BAB IV

HASIL DAN PEMBAHASAN

4.1 Implementasi Perangkat Lunak

Implementasi merupakan tahapan dimana sistem dalam keadaan siap untuk dioperasikan, sehingga sudah dapat diketahui apakah sistem yang dibuat sesuai dengan rancangan-rancangan yang telah dibuat sebelumnya. Pada bagian ini akan menjelaskan tentang kinerja sistem, dengan menampilkan *form-form* yang telah dibuat.

4.1.1 Batasan Implementasi

Dalam proses pembuatan aplikasi pengingat waktu sholat membutuhkan perangkat lunak dan perangkat keras. Perangkat lunak seperti software yang digunakan, bahasa pemrograman dan database sedangkan perangkat keras terdiri dari spesifikasi komputer yang digunakan dalam pembuatan dan pengaplikasiannya. Perangkat lunak yang dibutuhkan dalam pembuatan aplikasi pengingat sholat meliputi: JAVA jdk-6u3-windows-i586-p sebagai bahasa pemrograman, Netbeans IDE 6.0 sebagai antarmuka dan XAMPP 1.6.3 sebagai database. Untuk perangkat keras yang dibutuhkan dalam membangun dan menjalankan aplikasi pengingat waktu sholat meliputi: Intel P4 2.24Ghz, Memori 512 MB, VGA 64 MB, Hardisk 40 GB, Monitor resolusi 1152 x 864, Mouse dan Keyboard, Ponsel penghubung.

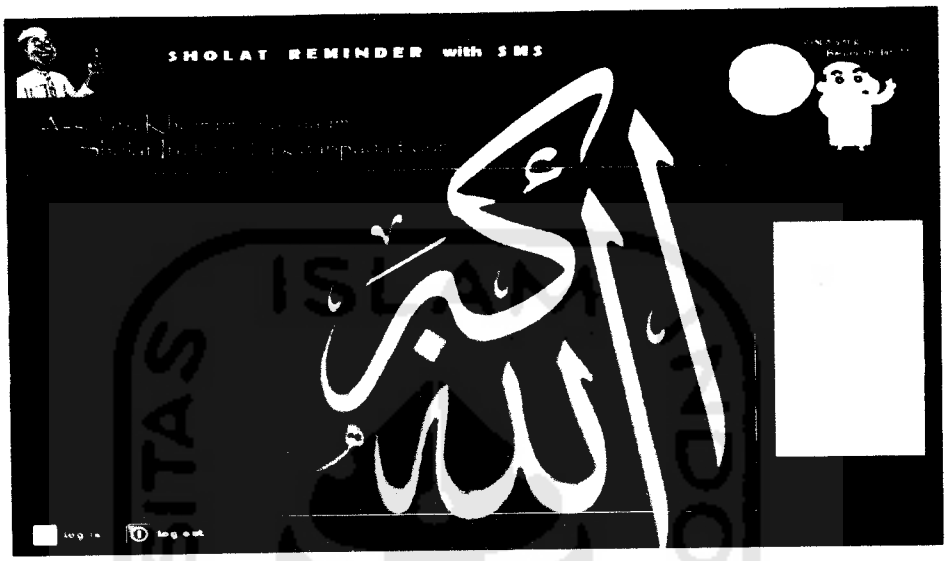
Sistem pengingat waktu sholat dalam implementasinya dibatasi pada tambah, ubah, hapus dan tampil data. Data tersebut meliputi data SMS, data member, data waktu sholat dan data modem. Data waktu sholat berisi waktu sholat dan nama kota beserta letak geografis. Data ini hanya berisi nama kota tidak ada nama kabupaten, contohnya: kota jogja, jika member berada di kota jogja maka member hanya memasukkan nama kota jogja dan itu sudah cukup tidak perlu memasukkan nama kabupaten dimana member berada misalnya sleman dan ini juga berlaku di kota lainnya.

4.1.2 Implementasi Antarmuka

Implementasi dari aplikasi Sistem pengingat waktu sholat terdiri dari beberapa form dengan masing-masing fungsinya. Tiap-tiap form memiliki fungsi sesuai dengan namanya masing-masing.

4.1.2.1 Halaman Utama

Halaman utama akan muncul pertama kali setelah sistem berhasil dijalankan. Terdapat beberapa menu pada halaman utama, antara lain: menu waktu sholat, member, sms, admin, my account dan modem. Menu-menu tersebut untuk menampilkan data yang terdapat dalam database yaitu dengan mengklik salah satu menu maka dengan sendirinya data dalam database dapat ditampilkan. Tampilan halaman utama sistem aplikasi pengingat sholat berbasis SMS dapat dilihat pada gambar 4.1



Gambar 4.1 Tampilan halaman utama

Halaman utama difungsikan menampilkan semua data sesuai menu yang dipilih. Jika admin mengklik menu waktu sholat maka disebelah kiri atau ditengah halaman utama akan menampilkan semua data sholat begitu juga dengan menu-menu lainnya. Sebagian sintaks dari halaman utama adalah sebagai berikut :

```
public class SholatRemindUI extends javax.swing.JFrame{
        /* Instance of */
        ManageData manData=new ManageData();
        MemberUI memberPanel=new MemberUI(this);
        UserUI userPanel=new UserUI(this);
        JadwalSholatUI sholatPanel=new JadwalSholatUI(this);
        SmsUI smsPanel=new SmsUI(this);
        AtCommand at=new AtCommand();
        ModemUI modem=new ModemUI(this,true,this.at);
        UserEditFormUI userEditForm=new UserEditFormUI(this,true);
        LoginFormUI loginForm=new LoginFormUI(this,true);
```

Nama *class* adalah SholatRemindUI dengan *modifier public*, yang diperluas *(extend) ke javax.swing.JFrame*. Ada 2 *type* kelas didalam pemrograman berorientasi objek yaitu kelas *public* dan *private*. Kelas *public* memiliki kelebihan karena kelas ini dapat

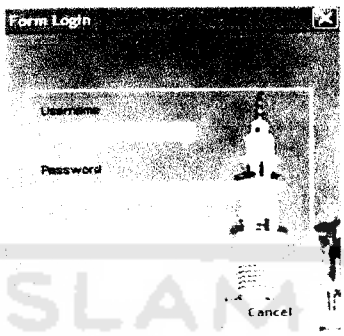
dipanggil oleh kelas lainnya menggunakan sebuah *method* yang disesuaikan dengan nama dan fungsinya. Sedangkan kelas *private* tidak seperti kelas *public* karena kelas *private* hanya dapat dipanggil oleh fungsi yang hanya berada didalam kelas tersebut. Pada bagian bawah ada sintaks:

```
MemberUI memberPanel=new MemberUI(this);
```

Maksudnya `memberUI` merupakan nama sebuah kelas panel yang berada dikelas lain yang terhubung dengan kelas `SholatRemindUI`, `memberPanel` merupakan nama lain sebuah panel yang diberikan untuk dipanggil didalam kelas `SholatRemindUI` begitu juga dengan panel yang lainnya. Dalam halaman utama ada sekitar 8 buah panel yang akan menampilkan data, fungsi panel akan disesuaikan dengan nama sehingga dapat mempermudah dalam penggunaan. Dalam tiap-tiap panel terdapat fasilitas tambah, ubah dan hapus data, dengan cara mengklik button kemudian button akan menghubungkan ke sebuah form dan dengan form tersebut admin dapat memanipulasi data.

4.1.2.2 Login

Proses login digunakan sebagai antarmuka admin untuk memasuki halaman utama sistem. Proses ini akan selalu memunculkan tampilan form login ketika sistem berhasil dijalankan. Tampilan *form* login dapat dilihat pada gambar 4.2



Gambar 4.2 Tampilan form login

Halaman form login difungsikan untuk menangani proses autentikasi login ketika admin ingin memasuki aplikasi pengingat sholat. Terdapat 2 buah *JTextField* yang dimanfaatkan utnuk menerima input username dan password admin ketika ingin mengelola sistem. Sebagian sintaks yang ada dalam form login untuk menangani proses login adalah sebagai berikut:

```
    // sintaks login ke sistem dengan nama method doLogin
    public void doLogin(){

        // proses pengecekan username dan password

if(!login.doLogin(usernameTField.getText(),passwordTField.getText()))
{

JOptionPane.showMessageDialog(this,errorMessage,"Information",2);
        }else{

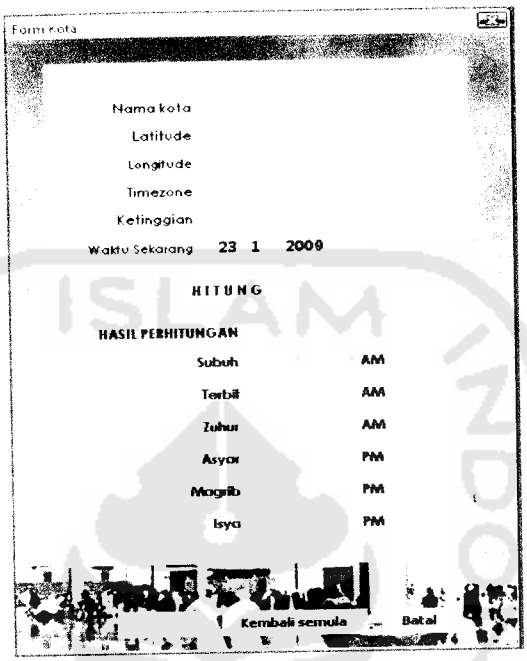
            // hasil yg didapat
            session=true;
            this.sessionData=login.getSessionData();
            if(login.isAdmin()){
                USER_PRIVILEGE=0;
            }else{
                USER_PRIVILEGE=1;
            }

            // untuk menghilangkan form
            dispose();
        }
    }
```

Nama metode adalah `doLogin`. Metode ini akan mengembalikan nilai berupa void. Sistem menerima 2 buah input username dan password yang diletakkan pada `usernameTField` untuk menerima input username dan `passwordTField` untuk menerima input password. Pada sintaks ada tertulis `USER_PRIVILEGE` untuk menentukan hak akses dengan diberi angka 0 dan 1. Dengan adanya sintaks `doLogin` data akan diautentikasi jika username dan password valid maka admin dapat masuk ke sistem dan memanipulasi data jika data tidak valid admin tidak dapat memasuki sistem untuk memanipulasi data.

