BAB III

METODOLOGI

3.1 Analisis Kebutuhan Perangkat Lunak

3.1.1 Metode Analisis

Aplikasi pengingat sholat dibangun menggunakan konsep SMS *gateway*. Untuk melihat aliran proses aplikasi yang meliputi *input* dan proses *output* digambarkan dengan menggunakan bahasa pemodelan UML *(Unified Modeling Language)*. UML menggunakan notasi-notasi grafis yang menggambarkan aliran proses sistem, dimana keberadaanya dapat membantu mengkomunikasikan fungsi sistem kepada pemakai dan juga sebagai alat bantu bagi *programmer* (pemrogram) dalam membangun sistem. UML digunakan untuk menggambarkan sistem yang akan dikembangkan secara logis tanpa mempertimbangkan lingkungan fisik dimana sistem ini akan digunakan. UML aplikasi pengingat sholat menjelaskan urutan interaksi yang terjadi antara user dan sistem.

Text editor penulisan sintaks menggunakan Netbeans IDE 6.0 karena sangat mendukung bahasa pemrograman java. Pengolah data sistem memakai DBMS *(Data Base Management System)* XAMPP.

3.1.2 Hasil Analisis

Dari data hasil analisis selama penelitian dapat ditetapkan kebutuhan-kebutuhan yang akan digunakan dalam aplikasi ini. Kebutuhan tersebut meliputi: *Input* (masukan) dan *Output* (keluaran), diantaranya yaitu:

3.1.2.1 Analisis Kebutuhan Proses

Kebutuhan proses aplikasi pengingat sholat, meliputi:

a) Olah Data Member

Proses pengolahan data member meliputi: tambah, ubah dan hapus data member. Tambah data meliputi: nama, nomor *handphone* dan kota member berada. Proses pengolahan data member bertujuan agar member dapat mengolah data pribadinya. Pengolahan ini hanya diperuntukkan buat member untuk mengolah data pribadinya, seperti merubah password dan kota member berada. Pengolahan data dilakukan dengan cara member mengirimkan SMS ke sistem dengan format yang telah ditetapkan.

b) Olah data waktu sholat

Proses ini dilakukan oleh Admin dalam menentukan dan merubah jadwal waktu sholat. Jadwal sholat setiap kota berbeda tergantung letak geografis dan waktu sekarang. Untuk pengolahannya diberikan kepada admin karena dia yang bertanggung jawab. Proses pengolahan data waktu sholat meliputi: tambah, ubah dan hapus. Pemasukan data waktu sholat meliputi: nama kota,

latitude, longitude, zona waktu dan ketinggian daratan diatas permukaan laut. Sedangkan hasilnya berupa: waktu subuh, waktu terbit, waktu zuhur, waktu asyar, waktu magrib dan waktu isya.

c) Olah data SMS

Proses pengolahan SMS dilakukan oleh sistem, seperti: menerima sms yang masuk dan mengirim ulang pesan SMS kepada user. Sms yang masuk akan diproses setelah itu sistem akan mengirim SMS balasan kepada user berupa teks singkat. Jika isi SMS valid sesuai format yang telah ditetapkan maka user yang mengirimkan pesan akan dijadikan member, sedangkan jika isi SMS tidak valid sistem akan meminta user mengirimkan pesan ulang sesuai format yang telah ditetapkan.

d) Olah data admin

Proses pengolahan data admin meliputi: tambah, ubah dan hapus. Proses pengolahan data admin bertujuan agar admin dapat mengolah data pribadinya. Admin merupakan pemegang penuh aplikasi yang bertanggung jawab menjaga keamanan data sehingga admin juga perlu mengolah data pribadinya sehingga sistem menjadi lebih aman ketika admin dapat mengolah data pribadinya, seperti: ubah *username* dan ubah *password.*

e) Olah data modem

Proses pengaturan modem bertujuan menghubungkan ponsel dan komputer. Proses penyambungan ini dilakukan agar dapat mengetahui ponsel dan kabel

data yang akan digunakan dalam aplikasi pengingat sholat berbasis SMS dapat berjalan dengan baik. Pengaturan untuk setiap merek dan jenis ponsel dapat berbeda-beda tergantung dari kapasitas dan kecepatan dari ponsel tersebut. Pengaturannya meliputi: nama ponsel, port yang digunakan, bits per second, data bits, parity, stop bits dan flow control. Karena tiap ponsel berbeda maka dalam pengaturannya juga berbeda tergantung merk dan jenis ponsel yang digunakan.

f) Pendaftaran

Sebelum user ingin memanfaatkan aplikasi pengingat sholat terlebih dahulu sistem akan meminta user mengirimkan data pribadinya untuk dijadikan sebagai member, proses ini dinamakan proses pendaftaran atau *registrasi*.

g) Login

Proses login dilakukan admin sebelum memasuki sistem aplikasi pengingat sholat. Manfaat dibuatkan proses login agar keamanan data semakin terjaga. Proses login dimulai setelah admin memasukkan *username* dan *password*

h) Ubah password

Fasilitas ini dapat dimanfaatkan admin dan member untuk dapat merubah password ketika login. Fasilitas ubah password dibuat agar admin atau member lebih mudah dalam penggunaannya.

i) Waktu sholat

Fasilits ini khusus diberikan kepada member yang dikirim dari sistem. Sistem akan mengirimkan pesan teks kepada member yang berisi data waktu sholat jika ± 10 atau 15 menit sebelum waktu sholat dimulai.

3.1.2.2 Analisis Kebutuhan Input

Input dari aplikasi pengingat sholat terdiri terdiri dari 2 user, yaitu Admin dan Member. Admin memiliki wewenang penuh dalam pengolahan data yang ada didalam database. Sedangkan member hanyalah user yang hanya bisa memasukkan data pribadi dan perubahan password.

a. Input Admin

Input ini merupakan masukkan dari admin yaitu data *modem* dan data waktu sholat.

1. Input data modem

Input data *modem* merupakan segala sesuatu yang berhubungan dengan koneksi antara ponsel dan komputer. Kedua perangkat keras tersebut dihubungkan sehingga dapat digunakan dalam menjalankan aplikasi pengingat waktu sholat yang berbasiskan SMS.

2. Input data waktu sholat

Input data waktu sholat meliputi segala sesuatu yang berhubungan dengan data kota, seperti: nama, letak lintang, letak bujur, zona

waktu, ketinggian daratan diatas permukaan laut. Untuk menentukan waktu sholat terlebih dahulu harus diketahui letak geografis kota yang bersangkutan karena waktu sholat tiap-tiap kota berbeda.

b. Input Pengguna

Input pengguna adalah masukan data pribadi yang dikirimkan melalui SMS berupa: nama, nomor *handphone* dan kota dimana dia tinggal. Sehingga dengan data pribadi yang dikirimkan tersebut sistem dapat menentukan jadwal waktu sholat masing-masing member.

3.1.2.3 Analisis Kebutuhan Output

Output atau keluaran yang diterima member berupa teks pengingat waktu sholat bahwa sholat akan dimulai sesuai waktu yang telah ditentukan melalui pesam singkat SMS. Teks pengingat ini akan dikirimkan kepada member ± 5 atau 10 menit sebelum sholat dilaksanakan.

3.1.3 Kebutuhan Antarmuka

Antarmuka menggunakan NETBEANS IDE 6.0 karena termasuk *software Open Source* sekaligus sangat mendukung bahasa pemrograman java. Untuk membangun aplikasi membutuhkan *form* yang berfungsi mengolah data, seperti: ubah, tambah dan hapus. Dengan memanfaatkan *tool* yang ada pada NETBEANS IDE 6.0 *form* yang dibutuhkan dapat dibuat.

3.2 Perancangan Perangkat Lunak

3.2.1 Metode Perancangan

Dalam pembuatan aplikasi pengingat waktu sholat secara keseluruhan menggunakan 2 metode yaitu pemrograman berorientasi objek. Pemrograman berorientasi objek pada dasarnya menekankan pada kemudahan perawatan serta pengembangan sistem tetapi agak lebih rumit ketimbang pemrograman secara prosedural. Pemrograman berorientasi objek memiliki kelas yang berbeda-beda, ada kelas yang khusus berfungsi untuk menampilkan data, meyambung ke database, mengedit dan menghapus data. Untuk bahasa pemodelan yang digunakan dalam menspesifikasikan, memvisualisasikan membangun dan dokumentasi, adalah bahasa pemodelan UML *(Unified Modeling Language).*

3.2.2 Hasil Perancangan

Berdasarkan hasil analisis yang telah dilakukan dapat dirumuskan apa-apa saja yang menjadi masukan, proses, keluaran dan antarmuka sistem. Hasil perancangan dibedakan menjadi beberapa bagian sesuai dengan bagian-bagian yang digunakan pada metode perancangan dan bahasa pemodelan UML.

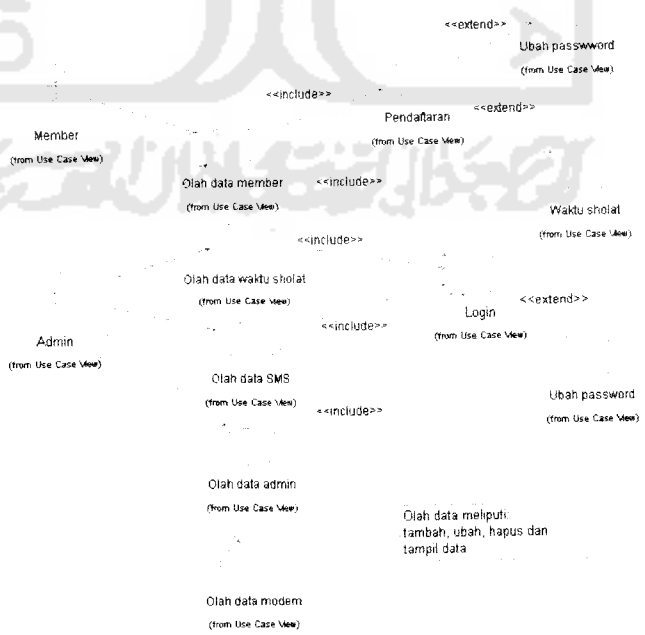
3.2.2.1 Perancangan UML *(Unified Modeling Language)*

3.2.2.1.1 Use Case Diagram

Use case diagram bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) sebuah sistem dengan suatu sistem tersendiri melalui sebuah cerita bagaimana sebuah sistem digunakan. Urutan langkah-langkah yang

menerangkan antar pengguna dan sistem disebut skenario. Setiap skenario mendeskripsikan urutan kejadian. Setiap urutan diinisialisasi oleh orang, sistem yang lain, perangkat keras atau urutan waktu. Selain itu *use case* juga diartikan sebagai abstraksi dari interaksi antara sistem dan *aktor*.

