakan benda langit, termasuk Bulan, ketika diamati dari Bumi bergerak secara semu dari arah timur ke barat. Pada saat yang bersamaan Bulan juga melakukan gerak revolusi. Akibatnya, setiap harinya Bulan terlambat terbit dari bintang tertentu sekitar 50 menit atau sekitar 13° busur. Terhadap Matahari, setiap hari Bulan terlambat sekitar 12° busur atau 0,5° setiap jamnya.[56]

Hal tersebut kemudian menimbulkan penampakan Bulan yang berubah-ubah setiap

[54] A. Gunawan Admiranto, *Eksplorasi Tata Surya*, (Bandung: PT Mizan Pustaka, 2017), 105.

[55] Slamet Hambali, *Pengantar Ilmu Falak*, (Banyuwangi: Bismillah Publisher, 2012), 223.

[56] Slamet Hambali, *Pengantar Ilmu Falak*,…224.

harinya, mulai dari sebatas garis kecil melengkung hingga semakin membesar membentuk bulatan sempurna kemudian mengecil kembali. Peristiwa perubahan fasefase penampakan semu Bulan tersebut diakibatkan oleh fungsi elongasi Bulan, yakni sudut yang dibentuk Bulan dari Matahari ketika diamati dari Bumi. Bulan mencapai fase purnama ketika sudut elongasinya sebesar 180° dan fase bulan mati pada sudut 0°.[57]

Periode revolusi Bulan yang disertai dengan fase-fase permukaannya berbeda dengan periode sideris Bulan. Waktu yang dibutuhkan oleh Bulan untuk kembali ke fase awal adalah sekitar 29,5305882 hari. Lama waktu tersebut kemudian disebut dengan 1 periode sinodis Bulan.[58]

Adanya teori heliosentris, ditemukannya hukum lintasan planet oleh Kepler, berserta fakta mengenai ketepatan jarak antara Bumi, Bulan dan Matahari serta pengaruhnya bagi kehidupan di Bumi, merupakan bukti bahwa alam semesta ini dirancang

[57] Slamet Hambali, *Pengantar Ilmu Falak*,…225.
[58] Slamet Hambali, *Pengantar Ilmu Falak*,…219-222.

dengan sistem yang rumit nan teliti namun teratur dan harmonis. Hal tersebut sesuai dengan apa yang telah dinyatakan dalam surat al-An'am ayat 96:

فَالِقُ ٱلۡإِصۡبَاحِ وَجَعَلَ ٱلَّيۡلَ سَكَنٗا وَٱلشَّمۡسَ وَٱلۡقَمَرَ حُسۡبَانٗاۚ ذَٰلِكَ تَقۡدِيرُ ٱلۡعَزِيزِ ٱلۡعَلِيمِ ٩٦

Artinya: "Dia yang menyingsingkan pagi dan menjadikan malam untuk istirahat, dan (menjadikan) Matahari dan Bulan untuk perhitungan. Itulah ketentuan Allah Yang Maha Perkasa lagi Maha Mengetahui"[59]

Terdapat juga di dalam surat Ar-Rahman ayat 5:

ٱلشَّمۡسُ وَٱلۡقَمَرُ بِحُسۡبَانٖ ٥

Artinya: "Matahari dan Bulan (beredar) menurut perhitungan"[60]

Kedua ayat di atas menggunakan pilihan kata حسبانuntuk menjelaskan perhitungan Matahari dan Bulan. Kata حسبانberasal dari kata حسابartinya perhitungan, penambahan alif dan nun pada kata tersebut menunjukkan arti kesempurnaan dan

[59] QS. al-An'am: 96, lihat : Tim Penerjemah, Al-Qur"an dan Terjemahannya (Terjemahan Departermen Agama RI), Bandung: CV Diponegoro, tt., 111.

[60] QS. ar-Rahman: 5, lihat : Tim Penerjemah, Al-Qur"an dan Terjemahannya (Terjemahan Departermen Agama RI), Bandung: CV Diponegoro, tt., 425.

ketelitian.[61] Oleh karena itu kata حسبانpada kedua ayat diatas dapat diartikan bahwa Matahari dan Bulan sejak awal penciptaannya telah berada pada sistem yang sangat teliti dan rumit namun akurat dan teratur.[62] Pendapat lain mengatakan bahwa حسبان menunjukkan arti bahwa pergerakan Matahari dan Bulan adalah dapat diketahui kadar perhitungannya oleh manusia.[63] Dari kedua penafsiran tersebut dapat disimpulkan bahwa manusia dapat memperhitungkan posisi dan pergerakan Matahari dan Bulan karena keduanya bergerak secara teratur, kemudian mengambil manfaat dari hal-hal yang ditimbulkan oleh keteraturan pergerakan tersebut bagi kehidupan mereka, salah satunya yakni dalam perhitungan waktu.

3. Data Ephemeris Dalam Ilmu Falak

Di bidang astronomi pengamatan benda langit menjadi kebutuhan khususnya di kalangan akademisi.[64]

---

[61] Quraish Shihab, Tafsir Al-Mishbah, vol. 4, Jakarta: Lentera Hati, 2001, 205.

[62] Quraish Shihab, Tafsir Al-Mishbah, vol. 12, Jakarta: Lentera Hati, 2001, 496-498.

[63] Tantowi Jauhari, *Jawahir fi Tafsir al-Qur"an al-Karim*, juz 14, (Mesir: Musthofa al-Baaby al-Khaaly wa Awladuhu, tt.), 15.

[64] M. Burhanuddin Latief, "Sistem Pelacak Otomatis Gerakan Benda Langit Pada Teleskop Refraktor Berbasis Mikrokontroler", Jurnal Fisika Indonesia, Vol. 18, No. 54, (Desember 2014), 82.

Pemahaman tentang pergerakan benda langit juga menjadi kebutuhan bagi umat Islam yang dikaji didalam ilmu hisab atau ilmu falak.

1. Metode perhitungan dalam ilmu falak

   Seperti hal nya astronomi, perkembangan ilmu hisab dalam algoritmanya mengalami koreksi untuk menghasilkan perhitungan yang presisi.[65] Dalam perkembangan metode hisab yang dimulai dari metode hisab urfi yang bersifat sederhana, karena hanya berdasarkan rata-rata peredaran dari Bumi, Bulan dan Matahari.[66] Kemudian hisab tersebut mengalami koreksi sampai menjadi hisab hakiki yang lebih teliti dalam perhitungannya dikarenakan mempertimbangkan koreksi disetiap nilai rata-ratanya.[67] Hisab hakiki terbagi menjadi tiga dalam periodenya, yaitu periode metode hisab hakiki takribi, periode hisab hakiki tahkiki dan periode hisab hakiki

[65] M. Yakub Mubarok, "Pemrograman Data Ephemeris Matahari Dan Bulan Berdasarkan Perhitungan Jean Meeus Menggunakan Bahasa Program Php (Personal Homepage Hypertext Preprocessor) Dan Mysql (My Structure Query Language)", *Skripsi,* (Semarang: Institut Agama Islam Negeri Walisongo, 2013), 5.

[66] Asrini, "Studi Komparatif Hisab Kontemporer Ephemeris Dan Algoritma", *Skripsi,* (Makasar: Fakultas Syariah Dan Hukum Uin Alauddin, 2020), 22.

[67] Asrini, "Studi Komparatif Hisab Kontemporer Ephemeris Dan Algoritma"…23.

kontemporer.[68] Terdapat beberapa jenis algoritma yang termasuk ke dalam hisab hakiki kontemporer seperti, ephemeris, algoritma Nautical Almanac, dan algoritma New Comb.[69]

2. Perhitungan data ephemeris metode Jean Meeus

Data ephemeris merupakan data utama yang diperlukan dalam menghitung pergerakan benda langit yang memiliki relefansi terhadap waktu ibadah umat islam.[70] Terdapat beberapa metode yang digunakan dalam menghitung data ephemeris, salah satunya adalah algoritma Jean Meeus yang merupakan masuk dalam kelompok perhitungan akurasi tinggi.[71] Algoritma Jean Meeus merupakan algoritma hasil

---

[68] M. Yakub Mubarok, "Pemrograman Data Ephemeris Matahari Dan Bulan Berdasarkan Perhitungan Jean Meeus Menggunakan Bahasa Program Php (Personal Homepage Hypertext Preprocessor) Dan Mysql (My Structure Query Language)", *Skripsi*, (Semarang: Institut Agama Islam Negeri Walisongo, 2013), 5-6.

[69] Muhammad Al-Farabi Putra, "Algoritma Peter Duffett Smitth Dalam Hisab Awal Bulan Kamariah", Tesis, (Semarang: Fakultas Syariah Dan Hukum Universitas Islam Negeri Walisongo, 2019), 34.

[70] Zul Amri Fathinul Inshafi, "Aplikasi Data Ephemeris Matahari Dan Bulan Berdasarkan Perhitungan Jean Meeus Pada Smartphone Android" *Skripsi*, (Semarang: Universitas Islam Negeri Walisongo, 2013), 8.

[71] M. Yakub Mubarok, "Pemrograman Data Ephemeris Matahari Dan Bulan Berdasarkan Perhitungan Jean Meeus Menggunakan Bahasa Program Php (Personal Homepage Hypertext Preprocessor) Dan Mysql (My Structure Query Language)", *Skripsi*, (Semarang: Institut Agama Islam Negeri Walisongo, 2013), 6.

reduksi dari VSOP87 yang digunakan untuk menentukan posisi matahari.[72] Dalam algoritma Jean Meeus hanya mengambil suku-suku yang besar dan penting saja dari algoritma VSOP 87.[73] Meski demikian, perhitungan posisi benda langit hasil dari algoritma Jean Meeus masih dinilai akurat.[74]

B. Mikrokontroler

Mikrokontroler adalah *chip* yang digunakan dalam aplikasi prosesor mini untuk pengendalian dan otomatisasi skala kecil.[75] Pada dasarnya mikrokontroler merupakan sebuah komputer mini yang juga mempunyai unit pengelola pusat (*Central Processing Unit*) dapat juga menjalankan sebuah program pada ROM (*Read Only Memory*) dan RAM (*Random Access Memory*) serta memiliki beberapa saluran *input output* sebagai komunikasi timbal balik dengan dunila luar, namun

---

[72] Mira Musrini Barmawi, "Implementasi Algoritma Jean Meeus dalam Menentukan Waktu Shalat", *MIND Journal*, Vol. 2, No. 1 , (2017), 26-30. https://doi.org/10.26760/mindjournal

[73] Rinto Anugraha, Mekanika Benda Langit, (Yogyakarta: UGM, 2012), 68.

[74] Zul Amri Fathinul Inshafi, "Aplikasi Data Ephemeris Matahari Dan Bulan Berdasarkan Perhitungan Jean Meeus Pada Smartphone Android", *Skripsi*, (Semarang: Fakultas Syariah Dan Hukum Universitas Islam Negeri Walisongo, 2016), 8.

[75] Iswanto, *Design Dan Implementasi Sisitem Embedded Mikrokontroler ATMega8535 Dengan Bahasa Basic*, (Yogyakarta: Gava Media, 2008), 1.

mikrokontroler hanya dapat digunakan untuk suatu aplikasi saja atau hanya dapat menyimpan satu program saja.[76] Dengan kata lain, mikrokontroler merupakan alat elektronika digital yang mempunyai masukan dan keluaran serta kendali menggunakan program yang dapat ditulis dan dihapus dengan cara tersendiri, cara kerja mikrokontroler sebenarnya membaca dan menulis data.

1. Sejarah Dan Perkembangan Mikrokontroler

   Mikrokontroler pertama kali dikenalkan oleh Texas Instrument dengan seri TMS 1000 yang mulai dibuat pada tahun 1974 yang merupakan mikrokontroler 4 bit pertama dengan sebuah chip, lengkap dengan RAM dan ROM. Pada tahun 1976 Intel mengeluarkan mikrokontroler 8 bit dari keluarga MCS48 yang kelak menjadi populer dengan nama 8748. Sekarang di pasaran banyak sekali dijumpai mikrokontroler mulai dari 8 bit sampai dengan 64 bit, sehingga perbedaan antara mikrokontroler dan mikroprosesor sangat tipis. Masing-masing vendor mengeluarkan mikrokontroler dengan dilengkapi fasilitas-fasilitas yang cenderung memudahkan user

[76] Agfianto Eko Putra, *Belajar Mikrokontroler AT89C51/52/55 (Teori Dan Aplikasi)*, (Yogyakarta: Gava Media, 2010), 2.

untuk merancang sebuah sistem dengan komponen luar yang relatif lebih sedikit.[77]

Mikrokontroller dapat digunakan untuk berbagai aplikasi seperti pengendalian alat, otomasi industri, akuisisi data, telekomunikasi, dan lain-lain. Keuntungan menggunakan mikrokontroller yaitu harganya yang terjangkau, dapat diprogram berulang kali, dan dapat diprogram sesuai dengan kebutuhan. Saat ini keluarga mikrokontroller yang ada dipasaran yaitu intel 8048 dan 8051 (MCS51), Motorola 68HC11, microchip PI,hitachi H8, dan atmel AVR.[78]

2. Arsitektur Mikrokontroler

Berdasarkan arsitekturnya, mikrokontroler dibagi menjadi dua yaitu, CISC (*Complex Instruction Set Computer*) dan RISC (*Reduce Instruction Set Computer*). CISC Mikroprosesor merupakan jenis mikrokontroler yang memiliki jumlah instruksi yang kompleks dan lengkap seperti Intel 80C51 (MCS51) dan Motorola 68HC11 mengikuti arsitektur CISC. Sedangkan RISC merupakan jenis mikrokontroler yang

---

[77] Verdi Yasin, "Rancangan Miniatur Otomatisasi Bel Listrik Pada Gerbang Pintu Menggunakan Microkontroler Atmega8535", *Jisicom,* Vol.3, No.1, 2019, 14.

[78] Sutarsi Suhaeb, *Mikrokontroler Dan Interace,* (Universitas Negeri Makassar, 2017), 2-3.

memiliki jumlah instruksi yang terbatas dan sedikit. Pada arsitektur RISC jumlah instruksi lebih sedikit, namun memiliki banyak register dibandingkan dengan CISC. Selain itu pada arsitektur RISC kebanyakan instruksi dieksekusi hanya dalam satu *clock cycle* dan mode *addressing* memory yang sederhana seperti ATMELAVR, Microchip PIC2/16CXX dan National Semiconductor COP8.[79]

Mikrokontroler memiliki tiga komponen utama, yaitu: unit pengolahan pusat (CPU: Central Processing Unit), memori dan system I/O (Input/output) untuk dihubungkan ke perangkat luar. CPU yang mengatur sistem kerja komputer mikro, dibangun oleh sebuah mikroprosesor. Memori terdiri atas GEPROM untuk menyimpan program dan RAM untukmenyimpan data. Sistem I/O bisa dihubungkan dengan perangkat luar misalnya sebuah keyboard dan sebuah monitor, bergantung pada aplikasinya. Apabila CPU, memori dan sistem I/O dalam sebuah *chip* semikonduktor.[80]

Salah satu mikrokontroler yang banyak diminati saat ini yaitu mikrokotroler AVR. AVR adalah mikrokontroler

[79] Sutarsi Suhaeb, *Mikrokontroler Dan Interace…*6.

[80] Sumarsono, Dwiatmi Wahyu Saptaningtyas, "Pengembangan Mikrokontroler Sebagai Remote Control Berbasis Android", *Jurnal Teknik Informatika*, Vol 11, No. 1 (April 2018); Neliti, 68.

RSIC (*reduce instruction set compute*) 8 bit berdasarkan arsitektur Harvard , yang dibuat oleh Atmel pada tahun 1996. AVR mempunyai kepanjangan *Advanced versatile RSIC atau Alf and Vegards Risc*. AVR mempunyai kelebihan dibandingkan dengan mikrokontroller lain, keunggulan mikrokontroller AVR yaitu kecepatan eksekusi program yang lebih cepat karena sebagian besar instruksi dieksekusi dalam 1 siklus *clock*, lebih cepat dibandingkan dengan mikrokontroller MCS51 yang memiliki arsitektur CISC (*complex insruction set computer*) di mana mikrokontroller MSC51 membutuhkan 12 siklus *clock* untuk mengeksekusi 1 instruksi . selain itu, mikrokontroller AVR juga memiliki fitur yang lengkap (ADC *internal*, EEPROM *internal*, Timer/counter,Watchdog Timer, PWM ,Port I/O, Komunikasi serial, komparator, I2C,dll.), sehingga dengan fasilitas yang lengkap ini, programmer dan desainer dapat menggunakan untuk berbagai aplikasi sistem elektronika seperti robot, otomatis instruksi, peralatan telekomuniksi dan berbagai keperluan lainnya.[81]

[81] Sutarsi Suhaeb, *Mikrokontroler Dan Interace…*7.

Mikrokontroler merupakan sistem mikroprosesor lengkap yang terkandung di dalam sebuah chip. Mikrokontroler berbeda dari mikroprosesor serba guna yang digunakan dalam sebuah PC, karena sebuahmikrokontroler umumnya telah berisikomponen pendukung sistem minimal mikroprosesor, yakni memori dan pemrograman input-output.[82]

Dalam perkembangannya, modul atau minimum sistem dari mikrokontroler dibuat dalam bentuk chip yang lebih memudahkan pengguna untuk menggunakannya. Satu hal yang saat ini sedang atau banyak digemari oleh pengguna mikrokontroler adalah modul Arduino. Arduino tidak hanya digunakan pada tahapan desain, namun sampai produk jadi. Kita dapat berkreasi apapun dengan menggunakan Arduino, seperti aplikasi dalam bidang robotika, atau aplikasi-aplikasi *embedded system* lainnya. Arduino memberikan banyak kemudahan bagi pengguna untuk merealisasikan karya-karyanya. Arduino telah dilengapi dengan sistem IDE (*Intergrated Development Environment*) untuk menuliskan program aplikasi yang

[82] Arif Beni Santoso, Martinus, Sugiyanto, "Pembuatan Otomasi Pengaturan Kereta Api, Pengereman,Dan Palang Pintu Pada Rel Kereta Api Mainan BerbasisMikrokontroler", *Jurnal Fema*, Vol. 1, No.1 (Januari 2013); Neliti, 17.

kita buat. Selain itu, pada board Arduino juga telah dilengkapi dengan bebagai fasilitas sehingga lebih memudahkan para pencinta atau penggunaanya.[83]

Salah satu yang membuat Arduino memikat banya orang adalah karena sifatnya yang open source, baik untuk hardware maupun softwarenya. Diagram rangkaian elektronik Arduino digratiskan kepada semua orang. Orang-orang bebas untuk mengunduh gambarnya, membeli komponen-komponennya, membuat PCB[84] -nya dan merangkainya sendiri tanpa harus membayar kepada para pembuat Arduino. Sama halnya dengan IDE Arduino yang bisa didownload dan diinstal pada semua jenis komputer mulai dari Linux, Windows dan Os secara gratis.[85]

Arduino dikembangkan oleh sebuah tim yang beranggotakan orang-orang dari berbagai belahan dunia. Anggota inti dari tim ini adalah Massimo Bansi

---

[83] Junaidi, Yuliyan Dwi Prabowo, Project Sistem, 3.

[84] PCB adalah singkatan dari *Printed Circuit Board* yang dalam bahasa Indonesia sering diterjemahkan menjadi Papan Rangkaian Cetak atau Papan Sirkuit Cetak, PCB digunakan untuk menghubungkan komponen-komponen elektronika dengan lapisan jalur konduktornya. Lihat dalam Teknik Elektronika, "Pengertian PCB (Printed Circuit Board) dan jeni-jenis PCB", https://teknikelektronika.com/pengertian-pcb-printed-circuit-board-jenis-jenis-pcb/, diakses 19 Maret 2020.

[85] Mada Sanjaya, Dyah Anggraeni, Fikri Ibrahim Nurrahman, *Algortima Arah Kiblat AlBiruni Dalam Kitab Tahdid Nihayat Al-Amakin Listahih Masafat Al-Masakin* (Bandung: Bolabot, 2019), 322.

Milano (Italia), David Cuartielles Malmoe (Swedia), Tom Igoe (Amerika Serikat). Saat ini komunitas Arduino dikembangkan dengan pesat dan dinamis di berbagai belahan Dunia.[86]

[86] Mada Sanjaya, Dyah Anggraeni, Fikri Ibrahim Nurrahman, *Algortima Arah Kiblat AlBiruni Dalam Kitab Tahdid Nihayat Al-Amakin Listahih Masafat Al-Masakin*…322.

