

**APLIKASI JADWAL WAKTU SALAT
DENGAN STANDAR JAM ATOM BMKG BERBASIS ANDROID
SKRIPSI**

Diajukan untuk Memenuhi Tugas dan Melengkapi Syarat Guna Memperoleh
Gelar Sarjana Program Strata I (S.1)

Dalam Ilmu Syariah dan Hukum



Oleh :

OBI ROBI'A AL ASLAMI

NIM. 1502046101

**PROGRAM STUDI ILMU FALAK
FAKULTAS SYARIAH DAN HUKUM
UNIVERSITAS ISLAM NEGERI WALISONGO
SEMARANG
2019**

Dr. H. Ahmad Izzuddin, M.Ag
Jl. Bukit Beringin Lestari Barat Kav. C131
Wonosari, Ngaliyan, Semarang

PERSETUJUAN PEMBIMBING

Lamp : 4 (empat) eks.

Hal : Naskah Skripsi

An. Sdra. Obi Robi'a Al Aslami

Assalamualaikum. Wr. Wb.

Setelah saya mengoreksi dan mengadakan perbaikan seperlunya,
bersama ini saya kirimkan naskah skripsi saudara :

Nama : Obi Robi'a Al-Aslami

NIM : 1502046101

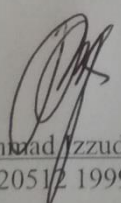
Judul Skripsi : Aplikasi Jadwal Waktu Salat dengan Standar Jam Atom
BMKG Berbasis Android.

Dengan ini saya mohon kepada Dekan Fakultas Syariah dan Hukum UIN
Walisongo, kiranya skripsi saudara tersebut dapat segera dimunaqosyahkan.

Demikian harap menjadikan maklum.

Wassalamualaikum. Wr. Wb.

Pembimbing I


Dr. H. Ahmad Izzuddin, M.Ag
NIP. 19720512 199903 1 003

Anthin Lathifah, M.Ag
Banjaran RT 04 RW 02 Beringin
Ngaliyan Semarang

PERSETUJUAN PEMBIMBING

Lamp : 4 (empat) eks.

Hal : Naskah Skripsi

An. Sdra. Obi Robi'a Al Aslami

Assalamualaikum. Wr. Wb.

Setelah saya mengoreksi dan mengadakan perbaikan seperlunya,
bersama ini saya kirimkan naskah skripsi saudara :

Nama : Obi Robi'a Al-Aslami

NIM : 1502046101

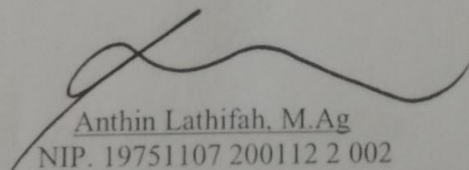
Judul Skripsi : Aplikasi Jadwal Waktu Salat dengan Standar Jam Atom
BMKG Berbasis Android.

Dengan ini saya mohon kepada Dekan Fakultas Syariah dan Hukum UIN
Walisongo, kiranya skripsi saudara tersebut dapat segera dimunaqosyahkan.

Demikian harap menjadikan maklum.

Wassalamualaikum. Wr. Wb.

Pembimbing II



Anthin Lathifah, M.Ag
NIP. 19751107 200112 2 002

MOTTO

إِنَّ الصَّلَاةَ كَانَتْ عَلَى الْمُؤْمِنِينَ كِتَابًا مَّوْقُوتًا

“Sesungguhnya salat itu adalah fardhu yang di tentukan waktunya atas orang orang yang beriman”.¹ (An-Nisa’ : 103)

¹ Departemen Agama RI, *Al-Quran dan Terjemahnya* (Surabaya: Pustaka Al-Kautsar, 2009), 89.

PERSEMBAHAN

Skripsi ini penulis persembahkan untuk kedua orang tua penulis. Bapak A. Sudja'i H.S dan Ibu Aliyah yang selalu mendoakan, membimbing dan mencurahkan kasih sayangnya kepada penulis. Semoga Allah mengasihi beliau berdua, aamiin.

Kepada seluruh guru penulis, terkhusus Ustadz Acep Saefuddin, M.Ed, Ustdadz Moch. Ridwan dan Ustadz Jajang Munjali, S.Pd.I Terima kasih atas keikhlasannya membimbing penulis agar menjadi insan yang lebih baik. Semoga ilmu-ilmu yang diberikan senantiasa berkah dan menjadi amal jariyah yang pahalanya selalu mengalir.

Keluarga besar Pesantren Life Skill Daarun Najaah Semarang terkhusus Dr. KH. Ahmad Izzuddin, M.Ag yang tidak pernah bosan untuk memotivasi penulis untuk menjadi manusia yang amanah dan bertanggung jawab.

Kementrian Agama Republik Indonesia, yang berkat beasiswa (PBSB) penulis dapat melanjutkan pendidikan ke jenjang Strata I (S1). Semoga amanah yang diberikan dapat penulis jalankan dengan sebaik-baiknya.

Kepada seluruh sahabat-sahabat penulis, terutama sahabat-sahabat Penerima PBSB Angkatan 2015 yang sejak pertama kuliah sampai saat ini selalu bersama-sama. Terima kasih atas semua bantuan, dukungan, bimbingan, dedikasi, doa serta semua pengalaman yang luar biasa selama ini, semua itu akan menjadi kenangan dan masa yang sangat indah bagi penulis.

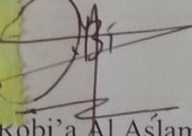
Kepada seluruh orang yang saya cintai dan yang mencintai saya, semoga cinta kita yang semu dapat menghantarkan pada cinta-Nya yang abadi.

DEKLARASI

Dengan Penuh kejujuran dan tanggung jawab, penulis menyatakan bahwa skripsi ini tidak berisi materi yang pernah ditulis oleh orang lain atau diterbitkan. Demikian juga skripsi ini tidak berisi satupun pemikiran orang lain, kecuali informasi yang terdapat dalam referensi yang dijadikan bahan rujukan.

Semarang, 07 Juli 2019
Deklerator,




Uti Kobi'a Al Aslami
NIM : 1502046101

PEDOMAN TRANSLITERASI¹

A. Konsonan

ء = ‘	ز = z	ق = q
ب = b	س = s	ك = k
ت = t	ش = sy	ل = l
ث = ts	ص = sh	م = m
ج = j	ض = dl	ن = n
ح = h	ط = th	و = w
خ = kh	ظ = zh	ه = h
د = d	ع = ‘	ي = y
ذ = dz	غ = gh	
ر = r	ف = f	

B. Vokal

اَ-	a
اِ-	i
اُ-	u

¹ Tim Fakultas Syari'ah IAIN Walisongo Semarang, *Pedoman Penulisan Skripsi*, (Semarang : Basscom Multimedia Grafika, 2012), hlm. 61

C. Diftong

اي	ay
او	aw

D. Syaddah (ّ-)

Syaddah dilambangkan dengan konsonan ganda, misalnya الطّبّ *at-thibb*.

E. Kata Sandang (... ال)

Kata Sandang (... ال) ditulis dengan *al-...* misalnya الصنّاعه = *al-shina'ah*.

Al- ditulis dengan huruf kecil kecuali jika terletak pada permulaan kalimat.

F. Ta' Marbutah (ة)

Setiap *ta' marbutah* ditulis dengan "h" misalnya المعيشه الطبيعیه = *al-ma'isyah al-thabi'iyah*.

ABSTRAK

Aplikasi waktu salat berbasis ponsel pintar Android merupakan salahsatu aplikasi yang berkembang saat ini. Namun perlu diperhatikan perancangan program-program waktu salat berbasis Android yang sudah banyak beredar saat ini perlu dikaji ulang mengenai pengambilan sumber data dan keakuratan hasilnya. Karena dalam perhitungan waktu salat terdapat hal-hal yang perlu diperhatikan yang erat kaitannya dengan penghitungan waktu salat itu sendiri. Seperti, lintang tempat, bujur tempat, tinggi tempat, dan zona waktu yang terkadang justru diabaikan. Selain proses penghitungan waktu salat, faktor lain yang juga tidak kalah penting adalah pentingnya standarisasi waktu. Karena terkadang jadwal yang digunakan sama namun jam yang menjadi standar berbeda. Pada kenyataanya jam yang digunakan di masjid-masjid adalah jam berjenis analog yang memiliki jeda sepersekian detik dalam setiap geraknya yang apabila diakumulasikan maka akan terdapat selisih yang cukup signifikan. Hal lain yang juga menjadi permasalahan adalah mengenai data koordinat tempat yang banyak sekali jenisnya. Padahal untuk waktu salat diperlukan data koordinat yang dapat mencakup luas satu kota/kabupaten sehingga dalam kota/kabupaten itu bisa diakomodir dengan satu jadwal waktu salat.

Berdasarkan permasalahan di atas penulis berinisiatif untuk meyusun sebuah aplikasi waktu salat dengan metode *ephemeris* dengan algoritma *Jean Meus* menggunakan standar jam atom BMKG. Aplikasi tersebut dirancang dalam bahasa pemrograman *java* dan diberi nama *Accurate Atomic Time for Salat*. Adapun rumusan masalah pada penelitian penulis ada dua, yakni bagaimana proses perancangan aplikasi android jadwal waktu salat dengan standar jam atom BMKG? bagaimana hasil uji fungsional aplikasi android jadwal waktu salat dengan standar jam atom BMKG ?

Dalam penyusunan penelitian ini, penulis melalui tiga tahap penyusunan, yakni: tahap pengumpulan data, tahap perancangan desain aplikasi dan tahap implementasi rancangan aplikasi kedalam bahasa pemrograman. Proses perhitungan aplikasi disusun menggunakan algoritma *Jean Meus*. Adapun desain antarmuka dibuat dengan *Corel Drax X7* yang kemudian diexport ke *Android Studio*.

Dari hasil uji coba, aplikasi ini dapat digunakan pada berbagai macam ponsel pintar Android dengan tipe, merek dan versi Android yang berebeda. Selanjutnya, berdasarkan uji verifikasi hasil perhitungan waktu salat pada aplikasi ini yang dikomparasikan dengan *website* waktu salat milik Bimas Islam Kementrian Agama RI, diketahui bahwa terdapat selisih pada beberapa waktu salat dengan nilai selisih satu menit hal ini diperkirakan karena nilai *ikhtiyat* yang berbeda. Oleh karenanya aplikasi ini memilki keakuratan dan layak untuk dijadikan sumber penentuan waktu salat.

Kata kunci : waktu salat, *Accurate Atomic Time for Salat*, jam atom BMKG

KATA PENGANTAR

Alhamdulillah, segala puji bagi Allah yang telah melimpahkan rahmat serta hidayah-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Aplikasi Jadwal Waktu Salat dengan Standar Jam Atom BMKG Berbasis Android”.

Salawat dan salam semoga senantiasa tercurahkan kepada Rasulullah Muhammad SAW, beserta keluarga, sahabat dan pengikut-pengikutnya yang telah menjadi suri tauladan yang baik dalam segala aspek kehidupan. Semoga kelak kita diakui sebagai umat beliau di hari kiamat.

Penulis menyadari bahwa penelitian ini selesai bukan semata-mata atas usaha dari penulis pribadi. Penyusunan penelitian ini tidak lepas dari usaha, bantuan dan doa dari berbagai pihak. Oleh karena itu penulis menyampaikan banyak terima kasih kepada :

1. Dr. H. Ahmad Izzuddin, M.Ag dan Anthin Lathifah, M.Ag., sebagai dosen pembimbing skripsi penulis. Terimakasih atas segala bimbingan dan arahnya mulai dari judul pertama sampai akhir penulisan skripsi.
2. Kedua orang tua penulis, bapak A. Sudja'i H.S dan ibu Aliyah atas segala doa dan perhatian yang selama ini mengalir kepada penulis, juga atas segala dukungan moril maupun materiil kepada penulis selama menempuh kuliah di UIN Walisongo Semarang.
3. Keluarga besar Pesantren Life Skill Daarun Najaah. Khususnya bapak Dr. KH. Ahmad Izzuddin, M.Ag. serta Ibu Nyai Hj. Aisah Andayani, M.Ag yang selalu

memberikan bimbingan dan motivasi kepada penulis untuk menjadi orang yang baik, amanah dan bertanggung jawab.

4. Keluarga besar Pesantren Persatuan Islam 92 Majalengka, terkhusus Ustadz Acep Saefuddin, M.Ed, Ustdadz Moch. Ridwan dan Ustadz Jajang Munjali, S.Pd.I yang doa serta ridho ilmu nya selalu penulis harapkan. Tak lupa kepada seluruh guru-guru penulis selama di pesantren, terima kasih atas segala bimbingan dan doa yang diberikan.
5. Bapak Dr. Suaidi, S.T, MT selaku narasumber dalam penelitian ini. Terima kasih atas ilmu serta arahannya selama penelitian skripsi ini. Semoga kebaikan bapak dicatat sebagai amal jariyah yang pahalanya selalu mengalir.
6. Kementrian Agama RI yang telah memberikan beasiswa PBSB kepada penulis selama menempuh pendidikan S1 di UIN Walisongo Semarang.
7. Ketua Jurusan Ilmu Falak sekaligus Ketua Pengelola PBSB UIN Walisongo beserta stafnya, terima kasih atas dukungan, bimbingan dan fasilitas yang diberikan kepada penulis.
8. Bapak Dr. Agus Nurhadi, M.A., selaku dosen wali penulis, yang selalu memberikan bimbingan, arahan, serta ilmunya kepada penulis.
9. Mas Ishom, Mas Riza, Mas Farid yang telah membimbing dalam penulisan skripsi ini.
10. Keluarga besar CSSMoRA UIN Walisongo dari seluruh angkatan, terima kasih telah menjadi wadah penulis dalam berorganisasi dan memberikan pengalaman yang akan selalu terkenang dalam hidup penulis.

11. Keluarga besar pengurus CSSMoRA Nasional “Kabinet Berkhidmat”, atas segala pengalaman berharganya selama satu tahun kepengurusan.
12. Keluarga pertamaku di Semarang, PBSB Angkatan 2015 : Shofa, Arif, Thoifur, Saldy, Masyfuk, Hajir, Firli, Iqbal, Fandi, Jamal, Shofi, Falih, Halimi, Cahyo, Ninik, Labib, Winda, Isma, Yuli, Raizza, Muslimah, Ana, Ilma, Dela, Indri, Rida, Mis, Amalia dan Nunuk. Terimakasih untuk semua kebersamaan kalian selama 4 tahun ini, semoga kita semua sukses dunia dan akhirat. .
13. Semua orang yang mencintai penulis dan yang penulis cintai. Semoga cinta kita yang semu dapat mengantarkan pada cinta-Nya yang abadi.

Atas semua kebaikannya, penulis hanya mampu mengucapkan terima kasih dan berdo'a semoga Allah SWT membalas semuanya. Pada akhirnya penulis mengharapkan saran dan kritik dari pembaca demi kesempurnaan skripsi ini agar kedepannya lebih baik lagi dan tentunya penulis berharap semoga skripsi ini membawa manfaat bagi penulis khususnya dan bagi pembaca umumnya. Amin.

Semarang, 07 Juli 2019
Penulis

Obi Robi'a Al Aslami

DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PERSETUJUAN PEMBIMBING.....	ii
HALAMAN PENGESAHAN.....	iv
HALAMAN MOTTO.....	v
HALAMAN PERSEMBAHAN.....	vi
HALAMAN DEKLARASI.....	vii
HALAMAN PEDOMAN TRANSLITERASI.....	viii
HALAMAN ABSTRAK.....	x
HALAMAN KATA PENGANTAR.....	xi
HALAMAN DAFTAR ISI.....	xiv
HALAMAN DAFTAR TABEL.....	xvii
HALAMAN DAFTAR GAMBAR.....	xviii

BAB I PENDAHULUAN

A. Latar Belakang Masalah.....	1
B. Rumusan Masalah.....	6
C. Tujuan Dan Manfaat Penelitian.....	6
D. Telaah Pustaka.....	6
E. Metode Penelitian.....	8
F. Sistematika Penulisan.....	11

BAB II KAIDAH HISAB AWAL WAKTU SALAT DAN PERKEMBANGAN ANDROID

A. Kaidah Hisab Awal Waktu Salat	
1. Definisi Salat dan Waktu Salat.....	13
2. Landasan Hukum Waktu Salat	15
3. Batasan Salat Maktubah.....	24
4. Komponen Untuk Hisab Awal Waktu Salat	32
B. Pola Kerja Program Android	
1. Perkembangan Android.....	39
2. Pola Kerja Android	43

BAB III DESAIN RANCANGAN DAN IMPLEMENTASI PROGRAM *ACCURATE ATOMIC TIME FOR SALAT*

A. Pola Kerja Jam Atom BMKG	47
B. Algoritma Hisab Waktu Salat	50
C. Aplikasi <i>Accurate Atomic Time for Salat</i>	61
D. Implementasi Aplikasi <i>Accurate Atomic Time for Salat</i>	66

BAB IV UJI COBA DAN EVALUASI APLIKASI *ACCURATE ATOMIC TIME FOR SALAT*

A. Uji Fungsional Aplikasi <i>Accurate Atomic Time for Salat</i>	71
B. Uji Verifikasi Hasil Perhitungan Aplikasi <i>Accurate Atomic Time for Salat</i>	80

BAB V :PENUTUP

A. Kesimpulan	85
B. Saran	86
C. Penutup	87

DAFTAR PUSTAKA

LAMPIRAN-LAMPIRAN

DAFTAR RIWAYAT HIDUP

DAFTAR TABEL

Tabel 4.1 Perbandingan Waktu Salat Kota Semarang 25 Juni 2019	81
Tabel 4.2 Perbandingan Waktu Salat Kota Semarang 03 Juli 2019	82
Tabel 4.3 Perbandingan Waktu Salat Kota Semarang 03 Juli 2019 dengan Berbagai Varian Tinggi Tempat menggunakan Aplikasi AAT for Salat	82
Tabel 4.4 Perbandingan Waktu Salat Kota Bandung 04 Juli 2019 dengan Berbagai Varian Tinggi Tempat menggunakan Aplikasi AAT for Salat	83

DAFTAR GAMBAR

Gambar 2.1 Arsitektur Android	43
Gambar 3.1 Jam Atom Cesium 4310 milik BMKG	49
Gambar 3.2 Skema Sistem Tanda Waktu BMKG	50
Gambar 3.3 Diagram Alur Aplikasi.....	61
Gambar 3.4 Tampilan Utama Menu Aplikasi	62
Gambar 3.5 Tampilan Hasil Perhitungan Waktu Salat	63
Gambar 3.6 Desain Antarmuka pada Tampilan Awal dan Halaman Input.....	69
Gambar 3.7 Desain Antarmuka pada Tampilan Utama Waktu Salat	70
Gambar 4.1 Tampilan <i>Loading</i> Android Studio	72
Gambar 4.2 Tampilan Lembar Kerja Android Studio	72
Gambar 4.3 Tampilan Setting Xiaomi Redmi 4X	73
Gambar 4.4 Proses Konfigurasi	74
Gambar 4.5 Proses Debugging	74
Gambar 4.6 Tampilan Utama Aplikasi	75
Gambar 4.7 Tampilan Cari Kota.....	75
Gambar 4.8 Tampilan Gunakan GPS.....	76
Gambar 4.9 Halaman Hasil Hitung Waktu Salat.....	77
Gambar 4.10 Tampilan Pengaturan Alarm.....	77
Gambar 4.11 Notifikasi Saat Memasuki Awal Waktu Salat	78
Gambar 4.12 Tampilan Server NTP jam atom BMKG.....	80

BAB I

PENDAHULUAN

A. Latar Belakang Masalah

Seiring dengan perkembangan teknologi setiap hari ada berbagai inovasi yang muncul di berbagai sektor tak terkecuali dalam kehidupan seorang muslim. Setiap harinya ada banyak teknologi yang digunakan termasuk didalamnya adalah ponsel pintar dengan berbagai aplikasi yang mampu memudahkan setiap penggunanya. Setiap pengembang berlomba-lomba untuk menyajikan aplikasi –aplikasi praktis yang dapat digunakan dimanapun dan kapanpun. Salah satu diantaranya adalah aplikasi waktu salat.

Aplikasi waktu salat sudah berkembang pesat berbasis desktop, web, maupun mobile. Aplikasi berbasis desktop dan web masih menyulitkan penggunanya karena dibutuhkan perangkat yang tidak praktis. Aplikasi berbasis mobile hadir sebagai solusi atas masalah ini. Terdapat banyak macam platform mobile saat ini namun salah satu platform yang memiliki banyak peminat adalah Android. Tercatat sekitar 85,1% ponsel yang beredar menggunakan platform ini.¹

Android merupakan sebuah sistem operasi untuk berbagai perangkat *mobile* seperti *handphone*, *netbook*, dan komputer tablet. Pada awalnya, android dikembangkan oleh perusahaan *Android*

¹ Nadia Firly, *Create Your Own Android Application* (Jakarta: Elex Medai Komputindo, 2018), 9.

Inc. namun kemudian perusahaan tersebut diakuisisi oleh *Google* sehingga menjadi produk *Google*. Sekarang ini pengembangan Android ditentukan oleh sebuah konsorsium bernama *Open Handset Alliance* (OHA)² yang terdiri atas berbagai vendor perangkat *mobile*, komputer, dan telekomunikasi seperti *Intel*, *Nvidia*, *Google*, *Samsung*, *Sprint*, *TMobile*, *Motorola*, *LG*, *Sony Ericsson*, *Toshiba*, *Vodafone*, serta masih banyak yang lain dan anggotanya terus bertambah³.

Platform ini dipilih karena terbukti penggunaannya semakin hari semakin meningkat dikutip dari situs resmi www.android.com, setidaknya ada satu juta perangkat Android yang diaktifkan setiap harinya. Dengan demikian, sangat terbuka peluang bagi para programmer untuk ikut andil dalam mengembangkan aplikasi Android.⁴

Aplikasi waktu salat berbasis Android yang beredar saat ini masih memiliki beberapa kekurangan. Pertama adalah input data yang memanfaatkan Global Positioning System (GPS) milik Android. Hal ini tentu akan mempermudah pengguna untuk mendapatkan waktu salat, namun untuk mendapatkan data diperlukan koneksi internet. Hal ini akan menimbulkan permasalahan bagi pengguna yang berada di tempat yang

²*Open Handset Alliance* (OHA) yaitu aliansi perangkat selular yang terdiri dari 47 perusahaan *hardware*, *software* dan perusahaan telekomunikasi ditujukan untuk mengembangkan standar terbuka bagi perangkat selular.

³ http://droid-indonesia.blogspot.co.id/2012/01/sejarah-android_11.html diakses 15 Januari 2019

⁴ Arif Akbarul Huda, *Live Coding! 9 Aplikasi Android Buatan Sendiri* (Yogyakarta: Andi Offset, 2013), 2.

sulit mendapatkan akses internet. Kekurangan yang kedua adalah minimnya aplikasi yang menggunakan data daerah di Indonesia yang tersimpan di database. Jika daerah tidak ditemukan, maka aplikasi tidak bisa digunakan sebagaimana mestinya. Padahal hal ini sangat berguna bagi umat Islam yang sulit mendapatkan akses internet.

Selain itu, masalah yang muncul adalah mengenai banyaknya ragam model perhitungan waktu salat dan sumber data. Banyak diantara pengembang yang tidak secara langsung menghitung waktu salat dengan algoritma tertentu. Namun mereka hanya mengambil hasil perhitungan dari sumber lain. Dalam hal ini setidaknya penulis menemukan dua aplikasi yang tidak menghitung secara langsung waktu salatnya. Berdasarkan wawancara kepada tim IT aplikasi Yaumeer, pengembang ini tidak melakukan perhitungan waktu salat dengan algoritma tertentu. Mereka hanya mengambil data dari PrayTimes.org.⁵ Aplikasi lain yang juga tidak secara langsung menghitung waktu salatnya adalah aplikasi Kesan (Kedaulatan Santri). Berdasarkan wawancara dengan tim IT Kesan mereka menyebutkan bahwa waktu salat yang ada pada aplikasinya adalah waktu salat yang bersumber dari Kementerian Agama RI.⁶

Masalah lain yang juga muncul adalah mengenai keakuratan data koordinat yang ada. Lembaga manakah yang layak menjadi standar acuan. Dalam penelitian ini penulis menggunakan koordinat tengah kabupaten

⁵ Hasil wawancara dengan Abid (Tim IT Yaumeer) pada Jumat, 21 Juni 2019 pukul 10:47 WIB.

⁶ Hasil wawancara dengan Fadhli (Tim IT Kesan) pada Sabtu, 8 Juni 2019 pukul 16:09 WIB.

kota yang telah dirilis oleh BIG (Badan Informasi Geospasial)⁷ hal ini agar waktu salat dapat digunakan untuk kabupaten kota secara menyeluruh. Karena jadwal waktu salat yang diperhitungkan menggunakan selain titik koordinat tengah belum tentu dapat diberlakukan untuk satu wilayah kabupaten atau kota, apalagi selisih koordinatnya diatas $0,5^\circ$ dan posisinya berada di sebelah selatan dan timur dari titik koordinat tengah.⁸

Standar jam juga memiliki peranan penting dalam penentuan waktu salat. Jam yang saat ini banyak digunakan di masjid-masjid adalah jam analog dengan tenaga baterai. Mekanisme jam jenis ini memanfaatkan daya arus listrik dari baterai yang menghasilkan getaran untuk menggerakkan jarum jamnya. Jam jenis ini mempunyai kelemahan karena disetiap detiknya ada jeda dan menyebabkan keterlambatan sepersekian detik.⁹ Jika keterlambatan ini diakumulasikan dalam waktu yang lama maka akan terjadi keterlambatan yang cukup signifikan. Oleh karenanya diperlukan sebuah standar jam yang mampu bergerak dengan stabil.

Berdasarkan Undang-Undang (UU) No.31 tahun 2009 tentang Meteorologi, Klimatologi, dan Geofisika (MKG)¹⁰, suatu badan yang bernama Badan Meteorologi, Klimatologi, dan Geofisika (BMKG) adalah

⁷ Badan Informasi Geospasial (BIG) sebelumnya dikenal dengan nama Badan Koordinasi Survei dan Pemetaan Nasional atau Bakosurtanal. Badan ini pertama kali didirikan berdasarkan Keputusan Presiden Nomor 63 Tahun 1969, tepatnya pada 17 Oktober 1969. Status hukumnya kemudian diperkuat dengan keluarnya Undang-Undang Nomor 4 Tahun 2011 tentang Informasi Geospasial pada 27 Desember 2011 dan dijabarkan dalam Peraturan Presiden Nomor 94 Tahun 2011. Keluarnya undang-undang inilah menjadi salah satu dasar perubahan nama menjadi BIG. Lihat <http://www.big.go.id/sejarah/>

⁸ <http://journal.walisongo.ac.id/index.php/ahkam/article/view/1981> diakses 15 Januari 2019

⁹ <https://www.tifannywatch.com/mengenal-dua-tipe-dari-mesin-penggerak-jam-tangan-analog/> diakses 04 Juli 2019

¹⁰ http://www.dpr.go.id/dokjdi/document/uu/UU_2009_31.pdf diakses 15 Januari 2019

lembaga yang ditugaskan oleh negara sebagai penjaga waktu di Indonesia. Artinya BMKG memiliki wewenang juga kewajiban untuk menyediakan standar waktu.

Berdasarkan sumbernya, sistem tanda waktu dibagi menjadi dua, yaitu sistem waktu atomis dan sistem waktu astronomis. Selanjutnya kita akan bahas sumber waktu atomis. Sumber waktu atomis adalah dari transisi energi elektron pada atom yang digunakan pada jam atom. Mengingat transisi ini terjadi secara stabil, maka jam atom dapat dikatakan sebagai penanda waktu yang standar, sehingga informasi yang dihasilkannya adalah informasi waktu standar. Dengan menggunakan jam atom, pada saat ini kita dapat menjawab waktu dengan tepat.¹¹ Dalam menyusun penelitian ini nantinya akan digunakan standar jam atom BMKG.

Menurut Rinto (2012) diperlukan berbagai data dalam penentuan awal waktu salat, yaitu posisi lintang tempat, bujur tempat, *julian day*, *deklinasi* Matahari, dan *equation of time*. Dengan menggunakan algoritma Jean Meus diharapkan jadwal salat yang peneliti bangun dapat menyediakan data lebih banyak dan akurasi waktu yang mengacu pada jadwal salat milik pemerintah.

Berangkat dari paparan diatas, peneliti bermaksud untuk mengangkat judul APLIKASI JADWAL WAKTU SALAT DENGAN STANDAR JAM ATOM BMKG BERBASIS ANDROID dengan *output*

¹¹ http://webdata.bmkg.go.id/web/standar_waktu_indonesia.pdf diakses 15 Januari 2019

sebuah aplikasi *mobile* waktu salat yang diberi nama *Accurate Atomic Time for Salat*.

B. Rumusan Masalah

1. Bagaimana proses rancangan aplikasi android jadwal waktu salat dengan standar jam atom BMKG ?
2. Bagaimana hasil uji fungsional aplikasi android jadwal waktu salat dengan standar jam atom BMKG ?

C. Tujuan dan Manfaat Penelitian

1. Merancang aplikasi android jadwal waktu salat dengan standar jam atom BMKG
2. Mengetahui tingkat akurasi aplikasi android jadwal waktu salat dengan standar jam atom BMKG

D. Telaah Pustaka

Sejauh penelusuran penulis secara garis besar dalam keilmuan falak memang sudah banyak para pengembang aplikasi android yang mengembangkan aplikasi penentu awal waktu salat. Hanya saja ketidakjelasan metode dan sumber data yang digunakan membuat penulis merasa perlu mengembangkan aplikasi android yang didasarkan pada data standar yang digunakan pemerintah. Adapun beberapa penelitian yang relevan dengan penelitian yang akan penulis lakukan antara lain :

Skripsi, Farid Abdillah Hasan, 2013, berjudul “Rancang Bangun Aplikasi Jadwal Shalat Metode Ephemeris Berbasis Android”. Dalam skripsinya Farid Abdillah menghasilkan sebuah aplikasi android dengan metode Ephemeris dengan input data berupa database kabupaten kota yang berasal dari Direktorat Jendral Badan Peradilan Agama Indonesia.¹² Perbedaan yang penulis teliti dengan skripsi ini adalah penulis menggunakan data koordinat yang dikeluarkan oleh BIG (Badan Informasi Geospasial) dan standar waktu atom BMKG.

