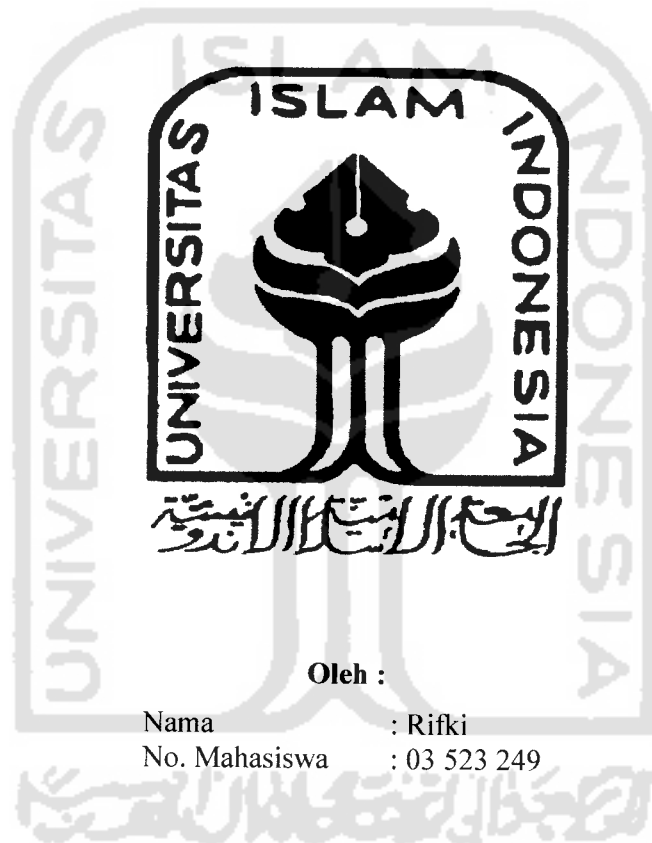


**SISTEM PENGINGAT UNTUK TINDAKAN TERAPI *INSULIN*
BAGI PENDERITA *DIABETES MELLITUS*
BEBASIS SMS**

TUGAS AKHIR

**Diajukan sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana
Jurusan Teknik Informatika**



Oleh :

Nama : Rifki

No. Mahasiswa : 03 523 249

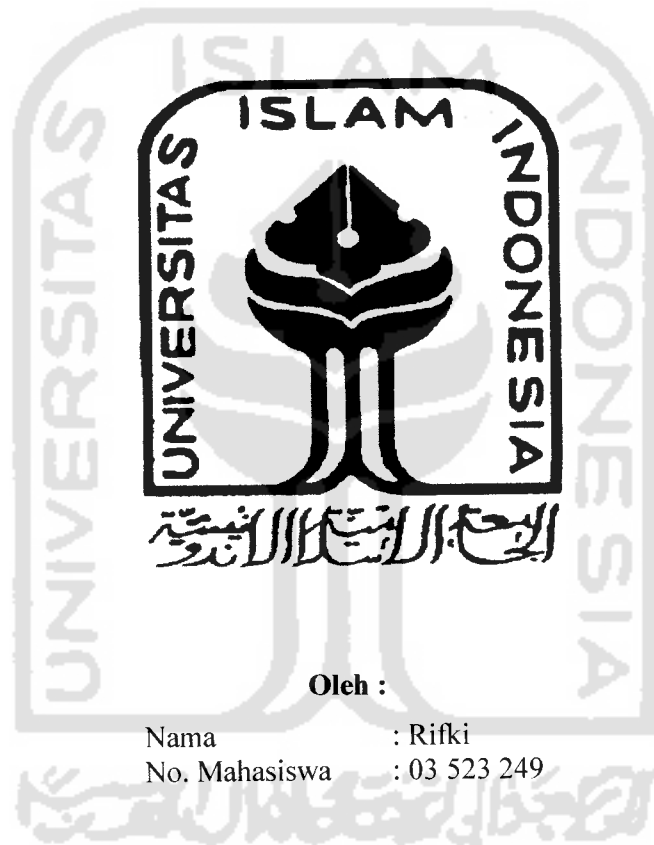
**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA**

2007

**SISTEM PENGINGAT UNTUK TINDAKAN TERAPI *INSULIN*
BAGI PENDERITA *DIABETES MELLITUS*
BEBASIS SMS**

TUGAS AKHIR

**Diajukan sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana
Jurusan Teknik Informatika**



Oleh :

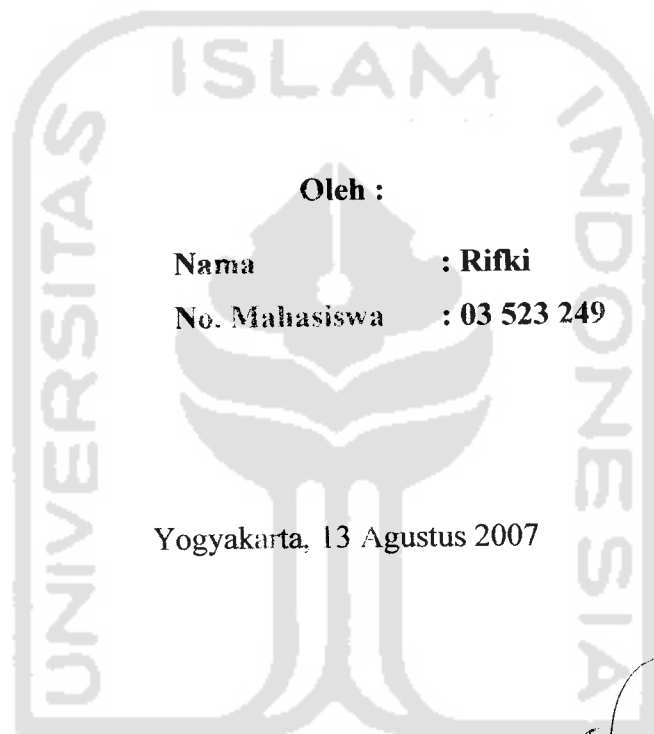
Nama : Rifki
No. Mahasiswa : 03 523 249

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA
2007**

LEMBAR PENGESAHAN PEMBIMBING

**SISTEM PENGINGAT UNTUK TINDAKAN TERAPI *INSULIN*
BAGI PENDERITA *DIABATES MELLITUS*
BERBASIS SMS**

TUGAS AKHIR



Oleh :

Nama : Rifki

No. Mahasiswa : 03 523 249

Yogyakarta, 13 Agustus 2007

Pembimbing,

(Sri Kusumadewi, S.Si., MT)

HALAMAN PERSEMBAHAN

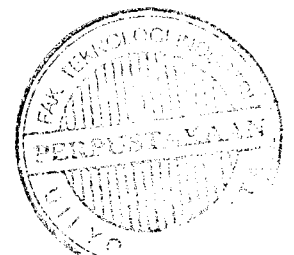
Alhamdulillahilahi Rabbil 'Aalamiin
Puji dan syukur kami panjatkan kehadiran Allah SWT
yang telah memberikan rahmat dan hidayah-Nya serta
memberikan kemudahan sehingga dapat menyelesaikan
tugah akhir ini dengan baik

Dengan izin-Nya pula,
kupersembahkan karyaku ini untuk:

Kedua orang tuaku dan kakak-ku
Terima kasih untuk doa, kasih sayang dan nasehatnya

Saudara-saudaraku
Nyonk, geriting, pank ndut, ivan, adin, faisal, yus
dan antonio kodir

Terima kasih atas kasih sayang, curahan perhatian
dan kesabarannya selama ini. Semoga Allah SWT
membalasnya dengan yang lebih baik



Temen-temen seperjuangan informatika '03
Nico, why, ade, bayu, kenthus, fajar, rian, medi
Dan juga semua temen2 informatika '03.
Terima kasih banget untuk pertemanan dan perjuangan yang
indah selama ini



HALAMAN MOTTO

“Sesungguhnya sesudah kesulitan itu ada kemudahan”

(QS. Al Insyroh : 6)

”.....jadikanlah sabar dan shalat sebagai penolongmu”

(QS. Al Baqoroh :153)

”sesungguhnya Allah tidak mengubah keadaan suatu kaum sehingga mereka
mengubah keadaan yang ada pada diri mereka sendiri”

(QS. Ar Ra'd :11)

”Kreatifitas, kegigihan dan keuletan dalam melakukan percepatan diri adalah kunci
sukses karya besar orang-orang biasa dalam meluarbiasakan dirinya”

KATA PENGANTAR



Assalamu'alaikum Wr.Wb.

Dengan memanjatkan puji dan syukur kehadiran Allah SWT atas segala rahmat dan hidayah-Nya, sehingga penulisan laporan tugas akhir ini dapat terselesaikan dengan baik dan lancar. Sholawat serta salam senantiasa kita sampaikan kepada junjungan kita Nabi besar Muhammad SAW, beserta para keluarga, sahabat, dan pengikutnya sampai akhir zaman.

Laporan tugas akhir ini adalah salah satu syarat guna memperoleh gelar kesarjanaan Strata 1 (S1) pada jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia.

Selama dalam pelaksanaan tugas akhir dan penulisan laporan ini, penulis telah mendapat bimbingan dan bantuan dari berbagai pihak. Sehingga penulis merasa perlu untuk memberikan ucapan terima kasih yang sebesar-besarnya kepada :

1. Pimpinan Rektor Universitas Islam Indonesia.
2. Bapak Fathul Wahid, ST., M.Sc selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia.
3. Bapak Yudi Prayudi, S.Si., M.Kom selaku Ketua Jurusan Teknik Informatika Universitas Islam Indonesia.

4. Ibu Sri Kusumadewi, S.Si., MT selaku dosen pembimbing yang telah memberikan pengarahan dan bimbingan selama pengerjaan tugas akhir dan penulisan laporan ini.
5. Ibu Suci Hanifah, S.Farm., Apt yang telah membimbing penulis untuk mengetahui dan mempelajari *diabetes mellitus* sebagai landasan teori.
6. Seluruh dosen yang telah memberikan ilmu kepada penulis dan membimbing penulis untuk mengetahui dan mempelajari ilmu informatika
7. Kedua orang tuaku, terima kasih atas doa, dukungan, bimbingan, kepercayaan, kesabaran serta nasehat yang tidak ternilai harganya.
8. Teman-teman yang selalu membantu, mendo'akan dan memberikan dukungan kepada kami.
9. Serta berbagai pihak yang tidak mungkin kami sebutkan satu persatu.

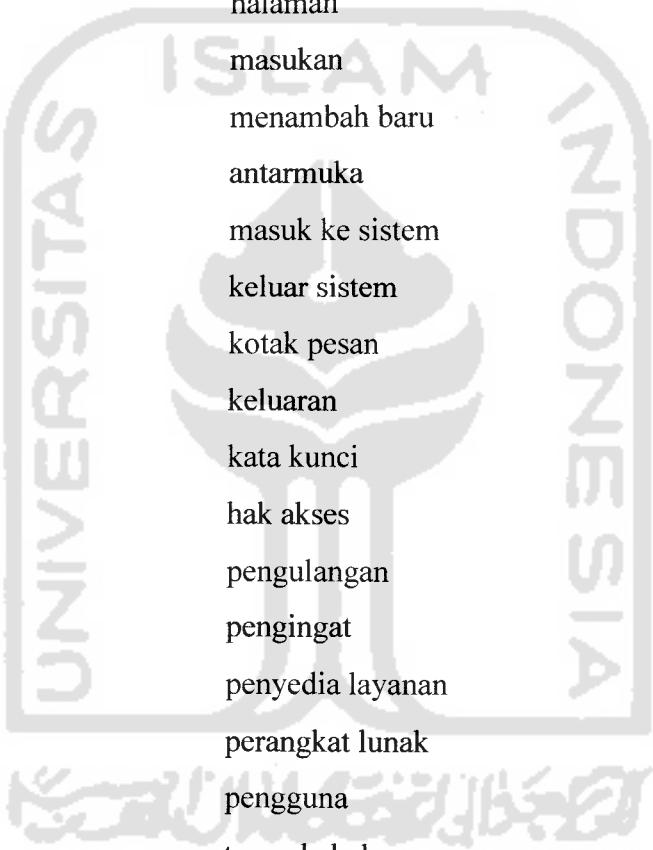
Segegap daya upaya telah kami curahkan untuk menyelesaikan laporan tugas akhir ini. Penulis menyadari sepenuhnya bahwa laporan ini masih jauh dari kesempurnaan. Sehingga untuk perbaikan laporan ini dan kebenaran ilmiahnya maka penyusun sangat mengharapkan adanya masukan, kritik, serta saran dari para pembaca. Semoga laporan tugas akhir ini dapat memberikan manfaat. Amien.

Wassalamu'alaikum Wr.Wb..

Yogyakarta, Agustus 2007

Penyusun

TAKARIR



<i>admin</i>	administrator
<i>edit</i>	merubah
<i>delete</i>	menghapus
<i>field</i>	kolom
<i>form</i>	halaman
<i>input</i>	masukan
<i>insert</i>	menambah baru
<i>interface</i>	antarmuka
<i>login</i>	masuk ke sistem
<i>logout</i>	keluar sistem
<i>message box</i>	kotak pesan
<i>output</i>	keluaran
<i>password</i>	kata kunci
<i>privilege</i>	hak akses
<i>redundancy</i>	pengulangan
<i>reminder</i>	pengingat
<i>server</i>	penyedia layanan
<i>software</i>	perangkat lunak
<i>user</i>	pengguna
<i>wireless</i>	tanpa kabel

ABSTRAKSI

Teknologi Komunikasi berkembang sangat pesat seiring dengan semakin banyaknya permintaan akan teknologi tersebut. Salah satu teknologi yang telah menjadi penunjang aktivitas kehidupan harian masyarakat adalah *Global System for Mobile Communication (GSM)* atau yang lebih dikenal dengan telepon selular digital. *Short Messaging Service (SMS)* merupakan salah satu layanan dari GSM.

SMS merupakan salah satu media yang banyak digunakan oleh masyarakat sekarang ini, karena SMS hadir dengan tarif yang sangat murah dibanding dengan layanan panggilan. Selain untuk mengirim pesan (komunikasi), SMS juga baik digunakan untuk transfer data atau informasi dalam kapasitas kecil. Misalnya, aplikasi pengingat terapi *insulin* berbasis SMS pada penderita *diabetes mellitus*.

Fungsi dari aplikasi ini, yaitu sebagai pengingat bagi para pasien penderita *diabetes mellitus* agar lebih teratur dalam melakukan terapi *insulin*. Sebab dengan semakin banyaknya aktivitas yang dilakukan, akan lebih sering mengakibatkan terjadinya kelalian terhadap sesuatu hal. Padahal dalam terapi *insulin* diperlukan adanya keteraturan dan konsistensi, jika tidak maka akan dapat memberlambat proses penyembuhan. Terlebih lagi jika hal tersebut dilakukan secara berulang kali, maka akan dapat mengakibatkan semakin buruknya kondisi pasien.

Kata kunci : SMS gateway, *diabetes mellitus*, *insulin*

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN PEMBIMBING	ii
LEMBAR PERNYATAAN KEASLIAN	iii
LEMBAR PENGESAHAN PENGUJI	iv
LEMBAR PERSEMBAHAN	v
HALAMAN MOTTO	vii
KATA PENGANTAR	viii
ABSTRAKSI	x
TAKARIR	xi
DAFTAR ISI	xii
DAFTAR GAMBAR	xvi
DAFTAR TABEL	xx
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3

1.6	Metodologi Penelitian	3
1.6.1	Metode Pengumpulan Data	3
1.6.2	Metode Pengembangan Sistem	4
1.7	Sistematika Penulisan	4
BAB II LANDASAN TEORI		7
2.1	SMS (<i>Short Message Service</i>)	7
2.1.1	Pengertian SMS	7
2.1.2	Mekanisme Dan Cara Kerja SMS	9
2.2	<i>Mobile</i> atau <i>Wireless</i>	10
2.2.1	Perkembangan Teknologi Jaringan <i>Wireless</i>	10
2.2.2	<i>GSM Cable Data</i>	12
2.3	<i>Access And Terminals (AT) Commaand</i>	13
2.4	Basis Data	14
2.4.1	Obyektif Basis Data	15
2.4.2	Sistem Basis Data	16
2.4.3	Komponen Basis Data	16
2.4.4	Perancangan Basis Data	17
BAB III METODOLOGI		18
3.1	Analisis Kebutuhan Perangkat Lunak	18
3.1.1	Metode Analisis	18
3.1.2	Hasil Analisis	18
3.1.2.1	Analisis Kebutuhan Proses	19
3.1.2.2	Analisis Kebutuhan <i>Input</i>	19

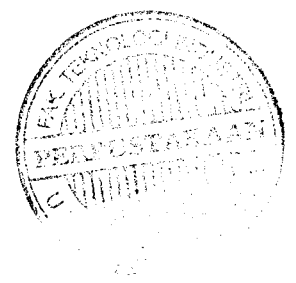
3.1.2.3	Analisis Kebutuhan <i>Output</i>	20
3.1.3	Kebutuhan Antarmuka	20
3.1.4	Analisis Kebutuhan Perangkat Lunak	21
3.1.5	Analisis Kebutuhan Perangkat Keras	21
3.2	Perancangan Perangkat Lunak	22
3.2.1	Metode Perancangan	22
3.2.2	Hasil Perancangan	22
3.2.2.1	Perancangan UML (<i>Unified Modeling Language</i>) ...	23
3.2.2.1.1	<i>Use Case Diagram</i>	23
3.2.2.1.2	<i>Class Diagram</i>	25
3.2.2.1.3	<i>Sequence Diagram</i>	28
3.2.2.1.4	<i>Activity Diagram</i>	46
3.2.2.2	Perancangan Basis Data	50
3.2.2.2.1	Struktur Tabel	50
3.2.2.2.2	Relasi Antar Tabel	55
3.2.2.3	Perancangan Antarmuka	56
3.2.2.3.1	Perancangan Halaman Utama	57
3.2.2.3.2	Perancangan Input	57
3.2.2.4	Perancangan Format SMS	64
3.3	Implementasi Perangkat Lunak	66
3.3.1	Batasan Implementasi	67
3.3.2	Implementasi Antarmuka	67
3.3.2.1	Halaman Utama	67

3.3.2.2	Form Login <i>Admin</i>	68
3.3.2.3	Form Input Data Pasien	69
3.3.2.4	Form Input Data Rekam Medis	69
3.3.2.5	Form Input Data Obat	70
3.3.2.6	Form Input Data User	71
3.3.2.7	Halaman Ubah Password	71
3.3.2.8	Form Ubah Pesan	72
3.3.2.9	Form Input Data Koneksi	73
3.3.2.10	Form Pencarian Data	74
BAB IV HASIL DAN PEMBAHASAN		75
4.1	Pengujian Program	75
4.2	Analisis Kinerja Sistem	75
4.2.1	Penanganan Kesalahan	75
4.2.1.1	Penanganan Kesalahan Pada <i>Admin</i>	76
4.2.1.2	Penangan Kesalahan Pada <i>User</i>	80
4.2.2	Pengujian Dan Analisis	83
4.2.2.1	Pengujian Pada <i>Admin</i>	83
4.2.2.2	Pengujian Pada <i>User</i>	88
BAB V SIMPULAN DAN SARAN		93
5.1	Simpulan	93
5.2	Saran	94
DAFTAR PUSTAKA		95

DAFTAR GAMBAR

Gambar 2.1 <i>short message mobile terminated</i>	8
Gambar 2.2 <i>short message mobile oriented</i>	8
Gambar 2.3 mekanisme cara kerja sms	10
Gambar 3.1 <i>use case diagram</i>	24
Gambar 3.2 <i>class diagram modul data management</i>	26
Gambar 3.3 <i>class diagram modul system management</i>	27
Gambar 3.4 <i>class diagram modul user management</i>	28
Gambar 3.5 <i>sequence diagram</i> tambah pasien	30
Gambar 3.6 <i>sequence diagram</i> ubah pasien	31
Gambar 3.7 <i>sequence diagram</i> hapus pasien	32
Gambar 3.8 <i>sequence diagram</i> tambah data rekam medis	32
Gambar 3.9 <i>sequence diagram</i> ubah data rekam medis	33
Gambar 3.10 <i>sequence diagram</i> hapus data rekam medis	34
Gambar 3.11 <i>sequence diagram</i> tambah data obat	35
Gambar 3.12 <i>sequence diagram</i> ubah data obat	36
Gambar 3.13 <i>sequence diagram</i> hapus data obat	37
Gambar 3.14 <i>sequence diagram</i> tambah data koneksi modem	38
Gambar 3.15 <i>sequence diagram</i> ubah data koneksi modem	39
Gambar 3.16 <i>sequence diagram</i> hapus data koneksi modem	40

Gambar 3.17 <i>sequence diagram</i> ubah data SMS	41
Gambar 3.17 <i>sequence diagram</i> hapus data SMS	42
Gambar 3.18 <i>sequence diagram</i> tambah user	43
Gambar 3.20 <i>sequence diagram</i> ubah user	44
Gambar 3.21 <i>sequence diagram</i> hapus user	45
Gambar 3.22 <i>sequence diagram</i> login	46
Gambar 3.23 <i>activity diagram</i> penambahan data	47
Gambar 3.24 <i>activity diagram</i> pengubahan data	48
Gambar 3.25 <i>activity diagram</i> penghapusan data	49
Gambar 3.26 relasi tabel basis data	56
Gambar 3.27 rancangan halaman depan	57
Gambar 3.28 rancangan input data pasien	58
Gambar 3.29 rancangan input rekam medis	59
Gambar 3.30 rancangan input obat	60
Gambar 3.31 rancangan input modem panel pertama	60
Gambar 3.32 rancangan input modem panel kedua	61
Gambar 3.33 rancangan input user	61
Gambar 3.34 rancangan ubah pesan masuk	62
Gambar 3.35 rancangan ubah pesan keluar	63
Gambar 3.36 rancangan input ubah password	63
Gambar 3.37 rancangan input pencarian	64



Gambar 3.38 tampilan halaman utama	68
Gambar 3.39 tampilan login <i>admin</i>	68
Gambar 3.40 tampilan <i>form</i> input data pasien	69
Gambar 3.41 tampilan <i>form input</i> data rekam medis	70
Gambar 3.42 tampilan <i>form input</i> data obat	70
Gambar 3.43 tampilan <i>form input</i> data <i>user</i>	71
Gambar 3.44 tampilan <i>form</i> ubah password	72
Gambar 3.45 tampilan <i>form</i> ubah pesan	72
Gambar 3.46 tampilan <i>form input</i> modem panel pertama	73
Gambar 3.47 tampilan <i>form input</i> modem panel kedua	73
Gambar 3.48 tampilan <i>form</i> pencarian data	74
Gambar 4.1 kotak dialog jika <i>password</i> salah	76
Gambar 4.2 kotak dialog jika <i>field</i> nama koneksi kosong	77
Gambar 4.3 kotak dialog jika <i>field</i> nama diisi dengan angka	77
Gambar 4.4 kotak dialog jika no hp terjadi duplikasi	78
Gambar 4.5 kotak dialog jika input terlalu panjang	78
Gambar 4.6 kotak dialog ubah data modem dalam keadaan aktif	79
Gambar 4.7 kotak dialog aktivasi modem dalam keadaan aktif	79
Gambar 4.8 kotak dialog jika tabel tidak terpilih	80
Gambar 4.9 kotak dialog jika data tidak ditemukan	80
Gambar 4.10 antarmuka masukan login	84

Gambar 4.11 antarmuka masukan data pasien	85
Gambar 4.12 antarmuka masukan data rekam medis	86
Gambar 4.13 antarmuka masukan data obat	86
Gambar 4.14 antarmuka masukan data <i>user</i>	87
Gambar 4.15 antarmuka masukan data modem	88



DAFTAR TABEL

Tabel 2.1 Tabel AT <i>Command</i>	14
Tabel 3.1 Tabel koneksi_data	50
Tabel 3.2 Tabel koneksi_log	51
Tabel 3.3 Tabel pasien_data	51
Tabel 3.4 Tabel pasien_terapi	51
Tabel 3.5 Tabel pasien_auth	52
Tabel 3.6 Tabel reminder_log	52
Tabel 3.7 Tabel rekam_medis	53
Tabel 3.8 Tabel obat	53
Tabel 3.9 Tabel inbox	53
Tabel 3.10 Tabel outbox	54
Tabel 3.11 Tabel admin_data	54
Tabel 3.12 Tabel admin_auth	55
Tabel 3.13 Tabel time_out	55
Tabel 4.1 Tabel <i>sleding scale</i> dosis	91

**LEMBAR PERNYATAAN KEASLIAN
HASIL TUGAS AKHIR**

Saya yang bertandatangan di bawah ini,

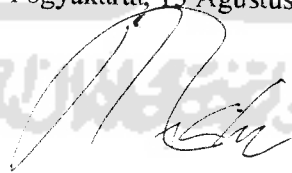
Nama : Rifki

No. Mahasiswa : 03 523 249

Menyatakan bahwa seluruh komponen dan isi dalam Laporan Tugas Akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti bahwa ada beberapa bagian dari karya ini adalah bukan hasil karya saya sendiri, maka saya siap menanggung resiko dan konsekuensi apapun.

