

## BAB II

### LANDASAN TEORI

#### 2.1 Penentuan Waktu Sholat

Perhitungan waktu sholat dapat diketahui melalui dua langkah, yaitu penyediaan data, dan penyediaan rumus-rumus yang digunakan. Setelah data dan rumus tersedia, proses selanjutnya adalah melakukan perhitungan dengan data tersebut (Majelis Tarjih dan Tajdid PP Muhammadiyah, 2009:54).

##### 2.1.1 Penyediaan Data

Data-data yang digunakan untuk menghitung waktu sholat adalah sebagai berikut:

a. Koordinat Lintang Tempat (L)

Lintang tempat adalah jarak sepanjang meridian bumi diukur dari ekuator bumi (katulistiwa) sampai suatu tempat yang bersangkutan. Nilai dari lintang tempat adalah  $0^\circ$  sampai  $90^\circ$ . Lintang tempat bagi tempat-tempat yang berada di belahan bumi utara bertanda positif (+) dan tempat-tempat yang berada di belahan bumi selatan bertanda negatif (-). Dalam astronomi disebut *latitude* dan biasanya digunakan lambang  $\phi$  (*phi*).

b. Koordinat Bujur Tempat (B)

Bujur tempat adalah jarak sepanjang equator bumi dihitung dari meridian yang melewati kota Greenwich sampai meridian yang melewati tempat bersangkutan. Dalam astronomi dikenal dengan nama *longitude* biasanya digunakan lambang  $\lambda$  (*lamda*). Nilai dari bujur tempat adalah  $0^\circ$  sampai  $180^\circ$ . Bagi tempat-tempat yang berada di sebelah barat Greenwich disebut Bujur Barat (BB) dan diberi tanda negatif (-). Sedangkan bagi tempat-tempat yang berada di sebelah timur Greenwich disebut Bujur Timur (BT) dan diberi tanda positif (+).

c. Zona Waktu Tempat ( $Z$ )

Zona Waktu (*Time Zone*) ini dibagi menjadi 24 zona waktu yang berbeda-beda, sesuai letak daerah tersebut. Waktu universal yang menjadi panutan adalah waktu GMT, waktu yang ada di Greenwich, Inggris.

d. Ketinggian Lokasi ( $H$ )

Ketinggian lokasi adalah ketinggian suatu lokasi dihitung dari permukaan laut. Hal ini menentukan waktu kapan terbit dan terbenamnya matahari. Tempat yang berada tinggi di atas permukaan laut akan lebih awal melihat matahari terbit serta lebih akhir melihat matahari terbenam, dibandingkan dengan tempat yang lebih rendah. Ketinggian lokasi ini biasanya menggunakan satuan meter (m).

e. Tanggal ( $D$ ), Bulan ( $M$ ), dan Tahun ( $Y$ )

Tanggal, bulan dan tahun akan sangat berpengaruh pada perhitungan waktu shalat. Kalender yang dianut sekarang adalah kalender gregorian, namun untuk menentukan waktu shalat digunakan kalender julian ( $JD$ ). Untuk itu, dibutuhkan sebuah rumus untuk melakukan konversi dari kalender gregorian ke dalam kalender julian. Rumus tersebut dapat dituliskan sebagai berikut:

$$JD = 1720994,5 + (365,25 \times Y) + (30,6001 \times (M + 1)) + (2 + \frac{Y}{4} - \frac{Y}{100} + D + \frac{12}{24}) \dots\dots\dots(2.1)$$

Jika  $M > 2$ , maka nilai  $M$  dan  $Y$  tidak berubah. Jika  $M = 1$  atau  $2$ , maka nilai  $M$  ditambah 12 sedangkan nilai  $Y$  dikurangi 1. Nilai  $JD$  di atas berlaku untuk pukul 12.00 GMT atau saat tengah hari di Greenwich. Adapun  $JD$  untuk pukul 12.00 waktu lokal, maka  $JD$  pukul 12.00 GMT waktu Greenwich tersebut harus dikurangi dengan  $Z/24$  dimana  $Z$  adalah zona waktu lokal tersebut.