Skripsi, Aditya Nurahman, 2014, berjudul “Aplikasi Mobile Penentuan Arah Kiblat Berdasarkan Global Positioning System (GPS) Beserta Pengingat Waktu Shalat dan Kumpulan Artikel Berbasis Android Hybrid”. Dalam skripsi ini perancang menggunakan data koordinat secara online melalui GPS sehingga ponsel harus terkoneksi internet sebelum menggunakannya.¹³

Skripsi, Bangkit Riyanto, 2016, “Studi Analisis Algoritma Waktu Sholat dalam Aplikasi Android Digital Falak Karya Ahmad Tholhah Ma’ruf”.¹⁴ Dalam skripsinya Bangkit Riyanto meneliti algoritma yang digunakan oleh Ahmad Tholhah Ma’ruf dalam Aplikasi Digital Falak.

¹² Farid Abdillah Hasan, “Rancang Bangun Aplikasi Jadwal Shalat Metode Ephemeris Berbasis Android”, Skripsi Fakultas Sains dan Teknologi UIN Maulana Malik Ibrahim, 2013.

¹³ Aditya Nurahman, “Aplikasi Mobile Penentuan Arah Kiblat Berdasarkan Global Positioning System (GPS) Beserta Pengingat Waktu Shalat dan Kumpulan Artikel Berbasis Android Hybrid”, Skripsi Fakultas Ilmu Komputer Universitas Gunadarma, 2014.

¹⁴ Bangkit Riyanto, “Studi Analisis Algoritma Waktu Sholat dalam Aplikasi Android Digital Falak Karya Ahmad Tholhah Ma’ruf”, Skripsi Fakultas Syariah UIN Walisongo, 2016.

Karya Ilmiah, Riki Anto, 2015, “Aplikasi Pengingat Waktu Shalat (Adzan) Menggunakan Eclips”.¹⁵ Dalam karya ilmiah ini penulis menggunakan bantuan Android Studio untuk pembuatan aplikasi android data yang digunakan berbasis database offline yang telah disiapkan developer.

Dalam telaah pustaka tersebut, penulis belum menemukan tulisan yang membahas secara spesifik mengenai aplikasi android jadwal waktu salat metode *ephemeris* dengan standar jam atom BMKG sesuai apa yang ingin diteliti oleh penulis.

E. Metode Penelitian

1. Jenis Penelitian

Penelitian ini termasuk dalam jenis penelitian kualitatif numerik yang bersifat deskriptif (*descriptive research*), yang dalam hal ini lebih ditekankan pada metode pemrograman.

2. Sumber Data

a. Data Primer

Sumber primer yang penulis gunakan adalah Konsep aplikasi *Accurate Atomic Time for Salat yang terambil dari data* buku *Astronomical Algorithms* sebagai rujukan dasar perhitungan waktu salat dengan metode *ephemeris* dan buku

¹⁵ Riki Anto, , “Aplikasi Pengingat Waktu Shalat (Adzan) Menggunakan Eclips”, Karya Ilmiah Fakultas Ilmu Komputer Universitas Gunadarma, 2015.

pemrograman Android sebagai rujukan dasar pemrogramannya.

b. Data Sekunder

Sumber sekunder yang penulis gunakan yakni berupa buku-buku, makalah-makalah hingga tulisan-tulisan yang berkaitan dengan keilmuan falak dan Android programming baik yang berupa dokumen maupun *e-book* penulis juga menggunakan sumber *sekunder* berupa web seperti www.big.go.id dan www.bmkg.go.id.

3. Metode Pengumpulan Data

Adapun metode pengumpulan data yang penulis gunakan adalah :

a. Metode Dokumentasi

Metode ini digunakan untuk memperoleh data-data yang dibutuhkan dalam penelitian, penulis mendapatkan data-data dengan menelusuri literatur-literatur yang berkaitan dengan penelitian penulis. Penulis memulai dengan menelusuri literatur-literatur yang berkaitan dengan pembahasan mengenai sejarah jam atom serta perkembangannya. Kemudian berlanjut kepada penelusuran mengenai data-data yang diperlukan beserta proses perhitungannya, dan yang terakhir adalah penelusuran mengenai tata cara pemrograman Android.

b. Metode Wawancara

wawancara merupakan pertemuan dua orang untuk bertukar informasi melalui tanya jawab, sehingga dapat dikonstruksikan makna dalam suatu topik tertentu.¹⁶ Dalam hal ini penulis melakukan wawancara berkaitan dengan jam atom kepada Dr. Suaidi Ahadi, M.T sebagai kepala sub bidang tanda waktu BMKG.

4. Metode Analisis Data

Dalam penelitian ini penulis menggunakan metode analisis deskriptif¹⁷ dan komparatif. Deskriptif yakni sebuah metode pemecahan masalah dengan mengumpulkan data dan melukiskan objek penelitian. Dalam hal ini penulis menggambarkan secara umum tentang aplikasi waktu salat berbasis android dengan standar jam atom, guna untuk mengetahui rancangan program, serta kelemahan dan kelebihan aplikasi tersebut.

Selanjutnya penulis menggunakan metode analisis komparatif untuk menarik kesimpulan berupa tingkat akurasi hisab awal waktu salat. Adapun untuk mengetahui informasi

¹⁶ Sugiyono, *Metode Penelitian Pendidikan Oendekatan Kuantitatif, Kualitatif, dan R&D* (Bandung: Alfabeta, 2007), 317.

¹⁷ Noeng Muhadjir, *Metodologi Penelitian Kualitatif*, (Yogyakarta: Rake Sarasin, 1996), 187.

nilai akurasi yang dimaksud penulis membandingkan waktu salat yang dihasilkan aplikasi ini dengan website waktu salat milik Direktorat Bimbingan Masyarakat Islam Kementerian Agama RI. *Website* ini dipilih karena Kementerian Agama merupakan lembaga resmi negara yang secara resmi memiliki wewenang untuk menyediakan layanan waktu salat.

F. Sistematika Penulisan

Dalam penyusunan skripsi ini, penulis akan membagi menjadi lima bab dengan rincian sebagai berikut :

Bab pertama berisi pendahuluan. Pada bab ini akan dikemukakan mengenai permasalahan yang melatar belakangi penelitian tentang aplikasi android jadwal waktu salat metode *ephemeris* dengan standar jam atom BMKG, kemudian di lanjutkan dengan tujuan penelitian sebagai arah penelitian, kajian pustaka dan penelitian terdahulu, metode penelitian sebagai cara sasaran penelitian dan sistematika penulisan.

Bab kedua berisi landasan teori yang akan memaparkan beberapa teori yang mendasari dalam penyusunan tugas akhir ini. Dasar teori yang berkaitan dengan pembahasan tentang aplikasi android jadwal waktu salat dengan standar jam atom BMKG.

Bab ketiga berisi perancangan dan implementasi program aplikasi android jadwal waktu salat dengan standar jam atom BMKG. Pembahasan

dalam bab ini meliputi alur algoritma perhitungan waktu salat metode *ephemeris*, rancangan program, desain dan skema prosedurnya, serta implementasi dari rancangan program tersebut.

Bab keempat berisi uji coba dan evaluasi, bab ini berisi beberapa tahap pengujian sebagaimana telah ditetapkan sebelumnya yang dilakukan untuk mengetahui kelayakan aplikasi android jadwal waktu salat dengan standar jam atom BMKG. Selanjutnya pada bab ini juga disertakan evaluasi hal-hal penting yang diketahui setelah pelaksanaan berbagai macam pengujian terhadap program tersebut.

Bab kelima berisi penjelasan mengenai kesimpulan, saran/rekomendasi terkait dengan hasil penelitian penulis, berupa program android jadwal waktu salat dengan standar jam atom BMKG berikut algoritma pemrogramannya dan penutup.

BAB II

KAIDAH HISAB AWAL WAKTU SALAT DAN POLA KERJA PROGRAM ANDROID

A. Kaidah Hisab Awal Waktu Salat

1. Definisi Salat dan Waktu Salat

Salat adalah kata dalam bahasa Indonesia yang merupakan kata serapan dari bahasa Arab¹. Dalam kamus besar bahasa Indonesia salat diartikan rukun Islam kedua, berupa ibadah kepada Allah SWT, doa kepada Allah SWT². Sedangkan dalam bahasa Arab salat berasal dari kata *shalla yushalli shalatan* yang berarti doa³ sedangkan menurut syara adalah rangkaian kata dan perbuatan yang ditentukan, dimulai dengan membaca takbir dan diakhiri dengan salam⁴.

Salat merupakan salah satu dari lima rukun Islam yang dilaksanakan dalam waktu-waktu yang ditentukan sehingga salat dinamakan ibadah *muwaqat*. Perintah salat sendiri diterima langsung oleh Rasulullah SAW dari Allah SWT dalam peristiwa *isra' mi'raj*, yaitu peristiwa dijalankannya Nabi Muhammad SAW dari *Masjid al Haram* ke *Masjid al Aqsha* kemudian dinaikkan ke *sidratul muntaha*. Peristiwa ini terjadi

¹ Kata salat ini tergolong kata serapan dari bahasa Arab yang masih sesuai dengan aslinya, baik lafal maupun artinya. Lihat Pusat Pengembangan Bahasa, *Bahasa Indonesia Bahasa Bangsa*, Semarang: IAIN Walisongo, 2014, hal. 40

² Tim Penyusun Kamus Bahasa Indonesia, *Kamus Bahasa Indonesia*, (Jakarta: Pusat Bahasa, 2008), 1249.

³ Ahmad Warson Munawwir, *Kamus Al-Munawwir Arab-Indonesia Terlengkap*, (Surabaya: Pustaka Progresif, 1997), 792.

⁴ Ahmad Zainuddin bin Abdul Aziz, *Fathul Muin*, (Cyprus: Daar Ibn Hazm, 2004), 36.

pada bulan Rajab tahun ke-11 kenabian. Dalam peristiwa itu Nabi Muhammad SAW ditemani oleh Jibril bertemu dengan nabi-nabi terdahulu.⁵

Salat memiliki keterkaitan dengan gerak semu harian Matahari yang mengakibatkan perbedaan waktu salat. Hal ini sebagaimana telah disebutkan dalam Quran Surat an-Nisa ayat 103

إِنَّ الصَّلَاةَ كَانَتْ عَلَى الْمُؤْمِنِينَ كِتَابًا مَّوْقُوتًا

“*Sesungguhnya salat itu adalah fardhu yang di tentukan waktunya atas orang orang yang beriman*”.⁶ (An-Nisa’ : 103)

Menurut Ibnu Abbas kata مَوْقُوتًا dalam firman Allah surat an-Nisa ayat 103 diatas bermakna diwajibkan. Ibnu Abbas R.A berkata, “sesungguhnya salat itu memiliki waktu seperti waktu haji”, begitu pula halnya dengan apa yang diriwayatkan dari Mujahid, Salim bin Abdillah, dan selain mereka berdua. Zaid bin Aslam *Rahimahullah* berkata berkenaan dengan firman Allah SWT مَوْقُوتًا yaitu teratur. Maksudnya adalah setiap kali lewat satu waktu, maka waktu yang lainpun akan datang.⁷

⁵ Dalam peristiwa Isra Mi’raj nabi Muhammad bertemu dengan nabi Adam AS, nabi Yahya AS, nabi Isa AS, nabi Yusuf AS, nabi Idris AS, nabi Harun AS, nabi Musa AS dan nabi Ibrahim AS. Lihat hadis Bukhori nomor 349, Muhammad bin Ismail al-Bukhari, *Shahih Bukhari*, juz 1, Beirut: Beirut: Daar al-Kutub al-Ilmiyah, 1992, hal. 115-116

⁶ Departemen Agama RI, *Al –Quran dan Terjemahnya* (Surabaya: Pustaka Al-Kautsar, 2009), 89.

⁷ Ismail bin Umar bin Katsir Al-Quraisy Ad-Dimasyqi, *Tafsir al-Quranul ‘Azhim*, (Beirut Libanon, 2000), 528.

Makna waktu salat (al-waqt) adalah batasan sesuatu. Baik dari segi masa ataupun esensi. Adanya waktu salat ini memudahkan umat Islam untuk menjalankan ibadah salat sesuai dengan waktu-waktu yang telah ditentukan.⁸

2. Landasan Hukum Waktu Salat

Landasan hukum waktu salat terdapat dalam dua sumber utama umat Islam, yaitu Al-Quran dan Hadits nabi Muhammad SAW.

a. Landasan Hukum Dari Al- Quran

1). Al – Quran surah an- Nisa’ ayat 103.

فَإِذَا قَضَيْتُمُ الصَّلَاةَ فَادْكُرُوا اللَّهَ فِيمَا وُقِعْتُمْ وَعَلَىٰ جُنُوبِكُمْ فَإِذَا اطْمَأْنَنْتُمْ فَأَقِيمُوا
الصَّلَاةَ إِنَّ الصَّلَاةَ كَانَتْ عَلَى الْمُؤْمِنِينَ كِتَابًا مَّوْقُوتًا

“Selanjutnya, apabila kamu telah menyelesaikan salat (mu), ingatlah Allah di waktu berdiri, di waktu duduk dan di waktu berbaring, kemudian apabila kamu telah merasa aman, maka dirikanlah salat itu (sebagaimana biasanya). Sesungguhnya salat itu adalah fardhu yang di tentukan waktunya atas orang orang yang beriman”.⁹ (An-Nisa’ : 103)

Para ahli tafsir berbeda pendapat dalam menakwilkan ayat ini. Sebagian berpendapat bahwa maksudnya adalah, “Sesungguhnya salat adalah kewajiban yang telah ditentukan waktunya atas orang-orang mukmin”.¹⁰ Ada yang berpendapat bahwa maksudnya adalah “Sesungguhnya salat adalah ketetapan yang telah diwajibkan atas

⁸ Maulidatun Nur Azizah, “Analisis Hisab Awal Waktu Salat Dalam Kitab Asy-Syahrū” Skripsi Fakultas Syariah UIN Walisongo, 2018, 23.

⁹ Departemen Agama RI, *Al-Quran...*, h. 89.

¹⁰ Abu Ja’far bin Muhammad Jarir Ath Thabari, *Jami Al Bayan an Ta’wil Ayyi Al Quran* Jilid VII, 2003, 449.

orang-orang beriman”. Sedangkan pendapat lain menyebutkan bahwa maksud ayat ini adalah “Sesungguhnya salat adalah kewajiban yang ditetapkan waktunya secara jelas mengenai pelaksanaannya didalam sumbernya (Al-Quran dan hadits).¹¹

Pendapat ini saling berdekatan dari sisi makna, karena apa yang telah difardhukan berarti wajib, dan apa yang diwajibkan pelaksanaannya dari waktu ke waktu berarti telah ditentukan secara bertahap. Hanya saja, mereka yang menakwilkan bahwa salat adalah kewajiban yang waktu pelaksanaannya memiliki tahapan dari waktu ke waktu bersandar pada lafaz مَوْفُوتًا yang diambil dari bentuk *maf'ul* dari ungkapan yang biasa diucapkan Allah telah menentukan waktu kewajiban atas kamu, dan Dia yang menentukannya. Apabila kamu melalaikannya maka Dia menentukan waktu lain untuk melaksanakannya. Demikian pula maksud ayat ini, salat bagi orang mukmin merupakan kewajiban yang telah ditetapkan waktu pelaksanaannya.¹²

2). Al-Quran surah Hud ayat 114

وَأَقِمِ الصَّلَاةَ طَرَفَيْ النَّهَارِ وَزُلْفًا مِّنَ اللَّيْلِ إِنَّ الْحُسْنَى يُذْهِبُ السَّيِّئَاتِ ذَلِكَ ذِكْرَى
لِلذَّكِرِينَ

“Dan laksanakanlah salat pada kedua ujung siang (pagi dan petang) dan pada bagian permulaan malam. Perbuatan-perbuatan

¹¹ *Ibid.*, 451.

¹² *Ibid.*, 452.

baik itu menghapus kesalahan kesalahan. Itulah peringatan bagi orang-orang yang selalu mengingat (Allah)”.¹³ (Hud : 114)

Setelah para ahli tafsir sepakat bahwa yang dimaksud salat pagi adalah salat Subuh, para ahli tafsir berbeda pendapat mengenai makna salat petang sebagian mereka berpendapat yang dimaksud salat petang adalah salat Zuhur dan Asar. Sementara itu Abu Ja'far Muhammad bin Jarir Ath-Thabari dalam tafsirnya berpendapat bahwa yang dimaksud salat petang adalah salat Magrib.

Telah kami katakan bahwa pendapat tersebut merupakan pendapat yang paling benar menurut kesepakatan semua ulama yang menyatakan bahwa salah satu dari kedua tepi petang itu adalah salat Fajar, dan itu adalah salat yang dilaksanakan sebelum terbitnya Matahari. Jika demikian, maka sudah seharusnya semua sepakat bahwa salah satu dari kedua tepi tersebut adalah salat Magrib, karena salat tersebut dilaksanakan sesudah terbenamnya Matahari.¹⁴

Sekiranya wajib, bahwa maksud dari salah satu kedua tepi itu adalah sebelum terbit Matahari, maka sudah seharusnya untuk menjadikan maksud salat yang dilakukan pada salah satu kedua tepi yang lain itu adalah salat yang dilaksanakan setelah Matahari terbenam.

Sementara itu mengani takwil kalimat *وَرُفْعًا مِّنَ اللَّيْلِ* bagian permulaan daripada malam maksudnya adalah bagian-bagian waktu

¹³ Departemen Agama RI, *Al-Quran...*, h. 234.

¹⁴ Abu Ja'far, *Jami...* Jilid XII, 2003, 605.

dari malam hari, yang merupakan bentuk jamak dari lafaz *zulfah* yang memiliki arti saat, kedudukan dan kedekatan. Ath-Thabari berpendapat makna *zulfah* sebagaimana lafaz *gurfah* dijamakan menjadi *guraf* dan *hujrah* menjadi *hujar*. Ath-Thabari memilih pendapat tersebut karena Isya adalah salat yang terakhir, dilaksanakan sesudah melewati bagian dari permulaan malam.¹⁵

3). Al-Quran surah al-Isra ayat 78

أَقِمِ الصَّلَاةَ لِذُلُوكِ الشَّمْسِ إِلَى غَسَقِ اللَّيْلِ وَقُرْآنِ الْفَجْرِ إِنَّ قُرْآنَ الْفَجْرِ كَانَ مَشْهُودًا

“Laksanakanlah salat sejak Matahari tergelincir sampai gelapnya malam dan (laksanakan pula salat) Subuh. Sungguh salat Subuh itu di saksikan (oleh Malaikat)”.¹⁶ (Al-Isra : 78)

Allah berfirman kepada Nabi Muhammad SAW Dirikanlah salat wahai Muhammad, setelah Matahari tergelincir. Para ahli tafsir berbeda pendapat tentang waktu yang dimaksud dengan tergelincirnya matahari. Sebagaimana berpendapat bahwa maksudnya adalah waktu terbenamnya, dan salat yang diperintahkan waktu itu adalah salat Magrib.¹⁷

¹⁵ *Ibid.*, 607.

¹⁶ Departemen Agama RI, *Al-Quran...*, h. 290.

¹⁷ Mereka bersandar pada sebuah riwayat, yaitu : Washil bin Abdul A'la Al Asadi menceritakan padaku, ia berkata : Ibnu Fudhai; menvaritakan kepada kami dari Abu Ishaq, yaitu Asy Syaibani, dari Abdurrahman bin Al Aswad, dari ayahnya, bahwa dia bersama dengan Abdullah bin Mas'ud di atap rumah ketika matahari terbenam, kemudian Ibnu Mas'ud membaca *Aqimishshalata lidulukisy Syamsi ila Gasaqillaili* sampai akhir ayat . Ia lalu berkata, Demi jiwaku yang berada dalam genggam tangan-Nya, inilah waktu matahari tergelincir, waktu nerukanya orang puasa, serta waktu ditegakkannya salat. lihat Abu Ja'far, *Jami...* Jilid XV, 2003, 22

Ulama lainnya berpendapat bahwa tergelincirnya Matahari adalah ketika condong ke arah tergelincirnya (terbenamnya), dan salat yang diperintahkan kepada Rasulullah untuk menegakkannya pada waktu tergelincirnya Matahari adalah salat Zuhur. Menurut Ath Thabari dari dua pendapat tersebut, yang lebih tepat adalah yang mengatakan bahwa maksud firman Allah ini adalah salat Zuhur karena lafaz *duluk* dalam bahasa Arab artinya condong (bukan terbenam) seperti ucapan *dalaka fulan ila kadza* jika dia condong kepadanya.¹⁸

Sementara itu, Wahbah Zuhaili dalam tafsirnya al-Munir menyebutkan makna sesudah Matahari tergelincir adalah tergelincirnya matahari dari titik tengah langit ketika siang hari dan beralihnya matahari dari arah timur ke arah barat. Yaitu hingga datangnya gelap malam. Ini mencakup empat salat, yaitu Zuhur, Asar, Magrib, dan Isya¹⁹. Lebih lanjut beliau menjelaskan *وَقُرْءَانَ الْفَجْرِ* adalah *majaaz mursal* dimana Allah menyebutkan sebagian unsur dari sesuatu sedangkan yang dimaksud ialah keseluruhannya. Dan arti asal *وَقُرْءَانَ الْفَجْرِ* adalah bacaan al-Quran pada salat subuh. Namun, yang dimaksud adalah salat subuh karena bacaan al-Quran adalah sebagian darinya.²⁰

¹⁸ Abu Ja'far, *Jami...* Jilid IV, 27

¹⁹ Wahbah Zuhaili, *At-Tafsirul Munir: Fil 'Aqidah wasy-Syarri'ah wal Manhaj* (Dimsyq: Daarul Fikr, 2003) Jilid VIII, 151.

²⁰ *Ibid.*

4). Al-Quran surah Thaha ayat 130

فَاصْبِرْ عَلَىٰ مَا يَقُولُونَ وَسَبِّحْ بِحَمْدِ رَبِّكَ قَبْلَ طُلُوعِ الشَّمْسِ وَقَبْلَ غُرُوبِهَا وَمِنْ أَنَاءِ
الَّيْلِ فَسَبِّحْ وَأَطْرَافَ النَّهَارِ لَعَلَّكَ تَرْضَىٰ

“Maka sabarlah engkau (Muhammad) atas apa yang mereka katakan, dan bertasbihlah dengan memuji tuhanmu, sebelum Matahari terbit dan sebelum terbenam, dan bertasbihlah (pula) pada waktu tengah malam dan di ujung siang hari, agar engkau merasa tenang.²¹(Thaha : 130)

Firman Allah قَبْلَ طُلُوعِ الشَّمْسِ وَقَبْلَ غُرُوبِهَا maksudnya adalah salat Subuh dan Asar. Firman Allah وَمِنْ أَنَاءِ اللَّيْلِ maksudnya adalah salat Isya, karena dilaksanakan setelah berlalunya beberapa saat pada malam hari. Sedangkan firman Allah وَأَطْرَافَ النَّهَارِ maksudnya adalah salat Zuhur dan Magrib. Disebut demikian karena salat Zuhur dilakukan pada akhir sisi siang yang pertama dan pada sisi siang yang kedua. Jadi, ia berada dalam dua sisi siang. Sedangkan sisi yang ketiga yaitu terbenamnya Matahari, dan ketika itulah dilakukan salat Magrib karena itulah dikatakan *athrafa* atau sisi-sisi.²²

Sementara itu dalam al-Munir, Wahbah Zuhaili berpendapat lain. Menurutnya yang dimaksud salat sebelum terbit Matahari adalah salat Subuh dan sebelum terbenamnya adalah salat Zuhur dan Asar atau salat Asar saja. Sedangkan salat pada waktu-waktu dimalam hari adalah salat Magrib dan Isya karena lafaz أَنَاءِ adalah bentuk jamak

²¹ Departemen Agama RI, *Al-Quran...*, h. 321.

²² Abu Ja'far, *Jami...* Jilid XVI, 209.

sehingga mencakup dua salat yaitu Magrib dan Isya. Sedangkan salat pada waktu-waktu disiang hari maksudnya adalah salat Zuhur.²³

5). Al-Quran surah ar-Rum ayat 17-18

فَسُبْحَانَ اللَّهِ حِينَ تُمْسُونَ وَحِينَ تُصْبِحُونَ (١٧) وَلَهُ الْحَمْدُ فِي السَّمَاوَاتِ وَالْأَرْضِ
وَعَشِيًّا وَحِينَ تُظْهِرُونَ (١٨)

“Maka bertasbihlah kepada Allah pada petang hari dan pada pagi hari (waktu subuh). Dan segala puji bagi-Nya baik di langit, di bumi, pada malam hari dan pada waktu waktu Zuhur (tengah hari)”.²⁴ (Ar-Rum : 17-18)

Maksudnya adalah Allah berfirman, Wahai manusia bertasbihlah kamu kepada Allah. Artinya berdoalah kepada Allah pada waktu malam, yaitu saat salat Magrib, dan waktu pagi yaitu saat salat Subuh, pada waktu petang maksudnya adalah bertasbih jugalah kamu kepada Allah pada waktu petang, yaitu saat salat Asar dan diwaktu kamu berada diwaktu Zuhur.²⁵

Ibnu Basysyar menceritakan kepada kami, ia berkata Abdurrahman menceritakan kepada kami, ia berkata Sufyan menceritakan kepada kami dari Ashim, dari Abu Razin, ia berkata Nafi bin Al Azraq bertanya kepada Ibnu Abbas tentang salat, adakah salat lima waktu dalam al-Quran? Ibnu Abbas menjawab, ya *“maka bertasbihlah kepada Allah diwaktu kamu berada dipetang hari”* maksudnya adalah salat Magrib *“dan waktu kamu berada diwaktu*

²³ Wahbah, *At-Tafsirul* Jilid VIII, 664.

²⁴ Departemen Agama RI, *Al-Quran...*, h. 406.

²⁵ Abu Ja'far, *Jami...* Jilid XVIII, 474.

Subuh” maksudnya adalah salat subuh. “*petang hari*” maksudnya adalah salat Asar. “*dan diwaktu kamu berada diwaktu Zuhur*” maksudnya adalah salat Zuhur. “*dan sesudah sembahyang Isya. (itulah) tiga aurat bagi kamu*”. (QS.An-Nuur:58).²⁶

Disebutkan sebuah keterangan dari Ibnu Abbas bahwa ayat ini mencakup salat lima waktu *تُسَوْنَ* menunjukkan waktu salat Magrib dan Isya, *نُصْبِحُونَ* menunjukkan waktu salat Subuh *وَعِشِيًّا* menunjukkan waktu salat Asar, *تُظْهِرُونَ* menunjukkan waktu Zuhur.²⁷

b. Dasar hukum dari Hadis

1). Hadis yang diriwayatkan Abdullah bin Amr r.a

عَنْ عَبْدِ اللَّهِ بْنِ عَمْرٍو رَضِيَ اللَّهُ عَنْهُ قَالَ أَنَّ رَسُولَ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ قَالَ وَ قَتُ الظُّهْرِ إِذَا زَالَتِ الشَّمْسُ وَكَانَ ظِلُّ الرَّجُلِ كَطُولِهِ مَا لَمْ يَخْضِرِ العَصْرُ وَوَقْتُ العَصْرِ مَا لَمْ تَصْفَرَ الشَّمْسُ وَوَقْتُ صَلَاةِ المَغْرِبِ مَا لَمْ يَغِبِ الشَّفَقُ وَوَقْتُ صَلَاةِ العِشَاءِ إِلَى نِصْفِ اللَّيْلِ الأَوْسَطِ وَوَقْتُ صَلَاةِ الصُّبْحِ مِنْ طُلُوعِ الفَجْرِ مَا لَمْ تَطْلُعِ الشَّمْسُ²⁸

“*Dari Abdullah bin Amr r.a berkata: Rasulullah SAW bersabda: waktu Zuhur apabila Matahari tergelincir, sampai bayang-bayang seseorang sama dengan tingginya, yaitu selama belum datang waktu Asar. Dan waktu Asar sebelum Matahari menguning. Dan waktu Magrib selama syafak (mega merah) belum terbenam. Dan waktu Isya sampai tengah malam yang pertengahan. Dan waktu Subuh mulai fajar menyingsing sampai selama Matahari belum terbit.*”

²⁶ *Ibid.*, 475.

²⁷ Wahbah, *At-Tafsirul* Jilid XI, 66.

²⁸ Imam Abi Al-Husain Muslim Bin al-Hajjaj al-Qusyairy, *Shahih Muslim*, (Beirut Libanon: Daar al-Alamiyah,t. Th), 427.