Dalam aplikasi pengingat waktu sholat berbasis SMS menjelaskan hubungan antara sistem dan *aktor*. Hubungan ini bisa digambarkan melalui input *aktor* ke sistem dan *output* yang dihasilkan sistem ke *aktor*, hubungan antara *aktor* dan sistem merupakan hubungan timbal balik. *Use case* merupakan dokumen yang menggambarkan urutan kejadian *aktor* dalam menggunakan sistem untuk menyelesaikan proses yang sedang dikerjakan. Berikut adalah gambar yang menjelaskan urutan kejadian hubungan timbal balik antara *aktor* ke sistem dalam aplikasi pengingat waktu sholat, seperti terlihat dalam gambar 3.1



Gambar 3.1 Use case diagram

Gambar 3.1 *use case* menjelaskan ada 2 user yang menggunakan aplikasi pengingat sholat yaitu admin dan member. Kedua user memiliki hak askes berbeda dalam menggunakan aplikasi pengingat sholat.

a) Olah Data Member

   Proses pengolahan data member meliputi: tambah, ubah dan hapus data member.

b) Olah data waktu sholat

   Proses ini dilakukan oleh Admin dalam menentukan dan merubah jadwal sholat. Proses pengolahan data waktu sholat meliputi: tambah, ubah dan hapus data.

c) Olah data SMS

   Proses pengolahan SMS dilakukan oleh sistem, seperti: menerima sms yang masuk dan mengirim ulang pesan SMS kepada user.

d) Olah data admin

   Proses pengolahan data admin meliputi: tambah, ubah dan hapus data admin.

e) Olah data modem

   Proses pengaturan modem bertujuan menghubungkan ponsel dan komputer. Pengaturannya meliputi: nama ponsel, port yang digunakan, bits per second, data bits, parity, stop bits dan flow control.

f) Pendaftaran

Sebelum user ingin memanfaatkan aplikasi pengingat sholat terlebih dahulu sistem akan meminta user mengirimkan data pribadinya untuk dijadikan sebagai member, proses ini dinamakan proses *registrasi* atau pendaftaran.

g) Login

Proses login dilakukan admin sebelum memasuki sistem aplikasi pengingat sholat. Manfaat dibuatkan proses login agar keamanan data semakin terjaga. Proses login dimulai setelah admin memasukkan *username* dan *password.*

h) Ubah password

Fasilitas ini dapat dimanfaatkan admin dan member untuk dapat merubah password.

i) Waktu sholat

Sistem akan mengirimkan pesan teks kepada member yang berisi data waktu sholat.

3.2.2.1.2 Class Diagram

Class diagram menggambarkan struktur dan deskripsi *class*, *package* dan *object* beserta hubungan satu dan yang lainnya. *Class diagram* digunakan untuk melakukan visualisasi struktur kelas-kelas dari suatu *system* dan merupakan tipe diagram yang paling banyak digunakan. *Class diagram* dapat memperlihatkan hubungan antar kelas dan penjelasan detail tiap-tiap *class* didalam model desain *(logical view)* dari sebuah *system*. Dalam proses pendesainan *Class diagram*

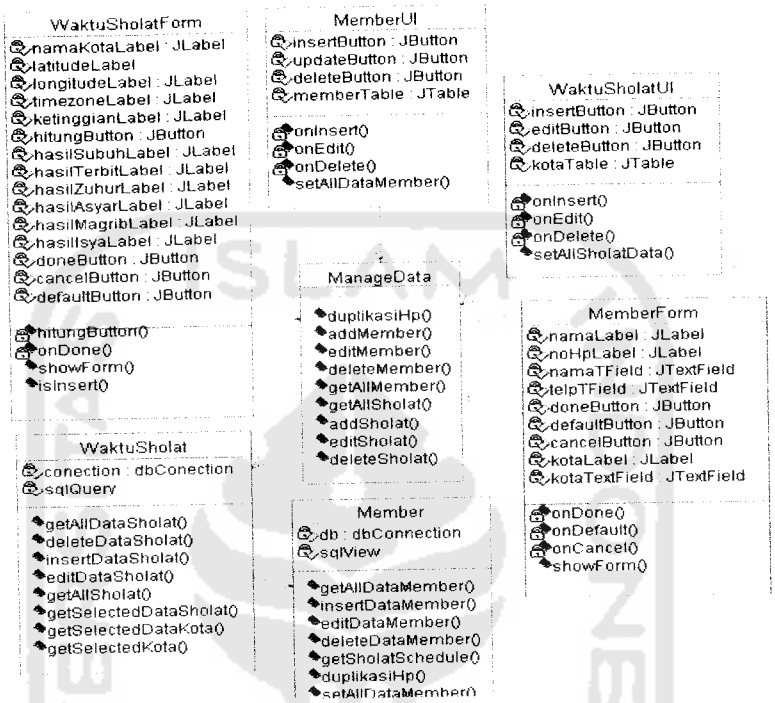
digunakan untuk menangkap struktur dari semua kelas yang merupakan komponen dari sebuah sistem.

Aplikasi pengingat sholat terdiri dari 3 buah modul utama yaitu: modul *data management,* modul *system management* dan modul *user management.* Kesemua modul tersebut memiliki fungsi yang berbeda-beda sesuai dengan namanya masing-masing. Berikut ini akan dijelaskan kelas-kelas apa saja yang ada didalam ketiga modul tersebut beserta hubungan antar kelasnya.

3.2.2.1.2.1 Modul *Data Management*

Modul *data management* terdiri dari data member dan data waktu sholat. Didalam data waktu sholat terdapat letak geografis dan zona waktu kota. Pengolahan data modul *data management* meliputi: tambah, ubah, tampilkan dan hapus. Ketika sistem masih dalam keadaan kosong dapat ditambahkan data baru, kemudian jika data yang sudah ada mengalami perubahan dapat diubah sekaligus bila data sudah tidak terpakai data dapat dihapus.

Sistem juga mampu menampilkan data yang ada dalam database sehingga proses pengolahan data semakin mudah. Dalam modul ini terdapat kelas-kelas yang saling terhubung antar satu kelas dan kelas yang lain. Kelas-kelas yang ada diberi nama sesuai dengan fungsinya masing-masing sehingga lebih mudah dalam penggunaanya berdasarkan nama kelas tersebut. Gambar di bawah ini menjelaskan hubungan antar kelas seperti pada gambar 3.2

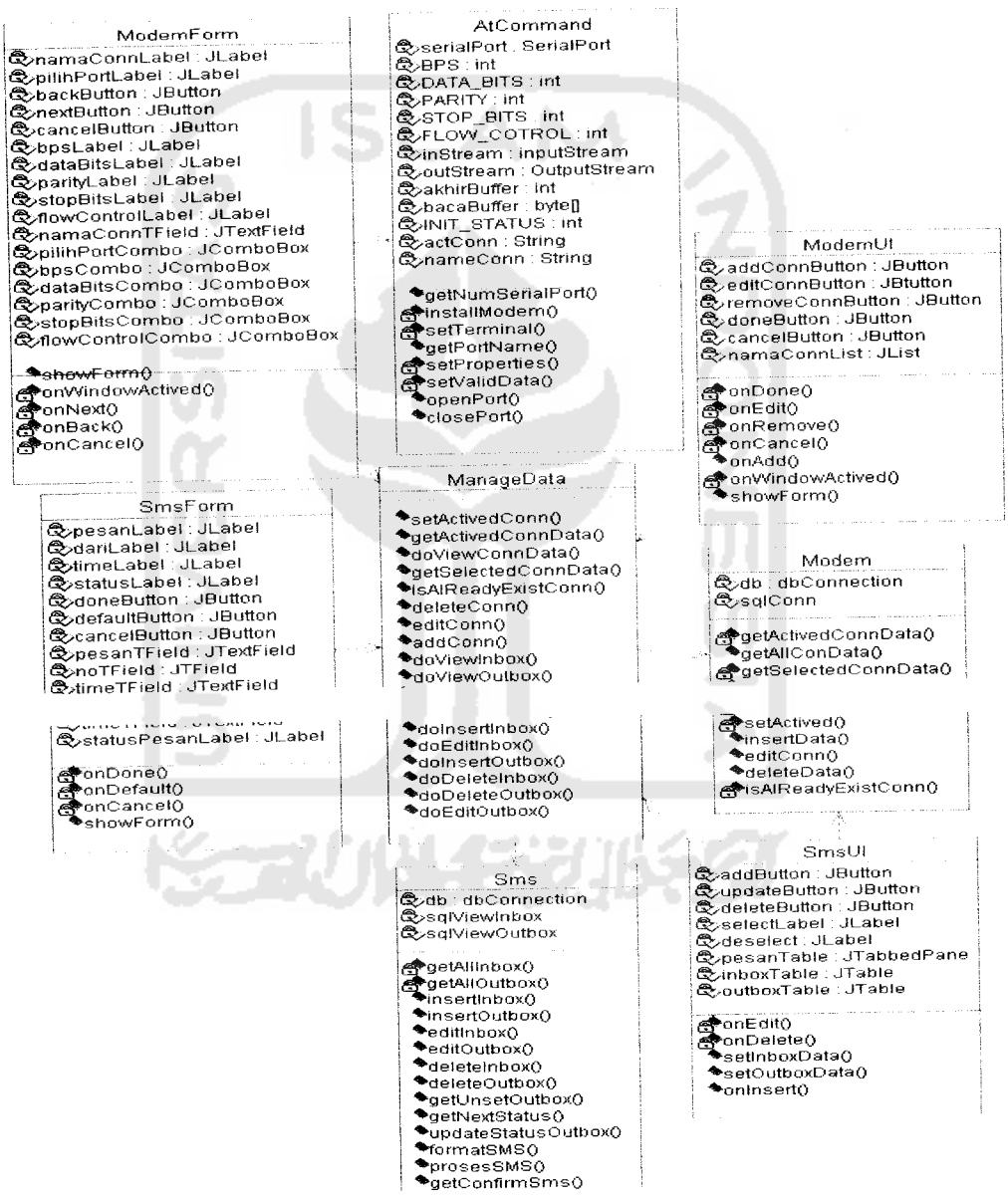


Gambar 3.2 Class diagram modul data management

3.2.2.1.2.2 Modul *System Management*

Modul *system management* berfungsi mengatur setiap sms yang masuk dan sms yang keluar. Modul ini juga memiliki kelas yang saling terelasi sama seperti modul *data management.* Modul *system management* menangani konfigurasi sistem dan data-data lain pendukung kinerja sistem, misalnya pencatatan SMS (SMS *log*). SMS *log* diperlukan untuk mengetahui apakah sistem sedang mengalami masalah atau tidak. Masalah tersebut dapat berupa kegagalan sistem dalam melakukan pengiriman SMS secara otomatis. Modul ini terdapat beberapa kelas yang saling