4.1.2.3 Olah Data Waktu Sholat

Proses ini digunakan untuk tambah, ubah dan hapus data waktu sholat. Proses ini akan memunculkan form waktu sholat sebagai interface dalam pengolahan data. Dalam pengisian data waktu sholat terdapat proses input data berupa angka, setelah data diinputkan admin maka form akan melakukan proses penghitungan data setelah user mengklik button yang diberi nama hitung. Hasil yang didapat kemudian disimpan kedalam database yang telah disiapkan. Input yang dibutuhkan harus berupa angka, jika input yang dimasukkan berupa huruf maka form tidak akan melakukan proses penghitungan dan memerintahkan user agar mengisikan data yang benar sehingga proses penghitungan berlangsung. Sintaks yang ada pada form disertakan rumus penghitungan waktu sholat yang sudah ditetapkan atau rujukan penulis dalam menentukan waktu sholat. Tampilan form waktu sholat dapat dilihat pada gambar 4.3



Gambar 4.3 Tampilan form waktu sholat

Sebagian sintaks yang ada dalam form waktu sholat untuk proses penghitungan waktu sholat adalah sebagai berikut:

```
// nama method
void HitungWaktuSholat() {

        // proses penghitungan mencari nilai beta,D,T,R,Z,U,Vd,Vn,W
        double beta = 2 * Math.PI * J / 365; // Year angle
        //mencari nilai D
        D = (180/Math.PI) * (0.006918 - (0.399912 * Math.cos(beta)) +
              (0.070257 * Math.sin(beta)) - (0.006758 * Math.cos(2 *
              beta)) + (0.000907 * Math.sin(2 * beta)) - (0.002697 *
              Math.cos(3 * beta)) + (0.001480 * Math.sin(3 * beta)));

        //mencari nilai T
        T = 229.18 * (0.000075 + (0.001868 * Math.cos(beta)) -
              (0.032077 * Math.sin(beta)) - (0.014615 * Math.cos(2 *
              beta)) - (0.040849 * Math.sin(2 * beta)));

        //mencari nilai R
        R = 15 * TZ; // 15 merupakan nilai tetap

        //mencari nilai Z
        double Z = 12 + ((R - L) / 15) - (T / 60);
```

```
        //mencari nilai U
        double U =  (180 / (15 * Math.PI)) * Math.acos((Math.sin((-
               0.8333 - 0.0347 * (H / Math.abs(H)) *
               Math.sqrt(Math.abs(H))) * (Math.PI / 180)) - Math.sin(D
               * (Math.PI / 180)) * Math.sin(B * (Math.PI / 180))) /
               (Math.cos(D * (Math.PI / 180)) * Math.cos(B *
               (Math.PI/180))));

        //mencari nilai Vd
        double Vd = (180 / (15 * Math.PI)) * Math.acos((-Math.sin(Gd *
               (Math.PI / 180)) - Math.sin(D * (Math.PI/180)) *
               Math.sin(B * (Math.PI / 180))) / (Math.cos(D *
               (Math.PI/180)) * Math.cos(B * (Math.PI/180))));

        //mencari nilai Vh
        double Vn = (180 / (15 * Math.PI)) * Math.acos((-Math.sin(Gn *
               (Math.PI / 180)) - Math.sin(D * (Math.PI/180)) *
               Math.sin(B * (Math.PI / 180))) / (Math.cos(D *
               (Math.PI/180)) * Math.cos(B * (Math.PI/180))));

        //mencari nilai W
        double W =  (180 / (15 * Math.PI)) *
               Math.acos((Math.sin(Math.atan(1/(Sh+Math.tan(Math.abs(B
               - D)*Math.PI/180))))-
               Math.sin(D*Math.PI/180)*Math.sin(B*Math.PI/180))/(Math.c
               os(D*Math.PI/180)*Math.cos(B*Math.PI/180)));

        // hasil waktu sholat yang didapat dan menampilkan
        subuhTextField.setText(FormatWaktu(String.valueOf(Z-Vd)));
        terbitTextField.setText(FormatWaktu(String.valueOf(Z-U)));
        zuhurTextField.setText(FormatWaktu(String.valueOf(Z)));
        asyarTextField.setText(FormatWaktu(String.valueOf(Z+W)));
        magribTextField.setText(FormatWaktu(String.valueOf(Z+U)));
        isyaTextField.setText(FormatWaktu(String.valueOf(Z+Vn)));
}
```

Maksudnya adalah method ini bernama `HitungWaktuSholat` dengan nilai kembalian berupa *void*. Nilai `J, H, B, L, TZ` merupakan masukkan yang diberikan admin dalam proses penghitungan waktu sholat sedangkan nilai `beta, D, T, R, Z, U, Vd, Vn, W` nilai yang dihasilkan setelah tombol dihitung diklik. Setelah nilai-nilai tersebut didapat barulah bisa diketahui waktu sholat dan ditampilkan pada kolom *JTextField* menggunakan sintaks sebagai berikut:

```
subuhTextField.setText(FormatWaktu(String.valueOf(Z-Vd)));
terbitTextField.setText(FormatWaktu(String.valueOf(Z-U)));
zuhurTextField.setText(FormatWaktu(String.valueOf(Z)));
asyarTextField.setText(FormatWaktu(String.valueOf(Z+W)));
magribTextField.setText(FormatWaktu(String.valueOf(Z+U)));
isyaTextField.setText(FormatWaktu(String.valueOf(Z+Vn)));
```

subuhTextField merupakan nama *JTextField* yang diberikan pemrogram dalam menamai *JTextField* agar dapat dibedakan antara satu dan yang lainnya. Begitu juga dengan yang lainnya seperti: *terbitTextField. zuhurTextField, asyarTextField, magribTextField* dan *isyaTextField.* Disebelahnya terdapat sintaks *setText,* maksudnya agar hasil perhitungan dapat ditampilkan didalam *JTextField.* Pada bagian kanan *setText* sintaks untuk menampilkan nilasi hasil perhitungan kekolom *JTextField.* Agar supaya hasil yang didapatkan memiliki format waktu berupa jam, menit dan detik dibuatkan sebuat *method* yang dapat mengaturnya menggunakan sintaks:

```
// nama method FormatWaktu
String FormatWaktu(String x) {
    double hour = Math.floor(Double.parseDouble(x));
    double minute = Math.floor((Double.parseDouble(x) - hour) * 60);
    double second = Math.floor((((Double.parseDouble(x) - hour) * 60)
- minute)     * 60);

    return (String.valueOf((int)hour) + ':' +
String.valueOf((int)minute) + ':' + String.valueOf((int)second));
}
```

Method ini bernama `FormatWaktu` dengan nilai kembalian berupa string. Menghasilkan output nilai jam, menit dan detik dalam sintaks diwakili menggunakan kata `hour, minute` dan `second`. Untuk bisa menghitung waktu sholat yang benar harus diketahui tanggal, bulan dan tahun sekarang untuk mengetahui dan menampilkan secara otomatis di *form* menggunakan sintaks:

```
// menampilkan tgl, bln, thn
tglTextField.setText(String.valueOf(tanggal));
blnTextField.setText(String.valueOf(bulan));
thnTextField.setText(String.valueOf(tahun));
```

sintaks akan menampilkan tanggal, bulan dan tahun sekarang pada form waktu sholat secara otomatis. Waktu sekarang digunakan untuk menentukan nilai `getDayOfYear` atau `J` untuk mengetahui nilai `J` dibuatkan sintaks sebagai berikut:

```
// method : mendapatkan nilai J
double getDayOfYear(int day, int month, int year) {

     //tentukan jumlah hari per bulan
      int[] jmlHari = {31,28,31,30,31,30,31,31,30,31,30,31};
     int dayOfYear = 0;

      //jika tahun kabisat
       if (year % 4 == 0) {
          jmlHari[1] = 29;
       }

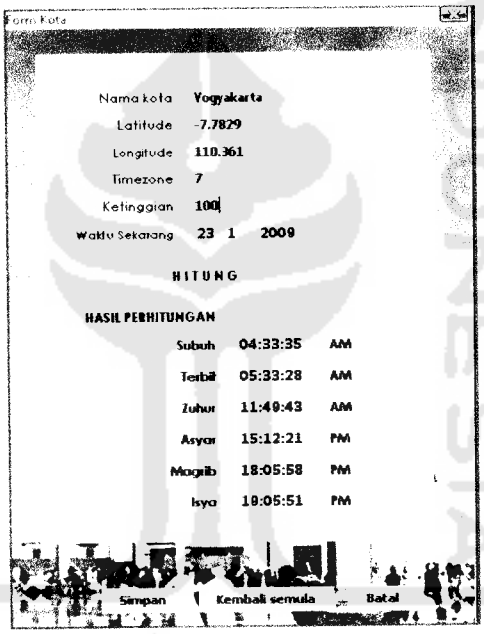
       for (int i=0; i<month-1;i++) {
           dayOfYear += jmlHari[i];
       }

       dayOfYear += day-1;

       return dayOfYear;
}
```

Nama method `getDayOfYear` dengan nilai kembalian `double` akan menghasilkan nilai `dayOfYear`. Antara satu method yang dibuat dan method yang lain saling terhubung sehinggga proses penghitungan bisa berjalan dengan lancar. Jika saja satu method mengalami error atau kesalaha maka form tidak dapat melakukan proses penghitungan sehingga dalam pembuatan sintaks diperlukan ketelitian yang amat sangat. Rumus yang digunakan merupakan rumus baku yang tidak dapat diubah karena jika diubah maka hasil perhitungan tidak valid, tetapi yang bisa diubah hanya

sintaksnya saja tergantung programer yang membuat. Logika antara satu programer dengan yang lainnya tentu berbeda, dalam mengaplikasikannya kedalam bahasa pemrograman tentunya tergantung dari programernya dengan bahasa pemrograman yang digunakan. Tampilan output form waktu sholat beserta hasil perhitungan setelah digunakan menghitung waktu sholat seperti pada gambar 4.4



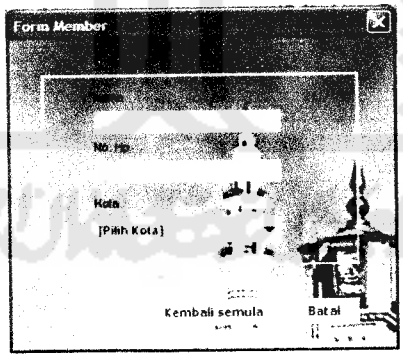
Gambar 4.4 Tampilan form hasil penghitungan waktu sholat

Didalam form tersebut menggunakan data kota Yogyakarta dengan nilai latitude, longitude, zona waktu, ketinggian yang sesuai dengan data dan waktu sekarang sesuai dengan tanggal, bulan tahun ketika sistem dijalankan. Setelah tombol *button* yang diberi nama HITUNG diklik akan melakukan proses penghitungan waktu sholat dan secara otomatis nilai hasil perhitungan akan ditampilkan pada kolom *JTextField.* Begitu juga jika ingin menghitung waktu sholat kota yang lain hanya saja

nilai masukkan yang berbeda berupa letak geografis kota. Berbeda dengan form lainnya dalam form waktu sholat terdapat masukan, proses penghitungan dan hasil setelah itu barulah data disimpan kedalam database berbeda dengan form-form yang lainnya yang hanya menerima input dan penyimpanan data.

4.1.2.4 Olah Data Member

Proses member digunakan sebagai pengolahan data member. Proses member dimanfaatkan oleh member untuk mengolah data pribadinya dan yang bisa memanfaatkan proses member adalah member sedangkan admin hanya dapat menampilkan dan melihat data tidak dapat merubah data member. Proses ini jika dijalankan akan menampilkan form member, tampilannya dapat dilihat pada gambar 4.5



Gambar 4.5 Tampilan form member

Halaman form member difungsikan untuk menangani proses tambah dan ubah data member. Jika data member belum ada sama sekali maka form akan melakukan proses penambahan data, sedangkan jika data member sudah ada maka hanya melakukan pengeditan data yang ada kemudian data tersebut disimpan ke database.