BAB III
PERANCANGAN BERBASIS MIKROKONTROLER UNTUK PERHITUNGAN POSISI MATAHARI DAN BULAN BERUPA PROGRAM *ORBITA*

1. **Proses Instalasi Mikrokontroler Arduino ke dalam Laptop**

   Untuk melakukan pemrograman pada Arduino maka perlu menghubungkan antara *board* Arduino dengan komputer atau laptop yang akan digunakan dalam memprogram dengan tujuan untuk memasukan perintah yang telah dibuat ke dalam Arduino. Berikut adalah langkah-langkah untuk melakukan instalasi Arduino IDE:

   1. *Download Software* IDE Arduino

      Download Software IDE Arduino sesuai operating system yang digunakan dalam komputer atau Laptop pada website resmi Arduino, https://www.arduino.cc/en/main/software.

Gambar 3.1
Download Software IDE Arduino

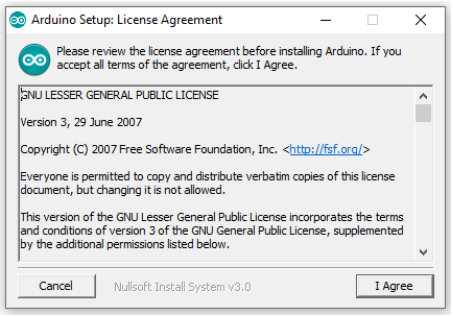
Gambar 3.2
Halaman *Download Software* IDE Arduino

Pada tab ini pilih *Just Download* atau *Contribute and Download* jika ingin berdonasi.

2. Persetujuan *Instalasi Software* IDE Arduino

Setelah selesai *download*. Silahkan buka dengan klik kiri dua kali atau klik kanan open kemudian akan

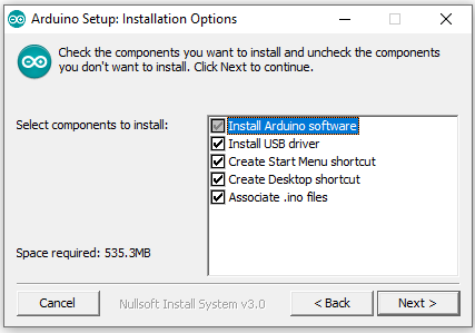
muncul *License Agreement* atau Persetujuan Instalasi, pilih tombol *I Agree* untuk memulai *install software* Arduino IDE.



Gambar 3.3
Proses instalasi *Software* IDE Arduino

3. Pilihan Opsi Instalasi

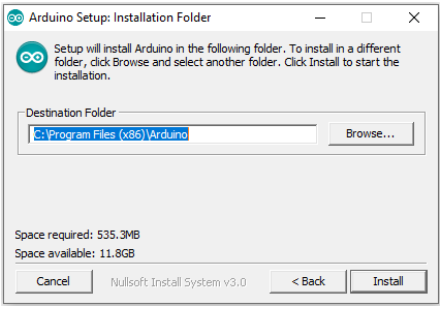
Untuk *Installation Option* centang semua option dan pilih tombol *Next*



Gambar 3.4
Pilihan Instalasi

4. Pilih Folder

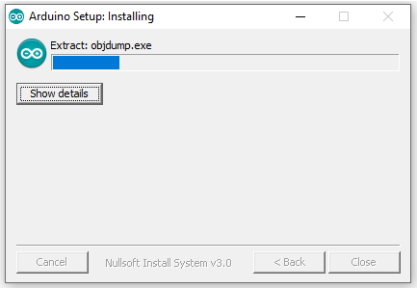
Installation Folder atau Pilihan Folder untuk memilih folder tempat menyimpan program arduino dan pilih tombol *install* untuk memulai proses instalasi *software*.



Gambar 3.5
Pilihan Penyimpanan Folder
Software

5. Proses *Extract* dan Installasi di mulai

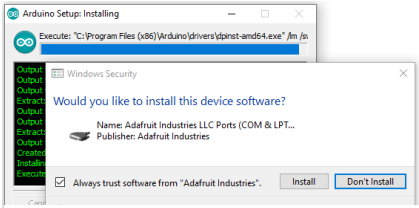
Proses installasi di mulai,program di *extract* ke Windows.



Gambar 3.6
Pemasangan *Software* IDE
Arduino

6. Install USB *Driver*

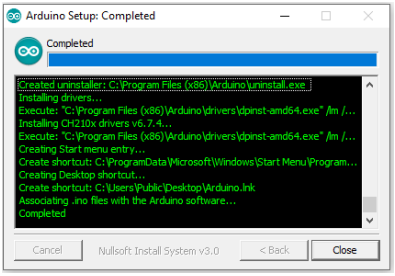
Saat proses instalasi sedang berlangsung akan muncul pilihan untuk install *driver*, pilih tombol install, proses ini untuk mengenali dan melakukan komunikasi dengan *board* arduino melalui *port* USB.



Gambar 3.7
Instalasi USB *Driver*

7. Proses Instalasi Selesai

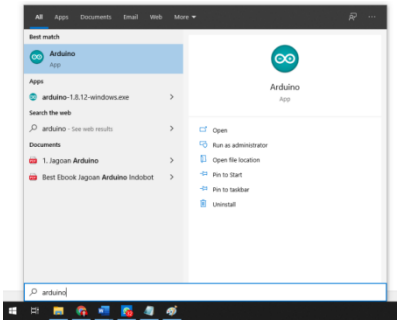
Setelah selesai (*complated*) silahkan pilih *close*.



Gambar 3.8
Instalasi USB *Driver*

8. *Start Software* IDE Arduino

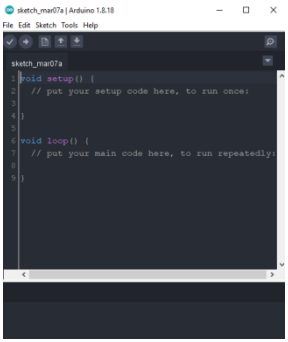
Proses instalasi *Software* Arduino sudah selesai terinstal di windows. Lihat di Desktop Windows atau Start Menu untuk menjalankan *Software* IDE Arduino.Double Klik icon Arduino di *desktop* atau klik di *Start Menu*.



Gambar 3.9
Memulai *Software* IDE Arduino

9. Jendela Awal *Software* IDE Arduino

Proses instalasi arduino selesai, Software IDE Arduino sudah siap untuk digunakan.



Gambar 3.10
Jendela Awal *Software* IDE
Arduino

2. **Penyusunan Program *Orbita***

1. Penulisan Bahasa C Pada Mikrokontroler Arduino

Pada fungsi setiap sintak pada Arduino adalah sebagai berikut:

• ”void setup()”: penanda atau sebagai penunjuk isi dari program dalam pemprograman, isi dari program dimulai dari tanda kurung kurawal ({) dan diakhiri pula dengan tanda tutup kurung kurawal (})

•”void loop()” Secara Prinsip, sketch selalu melibatkan dua fungsi, yaitu setup() dan loop().Baris yang mengandung void dan nama fungsi adalah judul fungsi, sedangkan bagian { } dinamakan tubuh fungsi. Semua definisi fungsi melibatkan judul fungsi dan tubuh fungsi. Fungsi sendiri adalah deretan instruksi yang diberi suatu anam, umumnya fungsi memberikan nilai ketika dipanggil. Nilai yang diberikan tersebut dinamakan nilai balik.

2. Tipe Data

Dalam Pemprograman Mikrokontroller, ram rom maupun register harus diisi data untuk menjalankan suatu program tertentu, data yang diisi pun beragam

tergantung besar bit data yang digunakan. Jenis data tersebut dapat dilihat di tabel berikut.

Tabel 3.1
Tipe Data IDE Arduino

Tipe	Ukuran (Bit)	Range
bit	1	0-1
char	8	-128 – 127
unsigned char	8	0-125
signed char	8	-128 – 127
int	16	-32768 – 32767
short int	16	-32768 – 32767
unsigned int	16	0 – 65535
signed int	16	-32768 - 32767
long int	32	-2147483648 - 21474836467
unsigned long int	32	0 - 4294967295
signed long int	32	-2147483648 - 21474836467
float	32	1.175e-38 - 3.402e38
double	32	1.175e-38 - 3.402e38

3. Operator

Operator adalah *source code* yang digunakan dalam mengelolah data baik itu membandingkan, menyamakan dan menjumlahkan.

Tabel 3.2
Operator Kondisi IDE Arduino

Operator Kondisi	Keterangan
<	lebih kecil
<=	lebih kecil atau sama dengan
>	lebih besar
>=	lebih besar atau sama dengan
==	sama dengan
!=	tidak sama dengan

Tabel 3.3
Operator Aritmatika IDE Arduino

Operator Aritmatika	Keterangan
+	Penjumlahan
-	Pengurangan

*	Perkalian
/	Pembagian
%	Sisa Bagi (Modulus)

Tabel 3.4
Operator Logika IDE Arduino

Operator Logika	Keterangan
!	Boolean NOT
&&	Boolean AND
——	Boolean OR

Tabel 3.5
Operator Bitwes IDE Arduino

Operator Bitwes	Keterangan
~	komplement betwise
&	Bitwisw AND
—	Bitwise OR
^	Bitwise Exclusive OR
>>	Right Shift
<<	Left Shift

Tabel 3.6
Operator Assigment IDE Arduino

Operator Assignment	Keterangan
=	Memasukkan nilai
+=	Menambahkan nilai dari keadaan semula
-=	Mengurangi nilai dari keadaan semula
*=	Mengalikan niliai dari keadaan semula
/=	Membagi nilai dari keadaan semula
%=	Mamasukkan nilai sisa dari pembagian
<<=	Memasukkan Shift Left
>>=	Memasukkan Shift Right
&=	Memasukkan Bitwise AND
\=	Memasukkan Bitwise OR

BAB IV
ANALISIS TERHADAP *ORBITA* SEBAGAI PROGRAM BERBASIS MIKROKONTROLER

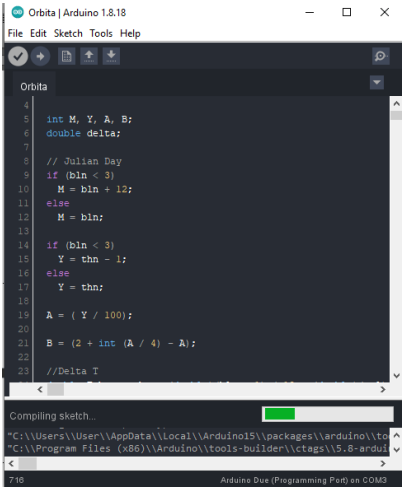
A. Pengujian Program

1. Uji Penggunaan Program *Orbita*

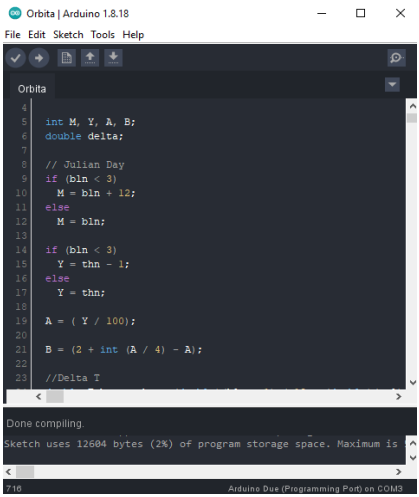
   Uji coba berfungsi untuk menguji kefektifan dari program *Orbita*, kemudian akan dilakukan evaluasi terhadap kesalahan maupun kekurangan pada program.

   Pengujian yang dilakukan pada program *Orbita* meliputi *Compile Sketch* dan *upload* program pada mikrokontroler Arduino.

   a. *Compile Sketch* Program *Orbita*



Gambar 4.1
Proses *Compile Sketch*



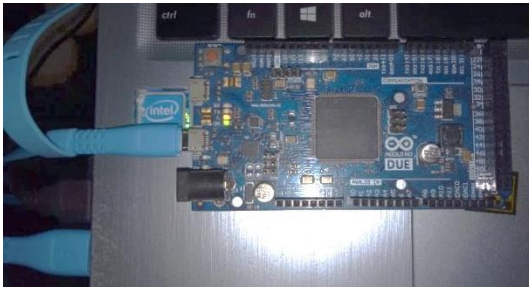
Gambar 4.2
Compile Sketch

Jika terdapat keterangan *Done Compiling* maka artinya *syntax* dalam program sudah benar.

2. Uji *Upload* Program *Orbita*

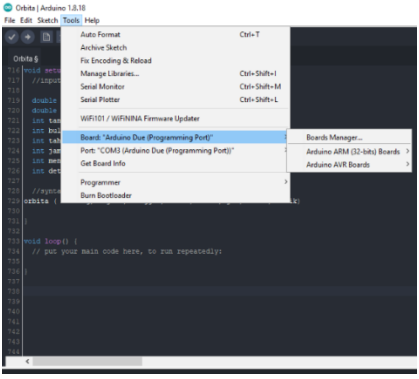
Dalam uji coba ini, penulis akan mencoba meng*input* data untuk diproses program *Orbita* menjadi data Matahari dan Bulan. Hasil perhitungan akan ditampilkan dalam fitur *Serial Monitor* yang terdapat pada *Software* Arduino IDE.

a. Menghubungkan Mikrokontroler Arduino dengan Laptop



Gambar 4.3
Menghubungkan Mikrokontroler
Arduino dengan Laptop

b. Buka *software* Arduino IDE, dan pilih jenis Arduino yang digunakan pada menu *Tools-Board*



Gambar 4.4
Tools-Board

Pada langkah ini, penulis menggunakan *board* Arduino Due.

c. Tulis sketch yang akan di *Upload* pada mikrokontroler Arduino

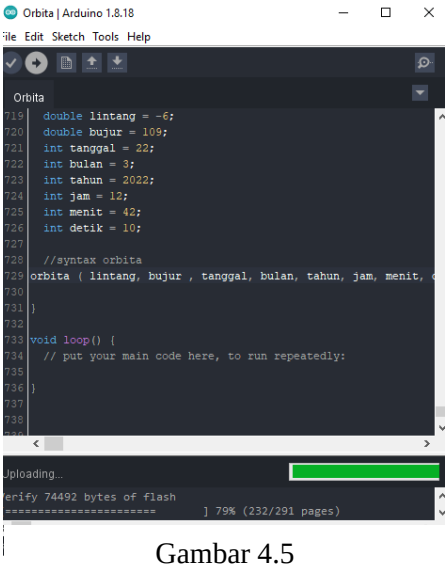
```
void setup() {
   Serial.begin(9600);
     //input data

     double lintang = -6;
     double bujur = 109;
     int tanggal = 22;
     int bulan = 3;
     int tahun = 2022;
     int jam = 12;
     int menit = 42;
     int detik = 10;

     //syntax orbita
   orbita ( lintang, bujur,
   tanggal, bulan, tahun,
   jam, menit, detik);
   }
   void loop() {}}
```

contoh sederhana dari *sketch* untuk memproses data Matahari dan Bulan pada program *Orbita*.

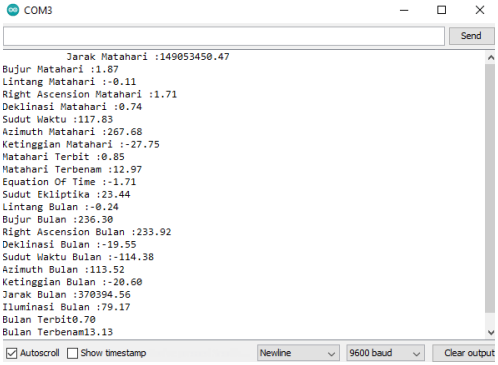
d. Klik *Upload* untuk mengunggah program



Gambar 4.5
Proses *Upload*

Tunggu proses *upload* sampai selesai.

e. Buka jendela *Serial Monitor* .



Gambar 4.6
Serial Monitor

Tampilan atau hasil dari pemrograman dapat dilihat pada *Serial Monitor*.

3. Uji *Input* Data Program

Uji *input* data digunakan untuk mengetahui seberapa lama kurun waktu yang dapat diproses program dengan hasil nilai yang normal. Selain itu, dalam tahap ini juga menguji penggunaan program yang memakai *input* modul GPS.

Dalam pengujian tahap ini, nilai dari hasil program *Orbita* dibandingkan dengan nilai dari perhitungan posisi Matahari dan Bulan berbasis excel karya Rinto Anugraha yang memiliki algoritma sama yaitu Jean Meeus, namun berbeda dalam koreksi Delta-T-nya. Selain itu, perbandingan nilai juga menggunakan aplikasi Stellarium yang mempunyai konsep serupa serta dapat menjangkau tahun minus sampai lebih dari 3000 tahun kedepan. Nilai yang dibandingkan mengambil sempel data azimuth serta ketinggian Matahari dan Bulan.

Tabel 4.1
Perbandingan *Output* Program *Orbita*
menggunakan modul GPS di lokasi luar dan
dalam ruangan

Data	Luar Ruangan	Dalam Ruangan
Satelit	Terdeteksi	Tidak Terdeteksi
Waktu	21:45:59	00:00
Bujur Matahari	79° 43' 20"	247° 19' 52"
EOT	00:28	11:38

Tabel 4.2
Komparasi Program *Orbita* dengan Excel Rinto
Anugraha dan Stellarium
Tanggal 1 Januari -1 pukul 00:00 UT.

Data	*Orbita*	Excel	Stellarium
Matahari			
Azimuth	112° 27' 51.1"	112° 42' 43"	112° 25' 13"
Ketinggian	18° 47' 44.7"	18° 59' 15"	14° 47' 13,8"
Bulan			
Azimuth	124° 5' 36.73"	164° 30' 19"	145° 8' 39,9"
Ketinggian	48° 37' 58.28"	77° 4' 39"	75° 47' 5,8"

Tabel 4.3
Komparasi Program *Orbita* dengan Excel Rinto
Anugraha dan Stellarium
Tanggal 1 Januari 3000 pukul 00:00 UT.

Data	*Orbita*	Excel	Stellarium
Matahari			
Azimuth	112° 15' 4.30"	112° 15' 6"	111° 54 5,1"

Ketinggian	20° 27' 5.26"	20° 27' 25"	20° 50' 57"
Bulan			
Azimuth	103° 19' 15.4"	103° 19' 47"	103° 31' 33"
Ketinggian	-13° 0' 34.98"	-12° 56' 37"	-13° 48' 1,3"

Tabel 4.4
Komparasi Program *Orbita* dengan Excel Rinto Anugraha dan Stellarium
Tanggal 31 Desember 2005 pukul 06:00 UT.

Data	***Orbita***	**Excel**	**Stellarium**
Matahari			
Azimuth	225° 23' 54.5"	225° 23' 54"	227° 4' 12,2"
Ketinggian	64° 46' 30.2"	64° 46' 30"	65° 24' 17,2"
Bulan			
Azimuth	214° 26' 13.8"	214° 26' 13"	215° 38' 1"
Ketinggian	62° 44' 46.58"	62° 44' 46"	63° 5' 6,3"

4. Uji Akurasi Program *Orbita*

Metode yang digunakan untuk menguji tingkat keakuratan perhitungan pada program *Orbita* yakni dengan cara mengkomparasikan dengan program perhitungan lainnya yang serupa.

Di tahap uji akurasi, penulis melakukan 3 kali proses uji coba perhitungan data yang dilakukan pada waktu, tanggal, bulan dan tahun yang berbeda. Adapun uji coba perhitungan masing-masing diterapkan pada waktu dan tanggal berikut ini

1. Tanggal 1 April 2022 pukul 17:46 WIB,
2. Tanggal 1 Januari 2100 pukul 07:00 WIB, dan
3. Tanggal 31 Desember 2005 pukul 13:00 WIB.

Uji coba pada tahun 2100 dipilih karena pada tahun tersebut merupakan tahun keseratus setelah epoch pada sistem perhitungan J-2000.[87] Selain itu pada tanggal 1 April tahun tersebut merupakan akhir bulan Sya'ban sebagai penentuan awal bulan Ramadhan, sehingga Bulan berada di posisi ufuk barat yakni ketika ketinggian mendekati nilai 0°. Adapun alasan lain pemilihan tahun 2100 karena tahun tersebut merupakan tahun ekstrim untuk perhitungan equation of time di mana pada tahun tersebut equation of time mencapai nilai minimum sekitar -14m 03d.[88]

Uji coba pada tanggal 31 Desember 2005 merupakan waktu dimana transisi dari nilai Delta-T.

[87] Rinto Anugraha, *Mekanika Benda Langit*,.. 59-66.
[88] Rinto Anugraha, *Mekanika Benda Langit*, 175.

Tabel 4.5
Komparasi Program *Orbita* Dengan Aplikasi Daff Moon dan Stellarium
Tanggal 1 April 2022 pukul 17:46 WIB.