2). Hadis yang diriwayatkan oleh Jabir bin Abdullah r.a

أَخْبَرَنَا سُؤْيُدُبْنُ نَصْرٍ قَالَ أَنْبَأَنَا عَبْدُ اللَّهِ بْنُ الْمُبَارَكِ عَنْ حُسَيْنِ بْنِ عَلِيٍّ بْنِ حُسَيْنٍ قَالَ أَخْبَرَنِي وَهْبُ بْنُ كَيْسَانَ قَالَ حَدَّثَنَا جَابِرُ بْنُ عَبْدِ اللَّهِ قَالَ جَاءَ جِبْرِيْلُ عَلَيْهِ السَّلَامُ إِلَى النَّبِيِّ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ حِينَ زَالَتِ الشَّمْسُ فَقَالَ فُؤْمُ يَامُحَمَّدَ فَصَلِّ الطُّحْرَ حِينَ مَالَتِ الشَّمْسُ ثُمَّ مَكَثَ حَتَّى إِذَا كَانَ فِي رِجْلِ الرَّجُلِ مِثْلُهُ جَاءَهُ اللَّعْصِرُ فَقَالَ فُؤْمُ يَامُحَمَّدَ فَصَلِّ الْعَصْرَ ثُمَّ مَكَثَ حَتَّى إِذَا غَابَتِ الشَّمْسُ جَاءَهُ فَقَالَ فُؤْمُ فَصَلِّ الْمَغْرِبَ فَقَامَ فَصَلَّاهَا حِينَ غَابَتِ الشَّمْسُ سِوَاءً ثُمَّ مَكَثَ حَتَّى إِذَا ذَهَبَ الشَّفَقُ جَاءَهُ فَقَالَ فُؤْمُ فَصَلِّ الْعِشَاءَ فَقَامَ فَصَلَّاهَا ثُمَّ جَاءَهُ حِينَ سَطَعَ الْفَجْرُ فِي الصُّبْحِ فَقَالَ فُؤْمُ يَامُحَمَّدَ فَصَلِّ فَقَامَ فَصَلَّى الصُّبْحَ ثُمَّ جَاءَهُ مِنَ الْعَدِ حِينَ كَانَ فِي رِجْلِ الرَّجُلِ مِثْلُهُ فَقَالَ فُؤْمُ يَامُحَمَّدَ فَصَلِّ فَصَلَّ الطُّحْرَ فَقَالَ فُؤْمُ فَصَلِّ الْعَصْرَ ثُمَّ جَاءَهُ جِبْرِيْلُ عَلَيْهِ السَّلَامُ حِينَ كَانَ فِي رِجْلِ الرَّجُلِ مِثْلُهُ فَقَالَ فُؤْمُ يَامُحَمَّدَ فَصَلِّ الْعَصْرَ ثُمَّ جَاءَهُ الْمَغْرِبُ حِينَ غَابَتِ الشَّمْسُ وَقَتًا وَاحِدًا لَمْ يَزُلْ عَنْهُ فَقَالَ فُؤْمُ فَصَلِّ فَصَلَّ الْمَغْرِبَ ثُمَّ جَاءَهُ الْعِشَاءُ حِينَ ذَهَبَ ثُلُثُ اللَّيْلِ الْأَوَّلِ فَقَالَ فُؤْمُ فَصَلِّ الْعِشَاءَ ثُمَّ جَاءَهُ لِلصُّبْحِ حِينَ اسْفَرَ جِدًّا فَقَالَ فُؤْمُ فَصَلِّ فَصَلَّى الصُّبْحَ فَقَالَ مَا بَيْنَ هَذَيْنِ وَقْتُ كُؤْمُ²⁹ (رواه احمد و النسائي و الترميذي ينحوه)

“Telah menceritakan kepada kami Jabir bin Abdullah r.a bahwasanya Jibril datang kepada Nabi Muhammad SAW, lalu berkata kepadanya: bangunlah dan bersalatlah, lalu Nabi SAW melakukan salat Zuhur pada saat Matahari telah tergelincir. Kemudian datang lagi Jibril kepada nabi pada waktu Asar, lalu berkata: bangunlah dan bersalatlah, kemudian nabi salat Asar di kala bayangnang Matahari sama dengan bendanya. Kemudian Jibril datang lagi kepada nabi di waktu Magrib lalu berkata: bangunlah dan bersalatlah, kemudian Nabi salat Magrib di waktu Matahari terbenam. Kemudian Jibril datang lagi di waktu Isya lalu berkata: bangunlah dan bersalatlah Kemudian Nabi salat Isya di kala mega merah telah hilang. Kemudian datang lagi Jibril di waktu Subuh lalu berkata: bangunlah dan bersalatlah, Kemudian Nabi salat Subuh di kala fajar shadiq telah menyingsing. Kemudian Jibril datang lagi esok harinya di waktu Zuhur, kemudian Jibril berkata: bangunlah dan bersalatlah, kemudian Nabi salat Zuhur di kala bayang-bayang sesuatu sama dengannya. Kemudian Jibril datang lagi di waktu Asar

²⁹ Al-Hafiz Jalal al-Din as-Suyuthi, *Sunan al-Nasa'i*, (Beirut: Daar al-Kutub al-Alamiah, t. th.), 263.

dan ia berkata: bangunlah dan bersalatlailah, Kemudian nabi salat Asar dikala bayang-bayang Matahari dua kali panjang dirinya. Kemudian Jibril datang lagi di waktu Magrib dalam waktu yang sama, pada saat ia datang kemarin. Lalu berkata: bangunlah dan bersalatlailah, kemudian nabi salat. Kemudian Jibril datang lagi di waktu Isya di kala telah lalu separuh malam atau sepertiga malam, kemudian Nabi salat Isya. Kemudian Jibril datang kembali kepadanya di waktu telah terbit fajar shadiq dan ia berkata: bangunlah dan bersalatlailah, Kemudian Nabi salat Subuh. Kemudian jibril berkata: waktu-waktu diantara kedua ini, itu adalah waktu salat.” (HR. Imam Ahmad, Nasa’i dan Thirmidzi).

3. Batasan Salat Maktubah

Dari berbagai keterangan dalil-dalil yang telah dipaparkan diatas, ditemukan bahwa teks yang dijadikan landasan dalam menentukan awal waktu salat bersifat interpretatif. Sebagai implikasinya muncul perbedaan dalam menetapkan awal waktu salat. Kelompok pertama menyebutkan bahwa awal waktu salat ada tiga. Kelompok kedua menyebutkan bahwa awal waktu salat ada lima.³⁰

Di Indonesia yang lebih berkembang adalah tipologi kedua, yaitu awal waktu salat terdiri dari lima. Hukum asal mengetahui waktu-waktu salat adalah dengan mengenali fenomena alam yang Allah SWT jadikan sebagai pertanda masuknya awal waktu salat.

Untuk mengetahui masuknya waktu salat Allah telah mengutus Jibril untuk memberi arahan kepada Rasulullah SAW tentang waktu-waktu salat tersebut dengan acuan Matahari dan fenomena cahaya langit yang notabene juga disebabkan oleh pancaran sinar Matahari.³¹

³⁰ Susiknan Azhari, *Ilmu Falak Perjumpaan Khazanah Islam dan Sains Modern*, (Yogyakarta: Suara Muhammadiyah, 2011), 64.

³¹ Ahmad Musonnif, *Ilmu Falak (Metode Hisab Awal Waktu Salat, Arah Kiblat, Hisab Urfi dan Hisab Hakiki Awal bulan*, (Yogyakarta: Teras, 2011), 58.

Dari pemahaman terhadap teks-teks tersebut dirinci ketentuan waktu-waktu salat sebagai berikut :

a. Waktu salat Subuh

Awal waktu salat Subuh dimulai sejak terbit fajar³² sampai terbitnya Matahari.³³ Waktu fajar ditandai dengan munculnya sebercak cahaya putih di langit bagian timur. Namun dalam menentukan waktu fajar kita mesti hati-hati karena kemunculan sebercak cahaya putih terjadi dua kali yang kemudian memunculkan istilah fajar *kazib* dan fajar *shadiq*. Fajar *kazib* adalah fajar yang bohong sesuai dengan namanya. Maksudnya, pada saat dini hari menjelang pagi, ada cahaya yang agak terang yang memanjang dan mengarah ke atas di tengah di langit. Bentuknya seperti ekor serigala.

Sedangkan fajar *shadiq* , yaitu fajar yang benar benar fajar yang berupa cahaya putih agak terang yang menyebar di ufuk timur yang muncul beberapa saat sebelum Matahari terbit. Fajar inilah yang menandakan masuknya waktu subuh.³⁴ Saat posisi Matahari -20° dibawah ufuk timur bintang-bintang sudah mulai redup disebabkan kuatnya cahaya fajar. Maka ditetapkan tinggi Matahari awal waktu Subuh adalah -20° .³⁵

³² Fajar dalam istilah bahasa Arab bukanlah matahari. Sehingga ketika disebutkan terbit fajar, bukanlah terbit matahari. Fajar adalah cahaya putih agak terang yang menyebar di ufuk Timur yang muncul beberapa saat sebelum matahari terbit. Lihat Slamet Hambali, *Ilmu Falak...*, 124.

³³ Ahmad Izzuddin, *Ilmu Falak Praktis Metode Hisab Rukyat Praktis dan Solusi Permasalahannya*, (Semarang: Pustaka Rizki Putra, 2012), 83.

³⁴ Slamet Hambali, *Ilmu Falak 1 Penentuan Awal Waktu Shalat dan Arah Kiblat Seluruh Dunia*, (Semarang: Pascasarjana IAIN Walisongo, 2011), 124.

³⁵ Muhyiddin Khazin, *Ilmu Falak dalam Teori dan Praktik*, (Yogyakarta: Buana Pustaka, 2005), 92.

Namun ada sebagian ahli hisab lainnya yang menggunakan acuan -18^0 , -18.5^0 , dan -19^0 . Disamping itu, jika tempat yang akan dihitung waktu shalatnya berada pada ketinggian tertentu maka tinggi Matahari tadi masih harus dikoreksi dengan angka kerendahan ufuk.³⁶

b. Waktu Salat Zuhur

Awal waktu salat Zuhur dimulai sejak Matahari tergelincir yaitu sesaat setelah Matahari mencapai titik kulminasi³⁷ dalam peredaran harian, sampai tibanya waktu Asar.³⁸ Peristiwa ini dapat digambarkan ketika panjang bayangan suatu tongkat tidak terbentuk baik itu di barat maupun di timur, maka itu menunjukkan waktu istiwa yaitu terjadi ketika Matahari berada pada titik tertinggi.³⁹ Yang perlu difahami adalah bahwa ketika Zuhur, Matahari berada disudut waktu 0^0 dan ketika itu waktu menunjukkan pukul 12 menurut waktu matahari hakiki.⁴⁰ Ketika tongkat tersebut menunjukkan bayangan di sebelah timur maka waktu Zuhur telah masuk.⁴¹

Tidak ada perselisihan diantara empat imam mazhab mengenai kapan awal waktu salat Zuhur. Akan tetapi mereka berselisih pendapat mengenai kapan akhir waktu salat zuhur. Imam Malik, asy-

³⁶ Ahmad Musonnif, *Ilmu Falak...*, 71.

³⁷ Biasanya posisi ini diambil sekitar 2 menit setelah lewat tengah hari. Saat berkulminasi atas pusat bundaran Matahari berada dimeridian. Lihat Hambali, *Ilmu Falak....*, 127.

³⁸ Izzuddin, *Ilmu Falak Praktis...*, 83.

³⁹ Watni Marpaung, *Pengantar Ilmu Falak*, (Jakarta: Prenadamedia Group, 2015), 43.

⁴⁰ Izzuddin, *Ilmu Falak Praktis...*, 85.

⁴¹ Setelah titik pusat Matahari, dalam perjalanan Matahari arah ke Barat. Melepaskan diri dari meridian, ujung bayang bayang benda yang terpancang tegak lurus, akan melepaskan diri dari garis utara selatan dan membelok kesebelah timur. Lihat, A. Jamil, *Ilmu Falak (Teori Dan Aplikasi)*, (Jakarta: Amzah, 2016), 33.

Syafi'i, dan Dawud berpendapat bahwa akhir waktu salat zuhur adalah ketika panjang bayangan suatu benda melebihi sedikit saja dari panjang benda tersebut. Adapun menurut Abu Hanifah akhir waktu zuhur adalah saat panjang bayangan suatu benda sama dengan dua kali panjang benda tersebut.⁴²

Hal ini sebenarnya bisa dijelaskan secara ilmiah. Dimana Saudi Arabia yang berlintang sekitar 20^0 - 30^0 utara, pada saat Matahari tergelincir panjang bayang-bayang dapat mencapai panjang bendanya bahkan lebih. Keadaan ini dapat terjadi ketika Matahari sedang berposisi jauh di selatan yaitu sekitar bulan Juni dan Desember.⁴³

Sudut waktu ketika Matahari berada di meridian adalah 00 dan pada saat itu waktu menunjukkan jam 12 menurut waktu Matahari hakiki. Namun pada kenyataannya waktu pertengahan belum tentu menunjukkan jam 12, terkadang waktu pertengahan masih kurang dari jam 12 atau bahkan lebih, Hal tersebut dipengaruhi oleh *equation of time*.⁴⁴ Oleh karenanya, waktu pertengahan waktu Matahari berada di meridian (meridian pass) dirumuskan dengan $MP = 12 - \text{equation of time}$.⁴⁵ Sesaat setelah waktu inilah sebagai permulaan waktu Zuhur

⁴² Muhammad bin Ahmad bin Muhammad bin Ahmad Ibnu Rusyd al-Qurtuby, *Bidayah al-Mujtahid wa Nihayah al-Muqtasid*, Juz II, (Baerut: Dar al-Kutub al-Ilmiyah, 1996), 116.

⁴³ Izzuddin, *Ilmu Falak Praktis...*, 83.

⁴⁴ Equation of time atau *Ta'dilul Waqti* yaitu selisih waktu antara waktu matahari hakiki dengan waktu matahari rata-rata. Dalam astronomi biasa disebut dengan *equation of time* yang diartikan dengan perata waktu. Lihat Muhyiddin Khazin, *Kamus Ilmu Falak*, (Yogyakarta: Buana Pustaka, 2005), 79.

⁴⁵ Muhyiddin Khazin, *Ilmu Falak...*, 89.

menurut waktu pertengahan dan waktu ini pula lah sebagai pangkal hitungan untuk waktu-waktu salat lainnya.

c. Waktu Salat Asar

Awal waktu Asar, berdasarkan literatur-literatur fikih tidak ada kesepakatan. Hal ini dikarenakan fenomena yang dijadikan dasar tidak jelas atau terkesan apa adanya.⁴⁶ Ketidakjelasan itu berpangkal pada akhir waktu Zuhur yang berbeda yang selanjutnya berpengaruh pula pada penentuan awal waktu salat Asar. Karena pada dasarnya berakhirnya waktu Zuhur adalah permulaan awal waktu Asar.

Awal waktu salat Asar adalah ketika bayangan benda sama panjangnya dengan benda itu sendiri dan pendapat lain menyebutkan bahwa awal waktu Asar adalah saat bayangan dua kali panjang dari benda sebenarnya. Tapi hal ini masih menimbulkan beberapa penafsiran karena fenomena seperti itu tidak dapat digeneralisasi sebab bergantung pada musim atau posisi tahunan Matahari. Pada musim dingin hal itu dapat dicapai pada waktu Zuhur, bahkan mungkin tidak pernah terjadi karena bayangan selalu lebih panjang daripada tongkatnya.⁴⁷

Di Indonesia khususnya Departemen Agama (sekarang Kementrian Agama) menganut kriteria waktu Asar adalah saat panjang bayangan sama dengan panjang benda ditambah panjang

⁴⁶ Slamet Hambali, *Ilmu Falak 1...*, 127.

⁴⁷ Hambali, *Ilmu Falak 1...*, 127.

bayangan saat istiwa. Dengan demikian, besarnya sudut tinggi matahari waktu Asar bervariasi setiap harinya.⁴⁸ Berdasarkan uraian diatas mengenai waktu Asar. Dapat di simpulkan bahwa waktu Asar di mulai saat panjang bayang-bayang suatu benda sama dengan panjang bayang-bayang pada saat Matahari berkulminasi sampai tiba waktu Magrib.⁴⁹

Kulminasi Matahari di titik zenit terjadi apabila harga lintang dan deklinasi Matahari sama. Jika tidak maka Matahari akan berkulminasi di selatan atau di utara titik zenit, sehingga benda yang berdiri tegak lurus sudah mempunyai panjang bayang-bayang tertentu. Untuk itu perlu ditakwil apabila keadaan seperti ini, yaitu awal waktu Asar masuk bila bayang bayang yang sudah ada pada saat kulmiansi Matahari sudah bertambah dengan sepanjang bendanya.⁵⁰

d. Waktu Salat Magrib

Awal waktu salat Magrib dimulai sejak Matahari terbenam sampai tibanya waktu Isya.⁵¹ Matahari dapat dikatakan sudah terbenam jika dalam pandangan mata piringan atas Matahari sudah bersinggungan dengan ufuk. Waktu Magrib dalam ilmu falak berarti saat terbenam matahari, artinya seluruh piringan matahari tidak terlihat oleh pengamat. Piringan Matahari berdiameter 32 menit busur, setengahnya berarti 16 menit busur, selain itu didekat horison

⁴⁸ Watni Marpaung, *Pengantar...*, 44.

⁴⁹ Ahmad Izzuddin, *Ilmu Falak Praktis...*, 83.

⁵⁰ Ahmad Musonnif, *Ilmu Falak....*, 72.

⁵¹ Ahmad Izzuddin, *Ilmu Falak Praktis...*, 83.

terdapat refraksi yang menyebabkan kedudukan Matahari lebih tinggi dari sebenarnya yang diperkirakan sekitar 34 menit.⁵²

Adapun mengenai rentang waktu diperbolehkannya salat Magrib para ulama berbeda pendapat, menurut Imam Maliki dan Imam Syafi'i waktu salat Magrib tidak luas yakni hanya mempunyai satu waktu. Rentang waktu ini dapat diibaratkan saat seseorang mengumandangkan azan, berwudu, menutup aurat (berpakai), melaksanakan salat Magrib, dan melaksanakan lima rakaat sunah. Sedangkan menurut Abu Hanifah, Imam Ahmad, Abu Tsaur dan Abu Dawud lama waktu Magrib itu luas yaitu diantara tenggelamnya Matahari sampai tenggelamnya mega.⁵³

Perhitungan tentang kedudukan maupun posisi benda benda langit termasuk Matahari, pada mulanya adalah perhitungan kedudukan atau posisi titik pusat Matahari diukur atau dipandang dari titik pusat Bumi, sehingga dalam melakukan perhitungan tentang kedudukan Matahari terbenam kiranya perlu memasukkan horisontal parallaks Matahari, kerendahan ufuk atau dip, refraksi cahaya, dan semidiameter Matahari. hanya saja karena parallaks Matahari itu terlalu kecil nilainya yakni sekitar $00^{\circ}00'08''$ sehingga parallaks Matahari dalam perhitungan waktu Magrib dapat diabaikan.⁵⁴

⁵² Slamet Hambali, *Ilmu Falak 1...*, 131.

⁵³ Al-Qurtuby, *Bidayah...*, 206.

⁵⁴ Khazin, *Ilmu Falak...*, 90.

e. Waktu Salat Isya

Awal waktu Isya dimulai sejak hilang mega merah sampai separuh malam. Ada juga pendapat yang mengatakan sepertiga, ada juga yang mengatakan akhir salat Isya adalah terbitnya fajar.⁵⁵

Saat Matahari terbenam di ufuk barat, permukaan Bumi tidak secara langsung menjadi gelap. Hal ini terjadi karena adanya partikel-partikel yang berada di angkasa yang mempengaruhi pembiasan sinar Matahari. sehingga meskipun sinar Matahari tidak mengenai Bumi namun masih ada bias dari cahaya Matahari dari partikel-partikel itu. Dalam ilmu falak dikenal dengan “cahaya senja” atau “*twilight*”.⁵⁶

Dalam ilmu astronomi istilah saat setelah Matahari terbenam dan sebelum Matahari terbit, yaitu *twilight* dibagi menjadi 3 tingkatan yaitu *Civil twilight*, *Nautical twilight*, *Astronomical twilight*. Batas *civil twilight* jika Matahari 06° di bawah horizon, benda-benda dilapangan terbuka masih tampak batas batas bentuknya, bintang yang paling terang dapat di lihat. Kemudian batas *nautical twilight* adalah 12° dibawah horizon, jika di laut ufuk hampir tidak kelihatan maka semua bintang terang dapat dilihat. Adapun batas *astronomical twilight* adalah jika Matahari senilai 18° dibawah ufuk maka gelap malam sudah sempurna (awal waktu Isya).⁵⁷

⁵⁵ Izzuddin, *Ilmu Falak Praktis...*, 83.

⁵⁶ Khazin, *Ilmu Falak...*, 91.

⁵⁷ A. Jamil, *Ilmu Falak* 45-46

Oleh karena pada posisi Matahari -18° di bawah ufuk malam sudah gelap karena telah hilang bias partikel (mega merah), maka ditetapkan bahwa awal waktu Isya apabila tinggi Matahari -18° . Oleh sebab itu $h \text{ Isya} = -18^{\circ}$.⁵⁸ Sementara itu ada ahli hisab yang menggunakan ketinggian -17° dan ada juga yang menggunakan -19° tentu saja ketinggian Matahari tersebut masih perlu dikoreksi lagi dengan kerendahan ufuk jika tempat yang akan di hisab berada pada diatas ketinggian tertentu.⁵⁹

4. Komponen untuk Hisab Waktu Salat

Secara umum data yang diperlukan untuk melakukan perhitungan waktu salat adalah data yang berkaitan dengan koordinat tempat dan posisi Matahari dimana waktu salat itu akan dihitung. Berikut penjelasannya:

a. Lintang Tempat

Lintang tempat didefinisikan sebagai jarak sepanjang meridian Bumi diukur dari khatulistiwa sampai suatu tempat dimaksud. Lintang tempat minimal 0° dan maksimal 90° . Lintang dalam bahasa Inggris bisa di istilahkan *latitude* dan dalam bahasa Arab di istilahkan *Urd al Balad*, sedangkan Siradj Dahlan mengistilahkan dengan malang. Dalam astronomi lintang tempat di beri tanda dengan huruf Yunani phi (π).⁶⁰

⁵⁸ Muhyiddin Khazin, *Ilmu Falak...*, 92.

⁵⁹ Ahmad Musonnif, *Ilmu Falak...*, 74.

⁶⁰ SusiknanAzhari, *Ensiklopedi Hisab Rukyat*, (Yogyakarta: Pustaka Pelajar, 2012), 134.

Daerah yang terletak di sebelah utara garis khatulistiwa memiliki lintang positif sedangkan disebelah selatan, lintangnya negatif. Misalnya Fukuoka (Jepang) memiliki lintang 33:35 derajat lintang utara (LU) maka nilainya positif. Sedangkan Jakarta berada di koordinat 6:10:0 derajat lintang selatan (LS) maka nilainya negatif.⁶¹

Lintang tempat dapat diperoleh data-datanya melalui tabel atau dapat dicari melalui *Global Position System (GPS)*, *google earth*, *google map*, peta, tabel dan lain-lain.⁶²

b. Bujur Tempat

Bujur tempat didefinisikan sebagai jarak yang diukur sepanjang busur ekuator dari bujur yang melalui kota Greenwich sampai bujur yang melalui tempat /negeri yang dimaksud. Daerah yang terletak disebelah timur Greenwich memiliki bujur positif. Misalnya Jakarta memiliki koordinat 106:51:0 derajat Bujur Timur, maka nilainya positif. Sedangkan disebelah barat Greenwich memiliki bujur negatif. Misalnya Los Angeles memiliki koordinat bujur 118:28 derajat Bujur Barat, maka nilainya negatif.⁶³

Bujur tempat dalam bahasa Inggris biasa diistilahkan dengan *longitude* dan dalam bahasa Arab di istilahkan *Thul al-Balad*, sedangkan Siradj Dahlan mengistilahkan *moedjoer*, tanda

⁶¹ Rinto Anugraha, *Mekanika Benda Langit*, (Yogyakarta: Jurusan Fisika FMIPA UGM, 2012), 88.

⁶² Ahmad Izzuddin, *Ilmu Falak Praktis...*, 84.

⁶³ Rinto Anugraha, *Mekanika...*, 88.

astronominya λ (lambda).⁶⁴ Dengan adanya pembagian bujur ini maka pembagian waktu tempat-tempat di Bumi dapat dilakukan, yaitu setiap 15^0 merupakan satu zona waktu. Untuk Indonesia terdapat 3 zona waktu yaitu WIB, WITA, dan WIT dimana WIB adalah UTC +7, WITA +8 dan WIT +9.

c. Deklinasi Matahari

Deklinasi Matahari atau *mail syams* adalah Jarak sepanjang lingkaran deklinasi di hitung dari equator sampai Matahari.⁶⁵ Untuk ketelitian perhitungan hendaknya diambil data *deklinasi* Matahari pada jam yang semestinya. Sebagai contoh jika akan menghitung waktu Zuhur maka ambillah data deklinasi jam 12.00 WIB atau jam 05.00 UT, Asar jam 15.00 WIB atau jam 08.00 UT, Maghrib jam 18.00 WIB atau jam 11 UT dan seterusnya.⁶⁶ Perlu difahami bahwa yang dimaksud deklinasi disini adalah deklinasi yang tampak (bukan Matahari hakiki)⁶⁷

Deklinasi di belahan langit bagian utara bernilai positif sedangkan di belahan selatan adalah negatif. Ketika Matahari melintasi khatulistiwa deklinasinya adalah 0 hal ini terjadi pada tanggal 21 Maret dan tanggal 23 September. Adapun nilai maksimal

⁶⁴ Susiknan Azhari, *Ensiklopedi...*, 47.

⁶⁵ Muhyiddin Khazin, *Kamus Ilmu Falak...*, 52.

⁶⁶ Ahmad Izzuddin, *Pengantar Ilmu Falak...*, 84.

⁶⁷ Kementerian Agama RI, *Ephemeris Hisab Rukyat 2018*, (Jakarta: Direktorat Urusan Agama Islam Dan Pembinaan Syariah Direktorat Jenderal Bimbingan Masyarakat Islam Kementerian Agama RI, 2017), 2.

deklinasi adalah $+23^{\circ} 27'$ ke sebelah utara dan $+23^{\circ} 27'$ ke sebelah selatan.⁶⁸

d. *Equation Of Time*

Equation of Time atau *Ta'dilul Auqat* yaitu selisih waktu antara waktu Matahari hakiki dengan waktu Matahari rata-rata.⁶⁹ *Equation of time* dalam bahasa Indonesia dikenal sebagai perata waktu simbolnya adalah “*e*”.⁷⁰

Sama halnya dengan deklinasi hendaknya *equation of time* diambil pada jam yang semestinya. Sebagai contoh jika akan menghitung waktu Zuhur maka ambillah data *equation of time* jam 12.00 WIB atau jam 05.00 UT, Asar jam 15.00 WIB atau jam 08.00 UT, Maghrib jam 18.00 WIB atau jam 11 UT dan seterusnya⁷¹

Equation of time bernilai positif (+) jika saat pukul 12.00 Matahari sudah melewati titik kulminasi atas, dan negatif (-) jika saat pukul 12.00 Matahari belum melewati titik kulminasi atas.⁷²

e. Ketinggian Tempat

Tinggi tempat secara geodetik yang dinotasikan dengan (h) disebut sebagai jarak titik yang bersangkutan dari *ellipsoid* referensi di dalam arah garis normal terhadap *ellipsoid* referensi tersebut. Kedataran dan ketinggian suatu tempat di permukaan bumi ini

⁶⁸ Slamet Hambali, *Ilmu Falak 1...*, 55.

⁶⁹ Muhyiddin Khazin, *Kamus Ilmu Falak...*, 79.

⁷⁰ Kementerian Agama RI, *Ephemeris Hisab ...*, 2.

⁷¹ Ahmad Izzuddin, *Pengantar Ilmu Falak...*, 84.

⁷² Slamet Hambali, *Ilmu Falak 1...*, 92.

diukur dengan menggunakan referensi kedataran air laut, sehingga disebut dengan istilah *mean sea level* atau di atas permukaan laut (dpl).⁷³

Ketinggian tempat ini mempengaruhi kapan terbit dan terbenamnya Matahari. Tempat yang lebih tinggi di atas permukaan laut akan lebih awal dalam menyaksikan terbitnya Matahari dan akan lebih akhir menyaksikan terbenamnya Matahari.⁷⁴ Ketinggian tempat selanjutnya akan berdampak pada sudut waktu Matahari. Sehingga tidak bisa dipungkiri ketinggian tempat dikatakan berpengaruh pada jadwal waktu salat yang berhubungan dengan kerendahan ufuk. Jadwal waktu salat yang pada perhitungannya tidak menggunakan koreksi ketinggian tempat menyebabkan samanya waktu salat antara yang berada di tempat rendah maupun tinggi. Padahal secara nyata perbedaan tinggi ini menyebabkan adanya perbedaan ketika melihat Matahari terbit dan tebenam.

f. Zona waktu

Zona waktu atau waktu daerah dalam astronomi dikenal dengan *time zone* adalah waktu yang digunakan di suatu daerah atau wilayah yang berpedoman pada bujur atau meridian perkelipatan 15° . Misalnya WIB = 115° , WITA = 120° , dan WIT = 135° .⁷⁵ Zona waktu

⁷³ Encep Abdul Rojak, Dkk, "Koreksi Ketinggian Tempat Terhadap Fikih Waktu Salat: Analisis Jadwal Waktu Salat Kota Bandung", dalam *Al- Ahkam*, Vol 27, no. 2, Oktober 2017, h. 254.

⁷⁴ Rinto Anugraha, *Mekanika...*, 88.

⁷⁵ Khazin, *Kamus Ilmu Falak...*, 90.

dihitung dari *Greenwich* dengan berpedoman pada *universal time* yaitu waktu Matahari menengah yang bereferensi ke *Greenwich Mean Solar Time* atau GMT.⁷⁶

Zona waktu ini sangat berpengaruh terhadap penyusunan jadwal waktu salat. Dimana waktu yang digunakan kita saat ini adalah waktu yang berbasis pada waktu daerah sedangkan dalam perhitungan awal waktu salat waktu yang digunakan adalah waktu berdasar pada fenomena yang benar-benar terjadi pada daerah tertentu dengan koordinat tertentu. Atas dasar itu maka perlu adanya koreksi waktu daerah dengan waktu setempat. Selain itu, garis bujur itu bernilai 0 – 180 derajat ke timur dan 0 – 180 derajat ke barat, artinya berjumlah 360 derajat. Apabila dalam sehari semalam Bumi berotasi 360 derajat selama 24 jam maka dapat diartikan setiap 1 jam akan menempuh 15 derajat garis bujur. Hal ini mengindikasikan bahwa untuk setiap 1 derajat bujur perbedaan waktunya adalah 4 menit..⁷⁷

g. Semidiameter

Semidiameter adalah jarak antara piringan benda langit dengan piringan luarnya titik, atau seperdua garis tengah piringan benda langit.⁷⁸ Dalam hal ini semidiameter yang dimaksud adalah semidiameter Matahari. Semakin jauh jarak Bumi dan Matahari

⁷⁶ Hasanuddin Z. Abidin, *Geodesi Satelit*, (Jakarta: Anem Kosong Anem, 2001), 55.

⁷⁷ Moelki Fahmi Ardiansyah “Implementasi Koordinat tengah Kabupaten atau Kota dalam Perhitungan Waktu Salat”, dalam *Al-Ahkam*, Vol 27, no. 2, Oktober 2017, 227.

⁷⁸ Muhyiddin Khazin, *Kamus Ilmu Falak...*, 61.

maka semakin kecil semidiamter Mataharinya begitu juga sebaliknya.

h. Refraksi

Refraksi atau pembiasan sinar adalah perbedaan antara tinggi suatu benda langit yang terlihat dengan tinggi sebenarnya. Refraksi ini diakibatkan oleh pembiasan sinar. Pembiasan ini terjadi karena sinar yang sampai ke mata kita telah melewati lapisan-lapisan atmosfer sehingga benda langit terlihat sedikit lebih tinggi dari posisi sebenarnya. Pembiasan bagi benda langit yang berada di zenith adalah 0° . Semakin dekat benda langit dengan ufuk maka semakin besar nilai refraksinya. Untuk benda langit yang sedang terbenam atau piringan atasnya bersinggungan dengan ufuk maka nilai refraksinya adalah $34' 34''$ ⁷⁹

i. Kerendahan Ufuk

Kerendahan ufuk atau *ikhtilaf ul ufuq* adalah perbedaan kedudukan antara ufuk yang sebenarnya (*hakiki*) dan ufuk yang terlihat (*mar'i*) oleh seorang pengamat, Dalam astronomi disebut *Dip*.⁸⁰ Kerendahan ufuk ini berkaitan erat dengan tinggi tempat oleh karenanya ntuk mencari *Dip* digunakan rumus $dip = 0^\circ 1,76' \sqrt{\text{tinggi tempat}}$.