Setelah mendapatkan nilai dari  $JD$  tersebut, dapat dihitung sudut tanggal ( $T$ ) dengan rumus sebagai berikut:

$$T = \frac{2 \times PHI \times (JD - 2451545)}{365,25} \dots\dots\dots(2.2)$$

f. Sudut Deklinasi Matahari (Delta)

Deklinasi matahari ialah jarak matahari dari lingkaran ekuator diukur sepanjang lingkaran waktu yang melalui matahari itu hingga ke titik pusat matahari tersebut.

$$\begin{aligned} \Delta = & 0,37877 + 23,264 \times \sin(57,297 \times T - 79,547) + 0,3812 \times \\ & \sin(2 \times 57,297 \times T - 82,682) + 0,17132 \times \sin(3 \times 57,297 \times T - \\ & 59,772) \dots \dots \dots (2.3) \end{aligned}$$

g. Perata Waktu (ET)

Perata waktu (*equation of time*) adalah selisih waktu antara waktu matahari hakiki dengan matahari rata-rata (pertengahan). Peredaran semua harian matahari dari arah timur ke barat itu tidaklah konstan, kadang-kadang cepat kadang-kadang lambat. Keadaan ini diakibatkan oleh percepatan bumi mengelilingi matahari tidak konstan karena bidang edarnya berbentuk elips.

Waktu matahari hakiki ialah waktu peredaran matahari senyatanya, sedangkan waktu matahari pertengahan adalah waktu peredaran semua matahari diandaikan ia beredar dengan konstan sebagaimana terlihat pada jam yang ada.

Rumus yang digunakan untuk menghitung ET adalah sebagai berikut:

$$\begin{aligned} ET = & \left( - \left( 1789 + 237 \times \frac{(JD-245145)}{36525} \right) \times \sin \left( 280,46607 + 36000,7689 \times \right. \right. \\ & \left. \left. \frac{(JD-245145)}{36525} \right) - \left( 7146 - 62 \times \frac{(JD-245145)}{36525} \right) \times \cos \left( 280,46607 + 36000,7689 \times \right. \right. \\ & \left. \left. \frac{(JD-245145)}{36525} \right) + \left( 9934 - 14 \times \frac{(JD-245145)}{36525} \right) \times \sin \left( 2 \times 280,46607 + \right. \right. \\ & \left. \left. 36000,7689 \times \frac{(JD-245145)}{36525} \right) - \left( 29 + 5 \times \frac{(JD-245145)}{36525} \right) \times \cos \left( 2 \times 280,46607 + \right. \right. \\ & \left. \left. 36000,7689 \times \frac{(JD-245145)}{36525} \right) + \left( 74 + 10 \times \frac{(JD-245145)}{36525} \right) \times \sin \left( 3 \times 280,46607 + \right. \right. \\ & \left. \left. 36000,7689 \times \frac{(JD-245145)}{36525} \right) + \left( 320 - 4 \times \frac{(JD-245145)}{36525} \right) \times \cos \left( 3 \times 280,46607 + \right. \right. \\ & \left. \left. 36000,7689 \times \frac{(JD-245145)}{36525} \right) - 212 \times \sin \left( 4 \times 280,46607 + 36000,7689 \times \right. \right. \end{aligned}$$

$$\left(\frac{JD-245145}{36525}\right)/1000\dots\dots\dots(2.4)$$

h. Tetapan Panjang Bayangan Ashar (KA)

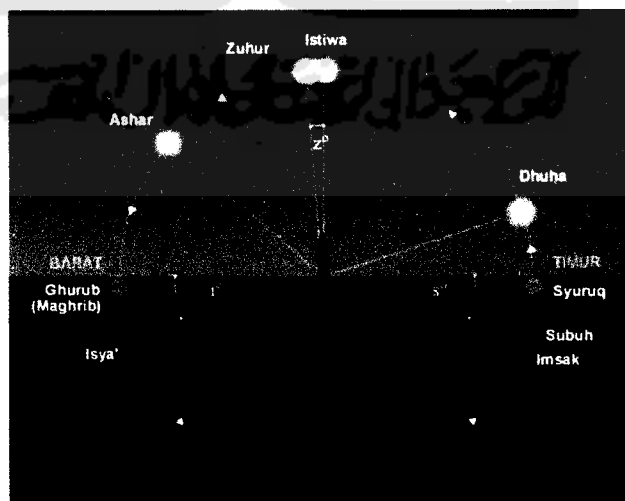
Tetapan panjang bayangan waktu sholat Ashar ini ada dua pendapat, yaitu pendapat mazhab Syafi'i dan pendapat madzhab Hanafi. Pendapat mazhab Syafi'i menyatakan panjang bayangan benda saat Ashar = tinggi benda + panjang bayangan saat Duhur. Sementara pendapat mazhab Hanafi menyatakan panjang bayangan benda saat Ashar = dua kali tinggi benda + panjang bayangan saat Duhur.