Data	*Orbita*	Daff Moon	Stellarium
Matahari			
Jarak	149483716,1	149483129,4	149597871
Bujur	11° 41' 18.24"	11° 41' 19"	-
Lintang	0° 27' 37.48"	-0° 0' 1"	-
Right. A	10° 44' 50.93"	10° 44' 0"	10° 44' 0"
Deklinasi	4° 37' 18.70"	4° 37' 19"	4° 37' 19"
Sudut Waktu	90° 31' 45.74"	-	90° 33' 0"
Azimuth	274° 32' 30.70"	274° 31' 32"	274° 31' 13,7"
Ketinggian	-1° 0' 26.63"	-1° 9' 59"	-1° 6' 53,4"
EOT	0° 0' 9.70"	-	-
Ekliptika	23° 26' 17.01"	-	-
Bulan			
Lintang	-3° 15' 43.83"	-3° 15' 41"	-
Bujur	13° 53' 5.93"	13° 53' 09"	-
Right. A	14° 2' 40.42"	13° 2' 0"	13° 6' 0"
Deklinasi	2° 28' 10.05"	2° 28' 14"	2° 35' 6,6"
Sudut Waktu	87° 13' 56.25"	-	88° 10' 0"
Azimuth	272° 44' 51.88"	272° 44' 29"	272° 47' 23"
Ketinggian	2° 29' 29.96"	1° 29' 11"	1° 29' 54"
Jarak	386930,5862	386922	386711,89
Iluminasi	0,11%	0%	-

Tabel 4.6
Komparasi Program *Orbita* Dengan Aplikasi Daff Moon dan Stellarium
Tanggal 1 Januari 2100 pukul 07:00 WIB

Data	*Orbita*	Daff Moon	Stellarium
Matahari			
Jarak	147108351,8	147108113,6	146605913,6
Bujur	280° 36' 20.94"	280° 36' 19"	-
Lintang	0° 16' 46.77"	0° 0' 0"	-
Right. A	281° 32' 3.25"	281° 31' 0"	281° 31' 0"
Deklinasi	-23° 0' 18.42"	-23° 00' 19"	-23° 0' 18"
Sudut Waktu	289° 12' 17.41"	-	289° 23' 0"
Azimuth	112° 19' 45.51"	112° 19' 52"	111° 59' 31"
Ketinggian	19° 59' 50.68"	20° 2' 41"	20° 32' 37"
EOT	0° 0' 7.93"	-	-
Ekliptika	23° 25' 43.20"	-	-
Bulan			
Lintang	1° 5' 21.44"	1° 5' 22"	-
Bujur	157° 26' 3.29"	157° 26' 3"	-
Right. A	159° 32' 19.97"	159° 32' 0"	158° 45' 0"
Deklinasi	9° 47' 15.05"	9° 47' 16"	10° 0' 46"
Sudut Waktu	51° 12' 0.70"	-	52° 2' 0"
Azimuth	286° 55' 7.01"	286° 54' 42"	287° 36' 15"
Ketinggian	36° 36' 26.77"	35° 47' 28"	35° 26' 48,6"
Jarak	371712,6601	371701	367945,11
Iluminasi	77,44%	77,4%	-

Tabel 4.7
Komparasi Program *Orbita* Dengan Aplikasi Daff Moon dan Stellarium
Tanggal 31 Desember 2005 pukul 13:00 WIB.

Data	*Orbita*	Daff Moon	Stellarium
Matahari			
Jarak	147109408,4	147109160,8	146605913,6
Bujur	279° 39' 23.58"	279° 39' 23"	-
Lintang	0° 38' 21.42"	0° 0' 0"	-
Right. A	280° 30' 23.84"	280° 30' 0"	280° 30' 0"
Deklinasi	-23° 5' 22.11"	-23° 5' 2"	-23° 5' 24,3"
Sudut Waktu	19° 15' 37.51"	-	19° 20' 0"
Azimuth	225° 23' 54.51"	225° 25' 0'	227° 4' 12,2"
Ketinggian	64° 46' 30.27"	64° 45' 52"	65° 24' 17,2"
EOT	0° 0' 7.39"	-	-
Ekliptika	23° 26' 26.98"	-	-
Bulan			
Lintang	-4° 59' 39.55"	-4° 59' 40"	-
Bujur	281° 15' 42.35"	281° 15' 52"	-
Right. A	282° 43' 18.51"	282° 43' 0"	282° 23' 0"
Deklinasi	-27° 56' 20.86"	-27° 56' 21"	-28° 16' 58"
Sudut Waktu	17° 2' 42.84"	-	17° 25' 0"
Azimuth	214° 26' 13.85"	214° 30' 25"	215° 38' 09,9"
Ketinggian	62° 44' 46.58"	62° 15' 6"	63° 5' 6,3"
Jarak	36.406.947.057	364058	358373,09
Iluminasi	0,21%	0%	-

5. Evaluasi Hasil Pengujian Program *Orbita*

Dari hasil uji penggunaan, diketahui bahwa program *Orbita* dapat berjalan dengan baik pada mikrontroler Arduino Due.

Dalam uji *input*, program tersebut dapat menjangkau perhitungan dari tahun -1 sampai tahun 3000. Namun hasil yang dibandingkan dengan program lainnya memiliki selisih yang cukup besar pada tahun -1 dan 3000, ini disebakan dari algoritma program *Orbita* yang kurang kompleks karena hanya mengoreksi nilai Delta-T dari tahun 2005-2150 saja. Sedangkan pada pengaplikasian program menggunakan *input* modul GPS nilai yang dihasilkan tidak janggal atau normal ketika modul berada di luar ruangan, namun ketika modul berada di dalam ruangan nilai yang dihasilkan terlihat ganjal atau abnormal.

Pada tahap uji akurasi program tersebut yang dibandingkan dengan program Daff Moon dan Stellarium menghasilkan data yang hanya berselisih kurang dari 1 derajat, serta terdapat beberapa data pada Stellarium selisihnya tidak lebih dari 2 derajat. Selisih hasil perhitungan tersebut terjadi karena algoritma sampai pembulatan angka dibelakang

koma. Ini terjadi karena terdapat beberapa algoritma untuk perhitungan data posisi Matahari dan Bulan serta basis pembulatan dari program *Orbita* yang berbahasa program C++ berbeda dengan pembulatan pada aplikasi lainnya.

6. Penjelasan Hasil Pengujian Program *Orbita*

   Dari beberapa uji coba yang telah dilakukan pada program *Orbita* di atas, diketahui bahwa program dapat digunakan sebagai referensi data untuk perhitungan pada Ilmu Falak. Nilai yang dihasilkan pada program tersebut masih cukup akurat untuk digunakan pengembangan instrumen falak berbasis elektro dan komputer, karena selisih nilai dengan program lain tidak terlalu besar. Sedangkan pada penggunaan modul GPS sebagai *input* data, hasil program masih normal jika sensor GPS memiliki sensitf yang kuat untuk dapat menghasilkan data yang benar.

B. Hasil Perancangan Program *Orbita*

Penelitian tersebut menghasilkan produk program penghitung data posisi Matahari dan Bulan berbasis mikrokotroler yang dapat dijalankan dengan baik pada Arduino Due. Program perhitungan tersebut disusun menggunakan algoritma Jean Meeus yang didapatkan dari

buku Mekanika Benda Langit karya Rinto Anugraha. Program tersebut berbentuk *sketch* berbahasa C++ yang dapat digunakan sebagai referensi pengembangan instrumen falak berbasis mikrokontroler.

BAB V
PENUTUP

A. Kesimpulan

Dari penelitian penulis yang berjudul "*Perancangan Program Perhitungan Data Posisi Matahari Dan Bulan Berbasis Mikrokontroler"* dihasilkan dua kesimpulan. Kedua kesimpulan tersebut yakni:

1. Proses perancangan program *Orbita* dengan menggunakan bahasa program C++ melalui beberapa tahapan, yakni: studi literatur dan pengumpulan data; perancangan perangkat lunak; dan implementasi perangkat lunak. Pada tahap studi literatur dan pengumpulan data didapatkan bahwa algoritma yang digunakan dan teknik pemrograman berbasis IDE (*Integrated Development Environment*) pada mikrokontroler layak dijadikan referensi utama dalam penyusunan program ini. Pada tahap perancangan perangkat lunak, penulis merancang alur

pemrograman sesuai dengan alur perhitungan posisi Matahari dan Bulan.

Tahap terakhir yakni tahap implementasi program. Pada tahap ini seluruh rancangan yang dibuat sebelumnya diimplementasikan ke dalam bahasa pemrograman C++. Alur perhitungan juga disusun menggunakan bahasa pemrograman yang sama.

2. Hasil uji coba penggunaan program yang dilakukan penulis terhadap program *Orbita* menghasilkan kesimpulan bahwa program tersebut dapat berjalankan dengan baik pada mikrokontroler Arduino DUE. program ini merupakan program yang berbasis offline, maka untuk menjalankannya tidak diperlukan ketersediaan akses internet pada mikrokontroler. Adapun hasil uji *input* data pada program, program masih dapat menjangkau perhitungan pada tahun 2005 sampai 2100. Sedangkan pada uji akurasi program dengan membandingkan hasil dari aplikasi Daff Moon dan Stellarium menghasilkan selisih yang tidak terlalu besar, yaitu kurang dari 1 derajat, serta terdapat beberapa

data pada Stellarium selisihnya tidak lebih dari 2 derajat.

B. Saran-saran

Setelah melakukan penelitian dari tahap perancangan, pengimplementasian dan pengujian terhadap program *Orbita*, penulis mempunyai beberapa saran, diantaranya:

1. Program *Orbita* dapat dijadikan alternatif pengambilan data ephemeris Matahari dan Bulan untuk perhitungan falak. Hal ini cukup mempermudah untuk pengembangan instrumen falak berbasis digital. Namun algoritma pada program tersebut hanya dapat menjangkau tahun 2005 sampai 2150 untuk hasil nilai yang akurat, sehingga perlu penambahan koreksi pada program tersebut supaya lebih kompleks.
2. Program ini menggunakan metode perhitungan algoritma Jean Meeus high accuracy. Namun demikian tingkat keakuratan hasil perhitungan masih dapat ditingkatkan lagi dengan menggunakan metode perhitungan data ephemeris yang lebih tinggi akurasinya seperti VSOP87 dan ELP2000/82.

C. Penutup

Segala puji bagi Allah yang dengan kehendak-Nya segala sesuatu dapat berakhir dengan baik. Oleh karena itu,

penulis bersyukur atas segala kenikmatan-Nya sehingga dapat menyelesaikan skripsi ini.

Penulis telah berusaha dengan sebaik-baiknya dalam proses pengerjaan skripsi ini. Namun hal tersebut tentunya tidak serta merta menutup kemungkinan masih adanya kesalahan pada hasil penelitian penulis. Oleh karena itu, penulis sangat berharap atas saran dan kritik dari para pembaca, khususnya untuk penelitian dan pengembangan *program Orbita* ke depan.

Pada akhirnya, penulis berharap hasil penelitian penulis ini dapat bermanfaat bagi diri penulis sendiri, komunitas falak di Indonesia dan umat Islam secara keseluruhan. Selain itu, penulis juga berharap penelitian penulis dapat menjadi sumbangasih pada khazanah keilmuan falak di Indonesia pada umumnya dan Prodi Ilmu Falak Fakultas Syariah dan Hukum UIN Walisongo pada khususnya. *Amin.*

DAFTAR PUSTAKA

A. Gunawan Admiranto, *Eksplorasi Tata Surya*, (Bandung: PT Mizan Pustaka, 2017)

Aldyrazor.com, "Software Arduino IDE: Cara Download, Instal, dan Fungsinya", https://www.aldyrazor.com/2020/05/software-arduino-ide.html, 31 Desember 2021.

Anugraha Rinto, *Mekanika Benda Langit*, (Yogyakarta: UGM, 2012).

Asrini, "Studi Komparatif Hisab Kontemporer Ephemeris Dan Algoritma", *Skripsi*, (Makasar: Fakultas Syariah Dan Hukum Uin Alauddin, 2020).

Barmawi Mira Musrini, "Implementasi Algoritma Jean Meeus dalam Menentukan Waktu Shalat", *MIND Journal*, Vol. 2, No. 1 , (2017). https://doi.org/10.26760/mindjournal

Bayong Tjasyono, *Ilmu Kebumian dan Antariksa*, (Bandung: Pascasarjana UPI, 2009).

Beetrona, Tutorial Menggunakan Arduino Due, https://beetrona.com/tutorial-arduino-Due/, 25 Januari 2022.

Fajrullah, "Konstruksi Instrumen Moon Verificator Dan Tingkat Akurasinya Dalam Rukyatulhilal", *Skripsi*, (Semarang: Fakultas Syari'ah Dan Hukum Universitas Islam Negeri Walisongo, 2020).

Google Terjemahan, https://translate.google.com/?sl=ru&tl=id&text=%D0%BE%D1%80%D0%B1%D0%B8%D1%82%D0%B0&op=translate, 7 Januari 2022.

Hambali Slamet, *Pengantar Ilmu Falak*, (Banyuwangi: Bismillah Publisher, 2012).

Hardani dkk, *Metode Penelitian Kualitatif & Kuantitatif*, (Yogyakarta: CV. Pustaka Ilmu, 2020), 247.

Harumy T. Henny Febriana, Belajar Dasar Algoritma dan Pemrograman C++, (Medan, 2017).

http://neo.jpl.nasa.gov/glossary/au.html diakses pada hari Sabtu, 5 Maret 2022.

http://rinto.staff.ugm.ac.id/. 11 Maret 2022, Lihat juga pada https://www.facebook.com/rinto.anugraha, 11 Maret 2022.

Inshafi Zul Amri Fathinul, "Aplikasi Data Ephemeris Matahari Dan Bulan Berdasarkan Perhitungan Jean Meeus Pada Smartphone Android" *Skripsi*, (Semarang: Fakultas Syariah Dan Hukum Universitas Islam Negeri Walisongo, 2016).

Iswanto, *Design Dan Implementasi Sisitem Embedded Mikrokontroler ATMega8535 Dengan Bahasa Basic*, (Yogyakarta: Gava Media, 2008).

Izzudin Ahmad, *Ilmu Falak Praktis*, (Semarang: PT. Pustaka

Rizki Putra, 2017).

Jauhari Tantowi, *Jawahir fi Tafsir al-Qur"an al-Karim*, juz 14, (Mesir: Musthofa al-Baaby al-Khaaly wa Awladuhu, tt.).

Junaidi, Yuliyan Dwi Prabowo, *Project Sistem.*

Kamus Besar Bahasa Indonesia (KBBI), https://www.kbbi.web.id/orbit, 7 Januari 2022.

Kantor Kementrian Agama Kabupaten Kulon Progo, https://kemenagkulonprogo.com/index.php/bimas-islam/, 4 Januari 2022.

Kasim A. Jusran, "Tingkat Akurasi Aplikasi Azimuth Matahari Pada Google Play Store (Analisis Perbandingan Rasd Al-Qiblah Harian Sistem Hisab Data Ephemeris Dan Software Hisab Komputer)", *ELFALAKY* , Vol. 4, No 2, (2020).

Khazin Muhyiddin, *Ilmu Falak Dalam Teori dan Praktik*, (Yogyakarta: Buana Pustaka, 2008).

Latief M. Burhanuddin, "Sistem Pelacak Otomatis Gerakan Benda Langit Pada Teleskop Refraktor Berbasis Mikrokontroler", *Jurnal Fisika Indonesia*, Vol 18, No: 54, (Desember 2014).

Lembaga Falakiyah Pengurus Besar Nahdlatul Ulama, Sambutan Ketua Lembaga Falakiyah PBNU, http://falakiyah.nu.or.id/Sambutan.aspx, 4Januari 2022.

Mubarok M. Yakub, "Pemrograman Data Ephemeris Matahari

Dan Bulan Berdasarkan Perhitungan Jean Meeus Menggunakan Bahasa Program Php (Personal Homepage Hypertext Preprocessor) Dan Mysql (My Structure Query Language)", *Skripsi*, (Semarang: Institut Agama Islam Negeri Walisongo, 2013).

Priyono, *Metode Penelitian Kuantitatif*, (Sidoarjo: Zifatama Publishing, 2016).

Putra Agfianto Eko, *Belajar Mikrokontroler AT89C51/52/55 (Teori Dan Aplikasi)*, (Yogyakarta: Gava Media, 2010).

Putra Muhammad Al-Farabi, "Algoritma Peter Duffett Smitth Dalam Hisab Awal Bulan Kamariah", *Tesis*, (Semarang: Fakultas Syariah Dan Hukum Universitas Islam Negeri Walisongo, 2019).

Rachim Abdur, *Ilmu Falak*, (Yogyakarta: Penerbit Liberty, 1983).

Raharto Moedji, *Sistem Penanggalan Syamsiah/Masehi*, (Bandung: Penerbit ITB, 2001).

Rahman Afif Aulia, "Sistem Otomatisasi Pelacakan Objek Astronomi Menggunakan Teleskop Berdasarkan Stellarium", *Skripsi*, (Surabaya: Fakultas Teknologi Industri Institut Teknologi Sepuluh Nopember, 2016).

Revolusitekno, Pengertian Personal Computer , Sejarah dan Komponennya, https://www.revolusitekno.com/1036/pengertian-personal-computer-sejarah-dan-komponennya.html,

5 Januari 2022.

Sanjaya Mada, Dyah Anggraeni, Fikri Ibrahim Nurrahman, *Algortima Arah Kiblat AlBiruni Dalam Kitab Tahdid Nihayat Al-Amakin Listahih Masafat Al-Masakin* (Bandung: Bolabot, 2019).

Santoso Arif Beni, Martinus, Sugiyanto, "Pembuatan Otomasi Pengaturan Kereta Api, Pengereman,Dan Palang Pintu Pada Rel Kereta Api Mainan BerbasisMikrokontroler", *Jurnal Fema*, Vol. 1, No.1 (Januari 2013).

Solikin Imam, "Implementasi Penggunaan Smartphone Android untuk Control PC (Personal Computer)", *Jurnal Informatika*, Vol.03, No.02, (Mei 2018).

Sugiyono, *Metode Penelitian Kuantitatif, Kualitatif Dan R & D*, (Bandung: Alfabeta, 2013).

Suhaeb Sutarsi, *Mikrokontroler Dan Interace*, (Universitas Negeri Makassar, 2017).

Sumarsono, Dwiatmi Wahyu Saptaningtyas, "Pengembangan Mikrokontroler Sebagai Remote Control Berbasis Android", *Jurnal Teknik Informatika*, Vol 11, No. 1 (April 2018); Neliti.

Syahrum, Salim, *Metodologi Penelitian Kuantitatif*, (Bandung: Citapustaka Media, 2012).

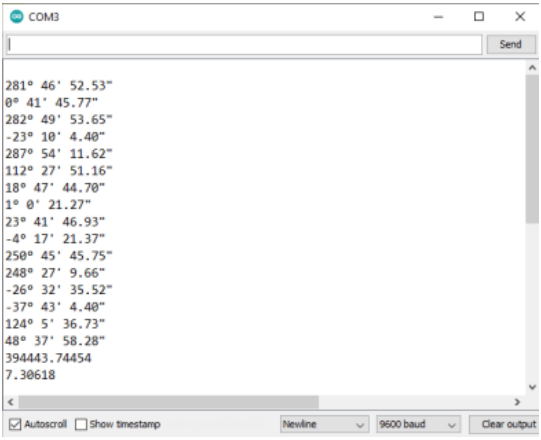
Teknik Elektronika, "Pengertian PCB (Printed Circuit Board) dan jeni-jenis PCB",

https://teknikelektronika.com/pengertian-pcb-printed-circuit-board-jenis-jenis-pcb/, 19 Maret 2020.

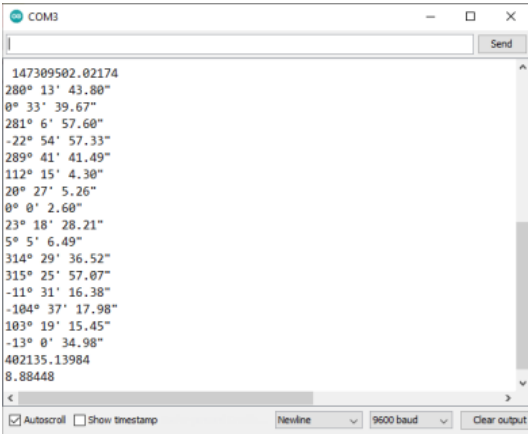
Tim Penerjemah, *Al-Qur'an dan Terjemahannya* (Terjemahan Departermen Agama RI), Bandung: CV Diponegoro, tt..

