

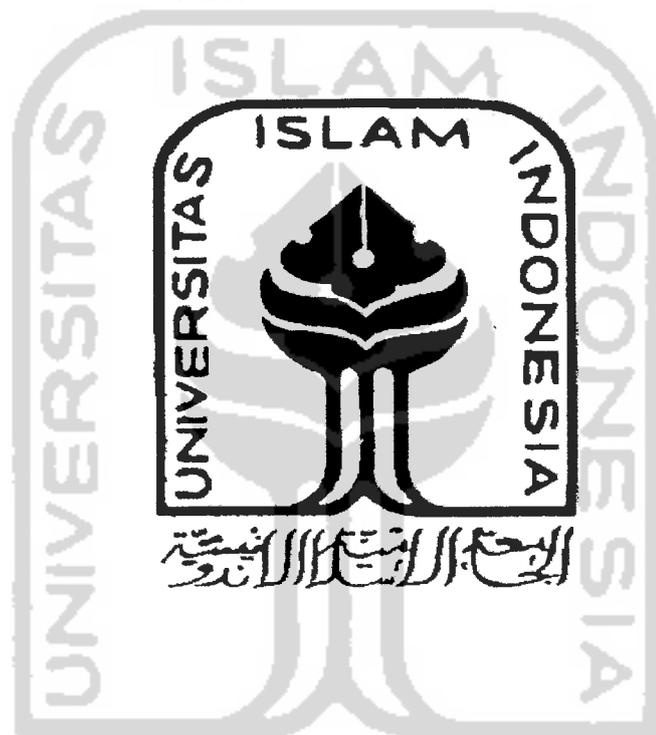
**SISTEM PENGAKSESAN DATA PRIBADI MAHASISWA  
BERBASIS SMS**

**TUGAS AKHIR**

**Diajukan sebagai Salah Satu Syarat**

**Untuk Memperoleh Gelar Sarjana**

**Jurusan Teknik Informatika**



Oleh :

Nama : Ade Irawan  
No. Mahasiswa : 03.523.217

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

**2007**

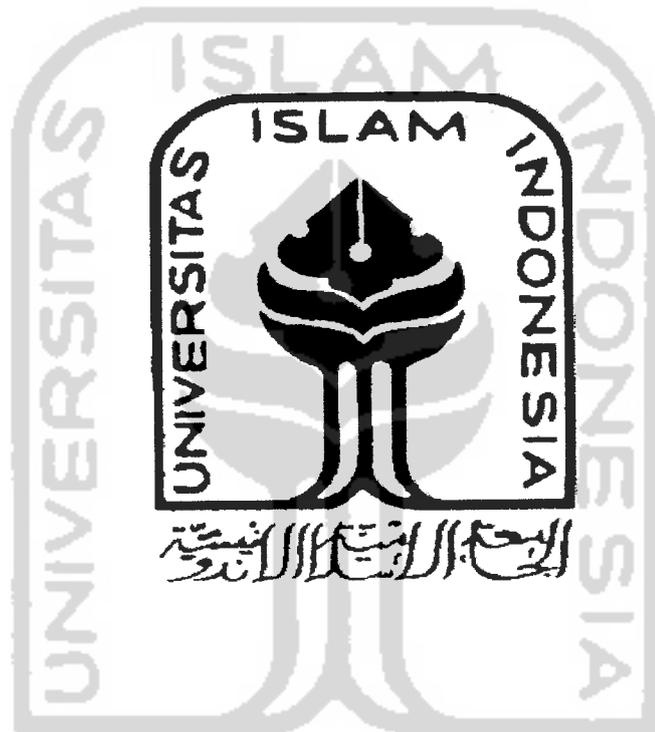
**SISTEM PENGAKSESAN DATA PRIBADI MAHASISWA  
BERBASIS SMS**

**TUGAS AKHIR**

**Diajukan sebagai Salah Satu Syarat**

**Untuk Memperoleh Gelar Sarjana**

**Jurusan Teknik Informatika**



**Oleh :**

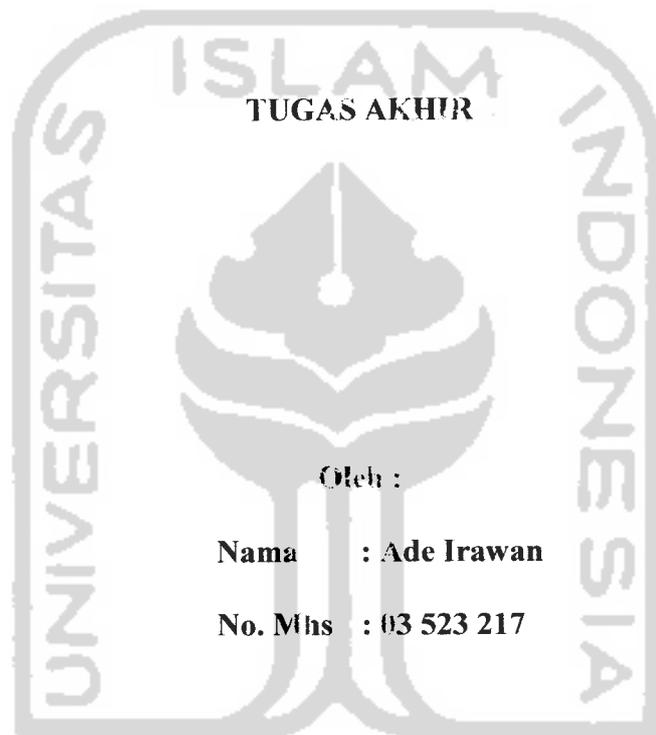
Nama : Ade Irawan  
No. Mahasiswa : 03.523.217

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

**2007**

**LEMBAR PENGESAHAN PEMBIMBING**

**SISTEM PENGAKSESAN DATA PRIBADI MAHASISWA  
BERBASIS SMS**



Yogyakarta, 5 September 2007

Pembimbing,

**(Taufiq Hidayat, ST., MCS.)**

# LEMBAR PENGESAHAN PENGUJI

## SISTEM PENGAKSESAN DATA PRIBADI MAHASISWA BEBASIS SMS

### TUGAS AKHIR

Oleh :

Nama : Ade Irawan

No. Mahasiswa : 03 523 217

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

Yogyakarta, 26 September 2007

**Tim Penguji**

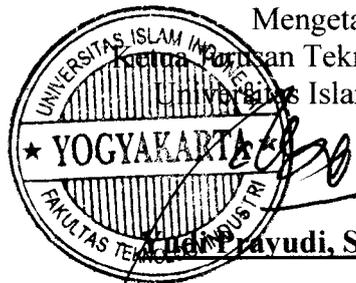
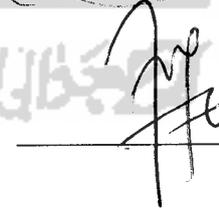
**Taufiq Hidayat, ST., MCS.**  
Ketua



**Hendrik, ST.**  
Anggota I



**Nur Wijayaning Rahayu, S.Kom.**  
Anggota II



Mengetahui,  
Ketua Jurusan Teknik Informatika  
Universitas Islam Indonesia

**Prayudi, S.Si., M.Kom**

**LEMBAR PERNYATAAN KEASLIAN  
HASIL TUGAS AKHIR**

Saya yang bertandatangan di bawah ini,

Nama : Ade Irawan

No. Mahasiswa : 03 523 217

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, 5 September 2007



Ade Irawan

## HALAMAN PERSEMBAHAN

*Dengan mengucapkan syukur Allhamdulillah kupersembahkan karya ini untuk.....*

*Allah SWT sang pencipta alam semesta, penguasa hati dan jiwa manusia, pemilik semua ilmu pengetahuan sebagai tanda pengabdian hamba dalam mempelajari ayat-ayat-Mu. Segala puji syukur kepada-Mu atas karunia Islam dan Iman serta segala kemudahan*

*Rasulullah SAW pembawa syafaat dan suri tauladan terpantas, Shalawat serta Salam hanya selalu tercurah kepadamu dan pengikutmu*

*Kedua Orangtuaku tercinta yang telah memberikan dorongan semangat, do'a, serta cinta kasih tulus yang selalu menemani perjuangan dan usaha ku selama ini, sebagai rasa hormat dan sayangku terimalah persembahanku.*

*Sahabat sahabatku tersayang*

*Dewi Lasrida P, Hanif, Rini, Mbak lia, Mbak indah, Dina, Anjar, Fais, Isal*

*Terima kasih atas kasih sayang, kesabaran, dan perhatiannya selama ini.*

*Semoga Allah SWT membalasnya dengan yang lebih baik.*

*Amin....*

*Temen-temen Informatika*

*Pak gor(Rifki), RudionXT9550(Rudi), Riki, Bayu, Agus, Fajar , Rian, Pipit,*

*Rio, Nico, Mul*

*Dan juga semua temen2 informatika Angkatan '03.*

*Terima kasih sebesar-besarnya atas pertemanan dan perjuangan yang indah nan berkesan selama ini.*

*Semoga Sukses dalam menatap masa depannya masing-masing.*

*Amin....*

## HALAMAN MOTTO

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

*Dengan menyebut nama Allah yang Maha Pemurah lagi Maha Penyayang  
Segala puji bagi Allah, Tuhan semesta alam, Yang Maha Pemurah lagi Maha Penyayang  
Penguasa tunggal pada hari pembalasan  
Hanya kepada-Mu kami menyembah dan hanya kepada-Mu kami mohon pertolongan  
Tunjukkan kami jalan yang benar, Jalan mereka yang Engkau beri nikmat, bukan jalan mereka  
yang Engkau murkai dan bukan pula jalan mereka yang sesat (Q.S Al Faatihah: 1-7)*

*Bacalah dengan nama Tuhanmu yang menciptakan.  
Dia telah menciptakan manusia dari segumpal darah.  
Bacalah, dan Tuhanmulah yang maha pemurah;  
Yang mengajarkan Qalam.  
Dialah yang mengajar manusia segala yang belum diketahui.*  
(QS. Al-Alaq 1-5)

*Al Qur'an menjadi petunjuk dan juga obat bagi segenap orang yang beriman (Q.S Fushilat: 44)*

*Sungguh, bersama kesukaran itu pasti ada kemudahan (QS. Asy Syarh : 6)*

## KATA PENGANTAR



*Assalamu'alaikum Wr.Wb.*

Puji syukur kehadiran Allah SWT. Yang telah melimpahkan rahmat dan hidayah-Nya sehingga laporan Kerja Paktek ini dapat diselesaikan. 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 menyelesaikan jenjang kesarjanaan Strata 1 (S1) pada jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia.

Selama dalam pelaksanaan Tugas Akhir dan penulisan laporan, penyusun 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. Bapak Taufiq Hidayat, ST., MCS. selaku dosen pembimbing yang telah memberikan pengarahan dan bimbingan selama pengerjaan tugas akhir dan penulisan laporan ini.
5. Seluruh dosen yang telah memberikan ilmu kepada penulis dan membimbing penulis untuk mengetahui dan mempelajari ilmu informatika
6. Kedua Orang tua saya yang telah membimbing, mendoakan, dan membantu saya hingga menjadi orang yang mudah-mudahan berguna dimasa depan.
7. Teman-teman yang selalu membantu, mendo'akan dan memberikan dukungan kepada kami.
8. Serta berbagai pihak yang tidak mungkin kami sebutkan satu persatu.

Segegap daya upaya telah saya curahkan untk terselesaikan laporan Tugas Akhir ini. Kami 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, 5 September 2007

Penyusun

## ABSTRAKSI

Kemajuan teknologi berkembang begitu pesat, terutama teknologi komputer dan telepon seluler. Selain diminati banyak kalangan, telepon seluler juga mempunyai banyak fasilitas yang dapat digunakan, terutama SMS yang dinilai sangat praktis, murah, dan efisien. SMS ( *Short Message Service* ) adalah layanan yang disediakan oleh operator ponsel untuk mengirim dan menerima pesan singkat. Selain untuk mengirim pesan antar pengguna ponsel, SMS juga cocok untuk diterapkan dan berinteraksi dengan suatu sistem informasi.

Pada kasus ini banyak mahasiswa yang mempunyai banyak keluhan ketika ingin mengakses ataupun mengubah tentang data-data pribadinya, seperti minimnya biaya, antri warnet dan sebagainya. Dengan adanya sistem ini mahasiswa hanya perlu mengirimkan SMS ke sistem untuk memperoleh dan mengubah data tersebut. Sistem ini menggunakan konsep SMS balik kepada user (*SMS gateway*), yang diharapkan dapat mengurangi keluhan-keluhan mahasiswa dalam mencari informasi tentang data-data akademiknya.

Oleh karena itu, penulis membangun sistem pengaksesan data pribadi mahasiswa berbasis sms ini yang bertujuan untuk mengatasi beberapa masalah diatas. Dengan harapan sistem ini dapat berdaya guna bagi mahasiswa untuk mengakses dan mengubah data pribadinya melalui sarana SMS.

