

BAB III

PERANCANGAN DAN IMPLEMENTASI PERANGKAT LUNAK

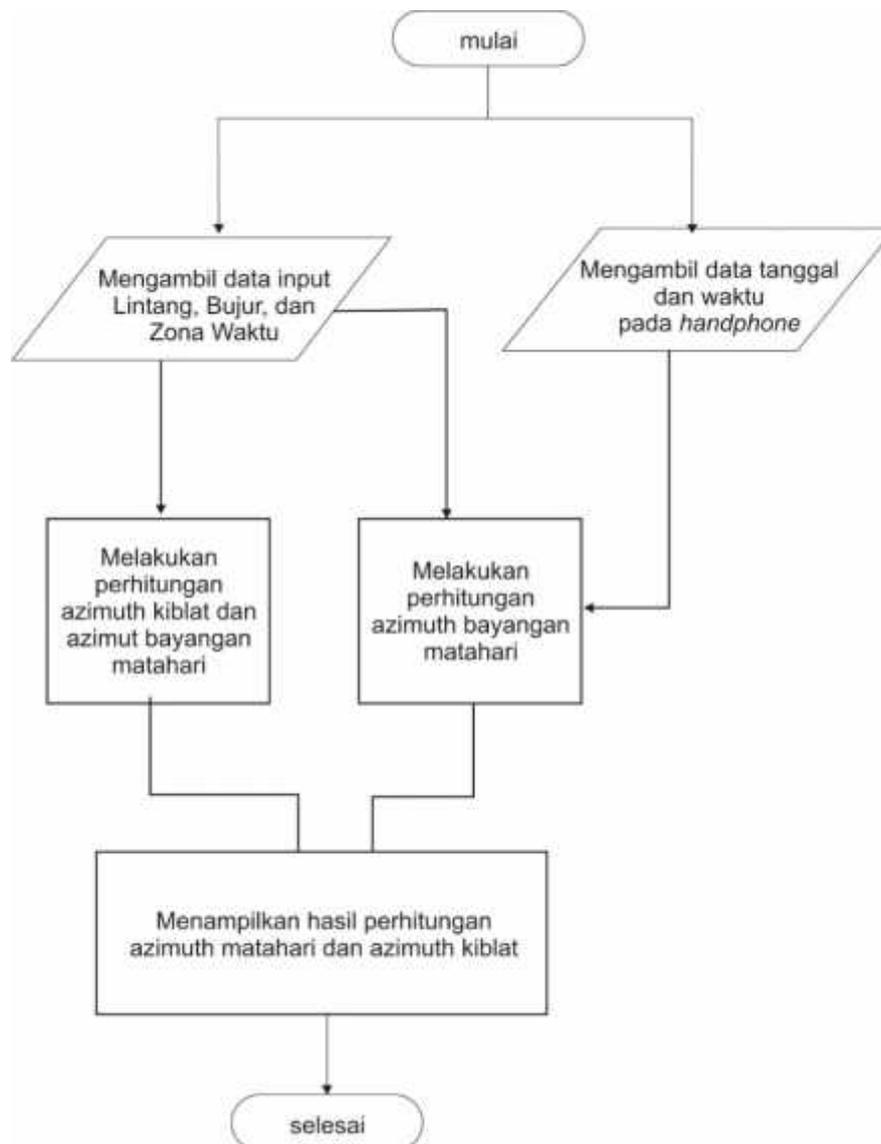
A. Deskripsi Umum Aplikasi Perhitungan *Mizwala Qibla Finder*

Dalam skripsi ini penulis merancang sebuah aplikasi dalam kajian ilmu falak. Aplikasi ini merupakan Aplikasi Perhitungan *Mizwala Qibla Finder*.

Mizwala Qibla Finder yang menjadikan Matahari sebagai acuan dalam menentukan arah kiblat, menjadikan dalam menggunakan alat ini membutuhkan sebuah proses perhitungan untuk mengetahui nilai azimuth bayangan Matahari dan terlebih azimuth kiblat. *Mizwala Qibla Finder* sendiri telah memberikan paket perhitungan dalam bentuk file *Microsoft Office Excel*. Dengan demikian, untuk menjalankannya membutuhkan sebuah komputer atau sejenisnya.

Sebagaimana mana telah disebutkan sebelumnya pada bab pendahuluan, aplikasi ini merupakan pengembangan dari *software Mizwah.xls* yang berfungsi sebagai alat hitung dari alat penentu kiblat dengan *Mizwala Qibla Finder*. Untuk mengembangkan dari alat penentu arah qiblat yang sederhana namun akurat ini, penulis merancang sebuah Aplikasi Perhitungan *Mizwala Qibla Finder* yang dapat dijalankan pada *Mobile Phone*. Sehingga nantinya pengguna *Mizwala Qibla Finder*, tidak perlu membawa laptop atau sejenisnya, cukup membawa *handphone* yang telah diinstal aplikasi ini.

Selain itu aplikasi ini didesain secara semi otomatis, di mana pengguna hanya cukup memasukkan zona waktu, data lintang tempat dan bujur tempat, untuk data waktu dan tanggal telah di *set* otomatis dari *mobile phone*.



Gambar 3.1 Alur sistem Aplikasi Perhitungan *Mizwala Qibla Finder* secara umum.

Dari gambar 3.1 di atas dapat dilihat alur sistem Aplikasi Perhitungan *Mizwala Qibla Finder* secara umum. Terlebih dahulu melakukan input data lintang dan bujur untuk melakukan perhitungan azimuth kiblat. Selanjutnya mengambil data tanggal dan waktu untuk melakukan proses perhitungan guna mendapatkan data astronomis posisi Matahari yang dilanjutkan dengan perhitungan azimuth bayangan Matahari atau Mizwah.

Aplikasi ini dikembangkan dengan menggunakan bahasa pemrograman *Java 2 Micro Edition* (J2ME). Di mana bahasa ini merupakan bahasa pemrograman yang berbasis *object*, yang diciptakan untuk perangkat yang memiliki sumber daya yang terbatas yang terbatas, baik memory maupun baterai. Sehingga untuk menjalankan aplikasi ini dibutuhkan *mobile phone* yang mampu menjalankan bahasa Java atau terdapat *Java Virtual Machine* (JVM).