⁷⁹ Muhyiddin Khazin, *Kamus...*, 19.

⁸⁰ *Ibid.*, 33.

j. *Ikhtiyat*

Ikhtiyat adalah suatu langkah pengamanan dengan cara menambahkan atau mengurangi waktu agar jadwal waktu salat tidak mendahului awal waktu atau melampaui akhir waktu.⁸¹ Tujuan *ikhtiyat* adalah untuk mengantisipasi adanya kesalahan dalam perhitungan. Nilai *ikhtiyat* berkisar antara 1 – 4 menit, tetapi karena semakin presisinya perhitungan yang ada saat ini maka dianjurkan untuk menggunakan *ikhtiyat* tidak lebih dari 2 menit kecuali untuk waktu Zuhur.⁸²

Ikhtiyat ini diharapkan bisa menjangkau daerah-daerah sekitar koordinat tempat yang dihitung dengan perkiraan kurang lebih 27,5 km permenitnya terutama bagian sebelah barat dari koordinat tempat yang dihitung waktu salatnya.⁸³

B. Pola Kerja Program Android

1. Perkembangan Android

Dalam sejarahnya Android sebenarnya merupakan perusahaan platform yang terbilang muda karena baru baru rilis pada Oktober 2003. Pada mulanya Android diprakarsai oleh empat orang ahli teknologi Andy Rubin, Rich Miner, Nick Sears dan Cris White dibawah sebuah perusahaan bernama Android Inc di Palo Alto, California. Pada mulanya tujuan

⁸¹ Susiknan Azhari, *Ilmu Falak...*, 73.

⁸² Musonnif, *Ilmu Falak...*, 66.

⁸³ Muhyiddin Khazin, *Ilmu Falak dalam Teori...*, 82.

platform ini adalah untuk mengembangkan sebuah sistem operasi yang lebih canggih bagi sebuah kamera digital. Namun seiring dengan perkembangannya dan tuntutan pasar global Adroid berubah menjadi sebuah perusahaan yang bergerak dibidang pengembangan sistem operasi *smartphone*.⁸⁴

Android adalah sistem operasi dan platform pemrograman yang dikembangkan oleh *Google* untuk ponsel cerdas dan perangkat seluler lainnya (seperti tablet). Android bisa berjalan di beberapa macam perangkat dari banyak produsen yang berbeda. Android menyertakan kit *development* perangkat lunak untuk penulisan kode asli dan perakitan modul perangkat lunak untuk membuat aplikasi bagi pengguna Android. Android juga menyediakan pasar untuk mendistribusikan aplikasi. Secara keseluruhan, Android menyatakan ekosistem untuk aplikasi seluler.⁸⁵

Visi Android yang terbilang baru membuat raksasa dunia maya *Google* berminat untuk mengakuisisi Android. Hingga pada Agustus 2005 Android resmi dipinang *Google*. Android sendiri dibangun berbasis *platform* Linux yang bersifat *open source*. Dengan nama besar *Google* dan konsep *open source* pada Android, *operating system* ini tidak membutuhkan waktu lama untuk bisa bersaing dengan *operating system* lainnya yang telah ada lebih dulu seperti *Symbian*, *Windows Mobile*, *Blackberry* dan *iOS*. Kini

⁸⁴ Nadia Firly, *Create Your Android...*, 2.

⁸⁵ Google Developer Training Team, *Android Developer Fundamentals Course* (2016), 6.

Android telah menjelma menjadi penguasa *operating system* bagi *smartphone* di dunia.⁸⁶

Setelah diakuisisi *Google*, pada 5 November 2007 untuk pertamakalinya Android meluncurkan versi beta yang bersamaan dengan berdirinya *Open Handset Alliance* atau OHA.⁸⁷ Hal tersebut dijadikan momentum dan ditetapkan sebagai hari Android. Satu minggu setelah peluncuran versi beta, Android meluncurkan *Software Development Kit* atau SDK tepat pada 12 November 2007. SDK ini memungkinkan siapapun untuk bisa mengembangkan aplikasi Android mereka sendiri.⁸⁸

Dalam rangka memenuhi kebutuhan global setiap tahunnya Android selalu melakukan inovasi dan peluncuran versi terbarunya. Berbagai fitur yang ditawarkan Android menjadikannya sebagai raja *platform* ponsel pintar sampai saat ini. Berikut adalah tabel yang menunjukkan berbagai versi Android yang telah dirilis oleh perusahaan ini :⁸⁹

⁸⁶ Hendra Nugraha Lengkong, “Perancangan Penunjuk Rute Pada Kendaraan Pribadi Menggunakan Aplikasi *Mobile GIS* Berbasis Android Yang Terintegrasi Pada *Google Maps*” (2015); Jurnal Teknik Elektro dan Komputer UNSRAT, 20.

⁸⁷ *Open Handset Alliance* (OHA) adalah konsorsium dari perusahaan-perusahaan teknologi seperti Google, produsen perangkat seluler seperti HTC, Sony dan Samsung, operator nirkabel seperti Sprint Nextel dan T-Mobile, serta produsen chipset seperti Qualcomm dan Texas Instruments. OHA sendiri bertujuan untuk mengembangkan standar terbuka bagi perangkat seluler. Saat itu, Android diresmikan sebagai produk pertamanya; sebuah platform perangkat seluler yang menggunakan kernel Linux versi 2.6. Telepon seluler komersial pertama yang menggunakan sistem operasi Android adalah HTC Dream, yang diluncurkan pada 22 Oktober 2008. https://id.wikipedia.org/wiki/Open_Handset_Alliance diakses 21.29 02 juli 2019

⁸⁸ Nadia Firly, *Create Your Android...*, 3.

⁸⁹ *Ibid.*

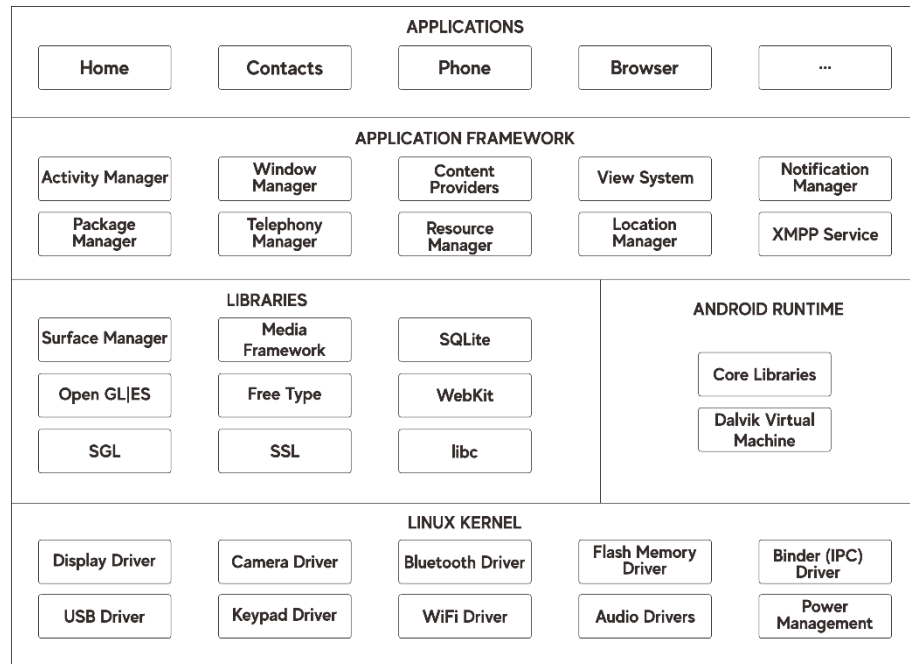
Tabel 2.1 Versi Android

Versi	Nama	Tanggal Rilis
1.0 (API level 1)	-	23 September 2008
1.1 (API level 2)	-	9 Februari 2009
1.5 (API level 3)	Cupcake	27 April 2009
1.6 (API level 4)	Donut	15 September 2009
2.0 (API level 5)	Eclair	26 Oktober 2009
2.0.1 (API level 6)	Eclair	3 Desember 2009
2.1 (API level 7)	Eclair	12 Januari 2010
2.2-2.2.3 (AP level 8)	Froyo	20 Mei 2010
2.3-2.3.2 (AP level 9)	Gigerbread	6 Desember 2010
2.3-2.3.7(AP level 10)	Gigerbread	9 Februari 2011
3.0 (API level 11)	Honeycomb	22 Februari 2011
3.1 (API level 12)	Honeycomb	10 Mei 2011
3.2 (API level 13)	Honeycomb	15 Juli 2011
4.0-4.0.2 (APlevel 14)	Ice Cream Sandwich	19 Oktober 2011
4.0.3.-4.0.4(APIlevel 15)	Ice Cream Sandwich	16 Desember 2011
4.1 (API level 16)	Jelly Bean	27 Juni 2012
4.2 (API level 17)	Jelly Bean	29 Oktober 2012
4.3 (API level 18)	Jelly Bean	24 Juli 2013
4.4 (API level 19)	Kitkat	31 Oktober 2013
5.0 (API level 21)	Lollipop	12 November 2014
6.0 (API level 23)	Marsmallow	5 Oktober 2015
7.0 (API level 24)	Nougat	9 Maret 2016
7.1 (API level 25)	Nougat	19 Oktober 2016
8.0 (API level 26)	Oreo	21 Maret 2017

2. Pola Kerja Android

Secara garis besar arsitektur Android dapat digambarkan sebagai berikut:

Gambar 2.1 Arsitektur Android



a. Applications dan Widgets

Applications dan Widgets ini adalah layer dimana kita berhubungan dengan aplikasi saja. Aplikasi tersebut biasanya kita download, lalu melakukan instalasi dan menggunakan aplikasi yang sudah terinstal. Di layer terdapat aplikasi inti seperti klien email (Gmail), program SMS, kalender, peta (Google Map), browser (Google Chrome), kontak, android market (Google Play), pemutar video (Youtube), dan lain-lain.

Semua aplikasi dibangun di atas *application layer* atau pada *library API* yang sama. *Application layer* ini berjalan di dalam android

runtime, menggunakan *class* dan layanan yang dibuat dari *application framework*.⁹⁰

b. *Application Framework*

Android adalah *Open Development Platform*. Android menawarkan kepada pengembang untuk membangun aplikasi yang bagus dan inovatif. Pengembang bebas untuk mengakses perangkat keras, akses informasi *resources*, menjalankan *service background* dan sebagainya. Sehingga bisa kita simpulkan *Application Framework* ini adalah layer yang dapat memungkinkan kita melakukan pengembangan/pembangunan aplikasi yang dapat dijalankan pada sistem operasi android.

Application framework menyediakan berbagai *class* yang digunakan untuk membuat aplikasi android. Termasuk juga berhubungan dengan akses ke *hardware* dan manajemen *user-interface* dan manajemen memori.⁹¹

c. *Libraries*

Libraries merupakan layer tempat fitur-fitur android berada. Pada umumnya *libraries* diakses untuk menjalankan aplikasi. Beberapa *library* yang terdapat pada android diantaranya adalah *libraries Media* untuk memutar media video atau audio, *libraries* untuk menjalankan tampilan,

⁹⁰ Nizar Rabbi Radliya, *Modul Kuliah Pemrograman Mobile (Android)*, (2016); UNIKOM, 2.

⁹¹ *Ibid.*

libraries Graphic, libraries SQLite untuk dukungan database, dan masih banyak *library* lainnya.⁹²

d. Android Runtime

Terdiri dari *Core Libraries* dan *Dalvik Virtual Machine (DVM)*. *Core Libraries* ini meliputi serangkaian *library Java* yang terdiri dari berbagai macam fungsi dasar pada bahasa pemrograman Java, sedangkan DVM merupakan *Java Virtual Machine* (semacam JVM pada Java ME) yang secara khusus dijalankan pada android. DVM dapat mengeksekusi file dengan format *Dalvik Executable (*.dex)* yang telah dioptimasi dengan menggunakan memori yang minimal. DVM telah di-register dan menjalankan berbagai class yang di-compile menggunakan compiler Java yang kemudian ditransformasi menjadi format *.dex dengan menggunakan 'dx' tool. DVM menggunakan kernel linux untuk menjalankan berbagai fungsi seperti *threading* dan manajemen memori.⁹³

e. Linux Kernel

Linux kernel adalah layer dimana ini dari sistem operasi dari Android itu berada. Berisi file-file sistem yang mengatur sistem *processing, memory, resource, drivers*, dan sistem-sistem lainnya. Fondasi *platform* Android adalah kernel Linux. Sebagai contoh, *Android Runtime (ART)* bergantung pada kernel Linux untuk

⁹² <https://www.twoh.co/2012/09/18/mengenal-arsitektur-sistem-operasi-android/> diakses 03 Juli pukul 15:12

⁹³ Nizar Rabbi Radliya, *Modul Kuliah Pemrograman...*, 3.

fungsi-fungsionalitas dasar seperti threading dan manajemen memori tingkat rendah.

Menggunakan kernel Linux memungkinkan Android untuk memanfaatkan fitur keamanan inti dan memungkinkan produsen perangkat untuk mengembangkan driver perangkat keras untuk kernel yang cukup dikenal.⁹⁴

⁹⁴ <https://developer.android.com/guide/platform?hl=id> diakses 03 Juli pukul 15:08

BAB III

DESAIN, RANCANGAN DAN IMPLEMENTASI PROGRAM

ACCURATE ATOMIC TIME FOR SALAT

A. Pola Kerja Jam Atom BMKG

Waktu atom (*Atomic Time AT*) didasarkan pada osilasi elektro magnetik yang dihasilkan oleh transisi kuantum suatu atom. Unit waktu secara internasional pada sistem waktu atom adalah detik yang didefinisikan sebagai berikut : “*The second is the duration of 9192631770 periods of the radiation corresponding to the transition between the two hyperfine levels of the ground state of the Cesium 133 atom*”.¹ Atom lainnya selain Cesium, seperti Rubidium dan Hydrogen Maser, juga dapat digunakan untuk merealisasikan sistem waktu atom.

Waktu Atom Internasional (*International Atomic Time*) ditetapkan dan dijaga oleh BIPM di Paris. Meskipun secara resmi diberlakukan sejak Januari 1972, waktu atom Internasional sebenarnya telah ada sejak Juli 1955. Sampai November 1999 waktu atom Internasional ditentukan berdasarkan data dari sekitar 50 laboratorium yang mengoperasikan sekitar 200 jam (osilator)² atom di seluruh dunia. Waktu atom Internasional

¹ Patrice F Dassonville, *The Invention of Time and Space: Origins, Definitions, Nature, Properties* (Springer,Switzerland), 83.

² Osilator (Oscillator) adalah suatu rangkaian elektronika yang menghasilkan sejumlah getaran atau sinyal listrik secara periodik dengan amplitudo yang konstan. Gelombang sinyal yang dihasilkan ada yang berbentuk Gelombang Sinus (*Sinusoid Wave*), Gelombang Kotak (*Square Wave*) dan Gelombang Gigi Gergaji (*Saw Tooth Wave*). <https://teknikelektronika.com/pengertian-osilator-prinsip-kerja-oscillato/> diakses 25 Juni 2019 23:51

ditentukan dengan mengambil nilai rata-rata (dengan pembobotan) dari pembacaan seluruh jam yang terlibat.

Pada prinsipnya ada beberapa jenis jam (osilator) atom, seperti yang ditunjukkan pada tabel berikut:³

Tipe Osilator	Frekuensi Osilasi (Hz)	Stabilitas per hari (df/f)	Waktu untuk kehilangan 1 detik (tahun)
Kristal Quartz	5.000.000 (tipikal)	10.E-09	30
Rubidium	6.834.682.613	10.E-12	30 ribu
Cesium	9.192.631.770	10.E-13	300 ribu
Hydrogen Maser	1.420.405.751	10.E-15	30 juta

Adapun jam atom milik BMKG adalah jam atom Cesium 4310 dengan *backup* jam atom Rubidium⁴ yang dikelola dibawah Sub Bidang Seismologi Teknik, Geofisika Potensial, dan Tanda Waktu yang merupakan bagian dari Deputi Geofisika. Bidang inilah yang berperan aktif dalam memberikan informasi tanda waktu yang berkaitan dengan Matahari dan Bulan, yang berupa penentuan awal bulan kamariah, gerhana, jam atom serta banyak informasi lainnya.

³ Hasanuddin Z Abidin, *Geodesi...*, 61.

⁴ Jam atom Rubidium berfungsi sebagai *backup* jika pada suatu saat jam atom Cesium mengalami kendala atau *error*. Wawancara Pak Devid Bidang Seismologi Teknik, Geofisika Potensial, dan Tanda Waktu. 20 Mei 2019 pukul 12:06

Gambar 3.1 Jam Atom Cesium 4310 milik BMKG



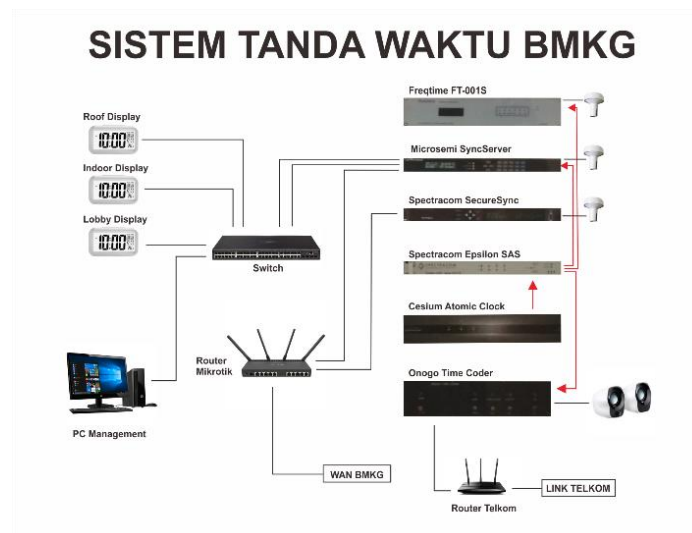
Jam atom BMKG disiarkan melalui jaringan *Local Area Network* (LAN)⁵ dan *Wide Area Network* (WAN)⁶. Jaringan LAN digunakan untuk keperluan lokal di area kantor BMKG, sedangkan jaringan WAN digunakan untuk menyiarkan jam atom kepada pihak-pihak terkait yang jangkauannya lebih luas. BMKG sendiri memanfaatkan WAN dalam hal ini melalui website resminya untuk menyiarkan jam atom melalui link <http://jam.bmkg.go.id/Jam.BMKG>. Melalui jaringan inilah aplikasi *AAT for*

⁵ LAN dapat didefinisikan sebagai network atau jaringan sejumlah system komputer yang lokasinya terbatas didalam satu gedung, satu kompleks gedung atau suatu kampus dan tidak menggunakan media fasilitas komunikasi umum seperti telepon, melainkan pemilik dan pengelola media komunikasinya adalah pemilik LAN itu sendiri dari definisi diatas dapat kita ketahui bahwa sebuah LAN dibatasi oleh lokasi secara fisik. Lihat Supriyanto, *Jaringan Dasar 1*, (Jakarta: Kementerian Pendidikan & Kebudayaan, 2013), 8.

⁶ WAN (Wide Area Network) merupakan jaringan komputer yang mencakup area yang besar sebagai contoh yaitu jaringan komputer antar wilayah, kota atau bahkan negara, atau dapat didefinisikan juga sebagai jaringan komputer yang membutuhkan router dan saluran komunikasi publik. Lihat Supriyanto, *Jaringan Dasar...*, 14.

Salat mendapatkan data jam atom BMKG. Secara lebih rinci, sistem tanda waktu BMKG akan digambarkan melalui skema berikut :⁷

Gambar 3.2 Skema Sistem Tanda Waktu BMKG



B. Algoritma Hisab Waktu Salat

Perhitungan waktu salat pada aplikasi ini menggunakan metode Ephemeris dengan algoritma Jean Meus. Algoritma ini sebenarnya merupakan reduksi dari perhitungan VSOP87⁸ yang lengkap. Dari ribuan suku koreksi dalam algoritma VSOP87 Jean Meus hanya mengambil beberapa ratus suku koreksi saja. Ia hanya mengambil suku-suku koreksi yang dinilai besar dan penting dan membuang suku-suku koreksi yang

⁷ Wawancara Pak Devid Bidang Seismologi Teknik, Geofisika Potensial, dan Tanda Waktu. 20 Mei 2019 pukul 12:06

⁸ VSOP87 atau Variations Seculaires des Orbites Planetaires merupakan teori lintasan planet-planet yang dipublikasikan oleh P.Bretagnon dan G Franco di Bureau des Longitudes, Paris pada tahun 1987. VSOP87 merupakan revisi dari VSOP82, karena pada perhitungan VSOP82 tidak mencantumkan koreksi-koreksi yang bisa ditinggalkan untuk perhitungan *full accuracy*. Total jumlah koreksi pada VSOP87 sebanyak 2425 buah; 1080 koreksi untuk bujur ekliptika, 348 untuk koreksi lintang ekliptika dan 997 koreksi untuk jarak Matahari dan Bumi. Lihat Jean Meus, *Astronomical Algorithms*, (Vigrinia: Willmann-Bell, Inc), 205.

kurang penting.⁹ Meskipun demikian algoritma Jean Meus mampu menghasilkan data posisi Matahari yang akurat. Dengan tingkat kesalahan tidak lebih dari 1 detik bujur dalam rentang waktu antara tahun -2000 hingga +6000 (sekitar 8000 tahun).¹⁰ Adapun alur proses perhitungan algoritma secara lengkap akan dijabarkan sebagai berikut :

1. Algoritma Perhitungan *Julian Day*, Selisih *Universal Time* dan *Dinamical Time* (ΔT), *Julian Day Ephemeris*, *Julian Centuries*, dan *Jullian Millenia*.

a. Algoritma Perhitungan Julian Day

Input data yang diperlukan untuk perhitungan data *ephemeris* yakni data tanggal dan waktu. Data tanggal berisi tanggal, bulan dan tahun. Sedangkan data waktu berisi jam, menit dan detik. Sebelum menghitung pertama-tama data tanggal harus diubah ke dalam sistem Julian. Sedangkan data waktu dirubah menjadi GMT. Untuk mengubah data waktu cukup mudah karena tinggal mengurangi atau menambahkan ke waktu GMT. Seperti Waktu Indonesia Barat GMT +7 maka WIB -7 sama dengan GMT (misal 09.00 WIB = 02.00 GMT). Sedangkan untuk konversi tanggal menjadi Julian Day diperlukan rumus sebagai berikut:¹¹

Jika bulan > 2 , maka $M = \text{bulan}$, dan $Y = \text{tahun}$

Jika bulan = 1 atau 2, maka $M = \text{bulan}+12$, dan $Y = \text{tahun}-1$

⁹ Rinto Anugraha, *Mekanika...*, 68.

¹⁰ Jean Meus, *Astronomical...*, 154.

¹¹ *Ibid.*, 61.

$$A = \text{INT} (Y/100)$$

$$B = 2-A+\text{INT}(A/4)$$

$$\text{JD} = \text{INT} (365,25* (Y+4716)) + \text{INT}(30,6001*(M+1)) + B + \\ \text{tanggal}+(\text{jam}+\text{menit}/60+\text{detik}/3600)/24-1524,5$$

b. Algoritma Perhitungan ΔT

Langkah selanjutnya adalah menghitung ΔT , terdapat banyak macam rumus yang diperkenalkan oleh Jean Meus untuk mencari ΔT . Sebagaimana telah diketahui bahwa nilai ΔT dari tahun ke tahun memiliki perubahan. Sehingga beberapa rumus yang terdapat dalam buku *Astronomical Algorithms* milik Jean Meus perlu adanya penyesuaian.¹² Adapun rumus terbaru yang dikenalkan oleh Jean Meus adalah rumus yang dimuat di website resmi NASA mengenai logika *polynomial expression for ΔT* .¹³

Rumus *polynomial expression for ΔT* merupakan rumus yang disusun oleh Jean Meus dan Fred Espenak. Rumus ini disusun untuk mengatasi masalah pada hasil perhitungan beberapa rumus sebelumnya. Adapun cakupan perhitungan dari rumus ini adalah tahun -1999 s/d +3000 atau sekitar lima milenium. Berikut adalah logika perhitungan *polynomial expression for ΔT* :

¹² *Ibid.*, 71-75

¹³ <https://eclipse.gsfc.nasa.gov/SEcat5/deltatpoly.html> diakses 25 juni 2019 pukul 22:39

Tahun 2005 s/d 2050

$$\Delta T = 62,92 + 0,32217 * t + 0,005589 * t^2$$

di mana: $t = y - 2000$

Logika diatas digunakan untuk mengoreksi perhitungan delta T pada tahun 2005 sampai dengan 2050. Rumus ini digunakan dalam perhitungan pada aplikasi ini. Karena pada aplikasi ini digunakan logika perhitungan otomatis menyesuaikan tanggal tidak bisa menghitung waktu yang telah lampau.

2050 s/d 2150

$$\Delta T = -20 + 32 * ((y-1820) / 100)^2 - 0,5628 * (2150 - y)$$

Logika diatas digunakan untuk rentang waktu 2050 hingga 2150. Perhitungan ini memungkinkan aplikasi masih bisa digunakan hingga 2150

Setelah 2150, maka :

$$\Delta T = -20 + 32 * u^2$$

di mana: $u = (y-1820) / 100$

- c. Menghitung *Julian Day Ephemeris*, *Julian Centuries* (T), dan *Julian Millenia* (T)¹⁴

Langkah selanjutnya adalah menghitung JDE, T dan T_J, berikut rumus yang dipergunakan:

¹⁴ Jean Meus, *Astronomical...*, 71-75.

$$\text{JDE} = \text{JD} + \Delta T$$

$$T \text{ (TD)} = (\text{JDE} - 2452545) / 36525$$

$$\mathcal{T} = T / 10^{15}$$

2. Algoritma Perhitungan Data *Ephemeris* Matahari

a. Koreksi Posisi Planet Bumi

Untuk mengetahui data *ephemeris* Matahari, sebelumnya dilakukan perhitungan posisi orbit planet Bumi dari Matahari. Terdapat 3 data posisi untuk masing-masing planet, yakni: L untuk bujur heliosentris planet, B untuk lintang heliosentris planet dan R untuk jarak planet-Matahari.¹⁶ Untuk perhitungan data posisi planet Bumi terdapat 6 kelompok koreksi untuk data L Bumi, 2 kelompok koreksi untuk data B Bumi dan 5 kelompok koreksi untuk data R Bumi. Data tabel koreksi *Periodic Terms* untuk planet Bumi dapat dilihat pada lampiran 6. Adapun untuk menggunakan data koreksi pada tabel tersebut menggunakan rumus:

$$A * \text{Cos}(B + C * \mathcal{T})$$

Koreksi pertama yakni koreksi bujur heliosentris Bumi (L), terdapat 6 kelompok koreksi pada tahap ini, yakni L0, L1, L2, L3, L4 dan L5. Di mana pada masing-masing kelompok koreksi terdapat suku-suku koreksi dengan jumlah yang berbeda-beda. Masing-masing suku koreksi tersebut dihitung menggunakan rumus yang

¹⁵ Rinto Anugraha, *Mekanika...*, 64.

¹⁶ Jean Meus, *Astronomical...*, 205.

telah disebutkan di atas. Selanjutnya suku-suku koreksi tersebut dijumlahkan pada masing-masing kelompok koreksi.

Terakhir, untuk mendapatkan bujur heliosentris Bumi digunakan rumus:

$$L = (L_0 + L_1 * T + L_2 * T^2 + L_3 * T^3 + L_4 * T^4 + L_5 * T^5) /$$

108

L merupakan bujur ekliptika Bumi diukur dari Matahari (heliosentris). Adapun untuk mengetahui bujur ekliptika Matahari dari Bumi (geosentris) menggunakan rumus berikut:

$$\Theta = L + 180, \text{ hasil dalam derajat.}$$

Setelah itu, menghitung koreksi untuk Θ dengan rumus:

$$\Theta_t = \Theta - 0,09033'', \text{ hasil dalam derajat.}$$

Hasilnya merupakan *true geometric longitude* matahari.

Koreksi kedua adalah koreksi lintang heliosentris Matahari (B), terdapat 2 kelompok koreksi pada tahap ini, yakni B0 dan B1. Proses koreksinya sama dengan koreksi L. Selanjutnya untuk mengetahui B menggunakan rumus berikut:

$$B = ((B_0 + B_1 * T) / 108) * 1, \text{ hasil dalam bentuk radian,}$$

kemudian dikonversi ke dalam detik busur.

Setelah itu menghitung koreksi B dengan rumus:

$$\lambda' = \Theta - 1,397 * T - 0,00031 * T^2$$

$\Delta B = 0,03916 * (\cos \lambda' - \sin \lambda')$, di mana untuk pemrograman java, λ' dirubah ke dalam bentuk desimal terlebih dahulu.

Selanjutnya untuk mengetahui *apparent latitude* Matahari menggunakan rumus:

$$\beta = B + \Delta B$$

Koreksi ketiga adalah koreksi jarak Matahari-Bumi (R). Terdapat 5 kelompok koreksi. Adapun proses koreksinya sama dengan koreksi L dan B. Selanjutnya untuk mengetahui *true geocentric distance* menggunakan rumus:

$$R = (R_0 + R_1 * T + R_2 * T^2 + R_3 * T^3 + R_4 * T^4) / 108$$

b. Algoritma Perhitungan *Nutasi* dan *True Obliquity*

Langkah pertama yakni menghitung *mean obliquity* dengan rumus:

$$E_0 = 23^{\circ} 26' 21,448'' + (-4680,93 * U - 1,55 * U^2 + 1999,25 * U^3 - 51,38 * U^4 - 249,67 * U^5 - 39,05 * U^6 + 7,12 * U^7 + 27,87 * U^8 + 5,79 * U^9 + 2,45 * U^{10}) / 3600, \text{ hasil dalam derajat.}$$

$$\text{Di mana } U = T/100$$

Untuk mengetahui *true obliquity* dan *nutasi*, diperlukan perhitungan koreksi $\Delta\epsilon$ dan $\Delta\psi$ menggunakan tabel *terms of the 1980 IAU Theory of Nutations*, sebagaimana terdapat pada lampiran 7 dan 8. Namun sebelum menghitung $\Delta\psi$ dan $\Delta\epsilon$ dengan tabel tersebut, diperlukan perhitungan *multiple arguments* terlebih dahulu. Berikut rumus perhitungan *multiple arguments* untuk $\Delta\psi$ dan $\Delta\epsilon$:

$$D = 297,85036 + 445267,11148 * T - 0,0019142 * T^2 + T^3/189474, \text{ hasil dalam derajat.}$$

$$M_o = 357,52772 + 35999,05034 * T - 0,0001603 * T^2 - T^3/300000, \text{ hasil dalam derajat.}$$

$$M_c = 134,96298 + 477198,867398 * T + 0,0086972 * T^2 + T^3/56250, \text{ hasil dalam derajat.}$$

$$F = 93,27191 + 483202,017538 * T - 0,0036825 * T^2 + T^3/327270, \text{ hasil dalam derajat.}$$

$$\Omega_c = 125,04452 - 1934,136261 * T + 0,0020708 * T^2 + T^3/450000, \text{ hasil dalam derajat.}$$

Langkah selanjutnya adalah menghitung koreksi $\Delta\psi$ pada tabel yang terdapat pada lampiran 7 menggunakan rumus:

$$(\text{Coefficient 1} + \text{Coefficient 2} * T) * \sin (\text{Multiple Arguments})$$

Setelah itu seluruh koreksi dijumlahkan, kemudian digunakan untuk menghitung *Nutasi* ($\Delta\psi$).

$$\Delta\psi = \text{jumlah koreksi}/10000/3600, \text{ hasil dalam derajat.}$$

Selanjutnya algoritma perhitungan berpindah ke tabel koreksi $\Delta\epsilon$ pada lampiran 8. Koreksi dihitung menggunakan rumus:

$$(\text{Coefficient 1} + \text{Coefficient 2} * T) * \cos (\text{Multiple Arguments})$$

Terakhir, yakni menghitung *apparent longitude* Matahari berdasarkan hasil perhitungan di atas dengan algoritma:

$\lambda = \Theta t + \Delta\psi + \text{Aberasi}$, hasil dalam bentuk derajat.