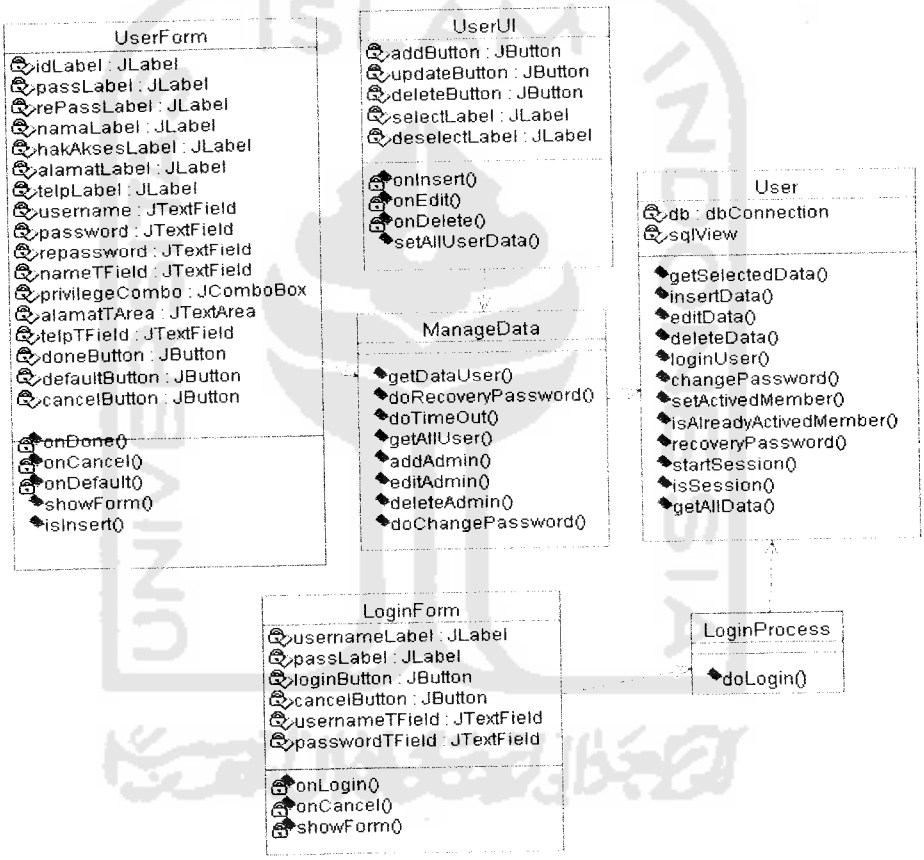
terelasi. Gambar di bawah ini menjelaskan relasi yang terjadi antar kelas, lebih jelasnya ditunjukkan oleh gambar 3.3



Gambag 3.3 Class diagram modul system management

3.2.2.1.2.3 Modul *User Management*

Modul *user management* menangani data *user, user* yang dimaksud yaitu *Admin.* Relasi yang terjadi akan dijelaskan pada gambar 3.4



Gambar 3.4 Class diagram modul user management

3.2.2.1.3 Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk *user,* antarmuka dan sebagainya). *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang

dilakukan sebagai *response* dari sebuah *event* untuk menghasilkan *output.* Dalam *sequence diagram* setiap *object* hanya memiliki garis yang digambarkan garis putus-putus kebawah. Pesan antar *object* digambarkan dengan anak panah dari *object* yang mengirimkan pesan ke *object* yang menerima pesan.

Pada bagian ini akan dijelaskan *sequence diagram* dari sistem pengingat sholat berbasis SMS. Dalam *sequence diagram* ini akan dijelaskan gambaran interaksi antara objek secara berurutan. Penjelasan *sequence diagram* akan dijelaskan berdasarkan modul yang digunakan yaitu modul *data management,* modul *system management* dan modul *user management.*

3.2.2.1.3.1 Modul Data Management

Modul *data management* terdiri dari data member dan data waktu sholat. Didalam data waktu sholat terdapat nama kota, letak lintang, letak bujur, zona waktu, ketinggian diatas permukaan laut, waktu subuh, waktu terbit, waktu zuhur, waktu asyar, waktu magrib dan waktu isya. Berikut ini akan dijelaskan mengenai *sequence diagram* dari *data management.*

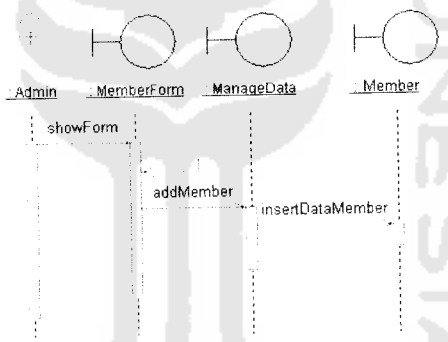
3.2.2.1.3.1.1 Tambah Data Member

Objek yang berhubungan dengan *sequence diagram* ini adalah sebagai berikut:

Aktor	: *Admin*
Kelas pembatas	: *MemberForm*
Kelas kontrol	: *ManageData*
Kelas *Entity*	: *Member*
Urutan proses	:

1. *Admin* memulai *sequence* ini dengan memanggil method *showForm()*.
2. *MemberForm* memanggil kelas *ManageData* dan memanggil *method addMember* untuk memproses *input* dari *Admin*.
3. *manageData* melakukan instansiasi ke kelas *member* dan memanggila *method insertDataMember()* untuk memasukkan data *member* ke dalam basis data *member*.

Sequence diagram tambah data *member* dapat dilihat pada gambar 3.5



Gambag 3.5 Sequence diagram tambah data member

3.2.2.1.3.1.2 Ubah Data Member

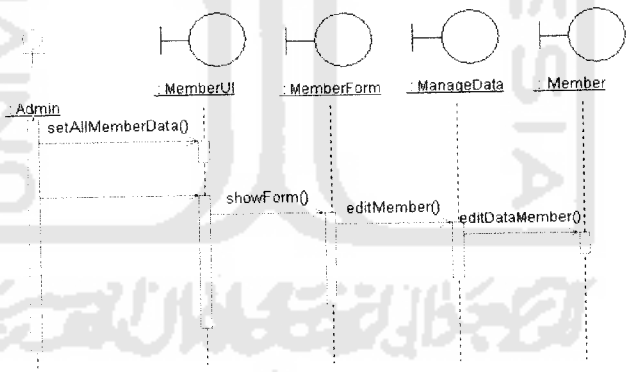
Objek yang berkaitan dengan *sequence diagram* ini adalah sebagai berikut:

Aktor : *Admin*
Kelas pembatas : *MemberForm, MemberUI*
Kelas kontrol : *ManageData*
Kelas *Entity* : *Member*
Urutan proses :

1. *Admin* memulai *sequence* ini dengan menampilkan semua data member menggunakan perintag *sqlView()*.

2. *MemberUI* melakukan instansiasi ke *manageData* dan memanggil *method setAllMember()* untuk menampilkan semua data *member.*
3. *MemberUI* melakukan instansiasi ke *MemberForm* dengan memanggil *method showForm()* untuk menampilkan *form member.*
4. *MemberForm* melakukan instansiasi ke *ManageData* dan memanggil *method editMember()* untuk memproses data yang telah diubah.
5. *ManageData* melakukan instansiasi ke *Member* dan memanggil *method editDataMember()* untuk mengubah data *member.*

Sequence diagram ubah data member dapat dilihat pada gambar 3.6



Gambar 3.6 Sequence diagram ubah data member

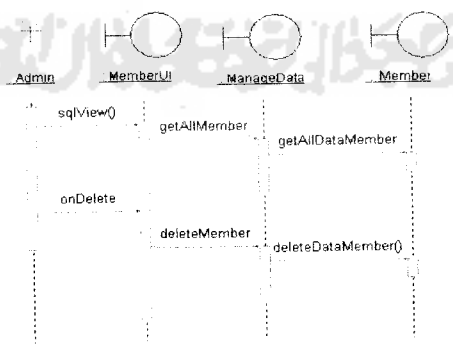
3.2.2.1.3.1.3 Hapus Data Member

Objek yang berkaitan dengan *sequence diagram* ini adalah sebagai berikut:

Aktor	: *Admin*
Kelas pembatas	: *MemberUI*
Kelas kontrol	: *ManageData*
Kelas *entity*	: *Member*
Urutan proses	:

1. *Admin* mengawali *sequence* ini dengan menampilkan semua data *member* menggunakan method *getAllMember()*.
2. *MemberUI* melakukan instansiasi ke *ManageData* dan memanggil *method getAllMemberData()* untuk menampilkan semua data *member*.
3. ManageData melakukan instansiasi ke Member dan memanggil method getAllDataMember() untuk mengambil semua data member
4. Admin kemudian memanggil *method deleteMember()*.
5. *MemberUI* melakukan instansiasi ke *ManageData* dan memanggil *method deleteMember()* untuk melakukan penghapusan data *member* yang telah diseleksi.
6. *ManageData* melakukan instansiasi ke *Member* dan memanggil *method deleteDataMember()* untuk melakukan penghapusan data *member* yang telah dipilih.