Demikian pernyataan ini saya buat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 13 Agustus 2007



Rifki

LEMBAR PENGESAHAN PENGUJI

**SISTEM PENGINGAT UNTUK TINDAKAN TERAPI *INSULIN*
BAGI PENDERITA *DIABETES MELLITUS*
BEBASIS SMS**

TUGAS AKHIR

Oleh :

Nama : Rifki

No. Mahasiswa : 03 523 249

Telah Dipertahankan di Depan Sidang Penguji Sebagai Salah Satu Syarat
untuk Memperoleh Gelar Sarjana Jurusan Teknik Informatika
Fakultas Teknologi Industri Universitas Islam Indonesia

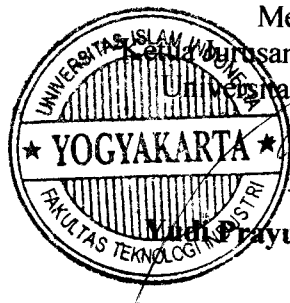
Yogyakarta, 20 Agustus 2007

Tim Penguji

Sri Kusumadewi, S.Si., MT
Ketua

Yudi Prayudi, S.Si., M.Kom
Anggota I

Syarif Hidayat, S.Kom
Anggota II



Mengetahui,
Jurusan Teknik Informatika
Universitas Islam Indonesia

Yudi Prayudi, S.Si., M.Kom

BAB I

PENDAHULUAN

1.1 Latar Belakang

Teknologi Komunikasi berkembang sangat pesat, seiring dengan semakin banyaknya permintaan akan teknologi tersebut. Salah satu teknologi komunikasi yang telah menjadi penunjang aktivitas kehidupan harian masyarakat adalah *Global System for Mobile Communication* (GSM) atau yang lebih dikenal dengan telepon selular digital. *Short Messaging Service* (SMS) merupakan salah satu layanan dari GSM. SMS merupakan salah satu media komunikasi yang banyak digunakan oleh masyarakat sekarang ini, karena hadir dengan tarif yang relatif murah. Selain untuk mengirim pesan (komunikasi), juga dapat digunakan untuk melakukan transfer data atau informasi dengan kapasitas kecil.

Misalnya, aplikasi pengingat terapi *insulin* berbasis SMS pada penderita *diabetes mellitus*. Fungsi dari aplikasi ini, yaitu sebagai alat pengingat bagi para pasien penderita *diabetes mellitus* agar lebih teratur dalam melakukan terapi *insulin*. Sebab dengan semakin banyaknya aktivitas yang dilakukan, akan lebih sering mengakibatkan terjadinya kelalian terhadap suatu hal. Padahal dalam terapi *insulin* diperlukan adanya keteraturan dan konsistensi, jika tidak maka akan dapat memperlambat proses penyembuhan. Terlebih lagi jika hal tersebut dilakukan secara berulang kali, maka akan dapat mengakibatkan semakin buruknya kondisi pasien.

Menyadari betapa pentingnya konsistensi dan keteraturan dalam melakukan terapi *diabetes mellitus*, maka sangatlah perlu dirancang aplikasi yang berfungsi sebagai pengingat. Aplikasi pengingat ini memanfaatkan teknologi yang telah mapan keberadaannya dan sering digunakan oleh masyarakat, yaitu SMS.

1.2 Rumusan Masalah

Melihat latar belakang permasalahan yang ada, maka masalah yang dapat dirumuskan, yaitu bagaimana merancang dan membuat sebuah sistem yang dapat membantu keteraturan dalam menjalankan terapi *insulin* bagi penderita *diabetes mellitus*.

1.3 Batasan Masalah

Dalam melakukan penelitian sangat diperlukan adanya batasan-batasan. Adapun batasan-batasan yang diperlukan, yaitu :

1. Sistem yang akan dibangun merupakan sistem yang berfungsi sebagai pengingat terapi *insulin* bagi para penderita *diabetes mellitus* berbasis SMS
2. *Output* (keluaran) sistem adalah pesan pengingat untuk melakukan terapi *insulin*.
3. Selain sebagai sistem pengingat, sistem ini juga dapat difungsikan sebagai sistem informasi untuk mengolah data-data riwayat kesehatan pasien.

1.4 Tujuan Penelitian

Tujuan penelitian ini adalah membangun aplikasi perangkat lunak yang berfungsi sebagai pengingat terapi *insulin* bagi para penderita *diabetes mellitus* berbasis SMS.

1.5 Manfaat Penelitian

Sistem ini diharapkan akan dapat dimanfaatkan untuk :

1. Membantu para pasien penderita *diabetes mellitus* agar lebih teratur dalam menjalankan terapi *insulin*.
2. Membantu para medis agar lebih mudah memantau kesehatan para pasiennya.
3. Memberikan informasi tentang data-data akurat tentang riwayat kesehatan pasien.

1.6 Metodologi Penelitian

Metodologi yang digunakan dalam penelitian ini meliputi metode pengumpulan data dan pengembangan sistem.

1.6.1 Metode pengumpulan data

Pengumpulan data yang diperlukan menggunakan metode sebagai berikut:

1. Observasi

Metode pengumpulan data ini digunakan untuk mendapatkan data yang berkaitan dengan kebutuhan sistem, untuk menentukan *input* (masukan) serta *output* yang efektif.

2. Studi Pustaka

Metode ini digunakan untuk mendapatkan informasi tambahan yang digunakan sebagai acuan dalam pembangunan sistem.

1.6.2 Metode pengembangan sistem

Metode pengembangan sistem yang digunakan meliputi analisis kebutuhan perangkat lunak, perancangan perangkat lunak, implementasi perangkat lunak dan analisis kinerja perangkat lunak

1.7 Sistematika Penulisan

Dalam penyusunan tugas akhir ini, sistematika penulisan dibagi menjadi beberapa bab sebagai berikut:

BAB I PENDAHULUAN

Bab ini berisi pembahasan masalah umum yang meliputi latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian dan sistematika penulisan

BAB II LANDASAN TEORI

Berisi tinjauan pustaka dan teori dasar berupa penjelasan secara terperinci mengenai teori-teori yang terkait dengan bagaimana membangun sistem informasi berbasis SMS (*SMS Gateway*) dan teknologi bahasa pemrograman *Java* yang digunakan sebagai landasan untuk pemecahan masalah. Memberikan garis besar metode yang digunakan oleh peneliti sebagai kerangka pemecahan masalah.

BAB III METODOLOGI

Bagian ini memuat uraian tentang metode analisis kebutuhan perangkat lunak yang dipakai, serta hasil analisis kebutuhan perangkat lunak yang berupa analisis kebutuhan proses, analisis kebutuhan masukan, analisis kebutuhan keluaran, kebutuhan perangkat lunak, kebutuhan perangkat keras dan kebutuhan antar muka

Pada bagian perancangan perangkat lunak ini membahas tentang metode perancangan yang digunakan, hasil perancangan yang berupa perancangan diagram arus data, perancangan basis pengetahuan dan perancangan tabel basis data.

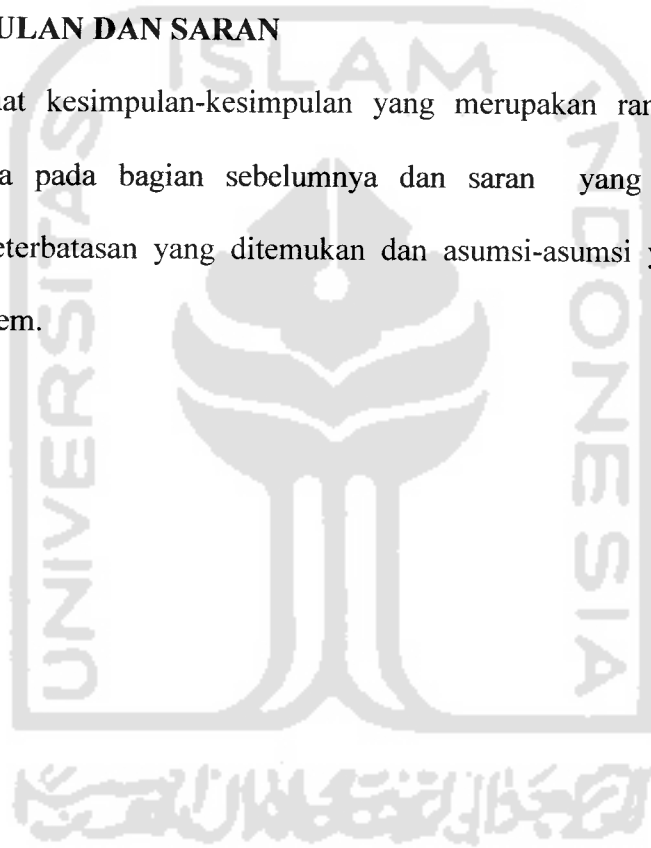
Pada bagian implementasi perangkat lunak membahas tentang batasan implementasi terhadap sistem yang dibuat dan memuat dokumentasi atau tampilan *form-form* yang telah dibangun

BAB IV HASIL DAN PEMBAHASAN

Bab ini membahas tentang analisis kinerja terhadap perangkat lunak yang telah dibangun. Bagian ini membahas analisis hasil pengujian sistem yang telah dilakukan perbandingan terhadap kebenaran dan kesesuaian kebutuhan perangkat lunak yang telah dituliskan pada bagian sebelumnya.

BAB V SIMPULAN DAN SARAN

Membuat kesimpulan-kesimpulan yang merupakan rangkuman dari hasil analisis kinerja pada bagian sebelumnya dan saran yang perlu diperhatikan berdasarkan keterbatasan yang ditemukan dan asumsi-asumsi yang dibuat selama pembuatan sistem.



BAB II

LANDASAN TEORI

2.1 SMS (*Short Message Service*)

2.1.1 Pengertian SMS

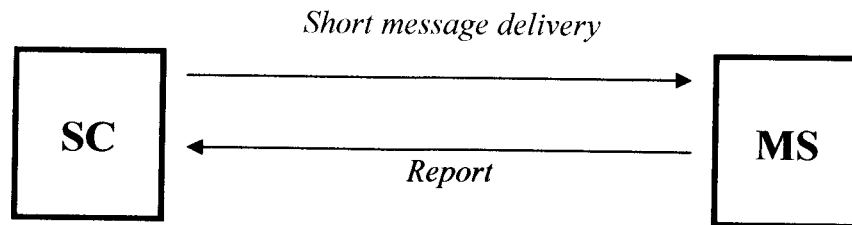
SMS (*Short Messages Services*) merupakan salah satu layanan dari berbagai operator GSM. Teknologi SMS dikembangkan dan distandarisasi oleh *European Telecommunications Standards Institute* (ETSI) untuk mengirim atau menerima pesan singkat [KOM05]. SMS merupakan layanan *messaging* yang pada umumnya terdapat pada setiap sistem jaringan *wireless digital*. SMS adalah layanan untuk mengirim dan menerima pesan tertulis (teks) dari maupun kepada perangkat bergerak (*mobile device*). Pesan teks yang dimaksud tersusun dari huruf, angka, atau karakter alfanumerik. Pesan teks dikemas dalam satu paket atau frame yang ynag berkapasitas maksimal 160 *byte* yang dapat direpresentasikan berupa 160 karakter huruf latin atau 70 karakter alfabet non-latin seperti alfabet Arab atau Cina.

Ada 2 (dua) layanan dasar pada SMS, antara lain :

1. *Short Message Mobile Terminated* (SMMT), yaitu kemampuan sistem GSM untuk menyampaikan pesan pendek yang dikirimkan dari *Service Centre* (SC) ke satu *Mobile Station* (MS), dan menghasilkan informasi tentang pengiriman itu

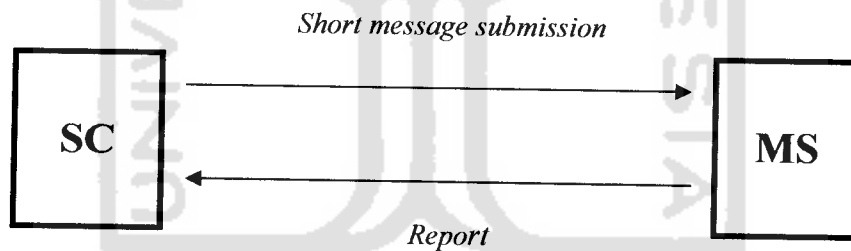
baik *delivery report* atau *status report*. Lebih jelasnya dapat dilihat pada gambar

2.1



Gambar 2.1 *short message mobile terminated*

2. *Short Message Mobile Oriented (SM-MO)*, merupakan salah satu bentuk layanan dari sistem GSM untuk menyampaikan pesan pendek yang dikirimkan dari *Mobile Station (MS)* ke satu *Short Message Entity (SME)* melalui sebuah *Service Centre (SC)*, dan menghasilkan informasi tentang pengiriman itu baik *delivery report* atau *failure report*. Lebih jelasnya dapat dilihat pada gambar 2.2



Gambar 2.2 *short message mobile oriented*

Beberapa karakteristik pesan SMS, antara lain :

1. Satu pesan SMS pasti sampai atau tidak sama sekali ke ponsel tujuan, seperti sebuah *e-mail*, sehingga jika terjadi kegagalan sistem, *time-out*, atau karena

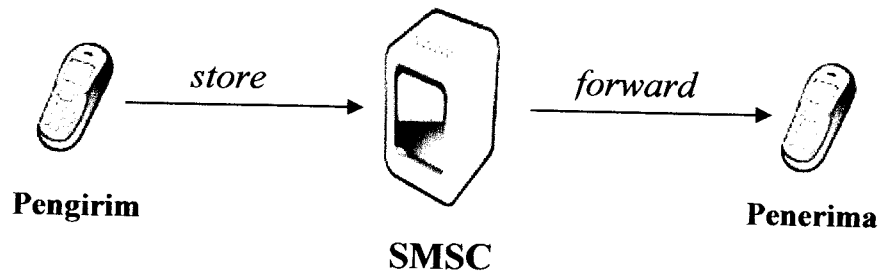
faktor lain yang menyebabkan pesan SMS gagal diterima, kemudian akan diberikan informasi (*report*) yang menyatakan pesan SMS gagal dikirim.

2. Berbeda dengan fungsi *Call* (pemanggilan), walaupun saat mengirimkan SMS ponsel tujuan dalam keadaan tidak aktif, bukan berarti pengiriman SMS akan gagal. Namun, SMS akan masuk ke antrian terlebih dahulu selama belum masa *time-out*, SMS akan segera dikirimkan jika ponsel tujuan sudah dalam keadaan aktif.
3. *Bandwith* yang digunakan rendah.

2.1.2 Mekanisme Dan Cara Kerja SMS (*Short Message Service*)

Pesan teks dikirim ke sebuah *Short Message Service Center* (SMSC) yang menyediakan mekanisme *store and forward*. Dengan SMSC pengguna dapat mengetahui status SMS yang dikirim, apakah telah sampai atau gagal diterima oleh ponsel tujuan. Jika ponsel tujuan dalam keadaan aktif dan menerima SMS yang dikirim, maka ia akan mengirimkan pesan konfirmasi ke SMSC yang menyatakan bahwa SMS telah diterima, kemudian SMSC mengirimkan kembali status tersebut kepada pengirim. Akan tetapi, jika ponsel tujuan tidak aktif atau di luar jangkauan, pesan yang dikirimkan akan disimpan pada SMSC sampai batas waktu validitas terpenuhi. Jika periode validitas waktu telah lewat, maka pesan tersebut akan dihapus dari SMSC. SMSC juga akan mengirim pesan informasi ke ponsel pengirim yang menyatakan SMS yang dikirim belum diterima atau gagal.

Untuk mekanisme cara kerja SMS dapat dilihat pada gambar 2.3



Gambar 2.3 mekanisme cara kerja sms

2.2 *Mobile* atau *Wireless*

Telepon *mobile nirkabel* merupakan turunan dari *wireless mobile phone*, *wireless handphone*, *cellular phone* atau telepon selular yang dihubungkan dengan jaringan. Oleh karena itu, disebut juga sebagai *mobile nirkabel* atau tanpa kabel sehingga memudahkan terjadinya komunikasi antar ponsel. Selain menyediakan layanan panggilan (*call*), *mobile phone* juga menyediakan beberapa layanan lain, seperti SMS, *e-mail*, *packet switching* untuk akses ke internet, dan MMS untuk mengirim dan menerima foto dan video [WIK07].

2.2.1 Perkembangan Teknologi Jaringan Wireless

Perkembangan teknologi *wireless* dibagi menjadi beberapa generasi teknologi, diantaranya : [WIK07]

1) Generasi Pertama (1G)

Generasi ini merupakan teknologi telepon selular analog atau telepon radio bergerak pertama, yang digunakan oleh kalangan maritim dan militer pada awal

abad ke-20. Termasuk dalam teknologi ini adalah teknologi selular AMPS, AMPS (*Advanced Mobile Phone Service*) dikembangkan oleh Laboratorium Bell dan penanganan distribusinya di Indonesia dilakukan oleh Komselindo, Metrosel, dan Telesera.

2) Generasi Kedua (2G)

Lahirnya generasi kedua (2G) berupa layanan telepon selular digital. Teknologi *Global System Mobile* (GSM) merupakan sistem yang paling luas digunakan di generasi ini, yang memberikan kemampuan komunikasi digital dan pengiriman data secara terbatas hingga kecepatan 9,6 kilobit/detik. Teknologi 2G yang lain diantaranya : *Digital AMPS* (D-AMPS) / *Time Division Multiple Access* (TDMA), IS-95 (*cdma-one*), CDMA, IS-136, dan *Personal Digital Celluller* (PDC).

3) Generasi Dua Setengah (2,5G)

Generasi 2,5G merupakan versi pengembangan dari teknologi sebelumnya (2G). Teknologi ini memberikan layanan paket data yang lebih tinggi. Pengembangan sistem GSM seperti GPRS dan EDGE yang termasuk dalam teknologi generasi 2,5G.

4) Generasi Ketiga (3G)

Generasi ini sering disebut juga dengan nama 3G, UMTS, dan W-CDMA, yang menjanjikan komunikasi bergerak yang memberikan dukungan kepada batas kecepatan baru yaitu penjelajahan internet bergerak (*mobile internet*) kecepatan tinggi, layanan bernilai tambah yang lebih maju, menambah efisiensi dan

kapasitas jaringan, telepon video, peningkatan kualitas layanan (Quality of Services / QOS), dan mencapai kecepatan transfer data yang lebih tinggi. Sistem *wireless* 3G akan menangani layanan sampai 384 kbps dalam aplikasi yang luas sampai 2Mbps untuk aplikasi internal.

5) Generasi Keempat (4G)

Generasi ini belum didefinisikan secara pasti. Teknologi 4G yang direncanakan akan memberikan kecepatan lebih tinggi, kapasitas tinggi, biaya rendah per bit, layanan berbasis *Internet Protocol* (IP). Tujuan yang akan dicapai yaitu kecepatan pengiriman data sampai 20Mbps, yang boleh jadi merupakan jaringan dengan kombinasi dari teknologi-teknologi berbeda yang bekerjasama dengan protokol yang sesuai (seperti *Mobile IP*).

2.2.2 GSM Cable Data

GSM cable adalah perangkat keras atau rangkaian kabel data yang berfungsi untuk menghubungkan antara ponsel dengan komputer agar bisa berkomunikasi lewat *Port Comm*. Kabel data biasanya digunakan sebagai perantara proses kegiatan antara ponsel ke komputer. Untuk bisa mengakses ponsel, kabel data ini harus dihubungkan ke COM1, COM2, ataupun Virtual Comm Port (misal : USB, IrDA, Bluetooth *connections*) pada komputer [WIK07].