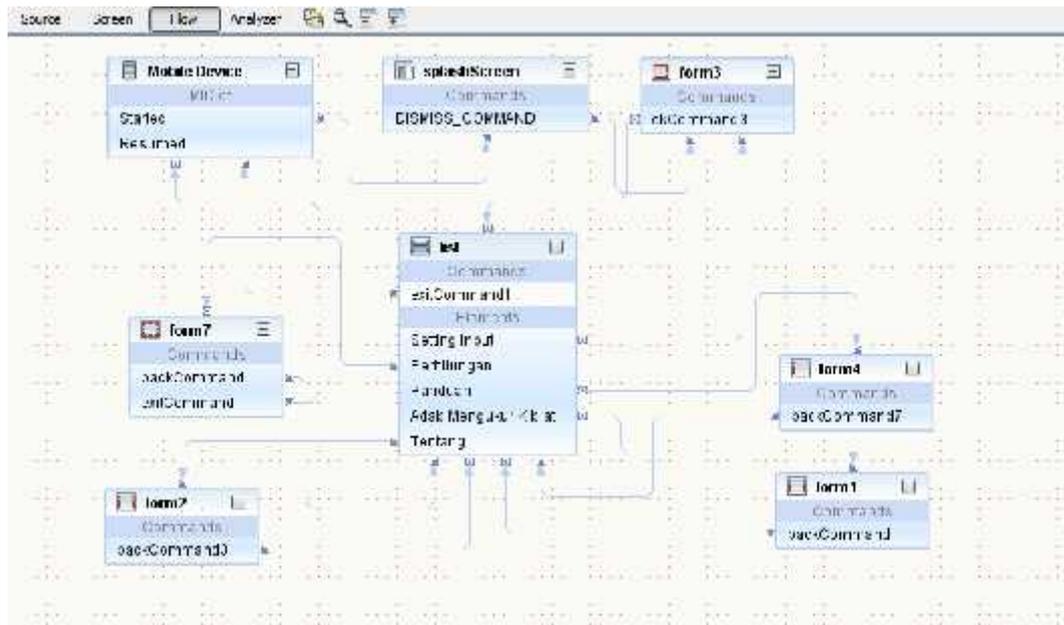
B. Rancangan Perangkat Lunak

Aplikasi yang penulis kembangkan ini merupakan aplikasi berbasis *Java 2 Micro Edition* (J2ME) yang diaplikasikan pada *mobile phone*. Dalam perangkat *mobile* rata-rata telah menyediakan JVM untuk menjalankan aplikasi berbasis Java ini. Aplikasi Perhitungan *Mizwala Qibla Finder* ini didesain dengan semi otomatis dimana untuk data tanggal dan waktu diambil langsung dari perangkat sesuai *realtime*. Dengan demikian pengguna cukup memasukkan data lintang dan bujur serta *time zone*, selanjutnya pengguna akan mendapatkan hasil perhitungan azimuth kiblat dan azimuth bayangan Matahari (*mizwah*) yang langsung diaplikasikan pada *Mizwala Qibla Finder*.

Untuk membangun aplikasi ini penulis menggunakan *Integrated Development Environment* (IDE) atau lingkungan pengembangan bernama Netbeans IDE 7.1.2. Yaitu sebuah software untuk pemrograman, menulis, mengompilasi, mencari kesalahan dan mengembangkan program yang berbasis Java.¹ Dengan menggunakan IDE ini, dalam perancangan aplikasi ini semakin mudah. Karena dalam IDE ini telah disediakan menu-menu penting yang

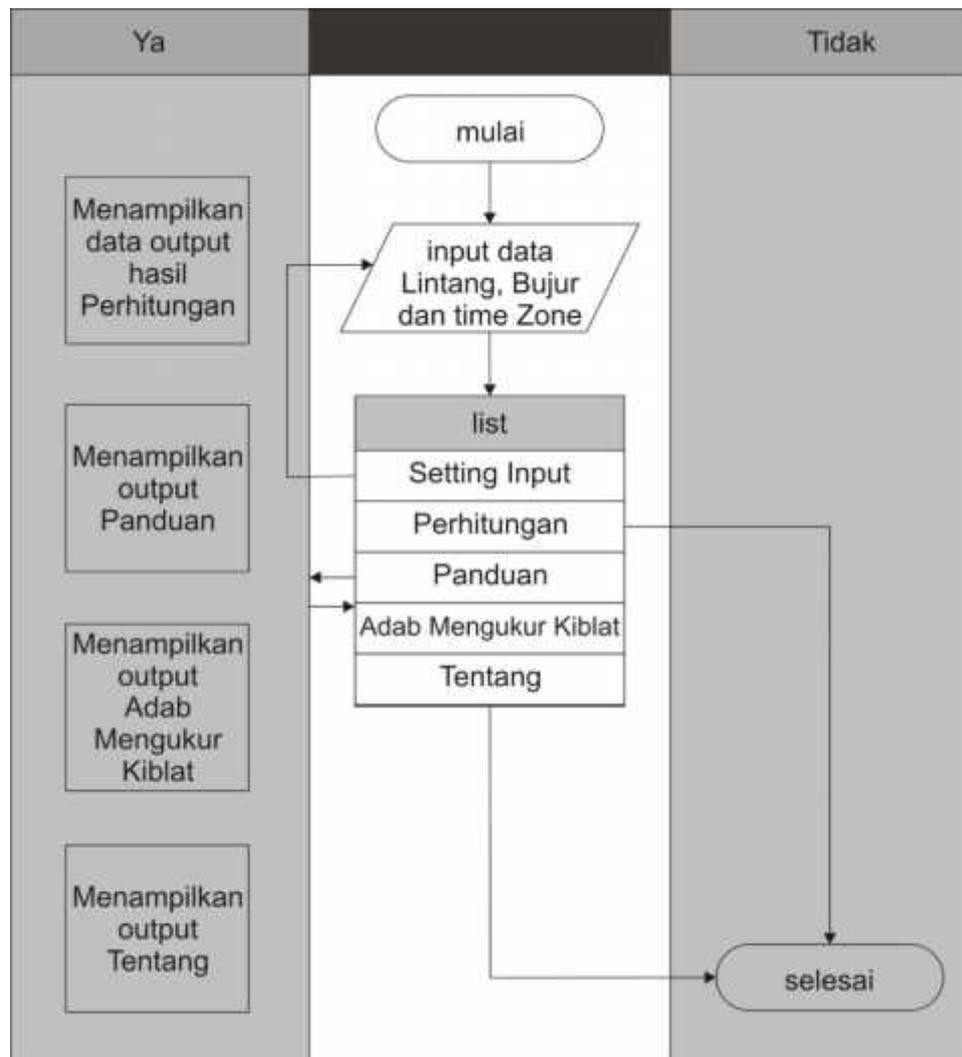
¹ https://netbeans.org/index_id.html?print=yes diakses pada tanggal 17 April 2013 pada 10:23 WIB

berfungsi dalam pemrograman. Termasuk Diagram alirnya, aplikasi yang dirancang dengan IDE ini, akan berjalan dengan baik dengan mengatur alur sistem pada menu *flow*. Diagram alir dari aplikasi yang dirancang penulis pada IDE adalah sebagai berikut :



Gambar 3.2 Diagram alir Aplikasi Perhitungan *Mizwala Qibla Finder* pada Netbeans IDE 7.1.2

Dengan demikian akan diketahui alur jalan dari aplikasi ini, sesuai dengan proses seperti gambar di atas. IDE akan mendeteksi adanya kesalahan apabila terdapat kesalahan dalam mengatur alur jalannya sistem. Sehingga apabila ada sedikit saja kesalahan, maka akan berakibat fatal dan aplikasi yang sedang dirancang tidak dapat berjalan. Diagram alir sistem Aplikasi Perhitungan *Mizwala Qibla Finder* secara umum dapat juga digambarkan dengan *flowchart*, sebagai berikut :