Sequence diagram hapus data *member* dapat dilihat pada gambar 3.7



Gambar 3.7 Sequence diagram hapus data member

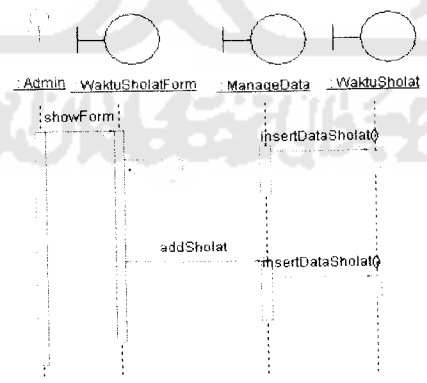
3.2.2.1.3.1.4 Tambah Data Waktu Sholat

Objek yang berhubungan dengan *sequence diagram* ini adalah sebagai berikut:

Aktor : *Admin*
Kelas pembatas : *WaktuSholatForm*
Kelas kontrol : *ManageData*
Kelas *Entity* : *WaktuSholat*
Urutan proses :

1. *Admin* memulai *sequence* ini dengan memanggil method *showForm()*.
2. *WaktuSholatForm* memanggil kelas *ManageData* dan memanggil *method addSholat* untuk memproses *input* dari *Admin*.
3. *manageData* melakukan instansiasi ke kelas kota dan memanggil *method insertDataSholat()* untuk memasukkan data *Waktu sholat* ke dalam basis data.

Sequence diagram tambah data waktu sholat dapat dilihat pada gambar 3.8



Gambar 3.8 Sequence diagram tambah data waktu sholat

Proses penambahan data waktu sholat berbeda dengan proses penambahan data lainnya. Karena form waktu sholat terdapat masukkan data dan keluaran data

yang kemudian data tersebut disimpan kedalam database. Masukkan atau *input* data berupa: nama kota, letak lintang *(latitude)*, letak bujur *(longitude)*, zona waktu dan ketinggian daratan diatas permukaan air laut. Sedangkan keluaran atau *output* yang dihasilkan form waktu sholat berupa: waktu subuh, waktu terbit, waktu zuhur, waktu asyar, waktu magrib dan waktu isya. Setelah semua data masukkan yang dibutuhkan lengkap maka proses penghitungan waktu sholat baru bisa dijalankan dan akan menghasilkan keluaran waktu sholat dalam format jam.

Algoritma penghitungan waktu sholat adalah sebagai berikut [SET08]:

1) Untuk dapat menghitung waktu sholat yang tepat pada suatu daerah terlebih dahulu harus mengetahui letak lintang (B), letak bujur (L), zona waktu (TZ) dan ketinggian daratan diatas permukaan air laut (H).

   Masukkan nilai:

   a. letak lintang *(latitude)* -> **B**
   b. letak bujur *(longitude)* -> **L**
   c. zona waktu *(timezone)* -> **TZ**
   d. ketinggian -> **H**

2) Dilanjutkan dengan proses mencari nilai `J`, `beta (β)`, `D`, `T`, `R`, `Gd`, `Gn`, `Z`, `U`, `Vd`, `Vn` dan `W`.

Nama	Keterangan	Nama	Keterangan
J	Waktu sekarang berupa: tanggal, bulan dan tahun.	Gn	Sudut awal matahari terbenam(15°-19°)
beta (β)	Sudut tahun	Z	Penyamaan untuk mengkalkulasi waktu
D	Deklinasi matahari	U	Penyamaan untuk mengkalkulasi waktu
T	Persamaan waktu	Vd	Penyamaan untuk mengkalkulasi waktu
R	Nilai referensi garis bujur	Vn	Penyamaan untuk mengkalkulasi waktu
Gd	Sudut awal matahari terbit(15°-19°)	W	Penyamaan untuk mengkalkulasi waktu

Di bawah ini algoritma dan rumus untuk mendapatkan nilai D dan T.

```
π = 3.1415926536 //nilai sudah ditentukan
```

Nilai yang dicari	Rumus	Sintaks Pemrograman
beta (β)	$\beta = \frac{2\pi(J)}{365}$	`beta = 2 * Math.PI * J / 365`

Nilai yang dicari	Rumus	Sintaks pemrograman
Deklinasi matahari (D)	$\frac{180}{\pi}*\{0.006918 - [0.399912*\text{Cos}[\beta]] + [0.070257*\text{Sin}[\beta]] - [0.006758*\text{Cos}[2*\beta]] + [0.000907*\text{Sin}[2*\beta]] - [0.002697*\text{Cos}[3*\beta]] + [0.001480*\text{Sin}[3*\beta]]\}$	**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**)));
Persamaan waktu (T)	$229.18*\{0.000075 + [0.001868*\text{Cos}[\beta]] - [0.032077*\text{Sin}[\beta]] - [0.014615*\text{Cos}[2*\beta]] - [0.040849*\text{Sin}[2*\beta]]\}$	**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**)));
Nilai referensi garis bujur (R)	15*TZ (Degrees)	**R** = 15 * **TZ**;
Nilai sudut matahari terbit (Gd)	15	15
Nilai sudut matahari terbenam (Gn)	15	15

Nilai yang dicari	Rumus	Sintaks pemrograman
Z	$12 + \frac{R - L}{15} - \frac{T}{60}$	`Z = 12 + ((R - L)/15) - (T/60);`
U	$U = \frac{180}{15 * \pi} * ArcCos \left[\frac{Sin\left[(-0.8333 - 0.0347 * Sign(H) * Abs(H)^{0.5}) * \frac{\pi}{180}\right] - Sin\left[D * \frac{\pi}{180}\right] * Sin\left[B * \frac{\pi}{180}\right]}{Cos\left[D * \frac{\pi}{180}\right] * Cos\left[B * \frac{\pi}{180}\right]} \right]$	`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))));`
Vd	$Vd = \frac{180}{15 * \pi} * ArcCos \left[\frac{-Sin\left[Gd * \frac{\pi}{180}\right] - Sin\left[D * \frac{\pi}{180}\right] * Sin\left[B * \frac{\pi}{180}\right]}{Cos\left[D * \frac{\pi}{180}\right] * Cos\left[B * \frac{\pi}{180}\right]} \right]$	`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))));`

Nilai yang dicari	Rumus	Sintaks pemrograman
Vn	$Vn = \frac{180}{15 * \pi} * ArcCos \left[\frac{-Sin\left[Gn * \frac{\pi}{180}\right] - Sin\left[D * \frac{\pi}{180}\right] * Sin\left[B * \frac{\pi}{180}\right]}{Cos\left[D * \frac{\pi}{180}\right] * Cos\left[B * \frac{\pi}{180}\right]} \right]$	`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))));`
W	$W = \frac{180}{15 * \pi} * ArcCos \left[\frac{Sin\left[ArcCot\left[\frac{Sh + Tan}{Abs(B - D) * \frac{\pi}{180}}\right]\right] - Sin\left[D * \frac{\pi}{180}\right] * Sin\left[B * \frac{\pi}{180}\right]}{Cos\left[D * \frac{\pi}{180}\right] * Cos\left[B * \frac{\pi}{180}\right]} \right]$	`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.cos(D*Math.PI/180)*Math.cos(B*Math.PI/180)));`

3) Setelah nilai diatas didapatkan dilanjutkan menghitung waktu sholat:

- waktu subuh -> **Z-Vd;**
- waktu matahari terbit -> **Z-U;**
- waktu zuhur -> **Z;**

- waktu asyar -> Z+W;
- waktu magrib -> Z+U;
- waktu isya -> Z+Vn;

4) Menampilkan hasil.

5) Selesai.

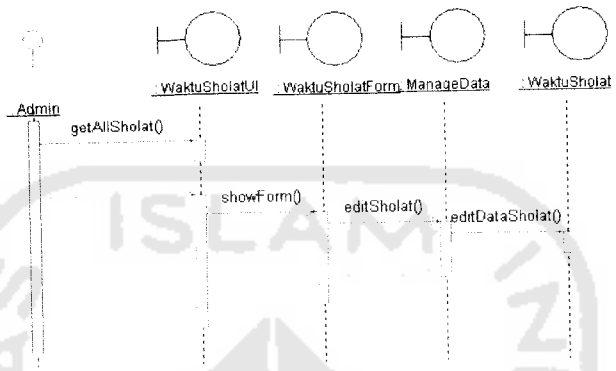
3.2.2.1.3.1.5 Ubah Data Waktu Sholat

Objek yang berkaitan dengan *sequence diagram* ini adalah sebagai berikut:

Aktor : *Admin*
Kelas pembatas : *WaktuSholatForm, WaktuSholatUI*
Kelas kontrol : *ManageData*
Kelas *Entity* : *WaktuSholat*
Urutan proses :

1. *Admin* memulai *sequence* ini dengan menampilkan semua data waktu sholat.
2. *WaktuSholatUI* melakukan instansiasi ke *manageData* dan memanggil *method getAllSholat()* untuk menampilkan semua data waktu sholat.
3. *WaktuSholatUI* melakukan instansiasi ke *WaktuSholatForm* dengan memanggil *method showForm()* untuk menampilkan *form* waktu sholat.
4. *WaktuSholatForm* melakukan instansiasi ke *ManageData* dan memanggil *method editSholat()* untuk memproses data yang ingin diubah.
5. *ManageData* melakukan instansiasi ke *WaktuSholat* dan memanggil *method editDataSholat()* untuk mengubah data sholat.

Sequence diagram ubah data waktu sholat dapat dilihat pada gambar 3.9



Gambar 3.9 Sequence diagram ubah data waktu sholat

3.2.2.1.3.1.6 Hapus Data Waktu Sholat

Objek yang berkaitan dengan *sequence diagram* ini adalah sebagai berikut:

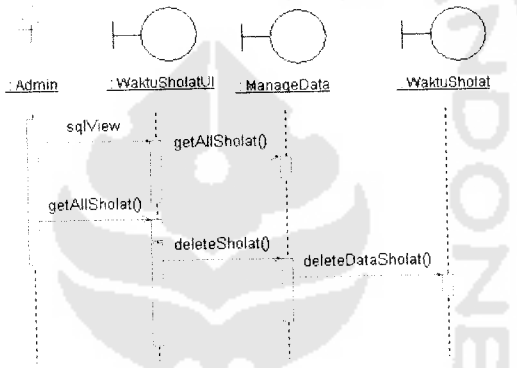
Aktor : *Admin*
Kelas pembatas : *WaktuSholatUI*
Kelas kontrol : *ManageData*
Kelas *entity* : *WaktuSholat*
Urutan proses :

1. *Admin* mengawali *sequence* ini dengan menampilkan semua data sholat menggunakan perintah *sqlView()*.
2. *WaktuSholatUI* melakukan instansiasi ke *ManageData* dan memanggil *method getAllSholat()* untuk menampilkan semua data sholat yang ada dalam data base.
3. *ManageData* melakukan instansiasi ke *WaktuSholat* dan memanggil method *getAllSholat()* untuk mengambil semua data sholat.
4. Admin kemudian memanggil *method deleteSholat()*.
5. *WaktuSholatUI* melakukan instansiasi ke *ManageData* dan memanggil *method deleteSholat()* untuk

melakukan penghapusan data sholat yang telah diseleksi.