i. *Altitude* matahari waktu Subuh dan Isya'

Dalam referensi standar astronomi, sudut *altitude* untuk waktu Subuh dan Isya' adalah 18 derajat di bawah ufuk, atau sama dengan minus 18 derajat. Namun demikian ada beberapa pendapat mengenai sudut *altitude* matahari di bawah ufuk saat Subuh dan Isya. Diantaranya berkisar antara 15 hingga 20 derajat. Dengan demikian, perbedaan sudut yang digunakan akan menyebabkan perbedaan datangnya waktu Subuh dan Isya'.

### 2.1.2 Penyediaan Rumus

Posisi matahari pada saat waktu sholat, dapat dilihat pada Gambar 2.1.



**Gambar 2.1** Posisi Matahari saat Waktu Sholat

Rumus-rumus yang digunakan untuk melakukan perhitungan waktu shalat adalah sebagai berikut:

a. Sholat Dhuhur

$$\text{Waktu shalat Dhuhur} = 12 + Z - \frac{B}{15} - \frac{ET}{60} \dots\dots\dots(2.5)$$

b. Sholat Ashar

$$\text{Waktu shalat Ashar} = \text{Dhuhur} + \frac{(\text{Hour Angle Ashar})}{15} \dots\dots\dots(2.6)$$

c. Sholat Maghrib

$$\text{Waktu shalat Maghrib} = \text{Dhuhur} + \frac{(\text{Hour Angle Maghrib})}{15} \dots\dots\dots(2.7)$$

d. Sholat Isya

$$\text{Waktu shalat Isya}' = \text{Dhuhur} + \frac{(\text{Hour Angle Isya}')}{15} \dots\dots\dots(2.8)$$

e. Sholat Subuh

$$\text{Waktu shalat Subuh} = \text{Dhuhur} - \frac{(\text{Hour Angle Subuh})}{15} \dots\dots\dots(2.9)$$

Rumus-rumus diatas dibutuhkan nilai *hour angle* (HA), sedangkan rumus untuk menghitung nilai HA tersebut adalah sebagai berikut:

$$\cos(HA) = \frac{\sin(\text{altitude}) - \sin(L) \cdot \sin(\Delta)}{\cos(L) \cdot \cos(\Delta)} \dots\dots\dots(2.10)$$

$$HA = \arccos(\cos(HA)) \dots\dots\dots(2.11)$$

Nilai HA diatas dipengaruhi juga oleh nilai variabel *altitude*, rumus untuk menghitung nilai tersebut berbeda untuk setiap waktu shalat. Rumus-rumus yang digunakan adalah sebagai berikut:

a. Sholat Ashar

$$\text{altitude} = \arccot(KA + \tan(|\Delta - L|)) \dots\dots\dots(2.12)$$

Keterangan:

KA = 1, untuk Syafi'i.

KA = 2, untuk Hanafi.

b. Sholat Maghrib

$$\text{altitude} = 0,8333 - 0,0347 \cdot \sqrt{H} \dots\dots\dots(2.13)$$

- c. Sholat Isya'

$$altitude = -(sudut Isya') \dots \dots \dots (2.14)$$

- d. Sholat Subuh

$$altitude = -(sudut Subuh) \dots \dots \dots (2.15)$$

## 2.2 Penentuan Arah Kiblat

Pada saat ini metode yang sering digunakan dalam pengukuran arah kiblat ada tiga macam, yaitu memanfaatkan bayang-bayang kiblat, memanfaatkan arah utara geografis, memperhatikan ketika matahari tepat berada di atas Ka'bah (Majelis Tarjih dan Tajdid PP Muhammadiyah, 2009:41).