2.3 Access and Terminals (AT) Command

AT *command* merupakan *command line* (baris perintah) yang berfungsi untuk komunikasi antara *handphone* dan komputer dengan menggunakan *serial port* pada komputer. Beberapa hal yang dapat dilakukan AT *command*, antara lain : dapat mengetahui *vendor handphone*, membaca pesan, mengirim pesan, deteksi pesan baru yang masuk secara otomatis, menghapus pesan, dan masih banyak lagi fungsi-fungsi lainnya

AT *command* pada prinsipnya mempunyai kesamaan dengan *command prompt* pada DOS, dimana setiap eksekusi perintahnya dilakukan perbaris. Untuk sintaksnya diawali dengan kata "AT", kemudian diikuti karakter lainnya sesuai dengan fungsinya.

Contoh perintah AT *command* :

- ATEI : untuk mengetahui kondisi port.
- AT+CGMI : untuk mengetahui vendor ponsel yang digunakan
- AT+CMGF : perintah untuk mengetahui format pengiriman SMS. Terdapat 2 (dua) jenis format SMS, yaitu format teks dan Protokol Data Unit (PDU).
- AT+CMGR : perintah untuk membaca salah satu SMS yang ada pada SIM *card*
- AT+COPS : perintah untuk memilih dan mendaftarkan jaringan operator GSM

Untuk melakukan perintah dengan AT *command* ke *handphone*, maka *handphone* terlebih dahulu harus dihubungkan ke komputer dengan menggunakan media transfer data (misal : kabel data, *Bluetooth* dan *IrDA*). Sintaks AT *command* yang digunakan pada *handphone* mengikuti standar dari ETSI GSM 07.05. beberapa

AT *command* yang dapat digunakan untuk menangani pesan SMS pada ponsel terdapat pada tabel 2.1

Tabel 2.1 Tabel AT command [MOT01]

Perintah	Fungsi	Tipe Perintah
AT	Mengecek apakah ponsel sudah terhubung	Pengecekan Ponsel
AT+CMGC	Mengirim perintah SMS	Pengiriman dan Penulisan Pesan
AT+CMGD	Menghapus SMS di memori SMS	Pengiriman dan Penulisan Pesan
AT+CMGF	Mengatur format mode SMS dari terminal	Pengiriman dan Penulisan Pesan
AT+CMGL	Menampilkan daftar SMS yang ada pada SIM card	Pengiriman dan Penulisan Pesan
AT+CMGR	Membaca sebuah pesan SMS	Pengiriman dan Penulisan Pesan
AT+CMGS	Mengirim sebuah pesan SMS	Pengiriman dan Penulisan Pesan
AT+CMGW	Menulis SMS ke memori SMS	Pengiriman dan Penulisan Pesan
AT+CNMA	Tanda terima dari keluaran langsung SMS	Pengiriman dan Penulisan Pesan
AT+CNMI	Menampilkan SMS baru yang masuk secara otomatis	Pengiriman dan Penulisan Pesan
AT+CPMS	Memilih penyimpanan pesan SMS	Konfigurasi Umum
AT+CSCS	Menetapkan jenis <i>encoding</i>	Konfigurasi Umum
AT+CSCA	Alamat SMS <i>service centre</i>	Konfigurasi Umum
AT+CSCB	Memilih pesan <i>cell broadcast</i>	Konfigurasi Umum
AT+CSMS	Pemilihan layanan pesan	Konfigurasi Umum

2.4 Basis Data

Basis data terdiri dari 2 kata, yaitu basis dan data. Basis Kurang lebihnya diartikan sebagai markas atau gudang, tempat bersarang/berkumpul. Sedangkan data adalah representasi fakta dunia nyata yang memiliki suatu obyek seperti manusia,

barang, hewan, konsep, keadaan dan sebagainya, yang disertakan dalam bentuk huruf, symbol teks, gambar, bunyi atau kombinasinya [FAT99].

Basisdata sendiri dapat diartikan dalam sejumlah sudut pandang seperti

- Himpunan kelompok data atau arsip yang saling berhubungan yang diorganisasikan sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
- Kumpulan data yang saling berhubungan yang disimpan secara bersamaan sedemikian rupa dan tanpa pengulangan (redundansi) yang tidak perlu untuk memenuhi berbagai kebutuhan.
- Kumpulan file/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

2.4.1 Obyektif Basis Data

Pemanfaatan Basis Data dilakukan untuk memenuhi sejumlah tujuan (obyektif) seperti berikut ini :[FAT99]

1. Kecepatan dan kemudahan (*speed*)
2. Efisiensi Ruang penyimpanan (*space*)
3. Keakuratan (*accuray*)
4. Ketersediaan (*availability*)
5. Kelengkapan (*completeness*)
6. Keamanan (*security*)
7. Kebersamaan Pemakaian (*sharibility*)

2.4.2 Sistem Basis Data

Sistem basis data merupakan lingkup yang lebih luas dari pada basis data. Sistem Basis Data memuat sekumpulan basisdata dalam suatu sistem yang mungkin tidak ada hubungan satu sama lain, tetapi secara keseluruhan mempunyai hubungan sebagai sebuah sistem dengan didukung oleh komponen lainnya.[SUT04]

Secara umum sebuah Sistem Basisdata merupakan sistem yang terdiri atas kumpulan *file* (tabel) yang saling berhubungan (dalam sebuah basis data didalam sistem computer) dan sekumpulan program (DBMS) yang memungkinkan beberapa pemakai dan atau program lain untuk mrngakses dan memanipulasi *file-file* (tabel-tabel) tersebut.[FAT99]

2.4.3 Komponen Basis Data

Komponen basis data terdiri dari [FAT99] :

- a. Perangkat keras (*Hardware*)
- b. Sistem Operasi (*Operating Sistem*)
- c. Basis Data (*Database*)
- d. Sistem Pengelola Basis Data (*Database Management Sistem*)
- e. Pemakai (*User*)
- f. Aplikasi (perangkat lunak) lain yang bersifat optimal

2.4.4 Perancangan Basis Data

Pada perancangan basis data diperlukan adanya model tertentu yang digunakan dalam perancangan sistem. Pemodelan sistem ini sangat diperlukan, karena :

- a. Dapat memfokuskan perhatian pada hal-hal penting dalam sistem
- b. Menggambarkan perubahan dan korelasi terhadap kebutuhan pemakai dengan resiko dan biaya mahal
- c. Menguji pengertian penganalisa sistem terhadap kebutuhan pemakai dan membantu pendisain sistem dan pemrogram membangun sistem



BAB III

METODOLOGI

3.1 Analisis Kebutuhan Perangkat Lunak

3.1.1 Metode Analisis

Aplikasi pengingat untuk tindakan terapi *insulin* bagi penderita *diabetes mellitus* dibangun dengan menerapkan konsep teknologi 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.

3.1.2 Hasil Analisis

Dari data hasil *survey* dan wawancara selama penelitian dan setelah melalui tahap analisis, maka telah dapat ditetapkan kebutuhan-kebutuhan apa saja yang meliputi proses, *input* dan *output*, diantaranya :



3.1.2.1 Analisis Kebutuhan Proses

Kebutuhan proses dalam sistem pengingat untuk tindakan terapi *insulin* bagi penderita *diabetes mellitus* :

- Proses pembuatan *gateway*, untuk komunikasi *handphone* yang berfungsi sebagai modem dengan komputer (sistem).
- Proses pengaturan modem.
- Proses pengolahan data pasien *diabetes mellitus*.
- Proses penentuan penjadwalan pemberian terapi *insulin*.
- Proses penentuan pemberian dosis *insulin*.

3.1.2.2 Analisis Kebutuhan Input

Input dari aplikasi pengingat untuk tindakan terapi *insulin* bagi penderita *diabetes mellitus*, terdiri dari 3 karakteristik masukan yaitu *Admin*, *Paramedis* dan Penderita *diabetes mellitus*.

a) Input Admin

Input Admin adalah suatu masukan yang diberikan oleh *Admin*, berupa data pasien dan data koneksi modem.

1. Input data pasien

Masukan data pasien *diabetes mellitus* tentang riwayat kesehatannya.

2. Input data koneksi modem

Masukan data properti modem, misalnya : nama *port* yang digunakan.

3. Password baru

Masukan untuk mengubah *password Admin*.

b) Input Pengguna

Input pengguna adalah masukan yang diberikan oleh penderita *diabetes mellitus* berupa SMS yang berisikan tentang kondisi pasien terbaru (misal, kadar gula darah dan berat badan).

3.1.2.3 Analisis Kebutuhan Output

Output yang diperoleh dari sistem pengingat untuk tindakan terapi *insulin* bagi penderita *diabetes mellitus* berbasis SMS ini adalah berupa jadwal terapi *insulin* beserta dosisnya.

3.1.3 Kebutuhan Antarmuka

Untuk perancangan antarmuka menggunakan NetBeans 5.5, merupakan pilihan yang tepat untuk mengimplementasikan sistem pengingat untuk tindakan terapi *insulin*. Selain karena tampilan *end-user* yang relatif mudah bagi para penggunanya dalam hal melakukan perancangan dan pembangunan aplikasi perangkat lunak. Juga adanya fasilitas *Matisse GUI Builder* yang berfungsi untuk lebih memudahkan dalam mendesain sistem yang menerapkan konsep WYSWYG (*What You See What You Get*), hanya cukup melakukan *drag* dan *drop* saja. Kelebihan lain yang dihadirkan NetBeans 5.5 adalah adanya fasilitas *code completion* berfungsi sebagai *code generator*, sehingga dapat menghilangkan

- c. VGA 32 MB
- d. *Hardisk* 40 GB
- e. Monitor resolusi 1152 x 864
- f. *Mouse* dan *Keyboard*
- g. *Handphone* yang berfungsi sebagai modem

3.2 Perancangan Perangkat Lunak

3.2.1 Metode Perancangan

Dalam membangun perangkat lunak sistem untuk tindakan terapi *insulin* bagi penderita *diabetes mellitus* menggunakan pendekatan berorientasi objek. Perancangan berorientasi objek pada dasarnya menekankan pada kemudahan perawatan serta pengembangan sistem lebih lanjut nantinya. Untuk bahasa pemodelan yang digunakan dalam menspesifikasikan, memvisualisasikan membangun dan dokumentasi, adalah bahasa pemodelan UML (*Unified Modeling Language*).

3.2.2 Hasil Perancangan

Berdasarkan analisis yang telah dilakukan, maka telah dapat ditetapkan apa saja yang menjadi masukan, proses, keluaran, atau antarmuka sistem yang dibuat agar sesuai dengan apa yang diharapkan. Hasil perancangan sistem ini dibedakan menjadi beberapa bagian sesuai dengan bagaian-bagaian yang digunakan pada metode perancangan dengan bahasa pemodelan UML.

kesulitan dalam pengetikan perintah-perintah JAVA dengan menggunakan manual teks.

3.1.4 Analisis Kebutuhan Perangkat Lunak

Perangkat keras komputer tidak berarti tanpa adanya perangkat lunak dan juga sebaliknya. Jadi perangkat lunak dan perangkat keras saling mendukung satu sama lain. Perangkat keras hanya berfungsi jika adanya instruksi-intruksi. Instruksi-instruksi inilah disebut dengan perangkat lunak. Dalam penelitian ini penyusun menggunakan perangkat lunak JAVA. Karena merupakan salah satu perangkat lunak yang dapat mengatasi permasalahan tentang aplikasi pengingat untuk tindakan terapi *insulin* bagi penderita *diabetes*

3.1.5 Analisis Kebutuhan Perangkat Keras

Komputer terdiri dari perangkat keras dan perangkat lunak yang saling berkomunikasi untuk menyelesaikan perintah tertentu. Fungsi komputer sebagai alat bantu manusia sudah banyak dijumpai, misalnya saja aplikasi yang menerapkan teknologi SMS *gateway*. Oleh karena itu penyusun berusaha untuk membuat salah satu aplikasi yang memanfaatkan teknologi tersebut, yaitu aplikasi pengingat untuk tindakan terapi *insulin* bagi penderita *diabetes mellitus*.

Untuk perangkat keras yang dibutuhkan dalam membangun dan menjalankan aplikasi pengingat untuk tindakan terapi *insulin* bagi penderita *diabetes mellitus* tersebut adalah :

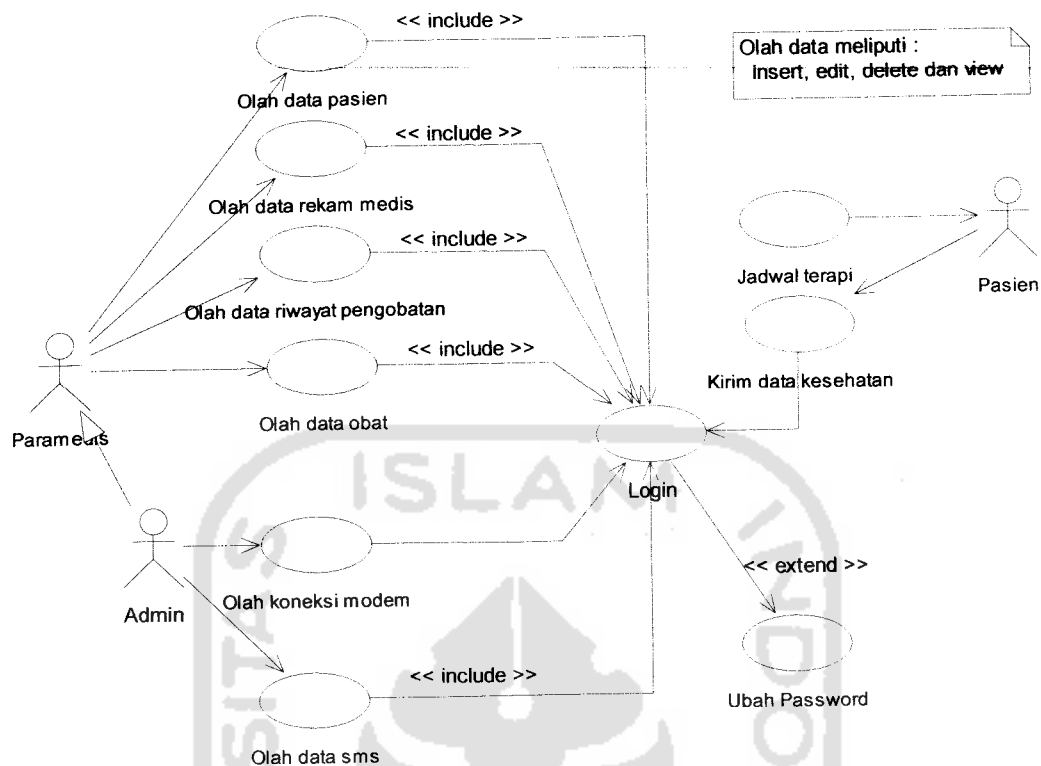
- a. Intel P4 2.0Ghz
- b. Memori 256 MB

3.2.2.1 Perancangan UML

3.2.2.1.1 Use Case Diagram

Merupakan diagram yang bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* buah sistem dengan suatu sistem tersendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. *Use case diagram* terdiri dari sebuah aktor dan interaksi yang dilakukannya, aktor tersebut dapat berupa manusia, perangkat keras, sistem lain, ataupun yang berinteraksi dengan sistem.

Pada sistem pengingat untuk tindakan terapi insulin bagi penderita *diabetes mellitus* menjelaskan tentang hubungan antara sistem dengan aktor. Hubungan ini dapat berupa *input* aktor ke sistem ataupun *output* ke aktor. *Use case* merupakan dokumen naratif yang mendeskripsikan kasus-kasus atau kejadian-kejadian daripada aktor dalam menggunakan *system* untuk menyelesaikan sebuah proses. Berikut ini adalah gambar yang menjelaskan sistem pengingat untuk tindakan terapi insulin bagi penderita *diabetes mellitus* dalam model *use case diagram*, seperti terlihat pada gambar 3.1



Gambar 3.1 use case diagram

Pada *use case diagram* diatas *user* Pasien (penderita *diabetes melitus*) dapat mengirimkan kondisi kesehatan terbaru apabila telah terdaftar sebagai anggota. Selain itu *user* Pasien juga secara pasif berinteraksi dengan sistem, seperti memperoleh jadwal terapi *insulin* dan dosis *insulin* yang digunakan. Sedangkan untuk *user Admin* dapat secara aktif berinteraksi dengan sistem, yaitu dalam kepentingan pengolahan data. Pengolahan data tersebut meliputi olah data pasien, olah data koneksi modem dan olah data obat.

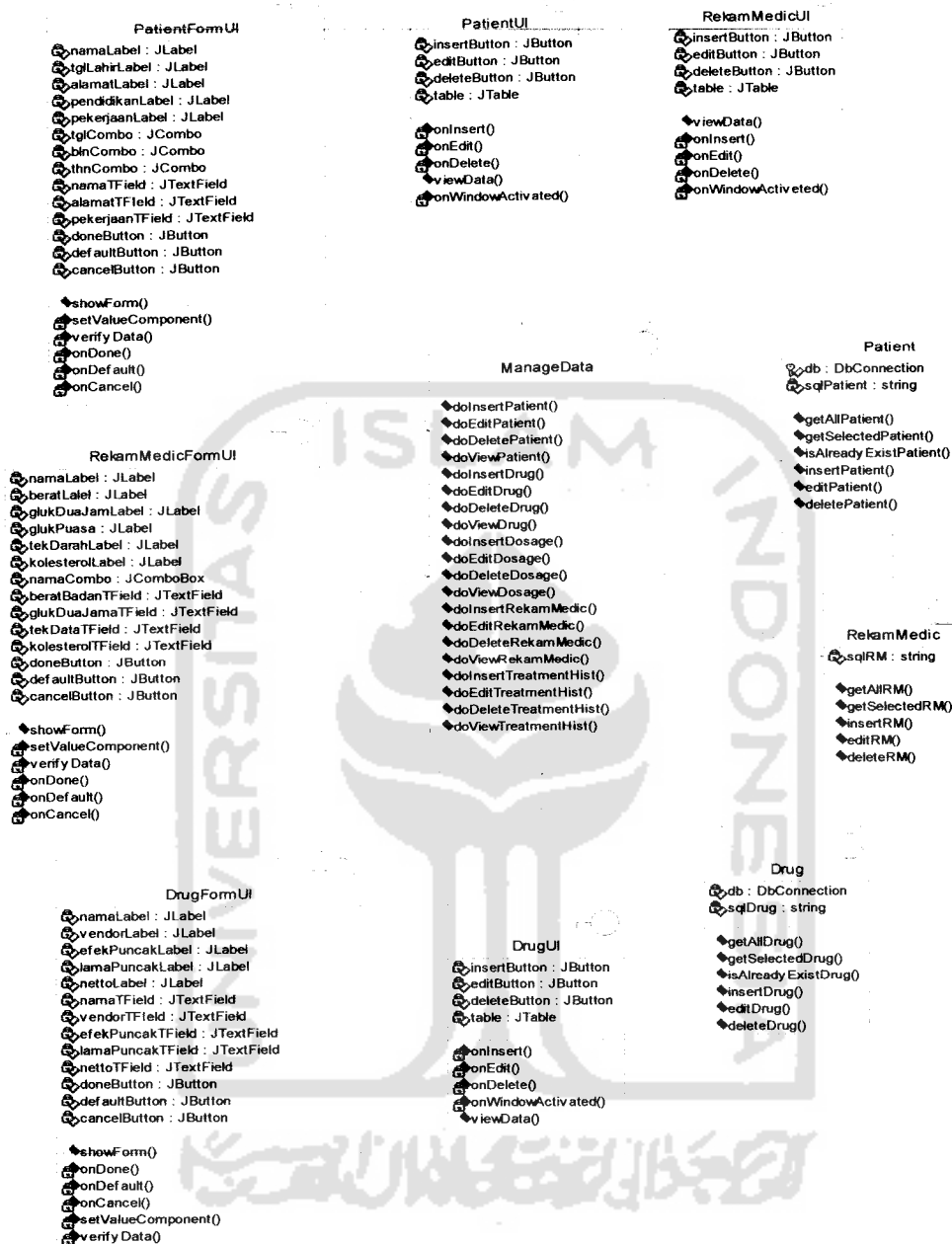
3.2.2.1.2 Class Diagram

Class diagram digunakan untuk melakukan visualisasi struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak digunakan. *Class diagram* juga dapat memperlihatkan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain (*logical view*) dari suatu sistem. Selama proses desain, *class diagram* berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat.

Berikut ini adalah akan dijelaskan *class diagram* yang digunakan untuk melakukan visualisasi struktur kelas-kelas yang terdapat dalam sistem pengingat untuk tindakan terapi *insulin* bagi penderita *diabetes mellitus* yang dibagi ke dalam 3 buah modul utama, yaitu modul *data management*, modul *system management* dan modul *user management*.

3.2.2.1.2.1 Modul Data Management

Modul *data management* mengolah data-data medis, yang terdiri dari data pasien, data rekam medis, data riwayat pengobatan, data obat dan data dosis yang diberikan. Proses pengolahan data meliputi *insert*, *edit*, *delete* dan *view*. Modul data management terdiri dari beberapa *class* yang saling terelasi. Lebih jelasnya dapat dilihat pada gambar 3.2



Gambar 3.2 class diagram modul data management

3.2.2.1.2.2 Modul System 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 sitem dalam melakukan pengiriman SMS secara otomatis. Modul ini terdapat beberapa *class* yang saling terelasi, lebih jelasnya dapat dilihat pada gambar 3.3



Gambar 3.3 class diagram modul system management

3.2.2.1.2.3 Modul *User Management*

Modul *user*. Artinya modul ini akan menangani pembuatan *user account* dan pembagian wewenang (admin, paramedis) masing-masing *user* terhadap penggunaan sistem itu.



Gambar 3.4 *Class diagram* modul *user management*

3.2.2.1.3 Sequence Diagram

Sequence diagram digunakan untuk menjelaskan interaksi objek yang disusun dalam suatu urutan waktu. Diagram ini secara khusus berasosiasi dengan *use case*. *Sequence diagram* juga dapat memperlihatkan tahap demi tahap proses yang seharusnya terjadi untuk menghasilkan sesuatu di dalam *use case*.

Pada bagian ini akan dijelaskan *sequence diagram* dari sistem pengingat untuk tindakan terapi *insulin* bagi penderita *diabetes mellitus*. Dalam *sequence diagram* ini menggambarkan interaksi antar objek pada sistem secara berurutan. Penjelasan *sequence diagram* akan dijelaskan berdasarkan atas modul-modul yang dibuat yaitu modul *data management*, modul *system management*, dan modul *user management*.