6. *ManageData* melakukan instansiasi ke *JadwalSholat* dan memanggil method *deleteDataSholat()* untuk melakukan penghapusan data yang telah dipilih.

Sequence diagram hapus data waktu sholat dapat dilihat pada gambar 3.10



Gambar 3.10 Sequence diagram hapus data waktu sholat

3.2.2.1.3.2 Modul System Management

Modul *system management* dalam aplikasi ini berfungsi mengatur segala transaksi yang terjadi di sistem baik berupa *input* dari *user* atau *output* yang dihasilkan sistem. Data-data itu disimpan didalam database modem dan database sms. Modul *system management* digunakan untuk mengatur data modem dan SMS. Berikut ini akan dijelaskan menggunakan *sequence diagram* cara kerja modul *system management* pada aplikasi pengingat sholat berbasis SMS.

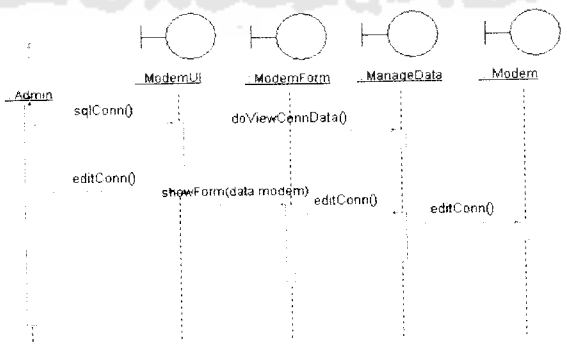
3.2.2.1.3.2.1 Tambah Data Modem

Objek yang akan menjelaskan *sequence* tambah data modem adalah sebagai berikut:

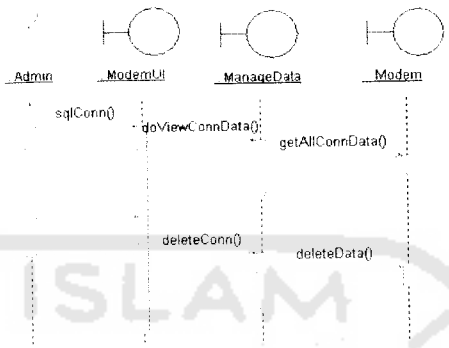
Aktor : *Admin*
Kelas pembatas : *ModemForm*

1. *Admin* mengawali *sequence* ini dengan menampilkan semua semua data modem .
2. *ModemUI* melakukan instansiasi ke manageData dan memanggil *method* *doViewConnData()* untuk menampilkan data.
3. *ManageData* melakukan instansiasi ke kelas Modem dan memanggil *method* *getAllConnData()* untuk mengambil semua data modem.
4. *Admin* kemudian memanggil method *editConn()* untuk melakukan pengeditan terhadap data modem.
5. *ModemUI* melakukan instansiasi ke ModemForm dengan memanggil *method* *showForm()* untuk menampilkan form modem .
6. *ModemForm* melakukan instansiasi ke ManageData dan memanggil method *editConn()* untuk memproses data yang diubah.
7. *ManageData* melakukan instansiasi ke modem dan memanggil method *editConn()* untuk mengubah data modem.

Sequence diagram ubah `data modem` dapat dilihat pada gambar 3.12



Gambar 3.12 Sequence diagram ubah data modem



Gambar 3.13 Sequence diagram hapus data modem

3.2.2.1.3.2.4 Ubah Data SMS

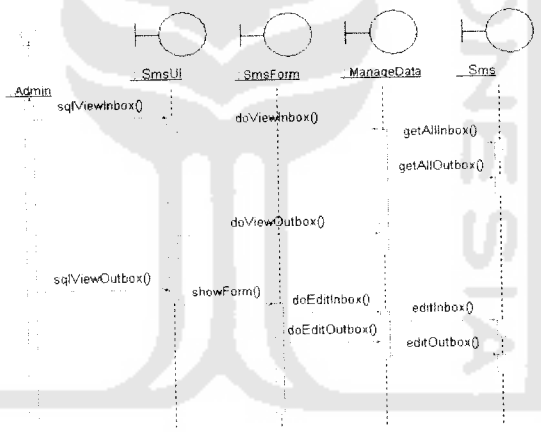
Objek yang berkaitan dengan *sequence* ini adalah sebagai berikut:

Aktor : *Admin*
Kelas pembatas : *SmsUI, SmsForm*
Kelas kontrol : *ManageData*
Kelas entity : *Sms*
Urutan proses :

1. *Admin* memulai dengan cara menampilkan data sms menggunakan perintah *sqlViewInbox()* dan *sqlViewOutbox()*.
2. *SmsUI* melakukan instansiasi ke *ManageData* dan memanggil method *doViewInbox()* dan *doViewOutbox()* untuk memerintahkan agar semua data SMS ditampilkan.
3. *ManageData* melakukan instansiasi ke *Sms* dan memanggil method *getAllInbox()* dan *getAllOutbox()* untuk mengambil semua data SMS.
4. *Admin* kemudian memanggil *method editInbox()* dan *editOutbox()*.

5. *SmsUI* melakukan instansiasi ke *SmsFormUI* dan memanggil method *showForm()* untuk menampilkan form.
6. *SmsFormUI* melakukan instansiasi ke *ManageData* dan memanggil method *doEditInbox()* dan *doEditOutbox()* untuk memproses data yang diubah.
7. *ManageData* melakukan instansiasi ke *Sms* dan memanggil method *editInbox()* dan *editOutbox()* untuk mengubah data SMS.

Sequence diagram ubah data sms dapat dilihat pada gambar 3.14



Gambar 3.14 Sequence diagram ubah data sms

3.2.2.1.3.2.5 Hapus Data Sms

Objek yang berkaitan dengan *sequence* ini adalah sebagai berikut:

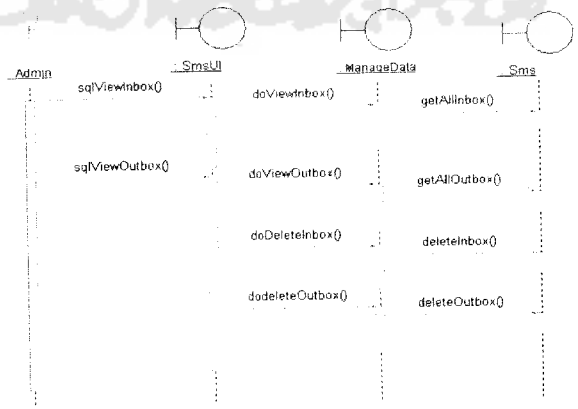
Aktor : *Admin*
Kelas pembatas : *SmsUI, SmsForm*
Kelas kontrol : *ManageData*
Kelas entity : *Sms*
Urutan proses :

1. *Admin* memulai *sequence* ini dengan menampilkan semua data sms menggunakan perintah *sqlViewInbox()*

untuk sms yang masuk dan *sqlViewOutbox()* untuk sms yang keluar.

2. *SmsUI* melakukan instansiasi ke *ManageData* dan memanggil method *doViewInbox()* dan *doViewOutbox()* untuk memerintahkan agar semua data SMS ditampilkan.
3. *ManageData* melakukan instansiasi ke *Sms* dan memanggil method *getAllInbox()* dan *getAllOutbox()* untuk mengambil semua data SMS.
4. *Admin* kemudian memanggil *method doDeleteInbox()* dan *doDeleteOutbox()*.
5. *SmsUI* melakukan instansiasi ke *ManageData* dan memanggil method *doDeleteInbox()* dan *doDeleteOutbox()* untuk melakukan proses penghapusan.
6. *ManageData* melakukan instansiasi ke Sms dan memanggil method *deleteInbox()* dan *deleteOutbox()* untuk menghapus data SMS.

Sequence diagram hapus data sms dapat dilihat pada gambar 3.15



Gambar 3.15 Sequence diagram hapus data sms

3.2.2.1.3.3 Modul User Management

Modul *User Management* difungsikan untuk mengatur atau mengelola penggunaan aplikasi pengingat waktu sholat. User yang dimaksud disini adalah *Admin.* Berikut ini akan dijelaskan menggunakan *sequence diagram* penggunaan modul *user management.*

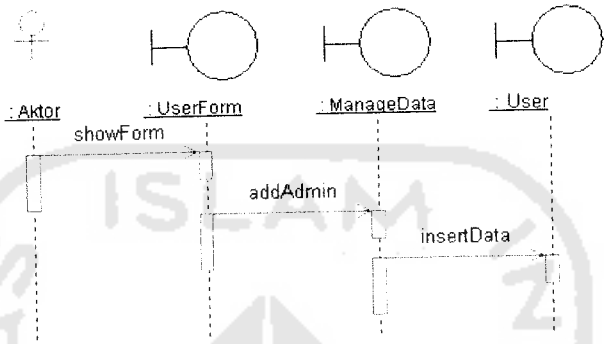
3.2.2.1.3.3.1 Tambah Data Admin

Objek yang berkaitan dengan *sequence diagram* ini adalah sebagai berikut:

Aktor : *Admin*
Kelas pembatas : *UserForm*
Kelas kontrol : *ManageData*
Kelas entity : *User*
Urutan proses :

1. *Admin* memulai *sequence* ini dengan memanggil method showForm.
2. UserUI memanggil kelas ManageData *insertData* untuk memproses input dari *Admin.*
3. ManageData memanggil kelas User dan memanggil *method insertData* untuk memasukkan data input dari *Admin.*

Sequence diagram tambah data *Admin* dapat dilihat pada gambar 3.16



Gambar 3.16 Sequence diagram tambah data admin

3.2.2.1.3.3.2 Ubah Data Admin

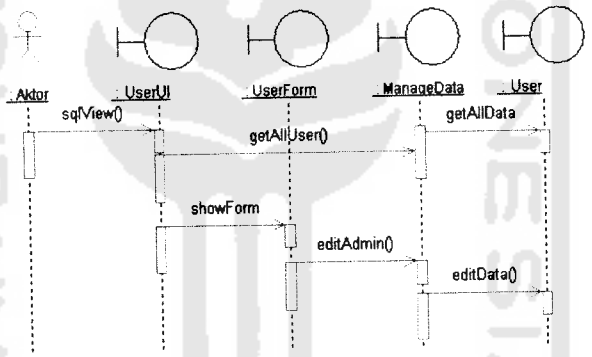
Objek dari *sequence diagram* ini adalah sebagai berikut:

Aktor : *Admin*
Kelas pembatas : *UserUI, UserForm*
Kelas kontrol : *ManageData*
Kelas entity : *User*
Urutan proses :

1. *Admin* mengawali sequence ini dengan memanggil method sqlView.
2. *LoginFormUI* melakukan instansiasi ke *ManageData* dan memanggil method *getAllUser()* untuk memerintahkan agar data ditampilkan.
3. *ManageData* melakukan instansiasi ke *User* dan memanggil method *getAllUser()* untuk mengambil semua data user.
4. *Admin* kemudian memanggil method *editAdmin()*.