Kata kunci : *Short Message Service (SMS), SMS gateway.*

## TAKARIR

<i>admin</i>	administrator
<i>byte</i>	satuan ukuran
<i>delivery report</i>	laporan pengiriman
<i>status report</i>	laporan status
<i>time-out</i>	batasan waktu
<i>source code</i>	kode program
<i>edit</i>	merubah
<i>delete</i>	menghapus
<i>field</i>	kolom
<i>form</i>	halaman
<i>input</i>	masukan
<i>interface</i>	antarmuka
<i>login</i>	masuk ke sistem
<i>message box</i>	kotak pesan
<i>dialog box</i>	kotak dialog
<i>output</i>	keluaran
<i>password</i>	kata kunci
<i>sms server</i>	penyedia layanan sms
<i>user</i>	pengguna
<i>mobile device</i>	perangkat bergerak
<i>wireless</i>	tanpa kabel

## DAFTAR ISI

<b>HALAMAN JUDUL</b> .....	i
<b>LEMBAR PENGESAHAN PEMBIMBING</b> .....	ii
<b>LEMBAR PENGESAHAN PENGUJI</b> .....	iii
<b>LEMBAR PERNYATAAN KEASLIAN</b> .....	iv
<b>LEMBAR PERSEMBAHAN</b> .....	v
<b>HALAMAN MOTTO</b> .....	vii
<b>KATA PENGANTAR</b> .....	viii
<b>ABSTRAKSI</b> .....	x
<b>TAKARIR</b> .....	xi
<b>DAFTAR ISI</b> .....	xii
<b>DAFTAR TABEL</b> .....	xvii
<b>DAFTAR GAMBAR</b> .....	xviii
<b>BAB I PENDAHULUAN</b> .....	1
1.1 Latar belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah .....	3
1.4 Tujuan Penelitian .....	3
1.5 Manfaat Penelitian .....	4
1.6 Metodologi Penelitian .....	4

1.6.1 Metode Pengumpulan Data .....	4
1.6.2 Metode Pengembangan Sistem .....	5
1.7 Sistematika Penulisan .....	5
<b>BAB II LANDASAN TEORI .....</b>	<b>7</b>
2.1 <i>SMS (Short Message Service)</i> .....	7
2.1.1 Pengertian <i>SMS (Short Message Service)</i> .....	7
2.1.2 Mekanisme cara kerja <i>SMS (Short Message Service)</i> .....	10
2.1.3 <i>SMS Gateway</i> .....	11
2.2 <i>Access and Terminals (AT) Commands</i> .....	12
2.3 <i>Mobile atau Wireless</i> .....	13
2.3.1 Perkembangan Teknologi Jaringan Wireless .....	14
2.3.2 <i>GSM Cable Data</i> .....	15
2.4 <i>UML (Unified Modeling Language)</i> .....	16
<b>BAB III METODOLOGI .....</b>	<b>17</b>
3.1 Analisis Kebutuhan Perangkat Lunak .....	17
3.1.1 Metode Analisis .....	17
3.1.2 Hasil Analisis .....	17
3.1.2.1 Analisis Kebutuhan Proses .....	18
3.1.2.2 Analisis Kebutuhan Input .....	18
3.1.2.3 Analisis Kebutuhan Output .....	19

3.1.3	Kebutuhan Antarmuka .....	20
3.1.4	Analisis Kebutuhan Perangkat Lunak .....	20
3.1.5	Analisis Kebutuhan Perangkat Keras .....	20
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> ) .....	22
3.2.2.1.1	<i>Use Case Diagram</i> .....	22
3.2.2.1.2	<i>Class Diagram</i> .....	24
3.2.2.1.3	<i>Sequence Diagram</i> .....	28
3.2.2.1.4	<i>Activity Diagram</i> .....	49
3.2.2.2	Perancangan Basis Data .....	54
3.2.2.2.1	Struktur Tabel .....	54
3.2.2.2.2	Relasi Antar Tabel .....	58
3.2.2.3	Perancangan Antarmuka .....	59
3.2.2.3.1	Perancangan Halaman Utama .....	59
3.2.2.3.2	Perancangan Input .....	60
3.2.2.3.2.1	Perancangan Format Sms ...	60
3.2.2.3.2.2	Perancangan Masukan Data	63
	<i>Sms Server</i> .....	
3.3	Implementasi Perangkat Lunak .....	70
3.3.1	Batasan implementasi .....	70

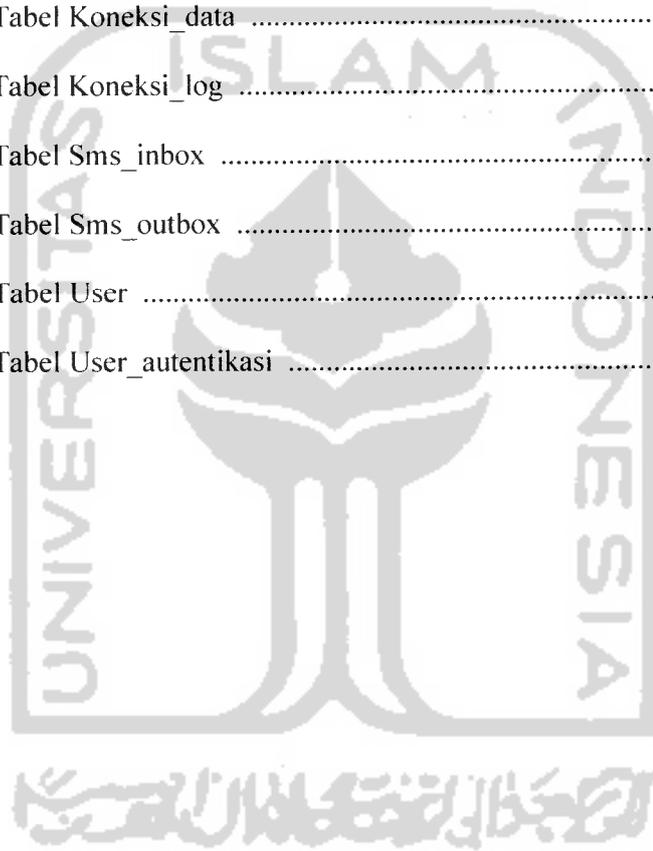
3.3.2	Implementasi Format Sms .....	70
3.3.3	Implementasi Antarmuka .....	72
3.3.3.1	Halaman Login .....	73
3.3.3.2	Halaman Utama <i>User</i> .....	73
3.3.3.3	Halaman <i>About</i> .....	74
3.3.3.4	Halaman <i>Help</i> .....	75
3.3.3.5	Input Data Koneksi .....	75
3.3.3.6	Input Data Mahasiswa .....	76
3.3.3.7	Input Data IPK .....	77
3.3.3.8	Input Data Jadwal Kuliah .....	77
3.3.3.9	Input Data Jadwal Ujian .....	78
3.3.3.10	Olah Data Sms .....	78
3.3.3.11	Halaman Ubah Password <i>User</i> .....	79
3.3.3.12	Halaman Ubah Data Sms .....	79
3.3.4	Implementasi Prosedural .....	80
<b>BAB IV HASIL DAN PEMBAHASAN .....</b>		<b>82</b>
4.1	Pengujian Program .....	82
4.2	Analisis Kinerja Sistem .....	82
4.2.1	Penanganan Kesalahan .....	82
4.2.1.1	Penanganan Kesalahan User Mahasiswa .....	83
4.2.1.2	Penanganan Kesalahan User Admin .....	85

4.2.2	Pengujian dan Analisis .....	88
4.2.2.1	Pengujian Untuk User Mahasiswa .....	88
4.2.2.2	Pengujian Untuk User Admin .....	92
4.2.3	Hasil Analisis .....	101
<b>BAB V SIMPULAN DAN SARAN .....</b>		
5.1	Simpulan .....	103
5.2	Saran .....	104
<b>DAFTAR PUSTAKA .....</b>		105
<b>LAMPIRAN A .....</b>		106
<b>LAMPIRAN B .....</b>		108



## DAFTAR TABEL

Tabel 2.1	Tabel AT Command Untuk SMS .....	13
Tabel 3.1	Tabel Mahasiswa .....	54
Tabel 3.2	Tabel Mahasiswa_otentikasi .....	55
Tabel 3.3	Tabel Koneksi_data .....	55
Tabel 3.4	Tabel Koneksi_log .....	55
Tabel 3.4	Tabel Sms_inbox .....	56
Tabel 3.5	Tabel Sms_outbox .....	56
Tabel 3.7	Tabel User .....	57
Tabel 3.8	Tabel User_otentikasi .....	57



## DAFTAR GAMBAR

Gambar 2.1 <i>Short Message Mobile Terminated</i> .....	8
Gambar 2.2 <i>Short Meessage Mobile Originated</i> .....	8
Gambar 2.3 Mekanisme Cara Kerja SMS .....	10
Gambar 3.1 <i>Use Case Diagram</i> .....	23
Gambar 3.2 <i>Class Diagram</i> Tampilan Kelompok Pengolahan Data .....	25
Gambar 3.3 <i>Class Diagram</i> Objek dan <i>Control</i> Kelompok Pengolahan Data ..	26
Gambar 3.4 Relasi <i>Classs Diagram</i> Kelompok Pengolahan Data .....	27
Gambar 3.5 <i>Class Diagram</i> Kelompok Pengolahan User .....	28
Gambar 3.6 <i>Sequence Diagram</i> Tambah Data Koneksi .....	30
Gambar 3.7 <i>Sequence Diagram</i> Ubah Data Koneksi .....	31
Gambar 3.8 <i>Sequence Diagram</i> Hapus Data Koneksi .....	32
Gambar 3.9 <i>Sequence Diagram</i> Ubah Data Sms .....	33
Gambar 3.10 <i>Sequence Diagram</i> Hapus Data Sms .....	34
Gambar 3.11 <i>Sequence Diagram</i> Tambah Data Mahasiswa .....	34
Gambar 3.12 <i>Sequence Diagram</i> Ubah Data Mahasiswa User Admin .....	35
Gambar 3.13 <i>Sequence Diagram</i> Ubah Data Mahasiswa User Mahasiswa .....	36
Gambar 3.14 <i>Sequence Diagram</i> Hapus Data Mahasiswa .....	37
Gambar 3.15 <i>Sequence Diagram</i> Tambah Data IPK .....	38
Gambar 3.16 <i>Sequence Diagram</i> Ubah Data IPK .....	39
Gambar 3.17 <i>Sequence Diagram</i> Hapus Data IPK .....	40
Gambar 3.18 <i>Sequence Diagram</i> Tambah Data Jadwal Kuliah .....	40
Gambar 3.19 <i>Sequence Diagram</i> Ubah Data Jaduial Kuliah .....	41

Gambar 3.20 <i>Sequence Diagram</i> Hapus Data Jadwal Kuliah .....	42
Gambar 3.21 <i>Sequence Diagram</i> Tambah Data Jadwal Ujian .....	43
Gambar 3.22 <i>Sequence Diagram</i> Ubah Data Jadwal Ujian .....	44
Gambar 3.23 <i>Sequence Diagram</i> Hapus Data Jadwal Ujian .....	45
Gambar 3.24 <i>Sequence Diagram</i> Tambah Data User .....	46
Gambar 3.25 <i>Sequence Diagram</i> Ubah Data User .....	47
Gambar 3.26 <i>Sequence Diagram</i> Hapus Data User .....	48
Gambar 3.27 <i>Sequence Diagram</i> Login User .....	49
Gambar 3.28 <i>Activity Diagram</i> Tambah Data .....	50
Gambar 3.29 <i>Activity Diagram</i> Ubah Data .....	51
Gambar 3.30 <i>Activity Diagram</i> Hapus Data .....	53
Gambar 3.31 Relasi Antar Tabel .....	58
Gambar 3.32 Tampilan Utama ( <i>Interface</i> ) Sistem .....	59
Gambar 3.33 Panel Pertama Input Koneksi .....	63
Gambar 3.34 Panel Kedua Input Koneksi .....	64
Gambar 3.35 Input Data Mahasiswa .....	65
Gambar 3.36 Input Data User .....	66
Gambar 3.37 Input Data IPK .....	66
Gambar 3.38 Input Data Jadwal Kuliah .....	67
Gambar 3.39 Input Data Jadwal Ujian .....	68
Gambar 3.40 Ubah Data Sms Masuk ( <i>inbox</i> ) .....	68
Gambar 3.41 Ubah Data Sms Keluar ( <i>outbox</i> ) .....	69
Gambar 3.42 Ubah Password User .....	69