3.2.2.1.3.1 Modul Data Management

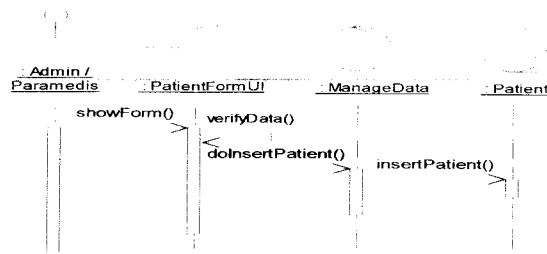
Modul *data management* adalah modul berfungsi untuk mengolah data-data pasien beserta data pendukung lainnya, seperti data rekam medis, riwayat pengobatan, obat dan dosis. Berikut ini akan jelaskan mengenai *sequence diagram* dari modul *Accounting*, antara lain :

3.2.2.1.3.1.1 Tambah Data Pasien

Objek yang berkaitan dengan *sequence* ini adalah sebagai berikut

Aktor	: <i>Admin, Paramedis</i>
Class Boundary	: <i>PatientFormUI</i>
Class Control	: <i>ManageData</i>
Class Entity	: <i>Patient</i>
Keterangan	: 1. <i>Admin</i> mengawali <i>sequence</i> ini dengan memanggil <i>method</i> <i>showForm()</i> . 2. <u><i>PatientFormUI</i></u> melakukan instansiasi ke <i>ManageData</i> dan memanggil <i>method</i> <i>doInsertPatient()</i> untuk memproses <i>input</i> dari <i>admin</i> . 3. <u><i>ManageData</i></u> melakukan instansiasi ke <u><i>Patient</i></u> dan memanggil <i>method</i> <i>insertPatient()</i> untuk memasukkan data ke basis data tabel pasien.

Sequence diagram tambah data pasien dapat dilihat pada gambar 3.5 :



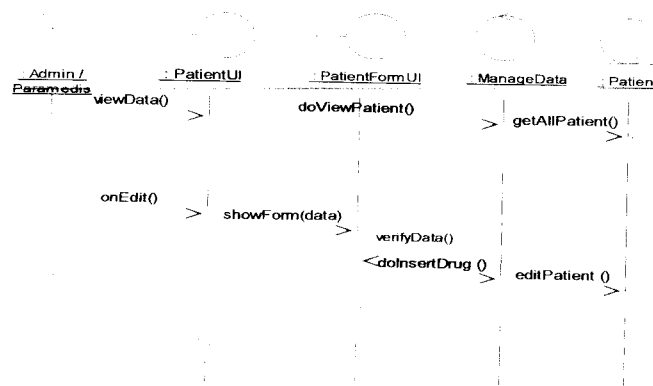
Gambar 3.5 *sequence diagram* tambah pasien

3.2.2.1.3.1.2 Ubah Data Pasien

Objek yang berkaitan dengan *sequence* ini adalah sebagai berikut

- | | | |
|-----------------------|---|--|
| Aktor | : | <i>Admin</i> |
| <i>Class Boundary</i> | : | <i>PatientUI, PatientFormUI</i> |
| <i>Class Control</i> | : | <i>ManageData</i> |
| <i>Class Entity</i> | : | <i>Patient</i> |
| Keterangan | : | <ol style="list-style-type: none"> 1. <i>Admin</i> mengawali <i>sequence</i> ini dengan memanggil <i>method</i> <i>viewData()</i>. 2. <u><i>PatientUI</i></u> melakukan instansiasi ke <u><i>ManageData</i></u> dan memanggil <i>method</i> <i>doViewPatient()</i> untuk memerintahkan agar semua data pasien ditampilkan 3. <u><i>ManageData</i></u> melakukan instansiasi ke <u><i>Patient</i></u> dan memanggil <i>method</i> <i>getAllPatient()</i> untuk mengambil semua data pasien. 4. <i>Admin</i> kemudian memanggil <i>method</i> <i>doEdit()</i>. 5. <i>PatientUI</i> melakukan instansiasi ke <u><i>PatientFormUI</i></u> dan memanggil <i>method</i> <i>showForm()</i> untuk menampilkan form. 6. <u><i>PatientFormUI</i></u> melakukan instansiasi ke <u><i>ManageData</i></u> dan memanggil <i>method</i> <i>doEditPatient()</i> untuk memproses data yang telah diubah. 7. <u><i>ManageData</i></u> melakukan instansiasi ke <u><i>Patient</i></u> dan memanggil <i>method</i> <i>editPatient()</i> untuk mengubah <i>record</i> data pasien. |

Sequence diagram ubah data pasien dapat dilihat pada gambar 3.6



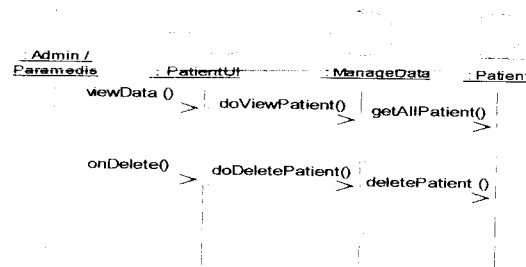
Gambar 3.6 *sequence diagram* ubah pasien

3.2.2.1.3.1.3 Hapus Data Pasien

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

- | | |
|----------------|--|
| Aktor | : Admin |
| Class Boundary | : PatientUI |
| Class Control | : ManageData |
| Class Entity | : Patient |
| Keterangan | : 1. Admin mengawali <i>sequence</i> ini dengan memanggil <i>method</i> <code>viewData()</code> . |
| | 2. <u>PatientUI</u> melakukan instansiasi ke <u>ManageData</u> dan memanggil <i>method</i> <code>doViewPatient()</code> untuk memerintahkan agar semua data pasien ditampilkan |
| | 3. <u>ManageData</u> melakukan instansiasi ke <u>Patient</u> dan memanggil <i>method</i> <code>getAllPatient()</code> untuk mengambil semua data pasien. |
| | 4. Admin kemudian memanggil <i>method</i> <code>doDelete()</code> . |
| | 5. <u>PatientUI</u> melakukan instansiasi ke <u>ManageData</u> dan memanggil <i>method</i> <code>doDelete()</code> untuk melakukan proses penghapusan. |
| | 6. <u>ManageData</u> melakukan instansiasi ke <u>Patient</u> dan memanggil <i>method</i> <code>deletePatient()</code> untuk menghapus <i>record</i> data pasien. |

Sequence diagram hapus data pasien dapat dilihat pada gambar 3.7



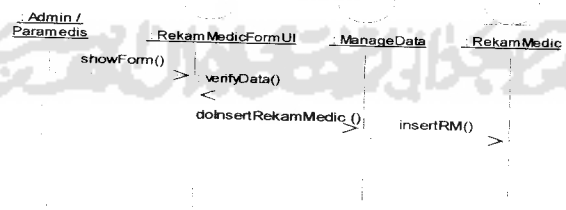
Gambar 3.7 *Sequence diagram* hapus pasien

3.2.2.1.3.1.4 Tambah Data Rekam Medis

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

- | | | |
|-----------------------|---|--|
| Aktor | : | <i>Admin, Paramedis</i> |
| <i>Class Boundary</i> | : | <i>RekamMedicFormUI</i> |
| <i>Class Control</i> | : | <i>ManageData</i> |
| <i>Class Entity</i> | : | <i>RekamMedic</i> |
| Keterangan | : | <ol style="list-style-type: none"> 1. <i>Admin</i> mengawali <i>sequence</i> ini dengan memanggil <i>method</i> <i>showForm()</i>. 2. <u><i>RekamMedicFormUI</i></u> melakukan instansiasi ke <i>ManageData</i> dan memanggil <i>method</i> <i>doInsertRekamMedic()</i> untuk memproses <i>input</i> data dari <i>admin</i>. 3. <u><i>ManageData</i></u> melakukan instansiasi ke <u><i>RekamMedic</i></u> dan memanggil <i>method</i> <i>insertRM()</i> untuk memasukkan data ke basis data tabel rekam medis. |

Sequence diagram tambah data rekam medis dapat dilihat pada gambar 3.8



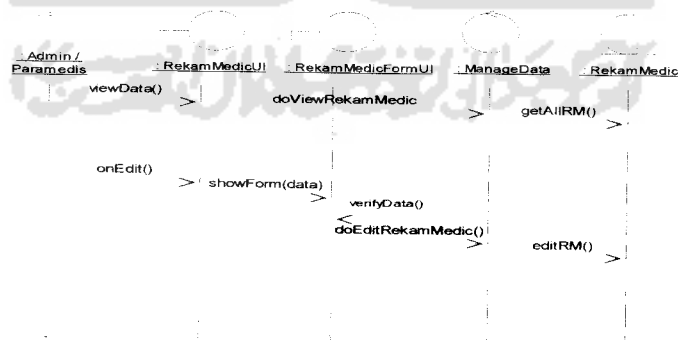
Gambar 3.8 *sequence diagram* tambah data rekam medis

3.2.2.1.3.1.5 Ubah Data Rekam Medis

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

- Aktor : *Admin, Paramedis*
 Class Boundary : *RekamMedicUI, RekamMedicFormUI*
 Class Control : *ManageData*
 Class Entity : *RekamMedic*
 Keterangan :
1. *Admin* mengawali *sequence* ini dengan memanggil *method* `viewData()`.
 2. *RekamMedicUI* melakukan instansiasi ke *ManageData* dan memanggil *method* `doViewRekamMedic()` untuk memerintahkan agar semua data rekam medis ditampilkan
 3. *ManageData* melakukan instansiasi ke *RekamMedic* dan memanggil *method* `getAllRM()` untuk mengambil semua data rekam medis.
 4. *Admin* kemudian memanggil *method* `doEdit()`.
 5. *RekamMedicUI* melakukan instansiasi ke *RekamMedicFormUI* dan memanggil *method* `showForm()` untuk menampilkan form.
 6. *RekamMedicFormUI* melakukan instansiasi ke *ManageData* dan memanggil *method* `doEditRekamMedic()` untuk memproses data yang telah diubah.
 7. *ManageData* melakukan instansiasi ke *RekamMedic* dan memanggil *method* `editRM()` untuk mengubah *record* data rekam medis.

Sequence diagram ubah data rekam medis dapat dilihat pada gambar 3.9



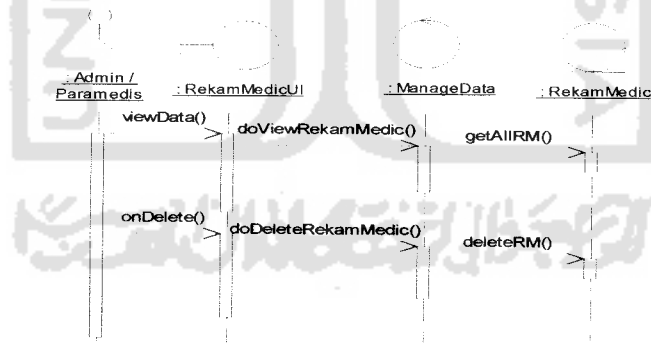
Gambar 3.9 *sequence diagram* ubah data rekam medis

3.2.2.1.3.1.6 Hapus Data Rekam Medis

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

- Aktor : *Admin*
 Class Boundary : *RekamMedicUI*
 Class Control : *ManageData*
 Class Entity : *RekamMedic*
 Keterangan :
1. *Admin* mengawali *sequence* ini dengan memanggil *method* `viewData()`.
 2. *RekamMedicUI* melakukan instansiasi ke *ManageData* dan memanggil *method* `doViewRekamMedic()` untuk memerintahkan agar semua data rekam medis ditampilkan
 3. *ManageData* melakukan instansiasi ke *RekamMedic* dan memanggil *method* `getAllRM()` untuk mengambil semua data rekam medis.
 4. *Admin* kemudian memanggil *method* `onDelete()`.
 5. *RekamMedicUI* melakukan instansiasi ke *ManageData* dan memanggil *method* `doDeleteRekamMedic()` untuk melakukan proses penghapusan.
 6. *ManageData* melakukan instansiasi ke *RekamMedic* dan memanggil *method* `deleteRM()` untuk menghapus *record* data rekam medis.

Sequence diagram hapus data rekam medis dapat dilihat pada gambar 3.10



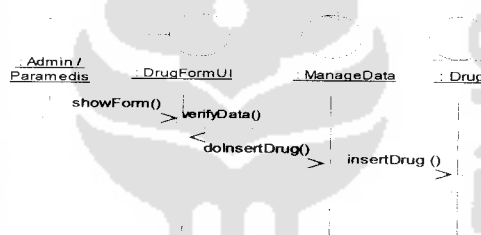
Gambar 3.10 *sequence diagram* hapus data rekam medis

3.2.2.1.3.1.7 Tambah Data Obat

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

- Aktor : *Admin, Paramedis*
 Class Boundary : *DrugFormUI*
 Class Control : *ManageData*
 Class Entity : *Drug*
 Descriptions : 1. *Admin* mengawali *sequence* ini dengan memanggil *method* *showForm()*.
 2. *DrugFormUI* melakukan instansiasi ke *ManageData* dan memanggil *method* *doInsertDrug()* untuk memproses *input* data dari *admin*.
 3. *ManageData* melakukan instansiasi ke *Drug* dan memanggil *method* *insertDrug()* untuk memasukkan data ke basis data tabel obat.

Sequence diagram tambah data obat dapat dilihat pada gambar 3.11



Gambar 3.11 *sequence diagram* tambah data obat

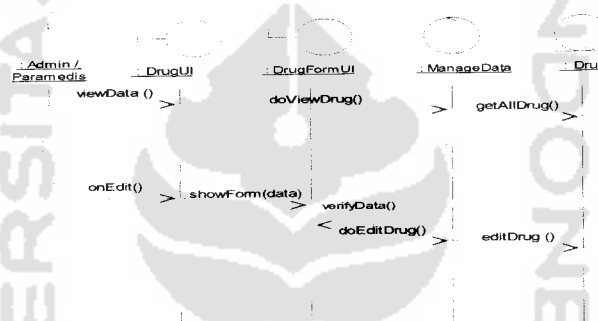
3.2.2.1.3.1.8 Ubah Data Obat

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

- Aktor : *Admin*
 Class Boundary : *DrugUI, DrugFormUI*
 Class Control : *ManageData*
 Class Entity : *Drug*
 Keterangan : 1. *Admin* mengawali *sequence* ini dengan memanggil *method* *viewData()*.
 2. *DrugUI* melakukan instansiasi ke *ManageData* dan memanggil *method* *doViewDrug()* untuk memerintahkan agar semua data obat ditampilkan
 3. *ManageData* melakukan instansiasi ke *Drug* dan

- memanggil *method* `getAllDrug()` untuk mengambil semua data obat.
4. Admin kemudian memanggil *method* `doEdit()`.
 5. DrugUI melakukan instansiasi ke DrugFormUI dan memanggil *method* `showForm()` untuk menampilkan form.
 6. DrugFormUI melakukan instansiasi ke ManageData dan memanggil *method* `doEditDrug()` untuk memproses data yang telah diubah.
 7. ManageData melakukan instansiasi ke Drug dan memanggil *method* `editDrug()` untuk mengubah *record* data obat.

Sequence diagram ubah data obat dapat dilihat pada gambar 3.12

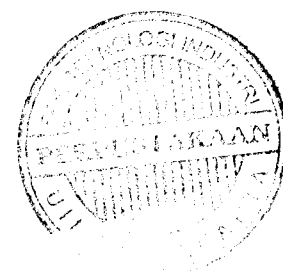


Gambar 3.12 *sequence diagram* ubah data obat

3.2.2.1.3.1.9 Hapus Data Obat

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

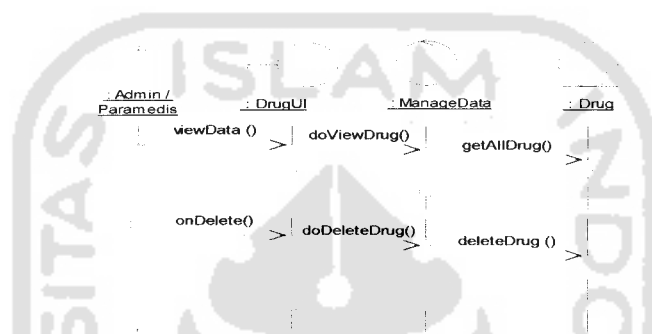
- | | | |
|----------------|---|---|
| Aktor | : | Admin |
| Class Boundary | : | DrugUI |
| Class Control | : | ManageData |
| Class Entity | : | Drug |
| Keterangan | : | <ol style="list-style-type: none"> 1. Admin mengawali <i>sequence</i> ini dengan memanggil <i>method</i> <code>viewData()</code>. 2. <u>DrugUI</u> melakukan instansiasi ke <u>ManageData</u> dan memanggil <i>method</i> <code>doViewDrug()</code> untuk memerintahkan agar semua data obat ditampilkan 3. <u>ManageData</u> melakukan instansiasi ke <u>Drug</u> dan memanggil <i>method</i> <code>getAllDrug()</code> untuk |



mengambil semua data obat.

4. Admin kemudian memanggil *method* `doDelete()`.
5. DrugUI melakukan instansiasi ke ManageData dan memanggil *method* `doDelete()` untuk melakukan proses penghapusan.
6. ManageData melakukan instansiasi ke Drug dan memanggil *method* `deleteDrug()` untuk menghapus *record* data obat.

Sequence diagram hapus data obat dapat dilihat pada gambar 3.13 :



Gambar 3.13 *sequence diagram* hapus data obat

3.2.2.1.3.2 Modul System Management

Modul *system management* adalah modul yang berfungsi untuk mengolah sistem agar lebih mudah disesuaikan dengan keadaan serta untuk lebih mudah dalam pemeliharannya. Data yang diolah dalam modul ini terdiri dari data koneksi modem dan data sms. Berikut ini akan jelaskan mengenai *sequence diagram* dari modul *Accounting*, antara lain :

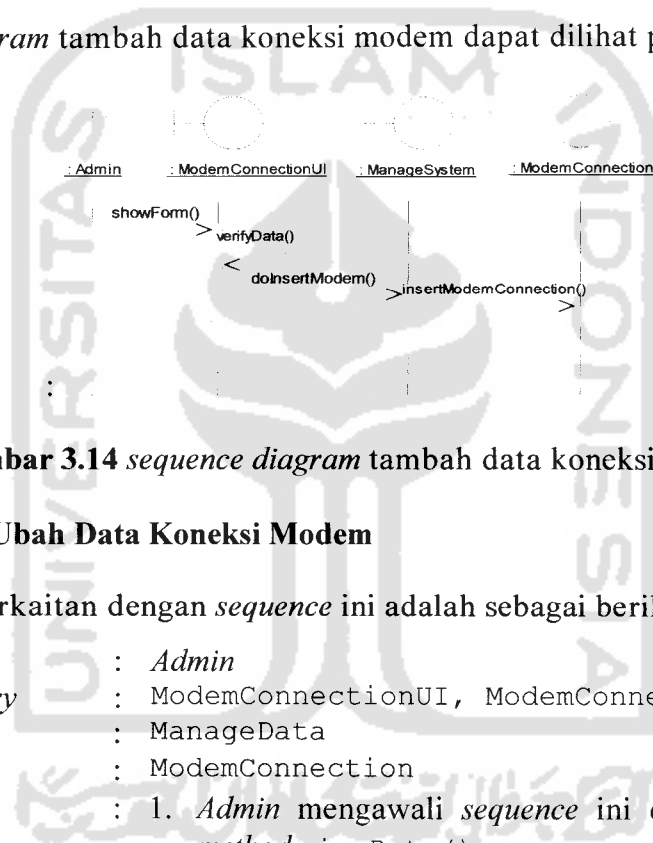
3.2.2.1.3.2.1 Tambah Data Koneksi Modem

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

Aktor	: Admin
Class Boundary	: ModemConnectionFormUI
Class Control	: ManageData
Class Entity	: ModemConnection

- Keterangan : 1. *Admin* mengawali *sequence* ini dengan memanggil *method* `showForm()`.
2. ModemConnectionFormUI melakukan instansiasi ke ManageData dan memanggil *method* `doInsertModem()` untuk memproses *input* data dari *admin*.
3. ManageData melakukan instansiasi ke ModemConnection dan memanggil *method* `insertModem()` untuk memasukkan data ke basis data tabel koneksi modem.

Sequence diagram tambah data koneksi modem dapat dilihat pada gambar 3.14



Gambar 3.14 *sequence diagram* tambah data koneksi modem

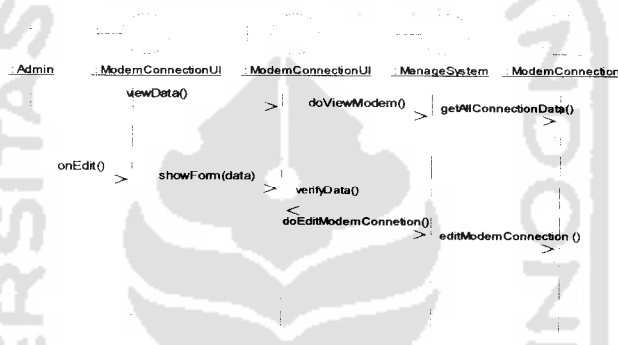
3.2.2.1.3.2.2 Ubah Data Koneksi Modem

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

- Aktor : *Admin*
- Class Boundary : ModemConnectionUI, ModemConnectionFormUI
- Class Control : ManageData
- Class Entity : ModemConnection
- Keterangan : 1. *Admin* mengawali *sequence* ini dengan memanggil *method* `viewData()`.
2. ModemConnectionUI melakukan instansiasi ke ManageData dan memanggil *method* `doViewModem()` untuk memerintahkan agar semua data modem ditampilkan
3. ManageData melakukan instansiasi ke ModemConnection dan memanggil *method* `getAllModem()` untuk mengambil semua data modem.

4. Admin kemudian memanggil *method* `doEdit()`.
5. ModemConnectionUI melakukan instansiasi ke ModemConnectionFormUI dan memanggil *method* `showForm()` untuk menampilkan form.
6. ModemConnectionFormUI melakukan instansiasi ke ManageData dan memanggil *method* `doEditModem()` untuk memproses data yang telah diubah.
7. ManageData melakukan instansiasi ke ModemConnection dan memanggil *method* `editModem()` untuk mengubah *record* data modem.