5. *UserUI* melakukan instansiasi ke *UserFormUI* dan memanggil method *showForm()* untuk menampilkan form.
6. *UserFormUI* melakukan instansiasi ke *ManageData* dan memanggil method *editAdmin()* untuk memproses data diubah.
7. *ManageData* melakukan instansiasi ke *User* dan memanggil method *editData()* untuk mengubah data.

Sequence diagram dapat dilihat pada gambar 3.17



Gambar 3.17 Sequence diagram ubah data admin

3.2.2.1.3.3.3 Hapus Data Admin

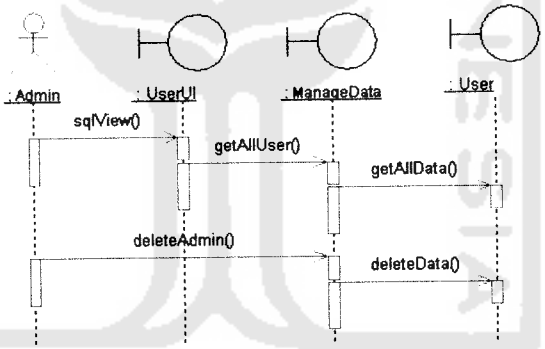
Objek dari *sequence diagram* ini adalah sebagai berikut:

Aktor	: *Admin*
Kelas pembatas	: *UserUI*
Kelas kontrol	: *ManageData*
Kelas entity	: *User*
Urutan proses	:

1. *Admin* mengawali *sequence* ini dengan memanggil method sqlView.

2. *UserUI* memanggil kelas *ManageData* dan memanggil method *getAllUser()* untuk menampilkan data.
3. *ManageData* memanggil kelas *User* dan memanggil method *getAllData()* untuk mengambil semua data.
4. *Admin* kemudian memanggil *method deleteAdmin()*.
5. *UserUI* memanggil kelas *ManageData* dan memanggil method *deleteAdmin()* untuk melakukan proses penghapusan.
6. *ManageData* memanggil kelas *User* dan memanggil method *deleteData()* untuk menghapus data.

Sequence diagram hapus data admin dapat dilihat pada gambar 3.18



Gambar 3.18 Sequence diagram hapus data admin

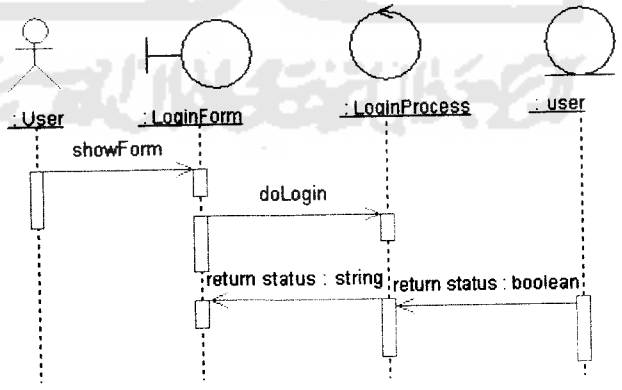
3.2.2.1.3.3.4 Login

Objek yang berkaitan dengan *sequence* ini akan di jelaskan sebagai berikut:

Aktor	: *Admin, Member*
Kelas pembatas	: *LoginFormUI*
Kelas kontrol	: *LoginProcess*
Kelas entity	: *User*
Keterangan	:

1. *User* mengawali *sequence* ini dengan memanggil method *ShowForm()*.
2. *LoginFormUI* memanggil kelas *LoginProcess* dan memanggil method *doLogin()* untuk memerintahkan agar username dan password diperiksa kebenarannya.
3. *LoginProcess* memanggil kelas *User* dan memanggil method *isValidUser()* untuk memeriksa kebenaran dari username dan password yang dimasukkan.
4. *User* menanggapi dengan mengembalikan status, tipe Boolean, ke *LoginProcess* untuk memberitahukan status bahwa username dan password yang dimasukkan benar/salah.
5. *LoginProcess* mengembalikan status, tipe String, ke FormUI untuk ditampilkan ke user bahwa berhasil/gagal login ke sistem.

Sequence diagram login dapat dilihat pada gambar 3.19



Gambar 3.19 Sequence diagram login

3.2.2.1.4 Activity Diagram

Activity diagram menggambarkan berbagai aliran aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing aliran berawal, decision yang mungkin terjadi, dan menggambarkan mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Diagram sangat mirip dengaan *flowchart* karena kita dapat memodelkan sebuah alur kerja dari satu aktivitas ke aktivitas lainnya atau dari aktivitas ke dalam keadaan sesaat *(state)*, tetapi perbedaan dengan *flowchart* adalah *activity diagram* bisa mendukung perilaku paralel sedangkan *flowchart* tidak bisa.

Membuat *activity diagram* dalam memodelkan terlebih dahulu dalam memodelkan sebuah proses dapat membantu kita memahami proses secara keseluruhan. *Activity diagram* adalah variasi dari *state* diagram yang mana *state* merepresentasikan operasi, dan transisinya merepresentasikan aktivitas yang terjadi pada saat operasi sudah selesai.

Pada bagian ini akan dijelaskan menggunakan *activity diagram* proses yang terjadi dalam sistem pengingat waktu sholat, meliputi: penambahan data, pengubahan data dan penghapusan data.

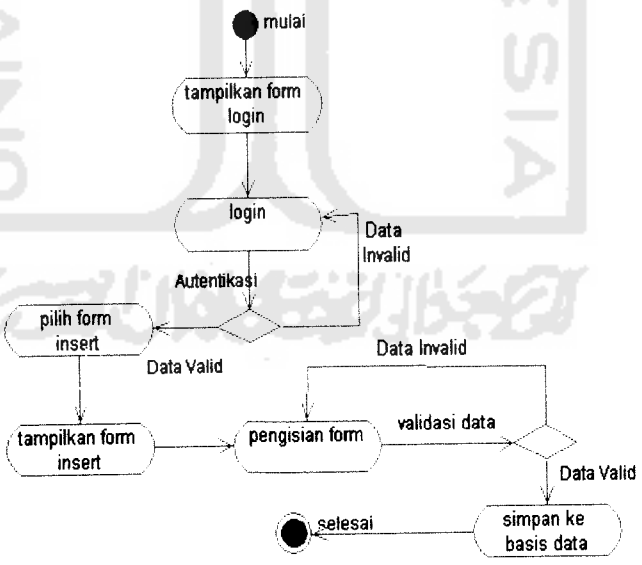
3.2.2.1.4.1 Penambahan Data

Kegiatan untuk menambahkan data baru ke dalam basis data. Untuk urutan aktivitas dijelaskan sebagai berikut:

1. User memilih menu insert.

2. Sistem menampilkan form pengisian data sebagai sarana agar user dapat menambah data yang diinginkan.
3. Kemudian user memasukkan data pada form yang telah disediakan sistem dan menekan tombol done untuk mengirimkan data.
4. Sistem melakukan pengecekan/validasi apakah data yang dimasukkan user valid atau tidak. Sesuai dengan format data yang ditetapkan sistem.
5. Jika valid data tersimpan sedangkan jika tidak valid user diperintahkan melakukan pengulangan. Kemudian data disimpan.

Activity diagram penambahan data dapat dilihat pada gambar 3.20



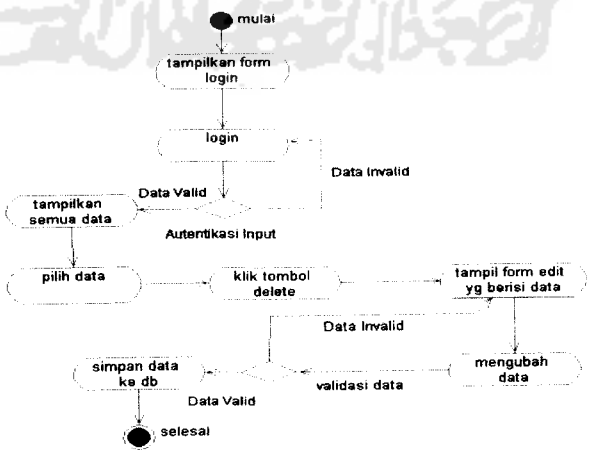
Gambar 3.20 Activity diagram panambahan data

3.2.2.1.4.2 Pengubahan Data

Pengubahan data merupakan proses pengubahan data sistem dengan urutan aktivitas sebagai berikut:

1. User memilih menu edit.
2. Sistem menampilkan semua data dari tabel tertentu.
3. User memilih data yang ingin diubah.
4. User menekan tombol edit.
5. Sistem menampilkan form edit dan isi data yang telah dipilih.
6. User mengedit data melalui form yang tampil dan pada field sudah terdapat data lalu menekan tombol done.
7. Sistem memvalidasi data, jika valid data tersimpan namun jika tidak valid muncul pesan kesalahan.
8. Sistem mengakhiri aktivitas ini.