Jika data belum ada sama sekali maka akan melakukan tambah data, logika sintaks penambahan data sebagai berikut:

```
// jika data kosong
if(selectedMemberData==null){
       // pemasukkan data
        insert=true;
        manData.addMember(data);

   }else{

      // pengubahan data
       data[3]=selectedMemberData[0];
       manData.editMember(data);
   }

  // menghilangkan form
   dispose();
}
```

Untuk proses pengecekan data jika data valid maka data akan disimpan ke dalam database sebaliknya jika data tidak valid sistem akan meminta pemasukkan data ulang, untuk mengantisipasi menggunakan method:

```
// nama method
private boolean isValidInput(){

     // untuk nomor hp
     String Hp;
     if(telpTField.getText().startsWith("0")){
          Hp=telpTField.getText().replaceFirst("0","62");
     }else if(telpTField.getText().startsWith("+")){
          Hp=telpTField.getText().substring(1);
     }else{
          Hp=telpTField.getText();
     }
     boolean status=true;

     // jika masukkan huruf
     if(!vld.isNumeric(telpTField.getText())){
         String ErrorTelp="<html>No.Hp Harus diisi dgn angka</html>";

JOptionPane.showMessageDialog(this,ErrorTelp,titleMessage,2);
         telpTField.requestFocus();
```

```
status=false;
```

Untuk mengantisipasi jika nomor handphone melebihi atau kurang dari yang sebenarnya atau bahkan ada nomor yang sama dibuatkan sintaks sebagai berikut:

```
// jika kurang dari 10 digit
if(vld.isMinLen(telpTField.getText(),10)){
    errorHp=errorHp+"No.Hp Min. 10 angka<br>";
}

// jika lebih dari 15 digit
if(vld.isMaxLen(telpTField.getText(),15)){
    errorHp=errorHp+"No.Hp Maks. 15 angka<br>";
}

// jika ada yang sama
if(manData.duplikasiHp(Hp,selectedMemberData)){
    errorHp=errorHp+"No.Hp ada yang sama<br>";
}
errorHp=errorHp+"</html>";
telpTField.requestFocus();

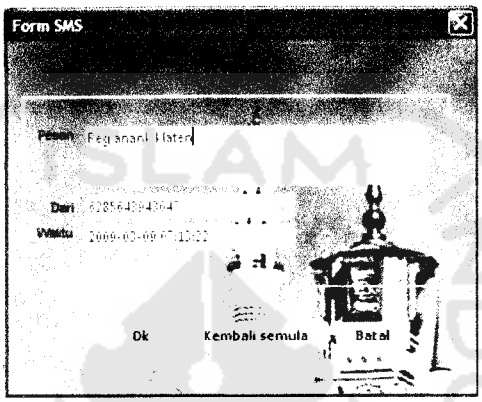
JOptionPane.showMessageDialog(this,errorHp,titleMessage,2);
status=false;
```

Method ini bernama `isValidInput` dengan nilai kembalian berupa `boolean` dengan *modifier* `private`. Berbeda dengan *modifier* `public`, *modifier* `private` hanya dapat dipanggil didalam kelas member saja tidak untuk kelas yang lain. *Modifier* `private` hanya dapat dipanggil pada kelas member tidak untuk kelas yang lain.

4.1.2.5 Olah data SMS

Proses sms digunakan untuk mengolah data SMS. Jika proses ini dijalankan akan menampilkan form SMS sebagai *interface* pengolahan data sms yang dikirim user ke sistem. Form SMS terbagi menjadi dua yaitu: form yang digunakan untuk

menerima SMS yang masuk dan form yang digunakan untuk mengirim SMS ke user. Tampilan form sms dapat dilihat pada gambar 4.6



Gambar 4.6 Tampilan form sms

Halaman form SMS difungsikan untuk menangani proses ubah dan hapus data sms yang masuk. Pada form SMS terdapat isi pesan yang dikirim user, nomor *handphone* dan waktu pengiriman. Sebagian sintaks yang ada dalam form SMS untuk mengolah data SMS adalah sebagai berikut:

```
//menampilkan form SMS
public void showForm(String[] data,int kind){
        TABLE_ID=kind;
        selectedData=data;
        setDefaultValue();
        setVisible(true);
    }
```

Maksudnya method ini bernama `showForm()` dengan *midifier public* dan nilai kembalian berupa void. Dalam proses SMS terjadi 2 aksi yaitu menerima dan mengirim pesan. Pesan yang dikirim member akan diterima kemudian pesan tersebut diolah selanjutnya sistem akan mengirim pesan balasan kepada member. Proses

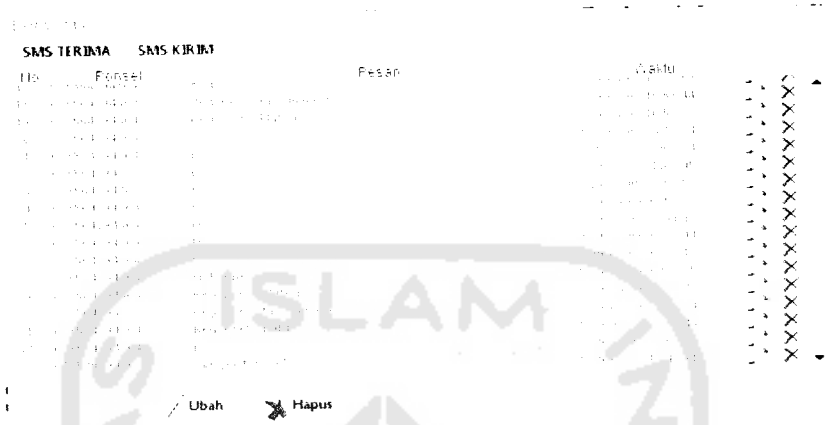
menerima dan mengirim pesan menggunakan perintah *At Command,* sebagian sintaks menerima pesan sebagai berikut:

```
/* Method : membuka dan membaca SMS inbox
 */
private void bacaSemuaSms(){
    kirimAt("AT+CMGL=0\15",1250);
}
/* Method : membaca SMS per index-nya
 */
private void bacaSms(int indexMsg){
    kirimAt("AT+CMGR="+indexMsg+"\15",1250);
}
```

Sebagian sintaks untuk mengirim pesan sebagai berikut:

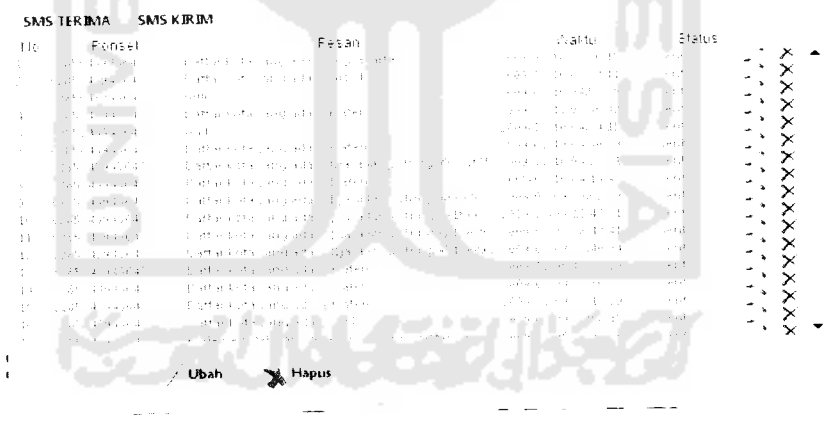
```
/* Method : kirim SMS
 */
private void kirimSms(String pesanPDUKirim){
      try{
            kirimAt("AT+CMGS=" + (pesanPDUKirim.length() / 2) + "\15",
500);
            kirimAt("00" + pesanPDUKirim, 2500); //Kirim Pesan Format
PDU
            kirimAt("\032", 100); // Ctrl + Z
            Thread.currentThread().sleep(10000);
      }catch (InterruptedException ex) {
            ex.printStackTrace();
      }
}
```

Sama halnya dengan form SMS yang terbagi menjadi dua yaitu: form terima dan form keluar begitu juga dengan tempat penyimpanan SMS juga terbagi dua yaitu: tempat penyimpanan SMS yang masuk dan tempat penyimpanan SMS yang keluar atau SMS yang terkirim. Form berbeda dengan dengan tempat penyimpanan karena form hanya menampilkan satu SMS yang akan diubah sedang tempat penyimpanan dapat menampilkan seluruh data SMS. Tampilan tempat penyimpanan SMS yang masuk seperti pada gambar 4.7



Gambar 4.7 Tampilan untuk menyimpan SMS masuk

Tampilan tempat penyimpanan SMS keluar yang dikirim ke member seperti pada gambar 3.40

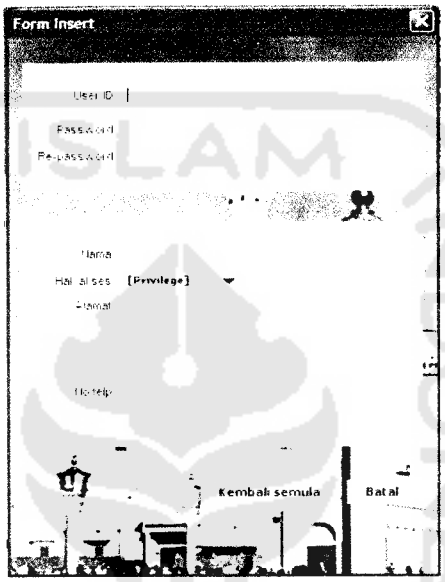


Gambar 4.8 Tampilan untuk menyimpan SMS terkirim

4.1.2.6 Olah Data Admin

Proses admin digunakan untuk mengolah data admin, seperti: tambah, ubah dan hapus. Sama halnya dengan data-data lainnya, data admin juga dapat menambah, menghapus dan mengubah datanya sendiri agar dapat disimpan kedalam sistem

pengingat waktu sholat. Jika proses ini dijalankan akan menampilkan form admin. Tampilan seperti pada gambar 4.9



Gambar 4.9 Tampilan form admin

Gambar 3.41 digunakan untuk menginputkan data admin jika admin ingin menambah data, karena bisa jadi admin tidak hanya satu tapi bisa saja ada 2 admin atau lebih. Sintaks yang digunakan untuk mengolah data admin sebagai berikut:

```
private void savedData(){
 if(isValidInput()){
     String[] data=new String[8];
     data[0]=username.getText();
     data[1]=password.getText();
     data[2]=nameTField.getText();
     data[3]=alamatTArea.getText();
     data[4]=telpTField.getText();
     data[5]=""+(privilegeCombo.getSelectedIndex()-1);
     manData.addAdmin(data);
     insert=true;
     dispose();
 }
```

Sebagian sintaks yang ada diatas digunakan untuk menerima inputan data admin, jika data yang diinputkan valid maka data akan tersimpan tetapi jika tidak valid maka data akan ditolak kemudian sistem akan meminta data baru yang valid. Untuk menyeleksi data yang diinputkan dibuatkanlah sintaks seperti dibawah ini:

```
//nama method
public boolean isValidInput(){
boolean status=true;

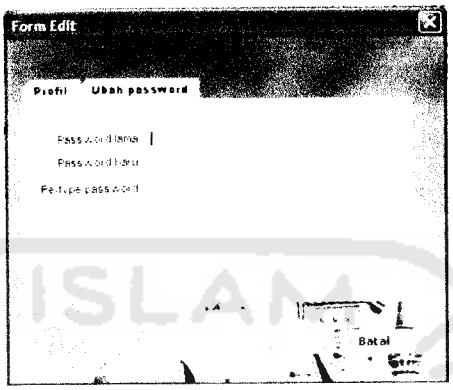
//untuk memvalidasi panjang dan pendek password
if(vld.isMinLen(password.getText(),6)||vld.isMaxLen(password.getText(
),50)){
String errorPassword="<html>";
if(vld.isMinLen(password.getText(),6)){
errorPassword=errorPassword+"Password minimal 6 !!<br>";
}
if (vld.isMaxLen(password.getText(),50)){
errorPassword=errorPassword+"Password : max 50 digit !!";
}
errorPassword=errorPassword+"</html>";
JOptionPane.showMessageDialog(this,errorPassword,titleMessage,2);
password.requestFocus();
status=false;

//memvalidasi jika password yang diinputkan tidak sama
}else if(!vld.isEqual(password.getText(),repassword.getText())){
String errorName="<html>Password dan re-password harus sama</html>";
JOptionPane.showMessageDialog(this,errorName,titleMessage,2);
password.requestFocus();
password.setText("");
repassword.setText("");
status=false;
```

Jika inputan data admin sesuai dengan yang diinginkan berdasarkan validasi sintaks, maka data admin tersimpan tetapi jika tidak akan terjadi proses pengulangan.