- c. Menghitung *Right Ascension* dan *Apparent Declination* (Deklinasi Matahari)¹⁷

Rumus *apparent right ascension*:

$\alpha = \text{Atan}((\sin \lambda * \cos \varepsilon - \tan \beta * \sin \varepsilon) / \cos \lambda)$, hasil dalam derajat.

Rumus *apparent declination*:

$\sin \delta = \sin \beta * \cos \varepsilon + \cos \beta * \sin \varepsilon * \sin \lambda$, hasil dalam derajat.

- d. Algoritma Perhitungan *Equation of Time*¹⁸

Sebelum menghitung *equation of time*, terlebih dahulu dilakukan perhitungan bujur rata-rata Matahari menggunakan rumus:

$$L_0 = 280,4664567 + 360007,6982779 * T - 0,03032028 * T^2 + T^3 / 49931 - T^4 / 15299 - T^5 / 1988000$$
, hasil dalam derajat dan di *modulus*-kan.

Rumus perhitungan *equation of time*:

$E_q = L_0 - 0,0057183 - \alpha + \Delta\psi * \cos \varepsilon$, hasil dalam menit waktu.

¹⁷ Jean Meus, *Astronomical...*, 89.

¹⁸ *Ibid.*, 171-175.

e. Algoritma Perhitungan Semi Diameter Matahari

$$Sd = 15' 59,63'' / R , \text{ hasil dalam derajat.}$$

3. Algoritma Perhitungan Waktu Salat

a. Waktu Zuhur

$$\text{Zuhur} = \text{pukul 12 Waktu Hakiki (WH)}$$

$$\text{WIB} = \text{WH} - e + (\lambda^d - \lambda^x) : 15^{19}$$

b. Waktu Asar

zm (jarak zenith) = $|\delta^m - \Phi^x|$ jarak antara zenit dan matahari seharga harga lintang mutlak lintang tempat dikurangi deklinasi matahari.

ha (tinggi matahari pada awal Ashar)

$$\text{Cotan ha} = \text{Tan zm} + 1$$

to (sudut waktu matahari) awal Ashar

$$\text{Cos to} = \text{Sin ha} : \text{Cos } \Phi^x : \text{Cos } \delta^m - \text{Tan } \Phi^x \times \text{Tan } \delta^m$$

$$\text{Asar} = \text{Pukul 12.00} + \text{to}^{20}$$

c. Waktu Magrib

ho (tinggi matahari) saat terbit/terbenam

to (sudut waktu matahari) awal Maghrib

$$\text{Cos to} = \text{Sin ho} : \text{Cos } \Phi^x : \text{Cos } \delta^m - \text{Tan } \Phi^x \times \text{Tan } \delta^m$$

$$\text{Asar} = \text{Pukul 12.00} + \text{to}^{21}$$

d. Waktu Isya

ho (tinggi matahari) untuk awal Isya

¹⁹ Ahmad Izzuddin, *Ilmu Falak...*, 85.

²⁰ *Ibid.*, 86.

²¹ *Ibid.*, 87.

t_o (sudut waktu matahari) awal Isya'

$$\cos t_o = \sin h_o : \cos \Phi^x : \cos \delta^m - \tan \Phi^x \times \tan \delta^m$$

$$\text{Isya} = \text{Pukul 12.00} + t_o^{22}$$

e. Waktu Subuh

h_o (tinggi matahari) untuk awal Shubuh

t_o (sudut waktu matahari) awal Shubuh

$$\cos t_o = \sin h_o : \cos \phi^x : \cos \delta^m - \tan \phi^x \times \tan \delta^m$$

$$\text{Isya} = \text{Pukul 12.00} + t_o$$

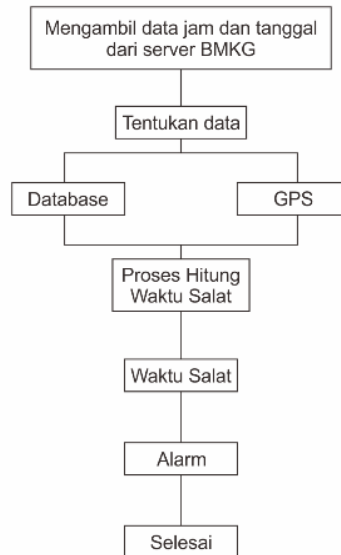
4. Diagram Alur Perhitungan Waktu Salat

Dalam skripsi ini penulis menyusun sebuah aplikasi waktu salat berbasis android yang menggunakan standar jam atom milik BMKG. Antarmuka pada aplikasi ini didesain dengan konsep *user friendly* sehingga pengguna dapat dengan mudah menggunakannya. Pengguna hanya perlu memasukkan data input berupa koordinat kota yang telah tersedia di database, tinggi tempat dan zona waktu. Atau dapat memilih input otomatis menggunakan bantuan GPS.

Agar dapat dengan mudah memahami alur perhitungannya berikut penulis berikan gambarannya:

²² *Ibid.*, 88.

Gambar 3.3 Diagram Alur Aplikasi



Dari gambar diatas dapat dilihat mengenai bagaimana aplikasi ini berjalan dan melakukan perhitungan. Setelah pengguna melengkapi data-data yang diperlukan di halaman utama maka sistem akan secara otomatis menghitungnya dan menampilkan dihalaman selanjutnya.

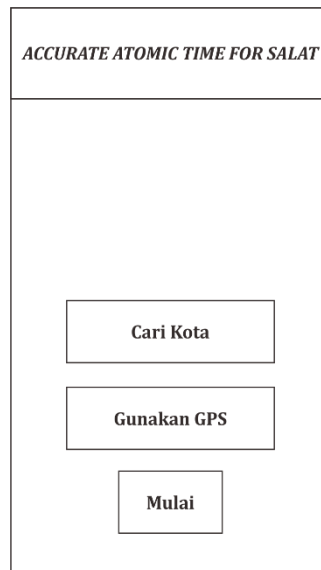
C. Aplikasi *Accurate Atomic Time for Salat*

1. Desain Utama dan Spesifikasi Perangkat

a. Desain Antarmuka (User Interface)

Desain antarmuka pada aplikasi *Accurate Atomic Time for Salat* dibuat dengan sesimpel mungkin agar bisa digunakan dengan mudah. Didesain menggunakan software grafis dengan menerapkan kaidah-kaidah desain grafis sehingga mampu menghasilkan tampilan yang menarik.

Gambar 3.4 Tampilan Utama Menu Aplikasi



1. Cari Kota

Pada *activity* ini akan menampilkan data koordinat kota secara offline yang merupakan data kota di seluruh Indonesia yang bersumber dari BIG (Badan Informasi Geospasial) *activity* ini dibuat secara otomatis, sehingga pengguna dapat dengan mudah mencari kota yang dimaksud. Pada *activity* ini juga akan muncul kolom tinggi tempat dan zona waktu yang dapat diisi secara manual.

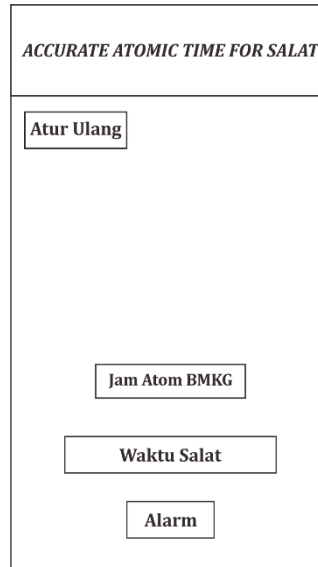
2. Gunakan GPS

Activity ini merupakan opsi untuk mencari koordinat tempat dengan memanfaatkan GPS (*Global Positioning System*) sehingga perangkat harus tersambung dengan koneksi internet.

3. Mulai

Tombol mulai merupakan tombol untuk melanjutkan *activity* pada tampilan utama ke tampilan selanjutnya.

Gambar 3.5 Tampilan Hasil Perhitungan Waktu Salat



4. Jam Atom BMKG

Activity ini menampilkan jam yang langsung terhubung dengan server jam atom BMKG <http://jam.bmkg.go.id/>.

5. Waktu Salat

Activity ini menampilkan hasil perhitungan waktu salat yang dihitung menggunakan metode *ephemeris*.

6. Alarm

Activity ini merupakan tombol yang berfungsi untuk mengatur alarm waktu salat yang akan mengeluarkan suara azan saat waktu salat tiba.

7. Atur Ulang

Activity ini merupakan tombol yang berfungsi untuk mengatur ulang program dan kembali ke halaman awal untuk input data.

b. Spesifikasi Perangkat Keras

Pada perancangan aplikasi *Accurate Atomic Time for Salat* menggunakan beberapa perangkat keras atau *hardware* dan perangkat lunak atau *software* untuk menyajikan aplikasi tersebut. Berikut adalah beberapa perangkat keras yang digunakan :

1. Gawai dengan Sistem Operasi Android

Gawai yang digunakan dalam proses pengembangan aplikasi *Accurate Atomic Time for Salat* adalah Xiaomi Redmi 4X 2/16 dengan operasi sistem android Nougat (7.1.2). Gawai ini juga yang digunakan untuk menguji aplikasi *Accurate Atomic Time for Salat*.

2. Laptop Dell Inspiron 5458

Laptop Dell Inspiron 5458 digunakan untuk merancang aplikasi *Accurate Atomic Time for Salat* dengan spesifikasi Intel (R) Core(TM) i3-5005U CPU @ 2.00GHz 2.00GHz, 4.00 RAM

c. Spesifikasi Perangkat Lunak

Adapun perangkat lunak atau *software* yang digunakan dalam proses perancangan aplikasi *Accurate Atomic Time for Salat* adalah sebagai berikut :

1. *Java Development Kit (JDK)* yang berfungsi untuk mengkompilasikan kode-kode java menjadi aplikasi java, serta *Java Runtime Environment (JRE)* yang berfungsi sebagai komponen utama yang bertugas untuk menjalankan aplikasi berbasis java.
2. *Android Studio (3.3)* merupakan *software* yang digunakan untuk melakukan *coding* aplikasi android, desain serta perhitungan yang dibutuhkan untuk pemrograman aplikasi AAT for Salat.
3. *Android Software Development Kit (Android SDK)* yang berfungsi sebagai alat pengembangan yang komperhensif. *Android SDK* berisi *debugger, libraries, handset emulator, dokumentasi, contoh kode, dan tutorial.*
4. *Windows 8.1 Pro* sebagai sistem operasi yang digunakan untuk *menginstall* perangkat lunak yang diperlukan dalam perancangan aplikasi *Accurate Atomic Time for Salat.*
5. *Corel Draw X7* merupakan software grafis berbasis vektor yang digunakan untuk merancang tampilan grafis *user interface (UI)* atau antarmuka yang mencakup *layout (tata letak), visual design* dan *branding* sehingga aplikasi mempunyai tampilan yang menarik dan dapat digunakan dengan mudah.

D. Implementasi Aplikasi *Acurate Atomic Time for Salat*

1. Implementasi Perhitungan Aplikasi

Setelah perancangan desain aplikasi *Acurate Atomic Time for Salat* sebagaimana telah dijelaskan diatas, bagian selanjutnya adalah implementasi perhitungan waktu salat ke bahasa pemrograman *java* menggunakan software Android Studio. Adapun coding bahasa pemrograman untuk aplikasi *Acurate Atomic Time for Salat* adalah sebagai berikut :

a. Implementasi Perhitungan Deklinasi

```
double deklinasi = 180 / Math.PI * (Math.asin(prodekm));
double elnol = MOD((280.4664567 + 360007.6982779 * tautau +
0.03032028
* Math.pow(tautau, 2) + Math.pow(tautau, 3) / 49931 -
Math.pow(tautau,4)/15299- Math.pow(tautau,5) / 1988000), 360);
```

b. Implementasi Perhitungan Equation of Time

```
double equationo = (elnol - 0.0057183 - alpha2 + segpe *
Math.cos(Math.PI / 180 * (segep)));
Double EoT;
if (equationo > 100) {
    EoT = (-360 + equationo) * 4;
} else {
```

```
EoT= equationo * 4;
```

```
}
```

c. Implementasi Waktu Zuhur

```
double dhuhur = 12 - EoT / 60 + ((tz * 15) - bt) / 15;
```

```
double zm = Math.abs(deklinasi - lt);
```

```
double ha = 180 / Math.PI * Math.atan((1 / (Math.tan(zm *  
Math.PI / 180) + 1)));
```

```
double sudutwaktu = (180 / Math.PI * Math.acos(Math.sin(ha *  
Math.PI / 180) / Math.cos(lt * Math.PI / 180) /
```

```
Math.cos(deklinasi * Math.PI / 180) - Math.tan(lt
```

```
* Math.PI / 180) * Math.tan(deklinasi * Math.PI / 180))) /
```

```
15;
```

d. Implementasi Waktu Asar

```
double ashar = 12 + sudutwaktu - EoT / 60 + (105 - bt) / 15;
```

```
double refrak = 0.575;
```

```
double semidi = 0.2667;
```

```
double kerendah = 0.02933333333 * Math.sqrt(tg);
```

```
double tinggimatahari = -(refrak + semidi + kerendah);
```

```
double sudutmataharimaghrib = (180 / Math.PI *  
Math.acos(Math.sin(tinggimatahari * Math.PI / 180) / Math.cos(lt
```

```
* Math.PI / 180)
```

```
/ Math.cos(deklinasi * Math.PI / 180) - Math.tan(lt *
```

```
Math.PI / 180) * Math.tan(deklinasi * Math.PI / 180))) / 15;
```

e. Implementasi Waktu Magrib

```

maghrib = 12 + sudutmataharimaghrib - EoT / 60 + (105 - bt) /
15;

double tinggimatahariisya = -17.0 + tinggimatahari;

double sudutmatahariisya = (180 / Math.PI *
Math.acos(Math.sin(tinggimatahariisya * Math.PI / 180) /
Math.cos(lt * Math.PI / 180) / Math.cos(deklinasi * Math.PI /
180) - Math.tan(lt * Math.PI / 180) * Math.tan(deklinasi *
Math.PI / 180))) / 15;

```

f. Implementasi Isya

```

isya = 12 + sudutmatahariisya - EoT / 60 + (105 - bt) / 15;

double tinggimataharishubuh = -19.0 + tinggimatahari;

double sudutmataharishubuh = (180 / Math.PI *
Math.acos(Math.sin(tinggimataharishubuh * Math.PI / 180) /
Math.cos(lt * Math.PI / 180) / Math.cos(deklinasi * Math.PI /
180) - Math.tan(lt * Math.PI / 180) * Math.tan(deklinasi *
Math.PI / 180))) / 15;

```

g. Implementasi Subuh

```

double shubuh = 12 - sudutmataharishubuh - EoT / 60 + (105 - bt)
/ 15;

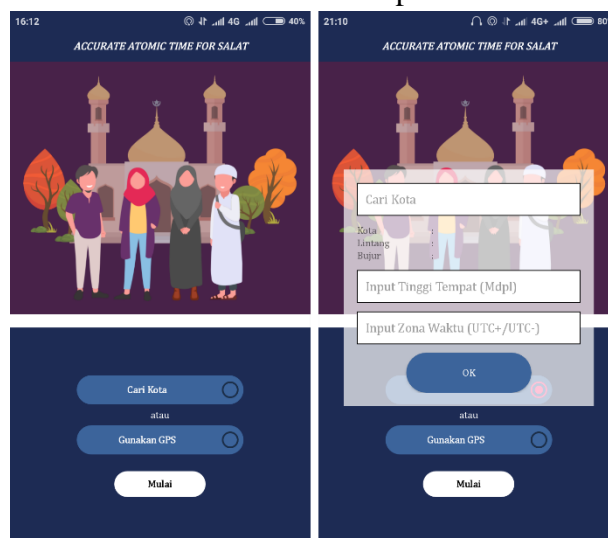
```

2. Implementasi Antarmuka Aplikasi

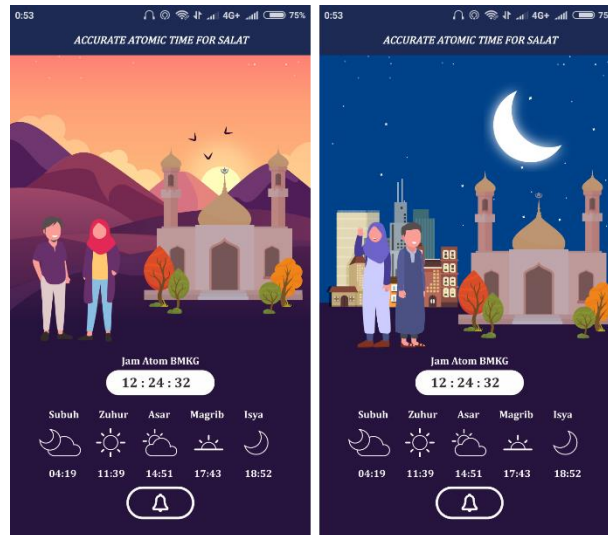
Pada bagian ini akan dijelaskan mengenai implementasi desain antarmuka dari aplikasi *Accurate Atomic Time for Salat* melalui bahasa pemrograman di Android Studio. Aplikasi ini merupakan aplikasi perhitungan waktu salat yang dikombinasikan dengan jam Atom milik BMKG. Karena jam Atom BMKG disiarkan melalui jaringan WAN, maka untuk mengambil data itu diperlukan koneksi internet.

Pengguna hanya perlu menginput data yang diperlukan, atau dapat secara otomatis memilih dengan bantuan GPS. Semua proses perhitungan akan dihitung melalui *activity* dibalik layar. Pengguna hanya dapat melihat hasil perhitungan waktu salat dan jam atom BMKG. Adapun implementasinya adalah sebagai berikut :

Gambar 3.6 Desain Antarmuka pada Tampilan Awal dan Halaman Input



Gambar 3.7 Desain Antarmuka pada Tampilan Utama Waktu Salat



Gambar 3.5 diatas merupakan tampilan antarmuka yang akan tampil saat pengguna membuka aplikasi *Accurate Atomic Time for Salat* saat pertama kali, sedangkan disampingnya adalah tampilan input dimana pengguna diminta untuk memasukkan nama kota, tinggi tempat dan zona waktu.

Adapun Gambar 3.6 adalah tampilan antar muka utama yaitu tampilan hasil dari perhitungan waktu salat serta tampilan jam atom BMKG. pada tampilan ini juga terdapat tombol alarm yang berfungsi mengaktifkan alarm berupa suara azan untuk setiap waktu salat. Jika ingin mengatur ulang aplikasi ini maka tinggal menekan tombol atur ulang pada pojok kiri atas.

BAB IV

UJI COBA DAN EVALUASI APLIKASI *ACCURATE ATOMIC TIME FOR SALAT*

A. Uji Fungsional Aplikasi *Accurate Atomic Time for Salat*

Pada bab ini penulis akan melakukan uji sistem aplikasi *Accurate Atomic Time for Salat*. Pengujian sistem merupakan proses pengujian perangkat lunak apakah mempunyai sistem yang cocok dengan spesifikasi sistem atau tidak. Pengujian sistem sering dikenal dengan istilah pencarian *bug*, yaitu pencarian ketidaksempurnaan program, kesalahan pada program yang menyebabkan kegagalan saat menjalankan perangkat lunak.

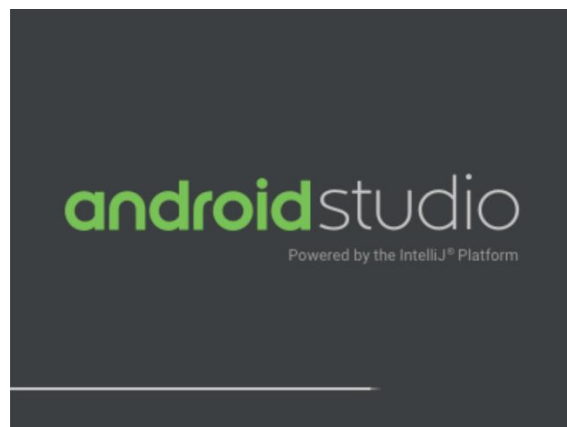
Pengujian ini dilakukan dengan menguji setiap proses dan kemungkinan terjadinya kesalahan pada setiap prosesnya. Adapun proses pengujian akan dibagi menjadi dua proses. Pengujian pertama adalah *White Box*, yaitu pengecekan terhadap detail perancangan menggunakan struktur kontrol dari desain program yang telah ditentukan sebelumnya. Sedangkan pengujian kedua adalah *Black Box* yaitu pengecekan terhadap detail aplikasi seperti tampilan aplikasi, fungsi-fungsi pada aplikasi dan kesesuaian alur fungsi yang ada pada aplikasi. Pengujian ini tidak melihat dan menguji *coding*.

Selain pengujian pada *coding* dan antarmuka, pada bab inipun akan memuat analisis hasil perhitungan aplikasi *Accurate Atomic Time for Salat* dengan perbandingan waktu salat milik Bimas Islam Kementerian Agama RI.

Karena terbatasnya spesifikasi laptop yang digunakan maka pada proses uji fungsional ini penulis menggunakan metode *debugging usb* dengan menyambungkan laptop secara langsung ke gawai. Berikut langkah-langkahnya :

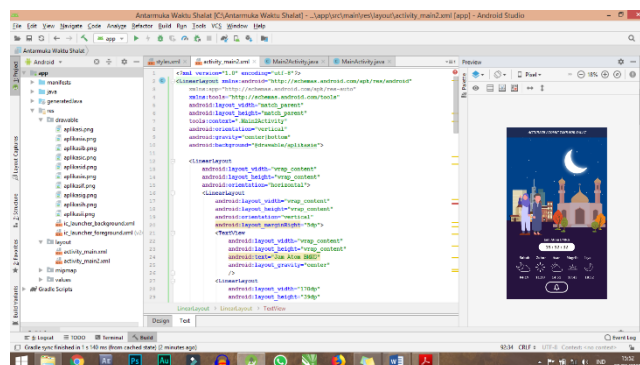
1. Jalankan *Software* Android Studio 3.3

Gambar 4.1 Tampilan *Loading* Android Studio



Setelah *software* Android Studio terbuka tunggu beberapa saat agar Android Studio bisa mengambil data-data yang telah kita susun. Setelah proses ini selesai, Android Studio akan menampilkan halaman kerja yang berisi *coding* dan halaman *activity layout*. Tunggu hingga semua proses selesai ditandai dengan munculnya *layout* antarmuka aplikasi.

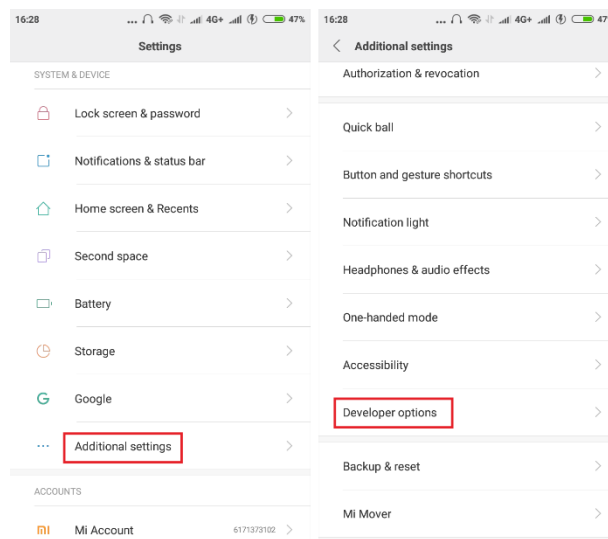
Gambar 4.2 Tampilan Lembar Kerja Android Studio



Karena pada *debugging* ini menggunakan metode *usb debugging* maka diperlukan sebuah ponsel dengan pengaturan khusus. Dalam hal ini penulis menggunakan Xiaomi Redmi 4X

2. Atur gawai menjadi *developer option*

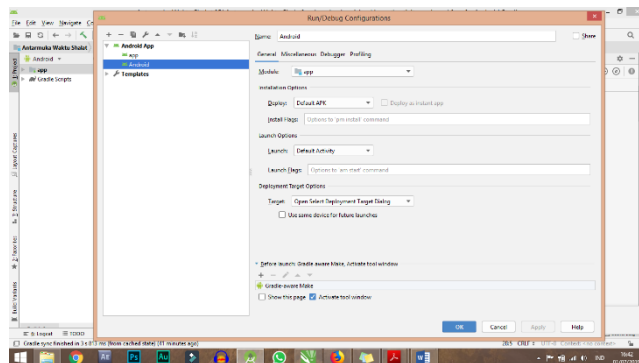
Gambar 4.3 Tampilan Setting Xiaomi Redmi 4X



Proses ini dilakukan agar gawai bisa digunakan untuk proses *debugging*. Pertama masuk ke halaman *Setting > Additional Setting > Developer option* pada halaman ini silahkan aktifkan *Debugging USB dan Install via USB*.

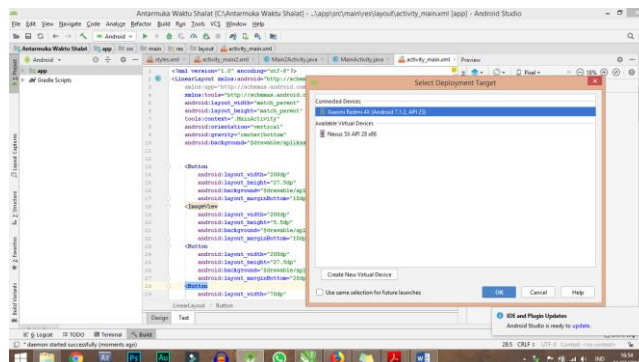
3. Sambungkan perangkat ke laptop menggunakan kabel USB
4. Buat konfigurasi untuk menjalankan projek. Proses konfigurasi ini dilakukan jika aplikasi benar-benar belum pernah dijalankan. Proses ini memerlukan beberapa tahapan *edit configuration* lalu pilih tambah untuk membuat konfigurasi baru, beri nama dan *module* yang sesuai

Gambar 4.4 Proses Konfigurasi



5. Jalankan proyek aplikasi dengan menekan tombol *Run* atau *Shift + F10*, maka akan muncul nama perangkat yang tersambung. Lalu pilih nama gawai yang dimaksud dan proses *debugging* dimulai.

Gambar 4.5 Proses Debugging



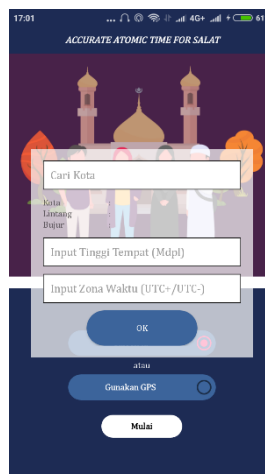
6. Setelah proses selesai maka akan tampil menu utama pada gawai. Pada tampilan ini kita dapat melihat tombol Cari Kota, Gunakan GPS dan Mulai.

Gambar 4.6 Tampilan Utama Aplikasi



7. Menguji fungsi tombol Cari Kota, Gunakan GPS dan Mulai. Saat pengguna menekan tombol Cari Kota maka akan muncul tampilan sebagai berikut:

Gambar 4.7 Tampilan Cari Kota

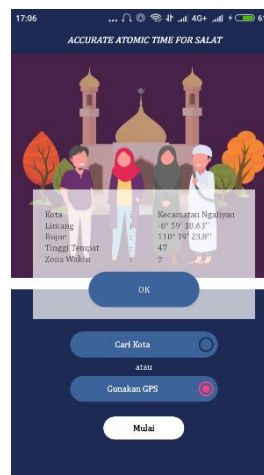


Pada halaman ini pengguna diminta untuk menginput data kota yang akan dihitung waktu shalatnya. Dalam aplikasi ini terdapat 511 data koordinat kota/kabupaten yang merupakan data milik Badan Informasi Geospasial. Pada halaman ini pula pengguna diminta untuk mengisi

tinggi tempat. Tinggi tempat ini menjadi opsional, bagi yang ingin menggunakannya maka cukup mengisikan tinggi tempat dalam satuan meter. Namun jika tidak, pengguna tinggal memasukkan angka nol pada tinggi tempat. Setelah itu, pengguna diminta memasukkan zona waktu tempat dimana waktu salat itu dihitung.

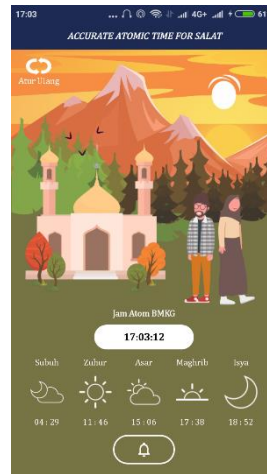
Selain menggunakan input data yang telah disediakan aplikasi berupa 511 koordinat kota/kabupaten, pengguna pun dapat menggunakan opsi lain yaitu dengan menggunakan bantuan GPS. Pengguna hanya perlu menekan tombol Gunakan GPS maka sistem akan secara otomatis mencari koordinat. Sebagai catatan penggunaan GPS akan jauh lebih baik jika digunakan diluar ruangan agar gawai bisa dengan mudah menemukan sinyal.