Activity diagram proses pengubahan data dapat dilihat pada gambar 3.21



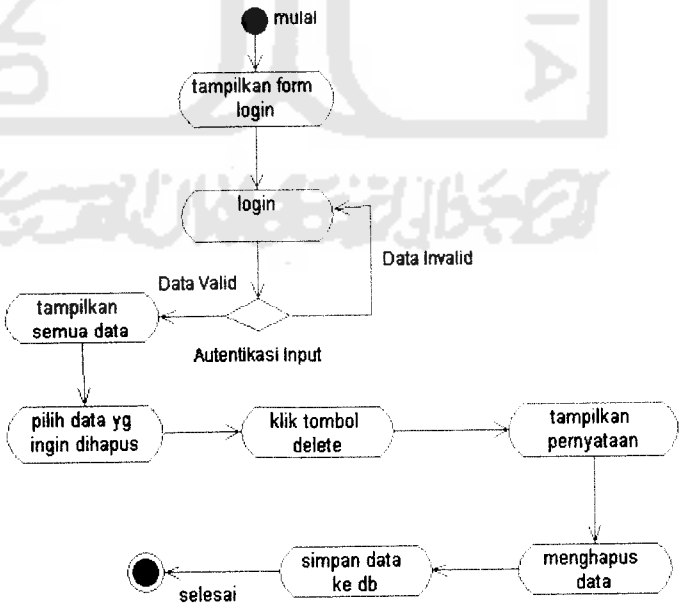
Gambar 3.21 Activity diagram pengubahan data

3.2.2.1.4.3 Penghapusan Data

Merupakan aktivitas user untuk menghapus data yang ada pada sistem. Urutan langkah-langkahnya adalah sebagai berikut:

1. *User* memilih menu *delete*.
2. Sistem menampilkan semua data dari tabel tertentu.
3. *User* memilih data yang ingin dihapus.
4. *User* menekan tombol *delete*.
5. Sistem menampilkan pernyataan apakah ingin menghapus data.
6. Sistem melakukan proses penghapusan data dalam basis data.
7. Sistem mengakhiri kegiatan ini.

Activity diagram dapat dilihat pada gambar 3.22



Gambar 3.22 Activity diagram penghapusan data

3.2.2.2 Perancangan Basis Data

Basis data adalah suatu kumpulan data-data yang saling terhubung yang dimanfaatkan untuk mengolah dan memanipulasi data menjadi informasi. Sifat basis data bertipe relasional yang terdiri dari beberapa tabel yang saling terhubung dan dapat melakukan relasi.

Setiap aplikasi memiliki data yang harus disimpan didalam tabel sehingga data tersebut dapat dikelola dengan baik. Berikut ini akan dijelaskan tabel yang digunakan dalam aplikasi pengingat sholat.

3.2.2.2.1 Struktur Tabel

1. Tabel koneksi_data

Tabel koneksi data digunakan untuk menyimpan data-data koneksi, misalnya: nama *port*, *bit per second* (*bps*), *data bits, parity, stop bits* dan *flow control.* Untuk struktur tabel koneksi_data dapat dilihat pada tabel 3.1

Tabel 3.1 Tabel koneksi_data

Nama Field	Tipe Data	Panjang	Keterangan
Nama	Varchar	50	Nama koneksi, *Primary Key* (PK)
Port	Varchar	30	Nama port yang digunakan
Bps	Integer	10	Nilai baud rate
Data_bits	Integer	3	Nilai data bits
Parity	Varchar	30	Nilai parity
Stop_bits	Double	double	Nilai stop bits
Flow_kontrol	Varchar	30	Nilai flow control

2. Tabel koneksi_log

Tabel koneksi_log digunakan untuk mencatat seluruh transaksi tentang koneksi, meliputi: waktu (*time stamp*), nama dan status. Struktur tabel dari koneksi_log dapat dilihat pada tabel 3.2

Tabel 3.2 Tabel koneksi_log

Nama Field	Tipe Data	Panjang	Keterangan
Nama	Varchar	50	Nama koneksi, *Foreign Key* (FK)
Waktu	Datetime		Nama port yang digunakan
Status	Integer	1	Nilai baud rate

3. Tabel member_data

Tabel member_data digunakan untuk menyimpan data pribadi member, seperti: id_member, Nama, Hp dan Kota. Struktur tabel dari member_data dapat dilihat pada tabel 3.3

Tabel 3.3 Tabel member_data

Nama Field	Tipe Data	Panjang	Keterangan
Id_member	Integer	7	Nomor urut member
Nama	Varchar	15	Nama member
Hp	Integer	15	Nomor *handphone* member
Kota	Varchar	15	Kota member berada

4. Tabel member_auth

Tabel member_auth digunakan untuk menyimpan data autentifikasi member. Struktur tabel member_auth dapat dilihat pada tabel 3.4

Tabel 3.4 Tabel member_auth

Nama Field	Tipe Data	Panjang	Keterangan
Id_member	Varchar	7	Nomor urut member
Password	Varchar	25	Password member
Status	Varchar	1	Status member

5. Tabel data_sholat

Tabel data_sholat digunakan untuk menyimpan data jadwal sholat. Waktu sholat ditentukan oleh letak geografis kota yang bersangkutan. Struktur tabelnya dapat dilihat pada gambar tabel 3.5

Tabel 3.5 Tabel data_sholat

Nama Field	Tipe Data	Panjang	Keterangan
Kode_kota	Varchar	7	Kode dari kota
Nama	Varchar	25	Nama kota
Latitude	Integer	15	Letak lintang kota
Longitude	Integer	15	Letak bujur kota
Timezone	Integer	15	Zona waktu kota
Ketinggian	Integer	15	Ketinggian kota dari permukaan laut
Subuh	time		Waktu sholat subuh
Terbit	Time		Waktu terbit matahari
Zuhur	Time		Waktu sholat zuhur
Asyar	Time		Waktu sholat asyar
Magrib	Time		Waktu sholat magrib
Isya	time		Waktu sholat isya

6. Tabel sms_inbox

Tabel sms_inbox digunakan untuk menyimpan pesan yang masuk ke sistem. Semua data pesan yang diterima sistem disimpan dan dikelola melalui tabel sms_inbox. Struktur tabel sms_inbox ada pada tabel 3.6

Tabel 3.6 Tabel sms_inbox

Nama Field	Tipe Data	Panjang	Keterangan
Id_inbox	Integer	11	nomor urut pesan yang masuk
No_hp	Varchar	20	Nomor *handphone* user
Pesan	Teks		Isi pesan user
Waktu	Timestamp		Waktu user kirim pesan

7. Tabel sms_outbox

Tabel sms_outbox digunakan untuk menyimpan dan mengelola pesan yang akan dikirim ke member. Struktur tabel sms_outbox bisa dilihat pada tabel 3.7

Tabel 3.7 Tabel sms_outbox

Nama Field	Tipe Data	Panjang	Keterangan
Id_inbox	Integer	11	Nomor urut pesan yang keluar
No_hp	Varchar	20	Nomor *handphone* user
Pesan	Teks		Isi pesan balasan yang dikirim sistem ke user
Waktu	Timestamp		Waktu sistem kirim pesan balasan
Status	Varchar	2	Status SMS

8. Tabel admin_data

Tabel admin_data digunakan untuk menyimpan data pribadi admin sebagai pengelola sistem. Struktur tabel admin_data bisa dilihat pada tabel 3.8

Tabel 3.8 Tabel admin_data

Nama Field	Tipe Data	Panjang	Keterangan
User_id	Varchar	255	Nomor urut admin
Nama	Varchar	50	Nama admin
Alamat	Varchar	100	Alamat admin
Telp.	Varchar	20	Nomor telepon admin

9. Tabel admin_auth

Tabel admin_auth digunakan untuk autentifikasi data admin. Struktur tabel admin_auth bisa dilihat pada tabel 3.9

Tabel 3.9 Tabel admin_auth

Nama Field	Tipe Data	Panjang	Keterangan
User_id	Varchar	255	Nomor urut admin
Password	Varchar	255	Password admin
Privilege	Varchar	1	Hak akses admin

10. Tabel reminder

Tabel reminder digunakan untuk mencatat pengiriman sms dari sistem ke member. Struktur tabelnya dapat dilihat di tabel 3.10

Tabel 3.10 Tabel reminder

Nama Field	Tipe Data	Panjang	Keterangan
id_member	Varchar	7	Nomor urut member
Subuh	Varchar	1	Sholat subuh
Zuhur	Varchar	1	Sholat zuhur
Asyar	Varchar	1	Sholat asyar
Magrib	Varchar	1	Sholat magrib
Isya	Varchar	1	Sholat isya
Today	date		Waktu hari ini

3.2.2.2.2 Relasi Antar Tabel

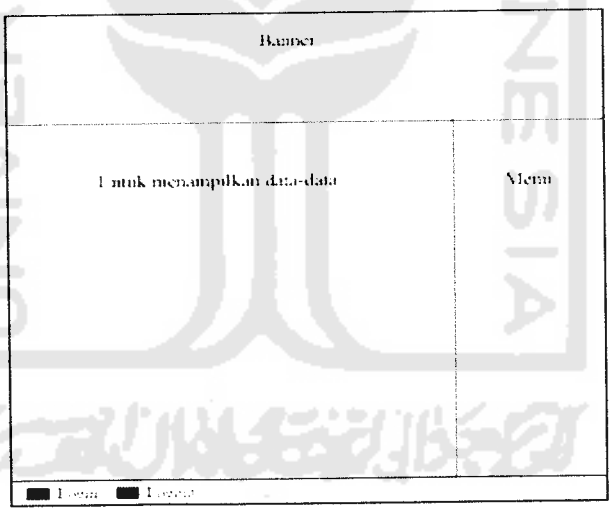
Basis data merupakan kumpulan dari data-data yang saling terhubung antara satu dengan yang lainnya. Data tersebut disimpan dalam sebuah file atau tabel dengan keterangan tertentu, dan apabila direlasikan akan menghasilkan informasi yang kompleks. Hubungan antar tabel ini dihasilkan dengan *relation key* (kunci relasi)

3.2.2.3 Halaman Antarmuka

Halaman antarmuka untuk sistem utama atau halaman *admin* dari aplikasi pengingat waktu sholat menggunakan NETBEANS IDE 6.0.