Verdi Yasin, "Rancangan Miniatur Otomatisasi Bel Listrik Pada Gerbang Pintu Menggunakan Microkontroler Atmega8535", *Jisicom*, Vol.3, No.1,(2019).

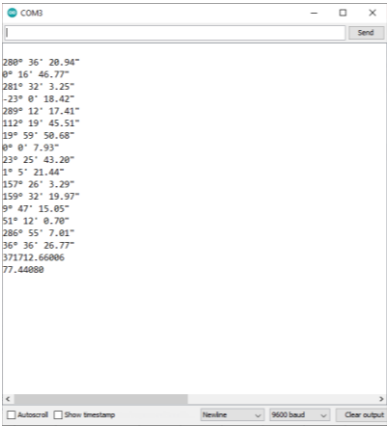
LAMPIRAN



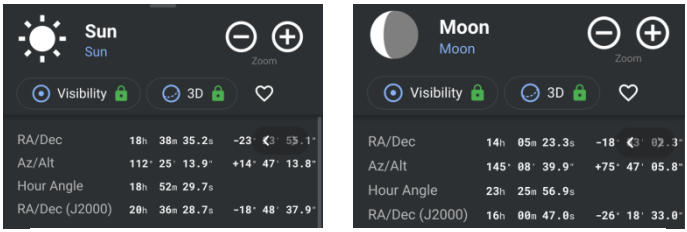
Data Matahari dan Bulan program *Orbita*
Tanggal 1 Januari -1 pukul 00:00 UT.



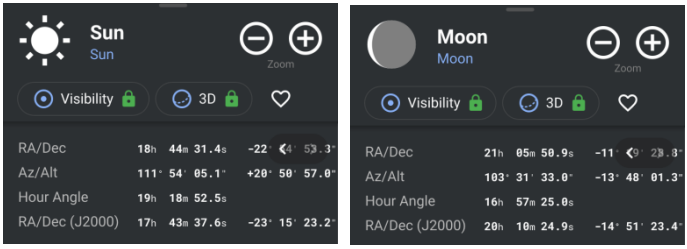
Data Matahari dan Bulan program *Orbita*
Tanggal 1 Januari 3000 pukul 00:00 UT.



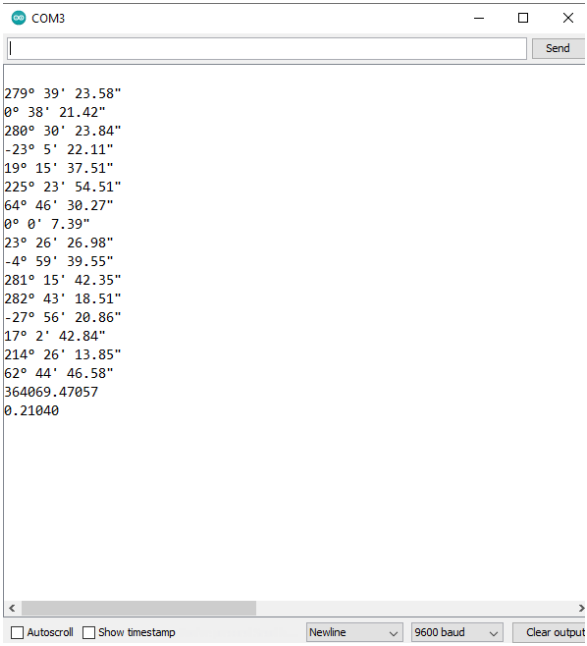
Data Matahari dan Bulan program *Orbita*
Tanggal 1 Januari 2100 pukul 17:00 WIB.



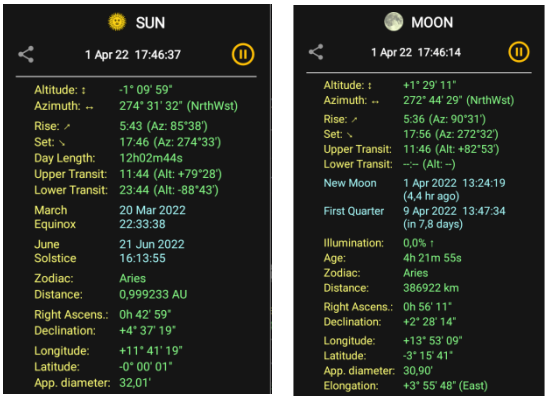
Data Matahari dan Bulan Aplikasi Stellarium
Tanggal 1 Januari -1 pukul 00:00 UT.



Data Matahari dan Bulan Aplikasi Stellarium
Tanggal 1 Januari 3000 pukul 00:00 UT.

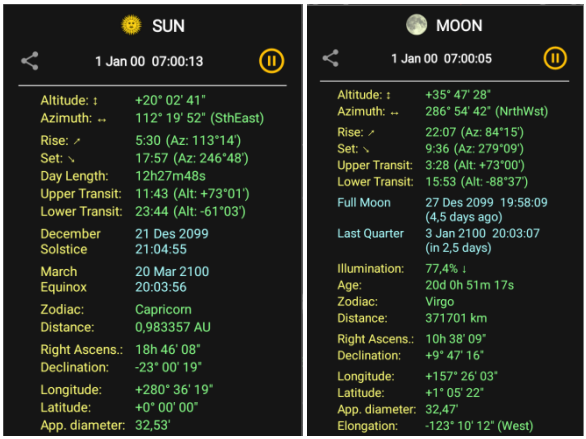


Data Matahari dan Bulan program *Orbita*
Tanggal 31 Desember 2005 pukul 13:00 WIB.

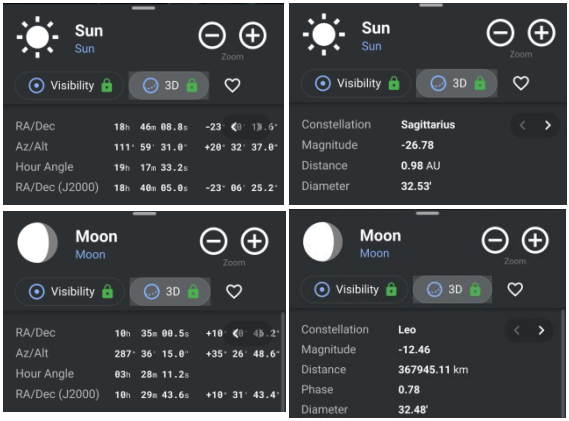


Data Matahari dan Bulan Aplikasi Daff Moon
Tanggal 1 April 2022 pukul 17:46 WIB.

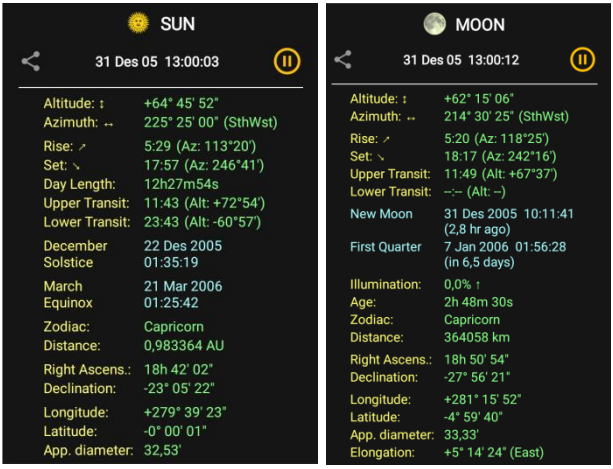
Data Matahari dan Bulan Aplikasi Stellarium
Tanggal 1 April 2022 pukul 17:46 WIB.



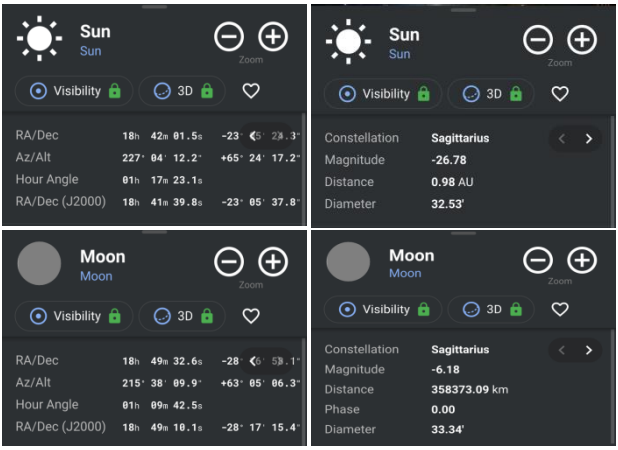
Data Matahari dan Bulan Aplikasi Daff Moon
Tanggal 1 Januari 2100 pukul 17:00 WIB.



Data Matahari dan Bulan Aplikasi Stellarium
Tanggal 1 Januari 2100 pukul 17:00 WIB.

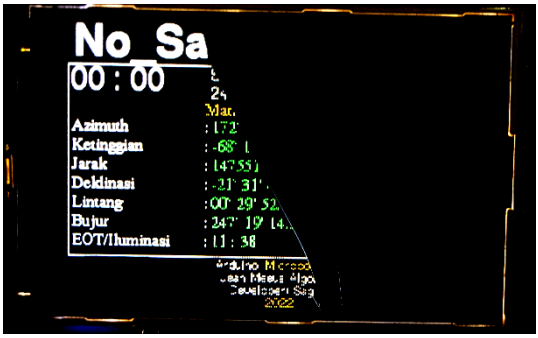


Data Matahari dan Bulan Aplikasi Daff Moon
Tanggal 31 Desember 2005 pukul 13:00 WIB.



Data Matahari dan Bulan Aplikasi Stellarium
Tanggal 31 Desember 2005 pukul 13:00 WIB.

Output LCD Program *Orbita* di luar ruangan Tanggal 10 Juni 2022 pukul 21:45:59 WIB.



Output LCD Program *Orbita* di dalam ruangan Tanggal 21 Juni 2022.