Gambar 3.43 Tampilan Halaman Login User .....	73
Gambar 3.44 Tampilan Halaman Utama User .....	74
Gambar 3.45 Tampilan Halaman <i>About</i> .....	74
Gambar 3.46 Tampilan Halaman <i>Help</i> .....	75
Gambar 3.47 Tampilan Halaman Input Data Koneksi Panel Pertama .....	75
Gambar 3.48 Tampilan Halaman Input Data Koneksi Panel Kedua .....	76
Gambar 3.49 Tampilan Halaman Input Data Mahasiswa .....	76
Gambar 3.50 Tampilan Halaman Input Data IPK .....	77
Gambar 3.51 Tampilan Halaman Input Data Jadwal Kuliah .....	77
Gambar 3.52 Tampilan Halaman Input Data Jadwal Ujian .....	78
Gambar 3.53 Tampilan Halaman Olah Data Sms .....	78
Gambar 3.54 Tampilan Halaman Ubah Password <i>User</i> .....	79
Gambar 3.55 Tampilan Halaman Ubah Data Sms .....	79
Gambar 4.1 <i>Dialog Box</i> Kesalahan Login .....	86
Gambar 4.2 <i>Dialog Box</i> Kesalahan Pemilihan Data .....	86
Gambar 4.3 <i>Dialog Box</i> Kesalahan Input Data .....	87
Gambar 4.4 <i>Dialog Box</i> Kesalahan Input Data Kosong .....	87
Gambar 4.5 <i>Dialog Box</i> Kesalahan Panjang Karakter .....	87
Gambar 4.6 Antarmuka Masukan Data Login .....	93
Gambar 4.7 Antarmuka Masukan Data Mahasiswa .....	95
Gambar 4.8 <i>Output</i> Masukan Data Mahasiswa .....	95
Gambar 4.9 Antarmuka Masukan Data IPK .....	96
Gambar 4.10 <i>Output</i> Masukan Data IPK .....	96

Gambar 4.11 Antarmuka Masukan Data Jadwal Kuliah .....	97
Gambar 4.12 <i>Output</i> Masukan Data Jadwal Kuliah .....	97
Gambar 4.13 Antarmuka Masukan Data Jadwal Ujian .....	98
Gambar 4.14 <i>Output</i> Masukan Data Jadwal Ujian .....	98
Gambar 4.15 Antarmuka Masukan Data <i>User</i> .....	99
Gambar 4.16 <i>Output</i> Masukan Data <i>User</i> .....	99
Gambar 4.17 Antarmuka Masukan Data <i>Setting</i> Koneksi Panel Pertama .....	100
Gambar 4.18 Antarmuka Masukan Data <i>Setting</i> Koneksi Panel Kedua .....	100
Gambar 4.19 <i>Output</i> Masukan Data <i>Setting</i> Koneksi .....	100



# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Kemajuan teknologi berkembang begitu pesat, terutama teknologi komputer dan telepon seluler. Seiring dengan semakin banyaknya kebutuhan manusia, keterlibatan teknologi komputer dan telepon seluler semakin mendukung seluruh aktifitas manusia. Selain diminati banyak kalangan, telepon seluler juga mempunyai banyak fasilitas yang dapat digunakan seperti : SMS ( *Short Message Service* ), MMS ( *Multimedia Messaging Service* ), WAP ( *Wireless Application Protocol* ), *Multimedia Streaming*, dan sebagainya. Selain itu, telepon seluler juga mempunyai teknologi yang semakin berkembang seperti : GSM ( *Global System for Mobile Communication* ), GPRS ( *General Packet Radio Services* ), CDMA ( *Code Division Multiple Access* ), dan yang terbaru 3G ( *Third-Generation Technology* ). Oleh karena itu, telepon seluler bisa digunakan untuk berbagai kebutuhan seperti berinteraksi dengan sistem komputer, hal inilah yang mendorong adanya teknologi komputer yang berbasis SMS atau sering disebut *SMS Gateway*.

SMS ( *Short Message Service* ) adalah layanan yang disediakan oleh operator ponsel untuk mengirim dan menerima pesan singkat. SMS dinilai sangat praktis, murah, dan efisien. Selain untuk mengirim pesan antar pengguna ponsel, SMS juga

cocok untuk diterapkan dan berinteraksi dengan suatu sistem informasi berbasis komputer.

Mengakses dan mengubah data pribadi mahasiswa adalah salah satu hal yang sangat penting dilakukan seorang mahasiswa setiap semester. Karena perubahan tersebut sebagai syarat untuk seorang mahasiswa melakukan registrasi ulang. Selain itu perubahan data pribadi tersebut dibutuhkan pihak universitas untuk melakukan pendataan seluruh mahasiswanya, dan memantau latar belakang mahasiswa yang bersangkutan.

Salah satu permasalahan yang dihadapi mahasiswa adalah keluhan-keluhan ketika ingin mengakses ataupun mengubah data pribadinya, seperti : minimnya biaya, antri warnet, dan lain-lain. Menyadari adanya beberapa keluhan tersebut maka perlu dirancang suatu sistem untuk mengakses dan mengubah data pribadi mahasiswa berbasis SMS. Sistem berbasis SMS yang menggunakan media telepon seluler dimaksudkan untuk dapat mempermudah sekaligus menghemat biaya mahasiswa dalam melakukan pengaksesan atau perubahan data pribadinya.

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang di atas dapat dirumuskan permasalahan yang akan diselesaikan adalah bagaimana membangun sebuah sistem untuk mengakses dan mengubah informasi data pribadi mahasiswa berbasis SMS sesuai dengan kebutuhan mahasiswa yang bersangkutan dan mampu menghasilkan informasi data yang lebih cepat dan efisien.

### 1.3 Batasan Masalah

Dalam membangun sistem tersebut terdapat beberapa batasan masalah yang diperlukan, diantaranya :

1. Sistem yang akan dibangun merupakan sistem untuk mengakses dan mengubah data pribadi mahasiswa berbasis SMS.
2. Data-data pribadi yang dapat diakses dan diubah berupa alamat yk, alamat asal, telp, password user, ipk, jadwal kuliah, dan jadwal ujian.
3. Output sistem adalah data pribadi yang terbaru.
4. Sistem ini hanya diperuntukkan untuk mengubah data pribadi saja.
5. Program aplikasi yang digunakan untuk membangun sistem tersebut menggunakan NetBeans dengan bahasa pemrograman Java.

### 1.4 Tujuan Penelitian

Tujuan dilakukannya penelitian ini adalah :

1. Membangun suatu sistem untuk mengakses dan mengubah data pribadi mahasiswa berbasis SMS agar informasi data lebih cepat dan efisien.
2. Memperdalam ilmu pengetahuan tentang perangkat lunak menggunakan java dalam membangun suatu sistem yang berbasis SMS ( *SMS Gateway* ).
3. Menambah perbendaharaan perangkat lunak yang sudah ada, terutama bahasa pemrograman java.

## **1.5 Manfaat Penelitian**

Sistem ini diharapkan akan dapat dimanfaatkan untuk :

1. Membantu dan mempermudah mahasiswa dalam mengakses dan mengubah data pribadinya setiap semester.
2. Biaya dalam mengakses dan mengubah data pribadi mahasiswa menjadi lebih murah dan terjangkau. Selain itu, waktu pengaksesan juga menjadi lebih cepat.

## **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 *SMS Gateway* untuk menentukan input serta output yang efektif.

## 2. Studi Pustaka

Metode ini digunakan untuk mendapatkan informasi tambahan berupa referensi buku-buku literatur 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 uraian beberapa bab untuk mempermudah pembahasan, diantaranya :

### **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. Tinjauan pustaka berisi penjelasan singkat mengenai penelitian-penelitian sebelumnya. Sedangkan, teori dasar mencakup penjelasan secara terperinci mengenai teori-teori yang berhubungan dengan *SMS Gateway* dan Java yang digunakan sebagai landasan untuk pemecahan masalah.

### **BAB III METODOLOGI**

Bagian ini memuat uraian langkah-langkah penyelesaian masalah, uraian tentang metode analisis kebutuhan perangkat lunak yang dipakai, hasil analisis kebutuhan perangkat lunak, perancangan antarmuka, metode dan hasil perancangan, implementasi perangkat lunak, dan hasil kinerja perangkat lunak.

### **BAB IV HASIL DAN PEMBAHASAN**

Bab ini membahas tentang analisis kinerja dari perangkat lunak. Bagian hasil memuat tentang hasil dan bagaimana hasil tersebut dicapai. Sedangkan bagian pembahasan memuat pembahasan hasil yang diperoleh. Pembahasan dapat berupa uraian tentang mengapa hasil diperoleh, kelebihan (keunggulan) dan kelemahan sistem yang dibuat.

### **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 yang berbasis SMS.

## BAB II

### LANDASAN TEORI

#### 2.1 SMS (*Short Message Service*)

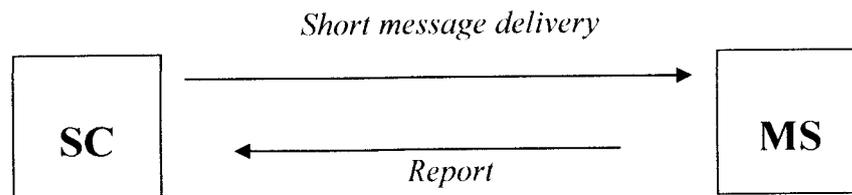
##### 2.1.1 Pengertian SMS (*Short Message Service*)

*Short Message Service* (SMS) merupakan salah satu fitur GSM yang dikembangkan dan distandarisasi oleh ETSI (*European Telecommunications Standards Institute*) untuk mengirim atau menerima pesan singkat [KOM05a]. SMS adalah layanan untuk mengirim dan menerima pesan tertulis (teks) dari maupun kepada perangkat bergerak (*mobile device*).

Dalam pengiriman dan penerimaan pesan sms terdapat dua mode, yaitu : mode teks dan mode Protocol Data Unit (PDU). Mode teks adalah format pesan dalam bentuk teks asli, sebenarnya mode teks merupakan pengkodean dari mode PDU. Sedangkan mode PDU adalah format pesan dalam bentuk oktet heksadesimal dan oktet semidesimal dengan panjang mencapai 160 (7 bit) atau 140 (8 bit) karakter. Pada pengiriman pesan terdiri dari 2 jenis *mobile* [KOM05a], yaitu :

1. SMS PDU Penerima atau *Short Message Mobile Terminated* (SM MT), merupakan 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* dengan mekanisme tertentu. Perhatikan gambar 2.1 :



**Gambar 2.1** *Short Message Mobile Terminated*

Pada prinsipnya pesan yang diterima dari SMSC (SMS centre) masih dalam format PDU setelah itu terminal handphone yang menerima pesan akan melakukan pengkodean menjadi teks, yang disebut proses *decode*. Cara pengkodean format PDU sudah diatur dan distandarkan oleh ETSI. Format PDU dari sms penerima adalah :

SCA | PDU Type | OA | PID | DCS | SCTS | UDL | UD

Contoh : menerima pesan dari 628122888374 dengan isi pesan "hellohello" tanggal 6 januari 2004 pukul 16:22 WIB. Maka format PDU pesan tersebut adalah :

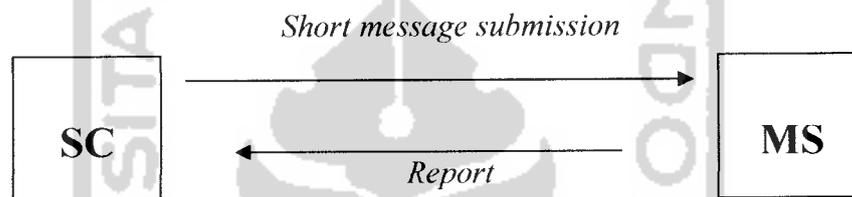
06912618010000040C912618228838470000401060612202820AE8329BFD4697  
D9EC37

Keterangan dari format pesan tersebut yaitu :

1. SCA (*Service Centre Address*) : Alamat nomor dari SMSC
2. PDU Type : Nilai *default* PDU Type untuk sms penerima yaitu 04 hexa
3. OA (*Originator Address*) : Alamat nomor dari pengirim
4. PID (*Protocol Identifier*) : Tipe atau format dari cara pengiriman pesan
5. DCS (*Data Coding Scheme*) : Rencana dari pengkodean data
6. SCTS (*Service Centre Time Stamp*) : Waktu penerimaan pesan

7. UDL (*User Data Length*) : Panjang pesan yang diterima dalam bentuk teks
8. UD (*User Data*) : Pesan yang diterima dalam format heksadesimal

2. SMS PDU Pengirim atau *Short Message Mobile Originated* (SM MO), yaitu kemampuan 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*. Perhatikan gambar 2.2 :



**Gambar 2.2** *Short Message Mobile Originated*

Sebelum melakukan pengiriman pesan sms, terminal atau handphone akan melakukan perubahan dari format teks menjadi menjadi format PDU, yang disebut proses *encodec*. Skema dari format PDU pengirim adalah :

SCA | PDU Type | MR | DA | PID | DCS | VP | UDL | UD

Contoh : mengirim pesan sms ke nomor 628122898840 dengan isi pesan “pesan pendek” dengan batas waktu pengiriman 5 hari. Maka format PDU pesan tersebut adalah :

0011000C912618229888040000AB0CD0F23CEC06C1CB6E72790D

Keterangan dari format pesan tersebut yaitu :

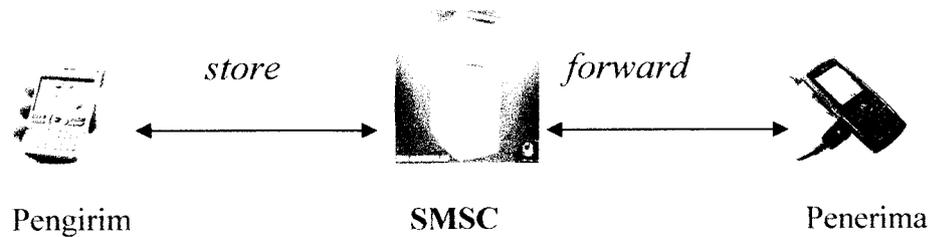
1. SCA (*Service Centre Address*) : Alamat nomor dari SMSC

2. PDU Type : Nilai *default* PDU Type untuk sms pengirim yaitu 11 hexa
3. MR (*Message Reference*) : Acuan dari pengaturan pesan sms
4. DA (*Destination Address*) : Alamat nomor tujuan
5. PID (*Protocol Identifier*) : Tipe atau format dari cara pengiriman pesan
6. DCS (*Data Coding Scheme*) : Rencana dari pengkodean data
7. VP (*Validity Period*) : Lama waktu pesan sms disimpan di SMSC
8. UDL (*User Data Length*) : Panjang pesan yang dikirim dalam bentuk teks
9. UD (*User Data*) : Pesan yang diterima dalam format heksadesimal

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

Dengan adanya suatu *server* SMS (*SMS-centre* / SMSC), pengguna dapat mengetahui status dari SMS yang dikirim, apakah telah sampai atau gagal diterima oleh ponsel tujuan. Apabila ponsel tujuan dalam keadaan aktif dan menerima SMS yang dikirim, maka ia akan mengirim kembali pesan konfirmasi ke SMSC yang menyatakan bahwa SMS telah diterima, kemudian SMSC mengirimkan kembali status tersebut kepada pengirim. Akan tetapi, jika ponsel tujuan dalam keadaan tidak aktif atau di luar jangkauan, SMS yang dikirimkan akan disimpan pada SMSC sampai periode validitas terpenuhi. Jika periode validitas waktu terlewati maka SMS itu akan dihapus dari SMSC dan tidak dikirimkan ke ponsel tujuan. Disamping itu, SMSC juga akan mengirim pesan informasi ke ponsel pengirim yang menyatakan pesan yang dikirim belum diterima atau gagal [KOM05a].