3.2.2.3.1 Halaman Utama

Pada sistem pengingat waktu sholat dalam pembuatan tampilannya terbagi dalam beberapa bagaian, bagian atas sebagai peletakkan *banner*, kolom kanan sebagai bagian menu dan bagian kiri sebagai tempat untuk menampilkan data untuk diolah lebih lanjut. Untuk halaman utama dapat dilihat pada gambar 3.24



Gambar 3.24 Rancangan halaman utama

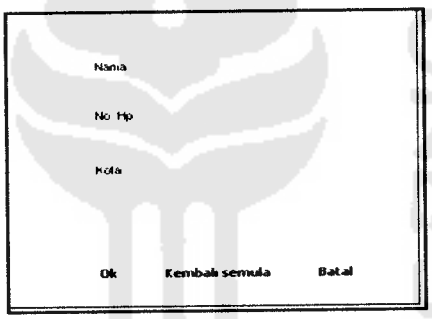
3.2.2.3.2 Halaman Input

Input (masukan) merupakan awal dimulainya proses informasi. Masukan untuk sistem informasi adalah data yang merupakan bahan mentah dari informasi. Halaman *input* atau masukan adalah halaman *form* yang digunakan untuk menangkap

data (*input*), kode-kode *input* yang digunakan dan bentuk dari tampilan *input*. Adapun halaman *input* dialog untuk sistem ini adalah sebagai berikut:

3.2.2.3.2.1 Halaman Form Input Data Member

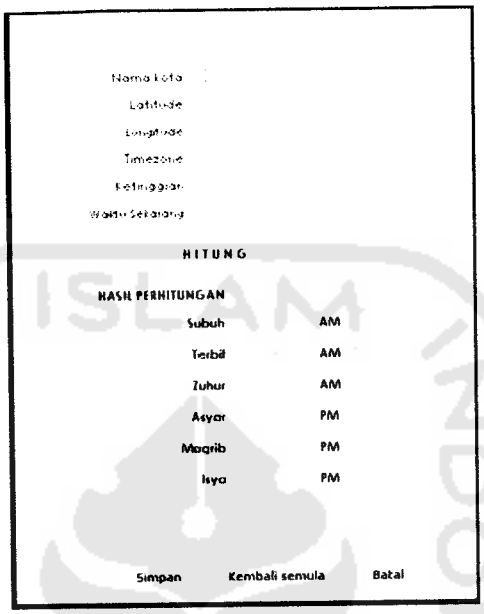
Halaman antarmuka digunakan untuk mempermudah dalam pemasukan, perubahan, penghapusan dan penyimpanan data. Data masukan member terdiri dari id member, nama dan nomor *handphone*. Halaman *input* data member dapat dilihat pada gambar 3.25



Gambar 3.25 Rancangan halaman input data member

3.2.2.3.2.2 Halaman Form Input Data Waktu Sholat

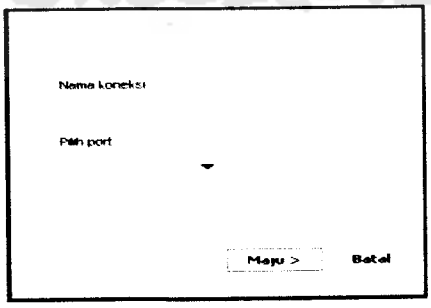
Halaman antarmuka pemasukkan data waktu sholat digunakan untuk memasukkan waktu sholat, karena didalam form sholat sudah terdapat form untuk menghitung waktu sholat berdasarkan letak geografis kota. Data masukkan kota terdiri dari nama kota, latitude, longitude, timezone, ketinggian dan hasilnya berupa data waktu sholat Subuh, waktu terbit matahari, waktu sholat Zuhur, waktu sholat Asyar, waktu sholat Magrib dan waktu sholat Isya. Hasilnya berbentuk jam, menit dan detik sesuai format jam. Halaman input data waktu sholat dapat dilihat pada gambar 3.26



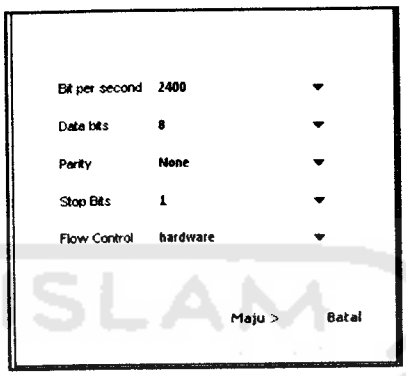
Gambar 3.26 Rancangan halaman input data waktu sholat

3.2.2.3.2.3 Halaman Form Input Data Modem

Halaman form input data modem digunakan untuk memasukkan data modem. Data masukkan modem terdiri dari nama koneksi, port, bit per second, data bits, parity, stop bits, flow control. Halaman input data modem dapat dilihat pada gambar 3.27 dan 3.28



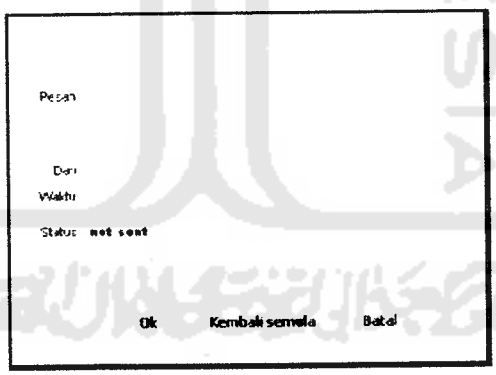
Gambar 3.27 Rancangan halaman input data modem 1



Gambar 3.28 Rancangan halaman input data modem 2

3.2.2.3.2.4 Halaman Form Input Data SMS

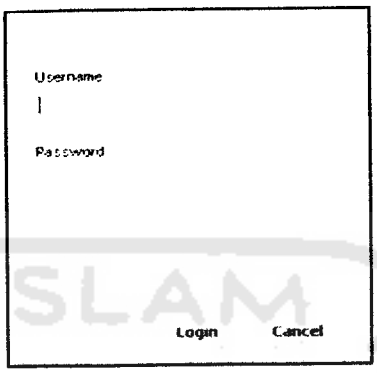
Halaman ini digunakan untuk pemasukan data pesan yang masuk ke sistem. Data masukkan terdiri dari isi pesan, waktu pengiriman dan nomor *handphone* pengirim. Halaman Ubah Input Data SMS Inbox dapat dilihat pada gambar 3.29



Gambar 3.29 Rancangan halaman ubah data sms

3.2.2.3.2.5 Halaman Form Login

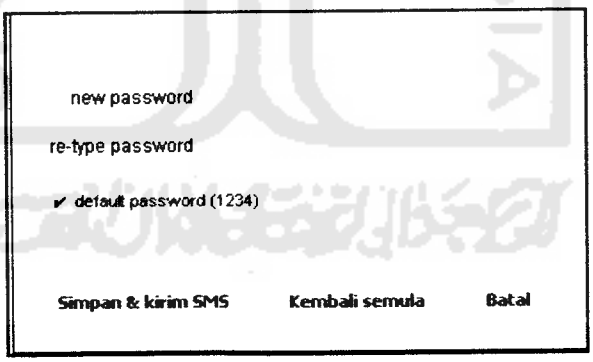
Halaman login digunakan sebagai antarmuka admin untuk login ke sistem. Sebelum masuk ke sistem admin diwajibkan login terlebih dahulu dengan cara memasukkan *username* dan *password.* Halaman form login dapat dilihat pada gambar 3.30



Gambar 3.30 Rancangan halaman login admin

3.2.2.3.2.6 Halaman Form Ubah Password

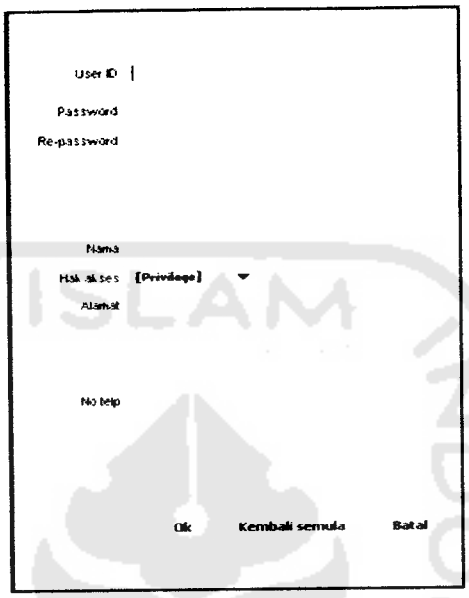
Antarmuka ubah password digunakan untuk mengubah password *user (admin)*. Terdapat tiga *input* pada antarmuka ini, antara lain: password lama, password baru dan *re-type* password. Adapun untuk halaman antarmuka ubah password dapat dilihat pada gambar 3.31



Gambar 3.31 Rancangan halaman ubah password

3.2.2.3.2.7 Halaman Form Input Data Admin

Antarmuka input *user* digunakan untuk membuat *user* baru. Pada antarmuka ini terdapat beberapa *input* antara lain: *user* id, password, nama, *privilege,* alamat dan nomor telepon. Untuk halaman antarmuka input *user* dapat dilihat pada gambar 3.32



Gambar 3.32 Rancangan halaman input data admin

3.2.2.4 Format SMS

Format SMS digunakan untuk menetapkan pengiriman pesan sebagai *input* yang dikirim user ke sistem. Sistem menetapkan format baku yang harus dipatuhi *user* sehingga apapun *input* dari *user* yang tidak sesuai format akan ditolak. Tujuan dari ditetapkannya format pengiriman SMS, yaitu agar *input* yang dikirimkan oleh *user* dapat dipastikan ke-*valid*-annya sebelum disimpan ke dalam basis data. Halaman *input* dialog untuk sistem ini adalah sebagai berikut:

3.2.2.4.1 Format Bantuan

Format bantuan dirancang sebagai panduan buat user dalam menggunakan aplikasi pengingat sholat. Format ini untuk menampilkan bantuan penggunaan perintah-perintah SMS yang terdapat pada sistem. Format yang telah ditetapkan sistem seperti pendaftaran, ubah password, ubah lokasi dan lihat data kota

secara keseluruhan. Adapun untuk format penulisan dari masing-masing perintah adalah sebagai berikut:

- Menampilkan semua perintah : HELP
- Pendaftaran : REG (Registrasi)
- Ubah password : UP (Ubah Password)
- Ubah lokasi : UK (Ubah Kota)
- Lihat Kota : LK (Lihat Kota)

3.2.2.4.2 Format Pendaftaran

Format ini digunakan untuk pendaftaran user menjadi member sistem pengingat waktu sholat. Setelah user memperoleh nomor id member. Pengetikan format ini yaitu "REG Nama Kota".

3.2.2.4.3 Format Ubah Password

Format ini digunakan untuk mengubah password member, terdapat beberapa input yang harus dikirimkan, yaitu password lama dan password baru. Adapun untuk format ubah password adalah " UP <Password Lama> <Password Baru>".

3.2.2.4.4 Format Ubah Kota

Format ini digunakan jika *member* pindah lokasi dari satu tempat ketempat lainnya. Lokasi yang dimaksud yaitu kota dimana *member* berada, jika *member* berpindah lokasi dari satu kota ke kota lainnya format difungsikan untuk

mengantisipasi hal itu. Format ubah password yaitu "UK <password> <kota anda berada sekarang>".

3.2.2.4.5 Format Lihat Kota

Format ini digunakan jika *user* ingin melihat data kota yang ada pada sistem. Format lihat kota yaitu " LK ".