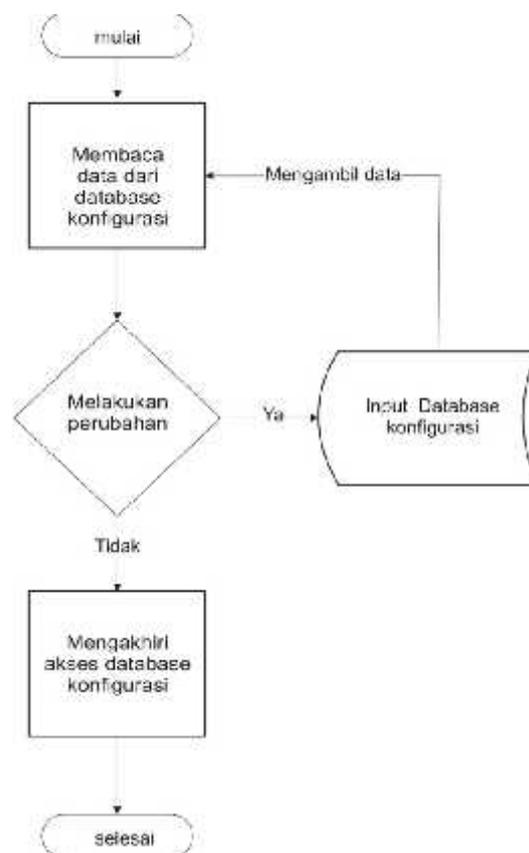


Gambar 3.3 Flowchart aplikasi perhitungan Mizwala Qibla Finder

Dari gambar 3.3 ini dapat diketahui bahwa saat aplikasi ini dijalankan pengguna akan diminta untuk mengisi data lintang, bujur lokasi dan time zone. Selanjutnya pengguna diminta untuk memilih beberapa menu, di antaranya menu *Setting input* yang berfungsi untuk merubah database konfigurasi yang diperlukan. Menu *Perhitungan* untuk melihat hasil dari perhitungan azimuth kiblat dan azimuth bayangan Matahari (mizwah) serta beberapa output penting lainnya. Menu *Panduan* untuk melihat panduan langkah demi langkah menentukan arah

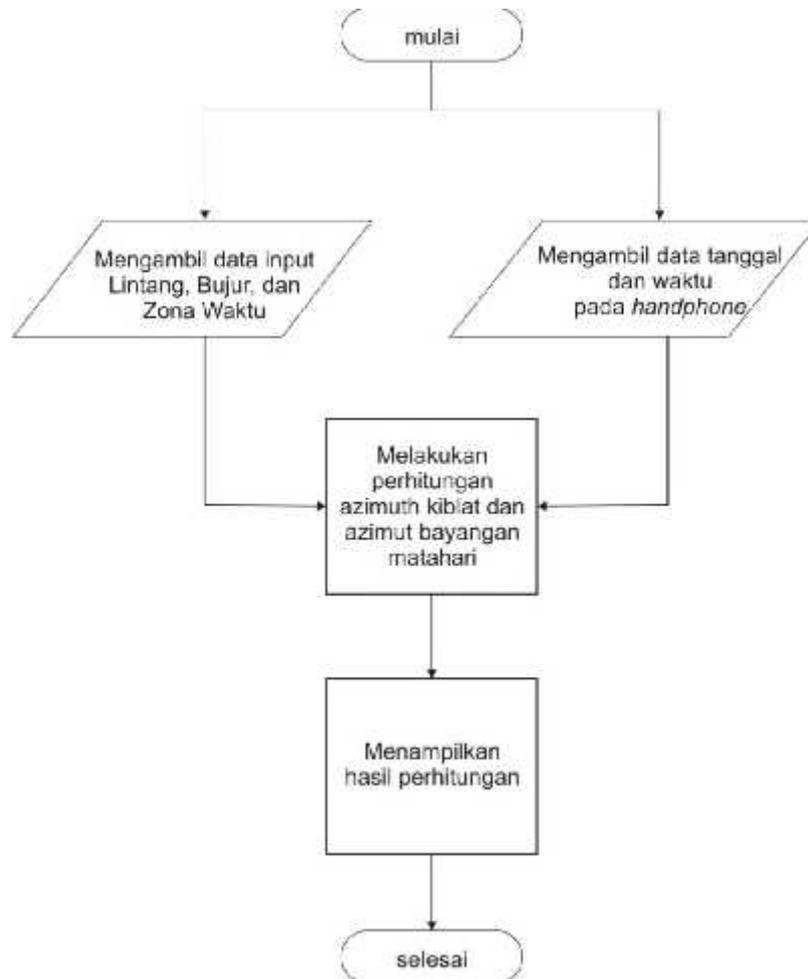
kiblat dengan menggunakan *Mizwala Qibla Finder*. Menu Adab Mengukur Kiblat yaitu tampilan tambahan yang ada dalam aplikasi ini yang berisi anjuran-anjuran sebelum menentukan arah kiblat. Dan menu Tentang, yaitu tampilan informasi tentang aplikasi dan perancang aplikasi.

Beberapa proses yang penting untuk dijelaskan tentang alur kerja dari Aplikasi Perhitungan *Mizwala Qibla Finder* ini. Yaitu, pertama alur setting input data atau database konfigurasi. Pada database konfigurasi ini, penulis telah membuat *default* data pada menggunakan data koordinat Fakultas Syariah. Namun apabila pengguna akan menggantinya bisa dilakukan seperti gambar 3.4 berikut ini :



Gambar 3.4 Diagram alur setting input data

Alur diagram yang penting kedua yaitu diagram alur proses perhitungan azimuth kiblat dan azimuth bayangan Matahari, yaitu sebagai berikut :



Gambar 3.5 Diagram alur perhitungan azimuth kiblat dan azimuth bayangan Matahari (mizwah)

Pada gambar 3.5 diterangkan bahwa untuk melakukan perhitungan azimuth kiblat dan azimuth bayangan Matahari (mizwah), terlebih dahulu aplikasi mengambil data input lintang dan bujur lokasi serta zona waktu. Selanjutnya mengambil data tanggal dan waktu yang berfungsi untuk perhitungan data astronomis posisi Matahari yang kemudian diimplementasikan pada penentuan

azimuth bayangan Matahari. Proses perhitungan azimuth kiblat di sini menggunakan teori trigonometri bola sebagaimana telah dijelaskan pada bab dua. Sementara untuk perhitungan data astronomis posisi Matahari menggunakan metode Jean Meeus dengan kriteria *high accuration* dengan ketelitian mencapai 0,01”.

C. Lingkungan Implementasi

Setelah rancangan aplikasi perhitungan *Mizwala Qibla Finder* telah dijabarkan di atas. Kemudian rancangan tersebut mulai diimplementasikan. Untuk mengimplementasikan rancangan di atas, memerlukan beberapa perangkat, yaitu perangkat lunak (*software*) yaitu kumpulan program-program dan data-data yang saling terelasi yang memerintahkan apa yang harus dilakukan komputer,² dan perangkat keras yang berfungsi sebagai alat untuk menjalankan perangkat lunak.