Mekanisme cara kerja SMS dapat dilihat pada gambar 2.3 :



**Gambar 2.3** Mekanisme Cara Kerja SMS

### 2.1.3 SMS Gateway

SMS *Gateway* merupakan salah satu perkembangan fungsi yang dimiliki SMS. Secara umum SMS *Gateway* adalah sebuah sistem yang dipergunakan untuk memudahkan seseorang atau sebuah perusahaan mengirimkan pesan SMS yang sama dalam waktu yang bersamaan pada banyak orang. Selain itu, semakin berkembangnya fungsi SMS, SMS *Gateway* juga bisa dimanfaatkan untuk keperluan lain seperti melakukan polling, transaksi dengan sebuah sistem, pemantauan, dan sebagainya [PCM07].

Selain menghemat waktu dan biaya, kebutuhan untuk SMS *Gateway* juga tidak terlalu berlebihan dan juga fleksibel karena hanya menggunakan sebuah PC dengan menggunakan modem sebuah ponsel, kartu GSM, dan sebuah program yang bisa dibangun sendiri khususnya menggunakan bahasa pemrograman JAVA yang berfungsi sebagai pengirim pesan [YON07].

## 2.2 Access and Terminals (AT) Command

AT *Command* adalah perintah-perintah yang digunakan dalam komunikasi dengan Serial port. Dengan AT *command*, vendor dari handphone yang digunakan, kekuatan sinyal, membaca pesan yang ada pada SIM *Card*, mengirim pesan, mendeteksi pesan SMS baru yang masuk secara otomatis, menghapus pesan pada SIM *card*, dan masih banyak lagi fungsi lainnya yang dapat dilihat dengan AT *command* [KOM05a].

AT *Command* sebenarnya hampir sama dengan perintah > (*prompt*) pada DOS. Perintah-perintahnya digunakan untuk penulisan ke port komputer, dan diawali dengan kata AT, kemudian diikuti karakter lainnya yang memiliki fungsi sendiri-sendiri. Selain digunakan untuk penulisan ke port, AT *Command* juga dapat digunakan untuk penulisan ke modem.

Contoh perintah AT *Command* :

- ATEI : mengetahui kondisi port jika siap untuk berkomunikasi
- AT+CGMI : perintah untuk mengetahui vendor ponsel yang digunakan
- AT+CMGR : perintah untuk membaca salah satu SMS yang ada pada SIM *card*
- AT+COPS : perintah untuk memilih dan mendaftarkan jaringan operator GSM

Untuk penulisan data ke ponsel, maka ponsel terlebih dahulu harus dihubungkan dengan suatu kabel data yang tersedia ke port serial pada komputer. AT *Command* yang digunakan pada ponsel mengikuti standar dari ETSI GSM 07.05. beberapa AT *Command* yang dapat digunakan untuk menangani pesan SMS pada ponsel terdapat pada tabel 2.1 berikut ini [NOK01] :

**Tabel 2.1** AT Command untuk SMS

<b>Perintah</b>	<b>Fungsi</b>	<b>Tipe Perintah</b>
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

### 2.3 Mobile atau Wireless

Telepon *mobile nirkabel* adalah suatu turunan atau padanan dari *wireless mobile phone*, *wireless handphone*, *cellular phone* atau telepon selular yang dihubungkan dengan jaringan. Oleh sebab itu, disebut juga sebagai *mobile nirkabel* atau tanpa kabel yang dapat memudahkan terjadinya komunikasi antar ponsel.

### 2.3.1 Perkembangan Teknologi Jaringan Wireless

Perkembangan teknologi *wireless* dibagi menjadi beberapa generasi teknologi, diantaranya :

#### 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, Metrocel, 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 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.

#### 4) 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*).

#### 5) Generasi Kelima (5G)

Generasi ini belum didefinisikan apa sebenarnya yang akan ditawarkan, ini akan merupakan sesuatu yang lebih maju dan canggih dari generasi sebelumnya.

### 2.3.2 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 com* [ADY07]. 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 USB pada komputer dan kabel data

yang tersedia harus adanya *compatible* antara ponsel yang ada dengan kecocokan kabel data itu sendiri.

#### 2.4 UML (*Unified Modeling Language*)

UML (*Unified Modeling Language*) pertama kali diperkenalkan pada tahun 1990-an ketika **Grady Booch** dan **Ivar Jacobson** dan **James Rumbaugh** mulai mengadopsi ide-ide serta kemampuan-kemampuan tambahan dari masing-masing metodonya dan berusaha membuat metodologi terpadu yang kemudian dinamakan UML (*Unified Modeling Language*). UML merupakan metode pengembangan perangkat lunak atau sistem informasi dengan metode grafis yang relatif mudah dipahami [NUG05a].

Secara umum UML (*Unified Modeling Language*) merupakan bahasa untuk visualisasi, spesifikasi, konstruksi, serta dokumentasi. UML ini digunakan oleh para pengembang sebagai sarana untuk mengkomunikasikan idenya kepada para pemrogram serta calon pengguna suatu sistem atau perangkat lunak. Secara khusus UML menspesifikasikan langkah-langkah penting dalam pengambilan keputusan analisis, perancangan, serta implementasi dalam sistem yang sangat bernuansa perangkat lunak (*software intensive system*) [NUG05a]. UML bukanlah suatu bahasa pemrograman tetapi merupakan model-model yang tercipta berhubungan langsung dengan berbagai macam bahasa pemrograman, sehingga memungkinkan melakukan pemetaan (*mapping*) langsung dari model-model yang dibuat dengan UML ke bahasa-bahasa pemrograman berorientasi objek seperti JAVA.

## BAB III

### METODOLOGI

#### 3.1 Analisis Kebutuhan Perangkat Lunak

##### 3.1.1 Metode Analisis

Sistem untuk mengakses dan mengubah data pribadi ini dirancang menggunakan konsep *SMS Gateway*. Untuk melihat aliran proses sistem ini yang mencakup *Input*, proses sistem, dan *output* maka digambarkan menggunakan UML (*Unified Modeling Language*). Pada UML akan digunakan notasi-notasi grafis yang akan menggambarkan aliran sistem dimana akan membantu dalam mengkomunikasikan fungsi sistem kepada pemakai maupun sebagai alat bantu pemrogram dalam membangun sistem.

UML (*Unified Modeling Language*) digunakan untuk mendefinisikan langkah-langkah penting dalam pengambilan keputusan analisis, perancangan, serta implementasi dalam proses pembuatan ataupun pengembangan perangkat lunak (*Software*).

##### 3.1.2 Hasil Analisis

Dari data pribadi yang diperoleh melalui survey selama penelitian dan setelah dilakukan proses analisis maka kebutuhan sistem terdiri dari kebutuhan proses, kebutuhan *Input* dan kebutuhan keluaran.

### 3.1.2.1 Analisis Kebutuhan Proses

Kebutuhan proses dalam sistem pengaksesan data pribadi berbasis SMS ini antara lain :

- Proses penyambungan antara ponsel dengan sistem
- Proses pengolahan data pribadi mahasiswa
- Proses pengaksesan data pribadi mahasiswa
- Proses perubahan data pribadi mahasiswa
- Proses pengolahan database data pribadi dan SMS terhadap sistem
- Proses pengiriman SMS balik data pribadi yang terbaru

### 3.1.2.2 Analisis Kebutuhan *Input*

*Input* atau masukan dari sistem untuk mengakses dan mengubah data pribadi mahasiswa berbasis SMS ini, terdiri dari 2 karakteristik masukan yaitu *admin* dan *pengguna*.

#### a) *Input Admin*

*Input* admin adalah suatu masukan yang diberikan oleh admin berupa data mahasiswa baru termasuk hak *edit* dan menghapus data tersebut, variabel data pribadi yang digunakan, informasi database baik data mahasiswa ataupun SMS yang akan digunakan, atau password admin.

#### 1. *Input Data Mahasiswa*

Masukan daftar data mahasiswa baru.

## 2. Input Variabel Data pribadi

Masukan variabel data pribadi yang digunakan sistem.

## 3. Input Informasi Database

Masukan informasi baru data database yang akan digunakan.

## 4. Password Baru

Masukan untuk mengubah password admin.

### b) Input Pengguna

*Input* pengguna adalah masukan yang diberikan oleh seorang mahasiswa berupa SMS melalui ponsel, proses apa yang diinginkan pengguna, mengakses atau mengubah data pribadi mahasiswa. Dalam mengakses atau mengubah data pribadi, mahasiswa tersebut melakukan proses input atau ubah terhadap data pribadinya masing-masing.

#### 1. Akses Data Pribadi

Mengakses data pribadinya sesuai dengan kebutuhan dengan SMS

#### 2. Ubah Data Pribadi

Mengubah data pribadinya menjadi data pribadi terbaru dengan SMS

### 3.1.2.3 Analisis Kebutuhan Output

Data keluaran (*Output*) yang diperoleh dari sistem pengaksesan dan pengubahan data pribadi berbasis SMS ini adalah berupa data pribadi mahasiswa terbaru berupa SMS balik dari sistem.

### 3.1.3 Kebutuhan Antar Muka

Perancangan antar muka dengan menggunakan NetBeans 5.5 merupakan pilihan yang tepat untuk mengimplementasikan sistem ini, selain karena tampilan yang memudahkan bagi penggunanya untuk menggunakan sistem ini, juga antar muka yang menggunakan NetBeans ini lebih dapat mendukung membuat sistem yang berdasar *WYSWYG (What You See What You Get)* untuk tampilan grafisnya sebagai visualisasi untuk menggambarkan kejadian yang sesungguhnya. Kelebihan lain dari tampilan antar muka menggunakan NetBeans ini adalah untuk menghilangkan kesulitan dari pengetikan perintah-perintah JAVA menggunakan manual teks.

### 3.1.4 Analisis Kebutuhan Perangkat Lunak

Perangkat lunak dalam pembuatan suatu sistem merupakan perintah-perintah yang digunakan agar dapat berinteraksi dengan perangkat keras komputer. Dalam penelitian ini penyusun menggunakan perangkat lunak JAVA. Karena merupakan salah satu perangkat lunak yang dapat mengatasi permasalahan tentang sistem informasi yang berbasis SMS (*SMS Gateway*) khususnya dalam sistem untuk mengakses dan mengubah data pribadi mahasiswa.

### 3.1.5 Analisis Kebutuhan Perangkat Keras

Penggunaan sistem komputer dan ponsel dalam teknologi yang berkembang saat ini merupakan alat bantu dalam menyelesaikan tugas-tugas,

pekerjaan, atau masalah yang dihadapi setiap harinya, karena dengan sistem komputer semua masalah tersebut dapat terselesaikan dengan cepat dan efisien.

Komputer terdiri dari perangkat keras dan perangkat lunak yang saling berkomunikasi untuk menyelesaikan tugas tertentu. Penggunaan komputer sebagai alat bantu yang sering digunakan oleh masyarakat umumnya dan mahasiswa khususnya, oleh karena itu penyusun berusaha untuk membuat salah satu kasus yang sering terjadi pada mahasiswa pada umumnya yaitu pengaksesan dan perubahan data pribadi mahasiswa berbasis komputer dan ponsel yaitu menggunakan SMS.