4.1.2.7 Ubah Password

Proses ubah password digunakan untuk mengubah data autentikasi admin. Password sangatlah penting karena dalam sistem terdapat proses login. Proses login juga dimanfaatkan untuk menjaga agar data dapat tersimpan dengan aman. Untuk tampilan ubah password dapat dilihat pada gambar 4.10



Gambar 4.10 Tampilan form ubah password

Sintaks yang digunakan untuk memvalidasi password, jika admin ingin mengubah password, adapun sintaksnya adalah sebagai berikut:

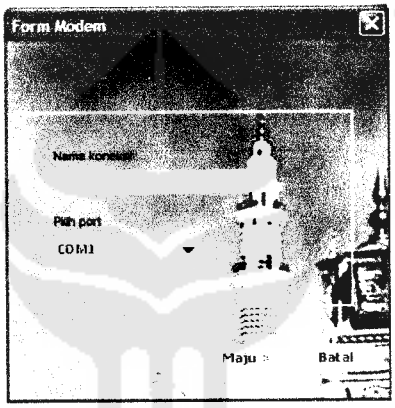
```
//untuk memvalidasi panjang dan pendek password
if(vld.isMinLen(password.getText(),6)||vld.isMaxLen(pass
word.getText(),50)){
String errorPassword="<html>";
if(vld.isMinLen(password.getText(),6)){
errorPassword=errorPassword+"Password minimal 6 !!<br>";
}
if (vld.isMaxLen(password.getText(),50)){
errorPassword=errorPassword+"Password : max 50 digit
!!";
}
errorPassword=errorPassword+"</html>";
JOptionPane.showMessageDialog(this,errorPassword,titleMe
ssage,2);
password.requestFocus();
status=false;

//memvalidai jika password yang diinputkan tidak sama
}else
if(!vld.isEqual(password.getText(),repassword.getText())
){
String errorName="<html>Password dan re-password harus
sama</html>";
```

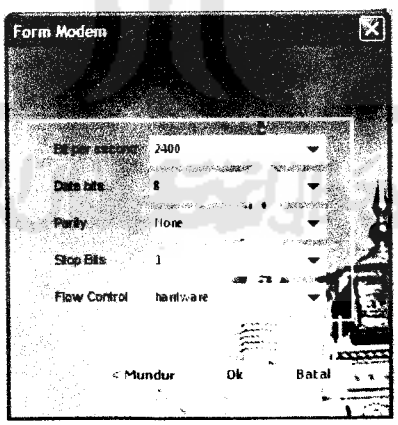
Dalam proses pengubahan password, password baru yang dimasukkan haruslah sesuai dengan format yang telah ditetapkan sehingga proses pengubahan berhasil.

4.1.2.8 Olah Data Modem

Proses modem digunakan untuk mengolah data modem, seperti: tambah, ubah, hapus dan pengaktifan modem. Jika proses ini dijalankan akan menampilkan form modem. Form modem terbagi menjadi 2 tampilan seperti pada gambar 4.11 dan 4.12



Gambar 4.11 Tampilan form modem 1



Gambar 4.12 Tampilan form modem 2

Form modem difungsikan untuk menangani proses pengaturan data modem. Pada form modem pertama terdapat pilihan nama koneksi maksudnya nama modem yang akan digunakan, bagian bawah terdapat pilih port memilih port penghubung

yang dapat menghubungkan antara ponsel dan komputer. Form modem yang kedua terdapat beberapa pilihan yang digunakan untuk mengatur koneksi ponsel karena tiap-tiap ponsel pengaturannya berbeda tergantung tipe dan merk. Sebagian sintaks yang ada dalam form modem adalah sebagai berikut:

```
//menentukan dan memilih port beserta nilainya
String[] values=new String[8];
        values[0]= connNameTField.getText();
        values[1]=(String)portNameCombo.getSelectedItem();
        values[2]=(String)bpsCombo.getSelectedItem();
        values[3]=(String)dataBitsCombo.getSelectedItem();
        values[4]=(String)parityCombo.getSelectedItem();
        values[5]=(String)stopBitsCombo.getSelectedItem();
        values[6]=(String)flowControlCombo.getSelectedItem();
        if(selectedConndata==null){// ini untuk insert

            //menetapkan nilai dan pengaktifan
            if(connNameTField.getText().equals("")){

JOptionPane.showMessageDialog(this,msgVoidConn,titleMessage,2);
                connNameTField.requestFocus();
            }else{

if(manData.isAlreadyExistConn(connNameTField.getText())){

JOptionPane.showMessageDialog(this,msgDupConn,titleMessage,2);
                }else{
mainPanel.add(addConnSecondPanel, "card2"); //panel ke 1
mainPanel.add(addConnFirstPanel, "card3"); //panel ke 2
       if(nextButton.getText().equals(" Done ")){
            manData.addConn(values);
            dispose();
            }
                backButton.setEnabled(true);
                nextButton.setText(" Done ");
              }
          }

        // edit modem
        }else{// ini untuk edit
            values[7]=selectedConndata[0];
            mainPanel.add(addConnSecondPanel, "card2");
            mainPanel.add(addConnFirstPanel, "card3");
            if(nextButton.getText().equals(" Done ")){
                manData.editConn(values);
                dispose();
```

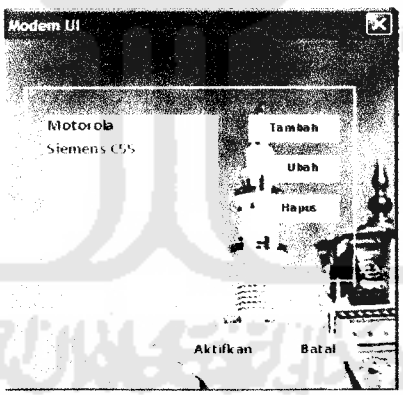
```
        }

        //menyimpan ke DB
        backButton.setEnabled(true);
        nextButton.setText(" Done ");
    }
```

Sebagian sintaks untuk mencari port adalah sebagai berikut:

```
//mencari port
comPortNames=at.getPortName();
pilihPortComboBox.setModel(new
javax.swing.DefaultComboBoxModel(comPortNames));
setDefaultValue();
```

Setelah modem berhasil terhubung maka nama koneksi akan berubah warna menjadi biru, menandakan nama koneksi sedang digunakan, tampilannya seperti pada gambar 4.13



Gambar 4.13 Tampilan modem aktif

4.2 Pengujian Program

Pada bab ini akan dijelaskan tentang pengujian terhadap aplikasi pengingat waktu sholat. Melalui tahap pengujian diharapkan dapat diketahui kekurangan-

kekurangan dari sistem, untuk kemudian dilakukan perbaikan sehingga kesalahan-kesalahan tersebut dapat diminimalisasi atau bahkan dihilangkan.

Pengujian dilakukan dengan mengisi *form-form* masukan yang sebelumnya telah ditampilkan pada BAB III, yaitu dengan mengisikan data admin, modem, waktu sholat dan member.

4.2.1 Pengujian pada admin

Pemasukkan data yang dilakukan oleh *admin* dan *member* terbagi menjadi 2 yaitu: data *valid* dan data *invalid*. Jika data masukkan *valid* maka *admin* atau *member* dapat masuk ke sistem pengingat waktu sholat serta menggunakannya. Untuk memastikan bahwa *input* data yang dimasukkan adalah *valid*, maka diperlukan adanya proses *validasi* terlebih dahulu sebelum data tersebut diproses lebih lanjut. Karena itu diperlukan adanya pesan kesalahan jika data yang dikirimkan tidak *valid*. Pada bagian ini terdapat beberapa hasil pengujian, antara lain : login, olah data member, olah data waktu sholat, olah data SMS, olah data admin, olah data modem dan ubah password.

4.2.1.1 Login

Terdapat 2 proses ketika login yaitu: berhasil dan gagal. Ketika sedang login admin diwajibkan mengisikan *username* dan *password*, kemudian jika kombinasi *username* dan *password* yang dimasukkan benar maka proses berhasil sebaliknya jika kombinasi salah maka proses login

gagal. Untuk menguji *output* yang akan dihasilkan, pada *form* login akan dilakukan pengisian data-data seperti dibawah ini.

Username : admin

Password : bismillah

Masukan data login tersebut dapat dilihat pada gambar 4.14



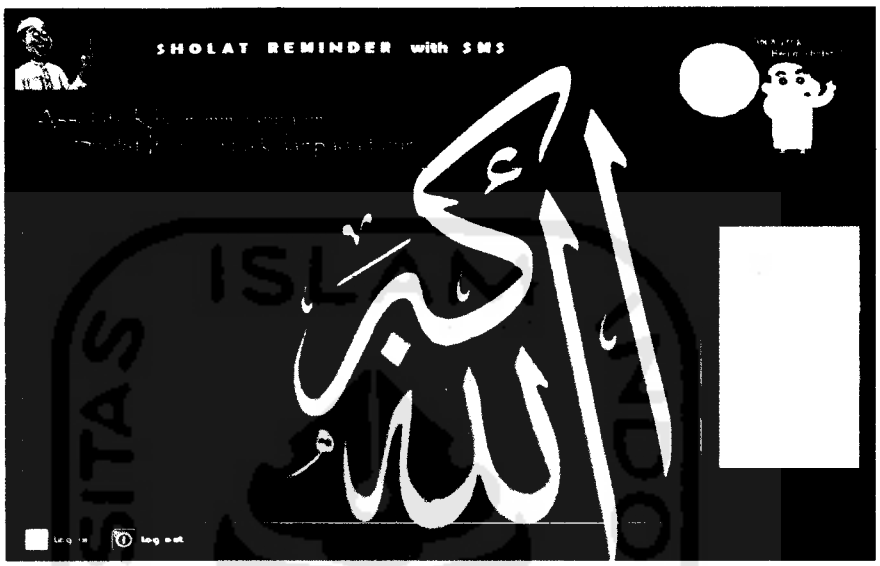
Gambar 4.14 Antarmuka masukan login

Jika *admin* salah mengisikan *username* dan *password*, akan muncul kotak dialog seperti pada gambar 4.15



Gambar 4.15 Kotak dialog username & password salah

Gambar 4.15 menandakan proses login gagal. Jika kombinasi *username* dan *password* benar proses login berhasil, maka admin dapat memasuki halaman utama aplikasi pengingat sholat untuk melakukan pengolahan data, tampilannya seperti pada gambar 4.16

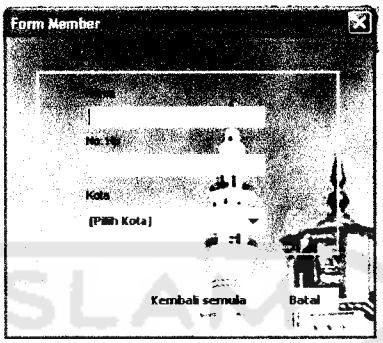


Gambar 4.16 Halaman utama

Pada bagian sebelah kanan terdapat pilihan menu w. Sholat, member, SMS, admin, my account, modem, help dan about. Jika salah satu menu diklik akan menampilkan data berdasarkan link yang dihubungkan dengan menu yang ada.