Sketch Program *Orbita*

```
int M, Y, A, B;
  double delta;
  // Julian Day
  if (bln < 3)
    M = bln + 12;
  else
    M = bln;

  if (bln < 3)
    Y = thn - 1;
  else
    Y = thn;
  A = ( Y / 100);
  B = (2 + int (A / 4) - A);

  //Delta T
  double Tahun = thn + (double)(bln - 1) / 12 +
(double)(tgl) / 365;
  if (Tahun > 2005)
    if (Tahun <= 2050)
      delta = 62.92 + 0.32217 * (Tahun - 2000) + 0.005589
* (Tahun - 2000) * (Tahun - 2000);
    else
      delta = 0;
  else
    delta = 0;

  double JulianDayWaktuUT = 1720994.5 + int32_t(365.25
* Y) + int32_t(30.60001 * (M + 1)) + B + tgl + (double)((j
+ (double)(mnt) / 60 + (double)(dtk) / 3600)) / 24;
  double DeltaT = ((double)(delta) / 86400);
  double T = (JulianDayWaktuUT - 2451545) / 36525;
  double JulianDayEphemeris = JulianDayWaktuUT + DeltaT;
  double TD = (JulianDayEphemeris - 2451545) / 36525;
  double tau = TD / 10;

  // Koreksi Nutasi
  double D = radians(fmod((297.85036 + 445267.11148 * T
- 0.0019142 * T * T + T * T * T / 189474) + 360, 360));
  double X = radians(fmod((357.52772 + 35999.05034 * T
- 0.0001603 * T * T - T * T * T / 300000) + 360, 360));
```

```
  double X_ = radians(fmod((134.96298 + 477198.867398 *
T + 0.0086972 * T * T + T * T * T / 56250) + 360, 360));
  double F =  radians(fmod((93.27191 + 483202.017538 *
T - 0.0036825 * T * T + T * T * T / 327270) + 360, 360));
  double O = radians(fmod((125.04452 - 1934.136261 * T
+ 0.0020708 * T * T + T * T * T / 450000) + 360, 360));

  double nutasi1 = (-171996 + (-174.2) * T) * sin(0 * D
+ 0 * X + 0 * X_ + 0 * F + 1 * O);
  double nutasi2 = (-13187 + (-1.6) * T) * sin(-2 * D +
0 * X + 0 * X_ + 2 * F + 2 * O);
  double nutasi3 = (-2274 + (-0.2) * T) * sin(0 * D + 0
* X + 0 * X_ + 2 * F + 2 * O);
  double nutasi4 = (2062 + 0.2 * T) * sin(0 * D + 0 * X
+ 0 * X_ + 0 * F + 2 * O);
  double nutasi5 = (1426 + (-3.4) * T) * sin(0 * D + 1
* X + 0 * X_ + 0 * F + 0 * O);
  double nutasi6 = (712 + 0.1 * T) * sin(0 * D + 0 * X
+ 1 * X_ + 0 * F + 0 * O);
  double nutasi7 = (-517 + 1.2 * T) * sin(-2 * D + 1 *
X + 0 * X_ + 2 * F + 2 * O);
  double nutasi8 = (-386 + (-0.4) * T) * sin(0 * D + 0
* X + 0 * X_ + 2 * F + 1 * O);
  double nutasi9 = (217 + (-0.5) * T) * sin(-2 * D + (-
1) * X + 0 * X_ + 2 * F + 2 * O);
  double nutasi10 = (129 + 0.1 * T) * sin(-2 * D + 0 *
X + 0 * X_ + 2 * F + 1 * O);
  double nutasi11 = (63 + 0.1 * T) * sin(0 * D + 0 * X
+ 1 * X_ + 0 * F + 1 * O);
  double nutasi12 = (-58 + (-0.1) * T) * sin(0 * D + 0
* X + (-1) * X_ + 0 * F + 1 * O);
  double nutasi13 = (17 + (-0.1) * T) * sin(0 * D + 2 *
X + 0 * X_ + 0 * F + 0 * O);
  double nutasi14 = (-16 + 0.1 * T) * sin(-2 * D + 2 *
X + 0 * X_ + 2 * F + 2 * O);
  double nutasi15 = (-301 + 0 * T) * sin(0 * D + 0 * X
+ 1 * X_ + 2 * F + 2 * O);
  double nutasi16 = (-158 + 0 * T) * sin(-2 * D + 0 * X
+ 1 * X_ + 0 * F + 0 * O);
  double nutasi17 = (123 + 0 * T) * sin(0 * D + 0 * X +
(-1) * X_ + 2 * F + 2 * O);
  double nutasi18 = (63 + 0 * T) * sin(2 * D + 0 * X +
0 * X_ + 0 * F + 0 * O);
```

```
  double nutasi19 = (-59 + 0 * T) * sin(2 * D + 0 * X +
(-1) * X_ + 2 * F + 2 * O);
  double nutasi20 = (-51 + 0 * T) * sin(0 * D + 0 * X +
1 * X_ + 2 * F + 1 * O);
  double nutasi21 = (48 + 0 * T) * sin(-2 * D + 0 * X +
2 * X_ + 0 * F + 0 * O);
  double nutasi22 = (46 + 0 * T) * sin(0 * D + 0 * X +
(-2) * X_ + 2 * F + 1 * O);
  double nutasi23 = (-38 + 0 * T) * sin(2 * D + 0 * X +
0 * X_ + 2 * F + 2 * O);
  double nutasi24 = (-31 + 0 * T) * sin(0 * D + 0 * X +
2 * X_ + 2 * F + 2 * O);
  double nutasi25 = (29 + 0 * T) * sin(0 * D + 0 * X +
2 * X_ + 0 * F + 0 * O);
  double nutasi26 = (29 + 0 * T) * sin(-2 * D + 0 * X +
1 * X_ + 2 * F + 2 * O);
  double nutasi27 = (26 + 0 * T) * sin(0 * D + 0 * X +
0 * X_ + 2 * F + 0 * O);
  double nutasi28 = (-22 + 0 * T) * sin(-2 * D + 0 * X
+ 0 * X_ + 2 * F + 0 * O);
  double nutasi29 = (21 + 0 * T) * sin(0 * D + 0 * X +
(-1) * X_ + 2 * F + 1 * O);
  double nutasi30 = (16 + 0 * T) * sin(2 * D + 0 * X +
(-1) * X_ + 0 * F + 1 * O);
  double nutasi31 = (-15 + 0 * T) * sin(0 * D + 1 * X +
0 * X_ + 0 * F + 1 * O);
  double nutasi32 = (-13 + 0 * T) * sin(-2 * D + 0 * X
+ 1 * X_ + 0 * F + 1 * O);
  double nutasi33 = (-12 + 0 * T) * sin(0 * D + (-1) *
X + 0 * X_ + 0 * F + 1 * O);
  double nutasi34 = (11 + 0 * T) * sin(0 * D + 0 * X +
2 * X_ + (-2) * F + 0 * O);
  double nutasi35 = (-10 + 0 * T) * sin(2 * D + 0 * X +
(-1) * X_ + 2 * F + 1 * O);
  double nutasi36 = (-8 + 0 * T) * sin(2 * D + 0 * X +
1 * X_ + 2 * F + 2 * O);
  double nutasi37 = (7 + 0 * T) * sin(0 * D + 1 * X + 0
* X_ + 2 * F + 2 * O);
  double nutasi38 = (-7 + 0 * T) * sin(-2 * D + 1 * X +
1 * X_ + 0 * F + 0 * O);
  double nutasi39 = (-7 + 0 * T) * sin(0 * D + (-1) * X
+ 0 * X_ + 2 * F + 2 * O);
  double nutasi40 = (-7 + 0 * T) * sin(2 * D + 0 * X +
0 * X_ + 2 * F + 1 * O);
```

```
  double nutasi41 = (6 + 0 * T) * sin(2 * D + 0 * X + 1
* X_ + 0 * F + 0 * O);
  double nutasi42 = (6 + 0 * T) * sin(-2 * D + 0 * X +
2 * X_ + 2 * F + 2 * O);
  double nutasi43 = (6 + 0 * T) * sin(-2 * D + 0 * X +
1 * X_ + 2 * F + 1 * O);
  double nutasi44 = (-6 + 0 * T) * sin(2 * D + 0 * X +
(-2) * X_ + 0 * F + 1 * O);
  double nutasi45 = (-6 + 0 * T) * sin(2 * D + 0 * X +
0 * X_ + 0 * F + 1 * O);
  double nutasi46 = (5 + 0 * T) * sin(0 * D + (-1) * X
+ 1 * X_ + 0 * F + 0 * O);
  double nutasi47 = (-5 + 0 * T) * sin(-2 * D + (-1) *
X + 0 * X_ + 2 * F + 1 * O);
  double nutasi48 = (-5 + 0 * T) * sin(-2 * D + 0 * X +
0 * X_ + 0 * F + 1 * O);
  double nutasi49 = (-5 + 0 * T) * sin(0 * D + 0 * X +
2 * X_ + 2 * F + 1 * O);
  double nutasi50 = (4 + 0 * T) * sin(-2 * D + 0 * X +
2 * X_ + 0 * F + 1 * O);
  double nutasi51 = (4 + 0 * T) * sin(-2 * D + 1 * X +
0 * X_ + 2 * F + 1 * O);
  double nutasi52 = (4 + 0 * T) * sin(0 * D + 0 * X + 1
* X_ + (-2) * F + 0 * O);
  double nutasi53 = (-4 + 0 * T) * sin(-1 * D + 0 * X +
1 * X_ + 0 * F + 0 * O);
  double nutasi54 = (-4 + 0 * T) * sin(-2 * D + 1 * X +
0 * X_ + 0 * F + 0 * O);
  double nutasi55 = (-4 + 0 * T) * sin(1 * D + 0 * X +
0 * X_ + 0 * F + 0 * O);
  double nutasi56 = (3 + 0 * T) * sin(0 * D + 0 * X + 1
* X_ + 2 * F + 0 * O);
  double nutasi57 = (-3 + 0 * T) * sin(0 * D + 0 * X +
(-2) * X_ + 2 * F + 2 * O);
  double nutasi58 = (-3 + 0 * T) * sin(-1 * D + (-1) *
X + 1 * X_ + 0 * F + 0 * O);
  double nutasi59 = (-3 + 0 * T) * sin(0 * D + 1 * X +
1 * X_ + 0 * F + 0 * O);
  double nutasi60 = (-3 + 0 * T) * sin(0 * D + (-1) * X
+ 1 * X_ + 2 * F + 2 * O);
  double nutasi61 = (-3 + 0 * T) * sin(2 * D + (-1) * X
+ (-1) * X_ + 2 * F + 2 * O);
  double nutasi62 = (-3 + 0 * T) * sin(0 * D + 0 * X +
3 * X_ + 2 * F + 2 * O);
```

```
  double nutasi63 = (-3 + 0 * T) * sin(2 * D + (-1) * X
+ 0 * X_ + 2 * F + 2 * O);

  double nutasiTotal = (nutasi1 + nutasi2 + nutasi3 +
nutasi4 + nutasi5 + nutasi6 + nutasi7 + nutasi8 + nutasi9
+ nutasi10 + nutasi11 + nutasi12 + nutasi13 + nutasi14
+ nutasi15 + nutasi16 + nutasi17 + nutasi18 + nutasi19
+ nutasi20 + nutasi21 + nutasi22 + nutasi23 + nutasi24
+ nutasi25 + nutasi26 + nutasi27 + nutasi28 + nutasi29
+ nutasi30 + nutasi31 + nutasi32 + nutasi33 + nutasi34
+ nutasi35 + nutasi36 + nutasi37 + nutasi38 + nutasi39
+ nutasi40 + nutasi41 + nutasi42 + nutasi43 + nutasi44
+ nutasi45 + nutasi46 + nutasi47 + nutasi48 + nutasi49
+ nutasi50 + nutasi51 + nutasi52 + nutasi53 + nutasi54
+ nutasi55 + nutasi56 + nutasi57 + nutasi58 + nutasi59
+ nutasi60 + nutasi61 + nutasi62 + nutasi63) / 10000;
  double nutasi = nutasiTotal / 3600;

  // Koreksi Epsilon
  double epsilon1 = (92025 + (8.9) * T) * cos((0) * D +
(0) * X + (0) * X_ + (0) * F + (1) * O);
  double epsilon2 = (5736 + (-3.1) * T) * cos((-2) * D
+ (0) * X + (0) * X_ + (2) * F + (2) * O);
  double epsilon3 = (977 + (-0.5) * T) * cos((0) * D +
(0) * X + (0) * X_ + (2) * F + (2) * O);
  double epsilon4 = (-895 + (0.5) * T) * cos((0) * D +
(0) * X + (0) * X_ + (0) * F + (2) * O);
  double epsilon5 = (54 + (-0.1) * T) * cos((0) * D +
(1) * X + (0) * X_ + (0) * F + (0) * O);
  double epsilon6 = (224 + (-0.6) * T) * cos((-2) * D +
(1) * X + (0) * X_ + (2) * F + (2) * O);
  double epsilon7 = (129 + (-0.1) * T) * cos((0) * D +
(0) * X + (1) * X_ + (2) * F + (2) * O);
  double epsilon8 = (-95 + (0.3) * T) * cos((-2) * D +
(-1) * X + (0) * X_ + (2) * F + (2) * O);
  double epsilon9 = (-7 + (0) * T) * cos((0) * D + (0)
* X + (1) * X_ + (0) * F + (0) * O);
  double epsilon10 = (200 + (0) * T) * cos((0) * D + (0)
* X + (0) * X_ + (2) * F + (1) * O);
  double epsilon11 = (-70 + (0) * T) * cos((-2) * D +
(0) * X + (0) * X_ + (2) * F + (1) * O);
  double epsilon12 = (-53 + (0) * T) * cos((0) * D + (0)
* X + (-1) * X_ + (2) * F + (2) * O);
```

```
  double epsilon13 = (-33 + (0) * T) * cos((0) * D + (0)
* X + (1) * X_ + (0) * F + (1) * O);
  double epsilon14 = (26 + (0) * T) * cos((2) * D + (0)
* X + (-1) * X_ + (2) * F + (2) * O);
  double epsilon15 = (32 + (0) * T) * cos((0) * D + (0)
* X + (-1) * X_ + (0) * F + (1) * O);
  double epsilon16 = (27 + (0) * T) * cos((0) * D + (0)
* X + (1) * X_ + (2) * F + (1) * O);
  double epsilon17 = (-24 + (0) * T) * cos((0) * D + (0)
* X + (-2) * X_ + (2) * F + (1) * O);
  double epsilon18 = (16 + (0) * T) * cos((2) * D + (0)
* X + (0) * X_ + (2) * F + (2) * O);
  double epsilon19 = (13 + (0) * T) * cos((0) * D + (0)
* X + (2) * X_ + (2) * F + (2) * O);
  double epsilon20 = (-12 + (0) * T) * cos((-2) * D +
(0) * X + (1) * X_ + (2) * F + (2) * O);
  double epsilon21 = (-10 + (0) * T) * cos((0) * D + (0)
* X + (-1) * X_ + (2) * F + (1) * O);
  double epsilon22 = (-8 + (0) * T) * cos((2) * D + (0)
* X + (-1) * X_ + (0) * F + (1) * O);
  double epsilon23 = (7 + (0) * T) * cos((-2) * D + (2)
* X + (0) * X_ + (2) * F + (2) * O);
  double epsilon24 = (9 + (0) * T) * cos((0) * D + (1)
* X + (0) * X_ + (0) * F + (1) * O);
  double epsilon25 = (7 + (0) * T) * cos((-2) * D + (0)
* X + (1) * X_ + (0) * F + (1) * O);
  double epsilon26 = (6 + (0) * T) * cos((0) * D + (-1)
* X + (0) * X_ + (0) * F + (1) * O);
  double epsilon27 = (5 + (0) * T) * cos((2) * D + (0)
* X + (-1) * X_ + (2) * F + (1) * O);
  double epsilon28 = (3 + (0) * T) * cos((2) * D + (0)
* X + (1) * X_ + (2) * F + (2) * O);
  double epsilon29 = (-3 + (0) * T) * cos((0) * D + (1)
* X + (0) * X_ + (2) * F + (2) * O);
  double epsilon30 = (3 + (0) * T) * cos((0) * D + (-1)
* X + (0) * X_ + (2) * F + (2) * O);
  double epsilon31 = (3 + (0) * T) * cos((2) * D + (0)
* X + (0) * X_ + (2) * F + (1) * O);
  double epsilon32 = (-3 + (0) * T) * cos((-2) * D + (0)
* X + (2) * X_ + (2) * F + (2) * O);
  double epsilon33 = (-3 + (0) * T) * cos((-2) * D + (0)
* X + (1) * X_ + (2) * F + (1) * O);
  double epsilon34 = (3 + (0) * T) * cos((2) * D + (0)
* X + (-2) * X_ + (0) * F + (1) * O);
```

```
  double epsilon35 = (3 + (0) * T) * cos((2) * D + (0)
* X + (0) * X_ + (0) * F + (1) * O);
  double epsilon36 = (3 + (0) * T) * cos((-2) * D + (-
1) * X + (0) * X_ + (2) * F + (1) * O);
  double epsilon37 = (3 + (0) * T) * cos((-2) * D + (0)
* X + (0) * X_ + (0) * F + (1) * O);
  double epsilon38 = (3 + (0) * T) * cos((0) * D + (0)
* X + (2) * X_ + (2) * F + (1) * O);

  double sumE = (epsilon1 + epsilon2 + epsilon3 +
epsilon4 + epsilon5 + epsilon6 + epsilon7 + epsilon8 +
epsilon9 + epsilon10 + epsilon11 + epsilon12 + epsilon13
+ epsilon14 + epsilon15 + epsilon16 + epsilon17 +
epsilon18 + epsilon19 + epsilon20 + epsilon21 +
epsilon22 + epsilon23 + epsilon24 + epsilon25 +
epsilon26 + epsilon27 + epsilon28 + epsilon29 +
epsilon30 + epsilon31 + epsilon32 + epsilon33 +
epsilon34 + epsilon35 + epsilon36 + epsilon37 +
epsilon38) / 10000;
  double sumED = sumE / 3600;
  double Uu = T / 100;
  double epsilonZero = 23 + (double)(26) / 60 + 21.448
/ 3600 + (-4680.93 * Uu - 1.55 * Uu * Uu + 1999.25 * Uu
* Uu * Uu - 51.38 * Uu * Uu * Uu * Uu - 249.67 * Uu * Uu
* Uu * Uu * Uu - 39.05 * Uu * Uu * Uu * Uu * Uu * Uu +
7.12 * Uu * Uu * Uu * Uu * Uu * Uu * Uu + 27.87 * Uu *
Uu * Uu * Uu * Uu * Uu * Uu * Uu + 5.79 * Uu * Uu * Uu
* Uu * Uu * Uu * Uu * Uu * Uu + 2.45 * Uu * Uu * Uu * Uu
* Uu * Uu * Uu * Uu * Uu * Uu) / 3600;
  double epsilon = epsilonZero + sumED;

  //Waktu
  double GST_UT = fmod((280.46061837 + 360.98564736629
* (JulianDayWaktuUT - 2451545) + 0.000387933 * T * T -
T * T * T / 38710000) + 360, 360) / 15;
  double GST_L = GST_UT + nutasi * cos(radians(epsilon))
/ 15;
  double LST = fmod((GST_L + (double)(b) / 15) + 24,
24);

  //Posisi Matahari
  double BujurEkliptikaA1 = 175347046 * cos(0 + 0 * tau);
  double BujurEkliptikaA2 = 3341656 * cos(4.6692568 +
6283.07585 * tau);
```

```
  double BujurEkliptikaA3 = 34894 * cos(4.6261 +
12566.1517 * tau);
  double BujurEkliptikaA4 = 3497 * cos(2.7441 +
5753.3849 * tau);
  double BujurEkliptikaA5 = 3418 * cos(2.8289 + 3.5231
* tau);
  double BujurEkliptikaA6 = 3136 * cos(3.6277 +
77713.7715 * tau);
  double BujurEkliptikaA7 = 2676 * cos(4.4181 +
7860.4194 * tau);
  double BujurEkliptikaA8 = 2343 * cos(6.1352 +
3930.2097 * tau);
  double BujurEkliptikaA9 = 1324 * cos(0.7425 +
11506.7698 * tau);
  double BujurEkliptikaA10 = 1273 * cos(2.0371 + 529.691
* tau);
  double BujurEkliptikaA11 = 1199 * cos(1.1096 +
1577.3435 * tau);
  double BujurEkliptikaA12 = 990 * cos(5.233 + 5884.927
* tau);
  double BujurEkliptikaA13 = 902 * cos(2.045 + 26.298 *
tau);
  double BujurEkliptikaA14 = 857 * cos(3.508 + 398.149
* tau);
  double BujurEkliptikaA15 = 780 * cos(1.179 + 5223.694
* tau);
  double BujurEkliptikaA16 = 753 * cos(2.533 + 5507.553
* tau);
  double BujurEkliptikaA17 = 505 * cos(4.583 + 18849.228
* tau);
  double BujurEkliptikaA18 = 492 * cos(4.205 + 775.523
* tau);
  double BujurEkliptikaA19 = 357 * cos(2.92 + 0.067 *
tau);
  double BujurEkliptikaA20 = 317 * cos(5.849 + 11790.629
* tau);
  double BujurEkliptikaA21 = 284 * cos(1.899 + 796.298
* tau);
  double BujurEkliptikaA22 = 271 * cos(0.315 + 10977.079
* tau);
  double BujurEkliptikaA23 = 243 * cos(0.345 + 5486.778
* tau);
  double BujurEkliptikaA24 = 206 * cos(4.806 + 2544.314
* tau);
```

```
  double BujurEkliptikaA25 = 205 * cos(1.869 + 5573.143
* tau);
  double BujurEkliptikaA26 = 202 * cos(2.458 + 6069.777
* tau);
  double BujurEkliptikaA27 = 156 * cos(0.833 + 213.299
* tau);
  double BujurEkliptikaA28 = 132 * cos(3.411 + 2942.463
* tau);
  double BujurEkliptikaA29 = 126 * cos(1.083 + 20.775 *
tau);
  double BujurEkliptikaA30 = 115 * cos(0.645 + 0.98 *
tau);
  double BujurEkliptikaA31 = 103 * cos(0.636 + 4694.003
* tau);
  double BujurEkliptikaA32 = 102 * cos(0.976 + 15720.839
* tau);
  double BujurEkliptikaA33 = 102 * cos(4.267 + 7.114 *
tau);
  double BujurEkliptikaA34 = 99 * cos(6.21 + 2146.17 *
tau);
  double BujurEkliptikaA35 = 98 * cos(0.68 + 155.42 *
tau);
  double BujurEkliptikaA36 = 86 * cos(5.98 + 161000.69
* tau);
  double BujurEkliptikaA37 = 85 * cos(1.3 + 6275.96 *
tau);
  double BujurEkliptikaA38 = 85 * cos(3.67 + 71430.7 *
tau);
  double BujurEkliptikaA39 = 80 * cos(1.81 + 17260.15 *
tau);
  double BujurEkliptikaA40 = 79 * cos(3.04 + 12036.46 *
tau);
  double BujurEkliptikaA41 = 75 * cos(1.76 + 5088.63 *
tau);
  double BujurEkliptikaA42 = 74 * cos(3.5 + 3154.69 *
tau);
  double BujurEkliptikaA43 = 74 * cos(4.68 + 801.82 *
tau);
  double BujurEkliptikaA44 = 70 * cos(0.83 + 9437.76 *
tau);
  double BujurEkliptikaA45 = 62 * cos(3.98 + 8827.39 *
tau);
  double BujurEkliptikaA46 = 61 * cos(1.82 + 7084.9 *
tau);
```

```
  double BujurEkliptikaA47 = 57 * cos(2.78 + 6286.6 *
tau);
  double BujurEkliptikaA48 = 56 * cos(4.39 + 14143.5 *
tau);
  double BujurEkliptikaA49 = 56 * cos(3.47 + 6279.55 *
tau);
  double BujurEkliptikaA50 = 52 * cos(0.19 + 12139.55 *
tau);
  double BujurEkliptikaA51 = 52 * cos(1.33 + 1748.02 *
tau);
  double BujurEkliptikaA52 = 51 * cos(0.28 + 5856.48 *
tau);
  double BujurEkliptikaA53 = 49 * cos(0.49 + 1194.45 *
tau);
  double BujurEkliptikaA54 = 41 * cos(5.37 + 8429.24 *
tau);
  double BujurEkliptikaA55 = 41 * cos(2.4 + 19651.05 *
tau);
  double BujurEkliptikaA56 = 39 * cos(6.17 + 10447.39 *
tau);
  double BujurEkliptikaA57 = 37 * cos(6.04 + 10213.29 *
tau);
  double BujurEkliptikaA58 = 37 * cos(2.57 + 1059.38 *
tau);
  double BujurEkliptikaA59 = 36 * cos(1.71 + 2352.87 *
tau);
  double BujurEkliptikaA60 = 36 * cos(1.78 + 6812.77 *
tau);
  double BujurEkliptikaA61 = 33 * cos(0.59 + 17789.85 *
tau);
  double BujurEkliptikaA62 = 30 * cos(0.44 + 83996.85 *
tau);
  double BujurEkliptikaA63 = 30 * cos(2.74 + 1349.87 *
tau);
  double BujurEkliptikaA64 = 25 * cos(3.16 + 4690.48 *
tau);

  double   BujurEkliptikaA   =   BujurEkliptikaA1   +
BujurEkliptikaA2 + BujurEkliptikaA3 + BujurEkliptikaA4
+ BujurEkliptikaA5 + BujurEkliptikaA6 + BujurEkliptikaA7
+     BujurEkliptikaA8     +     BujurEkliptikaA9     +
BujurEkliptikaA10      +      BujurEkliptikaA11      +
BujurEkliptikaA12      +      BujurEkliptikaA13      +
BujurEkliptikaA14      +      BujurEkliptikaA15      +
```

```
BujurEkliptikaA16      +      BujurEkliptikaA17      +
BujurEkliptikaA18      +      BujurEkliptikaA19      +
BujurEkliptikaA20      +      BujurEkliptikaA21      +
BujurEkliptikaA22      +      BujurEkliptikaA23      +
BujurEkliptikaA24      +      BujurEkliptikaA25      +
BujurEkliptikaA26      +      BujurEkliptikaA27      +
BujurEkliptikaA28      +      BujurEkliptikaA29      +
BujurEkliptikaA30      +      BujurEkliptikaA31      +
BujurEkliptikaA32      +      BujurEkliptikaA33      +
BujurEkliptikaA34      +      BujurEkliptikaA35      +
BujurEkliptikaA36      +      BujurEkliptikaA37      +
BujurEkliptikaA38      +      BujurEkliptikaA39      +
BujurEkliptikaA40      +      BujurEkliptikaA41      +
BujurEkliptikaA42      +      BujurEkliptikaA43      +
BujurEkliptikaA44      +      BujurEkliptikaA45      +
BujurEkliptikaA46      +      BujurEkliptikaA47      +
BujurEkliptikaA48      +      BujurEkliptikaA49      +
BujurEkliptikaA50      +      BujurEkliptikaA51      +
BujurEkliptikaA52      +      BujurEkliptikaA53      +
BujurEkliptikaA54      +      BujurEkliptikaA55      +
BujurEkliptikaA56      +      BujurEkliptikaA57      +
BujurEkliptikaA58      +      BujurEkliptikaA59      +
BujurEkliptikaA60      +      BujurEkliptikaA61      +
BujurEkliptikaA62      +      BujurEkliptikaA63      +
BujurEkliptikaA64;

  double BujurEkliptikaB1 = 628331966747 * cos(0 + 0 *
tau) / 10000;
  double  BujurEkliptikaB2  =  206059  *  cos(2.678235  +
6283.07585 * tau);
  double  BujurEkliptikaB3  =  4303  *  cos(2.6351  +
12566.1517 * tau);
  double BujurEkliptikaB4 = 425 * cos(1.59 + 3.523 *
tau);
  double BujurEkliptikaB5 = 119 * cos(5.796 + 26.298 *
tau);
  double BujurEkliptikaB6 = 109 * cos(2.966 + 1577.344
* tau);
  double BujurEkliptikaB7 = 93 * cos(2.59 + 18849.23 *
tau);
  double BujurEkliptikaB8 = 72 * cos(1.14 + 529.69 *
tau);
  double BujurEkliptikaB9 = 68 * cos(1.87 + 398.15 *
tau);
```

```
  double BujurEkliptikaB10 = 67 * cos(4.41 + 5507.55 *
tau);
  double BujurEkliptikaB11 = 59 * cos(2.89 + 5223.69 *
tau);
  double BujurEkliptikaB12 = 56 * cos(2.17 + 155.42 *
tau);
  double BujurEkliptikaB13 = 45 * cos(0.4 + 796.3 * tau);
  double BujurEkliptikaB14 = 36 * cos(0.47 + 775.52 *
tau);
  double BujurEkliptikaB15 = 29 * cos(2.65 + 7.11 * tau);
  double BujurEkliptikaB16 = 21 * cos(5.34 + 0.98 * tau);
  double BujurEkliptikaB17 = 19 * cos(1.85 + 5486.78 *
tau);
  double BujurEkliptikaB18 = 19 * cos(4.97 + 213.3 *
tau);
  double BujurEkliptikaB19 = 17 * cos(2.99 + 6275.96 *
tau);
  double BujurEkliptikaB20 = 16 * cos(0.03 + 2544.31 *
tau);
  double BujurEkliptikaB21 = 16 * cos(1.43 + 2146.17 *
tau);
  double BujurEkliptikaB22 = 15 * cos(1.21 + 10977.08 *
tau);
  double BujurEkliptikaB23 = 12 * cos(2.83 + 1748.02 *
tau);
  double BujurEkliptikaB24 = 12 * cos(3.26 + 5088.63 *
tau);
  double BujurEkliptikaB25 = 12 * cos(5.27 + 1194.45 *
tau);
  double BujurEkliptikaB26 = 12 * cos(2.08 + 4694 * tau);
  double BujurEkliptikaB27 = 11 * cos(0.77 + 553.57 *
tau);
  double BujurEkliptikaB28 = 10 * cos(1.3 + 6286.6 *
tau);
  double BujurEkliptikaB29 = 10 * cos(4.24 + 1349.87 *
tau);
  double BujurEkliptikaB30 = 9 * cos(2.7 + 242.73 * tau);
  double BujurEkliptikaB31 = 9 * cos(5.64 + 951.72 *
tau);
  double BujurEkliptikaB32 = 8 * cos(5.3 + 2352.87 *
tau);
  double BujurEkliptikaB33 = 6 * cos(2.65 + 9437.76 *
tau);
```

```
  double BujurEkliptikaB34 = 6 * cos(4.67 + 4690.48 *
tau);

  double BujurEkliptikaC1 = 52919 * cos(0 + 0 * tau);
  double  BujurEkliptikaC2  =  8720  *  cos(1.0721  +
6283.0758 * tau);
  double BujurEkliptikaC3 = 309 * cos(0.867 + 12566.152
* tau);
  double BujurEkliptikaC4 = 27 * cos(0.05 + 3.52 * tau);
  double BujurEkliptikaC5 = 16 * cos(5.19 + 26.3 * tau);
  double BujurEkliptikaC6 = 16 * cos(3.68 + 155.42 *
tau);
  double BujurEkliptikaC7 = 10 * cos(0.76 + 18849.23 *
tau);
  double BujurEkliptikaC8 = 9 * cos(2.06 + 77713.77 *
tau);
  double BujurEkliptikaC9 = 7 * cos(0.83 + 775.52 * tau);