Perangkat keras yang digunakan pada sistem untuk pengaksesan dan perubahan data pribadi mahasiswa berbasis SMS ini adalah :

- a. Intel P4 2.0Ghz
- b. Memori 256 MB
- c. VGA 32 MB
- d. Hardisk 40 GB
- e. Monitor resolusi 1152 x 864
- f. Mouse dan Keyboard
- g. Ponsel penghubung (Motorola)

## **3.2 Perancangan Perangkat Lunak**

### **3.2.1 Metode Perancangan**

Dalam membangun perangkat lunak Sistem untuk Mengakses atau Mengubah Data Pribadi Mahasiswa ini digunakan metode perancangan berorientasi objek. Perancangan berorientasi objek pada dasarnya menekankan pada kemudahan perawatan serta pengembangan sistem lebih lanjut nantinya.

Selain itu, perancangan sistem ini menggunakan metode perancangan berorientasi objek menggunakan UML (*Unified Modelling Language*) yang digunakan untuk visualisasi, spesifikasi, pembentukan, dan pendokumentasian informasi proses yang digunakan atau dihasilkan dalam pembuatan sistem.

### **3.2.2 Hasil Perancangan**

Berdasarkan analisis yang telah dilakukan dalam pembuatan sistem ini maka dapat dilihat apa yang menjadi masukan, proses, keluaran, atau antarmuka sistem yang dibuat agar sesuai dengan apa yang diharapkan.

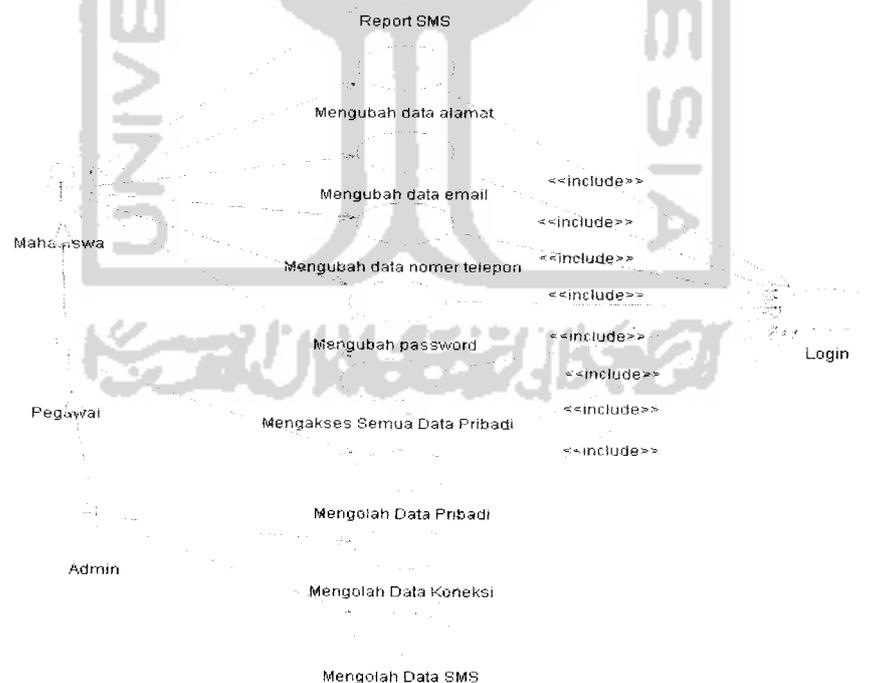
#### **3.2.2.1 Perancangan UML (*Unified Modelling Language*)**

##### **3.2.2.1.1 Use Case Diagram**

Diagram yang bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) sebuah sistem dengan suatu sistem tersendiri melalui sebuah cerita bagaimana sebuah sistem dipakai disebut *Use Case Diagram*. *Use case diagram* digunakan sebagai alat bantu untuk menstimulasi pengguna potensial untuk mengatakan tentang sistem dari sudut pandangnya, *Use case* mewakili pandangan di luar sistem. *Use case* terdiri dari sebuah aktor dan

interaksi yang dilakukannya, aktor tersebut dapat berupa manusia, perangkat keras, sistem lain, ataupun yang berinteraksi dengan sistem yang akan kita kembangkan. Dalam sistem ini aktor(pengguna)nya berupa manusia yang melakukan interaksi terhadap sistem melalui sms.

Pada sistem untuk mengakses atau mengubah data pribadi ini use case diagram menggambarkan seorang aktor(pengguna) sistem yaitu mahasiswa yang berinteraksi dengan sistem dengan melakukan permintaan berupa proses pengaksesan data pribadinya, dengan syarat user tersebut harus sudah daftar dan *login* terlebih dahulu terhadap sistem. Sedangkan user admin dapat melakukan olah-olah data berupa koneksi, sms, serta keseluruhan data pribadi mahasiswa melalui sistem secara langsung. Seperti terlihat pada gambar 3.1.



**Gambar 3.1** Use Case Diagram

Pada Use Case Diagram diatas user Mahasiswa dapat berinteraksi melalui sms dengan sistem apabila mahasiswa yang bersangkutan sudah terdaftar menjadi anggota. Mahasiswa dapat mengakses semua data pribadinya dan mengubah beberapa data yang memungkinkan bisa diubah, sedangkan user admin bisa mengakses dan mengubah semua data induk mahasiswa apabila terjadi kesalahan pengisian pada awal pengisian data.

#### 3.2.2.1.2 Class Diagram

*Class* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* dapat berhubungan dengan yang lain melalui berbagai cara: *associated* (berhubungan satu sama lain), *dependent* (satu *class* tergantung/menggunakan *class* yang lain), *specialized* (satu *class* merupakan spesialisasi dari *class* lainnya), atau *package* (group bersama sebagai satu unit). Sebuah sistem biasanya mempunyai beberapa *class diagram*.

*Class diagram* merupakan diagram yang digunakan untuk menampilkan beberapa kelas, objek serta paket-paket dan hubungannya satu sama lain seperti *containment*, pewarisan, asosiasi dan lain sebagainya yang ada dalam sistem/perangkat lunak yang sedang kita kembangkan.

Dalam sistem pengaksesan data pribadi mahasiswa ini terdiri dari dua pengelompokan *class diagram* yang tiap kelompoknya mempunyai *class control* masing-masing yang merelasikan antara tampilan sistem dengan entitas objek.

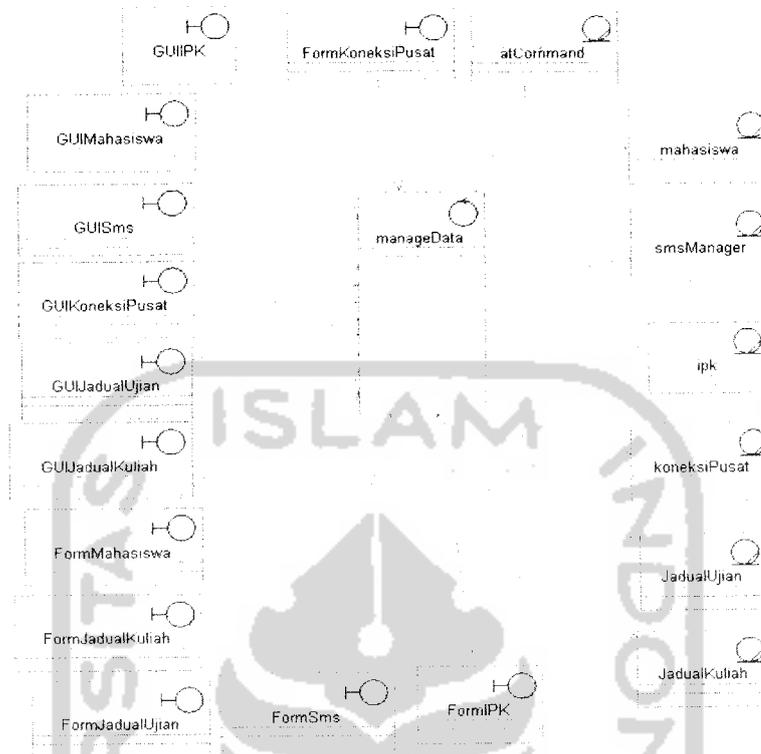


Gambar 3.3 menunjukkan *class* objek, class ini menunjukkan objek-objek yang berinteraksi dengan *class* tampilan (*boundary*) yang dihubungkan dengan sebuah *class* control yaitu *manageData*.



**Gambar 3.3** *Class Diagram* Objek dan *Control* Kelompok Pengolahan Data

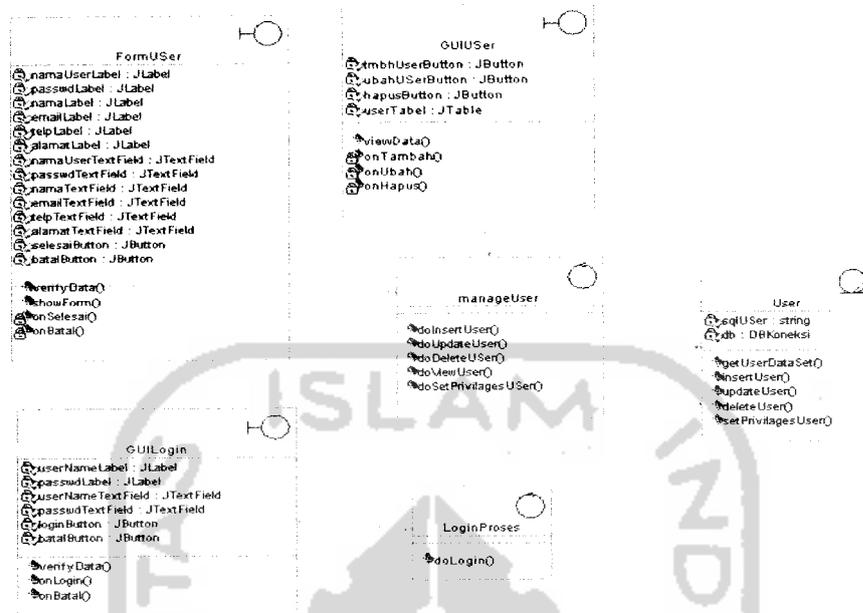
Gambar 3.4 menunjukkan relasi antar class yang berhubungan yaitu class tampilan (*boundary*) berelasi dengan class objek yang dihubungkan dengan sebuah class control.



**Gambar 3.4** Relasi *Class Diagram* Kelompok Pengolahan Data

### 3.2.2.1.3.2 Kelompok Class Diagram Pengolahan User

Dalam kelompok *class diagram* ini terdapat proses penanganan user dimana user admin merupakan user tertinggi yang berhak mengakses sekaligus mengatur sistem termasuk penanganan user mahasiswa. Dalam kelompok *class diagram* ini mengatur pembuatan user baru sekaligus pembagian hak akses terhadap sistem. Seperti terlihat pada gambar 3.5.



Gambar 3.5 Class Diagram Kelompok Pengolahan User

### 3.2.2.1.3 Sequence Diagram

Sequence diagram adalah *interaction diagram* yang memperlihatkan event-event yang berurutan sepanjang berjalannya waktu dari awal pengaksesan sistem sampai prosesnya selesai. Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem termasuk pengguna, display, dan sebagainya yang berupa *message-message* yang digambarkan terhadap waktu. Sequence diagram terdiri antara garis vertikal berupa waktu dan garis horizontal berupa objek-objek yang terkait.

Sequence diagram biasanya digunakan untuk menggambarkan skenario atau rangkaian-rangkaian yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu. Dalam sistem ini terdapat sequence diagram yang

menggambarkan aliran proses terhadap waktu dari pertama pengguna memulai sampai menghasilkan *output* yang diharapkan.

Dalam sistem pengaksesan data pribadi mahasiswa ini, *sequence diagram* dibagi berdasar pengelompokan *class diagram* diatas. Dimana setiap proses dalam *sequence diagram* ini memiliki interaksi antar objek yang berurutan.

Dalam penggambaran *sequence diagram* terdapat beberapa objek yang saling berinteraksi, yaitu :

- a. *Actor* : User yang bersangkutan.
- b. *Class Boundary* : *Class* tampilan sistem.
- c. *Class Control* : Merupakan *Class* yang merupakan penghubung antara *class* tampilan dan *class* entitas.
- d. *Clas Entity* : *Class* entitas dari sistem (Objek).

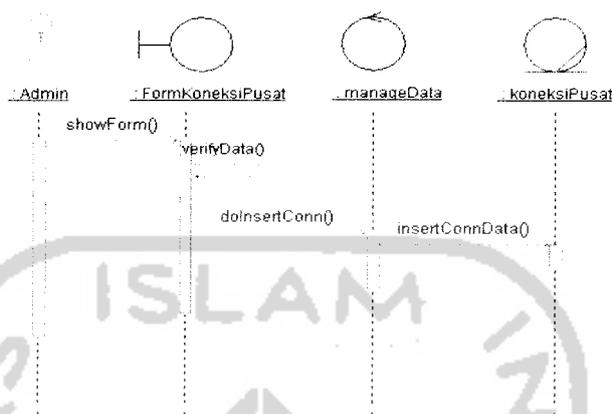
#### 3.2.2.1.3.1 Sequence Diagram Kelompok Pengolahan Data

Kelompok ini berfungsi untuk mengolah data-data sistem seperti mahasiswa serta data induknya, sms, dan koneksi. Berikut digambarkan penjelasan *sequence diagram* untuk kelompok ini.