1. Perangkat Lunak

Adapun perangkat lunak yang digunakan dalam proses pembuatan aplikasi perhitungan *Mizwala Qibla Finder* ini adalah sebagai berikut :

- a. *Netbeans IDE 7.1.2* yang berfungsi untuk mengolah bahasa pemrograman serta perhitungan yang dibutuhkan, sehingga menjadi aplikasi yang dapat dijalankan pada *mobile phone*.
- b. *Microsoft Windows 7* sebagai sistem operasi yang digunakan untuk merancang Aplikasi Perhitungan *Mizwala Qibla Finder*.

² [www. Anneahira.com/dasar-Software/jenis-jenis-Software-secara-umum.htm](http://www.Anneahira.com/dasar-Software/jenis-jenis-Software-secara-umum.htm) diakses pada tanggal 18 April 2013

- c. *Microsoft Office 2007*, di antaranya *Microsoft Office Word* berfungsi sebagai *word editor* dan *Microsoft Office Excel* sebagai *excel viewer* juga sebagai pembanding perhitungan.
- d. *Corel Draw X4* dan *Adobe Photoshop CS 8* untuk membuat tampilan yang menarik.
- e. Java MIDP 1.0 sampai 2.0 untuk emulator aplikasi perhitungan *Mizwala Qibla Finder*.

2. Perangkat Keras

Adapun perangkat keras yang digunakan adalah :

- a. Laptop Lenovo z460 dengan chipset Intel dualcore 2.13 GHZ, 2 GB Ram DDR II.
- b. Nokia 300 dengan spesifikasi Java MIDP 2.0 sebagai *emulator* Aplikasi Perhitungan *Mizwala Qibla Finder*.

D. Implementasi Perangkat Lunak

Pada bagian ini akan dijelaskan tentang implementasi perancangan Aplikasi Perhitungan *Mizwala Qibla Finder* dengan menggunakan *Java 2 Micro Edition*. Dalam implementasi perangkat lunak ini, pembahasan akan dibagi menjadi tiga bagian yang akan menjelaskan tentang implementasi input data, implementasi proses perhitungan, dan implementasi output data.

Namun sebelumnya perlu dibahas tentang beberapa komponen penting dalam pemrograman *Java 2 Micro Edition*, yaitu sebagai berikut :

- Statement dan Ekspresi

Statement merupakan baris perintah instruksi yang akan dijalankan oleh *Java Virtual Machine* (JVM). Akhir *statement* program Java ditandai dengan titik koma (;). *Statement* dapat berupa pemanggilan fungsi, operasi aritmetika, dan sebagainya. Contoh *statement* sebagai berikut ;

```
String Ld=textField1.getString();3
```

Eksresi merupakan kumpulan *variable* atau konstanta yang dihubungkan satu sama lain dengan operator. Contoh ekspresi sebagai berikut :

```
JDE = JD+DTT;
```

- Variable

Variable adalah tempat menampung data sementara, artinya selama program berjalan nilainya dapat diubah. Dalam pemrograman Java setiap *variable* harus dideklarasikan atau diperkenalkan dahulu. Contoh :

```
int tgl,bln,thn,jm,mnt,dtk,gmt;
double bujur,lintang;
```

Setelah *variable* diperkenalkan, kemudian diinisialisasi. Yaitu member nilai awal *variable*. Contoh :

```
int tgl ;
tgl = 13 ;
```

Dalam pemrograman Java, dapat memberi inisialisasi sekaligus deklarasi. Contoh :

```
int tgl = 13 ;
```

³ Contoh *statement* ini adalah *statement* yang berfungsi untuk mengambil data string pada *textfield*.

- Kata Kunci

Kata kunci (*keyword*) adalah nama yang dipakai oleh sistem Java. Nama tersebut bisa dipakai pada fungsi, method, package, instruksi, dan lain-lain. Contoh keyword antara lain `abstract`, `double`, `int`, `super`, `public`, `default`, `Clas`, dan lain sebagainya.

- Tipe Data (*Type Data*)

Tipe Data adalah jenis data yang dikandung oleh *variable*. Tipe data sederhana dalam pemrograman Java adalah tipe inti yang tidak diturunkan pada tipe lain. Ada delapan tipe data sederhana yaitu :

a. Tipe bilangan bulat

Tipe Data	Ukuran	Rentang Nilai
byte	8 bit	-128 sampai dengan +127
short	16 bit	-32768 sampai dengan 23767
int	32 bit	-2147483648 sampai dengan 2147483647
Long	64 bit	-9223372036854775808 sampai dengan 9223372036854775807

Tabel 3.1 Daftar Tipe bilangan bulat

b. Tipe angka titik mengambang (*floating point*)

Tipe Data	Ukuran	Rentang Nilai
float	32 bit	3.4e038 sampai dengan 3.4e+08
double	64 bit	1.7e-038 sampai dengan 1.7e+308

Tabel 3.2 Daftar Tipe angka titik mengambang (floating point)

c. Tipe karakter

Tipe karakter untuk karakter dengan pengkodean *Unicode char*.

Contoh :

```
Char kode ;
Kode = 'P' ;
```

Atau dalam tipe karakter char dapat menggunakan karakter khusus yang diawali dengan *backslash* (\). Karakter-karakter tersebut adalah sebagai berikut :

Kode	Nama
\b	Backspace
\t	Tab
\n	Linefeed
\r	Carriage return
*	Double quote
\'	Single quote
\\	Backslash

d. Tipe Logika (boolean)

Tipe logika untuk menentukan nilai benar atau salah. Tipe logika atau Boolean memiliki nilai *True* dan *False*. Contoh :

```
Boolean kondisi;
Kondisi = true;
```

e. Operator Aritmatika

Operator Aritmatika merupakan operator yang digunakan untuk ekspresi matematis (menggunakan rumus-rumus matematika). Contoh aritmatika yaitu + (penambahann), - (Pengurangan), * (perkalian), / (pembagian), % (modulus).

Adapun untuk fungsi Matematika, maka setiap fungsi diawali dengan "Math." Contohnya Math.sin, Math.abs, Math.cos, dan lain sebagainya.

1. Implementasi Input Data

Pada pembahasan implementasi input data ini akan membahas tentang penerapan pengaksesan input database konfigurasi dan pengambilan data tanggal dan jam pada perangkat *mobile phone*.

a. Implementasi Form Input Database Konfigurasi dan Pangaksesannya

Bagian ini merupakan yang terpenting dalam Aplikasi Perhitungan *Mizwala Qibla Finder* ini untuk menghasilkan azimuth kiblat maupun azimuth bayangan Matahari (mizwah). Dengan demikian aplikasi ini tidak akan menghasilkan output, sebelum melakukan input data lintang, bujur dan *time zone*.

Implementasi database konfigurasi ini pertama mendesain sebuah form yang berisi beberapa *Items text field* untuk memasukkan data-data lintang dan bujur, serta *Items choice group* untuk pemilihan zona waktu.