  double BujurEkliptikaC10 = 5 * cos(4.66 + 1577.34 *
tau);
  double BujurEkliptikaC11 = 4 * cos(1.03 + 7.11 * tau);
  double BujurEkliptikaC12 = 4 * cos(3.44 + 5573.14 *
tau);
  double BujurEkliptikaC13 = 3 * cos(5.14 + 796.3 * tau);
  double BujurEkliptikaC14 = 3 * cos(6.05 + 5507.55 *
tau);
  double BujurEkliptikaC15 = 3 * cos(1.19 + 242.73 *
tau);
  double BujurEkliptikaC16 = 3 * cos(6.12 + 529.69 *
tau);
  double BujurEkliptikaC17 = 3 * cos(0.31 + 398.15 *
tau);
  double BujurEkliptikaC18 = 3 * cos(2.28 + 553.57 *
tau);
  double BujurEkliptikaC19 = 2 * cos(4.38 + 5223.69 *
tau);
  double BujurEkliptikaC20 = 2 * cos(3.75 + 0.98 * tau);

  double   BujurEkliptikaC   =   BujurEkliptikaC1   +
BujurEkliptikaC2 + BujurEkliptikaC3 + BujurEkliptikaC4
+ BujurEkliptikaC5 + BujurEkliptikaC6 + BujurEkliptikaC7
+     BujurEkliptikaC8     +     BujurEkliptikaC9     +
BujurEkliptikaC10      +      BujurEkliptikaC11      +
BujurEkliptikaC12      +      BujurEkliptikaC13      +
BujurEkliptikaC14      +      BujurEkliptikaC15      +
```

```
BujurEkliptikaC16      +      BujurEkliptikaC17      +
BujurEkliptikaC18      +      BujurEkliptikaC19      +
BujurEkliptikaC20;

  double BujurEkliptikaD1 = 289 * cos(5.844 + 6283.076
* tau);
  double BujurEkliptikaD2 = 35 * cos(0 + 0 * tau);
  double BujurEkliptikaD3 = 17 * cos(5.49 + 12566.15 *
tau);
  double BujurEkliptikaD4 = 3 * cos(5.2 + 155.42 * tau);
  double BujurEkliptikaD5 = 1 * cos(4.72 + 3.52 * tau);
  double BujurEkliptikaD6 = 1 * cos(5.3 + 18849.23 *
tau);
  double BujurEkliptikaD7 = 1 * cos(5.97 + 242.73 * tau);

  double   BujurEkliptikaD   =   BujurEkliptikaD1   +
BujurEkliptikaD2 + BujurEkliptikaD3 + BujurEkliptikaD4
+     BujurEkliptikaD5     +     BujurEkliptikaD6     +
BujurEkliptikaD7;

  double BujurEkliptikaE1 = 114 * cos(3.142 + 0 * tau);
  double BujurEkliptikaE2 = 8 * cos(4.13 + 6283.08 *
tau);
  double BujurEkliptikaE3 = 1 * cos(3.84 + 12566.15 *
tau);

  double   BujurEkliptikaE   =   BujurEkliptikaE1   +
BujurEkliptikaE2 + BujurEkliptikaE3;

  double BujurEkliptikaF1 = 1 * cos(3.14 + 0 * tau);

  double BujurEkliptikaF = BujurEkliptikaF1;

  double totalBujurEkliptikaA = BujurEkliptikaA;

  double totalBujurEkliptikaC = BujurEkliptikaC * tau *
tau;
  double totalBujurEkliptikaD = BujurEkliptikaD * tau *
tau * tau;
  double totalBujurEkliptikaE = BujurEkliptikaE * tau *
tau * tau * tau;
  double totalBujurEkliptikaF = BujurEkliptikaF * tau *
tau * tau * tau * tau;
```

```
  double              totalBujurEkliptika              =
fmod((degrees((totalBujurEkliptikaA                    +
(((BujurEkliptikaB1   *   10000)   +   BujurEkliptikaB2   +
BujurEkliptikaB3 + BujurEkliptikaB4 + BujurEkliptikaB5
+ BujurEkliptikaB6 + BujurEkliptikaB7 + BujurEkliptikaB8
+     BujurEkliptikaB9     +     BujurEkliptikaB10     +
BujurEkliptikaB11       +       BujurEkliptikaB12       +
BujurEkliptikaB13       +       BujurEkliptikaB14       +
BujurEkliptikaB15       +       BujurEkliptikaB16       +
BujurEkliptikaB17       +       BujurEkliptikaB18       +
BujurEkliptikaB19       +       BujurEkliptikaB20       +
BujurEkliptikaB21       +       BujurEkliptikaB22       +
BujurEkliptikaB23       +       BujurEkliptikaB24       +
BujurEkliptikaB25       +       BujurEkliptikaB26       +
BujurEkliptikaB27       +       BujurEkliptikaB28       +
BujurEkliptikaB29       +       BujurEkliptikaB30       +
BujurEkliptikaB31       +       BujurEkliptikaB32       +
BujurEkliptikaB33   +   BujurEkliptikaB34)   *   tau)   +
totalBujurEkliptikaC     +     totalBujurEkliptikaD     +
totalBujurEkliptikaE     +     totalBujurEkliptikaF)    /
100000000)) + 360, 360);
  double theta = fmod((totalBujurEkliptika + 180) + 360,
360);
  double lamda =  fmod((theta - 1.397 * T - 0.00031 * T
* T) + 360, 360);
  double delta_theta = -0.0000250916666666667;
  double thetaTerkoreksi = fmod((theta + delta_theta) +
360, 360);

  //Jarak Matahari
  double jarakA1 = 100013989 * cos(0 + 0 * tau);
  double jarakA2 = 1670700 * cos(3.0984635 + 6283.07585
* tau);
  double  jarakA3  =  13956  *  cos(3.05525  +  12566.1517  *
tau);
  double jarakA4 = 3084 * cos(5.1985 + 77713.7715 * tau);
  double jarakA5 = 1628 * cos(1.1739 + 5753.3849 * tau);
  double jarakA6 = 1576 * cos(2.8469 + 7860.4194 * tau);
  double jarakA7 = 925 * cos(5.453 + 11506.77 * tau);
  double jarakA8 = 542 * cos(4.564 + 3930.21 * tau);
  double jarakA9 = 472 * cos(3.661 + 5884.927 * tau);
  double jarakA10 = 346 * cos(0.964 + 5507.553 * tau);
  double jarakA11 = 329 * cos(5.9 + 5223.694 * tau);
  double jarakA12 = 307 * cos(0.299 + 5573.143 * tau);
```

```
  double jarakA13 = 243 * cos(4.273 + 11790.629 * tau);
  double jarakA14 = 212 * cos(5.847 + 1577.344 * tau);
  double jarakA15 = 186 * cos(5.022 + 10977.079 * tau);
  double jarakA16 = 175 * cos(3.012 + 18849.228 * tau);
  double jarakA17 = 110 * cos(5.055 + 5486.778 * tau);
  double jarakA18 = 98 * cos(0.89 + 6069.78 * tau);
  double jarakA19 = 86 * cos(5.69 + 15720.84 * tau);
  double jarakA20 = 86 * cos(1.27 + 161000.69 * tau);
  double jarakA21 = 65 * cos(0.27 + 17260.15 * tau);
  double jarakA22 = 63 * cos(0.92 + 529.69 * tau);
  double jarakA23 = 57 * cos(2.01 + 83996.85 * tau);
  double jarakA24 = 56 * cos(5.24 + 71430.7 * tau);
  double jarakA25 = 49 * cos(3.25 + 2544.31 * tau);
  double jarakA26 = 47 * cos(2.58 + 775.52 * tau);
  double jarakA27 = 45 * cos(5.54 + 9437.76 * tau);
  double jarakA28 = 43 * cos(6.01 + 6275.96 * tau);
  double jarakA29 = 39 * cos(5.36 + 4694 * tau);
  double jarakA30 = 38 * cos(2.39 + 8827.39 * tau);
  double jarakA31 = 37 * cos(0.83 + 19651.05 * tau);
  double jarakA32 = 37 * cos(4.9 + 12139.55 * tau);
  double jarakA33 = 36 * cos(1.67 + 12036.46 * tau);
  double jarakA34 = 35 * cos(1.84 + 2942.46 * tau);
  double jarakA35 = 33 * cos(0.24 + 7084.9 * tau);
  double jarakA36 = 32 * cos(0.18 + 5088.63 * tau);
  double jarakA37 = 32 * cos(1.78 + 398.15 * tau);
  double jarakA38 = 28 * cos(1.21 + 6286.6 * tau);
  double jarakA39 = 28 * cos(1.9 + 6279.55 * tau);
  double jarakA40 = 26 * cos(4.59 + 10447.39 * tau);

  double jarakA = jarakA1 + jarakA2 + jarakA3 + jarakA4
+ jarakA5 + jarakA6 + jarakA7 + jarakA8 + jarakA9 +
jarakA10 + jarakA11 + jarakA12 + jarakA13 + jarakA14 +
jarakA15 + jarakA16 + jarakA17 + jarakA18 + jarakA19 +
jarakA20 + jarakA21 + jarakA22 + jarakA23 + jarakA24 +
jarakA25 + jarakA26 + jarakA27 + jarakA28 + jarakA29 +
jarakA30 + jarakA31 + jarakA32 + jarakA33 + jarakA34 +
jarakA35 + jarakA36 + jarakA37 + jarakA38 + jarakA39 +
jarakA40;

  double jarakB1 = 103019 * cos(1.10749 + 6283.07585 *
tau);
  double jarakB2 = 1721 * cos(1.0644 + 12566.1517 * tau);
  double jarakB3 = 702 * cos(3.142 + 0 * tau);
  double jarakB4 = 32 * cos(1.02 + 18849.23 * tau);
```

```
  double jarakB5 = 31 * cos(2.84 + 5507.55 * tau);
  double jarakB6 = 25 * cos(1.32 + 5223.69 * tau);
  double jarakB7 = 18 * cos(1.42 + 1577.34 * tau);
  double jarakB8 = 10 * cos(5.91 + 10977.08 * tau);
  double jarakB9 = 9 * cos(1.42 + 6275.96 * tau);
  double jarakB10 = 9 * cos(0.27 + 5486.78 * tau);

  double jarakB = jarakB1 + jarakB2 + jarakB3 + jarakB4
+ jarakB5 + jarakB6 + jarakB7 + jarakB8 + jarakB9 +
jarakB10;
  double jarakC1 = 4359 * cos(5.7846 + 6283.0758 * tau);
  double jarakC2 = 124 * cos(5.579 + 12566.152 * tau);
  double jarakC3 = 12 * cos(3.14 + 0 * tau);
  double jarakC4 = 9 * cos(3.63 + 77713.77 * tau);
  double jarakC5 = 6 * cos(1.87 + 5573.14 * tau);
  double jarakC6 = 3 * cos(5.47 + 18849.23 * tau);
  double jarakC = jarakC1 + jarakC2 + jarakC3 + jarakC4
+ jarakC5 + jarakC6;

  double jarakD1 = 145 * cos(4.273 + 6283.076 * tau);
  double jarakD2  = 7 * cos(3.92 + 12566.15 * tau);
  double jarakD = jarakD1 + jarakD2;
  double jarakE1 = 4 * cos(2.56 + 6283.08 * tau);
  double jarakE = jarakE1;
  double jarakTotalA = jarakA;
  double jarakTotalB = jarakB * tau;
  double jarakTotalC = jarakC * tau * tau;
  double jarakTotalD = jarakD * tau * tau * tau;
  double jarakTotalE = jarakE * tau * tau * tau * tau;

  double  jarakTotal  =  (jarakTotalA  +  jarakTotalB  +
jarakTotalC + jarakTotalD + jarakTotalE) / 100000000;
  double aberasi = -20.4898 / (3600 * jarakTotal);

  double bujurMatahariNampak = fmod((thetaTerkoreksi +
aberasi + (nutasiTotal / 3600)) + 360, 360);

  //Lintang Matahari

  double lintangA1 = 280 * cos(3.199 + 84334.662 * tau);
  double lintangA2 = 102 * cos(5.422 + 5507.553 * tau);
  double lintangA3 = 80 * cos(3.88 + 5223.69 * tau);
  double lintangA4 = 44 * cos(3.7 + 2352.87 * tau);
  double lintangA5 = 32 * cos(4 + 1577.34 * tau);
```

```
  double lintangA = lintangA1 + lintangA2 + lintangA3 +
lintangA4 + lintangA5;

  double lintangBb1 = 9 * cos(3.9 + 5507.55 * tau);
  double lintangBb2 = 6 * cos(1.73 + 5223.69 * tau);

  double lintangBm = (lintangBb1 + lintangBb2) * tau;
  double jarakMatahari = 149598000 * jarakTotal;
  double   lintangTotal   =   -((degrees((lintangA   +
lintangBm) / 100000000)) * 3600);
  double deltaBeta = 0.03916 * (cos(radians(lamda)) -
sin(radians(lamda)));
  double  lintangMatahariTerkoreksi  =  lintangTotal  +
deltaBeta;
  double    anomaliRataM    =    fmod((357.5291092    +
35999.0502909 * T - 0.0001536 * T * T + T * T * T /
24490000) + 360, 360);
  double                    alphaM                    =
fmod((degrees(atan2(sin(radians(bujurMatahariNampak)) *
cos(radians(epsilon))                                -
tan(radians(lintangMatahariTerkoreksi   /   3600))   *
sin(radians(epsilon)),
cos(radians(bujurMatahariNampak)))))) + 360, 360);
  double                    deltaM                    =
degrees(asin(sin(radians(lintangMatahariTerkoreksi    /
3600))        *        cos(radians(epsilon))        +
cos(radians(lintangMatahariTerkoreksi   /   3600))   *
sin(radians(epsilon))                                *
sin(radians(bujurMatahariNampak))));

  double hourAngle = fmod((LST - (alphaM / 15)) + 24,
24) * 15;
  double                azimuthMselatan                =
degrees(atan2(sin(radians(hourAngle)),
cos(radians(hourAngle))     *     sin(radians(l))     -
tan(radians(deltaM)) * cos(radians(l))));
  double azimuthM = fmod((azimuthMselatan + 180) + 360,
360);
  double  altitudeM  =  degrees(asin(sin(radians(l))  *
sin(radians(deltaM))      +      cos(radians(l))      *
cos(radians(deltaM)) * cos(radians(hourAngle))));

  //Eqiation Of Time
```

```
  double sunMeanLongitude = fmod((280.4664567 +
360007.6982779 * tau + 0.03032028 * tau * tau + tau *
tau * tau / 49931 - tau * tau * tau * tau / 15299 - tau
* tau * tau * tau * tau / 1988000) + 360, 360);
  double eot = sunMeanLongitude - 0.0057183 - alphaM +
nutasi * cos(radians(epsilon));

  //Posisi Bulan
  double bujurRataB = radians(fmod(218.3164591 +
481267.88134236 * T - 0.0013268 * T * T + T * T * T /
538841 - T * T * T * T / 65194000, 360));
  double elongasiRataB = radians(fmod(297.8502042 +
445267.1115168 * T - 0.00163 * T * T + T * T * T / 545868
- T * T * T * T / 113065000, 360));
  double anomaliRataB = radians(fmod(134.9634114 +
477198.8676313 * T + 0.008997 * T * T + T * T * T / 69699
- T * T * T * T / 14712000, 360));
  double argumenBujurB = radians(fmod(93.2720993 +
483202.0175273 * T - 0.0034029 * T * T - T * T * T /
3526000 + T * T * T * T / 863310000, 360));
  double eksentrisitasOrbit = 1 - 0.002516 * T -
0.0000074 * T * T;
  double argumenA1 = radians(fmod(119.75 + 131.849 * T,
360));
  double argumenA2 = radians(fmod(53.09 + 479264.29 * T,
360));
  double argumenA3 = radians(fmod(313.45 + 481266.484 *
T, 360));
  //Koreksi Bujur Bulan
  double bujur1 = 6288774 * (pow(eksentrisitasOrbit,
(abs(0))) * sin(0 * elongasiRataB + 0 *
(radians(anomaliRataM)) + 1 * anomaliRataB + 0 *
argumenBujurB);
  double bujur2 = 1274027 * (pow(eksentrisitasOrbit,
(abs(0))) * sin(2 * elongasiRataB + 0 *
(radians(anomaliRataM)) + -1 * anomaliRataB + 0 *
argumenBujurB);
  double bujur3 = 658314 * (pow(eksentrisitasOrbit,
(abs(0))) * sin(2 * elongasiRataB + 0 *
(radians(anomaliRataM)) + 0 * anomaliRataB + 0 *
argumenBujurB);
  double bujur4 = 213618 * (pow(eksentrisitasOrbit,
(abs(0))) * sin(0 * elongasiRataB + 0 *
```

```
(radians(anomaliRataM)) + 2 * anomaliRataB + 0 *
argumenBujurB);
  double bujur5 = -185116 * (pow(eksentrisitasOrbit,
(abs(1)))) * sin(0 * elongasiRataB + 1 *
(radians(anomaliRataM)) + 0 * anomaliRataB + 0 *
argumenBujurB);
  double bujur6 = -114332 * (pow(eksentrisitasOrbit,
(abs(0)))) * sin(0 * elongasiRataB + 0 *
(radians(anomaliRataM)) + 0 * anomaliRataB + 2 *
argumenBujurB);
  double bujur7 = 58793 * (pow(eksentrisitasOrbit,
(abs(0)))) * sin(2 * elongasiRataB + 0 *
(radians(anomaliRataM)) + -2 * anomaliRataB + 0 *
argumenBujurB);
  double bujur8 = 57066 * (pow(eksentrisitasOrbit,
(abs(-1)))) * sin(2 * elongasiRataB + -1 *
(radians(anomaliRataM)) + -1 * anomaliRataB + 0 *
argumenBujurB);
  double bujur9 = 53322 * (pow(eksentrisitasOrbit,
(abs(0)))) * sin(2 * elongasiRataB + 0 *
(radians(anomaliRataM)) + 1 * anomaliRataB + 0 *
argumenBujurB);
  double bujur10 = 45758 * (pow(eksentrisitasOrbit,
(abs(-1)))) * sin(2 * elongasiRataB + -1 *
(radians(anomaliRataM)) + 0 * anomaliRataB + 0 *
argumenBujurB);
  double bujur11 = -40923 * (pow(eksentrisitasOrbit,
(abs(1)))) * sin(0 * elongasiRataB + 1 *
(radians(anomaliRataM)) + -1 * anomaliRataB + 0 *
argumenBujurB);
  double bujur12 = -34720 * (pow(eksentrisitasOrbit,
(abs(0)))) * sin(1 * elongasiRataB + 0 *
(radians(anomaliRataM)) + 0 * anomaliRataB + 0 *
argumenBujurB);
  double bujur13 = -30383 * (pow(eksentrisitasOrbit,
(abs(1)))) * sin(0 * elongasiRataB + 1 *
(radians(anomaliRataM)) + 1 * anomaliRataB + 0 *
argumenBujurB);
  double bujur14 = 15327 * (pow(eksentrisitasOrbit,
(abs(0)))) * sin(2 * elongasiRataB + 0 *
(radians(anomaliRataM)) + 0 * anomaliRataB + -2 *
argumenBujurB);
  double bujur15 = -12528 * (pow(eksentrisitasOrbit,
(abs(0)))) * sin(0 * elongasiRataB + 0 *
```

```
(radians(anomaliRataM)) + 1 * anomaliRataB + 2 *
argumenBujurB);
  double bujur16 = 10980 * (pow(eksentrisitasOrbit,
(abs(0))) * sin(0 * elongasiRataB + 0 *
(radians(anomaliRataM)) + 1 * anomaliRataB + -2 *
argumenBujurB);
  double bujur17 = 10675 * (pow(eksentrisitasOrbit,
(abs(0))) * sin(4 * elongasiRataB + 0 *
(radians(anomaliRataM)) + -1 * anomaliRataB + 0 *
argumenBujurB);
  double bujur18 = 10034 * (pow(eksentrisitasOrbit,
(abs(0))) * sin(0 * elongasiRataB + 0 *
(radians(anomaliRataM)) + 3 * anomaliRataB + 0 *
argumenBujurB);
  double bujur19 = 8548 * (pow(eksentrisitasOrbit,
(abs(0))) * sin(4 * elongasiRataB + 0 *
(radians(anomaliRataM)) + -2 * anomaliRataB + 0 *
argumenBujurB);
  double bujur20 = -7888 * (pow(eksentrisitasOrbit,
(abs(1))) * sin(2 * elongasiRataB + 1 *
(radians(anomaliRataM)) + -1 * anomaliRataB + 0 *
argumenBujurB);
  double bujur21 = -6766 * (pow(eksentrisitasOrbit,
(abs(1))) * sin(2 * elongasiRataB + 1 *
(radians(anomaliRataM)) + 0 * anomaliRataB + 0 *
argumenBujurB);
  double bujur22 = -5163 * (pow(eksentrisitasOrbit,
(abs(0))) * sin(1 * elongasiRataB + 0 *
(radians(anomaliRataM)) + -1 * anomaliRataB + 0 *
argumenBujurB);
  double bujur23 = 4987 * (pow(eksentrisitasOrbit,
(abs(1))) * sin(1 * elongasiRataB + 1 *
(radians(anomaliRataM)) + 0 * anomaliRataB + 0 *
argumenBujurB);
  double bujur24 = 4036 * (pow(eksentrisitasOrbit,
(abs(-1))) * sin(2 * elongasiRataB + -1 *
(radians(anomaliRataM)) + 1 * anomaliRataB + 0 *
argumenBujurB);
  double bujur25 = 3994 * (pow(eksentrisitasOrbit,
(abs(0))) * sin(2 * elongasiRataB + 0 *
(radians(anomaliRataM)) + 2 * anomaliRataB + 0 *
argumenBujurB);
  double bujur26 = 3861 * (pow(eksentrisitasOrbit,
(abs(0))) * sin(4 * elongasiRataB + 0 *
```

```
(radians(anomaliRataM)) + 0 * anomaliRataB + 0 *
argumenBujurB);
  double bujur27 = 3665 * (pow(eksentrisitasOrbit,
(abs(0))) * sin(2 * elongasiRataB + 0 *
(radians(anomaliRataM)) + -3 * anomaliRataB + 0 *
argumenBujurB);
  double bujur28 = -2689 * (pow(eksentrisitasOrbit,
(abs(1))) * sin(0 * elongasiRataB + 1 *
(radians(anomaliRataM)) + -2 * anomaliRataB + 0 *
argumenBujurB);
  double bujur29 = -2602 * (pow(eksentrisitasOrbit,
(abs(0))) * sin(2 * elongasiRataB + 0 *
(radians(anomaliRataM)) + -1 * anomaliRataB + 2 *
argumenBujurB);
  double bujur30 = 2390 * (pow(eksentrisitasOrbit,
(abs(-1))) * sin(2 * elongasiRataB + -1 *
(radians(anomaliRataM)) + -2 * anomaliRataB + 0 *
argumenBujurB);
  double bujur31 = -2348 * (pow(eksentrisitasOrbit,
(abs(0))) * sin(1 * elongasiRataB + 0 *
(radians(anomaliRataM)) + 1 * anomaliRataB + 0 *
argumenBujurB);
  double bujur32 = 2236 * (pow(eksentrisitasOrbit,
(abs(-2))) * sin(2 * elongasiRataB + -2 *
(radians(anomaliRataM)) + 0 * anomaliRataB + 0 *
argumenBujurB);
  double bujur33 = -2120 * (pow(eksentrisitasOrbit,
(abs(1))) * sin(0 * elongasiRataB + 1 *
(radians(anomaliRataM)) + 2 * anomaliRataB + 0 *
argumenBujurB);
  double bujur34 = -2069 * (pow(eksentrisitasOrbit,
(abs(2))) * sin(0 * elongasiRataB + 2 *
(radians(anomaliRataM)) + 0 * anomaliRataB + 0 *
argumenBujurB);
  double bujur35 = 2048 * (pow(eksentrisitasOrbit,
(abs(-2))) * sin(2 * elongasiRataB + -2 *
(radians(anomaliRataM)) + -1 * anomaliRataB + 0 *
argumenBujurB);
  double bujur36 = -1773 * (pow(eksentrisitasOrbit,
(abs(0))) * sin(2 * elongasiRataB + 0 *
(radians(anomaliRataM)) + 1 * anomaliRataB + -2 *
argumenBujurB);
  double bujur37 = -1595 * (pow(eksentrisitasOrbit,
(abs(0))) * sin(2 * elongasiRataB + 0 *
```

```
(radians(anomaliRataM)) + 0 * anomaliRataB + 2 *
argumenBujurB);
  double bujur38 = 1215 * (pow(eksentrisitasOrbit,
(abs(-1)))) * sin(4 * elongasiRataB + -1 *
(radians(anomaliRataM)) + -1 * anomaliRataB + 0 *
argumenBujurB);
  double bujur39 = -1110 * (pow(eksentrisitasOrbit,
(abs(0)))) * sin(0 * elongasiRataB + 0 *
(radians(anomaliRataM)) + 2 * anomaliRataB + 2 *
argumenBujurB);
  double bujur40 = -892 * (pow(eksentrisitasOrbit,
(abs(0)))) * sin(3 * elongasiRataB + 0 *
(radians(anomaliRataM)) + -1 * anomaliRataB + 0 *
argumenBujurB);
  double bujur41 = -810 * (pow(eksentrisitasOrbit,
(abs(1)))) * sin(2 * elongasiRataB + 1 *
(radians(anomaliRataM)) + 1 * anomaliRataB + 0 *
argumenBujurB);
  double bujur42 = 759 * (pow(eksentrisitasOrbit, (abs(-
1)))) * sin(4 * elongasiRataB + -1 *
(radians(anomaliRataM)) + -2 * anomaliRataB + 0 *
argumenBujurB);
  double bujur43 = -713 * (pow(eksentrisitasOrbit,
(abs(2)))) * sin(0 * elongasiRataB + 2 *
(radians(anomaliRataM)) + -1 * anomaliRataB + 0 *
argumenBujurB);
  double bujur44 = -700 * (pow(eksentrisitasOrbit,
(abs(2)))) * sin(2 * elongasiRataB + 2 *
(radians(anomaliRataM)) + -1 * anomaliRataB + 0 *
argumenBujurB);
  double bujur45 = 691 * (pow(eksentrisitasOrbit,
(abs(1)))) * sin(2 * elongasiRataB + 1 *
(radians(anomaliRataM)) + -2 * anomaliRataB + 0 *
argumenBujurB);
  double bujur46 = 596 * (pow(eksentrisitasOrbit, (abs(-
1)))) * sin(2 * elongasiRataB + -1 *
(radians(anomaliRataM)) + 0 * anomaliRataB + -2 *
argumenBujurB);
  double bujur47 = 549 * (pow(eksentrisitasOrbit,
(abs(0)))) * sin(4 * elongasiRataB + 0 *
(radians(anomaliRataM)) + 1 * anomaliRataB + 0 *
argumenBujurB);
  double bujur48 = 537 * (pow(eksentrisitasOrbit,
(abs(0)))) * sin(0 * elongasiRataB + 0 *
```

```
(radians(anomaliRataM)) + 4 * anomaliRataB + 0 *
argumenBujurB);
  double bujur49 = 520 * (pow(eksentrisitasOrbit, (abs(-
1)))) * sin(4 * elongasiRataB + -1 *
(radians(anomaliRataM)) + 0 * anomaliRataB + 0 *
argumenBujurB);
  double bujur50 = -487 * (pow(eksentrisitasOrbit,
(abs(0)))) * sin(1 * elongasiRataB + 0 *
(radians(anomaliRataM)) + -2 * anomaliRataB + 0 *
argumenBujurB);
  double bujur51 = -399 * (pow(eksentrisitasOrbit,
(abs(1)))) * sin(2 * elongasiRataB + 1 *
(radians(anomaliRataM)) + 0 * anomaliRataB + -2 *
argumenBujurB);
  double bujur52 = -381 * (pow(eksentrisitasOrbit,
(abs(0)))) * sin(0 * elongasiRataB + 0 *
(radians(anomaliRataM)) + 2 * anomaliRataB + -2 *
argumenBujurB);
  double bujur53 = 351 * (pow(eksentrisitasOrbit,
(abs(1)))) * sin(1 * elongasiRataB + 1 *
(radians(anomaliRataM)) + 1 * anomaliRataB + 0 *
argumenBujurB);
  double bujur54 = -340 * (pow(eksentrisitasOrbit,
(abs(0)))) * sin(3 * elongasiRataB + 0 *
(radians(anomaliRataM)) + -2 * anomaliRataB + 0 *
argumenBujurB);
  double bujur55 = 330 * (pow(eksentrisitasOrbit,
(abs(0)))) * sin(4 * elongasiRataB + 0 *
(radians(anomaliRataM)) + -3 * anomaliRataB + 0 *
argumenBujurB);
  double bujur56 = 327 * (pow(eksentrisitasOrbit, (abs(-
1)))) * sin(2 * elongasiRataB + -1 *
(radians(anomaliRataM)) + 2 * anomaliRataB + 0 *
argumenBujurB);
  double bujur57 = -323 * (pow(eksentrisitasOrbit,
(abs(2)))) * sin(0 * elongasiRataB + 2 *
(radians(anomaliRataM)) + 1 * anomaliRataB + 0 *
argumenBujurB);
  double bujur58 = 299 * (pow(eksentrisitasOrbit,
(abs(1)))) * sin(1 * elongasiRataB + 1 *
(radians(anomaliRataM)) + -1 * anomaliRataB + 0 *
argumenBujurB);
  double bujur59 = 294 * (pow(eksentrisitasOrbit,