Penelitian kali ini akan menggunakan metode kedua, yaitu dengan memanfaatkan arah utara geografis. Pada metode ini, dibutuhkan data sebagai berikut:

- a. Koordinat Lintang Tempat (L)
- b. Koordinat Bujur Tempat (B)
- c. Koordinat Lintang Ka'bah (LK)
- d. Koordinat Bujur Ka'bah (BK)

Koordinat lintang dan bujur Ka'bah yang digunakan adalah  $21^{\circ}25'21.07''$  untuk lintang, dan  $39^{\circ}49'34.37''$  untuk nilai bujurnya (Jannah, 2009:11). Kemudian rumus yang digunakan adalah sebagai berikut (Anugraha, 2009):

$$\tan(K) = \frac{\sin(BK-B)}{\cos(L).\tan(LK)-\sin(L).\cos(B-BK)} \dots \dots \dots (2.16)$$

$$\cos(d) = \sin(L) \cdot \sin(LK) + \cos(L) \cdot \cos(LK) \cdot \cos(B - BK) \dots \dots (2.17)$$

Keterangan:

K = Arah kiblat

d = Jarak antara lokasi dan Ka'bah

### 2.3 Perangkat *iPhone*

*iPhone* adalah perangkat telepon genggam yang diproduksi oleh Apple untuk memenuhi kebutuhan konsumen akan sebuah perangkat yang menggabungkan antara perangkat komunikasi dan perangkat hiburan. *iPhone* adalah perangkat yang menggabungkan antara perangkat ponsel, *iPod*, dan perangkat untuk Internet (Apple Inc.).

*iPhone* generasi pertama dipasarkan pada tanggal 29 Juni 2007 di Amerika. *iPhone* terus berkembang, hingga pada tanggal 11 Juli 2008 Apple meluncurkan *iPhone* 3G, dan pada akhirnya pada tanggal 17 Juni 2009 meluncurkan generasi ketiganya, yaitu *iPhone* 3Gs.

Sampai saat ini, dua generasi terakhir itulah yang masih digemari oleh konsumen. Dua generasi tersebut memiliki fitur yang hampir sama, yaitu memiliki kamera, *bluetooth*, *Wi-Fi*, *accelerometer*, *voice memos*, *landscape keyboard*, dan GPS. Namun, *iPhone* 3Gs lebih unggul dengan fitur-fitur baru seperti *voice control*, *video recording*, *accessibility options*, kompas, dan ketahanan baterai yang melebihi *iPhone* 3G.

### 2.4 Bahasa Pemrograman Objective-C

Objective-C adalah sebuah bahasa pemrograman yang ditulis oleh Brad Cox pada awal tahun 1980. Bahasa pemrograman ini dirancang untuk pemrograman berorientasi obyek yang kompleks. Objective-C didefinisikan sebagai bahasa pemrograman berbasis C yang kecil, namun kuat. Pengembangan ini berdasarkan pada Smalltalk, sebuah bahasa pemrograman berorientasi obyek pertama. Objective-C dirancang untuk memberikan dukungan penuh terhadap pemrograman berorientasi obyek, namun dengan penulisan yang sederhana (Apple Inc., 2009:9).

Objective-C dipilih oleh Apple dalam pembuatan aplikasi pada *Mac*, *iPod* dan *iPhone* karena waktu yang digunakan untuk mengembangkan sebuah aplikasi menjadi lebih cepat jika dibandingkan dengan menggunakan bahasa lain. Hal ini

disebabkan karena terdapat beberapa kemudahan dalam penulisannya, dan memiliki *library* yang lengkap.

Perbedaan Objective-C dengan bahasa pemrograman lainnya adalah dalam pembuatan suatu kelas, diharuskan membuat minimal dua file, yaitu file *interface* dan file implementasinya. File *interface* berekstensi .h dan berisi deklarasi variabel dan fungsi dari kelas yang akan didefinisikan. File implementasi berekstensi .m dan berisi definisi dan implementasi dari fungsi-fungsi yang sudah dideklarasikan sebelumnya pada file *interface*.