Sequence diagram ubah data koneksi modem dapat dilihat pada gambar 3.15



Gambar 3.15 *sequence diagram* ubah data koneksi modem

3.2.2.1.3.2.3 Hapus Data Koneksi Modem

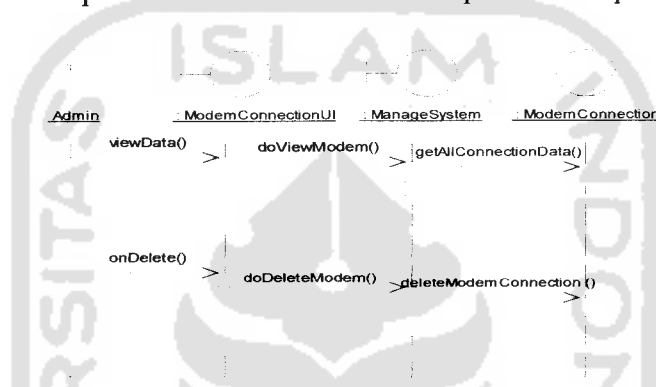
Objek yang berkaitan dengan *sequence* ini adalah sebagai berikut :

- | | |
|----------------|---------------------|
| Aktor | : Admin, Paramedis |
| Class Boundary | : ModemConnectionUI |
| Class Control | : ManageData |
| Class Entity | : ModemConnection |
- Keterangan :
1. Admin mengawali *sequence* ini dengan memanggil *method* `viewData()`.
 2. ModemConnectionUI melakukan instansiasi ke ManageData dan memanggil *method* `doViewModem()` untuk memerintahkan agar semua data modem ditampilkan
 3. ManageData melakukan instansiasi ke ModemConnection dan memanggil *method* `getAllModem()` untuk mengambil semua data

modem.

4. Admin kemudian memanggil *method* `doDelete()`...
5. ModemConnectionUI melakukan instansiasi ke ManageData dan memanggil *method* `doDelete()` untuk melakukan proses penghapusan.
6. ManageData melakukan instansiasi ke ModemConnection dan memanggil *method* `deleteModem()` untuk menghapus *record* data modem.

Sequence diagram hapus data koneksi modem dapat dilihat pada gambar 3.16



Gambar 3.16 *sequence diagram* hapus data koneksi modem

3.2.2.1.3.2.4 Ubah Data SMS

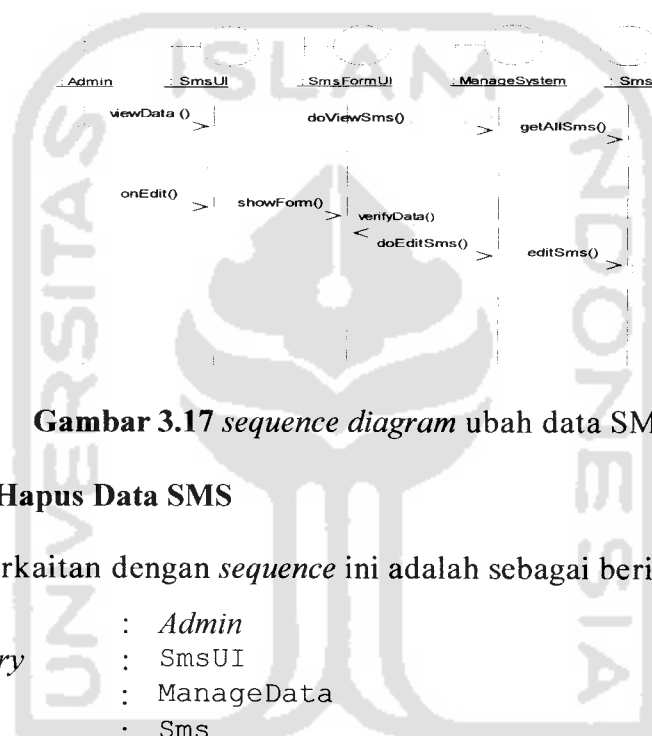
Objek yang berkaitan dengan *sequence* ini adalah sebagai berikut :

- | | | |
|----------------|---|---|
| Aktor | : | <i>Admin</i> |
| Class Boundary | : | <i>SmsUI, SmsFormUI</i> |
| Class Control | : | <i>ManageData</i> |
| Class Entity | : | <i>Sms</i> |
| Keterangan | : | <ol style="list-style-type: none"> 1. <i>Admin</i> mengawali <i>sequence</i> ini dengan memanggil <i>method</i> <code>viewData()</code>. 2. <u>SmsUI</u> melakukan instansiasi ke <u>ManageData</u> dan memanggil <i>method</i> <code>doViewSms()</code> untuk memerintahkan agar semua data SMS ditampilkan 3. <u>ManageData</u> melakukan instansiasi ke <u>Sms</u> dan memanggil <i>method</i> <code>getAllSms()</code> untuk mengambil semua data <i>SMS</i>. 4. Admin kemudian memanggil <i>method</i> <code>doEdit()</code>. 5. <u>SmsUI</u> melakukan instansiasi ke <u>SmsFormUI</u> dan |

memanggil *method* `showForm()` untuk menampilkan form.

6. SmsFormUI melakukan instansiasi ke ManageData dan memanggil *method* `doEditSms()` untuk memproses data yang telah diubah.
7. ManageData melakukan instansiasi ke Sms dan memanggil *method* `editSms()` untuk mengubah *record* data SMS.

Sequence diagram ubah data SMS dapat dilihat pada gambar 3.17



Gambar 3.17 *sequence diagram* ubah data SMS

3.2.2.1.3.2.5 Hapus Data SMS

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

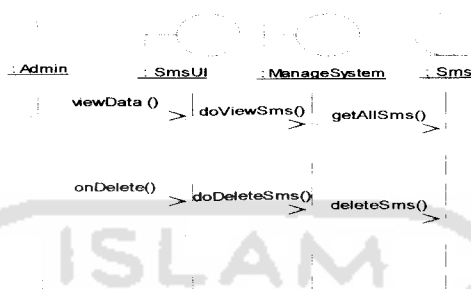
Aktor : Admin
 Class Boundary : SmsUI
 Class Control : ManageData
 Class Entity : Sms

- Keterangan :
1. Admin mengawali *sequence* ini dengan memanggil *method* `viewData()`.
 2. SmsUI melakukan instansiasi ke ManageData dan memanggil *method* `doViewSms()` untuk memerintahkan agar semua data SMS ditampilkan
 3. ManageData melakukan instansiasi ke Sms dan memanggil *method* `getAllSms()` untuk mengambil semua data SMS.
 4. Admin kemudian memanggil *method* `doDelete()`.
 5. SmsUI melakukan instansiasi ke ManageData dan memanggil *method* `doDelete()` untuk melakukan

proses penghapusan.

6. ManageData melakukan instansiasi ke Sms dan memanggil *method* `deleteSms()` untuk menghapus *record* data SMS.

Sequence diagram hapus data SMS dapat dilihat pada gambar 3.18



Gambar 3.18 *sequence diagram* hapus data SMS

3.2.2.1.3.3 Modul *User Management*

Modul *User Management* adalah modul pada sistem yang fungsinya adalah untuk menangani pengelolaan pengguna sistem. Berikut ini akan diuraikan mengenai semua *sequence diagram* dari modul *User Management*,

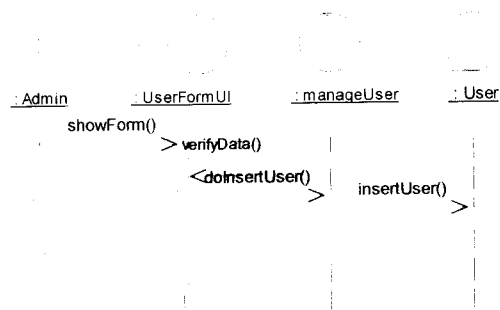
3.2.2.1.3.3.1: Tambah Data User

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

Aktor	: <i>Admin</i>
Class Boundary	: <i>UserFormUI</i>
Class Control	: <i>ManageUser</i>
Class Entity	: <i>User</i>
Keterangan	:

1. *Admin* mengawali *sequence* ini dengan memanggil *method* `showForm()`.
2. *UserFormUI* melakukan instansiasi ke *ManageUser* dan memanggil *method* `doInsertUser()` untuk memproses *input* data dari *admin*.
3. *ManageUser* melakukan instansiasi ke *User* dan memanggil *method* `insertUser()` untuk memasukkan data ke basis data tabel *user*.

Sequence diagram tambah user dapat dilihat pada gambar 3.19



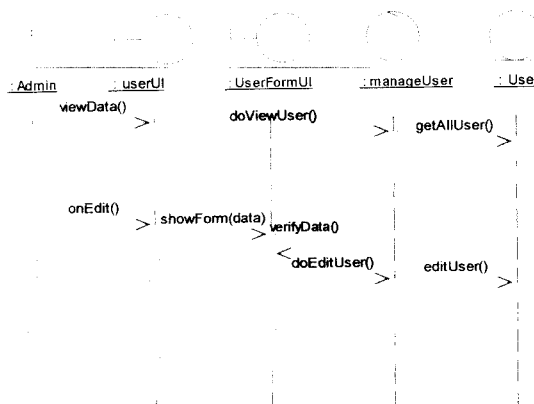
Gambar 3.19 *sequence diagram* tambah user

3.2.2.1.3.3.2 Ubah Data User

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

- | | | |
|----------------|---|--|
| Actor | : | <i>Admin</i> |
| Class Boundary | : | <i>UserUI, UserFormUI</i> |
| Class Control | : | <i>ManageUser</i> |
| Class Entity | : | <i>user</i> |
| Keterangan | : | <ol style="list-style-type: none"> 1. <i>ADMIN</i> mengawali <i>sequence</i> ini dengan memanggil <i>method</i> <i>viewData()</i>. 2. <i>LoginUI</i> melakukan instansiasi ke <i>ManageUser</i> dan memanggil <i>method</i> <i>doViewUser()</i> untuk memerintahkan agar semua data <i>user</i> ditampilkan 3. <i>ManageUser</i> melakukan instansiasi ke <i>User</i> dan memanggil <i>method</i> <i>getAllUser()</i> untuk mengambil semua data <i>user</i>. 4. Admin kemudian memanggil <i>method</i> <i>doEdit()</i>. 5. <i>UserUI</i> melakukan instansiasi ke <i>UserFormUI</i> dan memanggil <i>method</i> <i>showForm()</i> untuk menampilkan form. 6. <i>UserFormUI</i> melakukan instansiasi ke <i>ManageUser</i> dan memanggil <i>method</i> <i>doEditUser()</i> untuk memproses data yang telah diubah. 7. <i>ManageUser</i> melakukan instansiasi ke <i>User</i> dan memanggil <i>method</i> <i>editUser()</i> untuk mengubah <i>record</i> data <i>user</i>. |

Sequence diagram ubah data user dapat dilihat pada gambar 3.20



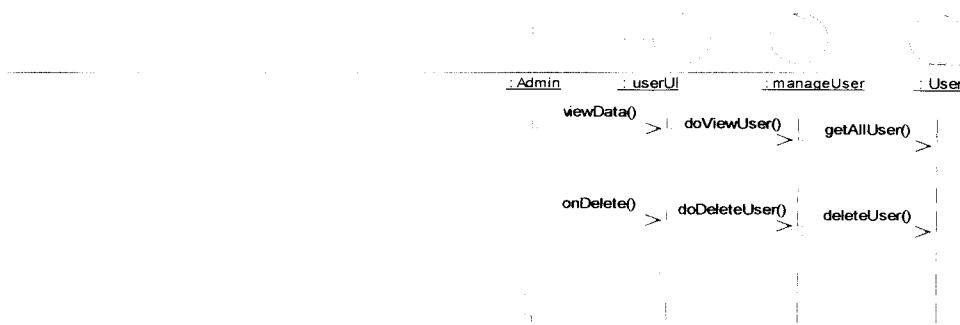
Gambar 3.20 *sequence diagram* ubah user

3.2.2.1.3.3.3 Hapus Data User

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

- | | | |
|----------------|---|--|
| Aktor | : | <i>Admin</i> |
| Class Boundary | : | <i>UserUI</i> |
| Class Control | : | <i>ManageData</i> |
| Class Entity | : | <i>User</i> |
| Keterangan | : | <ol style="list-style-type: none"> 1. <i>Admin</i> mengawali <i>sequence</i> ini dengan memanggil <i>method</i> <i>viewData()</i>. 2. <i>UserUI</i> melakukan instansiasi ke <i>ManageUser</i> dan memanggil <i>method</i> <i>doViewUser()</i> untuk memerintahkan agar semua data <i>user</i> ditampilkan 3. <i>ManageUser</i> melakukan instansiasi ke <i>User</i> dan memanggil <i>method</i> <i>getAlluser()</i> untuk mengambil semua data <i>user</i>. 4. <i>Admin</i> kemudian memanggil <i>method</i> <i>doDelete()</i>. 5. <i>UserUI</i> melakukan instansiasi ke <i>ManageUser</i> dan memanggil <i>method</i> <i>doDelete()</i> untuk melakukan proses penghapusan. 6. <i>ManageUser</i> melakukan instansiasi ke <i>User</i> dan memanggil <i>method</i> <i>deleteUser()</i> untuk menghapus <i>record</i> data <i>user</i>. |

Sequence diagram hapus data *user* dapat dilihat pada gambar 3.21



Gambar 3.21 *sequence diagram* hapus user

3.2.2.1.3.3.4 Login

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

Aktor : *Admin, Paramedis*

Class Boundary : *LoginFormUI*

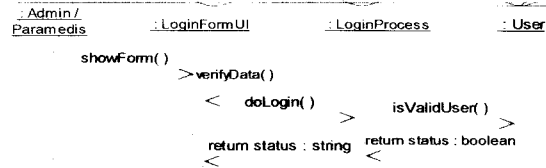
Class Control : *LoginProcess*

Class Entity : *User*

Keterangan :

1. *User* mengawali *sequence* ini dengan memanggil *method ShowForm()*.
2. *LoginFormUI* melakukan instansiasi ke *LoginProcess* dan memanggil *method doLogin()* untuk memerintahkan agar *username* dan *password* diperiksa kebenarannya.
3. *LoginProcess* melakukan instansiasi ke *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 3.22



Gambar 3.22 *sequence diagram login*

3.2.2.1.4 Activity Diagram

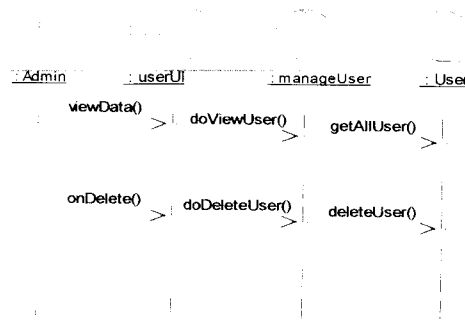
Activity diagram merupakan diagram yang menggambarkan alur kerja dari sistem dan diagram ini merupakan suatu diagram dinamis yang menunjukkan aktivitas beserta kejadian yang menyebabkan suatu objek berada dalam *state* tertentu. Diagram ini lebih menggambarkan transisi-transisi dan aktivitas-aktivitas yang menyebabkan perubahan pada *states* objek. Simbol lingkaran berisi warna hitam menandakan awal *state* sedangkan simbol lingkaran berisi warna hitam yang dilingkari oleh lingkaran bergaris hitam menandakan akhir *state*.

Pada bagian ini akan dijelaskan tentang aktivitas yang dilakukan oleh *user* (*admin*, *paramedis*) secara umum dalam pengolahan data. Kegiatan-kegiatan yang dilakukan *user* tersebut meliputi kegiatan tambah data, ubah data dan hapus 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* (*Admin*, *Paramedis*) memilih menu *insert*.



Gambar 3.21 *sequence diagram* hapus user

3.2.2.1.3.3.4 Login

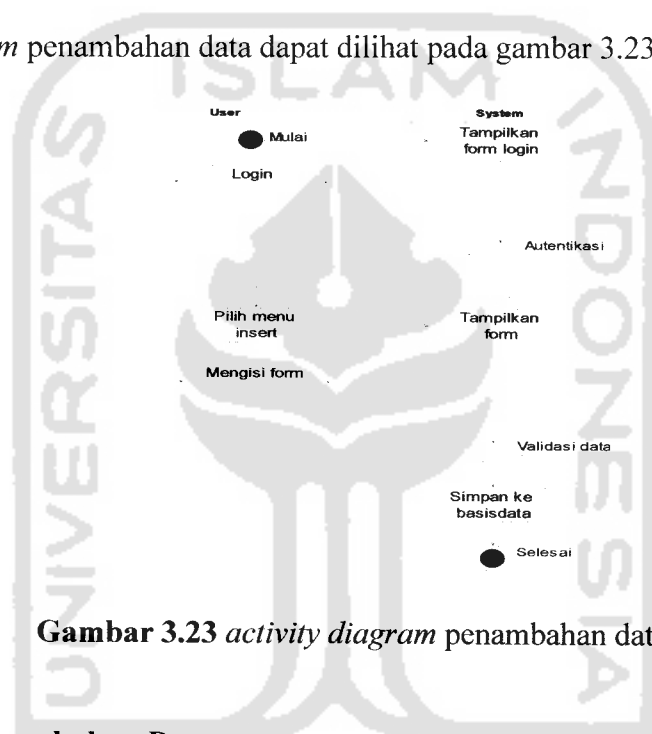
Objek yang berkaitan dengan *sequence* ini adalah sebagai berikut :

- | | |
|----------------|--|
| Aktor | : Admin, Paramedis |
| Class Boundary | : LoginFormUI |
| Class Control | : LoginProcess |
| Class Entity | : User |
| Keterangan | : 1. <u>User</u> mengawali <i>sequence</i> ini dengan memanggil <i>method</i> ShowForm().
2. <u>LoginFormUI</u> melakukan instansiasi ke <u>LoginProcess</u> dan memanggil <i>method</i> doLogin() untuk memerintahkan agar <i>username</i> dan <i>password</i> diperiksa kebenarannya.
3. <u>LoginProcess</u> melakukan instansiasi ke <u>User</u> dan memanggil <i>method</i> isValidUser() untuk memeriksa kebenaran dari <i>username</i> dan <i>password</i> yang dimasukkan.
4. <u>User</u> menanggapi dengan mengembalikan status, tipe Boolean, ke <u>LoginProcess</u> untuk memberitahukan status bahwa <i>username</i> dan <i>password</i> yang dimasukkan benar/salah.
5. <u>LoginProcess</u> mengembalikan status, tipe String, ke <u>FormUI</u> untuk ditampilkan ke <i>user</i> bahwa berhasil/gagal login ke sistem. |

Sequence diagram Login dapat dilihat pada gambar 3.22

2. Sistem menampilkan form untuk pengisian data.
3. Selanjutnya *user* mengisikan data pada form dan menekan tombol *done* untuk mengirimkan data
4. Sistem melakukan proses validasi, jika valid penyimpanan data akan dilakukan dan sebaliknya jika tidak valid akan muncul pesan kesalahan.
5. Kemudian sistem mengakhiri kegiatan ini.

Activity diagram penambahan data dapat dilihat pada gambar 3.23 :



Gambar 3.23 *activity diagram* penambahan data

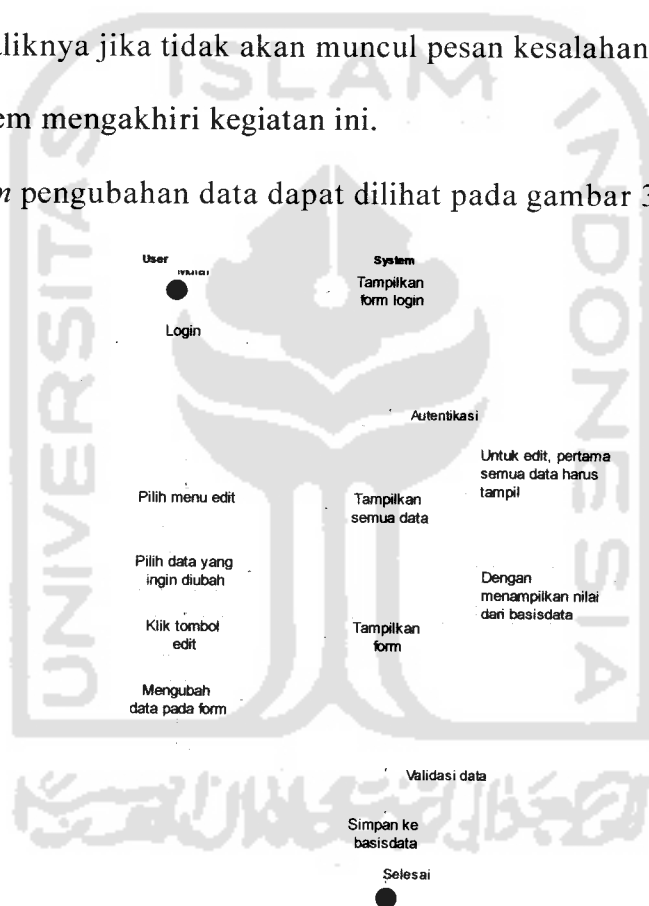
3.2.2.1.4.2 Perubahan Data

Merupakan kegiatan yang berfungsi untuk mengubah data pada basis data, dengan urutan aktivitas sebagai berikut :

1. *User* (*Admin*, *Paramedis*) memilih menu *edit*.
2. Selanjutnya sistem menampilkan semua data dari tabel tertentu.
3. *User* memilih data yang ingin diubah.

4. *User* menekan tombol *edit*.
5. Sistem menampilkan form dengan kolom-kolom yang terisi data dari basis data.
6. Selanjutnya *user* mengubah data yang dikehendaki dan menekan tombol *done* untuk mengirimkan data.
7. Sistem melakukan proses validasi, jika valid data akan disimpan dan sebaliknya jika tidak akan muncul pesan kesalahan.
8. Sistem mengakhiri kegiatan ini.