4.2.1.2 Pengujian Olah Data Member

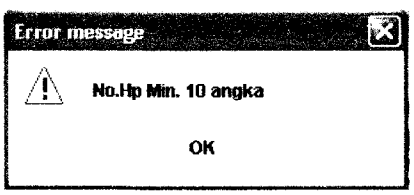
Seperti yang telah dijelaskan sebelumnya halaman utama memiliki beberapa menu yang bila salah satu menu diklik akan menampilkan data. Bila admin mengklik menu member akan menampilkan seluruh data member yang ada. kemudian jika admin mengklik menu tambah atau ubah maka akan terhubung ke *form* yang dapat digunakan mengolah data. *Form* member memerlukan masukkan data dari admin, tampilan *form* member seperti pada gambar 4.17



Gambar 4.17 From member

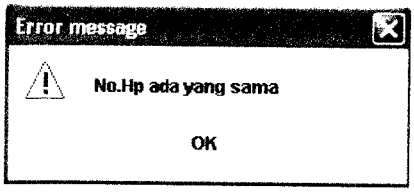
Form member memiliki 3 *field* yang menerima input data dari admin. Data yang dimasukkan haruslah benar sesuai format yang telah ditetapkan sistem jika masukkan data salah maka akan muncul kotak dialog kesalahan sesuai kesalahan data yang dimasukkan. Penanganan kesalahan olah data member berfungsi mengantisipasi dalam mengolah data member yang dilakukan admin. Contoh penanganan kesalahan seperti pada bagian dibawah ini:

Jika *field* nomor *handphone* jumlah karakter kurang dari yang sudah ditetapkan akan muncul akan muncul kotak dialog seperti pada gambar 4.18



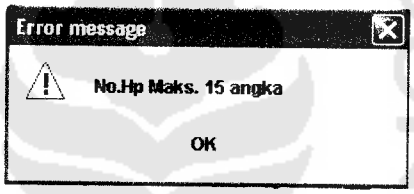
Gambar 4.18 Kotak dialog nomor handphone kurang

Jika nomor *handphone* ada yang sama akan muncul akan muncul kotak dialog seperti pada gambar 4.19



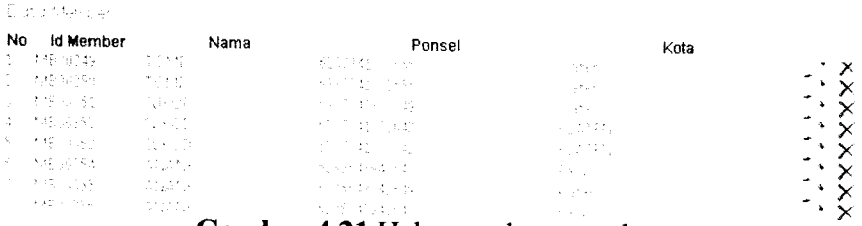
Gambar 4.19 Kotak dialog duplikasi nomor handphone

Jika *field* nomor *handphone* jumlah karakter lebih dari yang sudah ditetapkan akan muncul akan muncul kotak dialog seperti pada gambar 4.20



Gambar 4.20 Kotak dialog nomor handphone lebih

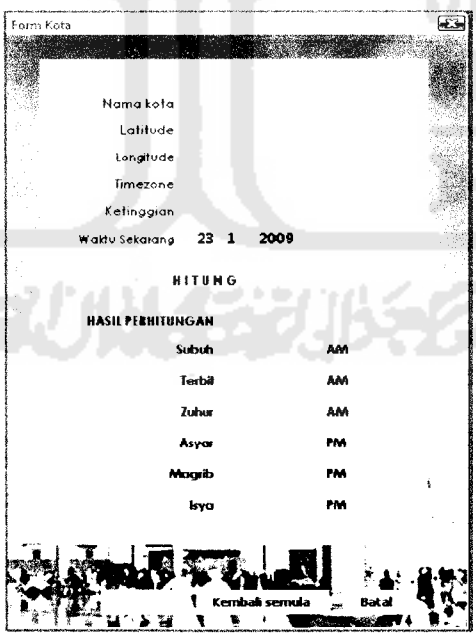
Jika admin menginputkan data yang benar pada *field* yang ada pada *form* member maka pengolahan data berhasil. Data dapat bertambah atau tetap tergantung admin mengolah datanya apakah akan menambah data baru atau hanya sekedar mengubah data yang sudah ada. Jika semua masukkan data benar maka proses pengolahan data member berhasil dan data hasil olahan akan tersimpan tampilan data yang tersimpan seperti pada gambar 4.21



Gambar 4.21 Halaman data member

4.2.1.3 Pengujian Olah Data Waktu Sholat

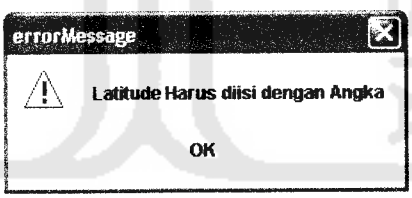
Pengujian olah data waktu sholat hampir sama dengan data lain tetapi bedanya dalam pengolahan data waktu sholat *form* waktu sholat akan melakukan proses penghitungan. Pada gambar 4.16 terdapat pilihan menu yang bila salah satu menu diklik akan menampilkan data. Bila admin mengklik menu waktu sholat sistem akan menampilkan seluruh data waktu sholat, kemudian jika admin mengklik menu tambah atau ubah maka akan terhubung ke *form* yang dapat digunakan mengolah data. *Form* waktu sholat memerlukan masukkan data dari admin untuk mengolah data tampilan *form* waktu sholat seperti pada gambar 4.22



Gambar 4.22 Form waktu sholat

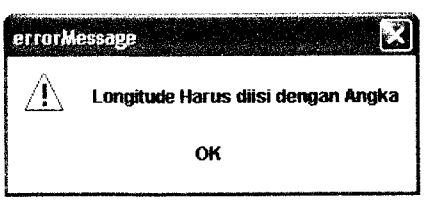
Pada *form* waktu sholat membutuhkan 4 masukkan pokok agar dapat menghitung waktu sholat, masukkannya berupa: *latitude, longitude, timezone* dan ketinggian. Pengujian olah data waktu sholat berfungsi menguji proses penghitungan waktu sholat, dimana *field* tempat pengisian data menerima data yang tidak seharusnya dimasukkan.

Contoh penanganan ini terdapat pada *form* waktu sholat. Jika pada *field* tempat pengisisian letak lintang atau *latitude,* letak bujur atau *longitude,* zona waktu atau *timezone* dan ketinggian dataran diatas permukaan laut dimana kesemuanya itu harus diisi angka bukan huruf. Jika data yang dimasukkan di *field* letak lintang atau *latitude* berupa huruf maka pada *form* waktu sholat akan muncul kotak dialog seperti pada gambar 4.23



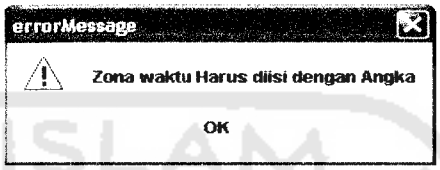
Gambar 4.23 Kotak dialog kesalahan latitude

Tampilan kotak dialog jika pada *field* letak bujur atau *longitude* diisi dengan huruf bukan dengan angka seperti pada gambar 4.24



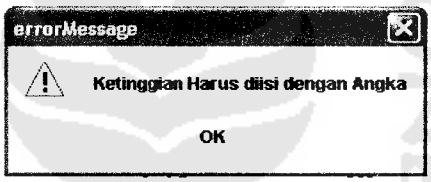
Gambar 4.24 Kotak dialog kesalahan longitude

Tampilan kotak dialog jika pada *field* zona waktu diisi dengan huruf bukan dengan angka seperti tampil pada gambar 4.25



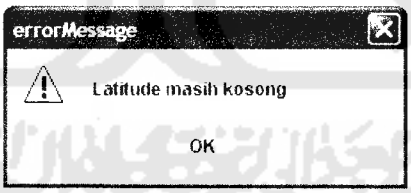
Gambar 4.25 Kotak dialog kesalahan zona waktu

Tampilan kotak dialog jika pada *field* ketinggian diisi dengan huruf bukan dengan angka seperti pada gambar 4.26



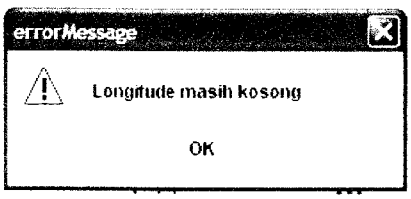
Gambar 4.26 Kotak dialog kesalahan ketinggian

Tampilan kotak dialog jika *field* latitude belum diisi dengan angka seperti pada gambar 4.27



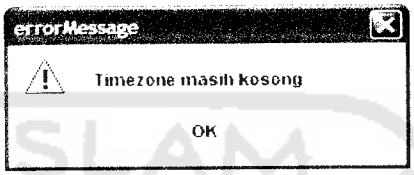
Gambar 4.27 Kotak dialog latitude kosong

Tampilan kotak dialog jika *field* longitude belum diisi dengan angka seperti pada gambar 4.28



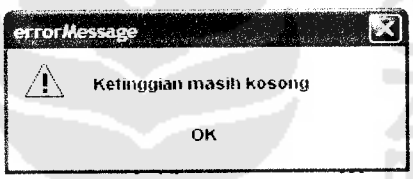
Gambar 4.28 Kotak dialog longitude kosong

Tampilan kotak dialog jika *field* timezone belum diisi dengan angka seperti pada gambar 4.29



Gambar 4.29 Kotak dialog zona waktu kosong

Tampilan kotak dialog jika *field* ketinggian belum diisi dengan angka seperti pada gambar 4.30



Gambar 4.30 Kotak dialog ketinggian kosong

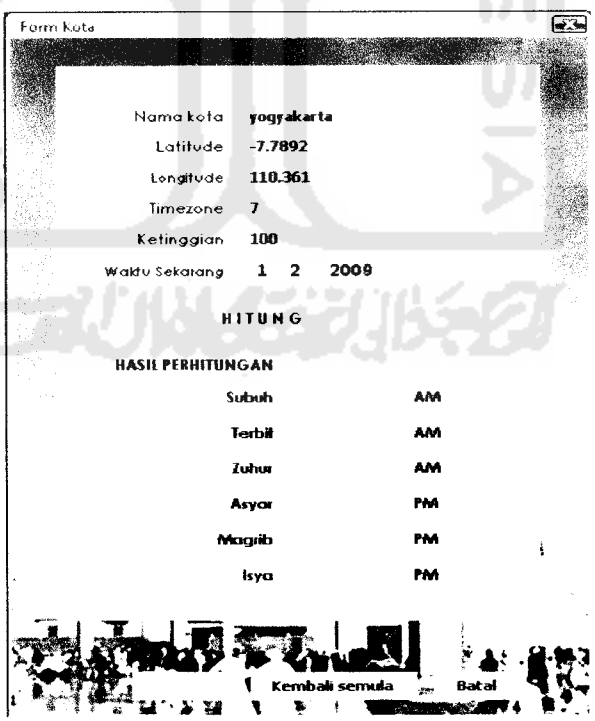
Jika admin menginputkan data yang benar pada *field* yang ada pada *form* waktu sholat maka pengolahan data waktu sholat berhasil. Data dapat bertambah atau tetap tergantung admin mengolah datanya apakah akan menambah data baru atau hanya sekedar mengubah data yang sudah ada. Jika semua masukkan data benar maka proses pengolahan data berhasil dan data hasil olahan akan tersimpan.