Gambar 4.8 Tampilan Gunakan GPS



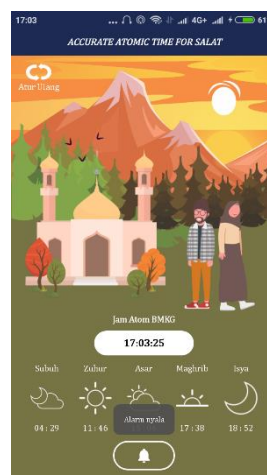
Jika sudah memilih salah satu dari dua metode input data maka aplikasi akan menghitung waktu salat di belakang layar dan akan ditampilkan pada halaman selanjutnya

Gambar 4.9 Halaman Hasil Hitung Waktu Salat



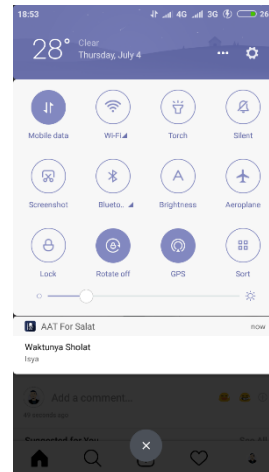
Pada halaman ini ditampilkan hasil hitungan waktu salat dan jam yang secara langsung terhubung dengan jam atom BMKG. selain itu ada pula tombol alarm yang berfungsi memberikan tanda berupa azan setiap memasuki awal waktu salat. Alarm yang aktif ditandai dengan tombol lonceng yang berubah dari yang semula hanya garis menjadi gambar penuh dan juga akan muncul pemberitahuan “Alarm nyala”.

Gambar 4.10 Tampilan Pengaturan Alarm



Alarm ini berfungsi untuk memberikan pemberitahuan jika sudah memasuki awal waktu salat. Sistem akan memberikan pemberitahuan berupa notifikasi dan suara azan jika waktu salat sudah tiba.

Gambar 4.11 Notifikasi Saat Memasuki Awal Waktu Salat



Jika ingin merubah koordinat tempat maka pengguna hanya perlu menekan tombol atur ulang dan aplikasi akan mengembalikan pengguna pada halaman awal untuk mengisi data input koordinat kota yang ingin dihitung waktu salatnya. Demikianlah proses *USB debugging* pada aplikasi *Accarute Atomic Time for Salat*.

Setelah melakukan *debugging*, jika dirasa cukup maka aplikasi bisa diexport menjadi file berjenis apk. Setelah proses *convert* selesai proses selanjutnya adalah memindahkan file apk dari laptop ke gawai. Proses ini dapat dilakukan dengan bantuan kabel USB atau dengan bantuan aplikasi tambahan seperti *Shareit* atau *AirDroid*.

Pada pengujian dengan *usb debugging* aplikasi dapat berjalan dengan baik. Untuk selanjutnya penulis menggunakan beberapa gawai

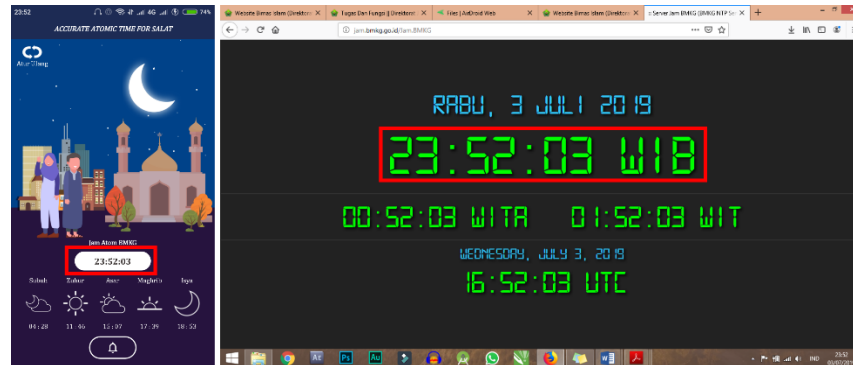
dengan berbagai macam *brand*, tipe dan versi android untuk melakukan pengujian aplikasi. Berikut rinciannya:

Nama dan Tipe Ponsel	Versi Android	Ukuran Layar	Memory
Samsung M20	Android 8.1 (Oreo)	2340 x 1080 (FHD+)	3 GB RAM
Realmi C1	Android 8.1 (Oreo)	720 x 1520 pixels	2 GB RAM
Xiaomi Redmi 4X	Android 6.0.1 (Marshmallow)	720 x 1280 pixels	2 GB RAM
Samsung A6	Android 8.1 (Oreo)	720 x 1480 pixels	3 GB RAM

Pada sub bab ini pula penulis mencoba menguji keserasian pengambilan data jam atom BMKG menggunakan aplikasi *Accurate Atomic Time for Salat* dibandingkan dengan perangkat lain berupa laptop. Pada pengujian ini diperlukan koneksi internet yang stabil agar data yang diambil bisa akurat. Karena stabilitas koneksi akan sangat berpengaruh pada data yang diambil dari server BMKG.¹ sebagai solusinya sistem harus senantiasa memperbarui pengambilan data dari NTP jam atom BMKG.

¹ Wawancara Pak Devid Bidang Seismologi Teknik, Geofisika Potensial, dan Tanda Waktu. 20 Mei 2019 pukul 12:06

Gambar 4.12 Tampilan Server NTP jam atom BMKG



B. Uji Verifikasi Hasil Perhitungan Aplikasi *Accurate Atomic Time for Salat*

Pada sub bab ini akan dibahas mengenai uji verifikasi hasil penghitungan waktu salat pada aplikasi *Accurate Atomic Time for Salat*. Uji verifikasi ini dilakukan untuk mengetahui apakah hasil penghitungan waktu salat pada aplikasi ini sudah tepat atau masih terjadi kesalahan, mengingat banyaknya fungsi-fungsi dan logika-logika rumus yang digunakan dalam perancangan aplikasi ini.

Adapun metode yang digunakan untuk memverifikasi hitungan ini dilakukan dengan membandingkan dengan hasil hitungan waktu salat *website* milik Direktorat Jendral Bimbingan Masyarakat Islam Kementerian Agama RI. *Website* ini dipilih karena Bimas Islam adalah lembaga resmi negara yang memiliki tugas merumuskan serta melaksanakan kebijakan dan standardisasi teknis di bidang bimbingan masyarakat Islam.² Termasuk

² <https://bimasislam.kemenag.go.id/site/profil/tugas-dan-fungsi> diakses 03 juli 2019

didalamnya memberikan layanan masyarakat untuk mengetahui jadwal waktu salat. Secara resmi waktu salat milik Bimas Islam ini dapat di akses melalui alamat <https://bimasislam.kemenag.go.id/jadwalshalat>.

Metode perhitungan yang digunakan *website* Bimas Islam dan aplikasi *Accurate Atomic Time for Salat* memiliki kemiripan. *Website* Bimas Islam menggunakan perhitungan astronomi modern dan menjadikan buku *Astronomical Alghorithm* milik *Jean Meus* sebagai acuan perhitungan serta menggunakan sistem *Ephemeris dalam* menghitung data Matahari.³ Selain itu *website* inipun menggunakan data Badan Informasi Geospasial sebagai acuan titik koordinat tengah kota.⁴ Hanya saja pada *website* ini faktor tinggi tempat tidak menjadi data yang dimasukkan kedalam perhitungan. Adapun pada aplikasi *Accurate Atomic Time for Salat* tinggi tempat menjadi opsi bagi pengguna yang merasa perlu memasukkan tinggi tempat. Berikut tabel perbandingan hasil perhitunga waktu salat:

Tabel 4.1 Perbandingan Waktu Salat Kota Semarang 25 Juni 2019

Salat	ATT for Salat Tinggi 0m	Bimas Islam	Selisih
Subuh	04:29	04:28	1 menit
Zuhur	11:45	11:44	1 menit
Asar	15:05	15:05	0 menit
Maghrib	17:35	17:36	1 menit
Isya	18:49	18:50	1 menit

³ Novi Arijatul Mufidoh, “Sistem Hisab Awal Waktu Salat Program Website Bimbingan Masyarakat Islam”, Skripsi Fakultas Syariah Dan Hukum UIN Walisongo Semarang, 2018, 55.

⁴ *Ibid.*, 54.

Tabel 4.2 Perbandingan Waktu Salat Kota Semarang 03 Juli 2019

Salat	ATT for Salat Tinggi 0m	Bimas Islam	Selisih
Subuh	04:31	04:30	1 menit
Zuhur	11:46	11:46	0 menit
Asar	15:07	15:06	1 menit
Maghrib	17:37	17:37	0 menit
Isya	18:51	18:51	0 menit

Dari hasil uji komparasi diatas, diketahui bahwa hasil perhitungan waktu salat *website* Bimas Islam dan *Aplikasi Accurate Atomic Time for Salat* dengan mengabaikan tinggi tempat untuk Kota Semarang pada 25 Juni 2019 dan 03 Juli 2019 hanya terpaut 1 menit. Hal ini membuktikan keduanya memiliki tingkat akurasi yang relatif sama. Perbedaan satu menit diperkirakan terjadi karena nilai *ikhtiyat* yang berbeda.

Untuk memastikan bahwa ketinggian tempat mempengaruhi hasil perhitungan jadwal waktu salat, penulis akan menjabarkan pula perbandingan setiap koordinat dengan menginput ketinggian berbeda-beda. Berikut perbandingannya:

Tabel 4.3 Perbandingan Waktu Salat Kota Semarang 03 Juli 2019 dengan Berbagai Varian Tinggi Tempat menggunakan Aplikasi AAT for Salat

Salat	Tinggi 0m	Tinggi 100m	Tinggi 200m	Tinggi 300m
Subuh	04:31	04:29	04:29	04:28
Zuhur	11:46	11:46	11:46	11:46

Asar	15:07	15:07	15:07	15:07
Maghrib	17:37	17:38	17:39	17:39
Isya	18:51	18:52	18:53	18:53

Tabel 4.4 Perbandingan Waktu Salat Kota Bandung 04 Juli 2019 dengan Berbagai Varian Tinggi Tempat menggunakan Aplikasi AAT for Salat

Salat	Tinggi 0m	Tinggi 200m	Tinggi 300m	Tinggi 400m
Subuh	04:42	04:40	04:39	04:39
Zuhur	11:57	11:57	11:57	11:57
Asar	15:18	15:18	15:18	15:18
Maghrib	17:48	17:50	17:50	17:51
Isya	18:02	18:04	18:04	18:05

Dari perbandingan tabel diatas membuktikan bahwa perbedaan tinggi tempat mempunyai pengaruh yang cukup signifikan pada penghitungan waktu salat. Perbedaan tinggi tempat masih bisa ditoleransi jika kota atau daerah yang dimaksud mempunyai permukaan yang relatif sama namun akan menjadi masalah jika dalam sebuah kota terdapat sebagian daerah yang dekat dengan laut dan sebagaian lainnya dekat dengan pegunungan.

C. Evaluasi Aplikasi *Accrurate Atomic Time for Salat*

Dari pemaparan uji fungsional dan uji verifikasi hasil hitungan waktu salat diatas dapat disimpulkan bahwa aplikasi *Accrurate Atomic Time for Salat* memiliki beberapa kelebihan dan kekurangan. Pemaparan

kelebihan diharapkan menjadi pacuan untuk pegiat falak lain agar bisa berinovasi, sedangkan kekurangannya dapat diperbaiki dikemudian hari.

Adapun kelebihan aplikasi *Accrurate Atomic Time for* adalah sebagai berikut:

1. Aplikasi dapat berjalan dengan baik pada berbagai macam ponsel pintar dengan berbagai macam versi Android.
2. Hasil perhitungan waktu salat merupakan hasil dengan tingkat akurasi yang tinggi. Karena proses perhitungannya menggunakan algoritma Jean Meus.
3. Terintegrasai secara langsung dengan server jam atom BMKG sehingga waktu yang dihasilkan akurat dan stabil.
4. Terdapat 511 data koordinat tengah kota/kabupaten di seluruh Indonesia yang bersumber dari Badan Informasi Geospasial.
5. Pengguna dapat memilih metode input data berupa koordinat dari Badan Informasi Geospasial atau otomatis dengan bantuan GPS.

Adapun kekurangannya adalah sebagai berikut:

1. ponsel harus terkoneksi dengan internet untuk mengambil data jam atom BMKG.
2. Ukuran aplikasi yang relatif besar 36 *megabyte*. Meskipun sebenarnya hal ini bisa dimaklumi karena banyaknya data koordinat tempat yang ada pada aplikasi ini.
3. Aplikasi hanya mempunyai fitur waktu salat sehingga pengguna hanya bisa mendapatkan satu manfaat dari aplikasi ini

BAB V

PENUTUP

A. Kesimpulan

Dari penelitian yang berjudul *Aplikasi Jadwal Waktu Salat Metode Ephemeris Dengan Standar Jam Atom BMKG Berbasis Android* dihasilkan dua kesimpulan, yaitu sebagai berikut:

1. Dalam proses pemrograman aplikasi *Accurate Atomic Time for Salat* ini penulis melalui beberapa tahap, yakni: studi literatur dengan mencari literatur yang berkaitan dengan waktu salat dan pengumpulan data berupa data-data yang diperlukan untuk menghitung waktu salat; desain dan perancangan program; dan implementasi perangkat lunak. Pada tahap studi literatur dan pengumpulan data didapatkan bahwa metode perhitungan waktu salat menggunakan algoritma Jean Meus layak digunakan sebagai referensi untuk penentuan waktu salat. Pada tahap desain dan perancangan program, penulis merancang alur pemrograman yang terstruktur agar alur kerja program bisa berjalan dengan baik dan menghasilkan data yang benar.

Tahap terakhir yakni tahap implementasi program. Pada tahap ini seluruh rancangan desain dan rancangan pemrograman yang telah dibuat sebelumnya diimplementasikan ke dalam bahasa pemrograman *Java*. Begitupun dengan alur perhitungan disusun menggunakan bahasa pemrograman *Java*. Adapun desain antarmuka dibuat dengan riset terhadap *style-style* desain yang sedang banyak digunakan pada saat ini.

Hingga terpilihlah desain dengan *flat design style*. Proses desain ini penulis menggunakan *software CorelDraw x7* yang kemudian diimport ke dalam *Android Studio*.

2. Hasil uji fungsionalitas terhadap aplikasi *Accurate Atomic Time for Salat* yang dilakukan penulis menghasilkan temuan bahwa aplikasi ini dapat berjalan dengan baik dan bisa dioperasikan melalui berbagai varian *smartphone* berbasis Android, mulai dengan *smartphone* berspesifikasi rendah hingga *smartphone* berspesifikasi tinggi. Aplikasi ini merupakan aplikasi yang berbasis *online*, maka pengguna perlu menggunakan koneksi internet untuk menjalankannya. Adapun pada uji verifikasi, perhitungan waktu salat yang dihasilkan aplikasi *Accurate Atomic Time for Salat* memiliki perbedaan dengan website milik Bimas Islam, yaitu selisih satu menit, hal ini diperkirakan karena nilai *ikhtiyat* yang berbeda.

B. Saran-saran

Setelah melakukan penelitian dari tahap pengumpulan data, perancangan desain dan program, pengimplementasian dan pengujian terhadap aplikasi *Accurate Atomic Time for Salat*, penulis mempunyai beberapa saran, diantaranya:

1. Aplikasi *Accurate Atomic Time for Salat* ini dapat dijadikan alternatif untuk mengetahui jadwal waktu salat harian dengan sangat praktis dan mudah, cukup dengan memiliki *smartphone* berbasis Android. Pengguna

nantinya bisa mengunduh aplikasi *Accurate Atomic Time for Salat* secara gratis dan langsung bisa digunakan.

2. Aplikasi ini menggunakan metode perhitungan Jean Meus. Namun demikian keakuratan aplikasi masih dapat ditingkatkan dengan menggunakan metode perhitungan yang lebih akurat.
3. Aplikasi *Accurate Atomic Time for Salat* masih dapat dikembangkan lagi dimana untuk saat ini fitur yang tersedia hanya waktu salat. Kedepan aplikasi bisa dikembangkan dengan menambahkan fitur-fitur lain seperti al-Quran, murottal, arah kiblat dan sebagainya sehingga pengguna dapat menikmati berbagai fitur hanya dalam satu aplikasi saja.

C. Penutup

Segala puji bagi Allah atas segala kehendak-Nya. Penulis bersyukur atas segala kenikmatan dan karunia-Nya sehingga dapat menyelesaikan skripsi ini. Penulis sudah berusaha secara maksimal dalam menyusun skripsi ini dengan segala upaya agar bisa sempurna, namun kesempurnaan hanyalah milik Allah semata. Penulis menyadari pada skripsi ini masih terdapat banyak kelemahan dan kekurangan. Oleh karenanya penulis sangat mengharapkan saran dan kritik dari para pembaca, khususnya untuk penelitian dan pengembangan aplikasi *Accurate Atomic Time for Salat* ke depan.

Pada akhirnya, hanya kepada Allah penulis memohon pertolongan dan ampunan. Penulis berharap hasil penelitian ini bermanfaat serta dapat menjadi sumbangsih kecil untuk khazanah keilmuan falak di Indonesia.

DAFTAR PUSTAKA

- Anugraha, Rinto, *Mekanika Benda Langit*, Yogyakarta: Universitas Gadjah Mada, 2012.
- Abi Al-Husain Muslim Bin al-Hajjaj al-Qusyairy, Imam, *Shahih Muslim*, Beirut Libanon: Daar al-Alamiyah, t. th.
- Abidin, Hasanuddin Z., *Geodesi Satelit*, Jakarta: Anem Kosong Anem, 2001.
- Abdul Rojak, Encep, Dkk, “Koreksi Ketinggian Tempat Terhadap Fikih Waktu Salat: Analisis Jadwal Waktu Salat Kota Bandung”, dalam *Al-Ahkam*, vol 27, no.2 Oktober 2007.
- Ad-Dimasyqi , Ismail bin Umar bin Katsir Al-Quraisyi, *Tafsir al-Quranul ‘Azhim*, Beirut Libanon, 2000.
- al-Bukhari, Muhammad bin Ismail, *Shahih Bukhari*, juz 1, Beirut: Beirut: Daar al-Kutub al-Ilmiyah, 1992.
- al-Qurtuby, Muhammad bin Ahmad bin Muhammad bin Ahmad ibnu Rusyd, *Bidayah al-Mujtahid wa Nihayah al-Muqtasid*, Juz II, Baerut: Dar al-Kutub al-Ilmiyah, 1996.
- Ardiansyah, Moelki Fahmi “Implementasi Koordinat tengah Kabupaten atau Kota dalam Perhitungan Waktu Salat”, dalam *Al-Ahkam*, Vol 27, no. 2, Oktober 2017.
- as-Suyuthi, Al-Hafiz Jalal al-Din, *Sunan al-Nasa’i*, Beirut: Daar al-Kutub al-Alamiah,t. th.
- Ath Thabari, Abu Ja’far bin Muhammad Jarir, *Jami Al Bayan an Ta’wil Ayi Al Quran* Jilid IV, 2003.
- _____ *Jami Al Bayan an Ta’wil Ayi Al Quran* Jilid VII, 2003.
- _____ *Jami Al Bayan an Ta’wil Ayi Al Quran* Jilid XII, 2003.
- _____ *Jami Al Bayan an Ta’wil Ayi Al Quran* Jilid XV, 2003.

- _____. *Jami Al Bayan an Ta'wil Ayi Al Quran* Jilid XVI, 2003.
- _____. *Jami Al Bayan an Ta'wil Ayi Al Quran* Jilid XVIII, 2003.
- Azhari, Susiknan, *Ensiklopedi Hisab Rukyat*, Yogyakarta: Pustaka Pelajar, 2012.
- _____, *Ilmu Falak Perjumpaan Khazanah Islam dan Sains Modern*, Yogyakarta: Suara Muhammadiyah, 2011.
- Azizah, Maulidatun Nur, "Analisis Hisab Awal Waktu Salat Dalam Kitab *Asy-Syahru*" Skripsi Fakultas Syariah UIN Walisongo, 2018.
- Aziz, Ahmad Zainuddin bin Abdul, *Fathul Muin*, Cyprus: Daar Ibn Hazm, 2004.
- Dassonville, Patrice F, *The Invention of Time and Space: Origins, Definitions, Nature, Properties*, Switzerland: Springer, t. th.
- Departemen Agama RI, *Al-Quran Dan Terjemahannya*, Surabaya: Pustaka Al-Kautsar, 2009.
- Firly, Nadia, *Create Your Own Android Application* Jakarta: Elex Medai Komputindo, 2018.
- Hambali, Slamet, *Ilmu Falak 1 Penentuan Waktu Salat Dan Arah Kiblat Seluruh Dunia*, Semarang: Program Pascasarjana IAIN Walisongo Semarang, 2011.
- Huda, Arif Akbarul *Live Coding! 9 Aplikasi Android Buatan Sendiri* Yogyakarta: Andi Offset, 2013.
- Izzuddin, Ahmad, *Ilmu Falak Praktis*, Semarang PT. Pustaka Rizki Putra, 2012.
- Jamil, A, *Ilmu Falak (Teori dan Aplikasi)*, Jakarta: Amzah, 2016.
- Kementrian Agama RI, *Ephemeris Hisab Rukyat*, Jakarta: Direktorat Urusan Agama Islam dan Pembinaan Syariah Direktorat Jendral Bimbingan Masyarakat Islam Kementrian Agama RI, 2017.
- Khazin, Muhyiddin, *Kamus Ilmu Falak*, Yogyakarta: Buana Pustaka, 2005.

- _____, *Ilmu Falak Dalam Teori dan Praktek*, Yogyakarta: Buana Pustaka, 2004.
- Marapaung, Wani, *Pengantar Ilmu Falak*, Jakarta: Prenadamedia Group, 2015.
- Meus, Jean, *Astronomical Algorithms*, Vigrinia: Willmann-Bell, Inc, 1991.
- Mufidoh, Novi Arijatul, “Sistem Hisab Awal Waktu Salat Program Website Bimbingan Masyarakat Islam”, Skripsi Fakultas Syariah Dan Hukum UIN Walisongo Semarang, 2018.
- Muhadjir, Noeng, *Metodologi Penelitian Kualitatif*, Yogyakarta: Rake Sarasin, 1996.
- Munawwir, Ahmad Warson, *Kamus Al-Munawwir Arab-Indonesia Terlengkap*, Surabaya: Pustaka Progresif, 1997.
- Musonif, Ahmad, *Ilmu Falak (Metode Hisab Awal Waktu Salat, Arah Kiblat, Hisab Urfi, Dan Hisab Hakiki Awal Bulan)*, Yogyakarta: Teras, 2011.
- Pusat Pengembangan Bahasa, *Bahasa Indonesia Bahasa Bangsa*, Semarang: IAIN Walisongo, 2014.
- Radliya, Nizar Rabbi, *Modul Kuliah Pemrograman Mobile (Android)*, UNIKOM, 2016.
- Sugiyono, *Metode Penelitian Pendidikan Oendekatan Kuantitatif, Kualitatif, dan R&D* Bandung: Alfabeta, 2007.
- Supriyanto, *Jaringan Dasar 1*, Jakarta: Kementerian Pendidikan & Kebudayaan, 2013.
- Tim Penyusun Kamus Bahasa Indonesia, *Kamus Bahasa Indonesia*, Jakarta: Pusat Bahasa, 2008.
- Zuhaili, Wahbah, *At-Tafsirul Munir: Fil ‘Aqidah wasy-Syarii’ah wal Manhaj* Dimsyq: Daarul Fikr Jilid VIII, 2003.
- _____, *At-Tafsirul Munir: Fil ‘Aqidah wasy-Syarii’ah wal Manhaj* Dimsyq: Daarul Fikr Jilid XI, 2003.

Abid (Tim IT Yaumee), *Wawancara*, Jumat, 21 Juni 2019.

Fadhli (Tim IT Kesan), *Wawancara*, Sabtu, 8 Juni 2019.

Devid Bidang Seismologi Teknik, Geofisika Potensial, dan Tanda Waktu),
Wawancara. Jakarta, 20 Mei 2019

http://droid-indonesia.blogspot.co.id/2012/01/sejarah-android_11.html

<http://www.big.go.id/sejarah/>

<http://journal.walisongo.ac.id/index.php/ahkam/article/view/1981>

<https://www.tifannywatch.com/mengenal-dua-tipe-dari-mesin-penggerak-jam-tangan-analog/>

http://www.dpr.go.id/dokjdih/document/uu/UU_2009_31.pdf

http://webdata.bmkg.go.id/web/standar_waktu_indonesia.pdf

<https://www.twoh.co/2012/09/18/mengenal-arsitektur-sistem-operasi-android/>

<https://developer.android.com/guide/platform?hl=id>

<https://teknikelektronika.com/pengertian-osilator-prinsip-kerja-oscillato/>

<https://eclipse.gsfc.nasa.gov/SEcat5/deltatpoly.html>

<https://bimasislam.kemenag.go.id/site/profil/tugas-dan-fungsi>

Lampiran 1

Source code Awal Waktu Salat pada Aplikasi Accurate Atomic Time for Salat

```
double waktux = jam1 + menit1 / 60 + detik1 / 3600;

Double year2;
if (bulan1 < 3) {
    year2 = tahun1 - 1;
} else {
    year2 = tahun1;
}

Double month2;
if (bulan1 < 3) {
    month2 = bulan1 + 12;
} else {
    month2 = bulan1;
}

double timezonel = bede / 15;
double timezone = intek(timezonel);

Double date2 = tanggal1;
Double waktu = waktux - timezone;

Double jede = intek(365.25 * (year2 + 4716)) +
intek(30.6001 * (month2 + 1)) + date2 + (2 -
(intek(year2 / 100)) + intek((intek(year2 / 100) /
4))) + waktu / 24 - 1524.5;

Double delta;
if (year2 < -500) {
    delta = (-20 + 32 * Math.pow((year2 / 100 - 18.2),
2)) / 3600 / 24;
} else if (year2 >= -500 && year2 < 500) {
    delta = (10583.6 - 1014.41 * (year2 / 100) +
33.78311 * Math.pow((year2 / 100), 2) - 5) / 3600 /
24;
} else if (year2 >= 500 && year2 < 1600) {
    - 0.01532 * Math.pow((year2 - 1600), 2) +
Math.pow((year2 - 1600), 3) / 7129) / 3600 / 24;
} else if (year2 >= 1700 && year2 < 1800) {
    delta = (8.83 + 0.1603 * (year2 - 1700) -
0.0059285 * Math.pow((year2 - 1700), 2) + 0.00013336 *
Math.pow((year2 - 1700), 3) - Math.pow((year2 - 1700),
4) / 1174000) / 3600 / 24;
} else if (year2 >= 1800 && year2 < 1860) {
    delta = (13.72 - 0.332447 * (year2 - 1800) +
```

```

0.0068612 * Math.pow((year2 - 1800), 2) + 0.0041116 *
Math.pow((year2 - 1800), 3) - 0.00037436 *
Math.pow((year2 - 1800), 4) + 0.0000121272 *
Math.pow((year2 - 1800), 5) - 0.0000001699 *
Math.pow((year2 - 1800), 6) + 0.000000000875 *
Math.pow((year2 - 1800), 7)) / 3600 / 24 / 3600 / 24;
} else if (year2 >= 1860 && year2 < 1900) {
    delta = (7.62 + 0.5737 * (year2 - 1860) - 0.251754
* Math.pow((year2 - 1860), 2) + 0.01680668 *
Math.pow((year2 - 1860), 3) - 0.0004473624 *
Math.pow((year2 - 1860), 4) + Math.pow((year2 - 1850),
5) / 233174) / 3600 / 24;
} else if (year2 >= 1900 && year2 < 1920) {
    delta = (-2.79 + 1.494119 * (year2 - 1900) -
0.0598939 * Math.pow((year2 - 1900), 2) + 0.0061966 *
Math.pow((year2 - 1900), 3) - 0.000197 *
Math.pow((year2 - 1900), 4)) / 3600 / 24;
} else if (year2 >= 1920 && year2 < 1941) {
    delta = (21.2 + 0.84493 * (year2 - 1920) - 0.0761
* / 3600 / 24;
} else if (year2 >= 2005 && year2 < 2050) {
    delta = (62.92 + 0.32217 * ((year2 + month2 / 12 +
date2 / 365.25 + waktu / 24 / 365.25) - 2000) +
0.005589 * Math.pow(((year2 + month2 / 12 + date2 /
365.25 + waktu / 24 / 365.25) - 2000), 2)) / 3600 /
24;
} else if (year2 >= 2050 && year2 < 2150) {
    delta = (-20 + 32 * Math.pow(((year2 - 1820) /
100), 2) - 0.5628 * (2150 - year2)) / 3600 / 24;
} else if (year2 >= 2150) {
    delta = (-20 + 32 * Math.pow(((year2 - 1820) /
100), 2)) / 3600 / 24;
} else {
    delta = Double.valueOf(00000);
}
;

```

```

Double jde = jede + delta;
Double tete = (jde - 2451545) / 36525;
Double tautau = tete / 10;

```

```

Double epsilon4 = 23.43929111;
Double epsilon33 = (46.815 / 3600 * tete);
Double epsilon22 = (0.00059 / 3600 * Math.pow(tete,
2));
Double epsilon1 = (0.001813 / 3600 * Math.pow(tete,
3));
Double epsilon = epsilon4 - epsilon33 - epsilon22 +
epsilon1;

```

```

Double elo = MOD((280.46645 + 36000.76983 * tete +
0.0003032 * Math.pow(tete, 2)), 360);

Double mbes = MOD((357.5291 + 35999.0503 * tete -
0.0001559 * Math.pow(tete, 2) - 0.00000048 *
Math.pow(tete, 3)), 360);
* (mbes)) + (0.019993 - 0.000101 * tete) *
Math.sin(Math.PI / 180 * (2 * mbes)) + 0.00029 *
Math.sin(Math.PI / 180 * (3 * mbes)), 360);
Double owe = MOD((elo + cbes), 360);
Double vcek = MOD((mbes + cbes), 360);

Double rbes = (1.000001018 * (1 - Math.pow(ecek, 2)))
/ (1 + ecek * Math.cos(Math.PI / 180 * (vcek)));
Double omega = MOD((125.04 - 1934.136 * tete), 360);
Double lambda = MOD((owe - 0.00569 - 0.00478 *
Math.sin(Math.PI / 180 * (omega))), 360);
Double alpha = MOD((180 / Math.PI *
(Math.atan(Math.cos(Math.PI / 180 * (epsilon)) *
Math.sin(Math.PI / 180 * (owe))) / Math.cos(Math.PI /
180 * (owe))))), 360);
Double delta1 = 180 / Math.PI *
(Math.asin(Math.sin(Math.PI / 180 * (epsilon)) *
//koreksilnol
double kor1 = (175347046 * Math.cos(0 + 0 * tautau));
double kor2 = (3341656 * Math.cos(4.6692568 +
6283.07585 * tautau));
double kor3 = (34894 * Math.cos(4.6261 + 12566.1517 *
tautau));
double kor4 = (3497 * Math.cos(2.7441 + 5753.3849 *
tautau));
double kor5 = (3418 * Math.cos(2.8289 + 3.5231 *
tautau));
double kor6 = (3136 * Math.cos(3.6277 + 77713.7715 *
tautau));
double kor7 = (2676 * Math.cos(4.4181 + 7860.4194 *
tautau));
double kor8 = (2343 * Math.cos(6.1352 + 3930.2097 *
tautau));
double kor9 = (1324 * Math.cos(0.7425 + 11506.7698 *
tautau));
double kor10 = (1273 * Math.cos(2.0371 + 529.691 *
tautau));
double kor11 = (1199 * Math.cos(1.1096 + 1577.3435 *
tautau));
double kor12 = (990 * Math.cos(5.233 + 5884.927 *
tautau));
double kor13 = (902 * Math.cos(2.045 + 26.298 *
tautau));
double kor14 = (857 * Math.cos(3.508 + 398.149 *
tautau));