Activity diagram perubahan data dapat dilihat pada gambar 3.24 :



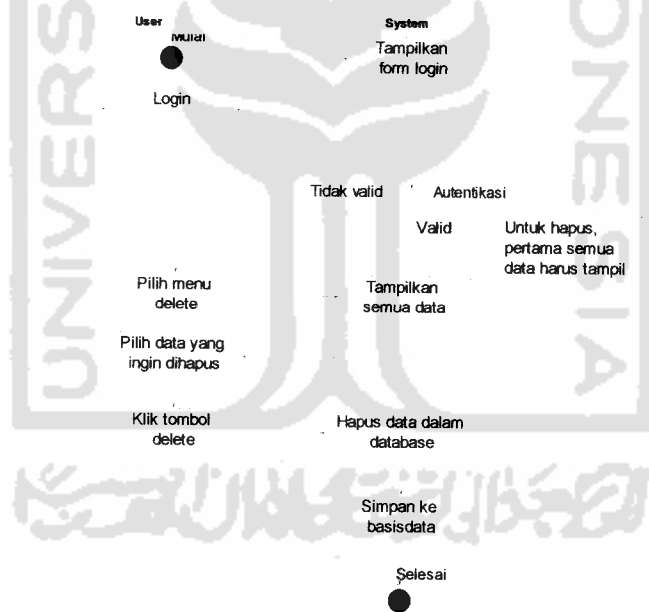
Gambar 3.24 *activity diagram* perubahan data

3.2.2.1.4.3 Penghapusan Data

Merupakan kegiatan dalam modul *data management* yang berfungsi menghapus data pada basis data, dengan urutan aksi sebagai berikut :

1. *User (Admin, Paramedis)* memilih menu *delete*.
2. Sistem menampilkan semua data dari tabel tertentu.
3. *User* memilih data pasien yang ingin dihapus.
4. *User* menekan tombol *delete*.
5. Sistem melakukan proses penghapusan data dalam basis data.
6. Sistem mengakhiri kegiatan ini.

Activity diagram hapus data dapat dilihat pada gambar 3.25 :



Gambar 3.25 *activity diagram* penghapusan data

3.2.2.2 Perancangan Basis Data

Basis data adalah salah satu komponen penting dalam suatu aplikasi perangkat lunak, karena mempunyai fungsi untuk mengolah dan memanipulasi data menjadi informasi. Sifat basis data bertipe relasional yang terdiri dari beberapa tabel yang saling terhubung dan dapat melakukan relasi.

Berikut ini akan dijelaskan tentang detail tabel yang digunakan pada sistem pengingat untuk tindakan terapi *insulin* bagi penderita *dabetes mellitus*.

3.2.2.2.1 Struktur Tabel

1) Tabel koneksi_data

Tabel koneksi data digunakan untuk menyimpan data-data properti 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, <i>Primary Key</i>
Port	Varchar	20	(PK) a port yang digunakan
Bps	Integer	11	Nilai baud rate
Data_bits	Integer	1	Nilai data bits
Parity	Varchar	20	Nilai parity
Stop_bits	Double	3	Nilai stop bits
Flow_kontrol	Varchar	20	Nilai flow control

2) Tabel koneksi_log

Tabel koneksi_log digunakan untuk mencatat seluruh transaksi tentang koneksi (modem), 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
Waktu	Datetime		Nama koneksi, <i>Foreign Key</i>
Nama	Varchar	50	Nama port yang digunakan
Status	Integer	1	Nilai baud rate

3) Tabel pasien_data

Tabel data_pasien digunakan untuk menyimpan data-data pribadi Pasien, terdiri dari nama pasien, alamat, tanggal lahir, status pendidikan dan pekerjaan. Adapun untuk struktur tabel dari data_pasien dapat dilihat pada

Tabel 3.3 Tabel pasien_data

Nama Field	Tipe Data	Panjang	Keterangan
Id_pasien	Varchar	10	Identitas pasien (PK)
Nama	Varchar	50	Nama pasien
Tgl_lahir	Date		Tanggal lahir pasien
Alamat	Varchar	50	Alamat tinggal pasien
No_telp	Varchar	20	No telp pasien
Hp	Varchar	20	No. <i>hand phone</i> pasien
Pendidikan	Varchar	30	Status pendidikan pasien
Pekerjaan	Varchar	50	Pekerjaan pasien

4) Tabel pasien_terapi

Tabel pasien_habit berfungsi untuk menyimpan data pola makan pasien, terdiri dari waktu makan pagi, makan siang, makan malam dan waktu tidur.

Untuk struktur tabel dari pasien_habit dapat dilihat pada tabel 3.4

Tabel 3.4 Tabel pasien_terapi

Nama Field	Tipe Data	Panjang	Keterangan
Id_pasien	Varchar	10	Identitas pasien (FK)
Tipe_regimen	Varchar	1	Jenis terapi pasien
Terapi_1	Time		Waktu terapi pertama
Terapi_2	Time		Waktu terapi kedua

Terapi_3	Time		Waktu terapi ketiga
Terapi_4	Time		Waktu terapi keempat

5) Tabel pasien_auth

Tabel pasien_auth digunakan untuk menyimpan data autentikasi pasien dan status keanggotaan pasien Untuk struktur tabel dari pasien_auth dapat dilihat pada tabel 3.5

Tabel 3.5 Tabel patient_auth

Nama Field	Tipe Data	Panjang	Keterangan
Id_pasien	Varchar	10	Identitas pasien (FK)
Password	Varchar	255	Password pasien
Status	Varchar	1	Status keanggotaan pasien

6) Tabel reminder_log

Tabel reminder_log digunakan untuk mencatat aktivitas pengiriman SMS pengingat. Untuk struktur tabel dari reminder log dapat dilihat pada tabel 3.6

Tabel 3.6 Tabel reminder_log

Nama Field	Tipe Data	Panjang	Keterangan
Id_pasien	Varchar	10	Identitas pasien (FK)
1	Varchar	1	Terapi pertama
2	Varchar	1	Terapi kedua
3	Varchar	1	Terapi ketiga
4	Varchar	1	Terapi keempat
Today	Datetime		Tanggal hari ini

7) Tabel rekam_medis

Tabel rekaman_medis digunakan untuk menyimpan data-data rekaman medis Pasien tentang perkembangan kesehatannya. Adapun untuk struktur tabel dari Data_pasien dapat dilihat pada tabel 3.7

Tabel 3.7 Tabel rekam_medis

Nama Field	Tipe Data	Panjang	Keterangan
No_RM	Varchar	10	Nomer data rekaman medis
Id_pasien	Varchar	10	Identitas pasien, <i>foreign key</i> (FK)
Berat_badan	Integer	3	Berat badan dalam kilo gram (kg)
Glukosa_puasa	Varchar	4	Kadar gula darah pasien saat puasa
Glukosa_2jam	Varchar	4	Kadar gula darah 2 jam sebelum / sesudah makan
Tekanan_darah	Varchar	3	Tekanan darah pasien
Kolesterol	Varchar	3	

8) Tabel obat

Tabel obat digunakan untuk menyimpan tentang detail obat *insulin*. Adapun untuk struktur tabel obat dapat dilihat pada tabel 3.8

Tabel 3.8 Tabel obat

Nama Field	Tipe Data	Panjang	Keterangan
Kode_obat	Varchar	8	Kode obat insulin
Nama	Varchar	50	Nama <i>merk</i> dagang
Vandor	Varchar	50	Produsen obat
Efek_puncak	Integer	2	Efek puncak
Lama_kerja	Integer	2	Lama kerja
Netto	Integer	10	Berat bersih

9) Tabel inbox

Tabel inbox digunakan untuk menyimpan *SMS* yang masuk. Untuk struktur tabel dari inbox dapat dilihat pada tabel 3.9

Tabel 3.9 Tabel inbox

Nama Field	Tipe Data	Panjang	Keterangan
No_inbox	Varchar	50	
No_telp	Varchar	20	Nomer telepon pengirim
Teks	Text		Format teks SMS
Waktu	Datetime	50	Waktu penerimaan SMS

10) Tabel outbox

Tabel outbox digunakan untuk menyimpan SMS yang keluar. Untuk struktur tabel dari outbox dapat dilihat pada tabel 3.10

Tabel 3.10 Tabel outbox

Nama Field	Tipe Data	Panjang	Keterangan
No_inbox	Varchar	50	
No_telp	Varchar	20	Nomer telepon penerima
Teks	Text		Format teks SMS
Waktu	Datetime	50	Wakut pengiriman SMS
Status	Varchar	1	Status SMS

11) Tabel Admin_data

Tabel admin_data digunakan untuk menyimpan data para *administrators*, yang terdiri dari admin sebagai *super admin* dan paramedis sebagai *operator*.

Struktur tabel dari admin_data dapat dilihat pada tabel 3.11

Tabel 3.11 Tabel admin_data

Nama Field	Tipe Data	Panjang	Keterangan
User_id	Varchar	10	Identitas user
Nama	Varchar	50	Nama user
Alamat	Varchar	50	Alamat tinggal user
Email	Varchar	50	Email user
No_telp	Varchar	20	No telp yang bisa dihubungi
Pekerjaan	Varvhar	50	Jabatan user

12) Tabel admin_auth

Tabel admin_auth digunakan untuk menyimpan data autentikasi *admin*, yang terdiri dari *username*, *password* dan hak akses. Struktur tabel dari tabel admin_auth

dapat dilihat pada tabel 3.12

Tabel 3.12 Tabel admin_auth

Nama Field	Tipe Data	Panjang	Keterangan
User_id	Varchar	50	Identitas admn (PK)
Password	Varchar	255	Password admin
Privilege	Varchar	1	Hak akses

13) Tabel time_out

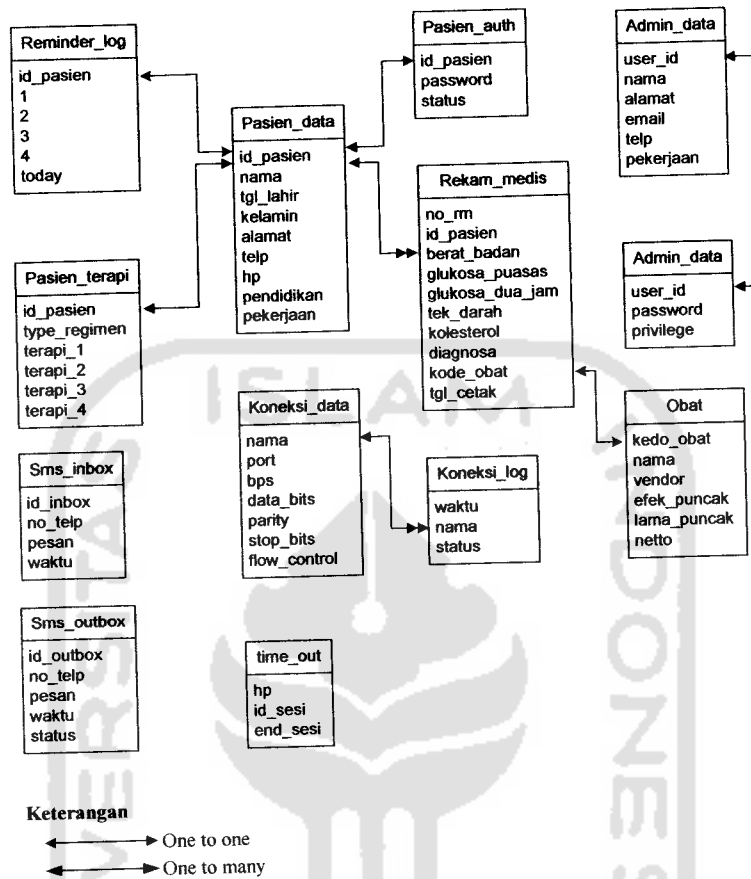
Tabel time_out digunakan untuk menyimpan data id user (no hp), waktu time out (akhir) sesi dan id sesi. Struktur tabel dari time out dapat dilihat pada tabel 3.13

Tabel 3.13 Tabel time_out

Nama Field	Tipe Data	Panjang	Keterangan
Hp	Varchar	20	No. hand phone user (PK)
Id_sesi	Varchar	1	Jenis sesi
End_sesi	Datetime		Waktu akhir sesi

3.2.2.2 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) yang merupakan kunci utama dari masing-masing file atau tabel. Keberadaan tabel relasi dapat memudahkan dalam pemeliharaan data dan menghindari *redudancy* (kerangkapan data), sehingga informasi yang diperoleh menjadi lebih akurat dan meminimalkan pembuatan *record*. Untuk Relasi antar tabel pada sitem untuk pengingat untuk tindakan terapi insulin bagi penderita *diabetes mellitus* dapat dilihat pada gambar 3.26



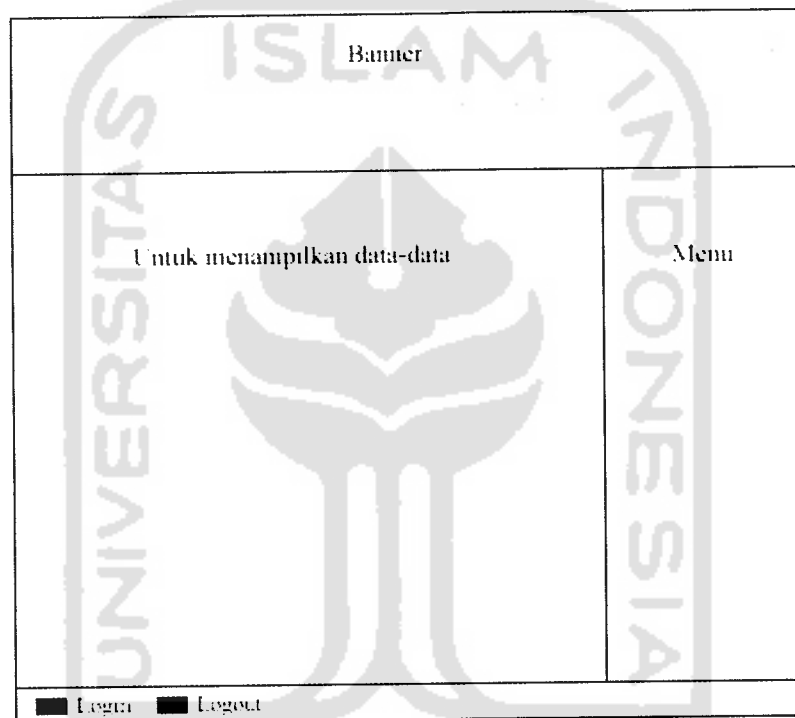
Gambar 3.26 relasi tabel basis data

3.2.2.3 Perancangan Antarmuka

Rancangan antarmuka untuk sistem utama atau halaman *admin* dari aplikasi pengingat untuk tindakan terapi *insulin* bagi penderita *diabetes mellitus* menggunakan NET BEANS 5.5.

3.2.2.3.1 Perancangan Halaman Utama

Pada sistem pengingat untuk tindakan terapi insulin bagi penderita *dibetes mellitus* ini dalam pembuatan tampilannya terbagi dalam beberapa bagian, bagian atas sebagai peletakkan *banner*, kolom kanan sebagai bagian menu dan bagian kiri sebagai tempat untuk menampilkan data untuk diolah lebih lanjut. Untuk rancangan antarmuka halaman utama dapat dilihat pada gambar 3.27



Gambar 3.27 rancangan halaman utama

3.2.2.3.2 Perancangan Input

Input (masukan) merupakan awal dimulainya proses informasi. Masukan untuk sistem informasi adalah data yang merupakan bahan mentah dari informasi. Perancangan masukan adalah rancangan dari *form* yang digunakan untuk menangkap

data (*input*), kode-kode *input* yang digunakan dan bentuk dari tampilan *input*.

Adapun rancangan *input* dialog untuk sistem ini adalah sebagai berikut :

3.2.2.3.2.1 Perancangan Input Data Pasien

Perancangan antarmuka digunakan untuk mempermudah dalam pemasukan, perubahan, penghapusan dan penyimpanan data. Data masukan pasien terdiri dari id pasien, nama, tanggal lahir, jenis kelamin, alamat, nomer telepon, nomer *hand phone* pendidikan dan pekerjaan. Rancangan *input* data pasien dapat dilihat pada gambar 3.28

The image shows a screenshot of a patient data input form. The form contains the following fields and controls:

- Nama pasien:** A text input field.
- Tel. lhu:** A text input field.
- Jenis kelamin:** Radio buttons for "Laki-laki" (selected) and "Perempuan".
- Alamat:** A text input field.
- No. telf:** A text input field.
- No. hand phone:** A text input field.
- Pendidikan:** A text input field with a dropdown arrow, and a "Lainnya:" label with a text input field.
- Pekerjaan:** A text input field with a dropdown arrow, and a "Lainnya:" label with a text input field.

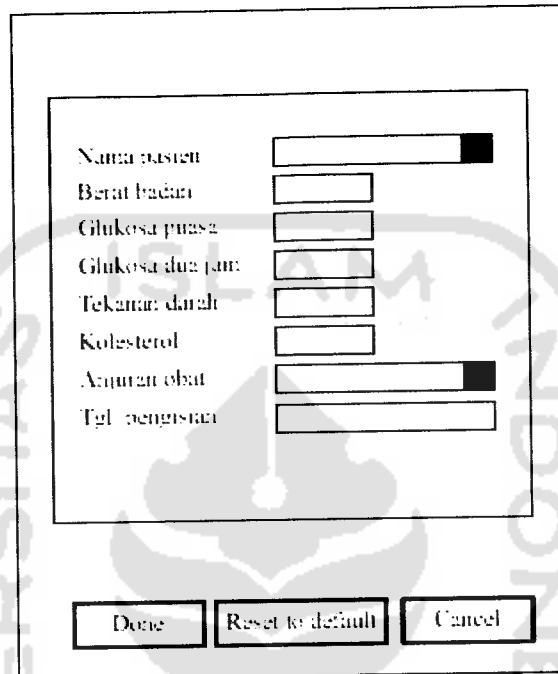
At the bottom of the form, there are three buttons: "Done", "Reset to default", and "Cancel".

Gambar 3.28 rancangan input data pasien

3.2.2.3.2.2 Perancangan Input Data Rekam Medis

Antarmuka *input* data rekam medis, digunakan untuk memasukan data-data kesehatan pasien. Pada antarmuka ini terdiri dari beberapa *input*, diantaranya : nama

pasien, berat badan, glukosa puasa, glukosa dua jam, tekanan darah, kolesterol, anjuran obat dan tanggal pengisian. Rancangan *input* data rekam medis dapat dilihat pada gambar 3.29



The image shows a screenshot of a medical record input form. The form is enclosed in a rectangular border and contains the following fields and buttons:

- Nama pasien:
- Berat badan:
- Glukosa puasa:
- Glukosa dua jam:
- Tekanan darah:
- Kolesterol:
- Anjuran obat:
- Tgl pengisian:

At the bottom of the form, there are three buttons: "Done", "Reset to default", and "Cancel".

Gambar 3.29 rancangan input rekam medis

3.2.2.3.2.3 Perancangan Input Data Obat

Antarmuka *input* obat digunakan untuk memasukan data-data tentang deskripsi obat, antara lain : nama obat, produsen, efek puncak, lama puncak dan *netto*. Untuk rancangan *input* data obat dapat dilihat pada gambar 3.30

A screenshot of a software dialog box for modem configuration. It contains five text input fields labeled 'Nama modem', 'Vendor', 'Efek bunyi', 'Nama port', and 'Netto'. At the bottom, there are three buttons: 'Done', 'Reset to default', and 'Cancel'.

Gambar 3.30 rancangan input obat

3.2.2.3.2.4 Perancangan Input Modem

Antarmuka *input* data modem digunakan untuk memasukan data properti *modem*. Pada antarmuka ini terdiri dari beberapa *input*, antara lain : nama koneksi, nama port dan data-data properti *modem*. Untuk rancangan *input* data modem dapat dilihat pada gambar 3.31 dan gambar 3.32

A screenshot of a software dialog box for modem configuration. It contains two text input fields labeled 'Nama koneksi' and 'Port'. At the bottom, there are three buttons: 'Back', 'Next', and 'Cancel'.

Gambar 3.31 rancangan input modem panel pertama

Bit per Second

Data bits

Parity

Stop bits

Flow control

Back Done Cancel

Gambar 3.32 rancangan input modem panel kedua

3.2.2.3.2.5 Perancangan Input Data User

Antarmuka input *user* digunakan untuk membuat *user* baru. Pada antarmuka ini terdapat beberapa *input* antara lain : *user* id, password, nama, *privilege*, alamat, email, nomer telepon dan pekerjaan. Untuk rancangan antarmuka input *user* dapat dilihat pada gambar 3.33

User id

Password

Re-password

Nama

Hak akses

Alamat

Email

No. Tele

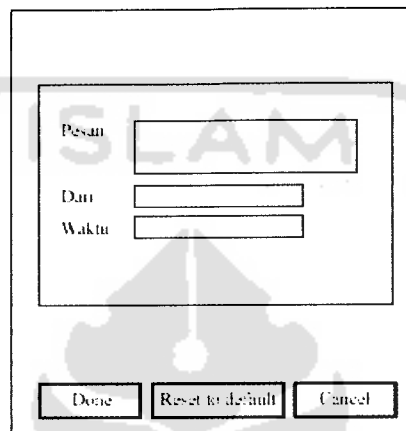
Pekerjaan

Done Reset to default Cancel

Gambar 3.33 rancangan input *user*

3.2.2.3.2.6 Perancangan Ubah Data Pesan Masuk (*Inbox*)

Antarmuka ini digunakan untuk mengubah pesan masuk yang telah tersimpan pada basis data. Terdapat beberapa *input* pada antarmuka ini, antara lain : pesan, nomor pengirim, waktu pengiriman. Untuk rancangan antarmuka input ubah pesan masuk dapat dilihat pada gambar 3.34

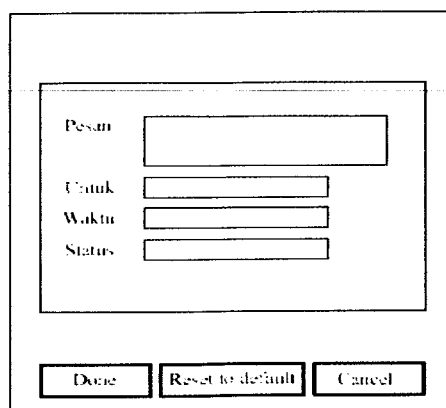


The image shows a screenshot of a web form for editing an incoming message. The form is enclosed in a rectangular border. Inside, there are three input fields stacked vertically. The first field is labeled 'Pesan' and contains the text 'SLAM'. The second field is labeled 'Dari' and is empty. The third field is labeled 'Waktu' and is empty. Below the input fields, there are three buttons: 'Done', 'Reset ke default', and 'Cancel'.