(abs(0)))) * sin(2 * elongasiRataB + 0 *
```

```
(radians(anomaliRataM)) + 3 * anomaliRataB + 0 *
argumenBujurB);
  double sumB = bujur1 + bujur2 + bujur3 + bujur4 +
bujur5 + bujur6 + bujur7 + bujur8 + bujur9 + bujur10 +
bujur11 + bujur12 + bujur13 + bujur14 + bujur15 + bujur16
+ bujur17 + bujur18 + bujur19 + bujur20 + bujur21 +
bujur22 + bujur23 + bujur24 + bujur25 + bujur26 + bujur27
+ bujur28 + bujur29 + bujur30 + bujur31 + bujur32 +
bujur33 + bujur34 + bujur35 + bujur36 + bujur37 + bujur38
+ bujur39 + bujur40 + bujur41 + bujur42 + bujur43 +
bujur44 + bujur45 + bujur46 + bujur47 + bujur48 + bujur49
+ bujur50 + bujur51 + bujur52 + bujur53 + bujur54 +
bujur55 + bujur56 + bujur57 + bujur58 + bujur59;
  double koreksiBujurB = (sumB + 3958 * sin(argumenA1)
+ 1962 * sin(bujurRataB - argumenBujurB) + 318 *
sin(argumenA2)) / 1000000;
  double bujurBterkoreksi = fmod((degrees(bujurRataB) +
koreksiBujurB) + 360, 360);
  //Koreksi Lintang Bulan
  double lintang1 = 5128122 * (pow(eksentrisitasOrbit,
(abs(0))) * sin(0 * elongasiRataB + 0 *
(radians(anomaliRataM)) + 0 * anomaliRataB + 1 *
argumenBujurB);
  double lintang2 = 280602 * (pow(eksentrisitasOrbit,
(abs(0))) * sin(0 * elongasiRataB + 0 *
(radians(anomaliRataM)) + 1 * anomaliRataB + 1 *
argumenBujurB);
  double lintang3 = 277693 * (pow(eksentrisitasOrbit,
(abs(0))) * sin(0 * elongasiRataB + 0 *
(radians(anomaliRataM)) + 1 * anomaliRataB + -1 *
argumenBujurB);
  double lintang4 = 173237 * (pow(eksentrisitasOrbit,
(abs(0))) * sin(2 * elongasiRataB + 0 *
(radians(anomaliRataM)) + 0 * anomaliRataB + -1 *
argumenBujurB);
  double lintang5 = 55413 * (pow(eksentrisitasOrbit,
(abs(0))) * sin(2 * elongasiRataB + 0 *
(radians(anomaliRataM)) + -1 * anomaliRataB + 1 *
argumenBujurB);
  double lintang6 = 46271 * (pow(eksentrisitasOrbit,
(abs(0))) * sin(2 * elongasiRataB + 0 *
(radians(anomaliRataM)) + -1 * anomaliRataB + -1 *
argumenBujurB);
```

```
  double lintang7 = 32573 * (pow(eksentrisitasOrbit,
(abs(0))) * sin(2 * elongasiRataB + 0 *
(radians(anomaliRataM)) + 0 * anomaliRataB + 1 *
argumenBujurB);
  double lintang8 = 17198 * (pow(eksentrisitasOrbit,
(abs(0))) * sin(0 * elongasiRataB + 0 *
(radians(anomaliRataM)) + 2 * anomaliRataB + 1 *
argumenBujurB);
  double lintang9 = 9266 * (pow(eksentrisitasOrbit,
(abs(0))) * sin(2 * elongasiRataB + 0 *
(radians(anomaliRataM)) + 1 * anomaliRataB + -1 *
argumenBujurB);
  double lintang10 = 8822 * (pow(eksentrisitasOrbit,
(abs(0))) * sin(0 * elongasiRataB + 0 *
(radians(anomaliRataM)) + 2 * anomaliRataB + -1 *
argumenBujurB);
  double lintang11 = 8216 * (pow(eksentrisitasOrbit,
(abs(-1))) * sin(2 * elongasiRataB + -1 *
(radians(anomaliRataM)) + 0 * anomaliRataB + -1 *
argumenBujurB);
  double lintang12 = 4324 * (pow(eksentrisitasOrbit,
(abs(0))) * sin(2 * elongasiRataB + 0 *
(radians(anomaliRataM)) + -2 * anomaliRataB + -1 *
argumenBujurB);
  double lintang13 = 4200 * (pow(eksentrisitasOrbit,
(abs(0))) * sin(2 * elongasiRataB + 0 *
(radians(anomaliRataM)) + 1 * anomaliRataB + 1 *
argumenBujurB);
  double lintang14 = -3359 * (pow(eksentrisitasOrbit,
(abs(1))) * sin(2 * elongasiRataB + 1 *
(radians(anomaliRataM)) + 0 * anomaliRataB + -1 *
argumenBujurB);
  double lintang15 = 2463 * (pow(eksentrisitasOrbit,
(abs(-1))) * sin(2 * elongasiRataB + -1 *
(radians(anomaliRataM)) + -1 * anomaliRataB + 1 *
argumenBujurB);
  double lintang16 = 2211 * (pow(eksentrisitasOrbit,
(abs(-1))) * sin(2 * elongasiRataB + -1 *
(radians(anomaliRataM)) + 0 * anomaliRataB + 1 *
argumenBujurB);
  double lintang17 = 2065 * (pow(eksentrisitasOrbit,
(abs(-1))) * sin(2 * elongasiRataB + -1 *
(radians(anomaliRataM)) + -1 * anomaliRataB + -1 *
argumenBujurB);
```

```
  double lintang18 = -1870 * (pow(eksentrisitasOrbit,
(abs(1))) * sin(0 * elongasiRataB + 1 *
(radians(anomaliRataM)) + -1 * anomaliRataB + -1 *
argumenBujurB);
  double lintang19 = 1828 * (pow(eksentrisitasOrbit,
(abs(0))) * sin(4 * elongasiRataB + 0 *
(radians(anomaliRataM)) + -1 * anomaliRataB + -1 *
argumenBujurB);
  double lintang20 = -1794 * (pow(eksentrisitasOrbit,
(abs(1))) * sin(0 * elongasiRataB + 1 *
(radians(anomaliRataM)) + 0 * anomaliRataB + 1 *
argumenBujurB);
  double lintang21 = -1749 * (pow(eksentrisitasOrbit,
(abs(0))) * sin(0 * elongasiRataB + 0 *
(radians(anomaliRataM)) + 0 * anomaliRataB + 3 *
argumenBujurB);
  double lintang22 = -1565 * (pow(eksentrisitasOrbit,
(abs(1))) * sin(0 * elongasiRataB + 1 *
(radians(anomaliRataM)) + -1 * anomaliRataB + 1 *
argumenBujurB);
  double lintang23 = -1491 * (pow(eksentrisitasOrbit,
(abs(0))) * sin(1 * elongasiRataB + 0 *
(radians(anomaliRataM)) + 0 * anomaliRataB + 1 *
argumenBujurB);
  double lintang24 = -1475 * (pow(eksentrisitasOrbit,
(abs(1))) * sin(0 * elongasiRataB + 1 *
(radians(anomaliRataM)) + 1 * anomaliRataB + 1 *
argumenBujurB);
  double lintang25 = -1410 * (pow(eksentrisitasOrbit,
(abs(1))) * sin(0 * elongasiRataB + 1 *
(radians(anomaliRataM)) + 1 * anomaliRataB + -1 *
argumenBujurB);
  double lintang26 = -1344 * (pow(eksentrisitasOrbit,
(abs(1))) * sin(0 * elongasiRataB + 1 *
(radians(anomaliRataM)) + 0 * anomaliRataB + -1 *
argumenBujurB);
  double lintang27 = -1335 * (pow(eksentrisitasOrbit,
(abs(0))) * sin(1 * elongasiRataB + 0 *
(radians(anomaliRataM)) + 0 * anomaliRataB + -1 *
argumenBujurB);
  double lintang28 = 1107 * (pow(eksentrisitasOrbit,
(abs(0))) * sin(0 * elongasiRataB + 0 *
(radians(anomaliRataM)) + 3 * anomaliRataB + 1 *
argumenBujurB);
```

```
  double lintang29 = 1021 * (pow(eksentrisitasOrbit,
(abs(0))) * sin(4 * elongasiRataB + 0 *
(radians(anomaliRataM)) + 0 * anomaliRataB + -1 *
argumenBujurB);
  double lintang30 = 833 * (pow(eksentrisitasOrbit,
(abs(0))) * sin(4 * elongasiRataB + 0 *
(radians(anomaliRataM)) + -1 * anomaliRataB + 1 *
argumenBujurB);
  double lintang31 = 777 * (pow(eksentrisitasOrbit,
(abs(0))) * sin(0 * elongasiRataB + 0 *
(radians(anomaliRataM)) + 1 * anomaliRataB + -3 *
argumenBujurB);
  double lintang32 = 671 * (pow(eksentrisitasOrbit,
(abs(0))) * sin(4 * elongasiRataB + 0 *
(radians(anomaliRataM)) + -2 * anomaliRataB + 1 *
argumenBujurB);
  double lintang33 = 607 * (pow(eksentrisitasOrbit,
(abs(0))) * sin(2 * elongasiRataB + 0 *
(radians(anomaliRataM)) + 0 * anomaliRataB + -3 *
argumenBujurB);
  double lintang34 = 596 * (pow(eksentrisitasOrbit,
(abs(0))) * sin(2 * elongasiRataB + 0 *
(radians(anomaliRataM)) + 2 * anomaliRataB + -1 *
argumenBujurB);
  double lintang35 = 491 * (pow(eksentrisitasOrbit,
(abs(-1))) * sin(2 * elongasiRataB + -1 *
(radians(anomaliRataM)) + 1 * anomaliRataB + -1 *
argumenBujurB);
  double lintang36 = -451 * (pow(eksentrisitasOrbit,
(abs(0))) * sin(2 * elongasiRataB + 0 *
(radians(anomaliRataM)) + -2 * anomaliRataB + 1 *
argumenBujurB);
  double lintang37 = 439 * (pow(eksentrisitasOrbit,
(abs(0))) * sin(0 * elongasiRataB + 0 *
(radians(anomaliRataM)) + 3 * anomaliRataB + -1 *
argumenBujurB);
  double lintang38 = 422 * (pow(eksentrisitasOrbit,
(abs(0))) * sin(2 * elongasiRataB + 0 *
(radians(anomaliRataM)) + 2 * anomaliRataB + 1 *
argumenBujurB);
  double lintang39 = 421 * (pow(eksentrisitasOrbit,
(abs(0))) * sin(2 * elongasiRataB + 0 *
(radians(anomaliRataM)) + -3 * anomaliRataB + -1 *
argumenBujurB);
```

```
  double lintang40 = -366 * (pow(eksentrisitasOrbit,
(abs(1))) * sin(2 * elongasiRataB + 1 *
(radians(anomaliRataM)) + -1 * anomaliRataB + 1 *
argumenBujurB);
  double lintang41 = -351 * (pow(eksentrisitasOrbit,
(abs(1))) * sin(2 * elongasiRataB + 1 *
(radians(anomaliRataM)) + 0 * anomaliRataB + 1 *
argumenBujurB);
  double lintang42 = 331 * (pow(eksentrisitasOrbit,
(abs(0))) * sin(4 * elongasiRataB + 0 *
(radians(anomaliRataM)) + 0 * anomaliRataB + 1 *
argumenBujurB);
  double lintang43 = 315 * (pow(eksentrisitasOrbit,
(abs(-1))) * sin(2 * elongasiRataB + -1 *
(radians(anomaliRataM)) + 1 * anomaliRataB + 1 *
argumenBujurB);
  double lintang44 = 302 * (pow(eksentrisitasOrbit,
(abs(-2))) * sin(2 * elongasiRataB + -2 *
(radians(anomaliRataM)) + 0 * anomaliRataB + -1 *
argumenBujurB);
  double lintang45 = -283 * (pow(eksentrisitasOrbit,
(abs(0))) * sin(0 * elongasiRataB + 0 *
(radians(anomaliRataM)) + 1 * anomaliRataB + 3 *
argumenBujurB);
  double lintang46 = -229 * (pow(eksentrisitasOrbit,
(abs(1))) * sin(2 * elongasiRataB + 1 *
(radians(anomaliRataM)) + 1 * anomaliRataB + -1 *
argumenBujurB);
  double lintang47 = 223 * (pow(eksentrisitasOrbit,
(abs(1))) * sin(1 * elongasiRataB + 1 *
(radians(anomaliRataM)) + 0 * anomaliRataB + -1 *
argumenBujurB);
  double lintang48 = 223 * (pow(eksentrisitasOrbit,
(abs(1))) * sin(1 * elongasiRataB + 1 *
(radians(anomaliRataM)) + 0 * anomaliRataB + 1 *
argumenBujurB);
  double lintang49 = -220 * (pow(eksentrisitasOrbit,
(abs(1))) * sin(0 * elongasiRataB + 1 *
(radians(anomaliRataM)) + -2 * anomaliRataB + -1 *
argumenBujurB);
  double lintang50 = -220 * (pow(eksentrisitasOrbit,
(abs(1))) * sin(2 * elongasiRataB + 1 *
(radians(anomaliRataM)) + -1 * anomaliRataB + -1 *
argumenBujurB);
```

```
  double lintang51 = -185 * (pow(eksentrisitasOrbit,
(abs(0))) * sin(1 * elongasiRataB + 0 *
(radians(anomaliRataM)) + 1 * anomaliRataB + 1 *
argumenBujurB);
  double lintang52 = 181 * (pow(eksentrisitasOrbit,
(abs(-1))) * sin(2 * elongasiRataB + -1 *
(radians(anomaliRataM)) + -2 * anomaliRataB + -1 *
argumenBujurB);
  double lintang53 = -177 * (pow(eksentrisitasOrbit,
(abs(1))) * sin(0 * elongasiRataB + 1 *
(radians(anomaliRataM)) + 2 * anomaliRataB + 1 *
argumenBujurB);
  double lintang54 = 176 * (pow(eksentrisitasOrbit,
(abs(0))) * sin(4 * elongasiRataB + 0 *
(radians(anomaliRataM)) + -2 * anomaliRataB + -1 *
argumenBujurB);
  double lintang55 = 166 * (pow(eksentrisitasOrbit,
(abs(-1))) * sin(4 * elongasiRataB + -1 *
(radians(anomaliRataM)) + -1 * anomaliRataB + -1 *
argumenBujurB);
  double lintang56 = -164 * (pow(eksentrisitasOrbit,
(abs(0))) * sin(1 * elongasiRataB + 0 *
(radians(anomaliRataM)) + 1 * anomaliRataB + -1 *
argumenBujurB);
  double lintang57 = 132 * (pow(eksentrisitasOrbit,
(abs(0))) * sin(4 * elongasiRataB + 0 *
(radians(anomaliRataM)) + 1 * anomaliRataB + -1 *
argumenBujurB);
  double lintang58 = -119 * (pow(eksentrisitasOrbit,
(abs(0))) * sin(1 * elongasiRataB + 0 *
(radians(anomaliRataM)) + -1 * anomaliRataB + -1 *
argumenBujurB);
  double lintang59 = 115 * (pow(eksentrisitasOrbit,
(abs(-1))) * sin(4 * elongasiRataB + -1 *
(radians(anomaliRataM)) + 0 * anomaliRataB + -1 *
argumenBujurB);
  double lintang60 = 107 * (pow(eksentrisitasOrbit,
(abs(-2))) * sin(2 * elongasiRataB + -2 *
(radians(anomaliRataM)) + 0 * anomaliRataB + 1 *
argumenBujurB);
  double sumL = lintang1 + lintang2 + lintang3 + lintang4
+ lintang5 + lintang6 + lintang7 + lintang8 + lintang9
+ lintang10 + lintang11 + lintang12 + lintang13 +
lintang14 + lintang15 + lintang16 + lintang17 +
```

```
lintang18 + lintang19 + lintang20 + lintang21 +
lintang22 + lintang23 + lintang24 + lintang25 +
lintang26 + lintang27 + lintang28 + lintang29 +
lintang30 + lintang31 + lintang32 + lintang33 +
lintang34 + lintang35 + lintang36 + lintang37 +
lintang38 + lintang39 + lintang40 + lintang41 +
lintang42 + lintang43 + lintang44 + lintang45 +
lintang46 + lintang47 + lintang48 + lintang49 +
lintang50 + lintang51 + lintang52 + lintang53 +
lintang54 + lintang55 + lintang56 + lintang57 +
lintang58 + lintang59 + lintang60;
  double lintangB = (sumL - 2235 * sin(bujurRataB) + 382
* sin(argumenA3) + 175 * sin(argumenA1 - argumenBujurB)
+ 175 * sin(argumenA1 + argumenBujurB) + 127 *
sin(bujurRataB - anomaliRataB) - 115 * sin(bujurRataB +
anomaliRataB)) / 1000000;
  //Jarak Bulan
  double jarakBln1 = -20905355 *
(pow(eksentrisitasOrbit, (abs(0)))) * cos(0 *
elongasiRataB + 0 * (radians(anomaliRataM)) + 1 *
anomaliRataB + 0 * argumenBujurB);
  double jarakBln2 = -3699111 * (pow(eksentrisitasOrbit,
(abs(0)))) * cos(2 * elongasiRataB + 0 *
(radians(anomaliRataM)) + -1 * anomaliRataB + 0 *
argumenBujurB);
  double jarakBln3 = -2955968 * (pow(eksentrisitasOrbit,
(abs(0)))) * cos(2 * elongasiRataB + 0 *
(radians(anomaliRataM)) + 0 * anomaliRataB + 0 *
argumenBujurB);
  double jarakBln4 = -569925 * (pow(eksentrisitasOrbit,
(abs(0)))) * cos(0 * elongasiRataB + 0 *
(radians(anomaliRataM)) + 2 * anomaliRataB + 0 *
argumenBujurB);
  double jarakBln5 = 246158 * (pow(eksentrisitasOrbit,
(abs(0)))) * cos(2 * elongasiRataB + 0 *
(radians(anomaliRataM)) + -2 * anomaliRataB + 0 *
argumenBujurB);
  double jarakBln6 = -204586 * (pow(eksentrisitasOrbit,
(abs(-1)))) * cos(2 * elongasiRataB + -1 *
(radians(anomaliRataM)) + 0 * anomaliRataB + 0 *
argumenBujurB);
  double jarakBln7 = -170733 * (pow(eksentrisitasOrbit,
(abs(0)))) * cos(2 * elongasiRataB + 0 *
```

```
(radians(anomaliRataM)) + 1 * anomaliRataB + 0 *
argumenBujurB);
  double jarakBln8 = -152138 * (pow(eksentrisitasOrbit,
(abs(-1)))) * cos(2 * elongasiRataB + -1 *
(radians(anomaliRataM)) + -1 * anomaliRataB + 0 *
argumenBujurB);
  double jarakBln9 = -129620 * (pow(eksentrisitasOrbit,
(abs(1)))) * cos(0 * elongasiRataB + 1 *
(radians(anomaliRataM)) + -1 * anomaliRataB + 0 *
argumenBujurB);
  double jarakBln10 = 108743 * (pow(eksentrisitasOrbit,
(abs(0)))) * cos(1 * elongasiRataB + 0 *
(radians(anomaliRataM)) + 0 * anomaliRataB + 0 *
argumenBujurB);
  double jarakBln11 = 104755 * (pow(eksentrisitasOrbit,
(abs(1)))) * cos(0 * elongasiRataB + 1 *
(radians(anomaliRataM)) + 1 * anomaliRataB + 0 *
argumenBujurB);
  double jarakBln12 = 79661 * (pow(eksentrisitasOrbit,
(abs(0)))) * cos(0 * elongasiRataB + 0 *
(radians(anomaliRataM)) + 1 * anomaliRataB + -2 *
argumenBujurB);
  double jarakBln13 = 48888 * (pow(eksentrisitasOrbit,
(abs(1)))) * cos(0 * elongasiRataB + 1 *
(radians(anomaliRataM)) + 0 * anomaliRataB + 0 *
argumenBujurB);
  double jarakBln14 = -34782 * (pow(eksentrisitasOrbit,
(abs(0)))) * cos(4 * elongasiRataB + 0 *
(radians(anomaliRataM)) + -1 * anomaliRataB + 0 *
argumenBujurB);
  double jarakBln15 = 30824 * (pow(eksentrisitasOrbit,
(abs(1)))) * cos(2 * elongasiRataB + 1 *
(radians(anomaliRataM)) + 0 * anomaliRataB + 0 *
argumenBujurB);
  double jarakBln16 = 24208 * (pow(eksentrisitasOrbit,
(abs(1)))) * cos(2 * elongasiRataB + 1 *
(radians(anomaliRataM)) + -1 * anomaliRataB + 0 *
argumenBujurB);
  double jarakBln17 = -23210 * (pow(eksentrisitasOrbit,
(abs(0)))) * cos(0 * elongasiRataB + 0 *
(radians(anomaliRataM)) + 3 * anomaliRataB + 0 *
argumenBujurB);
  double jarakBln18 = -21636 * (pow(eksentrisitasOrbit,
(abs(0)))) * cos(4 * elongasiRataB + 0 *
```

```
(radians(anomaliRataM)) + -2 * anomaliRataB + 0 *
argumenBujurB);
  double jarakBln19 = -16675 * (pow(eksentrisitasOrbit,
(abs(1)))) * cos(1 * elongasiRataB + 1 *
(radians(anomaliRataM)) + 0 * anomaliRataB + 0 *
argumenBujurB);
  double jarakBln20 = 14403 * (pow(eksentrisitasOrbit,
(abs(0)))) * cos(2 * elongasiRataB + 0 *
(radians(anomaliRataM)) + -3 * anomaliRataB + 0 *
argumenBujurB);
  double jarakBln21 = -12831 * (pow(eksentrisitasOrbit,
(abs(-1)))) * cos(2 * elongasiRataB + -1 *
(radians(anomaliRataM)) + 1 * anomaliRataB + 0 *
argumenBujurB);
  double jarakBln22 = -11650 * (pow(eksentrisitasOrbit,
(abs(0)))) * cos(4 * elongasiRataB + 0 *
(radians(anomaliRataM)) + 0 * anomaliRataB + 0 *
argumenBujurB);
  double jarakBln23 = -10445 * (pow(eksentrisitasOrbit,
(abs(0)))) * cos(2 * elongasiRataB + 0 *
(radians(anomaliRataM)) + 2 * anomaliRataB + 0 *
argumenBujurB);
  double jarakBln24 = 10321 * (pow(eksentrisitasOrbit,
(abs(0)))) * cos(2 * elongasiRataB + 0 *
(radians(anomaliRataM)) + 0 * anomaliRataB + -2 *
argumenBujurB);
  double jarakBln25 = 10056 * (pow(eksentrisitasOrbit,
(abs(-1)))) * cos(2 * elongasiRataB + -1 *
(radians(anomaliRataM)) + -2 * anomaliRataB + 0 *
argumenBujurB);
  double jarakBln26 = -9884 * (pow(eksentrisitasOrbit,
(abs(-2)))) * cos(2 * elongasiRataB + -2 *
(radians(anomaliRataM)) + 0 * anomaliRataB + 0 *
argumenBujurB);
  double jarakBln27 = 8752 * (pow(eksentrisitasOrbit,
(abs(0)))) * cos(2 * elongasiRataB + 0 *
(radians(anomaliRataM)) + -1 * anomaliRataB + -2 *
argumenBujurB);
  double jarakBln28 = -8379 * (pow(eksentrisitasOrbit,
(abs(0)))) * cos(1 * elongasiRataB + 0 *
(radians(anomaliRataM)) + -1 * anomaliRataB + 0 *
argumenBujurB);
  double jarakBln29 = -7003 * (pow(eksentrisitasOrbit,
(abs(1)))) * cos(0 * elongasiRataB + 1 *
```

```
(radians(anomaliRataM)) + -2 * anomaliRataB + 0 *
argumenBujurB);
  double jarakBln30 = 6322 * (pow(eksentrisitasOrbit,
(abs(0))) * cos(1 * elongasiRataB + 0 *
(radians(anomaliRataM)) + 1 * anomaliRataB + 0 *
argumenBujurB);
  double jarakBln31 = 5751 * (pow(eksentrisitasOrbit,
(abs(1))) * cos(0 * elongasiRataB + 1 *
(radians(anomaliRataM)) + 2 * anomaliRataB + 0 *
argumenBujurB);
  double jarakBln32 = -4950 * (pow(eksentrisitasOrbit,
(abs(-2))) * cos(2 * elongasiRataB + -2 *
(radians(anomaliRataM)) + -1 * anomaliRataB + 0 *
argumenBujurB);
  double jarakBln33 = -4421 * (pow(eksentrisitasOrbit,
(abs(0))) * cos(0 * elongasiRataB + 0 *
(radians(anomaliRataM)) + 2 * anomaliRataB + -2 *
argumenBujurB);
  double jarakBln34 = 4130 * (pow(eksentrisitasOrbit,
(abs(0))) * cos(2 * elongasiRataB + 0 *
(radians(anomaliRataM)) + 1 * anomaliRataB + -2 *
argumenBujurB);
  double jarakBln35 = -3958 * (pow(eksentrisitasOrbit,
(abs(-1))) * cos(4 * elongasiRataB + -1 *
(radians(anomaliRataM)) + -1 * anomaliRataB + 0 *
argumenBujurB);
  double jarakBln36 = 3258 * (pow(eksentrisitasOrbit,
(abs(0))) * cos(3 * elongasiRataB + 0 *
(radians(anomaliRataM)) + -1 * anomaliRataB + 0 *
argumenBujurB);
  double jarakBln37 = -3149 * (pow(eksentrisitasOrbit,
(abs(0))) * cos(0 * elongasiRataB + 0 *
(radians(anomaliRataM)) + 0 * anomaliRataB + 2 *
argumenBujurB);
  double jarakBln38 = 2616 * (pow(eksentrisitasOrbit,
(abs(1))) * cos(2 * elongasiRataB + 1 *
(radians(anomaliRataM)) + 1 * anomaliRataB + 0 *
argumenBujurB);
  double jarakBln39 = 2354 * (pow(eksentrisitasOrbit,
(abs(2))) * cos(2 * elongasiRataB + 2 *
(radians(anomaliRataM)) + -1 * anomaliRataB + 0 *
argumenBujurB);
  double jarakBln40 = -2117 * (pow(eksentrisitasOrbit,
(abs(2))) * cos(0 * elongasiRataB + 2 *
```

```
(radians(anomaliRataM)) + -1 * anomaliRataB + 0 *
argumenBujurB);
  double jarakBln41 = -1897 * (pow(eksentrisitasOrbit,
(abs(-1)))) * cos(4 * elongasiRataB + -1 *
(radians(anomaliRataM)) + -2 * anomaliRataB + 0 *
argumenBujurB);
  double jarakBln42 = -1739 * (pow(eksentrisitasOrbit,
(abs(0)))) * cos(1 * elongasiRataB + 0 *
(radians(anomaliRataM)) + -2 * anomaliRataB + 0 *
argumenBujurB);
  double jarakBln43 = -1571 * (pow(eksentrisitasOrbit,
(abs(-1)))) * cos(4 * elongasiRataB + -1 *
(radians(anomaliRataM)) + 0 * anomaliRataB + 0 *
argumenBujurB);
  double jarakBln44 = -1423 * (pow(eksentrisitasOrbit,
(abs(0)))) * cos(4 * elongasiRataB + 0 *
(radians(anomaliRataM)) + 1 * anomaliRataB + 0 *
argumenBujurB);
  double jarakBln45 = 1165 * (pow(eksentrisitasOrbit,
(abs(2)))) * cos(0 * elongasiRataB + 2 *
(radians(anomaliRataM)) + 1 * anomaliRataB + 0 *
argumenBujurB);
  double jarakBln46 = -1117 * (pow(eksentrisitasOrbit,
(abs(0)))) * cos(0 * elongasiRataB + 0 *
(radians(anomaliRataM)) + 4 * anomaliRataB + 0 *
argumenBujurB);
  double jarakBulan = 385000.56 + (jarakBln1 +
jarakBln2 + jarakBln3 + jarakBln4 + jarakBln5 +
jarakBln6 + jarakBln7 + jarakBln8 + jarakBln9 +
jarakBln10 + jarakBln11 + jarakBln12 + jarakBln13 +
jarakBln14 + jarakBln15 + jarakBln16 + jarakBln17 +
jarakBln18 + jarakBln19 + jarakBln20 + jarakBln21 +
jarakBln22 + jarakBln23 + jarakBln24 + jarakBln25 +
jarakBln26 + jarakBln27 + jarakBln28 + jarakBln29 +
jarakBln30 + jarakBln31 + jarakBln32 + jarakBln33 +
jarakBln34 + jarakBln35 + jarakBln36 + jarakBln37 +
jarakBln38 + jarakBln39 + jarakBln40 + jarakBln41 +
jarakBln42 + jarakBln43 + jarakBln44 + jarakBln45 +
jarakBln46) / 1000;
  double bujurB = fmod((bujurBterkoreksi + nutasi) +
360, 360);
  double alphaB =
fmod((degrees(atan2(sin(radians(bujurB)) *
cos(radians(epsilon)) - tan(radians(lintangB)) *
```

```
sin(radians(epsilon)), cos(radians(bujurB))))) + 360,
360);
  double deltaB = degrees(asin(sin(radians(lintangB)) *
cos(radians(epsilon)) + cos(radians(lintangB)) *
sin(radians(epsilon)) * sin(radians(bujurB))));
  double hourAngleB = LST * 15 - alphaB;
  double azimuthSelatanB =
degrees(atan2(sin(radians(hourAngleB)),
cos(radians(hourAngleB)) * sin(radians(l)) -
tan(radians(deltaB)) * cos(radians(l))));
  double azimuthB = fmod((azimuthSelatanB + 180) + 360,
360) ;
  double altitudeB = degrees(asin(sin(radians(l)) *
sin(radians(deltaB)) + cos(radians(l)) *
cos(radians(deltaB)) * cos(radians(hourAngleB))));
  double sudutFai = acos(sin(radians(deltaB)) *
sin(radians(deltaM)) + cos(radians(deltaB)) *
cos(radians(deltaM)) * cos(radians(alphaB)) -
radians(alphaM)));
  double sudutFase = atan2(jarakMatahari *
sin(sudutFai), jarakBulan - jarakMatahari *
cos(sudutFai));
  double iluminasiBulan = 100 * (1 + cos(sudutFase)) /
2;
```