##### 3.2.2.1.3.1.1 Tambah Data Koneksi

Aktor admin mengawali *sequence* ini dengan memanggil *method* `showForm()`, kemudian *class* `FormKoneksiPusat` melakukan instansiasi ke *class* `manageData` dan memanggil *method* `doInsertConn()` untuk melakukan proses *Input* data yang dilakukan aktor admin. Lalu, *class* `manageData` melakukan intansiasi ke *class* `koneksiPusat` dan memanggil *method* `insertConnData()`

untuk melakukan pemasukan data ke tabel koneksi. *Sequence diagram* tambah data koneksi terlihat pada gambar 3.6.

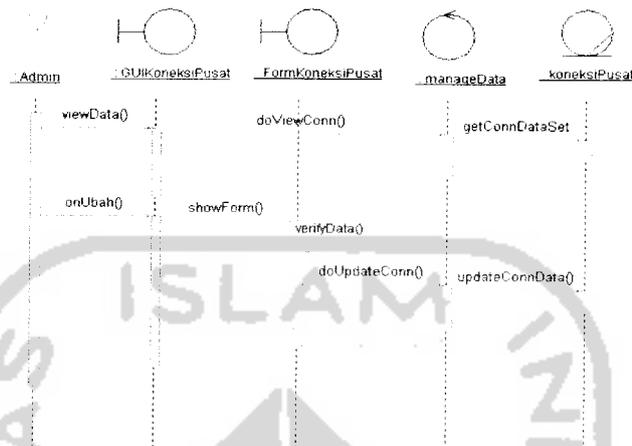


Gambar 3.6 *Sequence Diagram* Tambah Data Koneksi

#### 3.2.2.1.3.1.2 Ubah Data Koneksi

Aktor admin mengawali *sequence* ini dengan memanggil *method* `viewData()`, kemudian *class* `GUIKoneksiPusat` melakukan instansiasi ke *class* `manageData` dan memanggil *method* `doViewConn()` untuk memerintahkan agar seluruh data koneksi yang ada ditampilkan. Lalu, *class* `manageData` melakukan instansiasi ke *class* `koneksiPusat` dan memanggil *method* `getConnDataSet()` untuk mengambil semua data koneksi. Kemudian aktor admin memanggil *method* `onUbah()`, lalu *class* `GUIKoneksiPusat` melakukan instansiasi ke *class* `FormKoneksiPusat` dan memanggil *method* `showForm()` untuk menampilkan form. Setelah itu, *class* `FormKoneksiPusat` melakukan instansiasi ke *class* `manageData` dan memanggil *method* `doUpdateConn()` untuk memproses data yang telah diubah oleh admin. Lalu, *class* `manageData` melakukan instansiasi ke *class* `koneksiPusat` dan memanggil *method* `updateConnData()` untuk

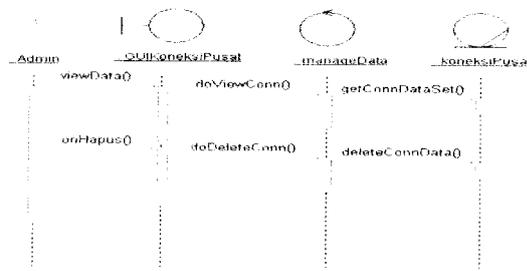
melakukan perintah *update* ke dalam basisdata tabel koneksi. *Sequence diagram* ubah data koneksi terlihat pada gambar 3.7.



Gambar 3.7 *Sequence Diagram* Ubah Data Koneksi

#### 3.2.2.1.3.1.3 Hapus Data Koneksi

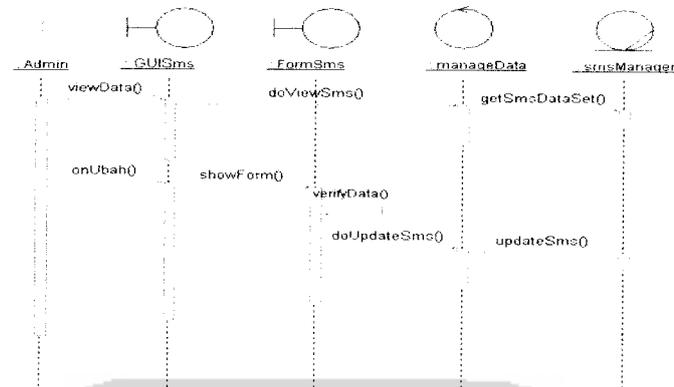
Aktor admin mengawali *sequence* ini dengan memanggil *method* `viewData()`, kemudian *class* `GUIKoneksiPusat` melakukan instansiasi ke *class* `manageData` dan memanggil *method* `doViewConn()` untuk memerintahkan agar seluruh data koneksi yang ada ditampilkan. Lalu, *class* `manageData` melakukan instansiasi ke *class* `koneksiPusat` dan memanggil *method* `getConnDataSet()` untuk mengambil semua data koneksi. Kemudian aktor admin memanggil *method* `onHapus()`. Setelah itu, *class* `GUIKoneksiPusat` melakukan instansiasi ke *class* `manageData` dan memanggil *method* `doDeleteConn()` untuk melakukan proses hapus data. Lalu, *class* `manageData` melakukan instansiasi ke *class* `koneksiPusat` dan memanggil *method* `deleteConnData()` untuk melakukan perintah *delete* ke dalam basisdata tabel koneksi. *Sequence diagram* hapus data koneksi terlihat pada gambar 3.8.



**Gambar 3.8** *Sequence Diagram* Hapus Data Koneksi

#### 3.2.2.1.3.1.4 Ubah Data Sms

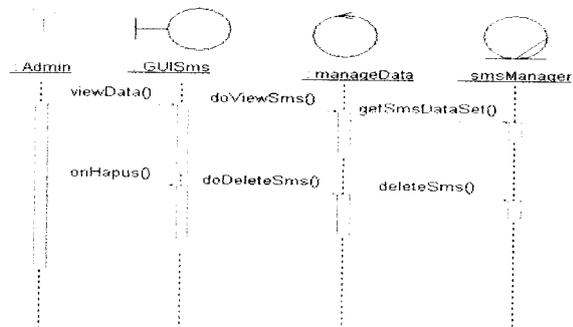
Aktor admin mengawali *sequence* ini dengan memanggil *method* `viewData()`, kemudian *class* `GUIKoneksiPusat` melakukan instansiasi ke *class* `manageData` dan memanggil *method* `doViewConn()` untuk memerintahkan agar seluruh data sms yang ada ditampilkan. Lalu, *class* `manageData` melakukan instansiasi ke *class* `smsManager` dan memanggil *method* `getSmsDataSet()` untuk mengambil semua data sms. Kemudian aktor admin memanggil *method* `onUbah()`, lalu *class* `GUIKoneksiPusat` melakukan instansiasi ke *class* `FormSms` dan memanggil *method* `showForm()` untuk menampilkan form. Setelah itu, *class* `FormSms` melakukan instansiasi ke *class* `manageData` dan memanggil *method* `doUpdateSms()` untuk memproses data sms yang telah diubah oleh admin. Lalu, *class* `manageData` melakukan instansiasi ke *class* `smsManager` dan memanggil *method* `updateSms()` untuk melakukan perintah *update* ke dalam basisdata tabel sms. *Sequence diagram* ubah data sms terlihat pada gambar 3.9.



**Gambar 3.9** *Sequence Diagram* Ubah Data Sms

#### 3.2.2.1.3.1.5 Hapus Data Sms

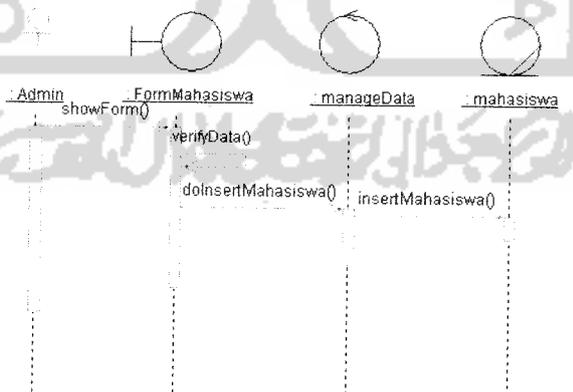
Aktor admin mengawali *sequence* ini dengan memanggil *method* `viewData()`, kemudian *class* `GUI SMS` melakukan instansiasi ke *class* `manageData` dan memanggil *method* `doViewSMS()` untuk memerintahkan agar seluruh data sms yang ada ditampilkan. Lalu, *class* `manageData` melakukan instansiasi ke *class* `smsManager` dan memanggil *method* `getSMSDataSet()` untuk mengambil semua data sms. Kemudian aktor admin memanggil *method* `onHapus()`. Setelah itu, *class* `GUI SMS` melakukan instansiasi ke *class* `manageData` dan memanggil *method* `doDeleteSMS()` untuk melakukan proses hapus data sms. Lalu, *class* `manageData` melakukan instansiasi ke *class* `smsManager` dan memanggil *method* `deleteSMS()` untuk melakukan perintah *delete* ke dalam basisdata tabel `Sms`. *Sequence diagram* hapus data sms terlihat pada gambar 3.10.



**Gambar 3.10** *Sequence Diagram* Hapus Data Sms

#### 3.2.2.1.3.1.6 Tambah Data Mahasiswa

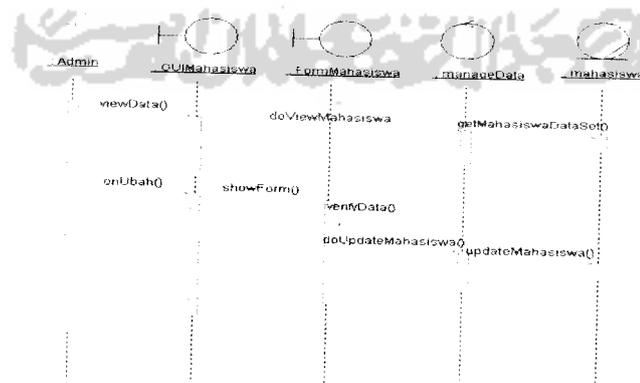
Aktor admin mengawali *sequence* ini dengan memanggil *method* `showForm()`, kemudian *class* `FormMahasiswa` melakukan instansiasi ke *class* `manageData` dan memanggil *method* `doInsertMahasiswa()` untuk melakukan proses input data yang dilakukan aktor admin. Lalu, *class* `manageData` melakukan intansiasi ke *class* `mahasiswa` dan memanggil *method* `insertMahasiswa()` untuk melakukan pemasukan data ke tabel Mahasiswa. *Sequence diagram* tambah data mahasiswa terlihat pada gambar 3.11.



**Gambar 3.11** *Sequence Diagram* Tambah Data Mahasiswa

### 3.2.2.1.3.1.7 Ubah Data Mahasiswa Untuk User Admin

Aktor admin mengawali *sequence* ini dengan memanggil *method* `viewData()`, kemudian *class* `GUIMahasiswa` melakukan instansiasi ke *class* `manageData` dan memanggil *method* `doViewMahasiswa()` untuk memerintahkan agar seluruh data induk mahasiswa yang ada ditampilkan. Lalu, *class* `manageData` melakukan instansiasi ke *class* `mahasiswa` dan memanggil *method* `getMahasiswaDataSet()` untuk mengambil semua data induk mahasiswa. Kemudian aktor admin memanggil *method* `onUbah()`, lalu *class* `GUIMahasiswa` melakukan instansiasi ke *class* `FormMahasiswa` dan memanggil *method* `showForm()` untuk menampilkan form. Setelah itu, *class* `FormMahasiswa` melakukan instansiasi ke *class* `manageData` dan memanggil *method* `doUpdateMahasiswa()` untuk memproses data induk mahasiswa yang telah diubah oleh admin. Lalu, *class* `manageData` melakukan instansiasi ke *class* `dataInduk` dan memanggil *method* `updateMahasiswa()` untuk melakukan perintah *update* ke dalam basisdata tabel Mahasiswa. *Sequence diagram* ubah data mahasiswa user admin terlihat pada gambar 3.12.

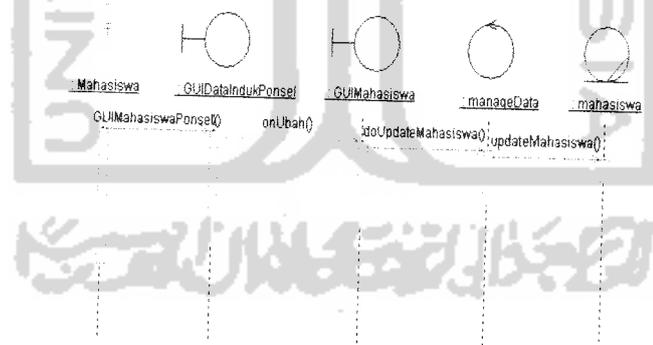


Gambar 3.12 *Sequence Diagram* Ubah Data Mahasiswa User Admin

### 3.2.2.1.3.1.8 Ubah Data Mahasiswa Untuk User Mahasiswa

*Sequence Diagram* untuk ubah data mahasiswa ini mencakup data-data yang hanya mungkin diubah oleh mahasiswa melalui Ponsel, seperti : alamat asal serta alamat tinggal, e-mail, dan telepon.