```

```
double kor15 = (780 * Math.cos(1.179 + 5223.694 *
tautau));
double kor16 = (753 * Math.cos(2.533 + 5507.553 *
tautau));
double kor17 = (505 * Math.cos(4.583 + 18849.228 *
tautau));
double kor18 = (492 * Math.cos(4.205 + 775.523 *
tautau));
double kor19 = (357 * Math.cos(2.92 + 0.067 *
tautau));
double kor20 = (317 * Math.cos(5.849 + 11790.629 *
tautau));
double kor21 = (284 * Math.cos(1.899 + 796.298 *
tautau));
double kor22 = (271 * Math.cos(0.315 + 10977.079 *
tautau));
double kor23 = (243 * Math.cos(0.345 + 5486.778 *
tautau));
double kor24 = (206 * Math.cos(4.806 + 2544.314 *
tautau));
double kor25 = (205 * Math.cos(1.869 + 5573.143 *
tautau));
double kor26 = (202 * Math.cos(2.458 + 6069.777 *
tautau));
double kor27 = (156 * Math.cos(0.833 + 213.299 *
tautau));
double kor28 = (132 * Math.cos(3.411 + 2942.463 *
tautau));
double kor29 = (126 * Math.cos(1.083 + 20.775 *
tautau));
double kor30 = (115 * Math.cos(0.645 + 0.98 *
tautau));
double kor31 = (103 * Math.cos(0.636 + 4694.003 *
tautau));
double kor32 = (102 * Math.cos(0.976 + 15720.839 *
tautau));
double kor33 = (102 * Math.cos(4.267 + 7.114 *
tautau));
double kor34 = (99 * Math.cos(6.21 + 2146.17 *
tautau));
double kor35 = (98 * Math.cos(0.68 + 155.42 *
tautau));
double kor36 = (86 * Math.cos(5.98 + 161000.69 *
tautau));
double kor37 = (85 * Math.cos(1.3 + 6275.96 *
tautau));
double kor38 = (85 * Math.cos(3.67 + 71430.7 *
tautau));
double kor39 = (80 * Math.cos(1.81 + 17260.15 *
tautau));
```



```

double jumlahkor = kor1 + kor2 + kor3 + kor4 + kor5 +
kor6 + kor7 + kor8 + kor9 + kor10 + kor11 + kor12 +
kor13
      + kor14 + kor15 + kor16 + kor17 + kor18 +
kor19 + kor20 + kor21 + kor22 + kor23 + kor24 + kor25
+ kor26 + kor27 + kor28
      + kor29 + kor30 + kor31 + kor32 + kor33 +
kor34 + kor35 + kor36 + kor37 + kor38 + kor39 + kor40
+ kor41 + kor42 + kor43
      + kor44 + kor45 + kor46 + kor47 + kor48 +
kor49 + kor50 + kor51 + kor52 + kor53 + kor54 + kor55
+ kor56 + kor57 + kor58
      + kor59 + kor60 + kor61 + kor62 + kor63 +
kor64;

```

```

//koreksibujur ekliptika

```

```

double korq1 = 6283319667471 * Math.cos(0 + 0 *
tautau);
double korq2 = 206059 * Math.cos(2.678235 + 6283.07585
* tautau);
double korq3 = 4303 * Math.cos(2.6351 + 12566.1517 *
tautau);
double korq4 = 425 * Math.cos(1.59 + 3.523 * tautau);
double korq5 = 119 * Math.cos(5.796 + 26.298 *
tautau);
double korq6 = 109 * Math.cos(2.966 + 1577.344 *
tautau);
double korq7 = 93 * Math.cos(2.59 + 18849.23 *
tautau);
double korq8 = 72 * Math.cos(1.14 + 529.69 * tautau);
double korq9 = 68 * Math.cos(1.87 + 398.15 * tautau);
double korq10 = 67 * Math.cos(4.41 + 5507.55 *
tautau);
double korq11 = 59 * Math.cos(2.89 + 5223.69 *
tautau);
double korq12 = 56 * Math.cos(2.17 + 155.42 * tautau);
double korq13 = 45 * Math.cos(0.4 + 796.3 * tautau);
double korq14 = 36 * Math.cos(0.47 + 775.52 * tautau);
double korq15 = 29 * Math.cos(2.65 + 7.11 * tautau);
double korq16 = 21 * Math.cos(5.34 + 0.98 * tautau);
double korq17 = 19 * Math.cos(1.85 + 5486.78 *
tautau);
double korq18 = 19 * Math.cos(4.97 + 213.3 * tautau);
double jumlahkorq = korq1 + korq2 + korq3 + korq4 +
korq5 + korq6 + korq7 + korq8 + korq9 + korq10
      + korq11 + korq12 + korq13 + korq14 + korq15 +
korq16 + korq17 + korq18 + korq19 + korq20
      + korq21 + korq22 + korq23 + korq24 + korq25 +
korq26 + korq27 + korq28 + korq29 + korq30
      + korq31 + korq32 + korq33 + korq34;

```

```

//koreksibujur L2
double korw1 = 52919 * Math.cos(0 + 0 * tautau);
double korw2 = 8720 * Math.cos(1.0721 + 6283.0758 *
tautau);
double korw3 = 309 * Math.cos(0.867 + 12566.152 *
tautau);
double korw4 = 27 * Math.cos(0.05 + 3.52 * tautau);
double korw5 = 16 * Math.cos(5.19 + 26.3 * tautau);
double korw6 = 16 * Math.cos(3.68 + 155.42 * tautau);
double korw7 = 10 * Math.cos(0.76 + 18849.23 *
tautau);
double korw8 = 9 * Math.cos(2.06 + 77713.77 * tautau);
double korw9 = 7 * Math.cos(0.83 + 775.52 * tautau);
double korw10 = 5 * Math.cos(4.66 + 1577.34 * tautau);
double korw11 = 4 * Math.cos(1.03 + 7.11 * tautau);
double korw12 = 4 * Math.cos(3.44 + 5573.14 * tautau);
double korw13 = 3 * Math.cos(5.14 + 796.3 * tautau);
double korw14 = 3 * Math.cos(6.05 + 5507.55 * tautau);
double korw15 = 3 * Math.cos(1.19 + 242.73 * tautau);
double korw16 = 3 * Math.cos(6.12 + 529.69 * tautau);
double jumlahkorw = korw1 + korw2 + korw3 + korw4 +
korw5 + korw6 + korw7 + korw8 + korw9
      + korw10 + korw11 + korw12 + korw13 + korw14 +
korw15 + korw16 + korw17 + korw18 + korw19 + korw20;

//koreksi bujur L3
double kore1 = 289 * Math.cos(5.844 + 6283.076 *
tautau);
double kore2 = 35 * Math.cos(0 + 0 * tautau);
double kore3 = 17 * Math.cos(5.49 + 12566.15 *
tautau);
double kore4 = 3 * Math.cos(5.2 + 155.42 * tautau);
double jumlahkore = kore1 + kore2 + kore3 + kore4 +
kore5 + kore6 + kore7;

//koreksi bujur L4
double korr1 = 114 * Math.cos(3.142 + 0 * tautau);
double korr2 = 8 * Math.cos(4.13 + 6283.08 * tautau);
double korr3 = 1 * Math.cos(3.84 + 12566.15 * tautau);
double jumlahkorr = korr1 + korr2 + korr3;

//koreksi bujur L5
double jumlahkort = 1 * Math.cos(3.14 + 0 * tautau);

double elbes = MOD((180 / Math.PI * ((jumlahkor +
jumlahkorq * tautau + jumlahkorw
      * Math.pow(tautau, 2) + jumlahkore *
Math.pow(tautau, 3) + jumlahkorr
      * Math.pow(tautau, 4) + jumlahkort *
Math.pow(tautau, 5)) / Math.pow(10, 8))), 360);

```

```

double owe2 = MOD((elbes + 180), 360);
double iaksen = (owe2 - 1.397 * tete - 0.00031 *
Math.pow(tete, 2)) - 0.09033 / 3600;
double segitiga = -0.09033 / 3600;

//koreksi bumi matahari R0
double kory1 = 100013989 * Math.cos(0 + 0 * tautau);
double kory2 = 1670700 * Math.cos(3.0984635 +
6283.07585 * tautau);
double kory3 = 13956 * Math.cos(3.05525 + 12566.1517 *
tautau);
double kory4 = 3084 * Math.cos(5.1985 + 77713.7715 *
tautau);
double kory5 = 1628 * Math.cos(1.1739 + 5753.3849 *
tautau);
double kory6 = 1576 * Math.cos(2.8469 + 7860.4194 *
tautau);
double kory7 = 925 * Math.cos(5.453 + 11506.77 *
tautau);
double kory8 = 542 * Math.cos(4.564 + 3930.21 *
tautau);
double kory9 = 472 * Math.cos(3.661 + 5884.927 *
tautau);
double kory10 = 346 * Math.cos(0.964 + 5507.553 *
tautau);
double kory11 = 329 * Math.cos(5.9 + 5223.694 *
tautau);
double kory12 = 307 * Math.cos(0.299 + 5573.143 *
tautau);
double kory13 = 243 * Math.cos(4.273 + 11790.629 *
tautau);
double kory14 = 212 * Math.cos(5.847 + 1577.344 *
tautau);
double kory15 = 186 * Math.cos(5.022 + 10977.079 *
tautau);
double kory16 = 175 * Math.cos(3.012 + 18849.228 *
tautau);
double kory17 = 110 * Math.cos(5.055 + 5486.778 *
tautau);
double kory18 = 98 * Math.cos(0.89 + 6069.78 *
tautau);
double kory19 = 86 * Math.cos(5.69 + 15720.84 *
tautau);
double kory20 = 86 * Math.cos(1.27 + 161000.69 *
tautau);
double kory21 = 65 * Math.cos(0.27 + 17260.15 *
tautau);
double kory22 = 63 * Math.cos(0.92 + 529.69 * tautau);
double kory23 = 57 * Math.cos(2.01 + 83996.85 *
tautau);

```

```

double kory24 = 56 * Math.cos(5.24 + 71430.7 *
tautau);
double kory25 = 49 * Math.cos(3.25 + 2544.31 *
tautau);
double kory26 = 47 * Math.cos(2.58 + 775.52 * tautau);
double kory27 = 45 * Math.cos(5.54 + 9437.76 *
tautau);
double kory28 = 43 * Math.cos(6.01 + 6275.96 *
tautau);
double kory29 = 39 * Math.cos(5.36 + 4694 * tautau);
double jumlahkory = kory1 + kory2 + kory3 + kory4 +
kory5 + kory6 + kory7 + kory8 + kory9
+ kory10 + kory11 + kory12 + kory13 + kory14 +
kory15 + kory16 + kory17 + kory18 + kory19 + kory20
+ kory21 + kory22 + kory23 + kory24 + kory25 +
kory26 + kory27 + kory28 + kory29 + kory30
+ kory31 + kory32 + kory33 + kory34 + kory35 +
kory36 + kory37 + kory38 + kory39 + kory40;

//koreksi bumi matahari R1
double koru1 = 103019 * Math.cos(1.10749 + 6283.07585
* tautau);
double koru2 = 1721 * Math.cos(1.0644 + 12566.1517 *
tautau);
double koru3 = 702 * Math.cos(3.142 + 0 * tautau);
double koru4 = 32 * Math.cos(1.02 + 18849.23 *
tautau);
double koru5 = 31 * Math.cos(2.84 + 5507.55 * tautau);
double koru6 = 25 * Math.cos(1.32 + 5223.69 * tautau);
double koru7 = 18 * Math.cos(1.42 + 1577.34 * tautau);
double koru8 = 10 * Math.cos(5.91 + 10977.08 *
tautau);
double koru9 = 9 * Math.cos(1.42 + 6275.96 * tautau);
double koru10 = 9 * Math.cos(0.27 + 5486.78 * tautau);
double jumlahkoru = koru1 + koru2 + koru3 + koru4 +
koru5 + koru6 + koru7 + koru8 + koru9 + koru10;

// koreksi bumi matahari R2
double kori1 = 4359 * Math.cos(5.7846 + 6283.0758 *
tautau);
double kori2 = 124 * Math.cos(5.579 + 12566.152 *
tautau);
double kori3 = 12 * Math.cos(3.14 + 0 * tautau);
double kori4 = 9 * Math.cos(3.63 + 77713.77 * tautau);
double kori5 = 6 * Math.cos(1.87 + 5573.14 * tautau);
double kori6 = 3 * Math.cos(5.47 + 18849.23 * tautau);
double jumlahkori = kori1 + kori2 + kori3 + kori4 +
kori5 + kori6;

//koreksi bumi matahari R3
double korol = 145 * Math.cos(4.273 + 6283.076 *

```

```

tautau);
double koro2 = 7 * Math.cos(3.92 + 12566.15 * tautau);
double jumlahkoro = koro1 + koro2;

//koreksi bumi matahari R4
double jumlahkorp = 4 * Math.cos(2.56 + 6283.08 *
tautau);

double erbes = (jumlahkory + jumlahkoru * tete +
jumlahkori * Math.pow(tete, 2) + jumlahkoro *
Math.pow(tete, 3) + jumlahkorp * Math.pow(tete, 4)) /
Math.pow(10, 8);
double debes = MOD((297.85036 + 445267.11148 * tete -
0.0019142 * Math.pow(tete, 2) + Math.pow(tete, 3) /
189474), 360);
double embes = MOD((357.52772 + 35999.05034 * tete -
0.0001603 * Math.pow(tete, 2) - Math.pow(tete, 3) /
300000), 360);
double emaks = MOD((134.96298 + 477198.867398 * tete +
0.0086972 * Math.pow(tete, 2) + Math.pow(tete, 3) /
56250), 360);
double efbes = MOD((93.27191 + 483202.017538 * tete -
0.0036825 * Math.pow(tete, 2) + Math.pow(tete, 3) /
327270), 360);
double omega2 = MOD((125.04452 - 1934.136261 * tete +
0.0020708 * Math.pow(tete, 2) + Math.pow(tete, 3) /
450000), 360);
double ubes = tete / 100;
double epsilon2 = 23.43929111 - (4680.93 * ubes - 1.55
* Math.pow(ubes, 2) + 1999.25 * Math.pow(ubes, 3) -
51.38 * Math.pow(ubes, 4) - 249.67 * Math.pow(ubes, 5)
- 39.05 * Math.pow(ubes, 6) + 7.12 * Math.pow(ubes, 7)
+ 27.87 * Math.pow(ubes, 8) + 5.79 * Math.pow(ubes, 9)
+ 2.45 * Math.pow(ubes, 10)) / 3600;

//koreksi asi
double korn1 = (-171996 + -174.2 * tete) * Math.sin(0
* Math.PI / 180 * (debes) + 0 * Math.PI / 180 *
(embes) + 0 * Math.PI / 180 * (emaks) + 0 * Math.PI /
180 * (efbes) + 1 * Math.PI / 180 * (omega2));
double korn2 = (-13187 + -1.6 * tete) * Math.sin(-2 *
Math.PI / 180 * (debes) + 0 * Math.PI / 180 * (embes)
+ 0 * Math.PI / 180 * (emaks) + 2 * Math.PI / 180 *
(efbes) + 2 * Math.PI / 180 * (omega2));
double korn3 = (-2274 + -0.2 * tete) * Math.sin(0 *
Math.PI / 180 * (debes) + 0 * Math.PI / 180 * (embes)
+ 0 * Math.PI / 180 * (emaks) + 2 * Math.PI / 180 *
(efbes) + 2 * Math.PI / 180 * (omega2));
double korn4 = (2062 + 0.2 * tete) * Math.sin(0 *
Math.PI / 180 * (debes) + 0 * Math.PI / 180 * (embes)

```

```

+ 0 * Math.PI / 180 * (emaks) + 0 * Math.PI / 180 *
(efbes) + 2 * Math.PI / 180 * (omega2));
double korn5 = (1426 + -3.4 * tete) * Math.sin(0 *
Math.PI / 180 * (debes) + 1 * Math.PI / 180 * (embes)
+ 0 * Math.PI / 180 * (emaks) + 0 * Math.PI / 180 *
(efbes) + 0 * Math.PI / 180 * (omega2));
double korn6 = (712 + 0.1 * tete) * Math.sin(0 *
Math.PI / 180 * (debes) + 0 * Math.PI / 180 * (embes)
+ 1 * Math.PI / 180 * (emaks) + 0 * Math.PI / 180 *
(efbes) + 0 * Math.PI / 180 * (omega2));
double korn7 = (517 + 1.2 * tete) * Math.sin(-2 *
Math.PI / 180 * (debes) + 1 * Math.PI / 180 * (embes)
+ 0 * Math.PI / 180 * (emaks) + 2 * Math.PI / 180 *
(efbes) + 2 * Math.PI / 180 * (omega2));
double korn8 = (-386 + -0.4 * tete) * Math.sin(0 *
Math.PI / 180 * (debes) + 0 * Math.PI / 180 * (embes)
+ 0 * Math.PI / 180 * (emaks) + 2 * Math.PI / 180 *
(efbes) + 1 * Math.PI / 180 * (omega2));
double korn9 = (217 + -0.5 * tete) * Math.sin(-2 *
Math.PI / 180 * (debes) + -1 * Math.PI / 180 * (embes)
+ 0 * Math.PI / 180 * (emaks) + 2 * Math.PI / 180 *
(efbes) + 2 * Math.PI / 180 * (omega2));
double korn10 = (129 + 0.1 * tete) * Math.sin(-2 *
Math.PI / 180 * (debes) + 0 * Math.PI / 180 * (embes)
+ 0 * Math.PI / 180 * (emaks) + 2 * Math.PI / 180 *
(efbes) + 1 * Math.PI / 180 * (omega2));
double korn11 = (63 + 0.1 * tete) * Math.sin(0 *
Math.PI / 180 * (debes) + 0 * Math.PI / 180 * (embes)
+ 1 * Math.PI / 180 * (emaks) + 0 * Math.PI / 180 *
(efbes) + 1 * Math.PI / 180 * (omega2));
double korn12 = (-58 + -0.1 * tete) * Math.sin(0 *
Math.PI / 180 * (debes) + 0 * Math.PI / 180 * (embes)
+ -1 * Math.PI / 180 * (emaks) + 0 * Math.PI / 180 *
(efbes) + 1 * Math.PI / 180 * (omega2));
double korn13 = (17 + -0.1 * tete) * Math.sin(0 *
Math.PI / 180 * (debes) + 2 * Math.PI / 180 * (embes)
+ 0 * Math.PI / 180 * (emaks) + 0 * Math.PI / 180 *
(efbes) + 0 * Math.PI / 180 * (omega2));
double korn14 = (-16 + 0.1 * tete) * Math.sin(-2 *
Math.PI / 180 * (debes) + 2 * Math.PI / 180 * (embes)
+ 0 * Math.PI / 180 * (emaks) + 2 * Math.PI / 180 *
(efbes) + 2 * Math.PI / 180 * (omega2));
double korn15 = (-301 + 0 * tete) * Math.sin(0 *
Math.PI / 180 * (debes) + 0 * Math.PI / 180 * (embes)
+ 1 * Math.PI / 180 * (emaks) + 2 * Math.PI / 180 *
(efbes) + 2 * Math.PI / 180 * (omega2));
double korn16 = (-158 + 0 * tete) * Math.sin(-2 *
Math.PI / 180 * (debes) + 0 * Math.PI / 180 * (embes)
+ 1 * Math.PI / 180 * (emaks) + 0 * Math.PI / 180 *
(efbes) + 0 * Math.PI / 180 * (omega2));
double korn17 = (123 + 0 * tete) * Math.sin(0 *

```

```

Math.PI / 180 * (debes) + 0 * Math.PI / 180 * (embes)
+ -1 * Math.PI / 180 * (emaks) + 2 * Math.PI / 180 *
(efbes) + 2 * Math.PI / 180 * (omega2));
double korn18 = (63 + 0 * tete) * Math.sin(2 * Math.PI
/ 180 * (debes) + 0 * Math.PI / 180 * (embes) + 0 *
Math.PI / 180 * (emaks) + 0 * Math.PI / 180 * (efbes)
+ 0 * Math.PI / 180 * (omega2));
double korn19 = (-59 + 0 * tete) * Math.sin(2 *
Math.PI / 180 * (debes) + 0 * Math.PI / 180 * (embes)
+ -1 * Math.PI / 180 * (emaks) + 2 * Math.PI / 180 *
(efbes) + 2 * Math.PI / 180 * (omega2));
double korn20 = (-51 + 0 * tete) * Math.sin(0 *
Math.PI / 180 * (debes) + 0 * Math.PI / 180 * (embes)
+ 1 * Math.PI / 180 * (emaks) + 2 * Math.PI / 180 *
(efbes) + 1 * Math.PI / 180 * (omega2));
double korn21 = (48 + 0 * tete) * Math.sin(-2 *
Math.PI / 180 * (debes) + 0 * Math.PI / 180 * (embes)
+ 2 * Math.PI / 180 * (emaks) + 0 * Math.PI / 180 *
(efbes) + 0 * Math.PI / 180 * (omega2));
double korn22 = (46 + 0 * tete) * Math.sin(0 * Math.PI
/ 180 * (debes) + 0 * Math.PI / 180 * (embes) + -2 *
Math.PI / 180 * (emaks) + 2 * Math.PI / 180 * (efbes)
+ 1 * Math.PI / 180 * (omega2));
double korn23 = (-38 + 0 * tete) * Math.sin(2 *
Math.PI / 180 * (debes) + 0 * Math.PI / 180 * (embes)
+ 0 * Math.PI / 180 * (emaks) + 2 * Math.PI / 180 *
(efbes) + 2 * Math.PI / 180 * (omega2));
double korn24 = (-31 + 0 * tete) * Math.sin(0 *
Math.PI / 180 * (debes) + 0 * Math.PI / 180 * (embes)
+ 2 * Math.PI / 180 * (emaks) + 2 * Math.PI / 180 *
(efbes) + 2 * Math.PI / 180 * (omega2));
double korn25 = (29 + 0 * tete) * Math.sin(0 * Math.PI
/ 180 * (debes) + 0 * Math.PI / 180 * (embes) + 2 *
Math.PI / 180 * (emaks) + 0 * Math.PI / 180 * (efbes)
+ 0 * Math.PI / 180 * (omega2));
double korn26 = (29 + 0 * tete) * Math.sin(-2 *
Math.PI / 180 * (debes) + 0 * Math.PI / 180 * (embes)
+ 1 * Math.PI / 180 * (emaks) + 2 * Math.PI / 180 *
(efbes) + 2 * Math.PI / 180 * (omega2));
double korn27 = (26 + 0 * tete) * Math.sin(0 * Math.PI
/ 180 * (debes) + 0 * Math.PI / 180 * (embes) + 0 *
Math.PI / 180 * (emaks) + 2 * Math.PI / 180 * (efbes)
+ 0 * Math.PI / 180 * (omega2));
double korn28 = (-22 + 0 * tete) * Math.sin(-2 *
Math.PI / 180 * (debes) + 0 * Math.PI / 180 * (embes)
+ 0 * Math.PI / 180 * (emaks) + 2 * Math.PI / 180 *
(efbes) + 0 * Math.PI / 180 * (omega2));
double korn29 = (21 + 0 * tete) * Math.sin(0 * Math.PI
/ 180 * (debes) + 0 * Math.PI / 180 * (embes) + -1 *
Math.PI / 180 * (emaks) + 2 * Math.PI / 180 * (efbes)
+ 1 * Math.PI / 180 * (omega2));

```

```

double korn30 = (16 + 0 * tete) * Math.sin(2 * Math.PI
/ 180 * (debes) + 0 * Math.PI / 180 * (embes) + -1 *
Math.PI / 180 * (emaks) + 0 * Math.PI / 180 * (efbes)
+ 1 * Math.PI / 180 * (omega2));
double korn31 = (-15 + 0 * tete) * Math.sin(0 *
Math.PI / 180 * (debes) + 1 * Math.PI / 180 * (embes)
+ 0 * Math.PI / 180 * (emaks) + 0 * Math.PI / 180 *
(efbes) + 1 * Math.PI / 180 * (omega2));
double korn32 = (-13 + 0 * tete) * Math.sin(-2 *
Math.PI / 180 * (debes) + 0 * Math.PI / 180 * (embes)
+ 1 * Math.PI / 180 * (emaks) + 0 * Math.PI / 180 *
(efbes) + 1 * Math.PI / 180 * (omega2));
double korn33 = (-12 + 0 * tete) * Math.sin(0 *
Math.PI / 180 * (debes) + -1 * Math.PI / 180 * (embes)
+ 0 * Math.PI / 180 * (emaks) + 0 * Math.PI / 180 *
(efbes) + 1 * Math.PI / 180 * (omega2));
double korn34 = (11 + 0 * tete) * Math.sin(0 * Math.PI
/ 180 * (debes) + 0 * Math.PI / 180 * (embes) + 2 *
Math.PI / 180 * (emaks) + -2 * Math.PI / 180 * (efbes)
+ 0 * Math.PI / 180 * (omega2));
double korn35 = (-10 + 0 * tete) * Math.sin(2 *
Math.PI / 180 * (debes) + 0 * Math.PI / 180 * (embes)
+ -1 * Math.PI / 180 * (emaks) + 2 * Math.PI / 180 *
(efbes) + 1 * Math.PI / 180 * (omega2));
double korn36 = (-8 + 0 * tete) * Math.sin(2 * Math.PI
/ 180 * (debes) + 0 * Math.PI / 180 * (embes) + 1 *
Math.PI / 180 * (emaks) + 2 * Math.PI / 180 * (efbes)
+ 2 * Math.PI / 180 * (omega2));
double korn37 = (7 + 0 * tete) * Math.sin(0 * Math.PI
/ 180 * (debes) + 1 * Math.PI / 180 * (embes) + 0 *
Math.PI / 180 * (emaks) + 2 * Math.PI / 180 * (efbes)
+ 2 * Math.PI / 180 * (omega2));
double korn38 = (-7 + 0 * tete) * Math.sin(-2 *
Math.PI / 180 * (debes) + 1 * Math.PI / 180 * (embes)
+ 1 * Math.PI / 180 * (emaks) + 0 * Math.PI / 180 *
(efbes) + 0 * Math.PI / 180 * (omega2));
double korn39 = (-7 + 0 * tete) * Math.sin(0 * Math.PI
/ 180 * (debes) + -1 * Math.PI / 180 * (embes) + 0 *
Math.PI / 180 * (emaks) + 2 * Math.PI / 180 * (efbes)
+ 2 * Math.PI / 180 * (omega2));
double korn40 = (-7 + 0 * tete) * Math.sin(2 * Math.PI
/ 180 * (debes) + 0 * Math.PI / 180 * (embes) + 0 *
Math.PI / 180 * (emaks) + 2 * Math.PI / 180 * (efbes)
+ 1 * Math.PI / 180 * (omega2));
double korn41 = (6 + 0 * tete) * Math.sin(2 * Math.PI
/ 180 * (debes) + 0 * Math.PI / 180 * (embes) + 1 *
Math.PI / 180 * (emaks) + 0 * Math.PI / 180 * (efbes)
+ 0 * Math.PI / 180 * (omega2));
double korn42 = (6 + 0 * tete) * Math.sin(-2 * Math.PI
/ 180 * (debes) + 0 * Math.PI / 180 * (embes) + 2 *
Math.PI / 180 * (emaks) + 2 * Math.PI / 180 * (efbes)

```



```

+ 2 * Math.PI / 180 * (omega2));
double korn43 = (6 + 0 * tete) * Math.sin(-2 * Math.PI
/ 180 * (debes) + 0 * Math.PI / 180 * (embes) + 1 *
Math.PI / 180 * (emaks) + 2 * Math.PI / 180 * (efbes)
+ 1 * Math.PI / 180 * (omega2));
double korn44 = (-6 + 0 * tete) * Math.sin(2 * Math.PI
/ 180 * (debes) + 0 * Math.PI / 180 * (embes) + -2 *
Math.PI / 180 * (emaks) + 0 * Math.PI / 180 * (efbes)
+ 1 * Math.PI / 180 * (omega2));
double korn45 = (-6 + 0 * tete) * Math.sin(2 * Math.PI
/ 180 * (debes) + 0 * Math.PI / 180 * (embes) + 0 *
Math.PI / 180 * (emaks) + 0 * Math.PI / 180 * (efbes)
+ 1 * Math.PI / 180 * (omega2));
double korn46 = (5 + 0 * tete) * Math.sin(0 * Math.PI
/ 180 * (debes) + -1 * Math.PI / 180 * (embes) + 1 *
Math.PI / 180 * (emaks) + 0 * Math.PI / 180 * (efbes)
+ 0 * Math.PI / 180 * (omega2));
double korn47 = (-5 + 0 * tete) * Math.sin(-2 *
Math.PI / 180 * (debes) + -1 * Math.PI / 180 * (embes)
+ 0 * Math.PI / 180 * (emaks) + 2 * Math.PI / 180 *
(efbes) + 1 * Math.PI / 180 * (omega2));
double jumlahkorn = korn1 + korn2 + korn3 + korn4 +
korn5 + korn6 + korn7 + korn8 + korn9 + korn10 +
korn11 + korn12 + korn13 + korn14 + korn15 + korn16 +
korn17 + korn18 + korn19 + korn20
      + korn21 + korn22 + korn23 + korn24 + korn25 +
korn26 + korn27 + korn28 + korn29 + korn30
      + korn31 + korn32 + korn33 + korn34 + korn35 +
korn36 + korn37 + korn38 + korn39 + korn40
      + korn41 + korn42 + korn43 + korn44 + korn45 +

//lanjutan
double segpe = jumlahkorn / 10000 / 3600;

//koreksi delta epsilon
double kord1 = (92025 + 8.9 * tete) * Math.cos(0 *
Math.PI / 180 * (debes) + 0 * Math.PI / 180 * (embes)
+ 0 * Math.PI / 180 * (emaks) + 0 * Math.PI / 180 *
(efbes) + 1 * Math.PI / 180 * (omega2));
double kord2 = (5736 + -3.1 * tete) * Math.cos(-2 *
Math.PI / 180 * (debes) + 0 * Math.PI / 180 * (embes)
+ 0 * Math.PI / 180 * (emaks) + 2 * Math.PI / 180 *
(efbes) + 2 * Math.PI / 180 * (omega2));
double kord3 = (977 + -0.5 * tete) * Math.cos(0 *
Math.PI / 180 * (debes) + 0 * Math.PI / 180 * (embes)
+ 0 * Math.PI / 180 * (emaks) + 2 * Math.PI / 180 *
(efbes) + 2 * Math.PI / 180 * (omega2));
double kord4 = (-895 + 0.5 * tete) * Math.cos(0 *
Math.PI / 180 * (debes) + 0 * Math.PI / 180 * (embes)
+ 0 * Math.PI / 180 * (emaks) + 0 * Math.PI / 180 *
(efbes) + 2 * Math.PI / 180 * (omega2));