Untuk menguji pengolahan data waktu sholat akan dilakukan percobaan pemasukkan data pada *form* tersebut. Hasil atau *output* yang akan dihasilkan tergantung dari data yang diinputkan admin, pada *form* data waktu sholat akan dilakukan pengisian data-data seperti dibawah ini.

1. Nama kota : Yogyakarta

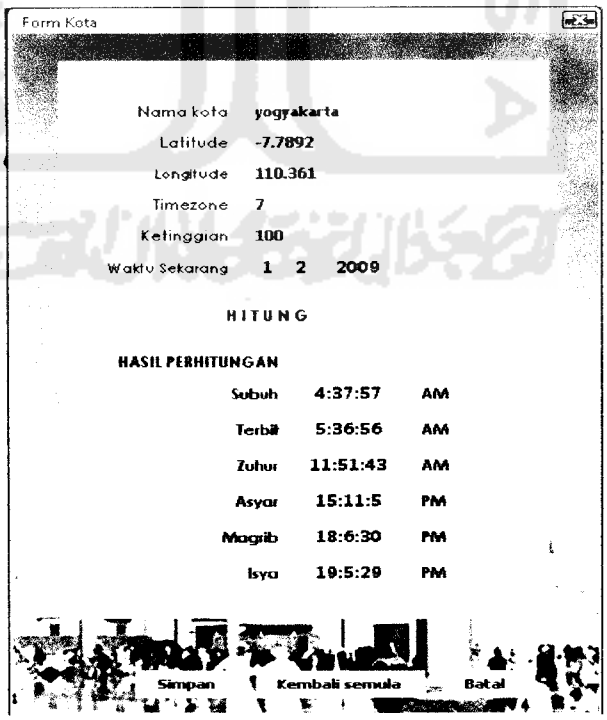
2. Latitude : -7.7892
3. Longitude : 110.361
4. Timezone : 7
5. Ketinggian : 100
6. Waktu sekarang : Waktu sekarang berupa tanggal, bulan dan tahun hari ini, akan tampil secara otomatis tanpa inputan dari admin seperti yang telah dijelaskan pada bab sebelumnya.

Data yang sudah ada kemudian diinputkan kedalam *field* yang terdapat pada *form,* kemudian masukkan data yang ada dihitung melalui proses penghitungan yang terdapat pada *form* agar dapat menghasilkan waktu sholat. *Form* waktu sholat dapat dilihat pada gambar 4.31



Gambar 4.31 Antarmuka masukan data menghitung waktu sholat

Gambar 4.32 adalah *form* untuk menghitung waktu sholat kota berdasarkan letak geografis. Pada *form* waktu sholat terlihat banyak *Button* dan *Textfield,* data yang sudah ada pada *form* belum bisa disimpan ke database karena waktu sholat belum diketahui. *Form* waktu sholat berbeda dengan *form* lainnya karena pada *form* waktu sholat terdapat proses input data, penghitungan waktu sholat dan penyimpanan data ke database. Pada *form* waktu sholat terdapat *button* yang diberi nama HITUNG untuk mengetahui waktu sholat *button* HITUNG haruslah diklik terlebih dahulu sehingga dapat diketahui hasil perhitungan waktu sholat kota berdasarkan letak geografis. Tampilan *form* setelah *button* HITUNG diklik seperti terlihat pada gambar 4.32



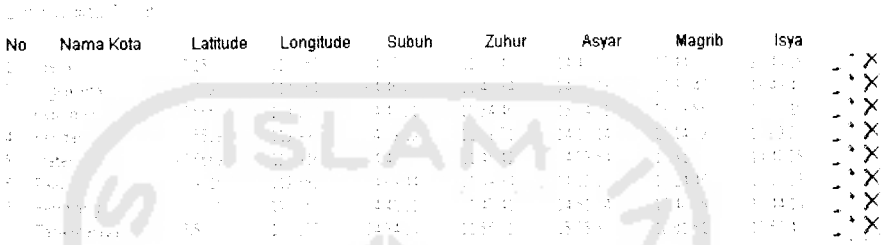
Gambar 4.32 Antarmuka hasil perhitungan waktu sholat

Gambar 4.33 terlihat hasil yang didapat setelah *button* HITUNG diklik, hasilnya berupa: waktu subuh, waktu terbit, waktu zuhur, waktu asyar, waktu magrib dan waktu isya. Jika isian pada kolom latitude, longitude, timezone dan ketinggian diisi dengan data yang tidak semestinya atau kolom masih kosong maka akan muncul kotak dialog kesalahan.

Setelah data yang diminta sistem dimasukkan kemudian admin mengklik *button* HITUNG barulah diketahui waktu sholat tetapi data ini belum dapat disimpan ke database. Pada *form* waktu sholat terdapat *button* SIMPAN, dengan mengklik *button* SIMPAN maka data secara otomatis akan tersimpan ke database sehingga data waktu sholat yang menjadi kunci ataupun nyawa dari aplikasi ini sudah terpenuhi. Tidak semua data pada *form* waktu sholat disimpan ke database, data yang tersimpan hanyalah Nama kota, Letak lintang atau *latitude,* Letak bujur atau *longitude,* waktu subuh, waktu terbit, waktu zuhur, waktu asyar, waktu magrib dan waktu isya.

Karena dirasa data tersebut sangat penting dan juga untuk menghemat *field* database. Data waktu sholat yang ada didalam database tidak semua ditampilkan tetapi yang ditampilkan hanyalah data nama kota, letak lintang atau *latitude,* letak bujur atau *longitude,* waktu subuh, waktu zuhur, waktu asyar, waktu magrib dan waktu isya sedangkan waktu terbit matahari tidak ditampilkan tapi hanya cukup disimpan di database. Hasil

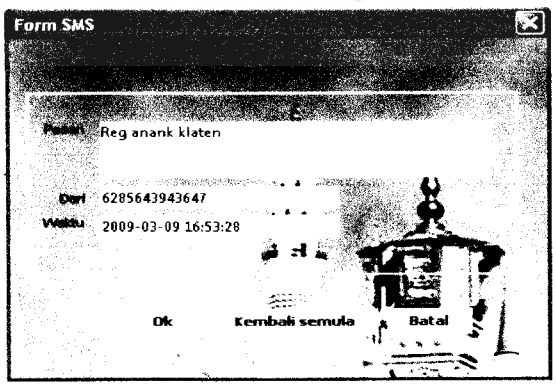
dari perhitungan waktu sholat yang tersimpan dapat dilihat pada gambar 4.33



Gambar 4.33 Halaman data waktu sholat

4.2.1.4 Pengujian Olah Data SMS

Dalam pengujian ini admin tidak dapat menambah data SMS tetapi hanya melakukan pengubahan dan penghapusan data. Sedangkan penambahan data hanya dapat dilakukan melalui proses SMS. Bila admin mengklik menu SMS akan menampilkan seluruh data SMS yang ada baik SMS yang diterima atau SMS yang dikirim. kemudian jika admin mengklik menu ubah maka akan terhubung ke *form* yang dapat digunakan mengolah data. Tampilan *form* SMS seperti pada gambar 4.34

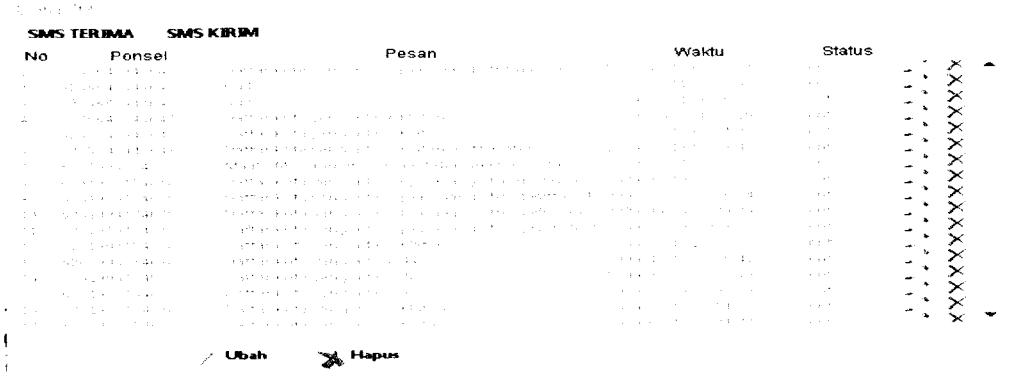


Gambar 4.34 Form SMS

Penanganan kesalahan olah data SMS digunakan untuk mengubah dan menghapus data SMS yang dikirimkan user. Data SMS terbagi menjadi 2 yaitu: data SMS terima dan data SMS kirim. Data SMS terima meupakan data SMS yang dikirimkan sistem ke user setelah user tersebut terlebih dahulu mengirimkan pesan SMS ke sistem kemudian sistem merespon SMS yang masuk dan memberikan SMS balasan. Semua SMS baik yang diterima maupun yang dikirim disimpan sistem, tampilan halaman penyimpanan SMS yang diterima seperti pada gambar 4.35 dan tampilan penyimpanan SMS yang dikirim seperti pada gambar 4.36



Gambar 4.35 Halaman data SMS terima



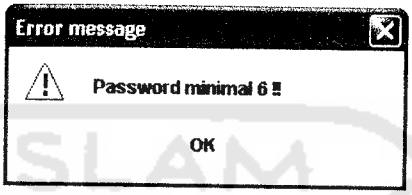
Gambar 4.36 Halaman data SMS kirim

Untuk menjaga keteraturan data SMS maka tempat penyimpanan data antara SMS terima dan SMS kirim harus dipisah. Data yang ada berupa: isi pesan, nomor *handphone* pengirim dan waktu pengiriman SMS. Sedangkan status yang ada pada tabel SMS kirim menandakan status SMS yang bersangkutan, jika SMS sudah dikirim maka status akan berubah menjadi *sent* atau sudah terkirim, sebaliknya jika masih belum terkirim status masih *pending* atau menunggu.