Selain itu cara pembuatan *method setter* dan *method getter* pada Objective-C berbeda dengan bahasa pemrograman lainnya. Untuk pembuatan dua *method* tersebut digunakan kode program `@property` dilanjutkan dengan `@synthesize`. Berbeda dengan bahasa pemrograman lainnya yang mengharuskan penulisan *method* tersebut secara manual.

## 2.5 Cocoa Touch

Cocoa merupakan salah satu *Application Programming Interface* (API) dari Apple yang digunakan untuk pembuatan aplikasi berorientasi obyek untuk sistem operasi Mac OS X. Cocoa ini adalah salah satu dari lima API utama yang tersedia untuk Mac OS X. Empat API yang lain adalah Carbon, POSIX, X11, dan Java(Apple Inc.,2010:15).

Cocoa Touch adalah sebuah pengembangan dari Cocoa yang dikhususkan untuk membangun aplikasi pada *iPhone*, *iPod Touch*, dan *iPad*. Cocoa Touch menyediakan lapisan abstraksi dari *iPhone OS*, sebuah sistem operasi untuk *iPhone*, *iPod Touch*, dan *iPad*. Cocoa Touch dikembangkan dengan menggunakan bahasa Objective-C.

Dalam penerapannya, Cocoa dibagi menjadi dua bagian, yaitu *Foundation framework* dan *UIKit framework*. Dua bagian ini tidak dapat dipisahkan, aplikasi tidak akan dapat dijalankan jika kita tidak menggunakan kedua bagian tersebut. Kelas, fungsi, tipe data, dan konstanta dalam *Foundation framework* menggunakan prefiks “NS”, sedangkan dalam *UIKit framework* menggunakan prefiks “UI”.