Setelah desain form tersebut selesai dibuat, kemudian dilakukan penulisan *pseudocode* untuk pemanggilan data yang telah diinput oleh pengguna pada form database konfigurasi ini.

Adapun cara memanggil data input tersebut adalah sebagai berikut :

```

getForm3();
int  bj1,bj2,lt1,lt2,tgl,bln,thn,jm,mnt,dtk,gmt;
double T_UT,bj3,lt3,bujur,lintang;

//setting timezone
int g = choiceGroup1.getSelectedIndex();
if (g<1){
gmt=-12; }else if(g<2) { gmt=-11; }else if(g<3){ gmt=-10;
}else if(g<4){ gmt=-9; }else if(g<5){ gmt=-8; }else
if(g<6){ gmt=-7; }else if(g<7){ gmt=-6; }else if(g<8){
gmt=-5; }else if(g<9){ gmt=-4; }else if(g<10){ gmt=-3;

```

```

}else if(g<11){ gmt=-2; }else if(g<12){ gmt=-1; }else
if(g<13){ gmt=0; }else if(g<14){ gmt=1; }else if(g<15){
gmt=2; }else if(g<16){ gmt=3; }else if(g<17){ gmt=4;
}else if(g<18){ gmt=5; }else if(g<19){ gmt=6; }else
if(g<20){ gmt=7; }else if(g<21){ gmt=8; }else if(g<22){
gmt=9; }else if(g<23){ gmt=10; }else if(g<24){ gmt=11;
}else { gmt=12; }
//setting Lintang
String Ld=textField1.getString();
String Lm=textField2.getString();
String Ldt=textField3.getString();

int ltg=choiceGroup.getSelectedIndex();
lt1 = Integer.parseInt(Ld);
lt2= Integer.parseInt(Lm);
lt3= Double.parseDouble(Ldt);

if(ltg>0){
lintang=(lt1+lt2/60.0+lt3/3600.0); }else{
lintang=(lt1+lt2/60.0+lt3/3600.0)*-1;
}
//setting Bujur
String Bd=textField5.getString();
String Bm=textField6.getString();
String Bdt=textField7.getString();
int B_T=choiceGroup2.getSelectedIndex();
bj1 = Integer.parseInt(Bd);
bj2 = Integer.parseInt(Bm);
bj3= Double.parseDouble(Bdt);
if(B_T>0){
bujur= (bj1+bj2/60.0+bj3/3600.0)*-1; }else{
bujur = bj1+bj2/60.0+bj3/3600.0; }

```

b. Implementasi Pengambilan Tanggal dan Waktu

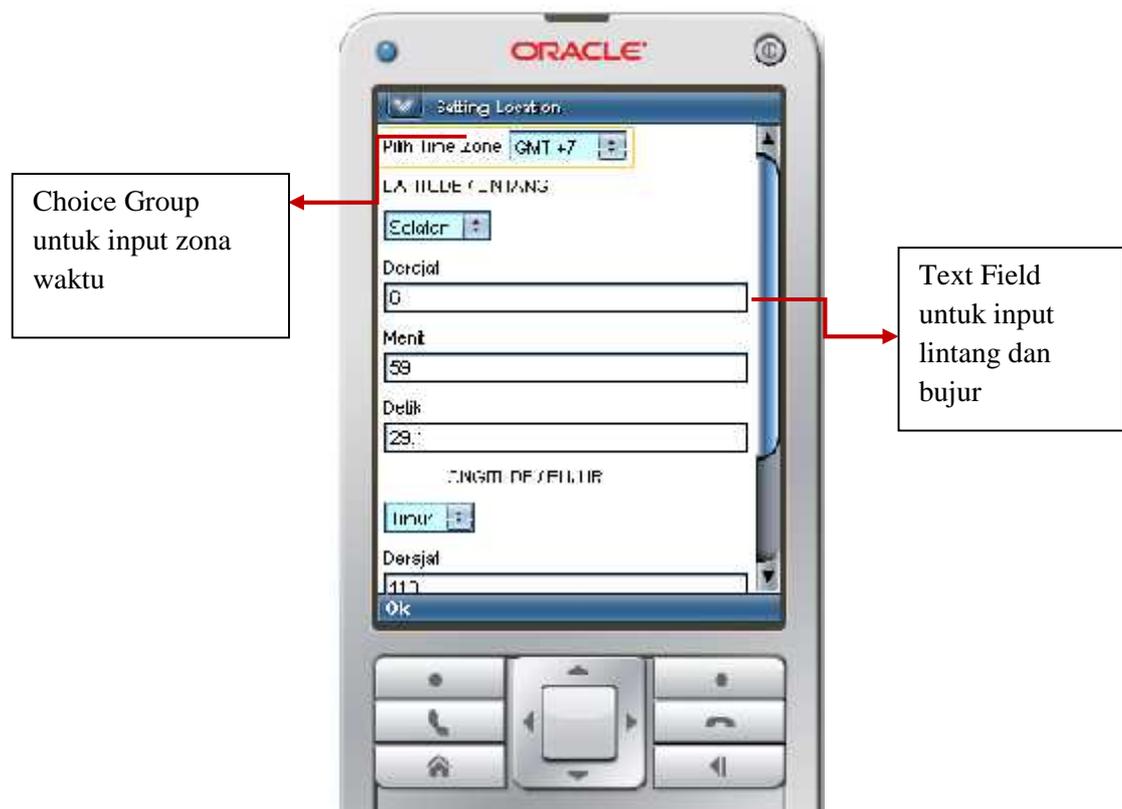
Bagian ini adalah untuk mengimplementasikan pengambilan tanggal dan waktu secara *realtime* pada perangkat *mobile*. Pada beberapa *mobile phone* terdapat setting tanggal dan waktu secara otomatis secara *realtime*, di mana setting otomatis ini adalah salah satu layanan dari operator telekomunikasi. Dengan demikian tanggal dan waktu tersebut dapat diambil dan diimplementasikan untuk melakukan perhitungan selanjutnya. Namun apabila pada *mobile phone* yang tidak terdapat menu setting tanggal dan waktu otomatis ini, sebelum menggunakan aplikasi perhitungan *Mizwala Qibla Finder* ini, dianjurkan untuk mengecek ulang

dan menyesuaikannya dengan menggunakan GPS atau penunjuk waktu lainnya.

Adapun *pseudocode* untuk mengambil data tanggal dan jam pada perangkat *mobile phone* adalah sebagai berikut :

```
import java.util.Calendar;
Calendar calendar = Calendar.getInstance();
calendar.setTime(new Date());
thn = calendar.get(Calendar.YEAR);
bln = calendar.get(Calendar.MONTH)+1;
tgl = calendar.get(Calendar.DATE);
jm = calendar.get(Calendar.HOUR_OF_DAY);
mnt = calendar.get(Calendar.MINUTE);
dtk = calendar.get(Calendar.SECOND);
```

Setelah proses implementasi di atas dilakukan, maka akan menghasilkan tampilan sebagai berikut :



Gambar 3.6 Desain antarmuka tampilan menu input database konfigurasi.