4.2.1.5 Pengujian Olah Data Admin

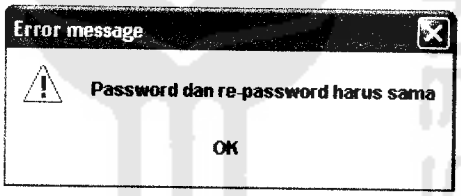
Halaman utama memiliki beberapa menu yang bila salah satu menu diklik akan menampilkan data. Bila admin mengklik menu admin akan menampilkan seluruh data admin yang ada. Kemudian jika admin mengklik menu tambah atau ubah maka akan terhubung ke *form* yang dapat digunakan mengolah data. Data yang sudah ada dapat diubah sesuai kehendak admin menggunakan *form* yang terdapat disistem pengingat sholat berbasis SMS. Admin memiliki 2 buah *form* yang dapat digunakan untuk menambah dan mengubah data. Tampilannya *form*nya berbeda tetapi memiliki fungsi sama. Tampilan *form* tambah data admin seperti pada gambar 4.37

kurang dari jumlah karakter yang sudah ditetapkan seperti pada gambar 4.39



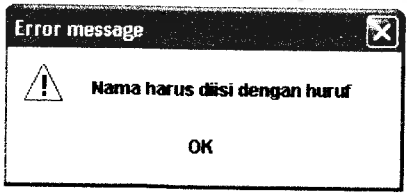
Gambar 4.39 Kotak dialog kesalahan password kurang

Karena dalam pengisian password memiliki 2 masukkan password yang harus sama maka jika data password tidak sama akan muncul kotak dialog seperti pada gambar 4.40



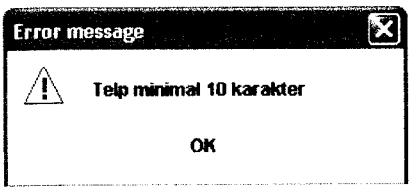
Gambar 4.40 Kotak dialog kesalahan password tidak sama

Contoh penanganan jika data yang diinputkan pada *field* nama tidak semestinya, maka akan muncul kotak dialog seperti pada gambar 4.41



Gambar 4.41 Kotak dialog kesalahan nama admin

Penanganan jika data yang diinputkan pada *field* nomor telepon tidak semestinya akan muncul kotak dialog seperti pada gambar 4.42



Gambar 4.42 Kotak dialog kesalahan nomor telepon

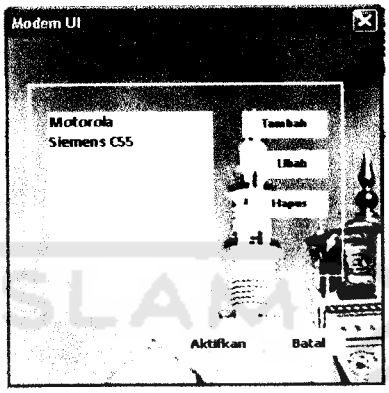
Jika semua masukkan data benar maka proses pengolahan data admin berhasil dan data hasil olahan akan tersimpan tampilan data yang tersimpan seperti pada gambar 4.43



Gambar 4.43 Halaman data admin

4.2.1.6 Pengujian Olah Data Modem

Bila admin mengklik menu modem akan menampilkan seluruh data modem yang ada. Halaman modem digunakan menyimpan data modem yang digunakan menghubungkan ponsel dan komputer. Ponsel yang digunakan dapat berbeda selama ponsel tersebut terdapat fasilitas yang dibutuhkan sistem kemudian dilakukan pengaturan seperti: nama modem, port penghubung, nilai bps, data bits, parity, stop bits dan flow kontrol. Tampilan halaman penyimpanan daftar data modem seperti pada gambar 4.44



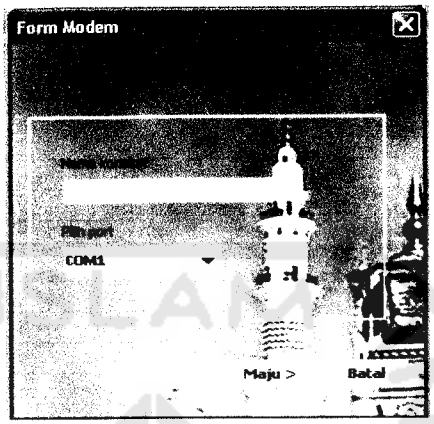
Gambar 4.44 Halaman modem

Kemudian jika admin mengklik menu tambah atau ubah maka akan terhubung ke *form* yang dapat digunakan mengolah data. Untuk menguji *output* yang akan dihasilkan, pada *form* data modem akan dilakukan pengisian data-data seperti dibawah ini.

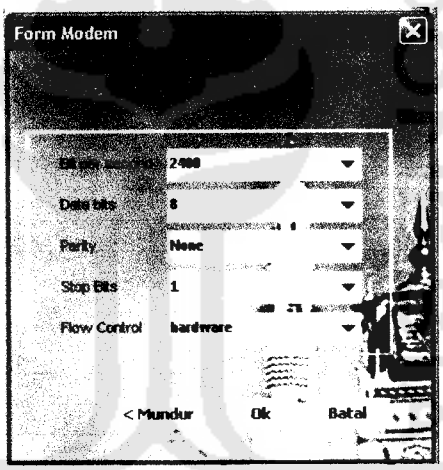
1. Nama koneksi : k510i
2. Port yang digunakan : COM3
3. Bit per second : 19200
4. Data bits : 8
5. Parity : none
6. Stop bits : 1
7. Flow control : hardware

Penjelasan dari data diatas, nama koneksi diisi dengan k510i karena menggunakan ponsel merk sony ericsson type k510i, port COM3 karena ponsel terhubung ke komputer melalui port COM3 dan dibagian bawah merupakan nilai bps, data bits, parity, stop bits dan flow kontrol

Tampilan *form* modem seperti pada gambar 4.45 dan 4.46

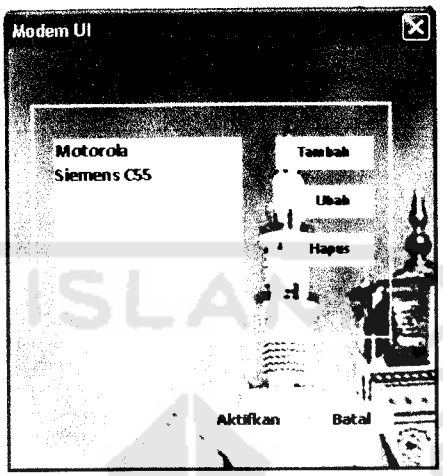


Gambar 4.45 Form modem 1



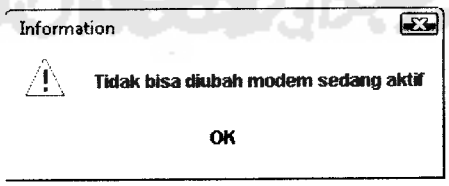
Gambar 4.46 Form modem 2

Kemudian admin mengklik button OK maka dengan sendirinya data modem akan tersimpan, setelah data modem tersimpan data modem dapat diubah dan diaktifkan. Nama modem yang sedang aktif akan berubah warna menjadi biru itu menandakan nama modem tersebut sedang digunakan seperti tampak pada gambar 4.47



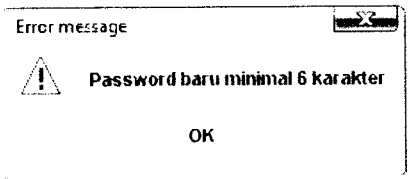
Gambar 4.47 Antarmuka data modem

Pengujian olah data modem juga terdapat penanganan kesalahan yang berfungsi untuk menangkap kesalahan yang terjadi pada proses tambah, ubah dan pengaktifan data modem dimana modem tersebut masih dalam keadaan aktif. Untuk penanganan kesalahan ini terdapat dua jenis penanganan kesalahan. Pertama, jika pada modem aktif dipilih untuk melakukan proses ubah data atau hapus data maka akan muncul kotak dialog seperti pada gambar 4.48



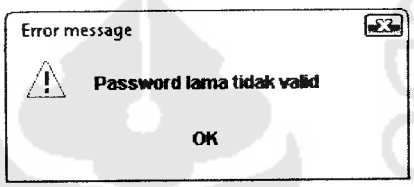
Gambar 4.48 Kotak dialog kesalahan ubah data modem keadaan aktif

Kedua, pada proses aktifasi modem dimana admin ingin melakukan aktifasi terhadap modem dan modem tersebut dalam keadaaan aktif, akan muncul kotak dialog seperti pada gambar 4.49



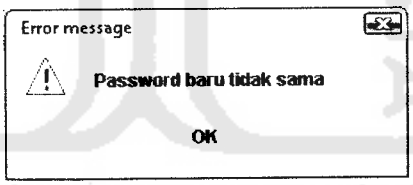
Gambar 4.51 Kotak dialog kesalahan password kurang

Jika password lama yang diminta tidak valid, maka akan muncul kotak dialog seperti pada gambar 4.52



Gambar 4.52 Kotak dialog kesalahan password tidak valid

Jika kombinasi antara password baru tidak sama, maka akan muncul tampilan seperti pada gambar 4.53



Gambar 4.53 Kotak dialog kesalahan password tidak sama

Data password tidak ditampilkan pada sistem tetapi cukup disimpan didalam database. Untuk mengecek data berhasil diubah dengan cara melihat pada bagian *field* password pada tabel admin_auth akan berubah sesuai masukkan password yang baru.

4.2.2 Pengujian Pada Member

Pengujian pada bagian member berupa proses validasi SMS yang dikirim oleh member ke sistem atau *server*. SMS tersebut berisikan perintah-perintah untuk melakukan interaksi dengan sistem utama. Pada pengujian kali ini *handphone* server menggunakan Motorola *type* sedangkan member menggunakan Motorola *type* C168.

4.4.2.1 Perintah Tidak Dikenal

Jika member mengirimkan SMS berisi perintah yang tidak dikenal, maka member akan memperoleh SMS balik dari sistem. Urutan prosesnya sebagai berikut:

User kirim SMS : "Halo".

Sistem membalas SMS : "Maaf SMS yang anda kirim tidak dikenal. Ketik Help untuk mengetahui semua kode perintah".

Tampilan SMS pada ponsel member sebagai berukut:



Gambar 4.54 Tampilan isi pesan perintah tidak dikenal

4.4.2.2 Masukkan Help

Beberapa urutan proses pada pengujian perintah help, antara lain:

User kirim SMS : "Help"

Sistem membalas SMS : "Kode perintah yang ada: REG? untuk daftar, LK? untuk lihat daftar kota, UK? untuk ubah kota anda berada dan UP? untuk ubah password".

Tampilan SMS pada ponsel member sebagai berikut:

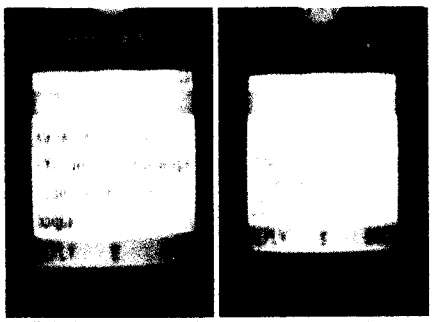


Gambar 4.55 Tampilan isi pesan perintah help

User kirim SMS : "UK?"

Sistem membalas SMS :"Untuk ubah lokasi kota anda berada ketik UK <Password> <KotaAndaSekarang>. Contoh UK 1234 Jogja".

Tampilan SMS pada ponsel member sebagai berukut:

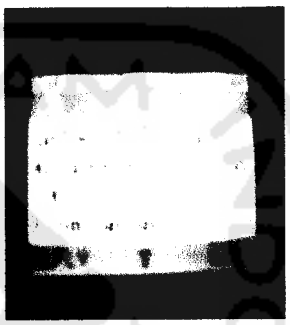


Gambar 4.56 Tampilan isi pesan panduan ubah kota

User kirim SMS : "LK?".

Sistem membalas SMS : "Untuk lihat kota, ketik LK. Contoh LK".

Tampilan SMS pada ponsel member sebagai berikut:



Gambar 4.57 Tampilan isi pesan panduan lihat kota

4.2.2.3 Pendaftaran

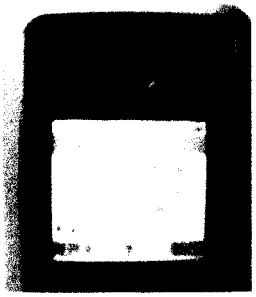
Untuk menguji hasil dari proses pendaftaran, dilakukan pengiriman SMS oleh *user* ke sistem. Terdapat beberapa urutan proses pada pengujian ini, antara lain sebagai berikut:

Jika format pendaftaran salah urutan proses SMS seperti berikut:

User kirim SMS : "REG".

Sistem membalas SMS : "Format salah. Ketik REG Nama Kota. Contoh REG Anang klaten".

Tampilan SMS pada ponsel member sebagai berikut:



Gambar 4.58 Tampilan isi pesan panduan pendaftaran

Jika pendaftaran berhasil urutan proses SMS seperti berikut:

User kirim SMS : "REG Anang klaten".