Gambar 3.34 rancangan ubah pesan masuk

3.2.2.3.2.7 Perancangan Ubah Pesan Keluar (*Outbox*)

Antarmuka ubah pesan keluar digunakan untuk mengubah pesan keluar. Terdapat beberapa kesamaan *input* data dengan antarmuka ubah pesan masuk, yang membedakan adalah adanya status pengiriman pada antarmuka ubah pesan keluar. Untuk antarmuka ini terdapat beberapa *input* antara lain : pesan, nomor tujuan, waktu pengiriman dan status. Rancangan antarmuka ubah pesan keluar dapat dilihat pada gambar 3.35



Pesan

Untuk

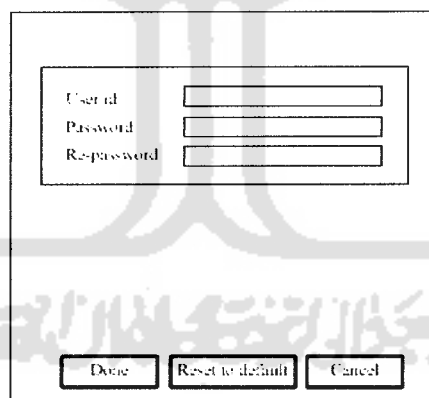
Waktu

Status

Gambar 3.35 rancangan ubah pesan keluar

3.2.2.3.2.8 Perancangan Ubah Password

Antarmuka ubah password digunakan untuk mengubah password *user* (*admin* dan *paramedis*). Terdapat tiga *input* pada antarmuka ini, antara lain : password lama, password baru dan *re-type* password. Adapun untuk rancangan antarmuka ubah password dapat dilihat pada gambar 3.36



User id

Password

Re-password

Gambar 3.36 rancangan input ubah password

3.2.2.3.2.9 Perancangan Pencarian Data

Antarmuka ini digunakan untuk melakukan pencarian terhadap data-data yang terdapat pada sistem utama. Pada antarmuka ini terdapat beberapa *input*, yaitu *keyword* (kata kunci), nama *field* (kolom), arah cursor dan jenis pencarian. Adapun untuk rancangan antarmuka pencarian dapat dilihat pada gambar 3.37

The image shows a search dialog box with the following elements:

- Two input fields for "Find what".
- A "Direction" section with two radio buttons: "up" (selected) and "down".
- A checkbox labeled "match word".
- Four buttons: "Find next", "Reset to default", and "Cancel".

Gambar 3.37 rancangan input pencarian

3.2.2.4 Perancangan Format SMS

Tujuan dari ditetapkannya format pengiriman SMS, yaitu agar *input* yang dikirimkan oleh *user* dapat dipastikan ke-*valid*-annya sebelum disimpan ke dalam basis data. Adapun rancangan *input* dialog untuk sistem ini adalah sebagai berikut :

3.2.2.4.1 Perancangan Format Panduan

Panduan berfungsi untuk menampilkan bantuan tentang penggunaan perintah-perintah SMS yang terdapat pada dari sistem. Perintah-perintah tersebut terdiri dari pendaftaran, ubah data kesehatan, ubah pola makan, ubah password dan jadwal terapi. Adapun untuk format penulisan dari masing-masing perintah adalah sebagai berikut :

- Daftar semua perintah : "PAN" atau "?"

- Pendaftaran : "REG?"
- Ubah data kesehatan : "RM?"
- Jadwal terapi : "JDW"
- Ubah pola makan : "PM?"
- Ubah password : "REG <id pasien>"

3.2.2.4.2 Perancangan Format Pendaftaran

Format ini digunakan untuk melakukan proses pendaftaran setelah *user* (pasien) memperoleh nomer id pasien. *Input* data yang harus dikirimkan , yaitu id pasien. Adapun rancangan format pendaftaran adalah "REG <id pasien>".

3.2.2.4.3 Perancangan Format Ubah Data Kesehatan

Format ini digunakan untuk mengubah data kesehatan pasien seperti gula darah dan berat badan. Pada pengisian gula darah dan berat badan harus menggunakan tanda "=", sebab tingkat kesulitan dapat menghasilkan ketelitian terhadap suatu hal. Adapun rancangan format ubah data kesehatah adalah "RM <password> <GD=gula darah> <BB=berat badan>".

3.2.2.4.4 Perancangan Format Ubah Pola Makan.

Pola makan adalah waktu makan pasien, misalnya waktu makan pagi, waktu makan siang dan waktu makan malam. Keberadaan dari variabel ini sangat penting, sebab dari variabel ini digunakan untuk menentukan kapan SMS *reminder* (peringat) dikirimkan. Adapun untuk rancangan pola makan adalah "PM <password>".

3.2.2.4.5 Perancangan Format Ubah Password.

Format ini digunakan untuk mengubah password *user*, terdapat beberapa input yang harus dikirimkan, yaitu password lama dan password baru. Adapun untuk format ubah password adalah "CP <password lama> <password baru>".

3.2.2.4.6 Perancangan Format Jadwal Terapi.

Format ini berisi perintah permintaan jadwal terapi *user*, adapun untuk format jadwal terapi adalah "JDW"

3.2.2.4.7 Perancangan Format Waktu Terapi.

Perintah pengisian waktu pola makan aktif setelah *user* melakukan proses pendaftaran atau ubah data pola makan. Untuk pola makan dari masing-masing pasien tidak sama, tergantung dengan tipe regimen (jenis terapi). Berikut ini adalah contoh format waktu pola makan dengan tipe regimen 2, yaitu "makan pagi(hh:mm)#makan malam(hh:mm)".

3.3 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.

3.3.1 Batasan Implementasi

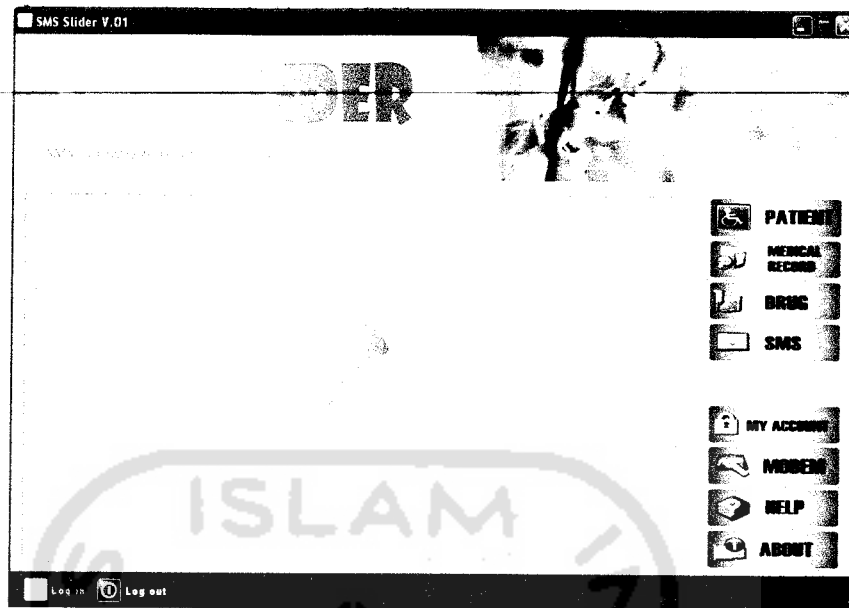
Sistem pengingat untuk tindakan terapi insulin bagi penderita *dibetes mellitus* ini, dalam implementasinya dibatasi pada *insert* (tambah), *update* (ubah) dan *delete* (hapus) data. Data tersebut meliputi data pasien, data rekam medis, data obat, data *user* dan data koneksi modem.

3.3.2 Implementasi antarmuka

Implementasi dari aplikasi Sistem pengingat untuk tindakan terapi insulin bagi penderita *dibetes mellitus* ini terdiri dari beberapa form dengan masing-masing fungsinya.

3.3.2.1 Halaman Utama

Halaman ini akan muncul pertama kali setelah sistem berhasil dijalankan. Terdapat beberapa menu pada halaman ini, antara lain : menu pasien, rekam medis, obat (*insulin*), *user*, modem, about dan bantuan. Adapun untuk tampilannya dapat dilihat pada gambar 3.38

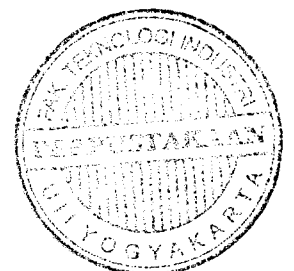


Gambar 3.38 tampilan halaman utama

3.3.2.2 Form Login Admin

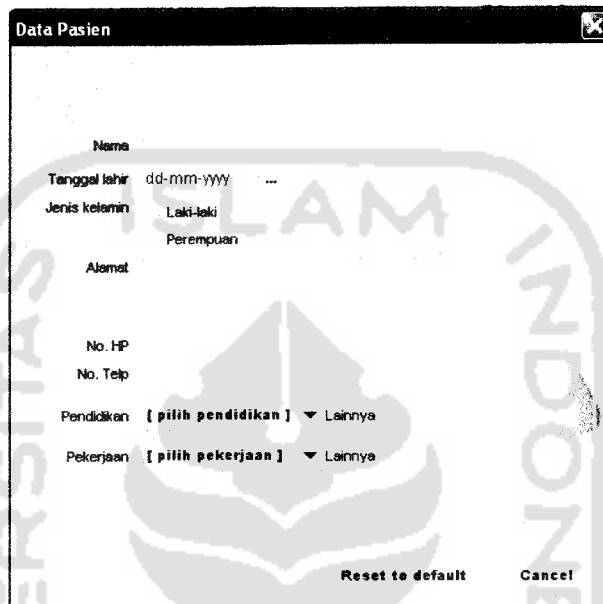
Halaman login akan selalu muncul ketika sistem berhasil dijalankan, sistem berfungsi media pengolahan data (*data manager*). Untuk Tampilan dari *form login admin* dapat dilihat pada gambar 3.39

Gambar 3.39 tampilan login *admin*



3.3.2.3 Form Input Data Pasien

Form input data pasien difungsikan untuk mengolah data-data pasien, olah data tersebut meliputi *insert*, *update* dan *delete*. Untuk tampilan form input data pasien dapat dilihat pada gambar 3.40



The image shows a screenshot of a software window titled "Data Pasien". The window contains a form with the following fields and options:

- Name: [text input field]
- Tanggal lahir: dd-mm-yyyy [calendar icon]
- Jenis kelamin: Radio buttons for "Laki-laki" and "Perempuan"
- Alamat: [text input field]
- No. HP: [text input field]
- No. Telp: [text input field]
- Pendidikan: [dropdown menu] with "pilih pendidikan" and "Lainnya" options
- Pekerjaan: [dropdown menu] with "pilih pekerjaan" and "Lainnya" options
- Buttons: "Reset to default" and "Cancel"

Gambar 3.40 tampilan *form input* data pasien

3.3.2.4 Form Input Data Rekam Medis

Form input data rekam medis digunakan untuk mengisikan data-data kesehatan terbaru pasien. Untuk *form input* data rekam medis dapat dilihat pada 3.41.

Gambar 3.41 tampilan *form input* data rekam medis

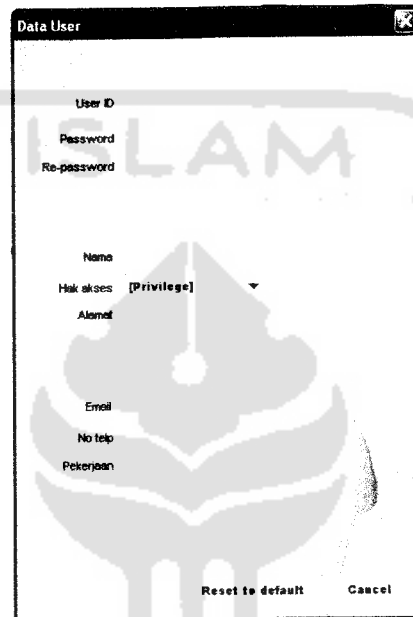
3.3.2.5 Form Input Data Obat

Form input data obat digunakan untuk mengisi data-data tentang deskripsi obat. Untuk dari *form input* data obat dapat dilihat pada gambar 3.42

Gambar 3.42 tampilan *form input* data obat

3.3.2.6 Form Input Data User

Form input data *user* digunakan untuk membuat *user* baru, dengan mengisi data-data pribadi *user*. Untuk tampilan *form input* data *user* dapat dilihat pada gambar 3.43

The image shows a screenshot of a software window titled "Data User". The window contains a form with the following fields: "User ID", "Password", "Re-password", "Name", "Hak akses (Privilege)" with a dropdown arrow, "Alamat", "Email", "No telepon", and "Pekerjaan". At the bottom of the form, there are two buttons: "Reset to default" and "Cancel". The background of the page features a large, faint watermark of the logo of Universitas Islam Indonesia.

Gambar 3.43 tampilan *form input* data *user*

3.3.2.7 Halaman Ubah Password

Halaman ini digunakan untuk mengubah data autentikasi *user*, akan tampil jika sebelumnya *user* berhasil *login* ke sistem. Untuk tampilan ubah password dapat dilihat pada gambar 3.44

The screenshot shows a window titled "Data User" with a close button in the top right corner. Below the title bar, there are two tabs: "Profil" and "Ubah password", with "Ubah password" being the active tab. The form contains three text input fields labeled "Password lama", "Password baru", and "Re-type password". At the bottom right of the form area, there is a "Cancel" button.

Gambar 3.44 tampilan *form* ubah password

3.3.2.8 Form Ubah Pesan

Form ubah pesan digunakan hanya untuk mengubah teks pesan, adapun untuk data-data lain, seperti waktu, no. hp dan status tidak dapat diubah. Tampilan *form* ubah pesan dapat pada gambar 3.45

The screenshot shows a window titled "Data SMS" with a close button in the top right corner. The form displays the following information:

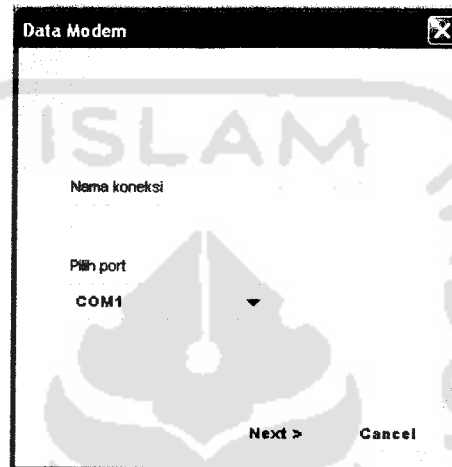
- Pesan:** Nomer ID pasien anda : PS00007
Simpan & segera aktifkan reminder anda.
Ketik panduan ketik PAN.]
- Dari:** 6285868277206
- Waktu:** 2007-08-09 07:11:34
- Status:** Pending

 At the bottom of the form, there are three buttons: "Done", "Reset to default", and "Cancel".

Gambar 3.45 tampilan *form* ubah pesan

3.3.2.9 Form Input Data Koneksi

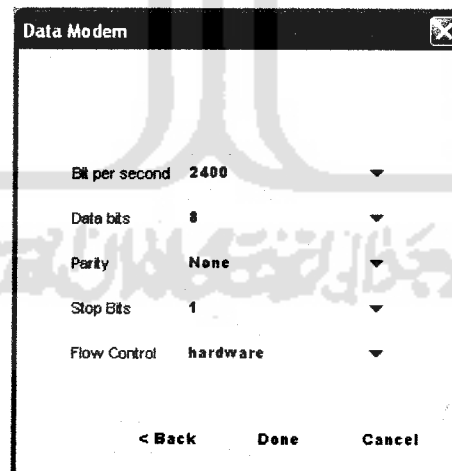
Halaman koneksi modem digunakan untuk menambahkan data koneksi modem, dengan beberapa *field* yang harus diisi sesuai dengan kemampuan dari modem itu sendiri. Untuk tampilan dari halaman ini dapat dilihat pada gambar 3.46 dan gambar 3.47



The screenshot shows a dialog box titled "Data Modem". It contains the following elements:

- A text input field labeled "Nama koneksi" which is currently empty.
- A dropdown menu labeled "Pilih port" with "COM1" selected.
- Two buttons at the bottom: "Next >" and "Cancel".

Gambar 3.46 tampilan *form input* modem panel pertama



The screenshot shows the second panel of the "Data Modem" dialog box. It contains the following settings:

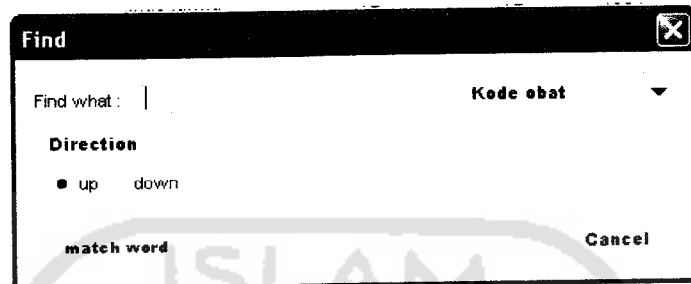
- Baud per second: 2400
- Data bits: 8
- Parity: None
- Stop Bits: 1
- Flow Control: hardware

At the bottom, there are three buttons: "< Back", "Done", and "Cancel".

Gambar 3.47 tampilan *form input* modem panel kedua

3.3.2.10 Form Pencarian Data

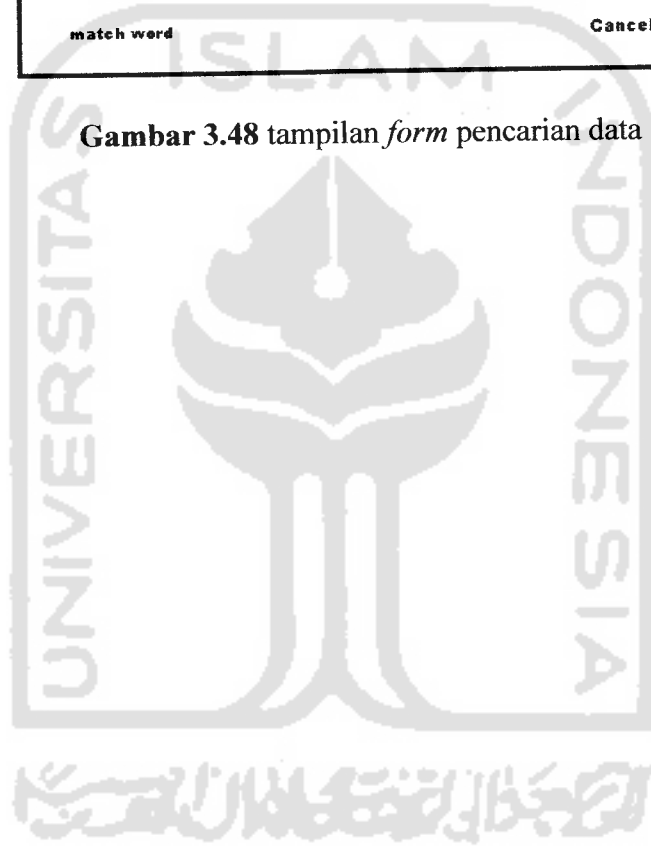
Halaman ini digunakan untuk melakukan pencarian data berdasarkan kategori yang dipilih. Untuk tampilan dapat dilihat pada gambar 3.48



The image shows a 'Find' dialog box with the following elements:

- Title bar: Find
- Find what: []
- Dropdown menu: Kode obat
- Direction section:
 - up (selected)
 - down
- match word []
- Cancel button

Gambar 3.48 tampilan *form* pencarian data



BAB IV

HASIL DAN PEMBAHASAN

4.1 Pengujian Program

Pada bagian ini menjelaskan tentang pengujian terhadap kinerja dari aplikasi pengingat untuk tindakan terapi *insulin* bagi penderita *diabetes mellitus*. Melalui tahap pengujian ini diharapkan dapat diketahui kekurangan-kekurangan dari sistem, untuk selanjutnya 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 mengisi data pasien, rekam medis, obat, *user* dan modem.

4.2 Analisis Kinerja Sistem

4.2.1 Penanganan kesalahan

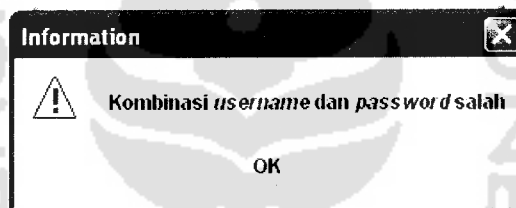
Untuk memastikan bahwa *input* data yang dimasukkan adalah valid, maka diperlukan adanya proses validasi terlebih dahulu sebelum data tersebut diproses lebih lanjut. Oleh karena itu diperlukan adanya pesan kesalahan jika data yang dikirimkan tidak valid. Untuk penanganan kesalahan ini dikelompokkan menjadi dua bagian, yaitu bagian *admin* dan *user*.