2. Implementasi Proses Perhitungan

Pada pembahasan implementasi proses perhitungan yang dilakukan dalam Aplikasi Perhitungan *Mizwala Qibla Finder* ini. Yakni proses perhitungan azimuth kiblat dan proses perhitungan azimuth Matahari (Mizwah).

a. Implementasi Perhitungan Azimuth Kiblat

Implementasi perhitungan azimuth kiblat merupakan penulisan *pseudocode* rumus arah kiblat. Rumus arah kiblat sendiri telah dijelaskan sebelumnya pada bab dua. Data yang perlu disiapkan terlebih dahulu adalah data lintang dan bujur Ka'bah, serta Data letak geografis tempat sebagaimana dalam form database konfigurasi. Setelah didapatkan data-data tersebut baru diintegrasikan dengan rumus segitiga bola :

$$\cotg B = \tan k \cdot \cos x \div \sin C - \sin x \div \tan C$$

Namun sebelumnya, karena dalam rumus tersebut ada fungsi Arcus, maka terlebih dahulu perlu membuat sebuah class untuk fungsi Arcus tersebut. Karena dalam Java 2 Micro Edition (J2ME) tidak terdapat *method* untuk fungsi tersebut. Adapun untuk membuatnya pertama harus membuat *Class* baru, kemudian diisi *pseudocode* sebagaimana terdapat pada lampiran 1. Class tersebut kemudian diberi nama Arc.java.

Setelah class tersebut dibuat, kemudian mengimplementasikan *pseudocode* rumus arah kiblat adalah sebagai berikut :

```
//Azimut Qiblat
double lm,bm,sbm,azq;
lm=21.42254722; //lintang makkah
```

```

bm=39.82626667; // bujur makkah
if(B_T >0){
    sbm= (Math.abs(bm-bujur));
}else if((B_T) <0 & Math.abs(bujur) <140.1737333){
    sbm= Math.abs(bujur-bm);
}else { sbm= 360+bujur-bm;}

Aq =(Arc.atan(1/(Math.tan(lm*Math.PI/180)*Math.cos(
    lintang*Math.PI/180)/Math.sin(sbm*Math.PI/180)-
    Math.sin(lintang*Math.PI/180) /Math.tan
    (sbm*Math.PI/180))))*180/Math.PI;

if(aq>0 & B_T<=0 &bujur<lm){azq=aq;
}else if(aq>0 & B_T<=0 &bujur>lm){azq= 360-aq;
}else if(aq>0 & B_T>0 & Math.abs(bujur) <140.1737333
){ azq=aq;
}else if(aq>0 & B_T>0 & Math.abs(bujur)>140.1737333
){azq=360-aq;
}else if(aq<0 & B_T<=0 & bujur<lm ){ azq=180+aq;
}else if(aq<0 & B_T<=0 & bujur<lm ){azq=180-aq;
}else if(aq<0 & B_T>0 & Math.abs(bujur)
<140.1737333){azq=180+aq;
}else{azq=180-aq; }

```

b. Implementasi Perhitungan Azimuth Bayangan Matahari (Mizwah)

Bayangan Matahari merupakan bayangan hitam suatu benda yang dibentuk oleh Matahari. Dengan kata lain azimuth bayangan Matahari adalah kebalikan (*opposite*) dari azimuth Matahari atau berada 180 derajat dari azimuth Matahari.

Untuk mendapatkan hasil dari azimuth Matahari, dibutuhkan beberapa data terlebih dahulu. Data-data tersebut di antaranya data deklinasi Matahari, sudut waktu Matahari, Asensio rekta, dan sebagainya. Sehingga untuk mendapatkan data-data tersebut dapat menggunakan metode algoritma Jean Meeus yang terdapat dalam buku *Astronomical Algorithms*. Langkah-langkahnya metode tersebut dalam *pseudocode* adalah sebagai berikut :

- 1) Menentukan Julian Day.

Karena untuk menentukan posisi Matahari pada tempat dan waktu tertentu maka waktu setempat harus dirubah ke *Julian day*. *Julian day* adalah perhitungan yang berlanjut berupa pecahan yang hitung mulai tahun -4712. *Julian day* dimulai tepat pada siang hari tanggal 1 Januari -4712 yaitu pukul 12 *Universal Time* (UT). *Julian Day* selalu berkorespondensi dengan *Dynamical Time* atau *Julian Ephemeris Day* (JDE).⁴ Adapun implementasi dalam *pseudocode* adalah sebagai berikut ⁵:

```
//menentukan Julian Day
int M,Y,A,B;
double JD;
if(bln<3){
    M =(int) bln+12;
}else{
    M =(int) bln;}
if(thn<3){
    Y= (int) thn-1;
}else{
    Y=thn; }

A  = (int)(Y/100);
B  = (int)2-A+((int)(A/4));
JD =(1720994.5+((int)(365.25*Y))+((int)(30.60001
*(M+1)))+B+tg1+(jm+mnt/60.0)/24-gmt/24.0);
```

Setelah melakukan perhitungan *Julian Day* selanjutnya melakukan perhitungan *Julian Ephemeris Day* (JDE) yang bersatuan TD (*Dynamical Times*) yaitu perhitungan waktu per 100 tahun. Yaitu sebagai berikut ⁶:

$$T_TD = (JDE-2451545)/36525.0;$$

⁴ Jean Meeus, *astronomical Algorithms*, Virgia : Willman-Bell, Inc, 1991, hlm 59

⁵ Rinto Angraha, *op. cit*, hlm. 9

⁶ *Ibid.* hlm. 151

2) Menentukan Bujur Matahari

Bujur Matahari yaitu busur sepanjang lingkaran ekliptika ke arah timur diukur dari titik Aries sampai Matahari.⁷ Dalam menentukan bujur Matahari terdapat beberapa koreksi untuk mendapatkan nilai *high accuracy*. Terdapat sekitar 129 suku koreksi terhadap bujur Matahari ini.⁸ Untuk mendapatkan nilai bujur Matahari yang dihitung dari pusat Bumi, menggunakan perhitungan yang tidak langsung. Berbeda dengan rumus yang *low accuracy*.

Untuk mendapatkan bujur ekliptika Matahari yang diukur melalui pusat bumi, terlebih dahulu dihitung bujur ekliptika Matahari yang diukur menurut pusat Matahari. Posisi Bumi diukur menurut Matahari, merupakan lawan dari posisi Matahari menurut Bumi. Setelah ekliptika Bumi telah diketahui, maka bujur ekliptika Matahari (Theta) = L +180 derajat. Dari hasil theta tersebut masih dilakukan koreksi lagi, yaitu dengan delta theta dan kemudian ditambahkan nilai nutasi⁹. Adapun *pseudocode* untuk menentukan bujur ekliptika Matahari sebagai berikut¹⁰ :

```
Double Aberasi, ALM ;
Aberasi = -20.4898 / (3600 * Jarak_BM) ;
ALM = (Thetax + nutasi + Aberasi) % 360 ;
```

Sehingga dari koreksi ini akan menemukan hasil bujur ekliptika Matahari yang dilihat dari pusat Bumi.