Aktor mahasiswa mengawali *sequence* ini dengan memanggil *method* `GUIMahasiswaPonsel()`, kemudian *class* `GUIMahasiswaPonsel` melakukan instansiasi ke *class* `GUIMahasiswa` dan memanggil *method* `onUbah()`. Lalu, *class* `GUIMahasiswa` melakukan instansiasi ke *class* `manageData` dan memanggil *method* `doUpdateMahasiswa()` untuk melakukan update data induk mahasiswa. Kemudian *class* `manageData` melakukan instansiasi ke *class* `dataInduk` dan memanggil *method* `updateMahasiswa()` untuk melakukan perintah *update* ke dalam basisdata tabel Mahasiswa. *Sequence diagram* ubah data mahasiswa user mahasiswa terlihat pada gambar 3.13.

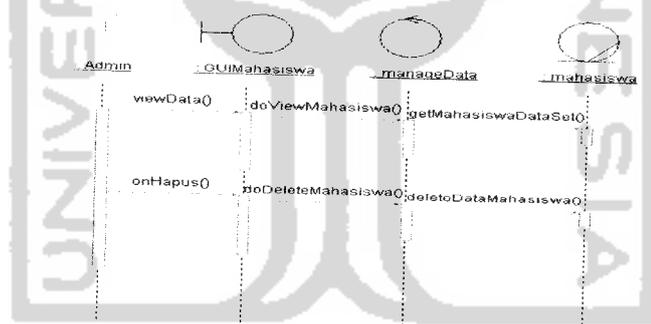


**Gambar 3.13** *Sequence Diagram* Ubah Data Mahasiswa User Mahasiswa

### 3.2.2.1.3.1.9 Hapus Data Mahasiswa

Aktor admin mengawali *sequence* ini dengan memanggil *method* `viewData()`, kemudian *class* `GUIMahasiswa` melakukan instansiasi ke *class*

`manageData` dan memanggil *method* `doViewMahasiswa()` untuk memerintahkan agar seluruh data induk mahasiswa yang ada ditampilkan. Lalu, *class* `manageData` melakukan intansiasi ke *class* `Mahasiswa` dan memanggil *method* `getMahasiswaDataSet()` untuk mengambil semua data induk mahasiswa. Kemudian aktor admin memanggil *method* `onHapus()`. Setelah itu, *class* `GUIMahasiswa` melakukan instansiasi ke *class* `manageData` dan memanggil *method* `doDeleteMahasiswa()` untuk melakukan proses hapus data induk mahasiswa. Lalu, *class* `manageData` melakukan instansiasi ke *class* `Mahasiswa` dan memanggil *method* `deleteMahasiswa()` untuk melakukan perintah *delete* ke dalam basisdata tabel `Mahasiswa`. *Sequence diagram* hapus data mahasiswa terlihat pada gambar 3.14.

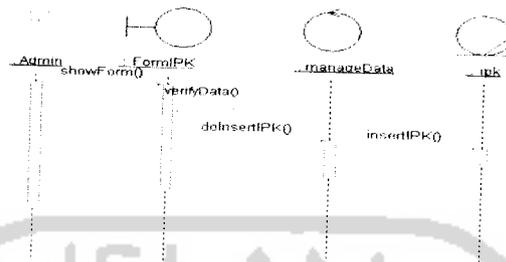


**Gambar 3.14** *Sequence Diagram* Hapus Data Mahasiswa

### 3.2.2.1.3.1.10 Tambah data IPK

Aktor admin mengawali *sequence* ini dengan memanggil *method* `showForm()`, kemudian *class* `FormIPK` melakukan instansiasi ke *class* `manageData` dan memanggil *method* `doInsertIPK` untuk melakukan proses input data yang dilakukan aktor admin. Lalu, *class* `manageData` melakukan intansiasi ke

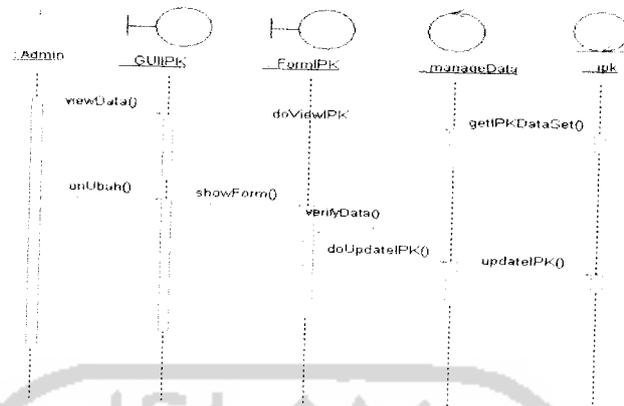
*class* IPK dan memanggil *method* `insertIPK()` untuk melakukan pemasukan data ke tabel IPK. *Sequence diagram* tambah data ipk terlihat pada gambar 3.15.



**Gambar 3.15** *Sequence Diagram* Tambah Data IPK

#### 3.2.2.1.3.1.11 Ubah data IPK

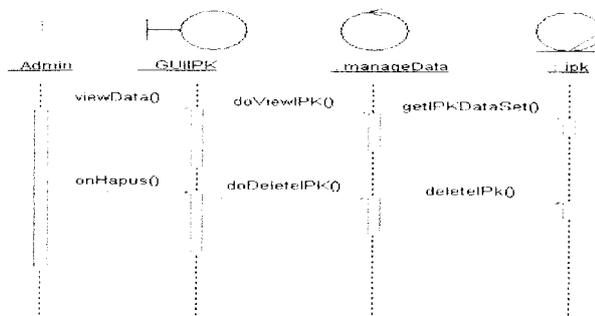
Aktor admin mengawali *sequence* ini dengan memanggil *method* `viewData()`, kemudian *class* GUIIPK melakukan instansiasi ke *class* `manageData` dan memanggil *method* `doViewIPK()` untuk memerintahkan agar seluruh data IPK yang ada ditampilkan. Lalu, *class* `manageData` melakukan instansiasi ke *class* IPK dan memanggil *method* `getIPKDataSet()` untuk mengambil semua data ipk. Kemudian aktor admin memanggil *method* `onUbah()`, lalu *class* GUIIPK melakukan instansiasi ke *class* `FormIPK` dan memanggil *method* `showForm()` untuk menampilkan form. Setelah itu, *class* `FormIPK` melakukan instansiasi ke *class* `manageData` dan memanggil *method* `doUpdateIPK()` untuk memproses data yang telah diubah oleh admin. Lalu, *class* `manageData` melakukan instansiasi ke *class* IPK dan memanggil *method* `updateIPK()` untuk melakukan perintah *update* ke dalam basisdata tabel ipk. *Sequence diagram* ubah data ipk terlihat pada gambar 3.16.



**Gambar 3.16** *Sequence Diagram* Ubah Data IPK

#### 3.2.2.1.3.1.12 Hapus data IPK

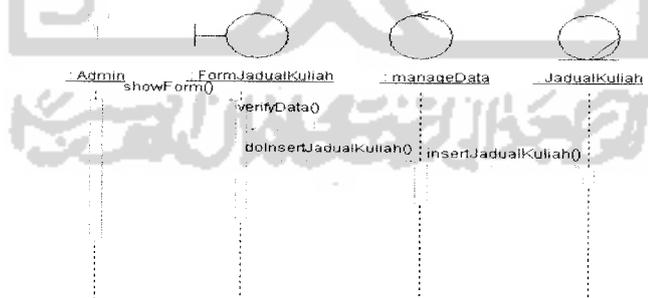
Aktor admin mengawali *sequence* ini dengan memanggil *method* `viewData()`, kemudian *class* `GUIPK` melakukan instansiasi ke *class* `manageData` dan memanggil *method* `doViewIPK()` untuk memerintahkan agar seluruh data `ipk` yang ada ditampilkan. Lalu, *class* `manageData` melakukan instansiasi ke *class* `ipk` dan memanggil *method* `getIPKDataSet()` untuk mengambil semua data `ipk`. Kemudian aktor admin memanggil *method* `onHapus()`. Setelah itu, *class* `GUIPK` melakukan instansiasi ke *class* `manageData` dan memanggil *method* `doDeleteIPK()` untuk melakukan proses hapus data `ipk`. Lalu, *class* `manageData` melakukan instansiasi ke *class* `ipk` dan memanggil *method* `deleteIPK()` untuk melakukan perintah *delete* ke dalam basisdata tabel IPK. *Sequence diagram* hapus data `ipk` terlihat pada gambar 3.17.



**Gambar 3.17** *Sequence Diagram* Hapus Data IPK

**3.2.2.1.3.1.13** Tambah data jadual kuliah

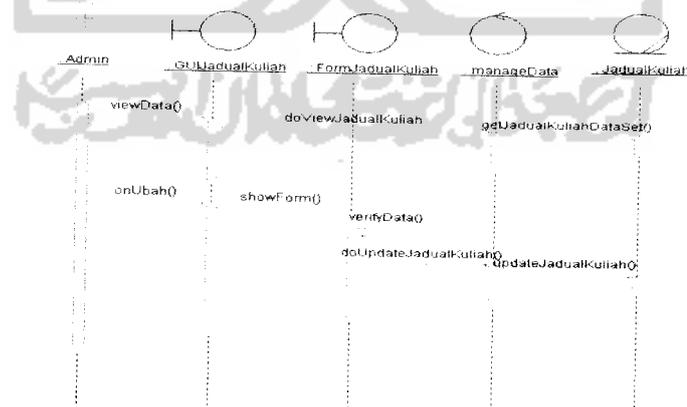
Aktor admin mengawali *sequence* ini dengan memanggil *method* `showForm()`, kemudian *class* `FormJKTinf` melakukan instansiasi ke *class* `manageData` dan memanggil *method* `doInsertJK` untuk melakukan proses input data yang dilakukan aktor admin. Lalu, *class* `manageData` melakukan intansiasi ke *class* `JKTinf` dan memanggil *method* `insertJK()` untuk melakukan pemasukan data ke tabel `JKTinf`. *Sequence diagram* tambah data jadual kuliah terlihat pada gambar 3.18.



**Gambar 3.18** *Sequence Diagram* Tambah Data Jadual Kuliah

### 3.2.2.1.3.1.14 Ubah data jadwal kuliah

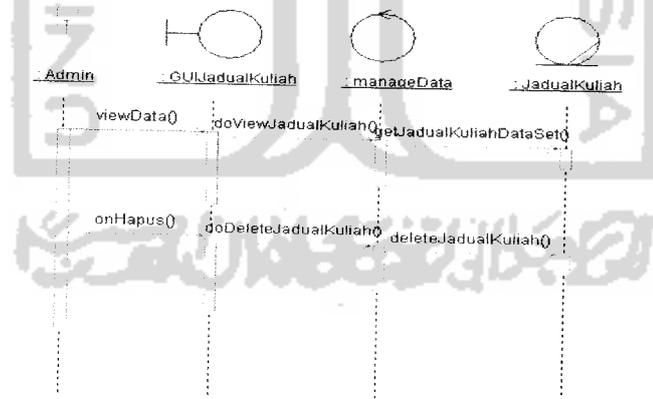
Aktor admin mengawali *sequence* ini dengan memanggil *method* `viewData()`, kemudian *class* `GUIJKTinf` melakukan instansiasi ke *class* `manageData` dan memanggil *method* `doViewJK()` untuk memerintahkan agar seluruh data jadwal kuliah yang ada ditampilkan. Lalu, *class* `manageData` melakukan intansiasi ke *class* `JKTinf` dan memanggil *method* `getJKDataSet()` untuk mengambil semua data jadwal kuliah. Kemudian aktor admin memanggil *method* `onUbah()`, lalu *class* `GUIJKTinf` melakukan instansiasi ke *class* `FormJKTinf` dan memanggil *method* `showForm()` untuk menampilkan form. Setelah itu, *class* `FormJKTinf` melakukan instansiasi ke *class* `manageData` dan memanggil *method* `doUpdateJK()` untuk memproses data yang telah diubah oleh admin. Lalu, *class* `manageData` melakukan instansiasi ke *class* `JKTinf` dan memanggil *method* `updateJK()` untuk melakukan perintah *update* ke dalam basisdata tabel `jktinf`. *Sequence diagram* ubah data jadwal kuliah terlihat pada gambar 3.19.