```

```

double kord5 = (54 + -0.1 * tete) * Math.cos(0 *
Math.PI / 180 * (debes) + 1 * Math.PI / 180 * (embes)
+ 0 * Math.PI / 180 * (emaks) + 0 * Math.PI / 180 *
(efbes) + 0 * Math.PI / 180 * (omega2));
double kord6 = (224 + -0.6 * tete) * Math.cos(-2 *
Math.PI / 180 * (debes) + 1 * Math.PI / 180 * (embes)
+ 0 * Math.PI / 180 * (emaks) + 2 * Math.PI / 180 *
(efbes) + 2 * Math.PI / 180 * (omega2));
double kord7 = (129 + -0.1 * tete) * Math.cos(0 *
Math.PI / 180 * (debes) + 0 * Math.PI / 180 * (embes)
+ 1 * Math.PI / 180 * (emaks) + 2 * Math.PI / 180 *
(efbes) + 2 * Math.PI / 180 * (omega2));
double kord8 = (-95 + 0.3 * tete) * Math.cos(-2 *
Math.PI / 180 * (debes) + -1 * Math.PI / 180 * (embes)
+ 0 * Math.PI / 180 * (emaks) + 2 * Math.PI / 180 *
(efbes) + 2 * Math.PI / 180 * (omega2));
double kord9 = (-7 + 0 * tete) * Math.cos(0 * Math.PI
/ 180 * (debes) + 0 * Math.PI / 180 * (embes) + 1 *
Math.PI / 180 * (emaks) + 0 * Math.PI / 180 * (efbes)
+ 0 * Math.PI / 180 * (omega2));
double kord10 = (200 + 0 * tete) * Math.cos(0 *
Math.PI / 180 * (debes) + 0 * Math.PI / 180 * (embes)
+ 0 * Math.PI / 180 * (emaks) + 2 * Math.PI / 180 *
(efbes) + 1 * Math.PI / 180 * (omega2));
double kord11 = (-70 + 0 * tete) * Math.cos(-2 *
Math.PI / 180 * (debes) + 0 * Math.PI / 180 * (embes)
+ 0 * Math.PI / 180 * (emaks) + 2 * Math.PI / 180 *
(efbes) + 1 * Math.PI / 180 * (omega2));
double kord12 = (-53 + 0 * tete) * Math.cos(0 *
Math.PI / 180 * (debes) + 0 * Math.PI / 180 * (embes)
+ -1 * Math.PI / 180 * (emaks) + 2 * Math.PI / 180 *
(efbes) + 2 * Math.PI / 180 * (omega2));
double kord17 = (-24 + 0 * tete) * Math.cos(0 *
Math.PI / 180 * (debes) + 0 * Math.PI / 180 * (embes)
+ -2 * Math.PI / 180 * (emaks) + 2 * Math.PI / 180 *
(efbes) + 1 * Math.PI / 180 * (omega2));
double kord18 = (16 + 0 * tete) * Math.cos(2 * Math.PI
/ 180 * (debes) + 0 * Math.PI / 180 * (embes) + 0 *
Math.PI / 180 * (emaks) + 2 * Math.PI / 180 * (efbes)
+ 2 * Math.PI / 180 * (omega2));
double kord19 = (13 + 0 * tete) * Math.cos(0 * Math.PI
/ 180 * (debes) + 0 * Math.PI / 180 * (embes) + 2 *
Math.PI / 180 * (emaks) + 2 * Math.PI / 180 * (efbes)
+ 2 * Math.PI / 180 * (omega2));
double kord20 = (-12 + 0 * tete) * Math.cos(-2 *
Math.PI / 180 * (debes) + 0 * Math.PI / 180 * (embes)
+ 1 * Math.PI / 180 * (emaks) + 2 * Math.PI / 180 *
(efbes) + 2 * Math.PI / 180 * (omega2));
double kord21 = (-10 + 0 * tete) * Math.cos(0 *
Math.PI / 180 * (debes) + 0 * Math.PI / 180 * (embes)
+ -1 * Math.PI / 180 * (emaks) + 2 * Math.PI / 180 *

```

```

(efbes) + 1 * Math.PI / 180 * (omega2));
double kord22 = (-8 + 0 * tete) * Math.cos(2 * Math.PI
/ 180 * (debes) + 0 * Math.PI / 180 * (embes) + -1 *
Math.PI / 180 * (emaks) + 0 * Math.PI / 180 * (efbes)
+ 1 * Math.PI / 180 * (omega2));
double kord23 = (7 + 0 * tete) * Math.cos(-2 * Math.PI
/ 180 * (debes) + 2 * Math.PI / 180 * (embes) + 0 *
Math.PI / 180 * (emaks) + 2 * Math.PI / 180 * (efbes)
+ 2 * Math.PI / 180 * (omega2));
double kord24 = (9 + 0 * tete) * Math.cos(0 * Math.PI
/ 180 * (debes) + 1 * Math.PI / 180 * (embes) + 0 *
Math.PI / 180 * (emaks) + 0 * Math.PI / 180 * (efbes)
+ 1 * Math.PI / 180 * (omega2));
double kord25 = (7 + 0 * tete) * Math.cos(-2 * Math.PI
/ 180 * (debes) + 0 * Math.PI / 180 * (embes) + 1 *
Math.PI / 180 * (emaks) + 0 * Math.PI / 180 * (efbes)
+ 1 * Math.PI / 180 * (omega2));
double kord26 = (6 + 0 * tete) * Math.cos(0 * Math.PI
/ 180 * (debes) + -1 * Math.PI / 180 * (embes) + 0 *
Math.PI / 180 * (emaks) + 0 * Math.PI / 180 * (efbes)
+ 1 * Math.PI / 180 * (omega2));
double kord27 = (5 + 0 * tete) * Math.cos(2 * Math.PI
/ 180 * (debes) + 0 * Math.PI / 180 * (embes) + -1 *
Math.PI / 180 * (emaks) + 2 * Math.PI / 180 * (efbes)
+ 1 * Math.PI / 180 * (omega2));
double kord28 = (3 + 0 * tete) * Math.cos(2 * Math.PI
/ 180 * (debes) + 0 * Math.PI / 180 * (embes) + 1 *
Math.PI / 180 * (emaks) + 2 * Math.PI / 180 * (efbes)
+ 2 * Math.PI / 180 * (omega2));
double kord29 = (-3 + 0 * tete) * Math.cos(0 * Math.PI
/ 180 * (debes) + 1 * Math.PI / 180 * (embes) + 0 *
Math.PI / 180 * (emaks) + 2 * Math.PI / 180 * (efbes)
+ 2 * Math.PI / 180 * (omega2));
double kord30 = (3 + 0 * tete) * Math.cos(0 * Math.PI
/ 180 * (debes) + -1 * Math.PI / 180 * (embes) + 0 *
Math.PI / 180 * (emaks) + 2 * Math.PI / 180 * (efbes)
+ 2 * Math.PI / 180 * (omega2));
double kord31 = (3 + 0 * tete) * Math.cos(2 * Math.PI
/ 180 * (debes) + 0 * Math.PI / 180 * (embes) + 0 *
Math.PI / 180 * (emaks) + 2 * Math.PI / 180 * (efbes)
+ 1 * Math.PI / 180 * (omega2));
double kord32 = (-3 + 0 * tete) * Math.cos(-2 *
Math.PI / 180 * (debes) + 0 * Math.PI / 180 * (embes)
+ 2 * Math.PI / 180 * (emaks) + 2 * Math.PI / 180 *
(efbes) + 2 * Math.PI / 180 * (omega2));
double kord33 = (-3 + 0 * tete) * Math.cos(-2 *
Math.PI / 180 * (debes) + 0 * Math.PI / 180 * (embes)
+ 1 * Math.PI / 180 * (emaks) + 2 * Math.PI / 180 *
(efbes) + 1 * Math.PI / 180 * (omega2));
double kord34 = (3 + 0 * tete) * Math.cos(2 * Math.PI
/ 180 * (debes) + 0 * Math.PI / 180 * (embes) + -2 *

```

```

Math.PI / 180 * (emaks) + 0 * Math.PI / 180 * (efbes)
+ 1 * Math.PI / 180 * (omega2));
double kord35 = (3 + 0 * tete) * Math.cos(2 * Math.PI
/ 180 * (debes) + 0 * Math.PI / 180 * (embes) + 0 *
Math.PI / 180 * (emaks) + 0 * Math.PI / 180 * (efbes)
+ 1 * Math.PI / 180 * (omega2));
Math.PI / 180 * (omega2));
double jumlahkord = kord1 + kord2 + kord3 + kord4 +
kord5 + kord6 + kord7 + kord8 + kord9 + kord10
      + kord11 + kord12 + kord13 + kord14 + kord15 +
kord16 + kord17 + kord18 + kord19 + kord20
      + kord21 + kord22 + kord23 + kord24 + kord25 +
kord26 + kord27 + kord28 + kord29 + kord30
      + kord31 + kord32 + kord33 + kord34 + kord35 +
kord36 + kord37 + kord38;

//lanjutan
double segep = jumlahkord / 10000 / 3600;

double koraberasi = -20.4898 / erbes / 3600;
double segdel = 3548.193 + 118.568 * Math.sin(87.5287
+ 359993.7286 * tautau) + 2.476
      * Math.sin(85.0561 + 719987.4571 * tautau) +
1.376 * Math.sin(27.8502 + 4452671.1152 * tautau)
      + 0.119 * Math.sin(73.1375 + 450368.8564 *
tautau) + 0.114
      * Math.sin(337.2264 + 329644.6718 * tautau) +
0.086
      * Math.sin(222.54 + 659289.3436 * tautau) +
0.078 * Math.sin(162.8136 + 9224659.7915 * tautau)
      + 0.01 * tautau * Math.sin(328.517 +
1079981.1857 * tautau) + 0.309 * Math.pow(tautau, 2)
      * Math.sin(241.4518 + 359993.7286 * tautau) +
0.021 * Math.pow(tautau, 2) * Math.sin(205.0482 +
719987.4571 * tautau)
      + 0.004 * Math.pow(tautau, 2) *
Math.sin(297.861 + 4452671.1152 * tautau) + 0.01 *
Math.pow(tautau, 3)
      * Math.sin(154.7066 + 359993.7286 * tautau);
double koraberasi2 = -0.005775518 * erbes * segdel /
3600;
double delta2 = MOD((owe2 + segitiga + koraberasi2),
360);
double epsilon3 = epsilon2 + segep;
double segbet = 0.03916 * (Math.cos(Math.PI / 180 *
(iaksen / 3600)) - Math.sin(Math.PI / 180 * (iaksen /
3600)));

//koreksi B0
double bnoll = 280 * Math.cos(3.199 + 84334.662 *

```

```

tautau);
double bnol2 = 102 * Math.cos(5.422 + 5507.553 *
tautau);
double bnol3 = 80 * Math.cos(3.88 + 5223.69 * tautau);
double bnol4 = 44 * Math.cos(3.7 + 2352.87 * tautau);
double bnol5 = 32 * Math.cos(4 + 1577.34 * tautau);
double koreksibnol = bnol1 + bnol2 + bnol3 + bnol4 +
bnol5;

//koreksi B1
double bsatu1 = 9 * Math.cos(3.9 + 5507.55 * tautau);
double bsatu2 = 6 * Math.cos(1.73 + 5223.69 * tautau);
double koreksibsatu = bsatu1 + bsatu2;

//lanjutan
double bebes = (180 / Math.PI * ((koreksibnol +
koreksibsatu * tautau) / Math.pow(10, 8))) * 3600 * -
1;
double beta = segbet + bebes;
double proalpha1 = Math.sin(Math.PI / 180 * (delta2))
* Math.cos(Math.PI / 180 * (epsilon3 + 0.00256
* Math.cos(Math.PI / 180 * (omega2)))) -
Math.tan(Math.PI / 180 * (beta / 3600)) *
Math.sin(Math.PI / 180 * (epsilon3
+ 0.00256 * Math.cos(Math.PI / 180 *
(omega2))));
double proalpha2 = Math.cos(Math.PI / 180 * (delta2));
double alpha2 = MOD((180 / Math.PI *
(Math.atan2(proalpha1, proalpha2))), 360);
double prodekm = Math.sin(Math.PI / 180 * (beta /
3600)) * Math.cos(Math.PI / 180 * (epsilon3))
+ Math.cos(Math.PI / 180 * (beta / 3600)) *
Math.sin(Math.PI / 180 * (epsilon3)) *
Math.sin(Math.PI / 180 * (delta2));
double deklinasi = 180 / Math.PI *
(Math.asin(prodekm));
double elnol = MOD((280.4664567 + 360007.6982779 *
tautau + 0.03032028
* Math.pow(tautau, 2) + Math.pow(tautau, 3) /
49931 - Math.pow(tautau, 4) / 15299
- Math.pow(tautau, 5) / 1988000), 360);
double equationo = (elnol - 0.0057183 - alpha2 + segpe
* Math.cos(Math.PI / 180 * (segpe)));
Double EoT;
if (equationo > 100) {
    EoT = (-360 + equationo) * 4;
} else {
    EoT= equationo * 4;
}
double sdm = 959.63 / 60 / erbes;

```

```

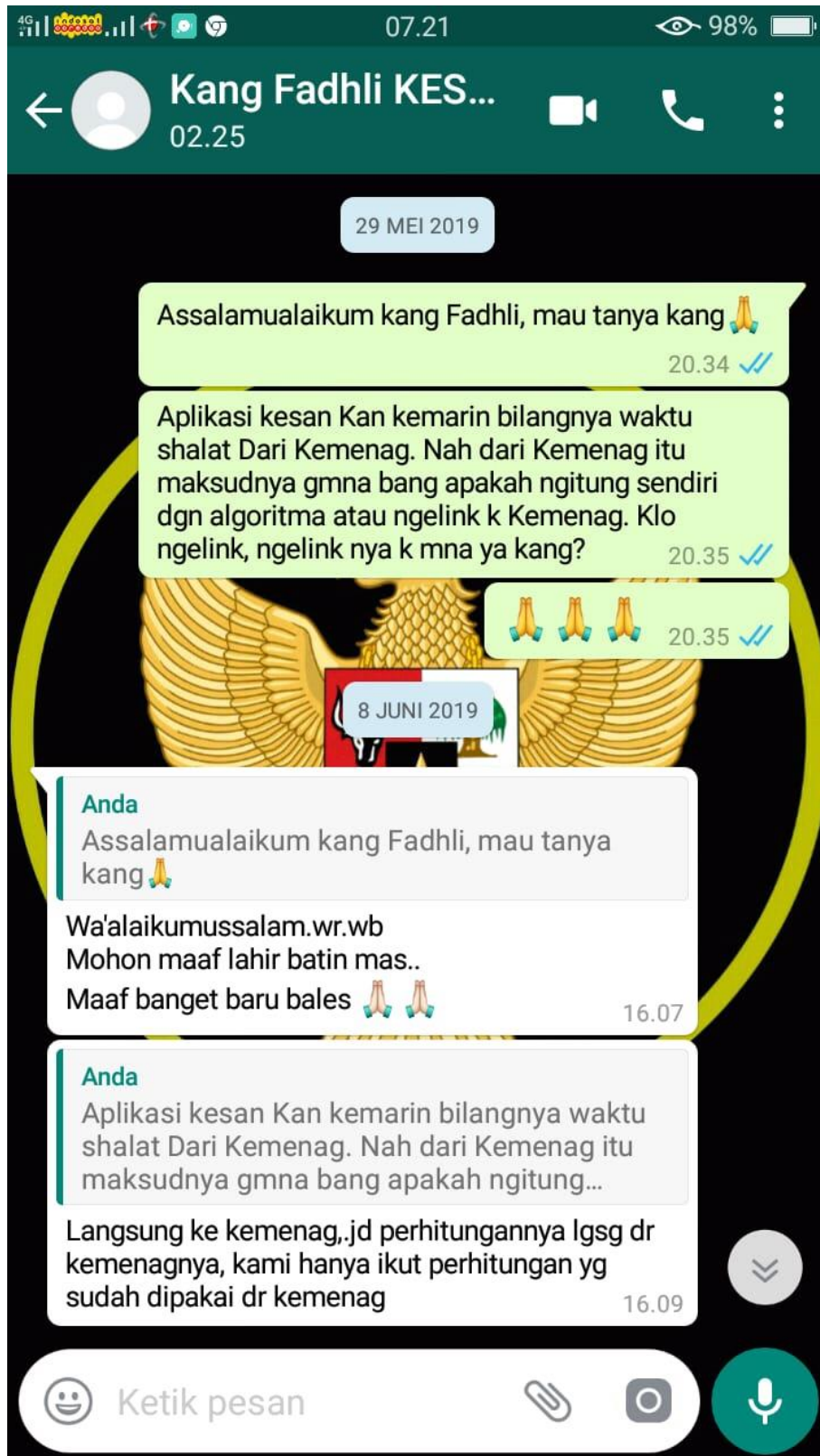
//menghitung waktu sholat
double dhuhur = 12 - EoT / 60 + ((tz * 15) - bt) / 15;
double zm = Math.abs(deklinasi - lt);
double ha = 180 / Math.PI * Math.atan((1 /
(Math.tan(zm * Math.PI / 180) + 1)));
double sudutwaktu = (180 / Math.PI *
Math.acos(Math.sin(ha * Math.PI / 180) / Math.cos(lt *
Math.PI / 180) / Math.cos(deklinasi * Math.PI / 180) -
Math.tan(lt
    * Math.PI / 180) * Math.tan(deklinasi *
Math.PI / 180))) / 15;
double ashar = 12 + sudutwaktu - EoT / 60 + (105 - bt)
/ 15;
double refrak = 0.575;
double semidi = 0.2667;
double kerendah = 0.02933333333 * Math.sqrt(tg);
double tinggimatahari = -(refrak + semidi + kerendah);
double sudutmataharimaghrib = (180 / Math.PI *
Math.acos(Math.sin(tinggimatahari * Math.PI / 180) /
Math.cos(lt * Math.PI / 180)
    / Math.cos(deklinasi * Math.PI / 180) -
Math.tan(lt * Math.PI / 180) * Math.tan(deklinasi *
Math.PI / 180))) / 15;
maghrib = 12 + sudutmataharimaghrib - EoT / 60 + (105
- bt) / 15;
double tinggimatahariisya = -17.0 + tinggimatahari;
double sudutmatahariisya = (180 / Math.PI *
Math.acos(Math.sin(tinggimatahariisya * Math.PI / 180)
/ Math.cos(lt * Math.PI / 180) / Math.cos(deklinasi *
Math.PI / 180) - Math.tan(lt * Math.PI / 180) *
Math.tan(deklinasi * Math.PI / 180))) / 15;
isya = 12 + sudutmatahariisya - EoT / 60 + (105 - bt)
/ 15;
double tinggimataharishubuh = -19.0 + tinggimatahari;
double sudutmataharishubuh = (180 / Math.PI *
Math.acos(Math.sin(tinggimataharishubuh * Math.PI /
180) / Math.cos(lt * Math.PI / 180) /
Math.cos(deklinasi * Math.PI / 180) - Math.tan(lt *
Math.PI / 180) * Math.tan(deklinasi * Math.PI / 180)))
/ 15;
double shubuh = 12 - sudutmataharishubuh - EoT / 60 +
(105 - bt) / 15;

```

Lampiran 2

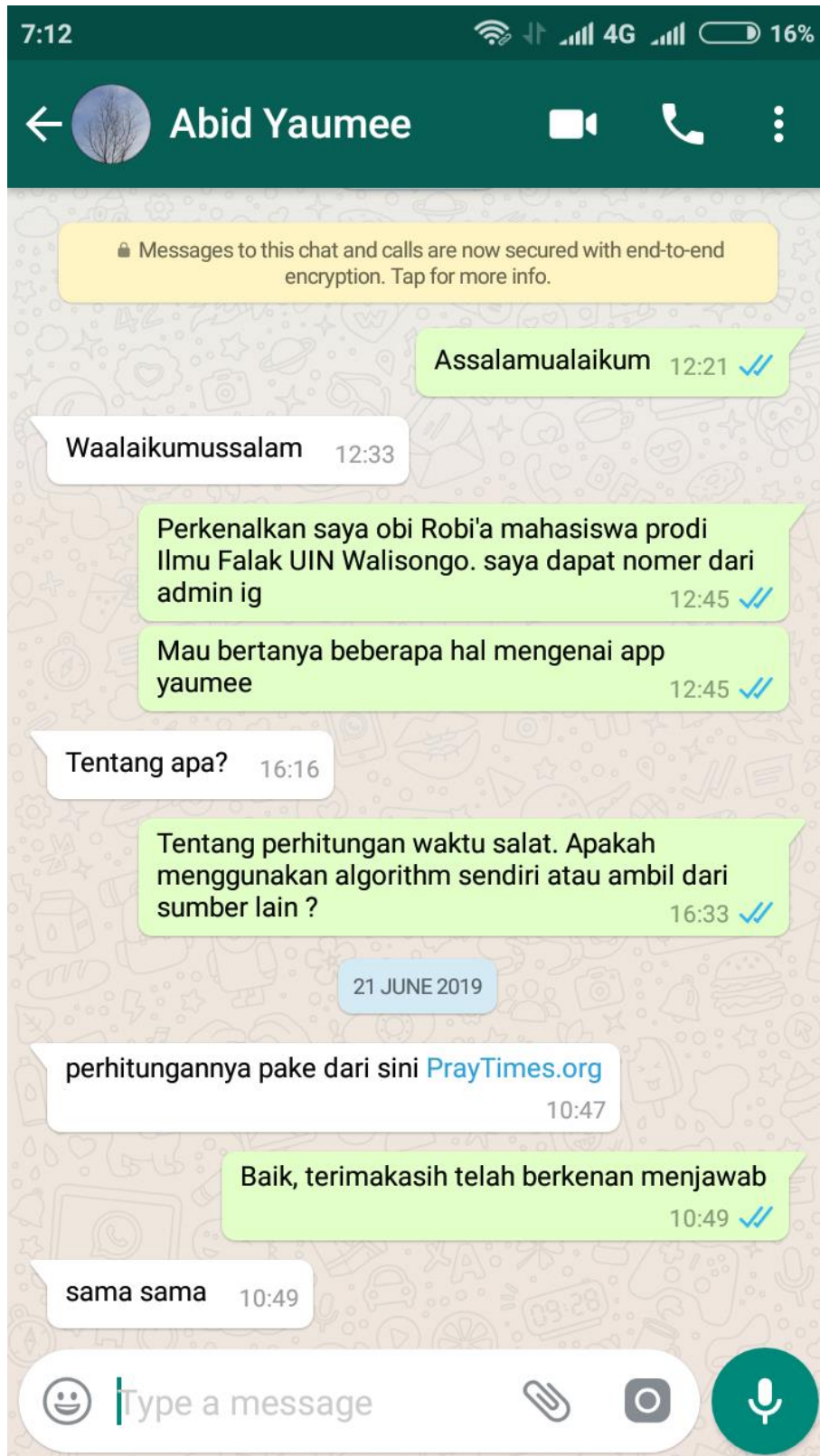
Wawancara 2 (8 Juni 2019)

Narasumber : Fadhli (Tim IT Aplikasi Kesan)



Wawancara 3 (21 Juni 2019)

Narasumber : Abid (Tim IT Aplikasi Yaumees)



Lampiran 4

Wawancara (20 Mei 2019)

Narasumber : Pa Devid (

Pertanyaan : Jam atom jenis apa yang dimiliki BMKG ?

Jawaban : Jam atom cesium tipe 431.

Pertanyaan : Apa alat yang digunakan untuk menyiarkan jam atom ?

Jawaban : Untuk menyiarkan jam ini digunakan *Onogo time coder* dengan kode tertentu untuk disiarkan ke RRI, TVRI dan Radio Pantai dan untuk sampai saat ini kami belum kerjasama dengan televisi swasta. Tapi kedepan akan dilakukan kerjasama dengan bank, bursa efek yang juga memerlukan untuk transaksi keuangan

Pertanyaan : Seberapa akurat jam atom ini?

Jawaban : Jam atom ini masa peluruhannya sekitar 8 tahun jadi sangat kecil sekali untuk perubahan detiknya.

Pertanyaan : Apakah sinyal *provider* mempengaruhi keakuratan jam atom BMKG ?

Jawaban : Tentunya *provider* sangat mempengaruhi keakuratan karena setiap *provider* memiliki kekuatan sinyal yang berbeda. Untuk itu diperlukan adanya *refresh*.

Pertanyaan : Bagaimana seandainya jika jam atom ini mati ?

Jawaban : Kami memiliki jam *backup* sehingga ketika jam yang satu mati akan bisa diganti dengan jam lain.

Pertanyaan : Apakah BMKG satu-satunya di Indonesia yang mempunyai jam atom ?

Jawaban : sebenarnya Lipi juga mempunyai jam atom hanya saja menurut undang-undang yang memiliki wewenang untuk menyiarkan adalah BMKG.

Lampiran 5

Rumus *polynomial expression for ΔT*

Sebelum tahun -500, maka :

$$\Delta T = -20 + 32 * u ^ 2$$

$$\text{di mana: } u = (y-1820) / 100$$

Tahun -500 s/d +500

$$\Delta T = 10583.6 - 1014.41 * u + 33.78311 * u ^ 2 - 5.952053 * u ^ 3 \\ - 0.1798452 * u ^ 4 + 0.022174192 * u ^ 5 + 0.0090316521 * u ^ 6$$

$$\text{dimana: } u = y / 100$$

Tahun +500 s/d +1600

$$\Delta T = 1574.2 - 556.01 * u + 71.23472 * u ^ 2 + 0.319781 * u ^ 3 \\ - 0.8503463 * u ^ 4 - 0.005050998 * u ^ 5 + 0.0083572073 * u ^ 6$$

$$\text{di mana: } u = (y-1000) / 100$$

Tahun +1600 s/d +1700

$$\Delta T = 120 - 0,9808 * t - 0,01532 * t ^ 2 + t ^ 3/7129$$

$$\text{di mana: } t = y - 1600$$

Tahun +1700 s/d +1800, hitung:

$$\Delta T = 8,83 + 0,1603 * t - 0,0059285 * t ^ 2 + 0,00013336 * t ^ 3 - t ^ 4/1174000$$

$$\text{di mana: } t = y - 1700$$

Tahun +1800 s/d +1860

$$\Delta T = 13,72 - 0,332447 * t + 0,0068612 * t^2 + 0,0041116 * t^3 - 0,00037436 * t^4 + 0,0000121272 * t^5 - 0,0000001699 * t^6 + 0,000000000875 * t^7$$

di mana: $t = y - 1800$

Tahun 1860 s/d 1900

$$\Delta T = 7.62 + 0.5737 * t - 0.251754 * t^2 + 0.01680668 * t^3 - 0.0004473624 * t^4 + t^5/233174$$

di mana: $t = y - 1860$

Tahun 1900 s/d 1920

$$\Delta T = -2,79 + 1,494119 * t - 0,0598939 * t^2 + 0,0061966 * t^3 - 0,000197 * t^4$$

di mana: $t = y - 1900$

Tahun 1920 s/d 1941

$$\Delta T = 21,20 + 0,84493 * t - 0,076100 * t^2 + 0,0020936 * t^3$$

di mana: $t = y - 1920$

Tahun 1941 s/d 1961

$$\Delta T = 29,07 + 0,407 * t - t^2/233 + t^3/2547$$

di mana: $t = y - 1950$

Tahun 1961 s/d 1986

$$\Delta T = 45,45 + 1,067 * t - t^2/260 - t^3/718$$

di mana: $t = y - 1975$

Tahun 1986 s/d 2005

$$\Delta T = 63,86 + 0,3345 * t - 0,060374 * t^2 + 0,0017275 * t^3 + 0,000651814$$

$$* t^4$$

$$+ 0,00002373599 * t^5$$

di mana: $t = y - 2000$

DAFTAR RIWAYAT HIDUP

Nama Lengkap : Obi Robi'a Al Aslami
Tempat Tanggal Lahir : Majalengka, 17 Juli 1997
Agama : Islam
Nama Orang Tua : A. Sudja'i HS, Aliyah
Alamat : Jl. S. Parman No. RT. 03 RW. 04 Blok Kamis Desa Bongas
Wetan Kec. Sumberjaya Majalengka
No Hp : 089503238470
Email : obirobia.alaslami@gmail.com

Riwayat Pendidikan :

- a. Formal
 1. RA Istqomah Bongas Wetan 2003
 2. SDN Bongas Wetan III lulus 2009.
 3. MTs Al-Ishlah Persis Majalengka lulus 2012.
 4. MA Al-Ishlah Persis Majalengka lulus 2015.
 5. UIN Walisongo Semarang
- b. Non Formal
 1. Pesantren Persatuan Islam 92 Majalengka.
 2. Pesantren Life Skill Daarun Najaah Semarang.
 3. Brilliant English Course Pare.
 4. FullBright English Course Pare.

Pengalaman Organisasi :

1. Ketua Rijalul Ghad Tsanawiyah Pesantren Persatuan Islam 92 Majalengka 2008-2009.
2. Sekretaris Rijalul Ghad Mu'allimien Pesantren Persatuan Islam 92 2011-2012.
3. Koordinator Departemen KOMINFO CSSMoRA UIN Walisongo 2017-2018.
4. Koordinator Bidang Eksternal Astronom Persis 2018-Sekarang.
5. Layouter LPM Zenith CSSMoRA UIN Walisongo periode 2016-2017.
6. Pimpinan Umum LPM Zenith CSSMoRA UIN Walisongo periode 2017-2018.
7. Koordinator Bidang Media mediasantri.id periode 2018-2019
8. Anggota Departemen PSDE CSSMoRA Nasional 2018-2019.