4.2.1.1 Penanganan Kesalahan Pada Admin

Admin bertugas melakukan pengolahan data, misalnya *insert*, *edit* dan *delete* data. Data-data tersebut terdiri dari data pasien, data rekam medis, data obat, data *user* dan data modem. Pada bagian ini terdapat beberapa penanganan kesalahan, antara lain :

1 Penanganan Kesalahan Login

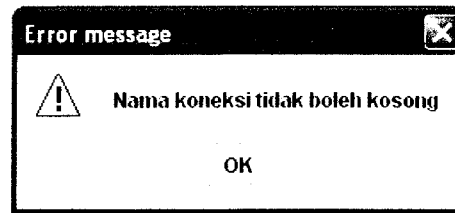
Penanganan kesalahan login berfungsi untuk menangkap kesalahan yang terjadi pada proses login. Contoh, jika *user* salah mengisikan *username* dan *password*, maka akan muncul *messagebox* seperti pada gambar 4.1



Gambar 4.1 kotak dialog jika *password* salah

2 Penganganan Kesalahan Input Kosong

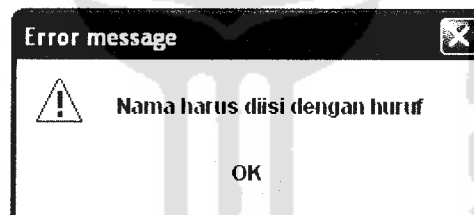
Penanganan kesalahan input kosong berfungsi untuk menangkap kesalahan yang terjadi pada proses pengisian data, dimana *field* masih dalam keadaan kosong. Contoh penanganan ini terdapat pada *form* koneksi modem. Jika pada *field* nama koneksi tetap dalam keadaan kosong, maka akan muncul *messagebox* seperti pada gambar 4.2



Gambar 4.2 kotak dialog jika *field* nama koneksi kosong

3 Penanganan Kesalahan Tipe Data

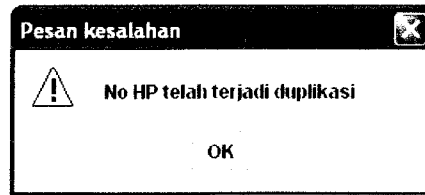
Penanganan kesalahan tipe data berfungsi untuk menangkap kesalahan-kesalahan yang terjadi ketika data yang diisikan tidak sesuai dengan tipe data yang seharusnya.. Contoh penanganan kesalahan ini terdapat pada pengisian *field* nama dalam *form* data pasien. Jika *field* nama terdapat karakter angka, maka akan muncul *messagebox* seperti pada gambar 4.3



Gambar 4.3 kotak dialog jika *field* nama diisi dengan angka

4 Penanganan Kesalahan Duplikasi Data

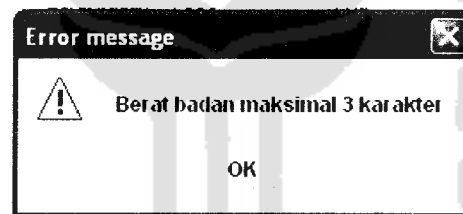
Penanganan kesalahan duplikasi data berfungsi untuk menangkap kesalahan yang terjadi ketika data baru terjadi duplikasi dengan data lama yang terdapat pada basis data. Contoh penanganan kesalahan input duplikasi data pada pengisian *field* dalam *form* data pasien. Jika *field* nomer hp diisi data yang terduplikasi, maka akan muncul *messagebox* seperti pada gambar 4.4



Gambar 4.4 kotak dialog jika nomer hp terjadi duplikasi

5 Penanganan Kesalahan Panjang Data

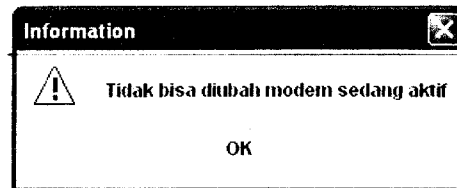
Penanganan kesalahan panjang data berfungsi untuk menangkap kesalahan yang terjadi ketika data yang dimasukkan terlalu panjang atau terlalu pendek. Contoh penanganan ini terdapat pada pengisian *field* berat badan *form* data rekam medis. Jika pada *field* berat badan diisi dengan angka yang lebih dari tiga karakter, maka akan muncul *messagebox* seperti pada gambar 4.5



Gambar 4.5 kotak dialog jika input terlalu panjang

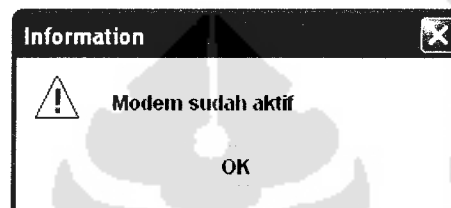
6 Penanganan kesalahan Modem Aktif

Penanganan kesalahan modem aktif berfungsi untuk menangkap kesalahan yang terjadi pada proses ubah 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 *messagebox* seperti pada gambar 4.6



Gambar 4.6 kotak dialog ubah data modem dalam keadaan aktif

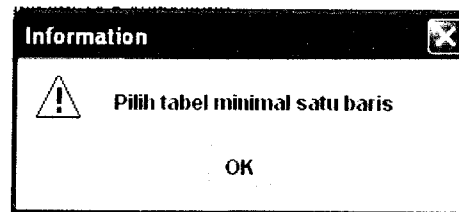
Kedua, pada proses aktivasi modem dimana *user* ingin melakukan aktivasi terhadap modem tertentu dan modem tersebut dalam keadaan aktif, maka akan muncul *messagebox* seperti pada gambar 4.7



Gambar 4.7 kotak dialog aktivasi modem dalam keadaan aktif

7 Penanganan Kesalahan Data Manager

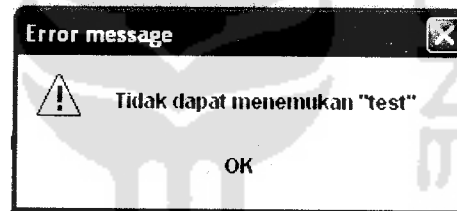
Penanganan kesalahan *data manager* ini berfungsi untuk menangkap kesalahan yang terjadi ketika *user* tidak memenuhi prosedur normal dalam proses olah. Contohnya terjadi pada proses ubah atau hapus data, dimana *user* tersebut tidak melakukan seleksi terhadap tabel, maka akan muncul *messagebox* seperti pada gambar 4.8



Gambar 4.8 kotak dialog jika tabel tidak terpilih

8 Penanganan kesalahan Pencarian Data

Penanganan kesalahan pencarian data berfungsi untuk menangkap kesalahan pada proses pencarian. Contoh, kesalahan tersebut terjadi jika data yang dicari tidak ditemukan, maka akan muncul *messagebox* seperti pada gambar 4.9



Gambar 4.9 kotak dialog jika data tidak ditemukan

4.2.1.2 Penanganan Kesalahan Pada User

Penanganan kesalahan pada bagian *user* ini berupa proses validasi SMS yang dikirim oleh *user* ke *server*. SMS tersebut berisikan perintah-perintah untuk melakukan interaksi dengan system utama. Terdapat beberapa penanganan kesalahan pada bagian ini, antara lain :

1 Perintah Tidak Dikenal

Jika *user* mengirimkan SMS berisi perintah yang tidak dikenal, maka *user* akan memperoleh SMS balik dari *server*. SMS tersebut adalah "Perintah yang tersedia REG, RM, PM, JDW, CP. Untuk bantuan ketik ? atau PAN".

2 Format Penulisan

Jika format penulisan SMS tidak sesuai dengan format yang ditentukan. Contoh penanganan kesalahan ini terdapat pada pengubahan data kesehatan. Jika dalam pengisian gula darah dan berat badan tidak menggunakan tanda "=", maka *server* akan mengirimkan SMS balik ke *user* yang berisi "Perintah salah. Ketik RM <password> <GD=guladarah> <BB=beratbadan>. Gunakan (=) pada isian GD dan BB".

3 Tipe Data

Penanganan kesalahan tipe data terdapat pada proses ubah data kesehatan. Kesalahan terjadi jika *user* mengirimkan data gula darah atau data berat badan dengan tipe data *string* (huruf), seharusnya dengan tipe data tersebut adalah *integer* (angka). Selanjutnya *server* akan mengirimkan SMS balik yang berisikan "GULA DARAH dan BERAT BADAN maksimal 3 angka".

4 Panjang Karakter

Jika panjang karakter dari data yang dikirimkan *user* terlalu panjang atau terlalu pendek. Contoh penanganan kesalahan ini terdapat pada pengubahan password. Jika password baru yang dikirim panjang karakternya lebih dari 10

(sepuluh), maka akan *server* akan mengirimkan SMS balik yang berisi "Password baru min 5 digit dan maks 10 digit"

5 Format Waktu

Penanganan kesalahan format waktu terdapat pada proses pengisian data pola makan. Kesalahan terjadi jika input waktu yang diisikan tidak valid, maka *server* akan mengirimkan SMS balik yang berisi "Isian waktu salah.Harus angka,2 digit dan pastikan t` dapat tanda (:).Contoh 10:10".

6 ID Pasien Tidak Dikenal

Penanganan kesalahan ini terdapat pada proses registrasi. Kesalahan terjadi jika ID Pasien yang dikirimkan tidak dikenal, maka *server* akan mengirimkan SMS balik ke *user*. SMS balik tersebut berisi "ID pasien tdk dikenal.Ulangi dengan mengisikan Id pasien yg benar".

7 Password Salah

Penanganan kesalahan ini terdapat pada proses ubah data kesehatan, ubah data pola makan dan ubah password. Contoh pada proses ubah data kesehatan, jika *password* tidak dikenal maka *server* akan mengirimkan SMS balik yang berisi "Password salah atau belum terdaftar.Untuk daftar,ketik REG <ID pasien>".

8 User Sudah Terdaftar

Penanganan kesalahan ini terdapat pada proses pendaftaran. Kesalahan terjadi jika *user* yang sudah terdaftar mengulangi proses pendaftaran, maka *server*

akan mengirimkan SMS balik yang berisi "Anda telah terdaftar.Update GULA DARAH & BERAT BADAN,krn menentukan dosis insulin"

9 Tipe Regimen

Tipe regimen merupakan variabel yang digunakan untuk menentukan intensitas pemberian terapi. Tipe regimen dari masing-masing pasien berbeda, tergantung pada umur pasien. Penanganan kesalahan ini terdapat pada proses ubah data pola makan. Kesalahan terjadi jika *user* mengirimkan data pola makan yang tidak sesuai dengan tipe regimennya, maka akan memperoleh SMS balik. SMS balik tersebut berisi "Pola makan tdk sesuai dgn regimen.Ketik makan pagi(hh:mm)#makan malam(hh:mm)".

4.2.2 Pengujian dan analisis

Pada tahap pengujian dan analisis program ini, akan dilakukan perbandingan antara kebenaran masukan terhadap kebutuhan sistem. Seperti halnya pada penanganan kesalahan, untuk pengujian dan analisis dikelompokkan menjadi dua bagian, yaitu *admin* dan *user*.

4.2.2.1 Pengujian Pada Admin

Pada bagian ini terdapat beberapa hasil pengujian, antara lain : masukan login, pasien, rekam medis, obat, *user* dan modem.

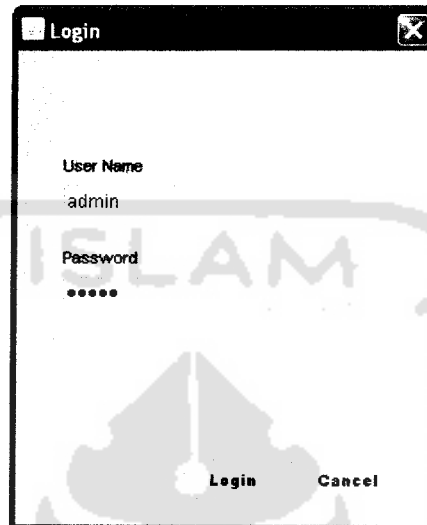
1 Masukan Login

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

Username : admin

Password : admin

Hasil dari masukan data login tersebut dapat dilihat pada gambar 4.10



Gambar 4.10 antarmuka masukan login

2 Masukan Data Pasien

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

1. Nama pasien : Iman handoyo
2. Tgl. lahir : 7 - Agustus - 1963
3. Jenis kelamin : Laki-laki
4. Alamat : Jln. Kaliurang KM 14,4 Yogyakarta
5. No. Ponsel : 085868277206
6. No. Telp : 0274896131

7. Pendidikan : Tamat SMU

8. Pekerjaan : Pegawai negeri

Hasil dari masukan data-data tersebut dapat pada gambar 4.11

No	Id Pasien	Nama	Alamat	Pekerjaan
1	PS00001	Hartono	Jl HOS Cokroaminoto no 81 Rembang	
2	PS00002	Ahmad Kurniawan	JL P Sudirman no 77 Rembang	
3	PS00003	Suhartini	Jl Kaliurang km 14,5 Yogyakarta	
4	PS00004	Imam handoyo	Jln. Kaliurang KM 14,4 Yogyakarta	Tamat SMU

Gambar 4.11 antarmuka masukan data pasien

3 Masukan Data Rekam Medis

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

1. Id pasien : PS00001
2. Berat badan : 70 Kg
3. Glukosa puasa : 250 mmol / liter
4. Glukosa dua jam : 270 mmol / liter
5. Tekanan darah : 160
6. Kolesterol : 300
7. Anjuran obat : OB00001
8. Tanggal pengisian : 29 - Agustus - 2007

Hasil dari masukan data rekam medis tersebut dapat dilihat pada gambar 4.12

No	Kode RM	Nama	Berat	GD Puasa	GD Dua Jam	Tek Darah	Kolesterol
1	RM00003	Hartono	70	200	300	120	200
2	RM00004	Imam Handoyo	67	270	300	170	500

Gambar 4.12 antarmuka masukan data rekam medis

4 Masukan Data Obat

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

1. Nama obat : Urispas 200
2. Vendor : Indo Farma
3. Efek puncak : 3 jam
4. Lama puncak : 7 jam
5. Netton : 400 ml

Hasil dari masukan data obat tersebut dapat dilihat pada gambar 4.13

No	Kode Obat	Nama	Vendor	Waktu Puncak	Durasi	Netto
1	OB00002	Urispas 200	Indo farma	12	12	4004

Gambar 4.13 antarmuka masukan data obat

5 Masukan Data User

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

1. *Username* : r1fk1

2. *Password* : bismillah
3. *Re-type password* : bismillah
4. Nama : Rifki
5. Hak akses : Admin
6. Alamat : Jln HOS Cokroaminoto no 71 Rembang
7. Email : k4467md@yahoo.com
8. No. Telp : 085868277205
9. Pekerjaan : Mahasiswa

Hasil dari masukan data *user* tersebut dapat dilihat pada gambar 4.14

No	Nama	Alamat	Telp	Status
1	rifki	Rembang		paramedis
2	Rifki	Jln HOS Cokroaminoto no 71 Rembang	085868277205	admin
3	Sugiro	Jl Magelang KM 5	ert	paramedis
4	Toni	Ring road utara		paramedis
5	Tri winardi	Ring road barat	test	admin

Gambar 4.14 antarmuka masukan data *user*

6 Masukan Data Modem

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

Nama koneksi : Motorola E1

Port yang digunakan : com 3

Bit per second : 19200

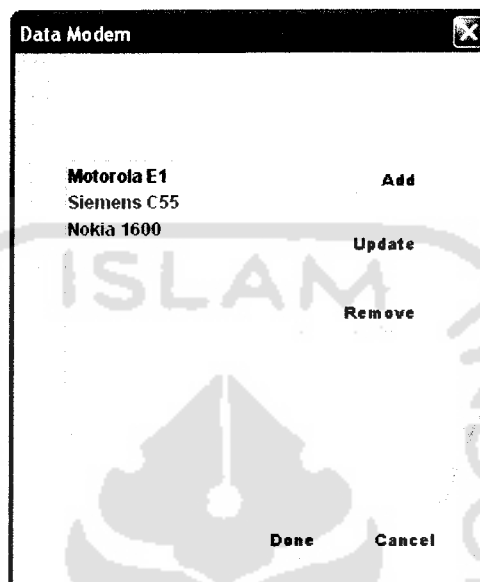
Data bits : 8

Parity : none

Stop bits : 1

Flow control : hardware

Hasil dari masukan data modem tersebut dapat dilihat pada gambar 4.15



Gambar 4.15 antarmuka masukan data modem

4.2.2.2 Pengujian Pada User

Pengujian pada bagian ini digunakan untuk mengetahui kebenaran perintah yang dikirim oleh *user* melalui SMS. Terdapat beberapa hasil pengujian, antara lain :

1 Masukan pendaftaran

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

User kirim SMS : "REG PS 00001"

Sistem kirim SMS : "Pendaftaran dalam proses.Selanjutnya,ketik makan pagi(hh:mm)#makan malam(hh:mm)".

Sistem kirim SMS : "07.30#20.30".

User kirim SMS : "Pendaftaran berhasil.Password anda (12345).Untuk keamanan,ubah password anda.Ketik CP <password lama> <password baru>".

2 Masukan Panduan

Beberapa urutan proses pada pengujian perintah panduan, antara lain :

User kirim SMS : "REG?"

Sistem kirim SMS : "Untuk daftar,ketik REG <ID Pasien>.Contoh REG PS00001".

3 Masukan Ubah Data Rekam Medis

Diasumsikan bahwa *user* telah terdaftar, pengujian dilakukan dengan mengirimkan *SMS* ke *server*. Terdapat beberapa urutan proses pada pengujian ini, antara lain :

User kirim SMS : "RM 12345 GD=250 BB=70"

Sistem kirim SMS : "Data rekam medis berhasil disimpan.Update terus GULA DARAH & BERAT BADAN,krn menentukan dosis insulin".

4 Masukan Ubah Pola Makan

Syarat untuk pengujian ini seperti halnya pada pengujian ubah data kesehatan, yaitu *user* sudah terdaftar. Terdapat beberapa urutan proses pada pengujian ini, antara lain :

User kirim SMS : "PM 12345"

Sistem kirim SMS : "Selanjutnya, ketik makan pagi(hh:mm)#makan malam(hh:mm)".

User kirim SMS : "07.30#20.30".

Sistem kirim SMS : "Data pola makan berhasil diubah"

5 Masukan Jadwal Terapi

Pengujian dilakukan dengan mengirimkan SMS yang berisi "JDW". Hasil dari pengujian ini, yaitu *user* akan memperoleh jadwal terapi sesuai dengan kondisi kesehatan terakhir.

6 Masukan Ubah Password

Terdapat beberapa proses pada pengujian ubah password, antara lain :

User kirim SMS : "CP 12345 abcde"

Sistem kirim SMS : "Password berhasil diubah".

7 SMS Peningat Terapi

SMS pengingat yang berisi nasehat untuk melakukan terapi beserta dosis merupakan output utama dari sistem pengingat untuk tindakan terapi *insulin* bagi penderita *diabetes mellitus*. Dosis yang diberikan kepada *user* (pasien) dalam penentuan keputusannya berdasarkan pada rumus baku penentuan dosis *insulin*

yang telah ditetapkan oleh pakar kesehatan. Untuk lebih jelasnya *sliding scale* dari rumus baku *insulin* dapat dilihat pada tabel 4.1

Tabel 4.1 Tabel *sliding scale* dosis

(mmol / liter)	Blood Glucose (mg / 100 ml)	Subcutaneous <i>Insulin</i> Dose (units)
< 7,8	< 140	0
7,8 – 11,8	140 – 200	2
11,1 – 16,6	201 – 300	5
16,6 – 22,2	301 – 400	10
> 22,2	> 400	12

SMS pengingat yang dikirimkan kepada pasien sesuai dengan pola makan dari masing-masing pasien. Intensitas terapi antar pasien tidak sama, tergantung pada tipe regimen dari masing-masing pasien. Penentuan tipe regimen berdasarkan umur pasien, berikut ini adalah daftar tipe regimen beserta intensitas terapinya :

1. Tipe regimen 1 untuk pasien dengan umur < 35 tahun. Terapi dilakukan 1 x sehari, yaitu sebelum makan pagi.
2. Tipe regimen 2 untuk pasien dengan umur 17 – 35 tahun. Terapi dilakukan sebanyak 2 x sehari, yaitu sebelum makan pagi dan sebelum makan malam.
3. Tipe regimen 3 untuk pasien dengan umur kurang dari 17 tahun. Terapi dilakukan sebanyak 4 x sehari, yaitu sebelum makan pagi, sebelum makan siang, sebelum makan malam dan sebelum tidur.

Berikut ini adalah contoh penghitungan *dosis insulin*, diketahui data-data kesehatan pasien seperti dibawah ini.

1. Nama : Hartono
2. Tanggal lahir : 06-08-1965
3. Berat badan : 70 Kg
4. Glukosa puasa : 250 mmol / liter
5. Glukosa dua jam : 220 mmol / liter
6. Tekanan darah : 160
7. Kolesterol : 500
8. Tipe regimen : 1 (ditentukan secara otomatis oleh sistem)
9. Waktu makan pagi : 7.30

Hasil output dosis harian yang diberikan adalah 140 unit, karena termasuk tipe regimen satu maka intensitas terapi hanya satu kali, yaitu sebelum makan. Sehingga SMS pengingat yang dikirimkan pada jam 7.00 berisi "Jangan lupa untuk terapi, dosis insulin anda = 140 unit".

BAB V

SIMPULAN DAN SARAN

5.1 Simpulan

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan, dapat disimpulkan bahwa :

1. Selain digunakan sebagai media SMS interaktif seperti *polling* SMS, kuis SMS, berita langganan dan lainnya. Teknologi SMS *gateway* juga dapat digunakan sebagai *reminder* (pengingat).
2. Teknologi SMS *gateway* dapat diimplementasikan sebagai alat pengingat untuk tindakan terapi insulin bagi penderita *diabetes mellitus*, sehingga konsistensi terapi dapat terus terjaga.
3. Kelemahan dari sistem ini adalah hasil perhitungan dosis yang masih bersifat *crisp* (tegas), sehingga *output* dosis yang dihasilkan belum bisa mewakili kondisi pasien yang sesungguhnya.
4. Belum adanya *synchronization* (sinkronisasi) waktu antara server dengan *client* (*handphone user*), sehingga waktu *reminder* tidak sama antara *server* dengan *user*.
5. Jika jaringan sibuk atau mengalami *congestion* (macet), misalnya pada waktu hari raya, fungsi SMS sebagai *reminder* tidak bisa difungsikan sebagaimana mestinya.

5.2 Saran

Mengingat berbagai keterbatasan yang dialami penulis terutama masalah pemikiran dan waktu, maka penulis menyarankan untuk pengembangan penelitian dimasa yang akan datang sebagai berikut :

1. Untuk perhitungan dosis, sebaiknya menggunakan pendekatan logika *fuzzy*.
2. Sebaiknya disertai *print out* (laporan cetak) untuk data-data yang diperlukan oleh kegiatan medis lain, misalnya data pasien dan data rekam medis.
3. Sebaiknya diintegrasikan teknologi *synchronization* waktu antara *server* dengan *client (handphone user)*.



DAFTAR PUSTAKA

- [BER86] Berger. *Insulin Therapy : Conventional*. Amstredam : Elsevier Science Publisher B. V, 1986.
- [FAT99] Fathansyah. *Basis Data*. Bandung : Informatika Bandung, 1999.
- [GEO04] George, Joey F. *Object-Oriented Systems Analysis and Design*. Prentice Hall, 2004.
- [KOM05] Komputer, Wahana. *Pengembangan Aplikasi Simtem Informasi Akademik Berbasis SMS Dengan Java*. Jakarta : Salemba Infotek, 2005.
- [LAR05] Larman, Craig. *Applying UML and Patterns : An Introduction to Object-Oriented Analysis and Design and Iterative Development*. Prentice Hall, 2005.
- [LET02] Lethbridge, Timothy C. *Object Oriented Software Engineering: Practical software development using UML and Java*. McGraw-Hil, 2002.
- [PRA04] Prasetyo, Didik D. *Tip dan Trik Pemrograman JAVA 2*. Jakarta : PT Elex Media Komputindo, 2004.
- [RAS86] Raskin, Rosenscock J. *Blood glucose control and diabetic complications*. Ann Intern Med, 1986.

- [SIN04] Sinaga, Benyamin. *Pemrograman Berorientasi Objek Dengan JAVA*. Yogyakarta : Gava Media, 2004.
- [STU05] Stumpf, Robert V. *Object-Oriented Systems Analisis and Design with UML*. Prentice Hall, 2005.