⁷ Muhyiddin Khozin, *Ilmu Falak Dalam Teori dan Praktik*, cet III, Yogyakarta: Buana Pustaka, 2004, hlm. 84

⁸ Daftar koreksi terlampir

⁹ Nutasi adalah gerak yang berubah-ubah pada sumbu rotasi bumi.

¹⁰ Jean Meeus, *op.cit*, hlm. 155

3) Menentukan lintang Matahari.

Seperti halnya bujur Matahari, untuk menentukan lintang Matahari diperlukan koreksi agar mendapatkan nilai *high accuracy*. Hal ini berbeda dengan nilai *low accuracy* yang menyebutkan bahwa lintang Matahari selalu bernilai nol. Hal ini dikarenakan nilai lintang Matahari tidak pernah melebihi 1 detik.¹¹ Adapun *pseudocode* untuk menentukan lintang ekliptika Matahari sebagai berikut¹²:

```
Lambda = Theta-1.397*T_UT-0.00031*T_UT*T_UT;
D_B     = 0.03916*(Math.cos(lambdax*Math.PI/180)-
           Math.sin(lambdax*Math.PI/180));
ALT     = Beta+D_B;
```

4) Menentukan *Right Ascension* Matahari.

Right Ascension atau assensio rekta¹³ adalah busur sepanjang lingkaran equator yang dihitung mulai titik Aries ke arah timur sampai titik perpotongan antara lingkaran equator dengan lingkaran deklinasi yang melalui benda langit.¹⁴ Adapun *pseudocode* untuk menentukan *Right Ascension* Matahari sebagai berikut¹⁵:

```
// Asensio Rekta
double alp,alpx,E,Alpha, ARA, right;
alp     = 125.04-1934.136*T_TD;
alpx    = 0.00256*Math.cos(alp*Math.PI/180);
E       =epsilon+0.00256*Math.cos(alp*Math.PI
           /180);
Alpha   =((Arc.atan2(Math.sin(ALM*Math.PI/180)*
           Math.cos(epsilon*Math.PI/180)-
           Math.tan(ALT/3600* Math.PI/180)*
           Math.sin(epsilon*Math.PI/180),
           Math.cos(ALM*Math.PI/180)))*
           180/Math.PI)%360 ;
```

¹¹ Rinto Anugraha, *Mekanika Benda Langit*, materi perkuliahan Studi Fisika Fakultas MIPA Universitas Gajah Mada, hlm. 68-69

¹² Jean Meeus, *op.cit*, hlm. 154

¹³ Disebut juga *Apparent Right Ascension* atau *Shu'udul Mustaqim*

¹⁴ Muhyiddin Khazin, *op. cit*, hlm. 54

¹⁵ Jean Meeus, *op.cit*, hlm. 89

```

if (Alpha<0){
  ARA = (Alpha+360);}
else {
  ARA=Alpha %360; }

```

5) Menentukan Deklinasi Matahari

Deklinasi Matahari adalah busur pada lingkaran waktu yang diukur mulai titik perpotongan antara lingkaran waktu dengan lingkaran ekuator ke arah utara atau selatan sampai ke titik pusat benda langit.¹⁶ Adapun *pseudocode* untuk menentukan Deklinasi Matahari sebagai berikut ¹⁷:

```

// Apparent declination
double Dec ;
Dec = Arc.asin(Math.sin(E*Math.PI/180)* Math.sin
  ALM*Math.PI/180))* 180/Math.PI;

```

6) Menentukan Equation of Time

Equation of Time atau perata waktu dalam bahasa arab disebut *Ta'dilu al-waqti* adalah selisih antara waktu hakiki dengan waktu Matahari rata-rata. Adapun *pseudocode* untuk menentukan *Equation of Time* sebagai berikut¹⁸ :

```

//equaton of Time
double tm, e, sm ;
tm = (JDE-2451545)/365250;
sm = (280.4664567 +360007.6982779*tm +
  0.03032028* tm*tm+tm*tm*tm/49931-
  tm*tm*tm*tm/15299-tm*tm*tm*tm*tm
  /1988000)%360;
e =(sm-0.0057183-ARA + nutasi * Math.cos
  (epsz*Math.PI /180))*4;

```

¹⁶ Muhyiddin Khazin, *op.cit*, hlm. 51

¹⁷ Jean Meeus, *Loc.cit*.

¹⁸ *Ibid*, hlm. 171

7) Menentukan Irtifa'

Irtifa' atau ketinggian Matahari yaitu ketinggian Matahari dihitung sepanjang lingkaran vertical dari ufuk sampai Matahari.¹⁹ Rentang nilai ketinggian Matahari adalah dalam rentang -90 hingga 90 derajat.²⁰ Adapun *pseudocode* untuk menentukan *Irtifa'* Matahari sebagai berikut ²¹:

```
//altitude
double altitude ;
altitude = (Arc.asin(Math.sin(lintang*Math.PI/180)*Math.sin (Dec*Math.PI/180) +
Math. cos (lintang*Math.PI/180)*
Math.cos(Dec* Math.PI/180)*Math.cos
(HA*Math.PI/180))) *180/Math.PI;
```

8) Menentukan Azimuth Bayangan Matahari

Setelah perhitungan diatas telah dilakukan, maka selanjutnya menentukan azimuth bayangan Matahari. Namun untuk mendapatkannya terlebih dahulu menentukan azimuth Matahari. Hal ini karena bayangan Matahari merupakan *opposite* dari Matahari. Perhitungan azimuth matahari pada algoritma Jean Meeus ini dihitung dari selatan. Dengan demikian untuk menjadikan azimuth matahari ini terhitung dari utara maka ia harus ditambahkan dengan 180 dan modulus 360. Kemudian ketika azimuth Matahari telah ditemukan, selanjutnya ditambahkan 180 derajat untuk menentukan bayangan Matahari.

¹⁹ Muhyiddin Khazin, *op.cit*, hlm. 37.

²⁰ Rinto Anugraha, *op. cit*, hlm. 66

²¹ Jean Meeus, *op.cit*, hlm. 89

Untuk menentukan azimuth Matahari dilakukan transformasi dari koordinat ekliptika ke koordinat equatorial. Adapun *pseudocode* transformasi azimuth ekliptika ke azimuth equatorial adalah sebagai berikut ²²:

```
//Azimuth Matahari
double azs, azz;
azs =(Arc.atan2(Math.sin(HA*Math.PI/180),
Math.cos
(HA*Math.PI/180)*Math.sin(lintang*Math.PI/1
80)-
Math.tan(Dec*Math.PI/180)*Math.cos(lintang*
Math.PI/180)))*180/Math.PI;
azz= (azs+180)%360;

//Azimuth bayangan Matahari
azb = (azz+180)%360;
```

3. Implementasi Penampilan Output Data

Implementasi penampilan output data dalam aplikasi ini adalah untuk menampilkan hasil proses perhitungan dan beberapa teks panduan agar dapat dibaca oleh pengguna. Dalam *Java 2 Micro Edition (J2ME)* untuk menampilkan hasil output dapat menggunakan fungsi `setText` pada sebuah `String` item untuk menampilkan berupa teks, atau menggunakan fungsi *Image Item* untuk menampilkan gambar.