DAFTAR RIWAYAT HIDUP

Nama	: Muhammad Sagise Ainun Naim
Tempat tanggal lahir	: Tegal, 18 September 1999
Alamat asal	: Ds. Lumingser Rt.16 Rw. 03 Kec. Adiwerna Kab. Tegal
Alamat sekarang	: Pondok Pesantern Raudlatut Thalibin Rt. 01 Rw.01 Tugurejo Kec. Tugu Kota Semarang
Handphone	: 083102311294
Email	: ainun_1802046117@student.walisongo.ac.id

Riwayat Pendidikan :

a. Pendidikan Formal
    1. SDN 2 Lumingser, Adiwerna, lulus tahun 2011
    2. MTS N Model Babakan, Tegal, lulus tahun 2014
    3. MAN 1 Tegal, lulus tahun 2018
b. Pendidikan Non Formal
    1. MDA Miftahul ‘Ulum Lumingser, lulus tahun 2010
    2. Pondok Pesantren Terpadu Al-Fatkhu Babakan, Tegal, lulus tahun 2014
c. Pengalaman Organisasi

1. Pengurus IPNU (Ikatan Pelajar Nahdlatul Ulama) Ranting Lumingser
2. Anggota Ansor NU Ranting Lumingser
3. Pengurus IRMA (Ikatan Remaja Masjid At-Taqwa) Lumingser
4. Anggota IPM (Ikatan Pemuda Melaten) Lumingser

Semarang, 8 Juni 2022
Penulis
Muhammad Sagise Ainun Naim

NIM: 1802046117