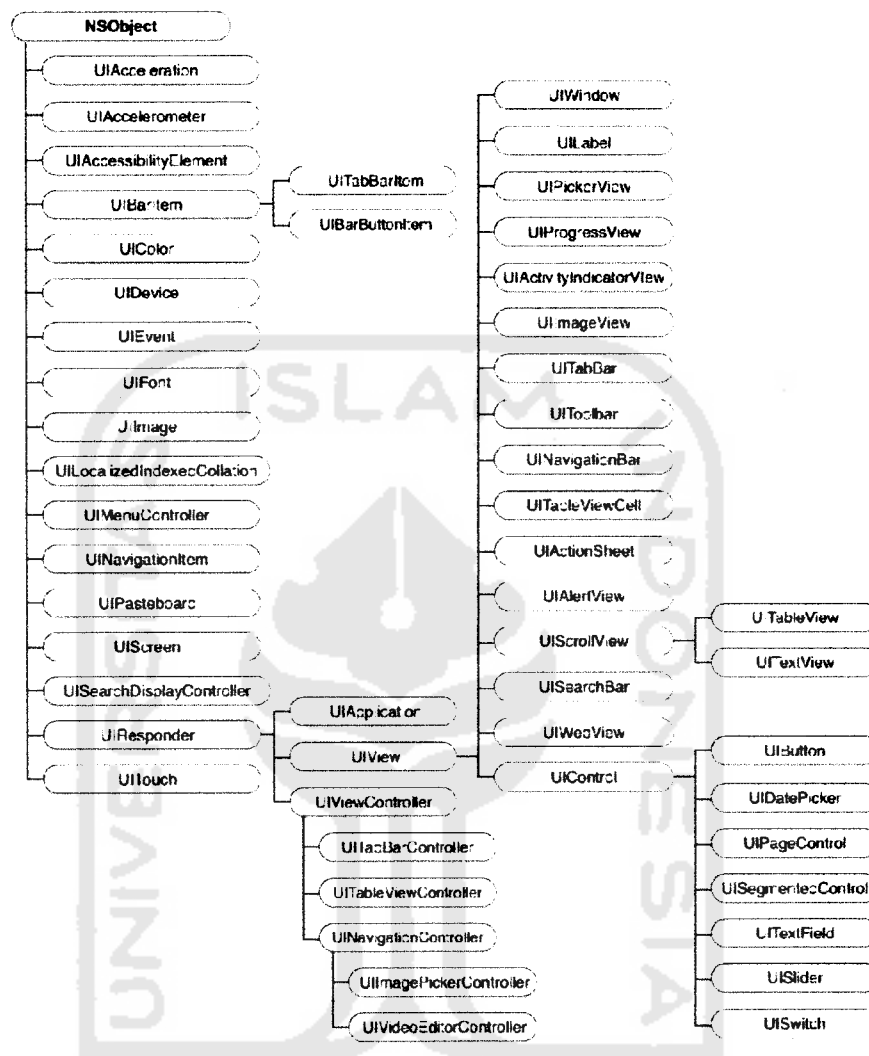


*Foundation framework* digunakan untuk mendefinisikan hal-hal umum dalam pemrograman, seperti manajemen memori dan memanipulasi obyek. Sedangkan *UIKit framework* digunakan untuk mendefinisikan dan memanipulasi *user interface*.

Kelas-kelas yang terdapat dalam *Foundation framework* dan *UIKit framework*, dapat dilihat pada gambar 2.2 dan gambar 2.3 di bawah ini:







Gambar 2.3 Hierarki Kelas pada *UIKit Framework*

Berikut ini adalah kelebihan-kelebihan yang dimiliki oleh Cocoa, yang akan memudahkan user dalam pembuatan aplikasi:

1. *User Interface Object*

Cocoa menyediakan banyak objek-objek untuk pembuatan antarmuka aplikasi. Objek-objek yang sudah tersedia dalam cocoa antara lain adalah *windows*, *text fields*, *image views*, *date pickers*, *progress indicators*, *table views*, *segmented controls*, *sliders*, *buttons*, dan *navigation bars*.

## 2. *Drawing and imaging*

Cocoa menyediakan fitur untuk menggambar dan melakukan manipulasi gambar secara manual. Hal ini dilengkapi dengan fitur untuk menjaga agar manipulasi gambar dapat berjalan dengan mudah, tanpa memikirkan adanya proporsional resolusi gambar, dan sebagainya.

## 3. *Performance*

Untuk menambah performa dari aplikasi yang akan dibangun, Cocoa menyediakan beberapa fitur yang mendukung untuk *multithreading*, *idle-time processing*, *lazy loading of resources*, *memory management*, dan *run-loop manipulation*.

## 4. *Networking*

Cocoa juga menyediakan fitur untuk berkomunikasi dengan *server* menggunakan koneksi *Internet Protocols*, maupun dengan koneksi *sockets*.

## 5. *Data Exchange*

Cocoa memudahkan *user* untuk pertukaran data dalam aplikasi maupun dengan aplikasi yang lainnya menggunakan metode *copy and paste*, maupun *drag and drop*.

## 2.6 GPS

*Global Positioning System* (GPS) adalah sistem navigasi berbasis satelit yang dapat mengirim dan menerima sinyal radio. Perangkat GPS menerima sinyal-sinyal kemudian memberikan sebuah informasi. Teknologi GPS berfungsi untuk menentukan lokasi, kecepatan, dan waktu (Garmin International, 2008:4).

GPS, secara resmi dikenal sebagai NAVSTAR (*Navigation Satellite Timing and Ranging*) *Global Positioning System*, dan awalnya dikembangkan untuk kebutuhan militer. Karena kemampuan navigasinya yang semakin populer dan dapat diakses menggunakan perangkat GPS yang kecil dan murah, pemerintah membuat sistem GPS yang disediakan untuk penggunaan secara umum.

Secara garis besar, prinsip kerja perangkat GPS adalah sebagai berikut (Garmin International, 2008:5):

1. Segmen kontrol melakukan pengecekan sinyal GPS, dan memberikan informasi kepada satelit GPS untuk mendapatkan akurasi maksimal.
2. Perangkat GPS akan mengumpulkan informasi dari satelit GPS.
3. Perangkat GPS melakukan pengecekan kesalahan dari satelit GPS, jika ternyata sinyal yang diterima oleh perangkat GPS mengalami gangguan.
4. Perangkat GPS akan menentukan lokasi, kecepatan, dan waktu.
5. Beberapa perangkat GPS dapat melakukan perhitungan lain seperti jarak perjalanan, jarak ke tujuan, waktu matahari terbit, waktu matahari terbenam, dan sebagainya.
6. Perangkat GPS akan menampilkan informasi pada layar.