Sistem membalas SMS : "Pendaftaran berhasil password 1234, demi keamanan data silahkan ubah password, ketik UP? dan ikuti petunjuknya".

Tampilan isi SMS pada ponsel member sebagai berikut:



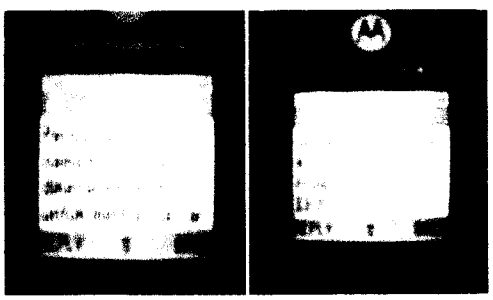
Gambar 4.59 Tampilan isi pesan pendaftaran berhasil

Jika pendaftaran gagal urutan proses SMS seperti berikut:

User kirim SMS : "REG Anang solo"

Sistem membalas SMS : " Pendaftaran gagal, nama kota tidak dikenal ketik LK untuk melihat data kota".

Tampilan SMS pada ponsel member sebagai berikut:



Gambar 4.60 Tampilan isi pesan nama kota tidak dikenal

Jika user mengirim SMS yang isinya LK, sistem membalas seperti berikut:

User kirim SMS	: "LK".
Sistem membalas SMS	: " Daftar kota yang ada = <daftar seluruh nama kota yang ada pada database>".

Tampilan SMS pada ponsel member sebagai berikut:



Gambar 4.61 Tampilan isi pesan daftar kota

Selanjutnya member melakukan pendaftaran ulang menggunakan nama kota yang ada dalam database sistem.

4.2.2.4 Login

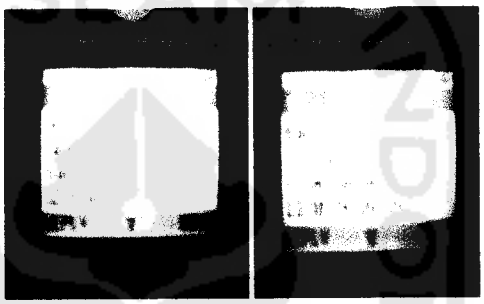
Pengujian login dilakukan pada saat user sudah terdaftar menjadi member. Pengujian proses login baru dapat dilakukan jika member ingin merubah password dan kota. Karena jika member ingin mengubah password dan kota ia harus menyertakan password terlebih dahulu. Untuk bisa mengetahui format SMS ubah password dam kota, urutan SMS sebagai berikut:

Untuk mengetahui format ubah password.

User kirim SMS : "UP?".

Sistem membalas SMS :"Untuk ubah password ketik UP <Password Lama> <Password Baru>. Contoh CP 1234 abcd".

Tampilan SMS pada ponsel member sebagai berikut:



Gambar 4.62 Tampilan isi pesan panduan ubah password

Untuk mengetahui format ubah kota.

User kirim SMS : "UK?".

Sistem membalas SMS : "Untuk ubah lokasi kota anda berada ketik UK <Password> <KotaAndaSekarang>. Contoh UK 1234 Jogja".

Tampilan SMS pada ponsel member seperti tampak pada gambar 4.56

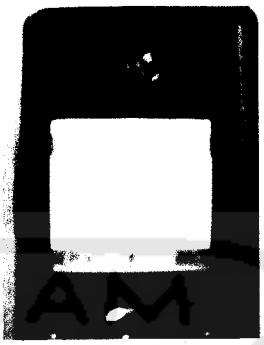
Ketika member sudah mengetahui format ubah lokasi kota dan ubah password barulah proses pengujian login pada member dapat dilakukan. Urutan prosesnya seperti berikut:

Urutan SMS ubah password.

User kirim SMS : "UP 1234 abcd".

Sistem membalas SMS : "Password berhasil diubah".

Tampilan SMS pada ponsel member sebagai berikut:



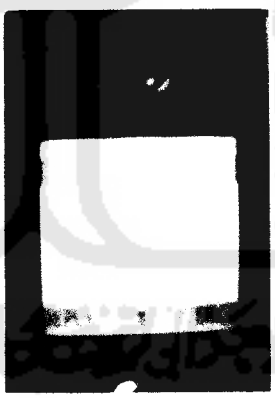
Gambar 4.63 Tampilan isi pesan ubah password berhasil

Urutan SMS ubah kota.

User kirim SMS : "UK 1234 klaten".

Sistem membalas SMS : "Ubah kota berhasil".

Tampilan SMS pada ponsel member sebagai berikut:



Gambar 4.64 Tampilan isi pesan ubah kota berhasil

Pada urutan proses ubah password dan ubah kota angka 1234 merupakan password *default* yang secara otomatis dimiliki user begitu ia sudah terdaftar, penjelasannya bisa dilihat pada proses pendaftaran. Sedangkan jika member mengirikan password salah maka akan menerima

SMS balasan yang isinya password salah, untuk lebih jelas urutan prosesnya sebagai berikut:

Urutan proses SMS ubah password jika member salah kirim password.

User kirim SMS : "UP 12345 abcd"

Sistem membalas SMS : "Password salah (case sensitive) atau anda belum terdaftar".

Tampilan SMS pada ponsel member sebagai berikut:



Gambar 4.65 Tampilan isi pesan password salah

Dengan sistem dapat mengetahui password salah dan benar maka proses login berhasil.

4.2.2.5 Olah Data Member

Diasumsikan bahwa user telah terdaftar, pengujian dilakukan dengan mengirimkan SMS ke sistem. Member dapat mengubah data kota dan password. Terdapat beberapa urutan proses pada pengujian ini, antara lain sebagai berikut:

- Untuk ubah data kota

  Jika nama kota benar, urutan SMS seperti berikut:

User kirim SMS : "UK abcd klaten".

Sistem kirim SMS : "Ubah kota berhasil".

Jika nama kota tidak sesuai, urutan SMS seperti berikut:

User kirim SMS : "UK abcd jogja"

Sistem kirim SMS :"Ubah kota gagal, Nama kota tidak dikenal ketik LK untuk melihat daftar kota".

Jika user mengirim SMS isinya LK, urutan SMS seperti berikut:

User kirim SMS : "LK".

Sistem membalas SMS : "Daftar kota yang ada = <daftar seluruh nama kota yang ada pada database>".

- Untuk ubah password

Jika proses benar, urutan SMS seperti berikut:

User kirim SMS : "UP 1234 abcd".

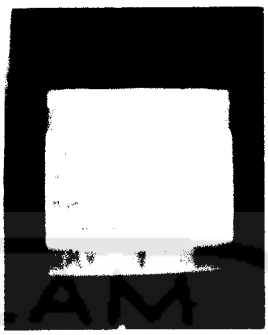
Sistem membalas SMS : "Password berhasil diubah".

Jika password baru panjang karakternya kurang dari format ditetapkan, urutan SMS seperti berikut:

User kirim SMS : "UP abcd abc".

Sistem membalas SMS : "Password baru minimal 4 digit dan maksimal 10 digit".

Tampilan SMS ponsel member sebagai berikut:



Gambar 4.66 Tampilan isi pesan password tidak sesuai format

Jika password baru panjang karakternya melebihi format ditetapkan, urutan SMS seperti berikut:

User kirim SMS : "UP abcd abcdefghijk".

Sistem membalas SMS : "Password baru minimal 4 digit dan maksimal 10 digit".

Jika password yang dikirim tidak valid:

User kirim SMS : "UP 12345 abcd"

Sistem membalas SMS : "Password salah (case sensitive) atau anda belum terdaftar".

4.2.2.6 Ubah Password

Untuk menguji hasil dari proses ubah password, dilakukan pengiriman SMS oleh *user* ke sistem. Terdapat beberapa proses pada pengujian ubah password, antara lain sebagai berikut:

Jika proses benar, urutan SMS seperti berikut:

User kirim SMS : "UP 1234 abcd".

Sistem membalas SMS : "Password berhasil diubah".

Jika password baru panjang karakternya kurang dari format ditetapkan, urutan SMS seperti berikut:

User kirim SMS : "UP 1234 abc".

Sistem membalas SMS : "Password baru minimal 4 digit dan maksimal 10 digit".

Jika password baru panjang karakternya melebihi format ditetapkan, urutan SMS seperti berikut:

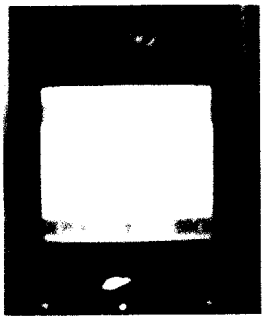
User kirim SMS : "UP 1234 abcdefghijk".

Sistem membalas SMS : "Password baru minimal 4 digit dan maksimal 10 digit".

4.2.2.7 User Sudah Terdaftar

Penanganan kesalahan ini terdapat pada proses pendaftaran. Kesalahan terjadi jika *member* yang sudah terdaftar mengulangi proses pendaftaran, maka server akan mengirimkan SMS balik yang berisi "Maaf Nomor Hp anda sudah terdaftar".

Tampilan SMS pada ponsel member sebagai berikut:



Gambar 4.67 Tampilan isi pesan member sudah terdaftar

4.2.2.8 Lihat Kota

Proses ini digunakan jika user ingin mengetahui daftar kota yang ada pada sistem. Urutan prosesnya sebagai berikut:

User kirim SMS : "LK".

Sistem membalas SMS : " Daftar kota yang ada = <daftar seluruh nama kota yang ada pada database>".

4.2.2.9 Waktu Sholat

SMS pengingat yang berisi pesan pengingat bahwa sebentar lagi sudah masuk waktu sholat bagi user yang telah terdaftar menjadi member. Waktu sholat tiap-tiap member berbeda berdasarkan kota member berada karena masing-masing kota memiliki perbedaan beberapa menit dari satu kota dan kota lainnya. Bila 10 atau 15 menit sebelum waktu sholat tiba member akan mendapat kiriman SMS berupa pesan pengingat waktu sholat.

Hasil output SMS pengingat yang dikirimkan pada member berisi "Jangan lupa untuk sholat, waktu sholat <nama sholat> pada pukul <sesuai hasil perhitungan waktu yang didapat>".

Setelah menjelajahi dunia internet, maka didapatkan program aplikasi pengingat waktu shalat untuk komputer bahkan ada program aplikasi pengingat shalat untuk pengguna *handphone*.

1. Shollu, versi 3.08.2, oleh **Ebta Setiawan,** yogyakarta 2008

Shollu merupakan *islamic software* yang mempunyai tujuan utama memberikan peringatan kepada pengguna komputer bahwa waktu sholat telah atau hampir tiba, sehingga pengguna bisa segera mempersiapkan diri untuk menunaikan sholat berjamaah. Kekurangan Aplikasi ini yaitu berlajan pada sistem operasi Windows.

2. Azan Times, Versi 2.5, oleh **Ebta Setiawan,** yogyakarta June 2008

   Azan Times merupakan *islamic java application* yang mempunyai tujuan utama memberikan peringatan kepada pengguna handphone bahwa waktu sholat telah tiba, sehingga pengguna bisa segera mempersiapkan diri untuk menunaikan sholat berjamaah. Namun kekurangan Adzan *Times* hanya berjalan di handphone yang mendukung Java-enabled.

   Dengan melihat kekurangan dari manfaat aplikasi tersebut, maka aplikasi pengingat sholat berbasis sms ini dapat digunakan di semua jenis *handphone* tanpa syarat apapun oleh semua masyarakat yang sudah terdaftar sebagai member baik yang memiliki komputer atau yang tidak memiliki komputer dapat memanfaatkannya.