a. Penampilan Data Output Proses Perhitungan

Terdapat sebuah kendala dalam implementasi penampilan hasil perhitungan ini, yaitu apabila output harus ditampilkan dalam format derajat. Dalam *Java 2 Micro Edition (J2ME)* tidak terdapat fungsi untuk menampilkan format tersebut. Oleh karena itu perlu adanya sebuah

²² *Ibid.*

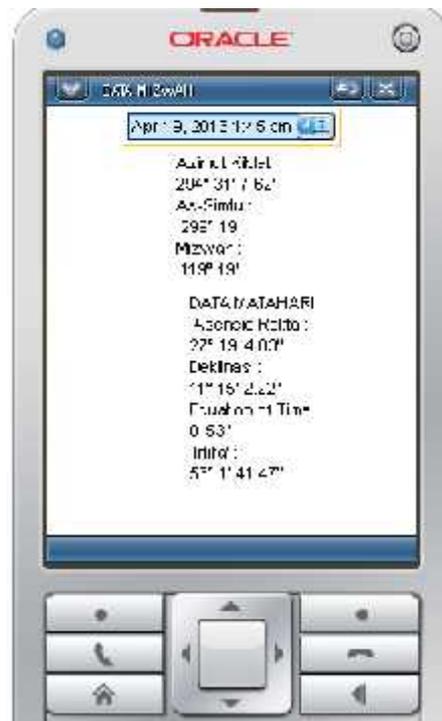
logika rumus baru sebelum hasil perhitungan ditampilkan dalam format derajat. Adapun logika rumus tersebut adalah sebagai berikut :

```
//derajat
Dazq = (int)(azq);
//menit
mazq = ((int)((Math.abs(azq)-
(int)(Math.abs(azq))*60));
//detik
dtazq = (int)((((Math.abs(azq)-(int)(Math.abs
(azq))*60)-mazq)*60);
//1 angka setelah koma
dsazq1 = (int)((((Math.abs(azq)-
(int)(Math.abs(azq))* 60)- mazq)*60)*10-
dtazq*10);
//2 angka dibelakang detik
dsazq2 = (int)((((Math.abs(azq)-(int)(Math.abs(azq)))
*60)-mazq)*60)*100-dtazq*100)-dsazq1*10;
```

Setelah dilakukan langkah di atas pada semua hasil perhitungan yang akan ditampilkan, kemudian menggunakan fungsi *setText* untuk menampilkan pada form tampilan, yaitu seperti berikut :

```
stringItem5.setText("Azimut Kiblat : \n" +dazq+"° "
+mazq+"'" +dtazq+"." +dsazq1+" "+dsazq2+"''\n"
+ "As-Simtu : \n"+dazm+"° " +mazm+"'\n"
+ "Mizwah : \n"+dazb+"° " +mazb+"'\n");
stringItem3.setText("DATA MATAHARI\n Asensio Rekta :
\n" + dara +"° " + mara +"'" +dtara+"." +dsara1+" "
+dsara2+ "' '\n" + "Deklinasi : \n" + ddec +"° " +
mdec +"'" +dtdec+"." +dsdecl1+" "+dsdec2+ "' '\n"
+ "Equation of Time \n" + meq +"'" + deq+ "' '\n "
+"Irtifa' : \n" + dalt +"° " + malt
+"'+dtalt+"." +dsalt1+" "+dsalt2+ "' '");
```

Dengan demikian akan menghasilkan tampilan sebagaimana berikut :



Gambar 3.7 Desain antarmuka tampilan form output hasil perhitungan

b. Implementasi Antarmuka Menu Bantuan

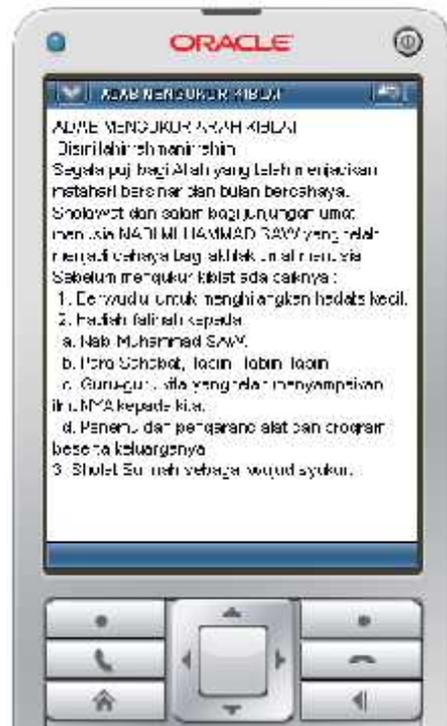
Bagian ini merupakan implementasi menu-menu yang bersifat bantuan. Menu-menu ini berisi diantaranya menu Panduan, Menu Adab mengukur Kiblat, dan menu Tentang.

Menu Panduan merupakan menu yang menampilkan panduan atau tutorial penentuan arah kiblat menggunakan *Mizwala Qibla Finder* dan aplikasi ini sebagai alat perhitungannya. Panduan ini dibuat secara *step by step*, sehingga memudahkan pengguna dalam penentuan arah kiblat. Seperti dalam proses penampilan data yang lainnya, dalam menu ini menggunakan fungsi `set text` pada sebuah `String Items`. Dengan demikian akan didapatkan tampilan sebagai berikut :



Gambar 3.8 Desain antarmuka tampilan menu Panduan

Menu Adab Mengukur Kiblat merupakan menu yang berisi anjuran sebelum menentukan arah kiblat. Anjuran ini terdapat pada software Mizwah.xls. Karena aplikasi ini merupakan transformasi dari software tersebut, maka apapun yang terdapat pada software tersebut diusahakan terdapat pada aplikasi ini. Adapun tampilan dari menu ini adalah sebagai berikut :



Gambar 3.9 Desain antarmuka tampilan menu Adab Mengukur Kiblat

Menu Tentang adalah menu yang memberikan keterangan tentang aplikasi sendiri serta informasi pembuat aplikasi. Seperti pada menu-menu sebelumnya, pada menu ini terdapat teks-teks dan juga terdapat sebuah image berupa foto pemrogram.



Gambar 3.10 Desain antarmuka tampilan menu Tentang

Setelah semua bahasa pemrograman telah ditulis dan disusun sedemikian rupa, selanjutnya Netbeans IDE 7.1.2 secara otomatis *compile* menjadi sebuah file **.Jar* yang dapat diaplikasikan langsung di *mobile phone*.

Karena aplikasi Java berbasis objek, maka ia menyediakan desain tampilan yang terbatas. Hal ini agar ukuran file aplikasi tidak terlalu besar, sehingga dapat diaplikasikan dalam perangkat *mobile phone* yang menyediakan memori dan daya yang terbatas. Dengan demikian desain antarmuka dibuat sesederhana mungkin dengan tidak meninggalkan faktor kemudahan dalam penggunaannya.