Gambar 3.19 *Sequence Diagram* Ubah Data Jadwal Kuliah

### 3.2.2.1.3.1.15 Hapus data jadwal kuliah

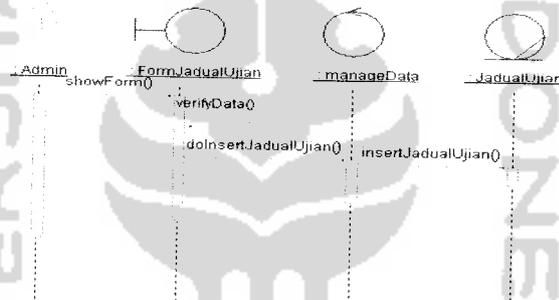
Aktor admin mengawali *sequence* ini dengan memanggil *method* `viewData()`, kemudian *class* `GUIJKTinf` melakukan instansiasi ke *class* `manageData` dan memanggil *method* `doViewJK()` untuk memerintahkan agar seluruh data jadwal kuliah yang ada ditampilkan. Lalu, *class* `manageData` melakukan instansiasi ke *class* `JKTinf` dan memanggil *method* `getJKDataSet()` untuk mengambil semua data jadwal kuliah. Kemudian aktor admin memanggil *method* `onHapus()`. Setelah itu, *class* `GUIJKTinf` melakukan instansiasi ke *class* `manageData` dan memanggil *method* `doDeleteJK()` untuk melakukan proses hapus data jadwal kuliah. Lalu, *class* `manageData` melakukan instansiasi ke *class* `JKTinf` dan memanggil *method* `deleteJK()` untuk melakukan perintah *delete* ke dalam basisdata tabel `jktinf`. *Sequence diagram* hapus data jadwal kuliah terlihat pada gambar 3.20.



Gambar 3.20 *Sequence Diagram* Hapus Data Jadwal Kuliah

### 3.2.2.1.3.1.16 Tambah data jadual ujian

Aktor admin mengawali *sequence* ini dengan memanggil *method* `showForm()`, kemudian *class* `FormJUTinf` melakukan instansiasi ke *class* `manageData` dan memanggil *method* `doInsertJU` untuk melakukan proses input data yang dilakukan aktor admin. Lalu, *class* `manageData` melakukan intansiasi ke *class* `JUTinf` dan memanggil *method* `insertJU()` untuk melakukan pemasukan data ke tabel `JUTinf`. *Sequence diagram* tambah data jadual ujian terlihat pada gambar 3.21.

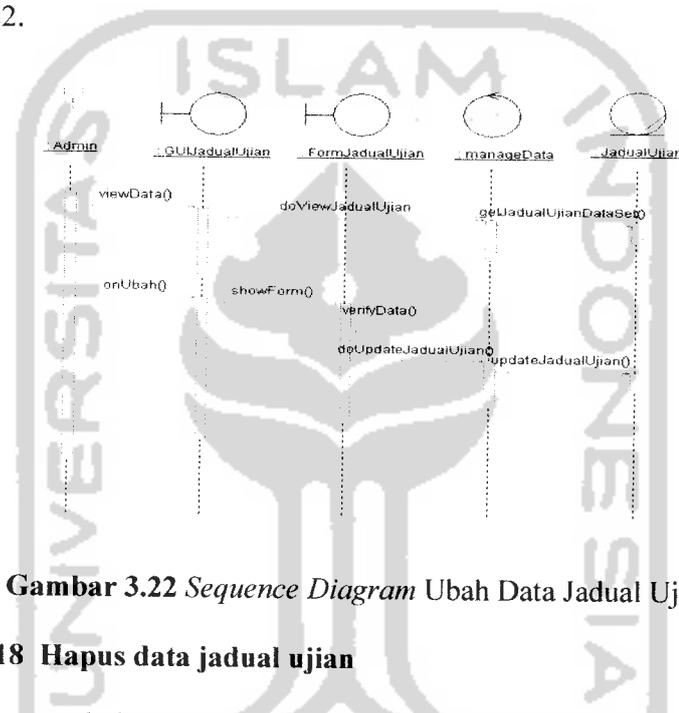


Gambar 3.21 *Sequence Diagram* Tambah Data Jadual Ujian

### 3.2.2.1.3.1.17 Ubah data jadual ujian

Aktor admin mengawali *sequence* ini dengan memanggil *method* `viewData()`, kemudian *class* `GUIJUTinf` melakukan instansiasi ke *class* `manageData` dan memanggil *method* `doViewJU()` untuk memerintahkan agar seluruh data jadual ujian yang ada ditampilkan. Lalu, *class* `manageData` melakukan intansiasi ke *class* `JUTinf` dan memanggil *method* `getJUDataSet()` untuk mengambil semua data jadual ujian. Kemudian aktor admin memanggil *method* `onUbah()`, lalu *class* `GUIJUTinf` melakukan instansiasi ke *class* `FormJUTinf` dan memanggil *method* `showForm()` untuk menampilkan form.

Setelah itu, *class* `FormJUTinf` melakukan instansiasi ke *class* `manageData` dan memanggil *method* `doUpdateJU()` untuk memproses data yang telah diubah oleh admin. Lalu, *class* `manageData` melakukan instansiasi ke *class* `JUTinf` dan memanggil *method* `updateJU()` untuk melakukan perintah *update* ke dalam basisdata tabel `jutinf`. *Sequence diagram* ubah data jadwal ujian terlihat pada gambar 3.22.

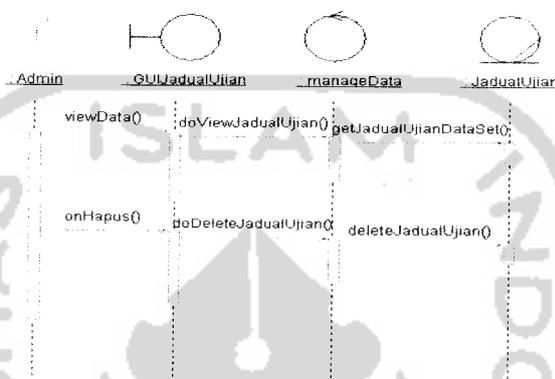


**Gambar 3.22** *Sequence Diagram* Ubah Data Jadwal Ujian

#### 3.2.2.1.3.1.18 Hapus data jadwal ujian

Aktor admin mengawali *sequence* ini dengan memanggil *method* `viewData()`, kemudian *class* `GUIJUTinf` melakukan instansiasi ke *class* `manageData` dan memanggil *method* `doViewJU()` untuk memerintahkan agar seluruh data jadwal ujian yang ada ditampilkan. Lalu, *class* `manageData` melakukan intansiasi ke *class* `JUTinf` dan memanggil *method* `getJUDDataSet()` untuk mengambil semua data jadwal ujian. Kemudian aktor admin memanggil *method* `onHapus()`. Setelah itu, *class* `GUIJUTinf` melakukan instansiasi ke *class* `manageData` dan memanggil *method* `doDeleteJU()` untuk melakukan proses

hapus data jadwal ujian. Lalu, *class* `manageData` melakukan instansiasi ke *class* `JUTinf` dan memanggil *method* `deleteJU()` untuk melakukan perintah *delete* ke dalam basisdata tabel `jutinf`. *Sequence diagram* hapus data jadwal ujian terlihat pada gambar 3.23.



**Gambar 3.23** *Sequence Diagram* Hapus Data Jadwal Ujian

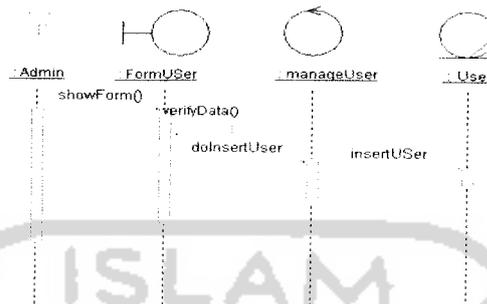
### 3.2.2.1.3.2 Sequence Diagram Kelompok Pengolahan User

*Sequence diagram* pada kelompok ini menggambarkan pengolahan user pengguna sistem atau perbedaan akses dari user mahasiswa dan admin. Berikut dijelaskan gambaran lengkap mengenai *sequence diagram* dari kelompok pengolahan user.

#### 3.2.2.1.3.2.1 Tambah Data User

Aktor admin mengawali *sequence* ini dengan memanggil *method* `showForm()`, kemudian *class* `FormUser` melakukan instansiasi ke *class* `manageUser` dan memanggil *method* `doInsertUser()` untuk melakukan proses input data yang dilakukan aktor admin. Lalu, *class* `manageUser` melakukan instansiasi ke *class* `user` dan memanggil *method* `insertUser()` untuk melakukan

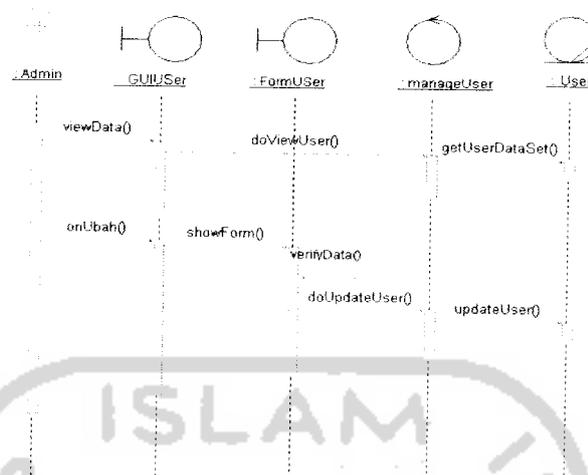
pemasukan data ke tabel User. *Sequence diagram* tambah data user terlihat pada gambar 3.24.



**Gambar 3.24** *Sequence Diagram* Tambah Data User

#### 3.2.2.1.3.2.2 Ubah Data User

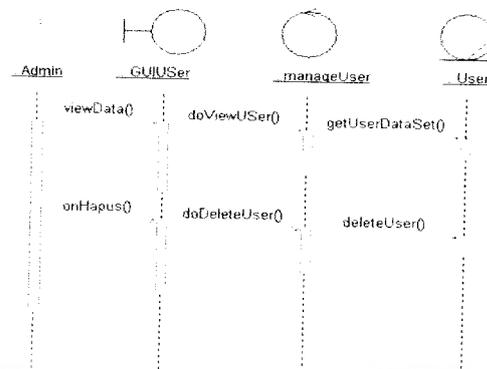
Aktor admin mengawali *sequence* ini dengan memanggil *method* `viewData()`, kemudian *class* `GUIUser` melakukan instansiasi ke *class* `manageUser` dan memanggil *method* `doViewUser()` untuk memerintahkan agar seluruh data user yang ada ditampilkan. Lalu, *class* `manageUser` melakukan instansiasi ke *class* `User` dan memanggil *method* `getUserDataSet()` untuk mengambil semua data user. Kemudian aktor admin memanggil *method* `onUbah()`, lalu *class* `GUIUser` melakukan instansiasi ke *class* `FormUser` dan memanggil *method* `showForm()` untuk menampilkan form. Setelah itu, *class* `FormUser` melakukan instansiasi ke *class* `manageUser` dan memanggil *method* `doUpdateUser()` untuk memproses data user yang telah diubah oleh admin. Lalu, *class* `manageUser` melakukan instansiasi ke *class* `User` dan memanggil *method* `updateUser()` untuk melakukan perintah *update* ke dalam basisdata tabel user. *Sequence diagram* ubah data user terlihat pada gambar 3.25.



Gambar 3.25 Sequence Diagram Ubah Data User

#### 3.2.2.1.3.2.3 Hapus Data User

Aktor admin mengawali *sequence* ini dengan memanggil *method* `viewData()`, kemudian *class* `GUIUser` melakukan instansiasi ke *class* `manageUser` dan memanggil *method* `doViewUser()` untuk memerintahkan agar seluruh data user yang ada ditampilkan. Lalu, *class* `manageUser` melakukan instansiasi ke *class* `User` dan memanggil *method* `getUserDataSet()` untuk mengambil semua data user. Kemudian aktor admin memanggil *method* `onHapus()`. Setelah itu, *class* `GUIUser` melakukan instansiasi ke *class* `manageUser` dan memanggil *method* `doDeleteUser()` untuk melakukan proses hapus data user. Lalu, *class* `manageUser` melakukan instansiasi ke *class* `User` dan memanggil *method* `deleteUser()` untuk melakukan perintah *delete* ke dalam basisdata user. *Sequence diagram* hapus data user terlihat pada gambar 3.26.

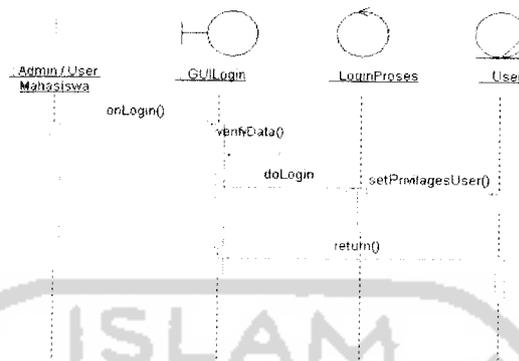


**Gambar 3.26** *Sequence Diagram* Hapus Data User

#### 3.2.2.1.3.2.4 *Login*

Aktor admin atau mahasiswa mengawali *sequence* ini dengan memanggil *method* `onLogin()`, kemudian *class* `GUILogin` melakukan instansiasi ke *class* `LoginProses` dan memanggil *method* `doLogin()` untuk memerintahkan agar memeriksa kebenaran `username` dan `password`. Lalu, *class* `LoginProses` melakukan intansiasi ke *class* `User` dan memanggil *method* `setPrivilagesUser()` untuk memeriksa kebenaran `username` dan `password` yang dimasukan user. Setelah itu, *class* `User` menanggapi dengan mengembalikan status apakah user berhasil atau gagal *login*, setelah diperiksa kebenaran `username` dan `password`nya oleh *class* `User` berdasarkan *record* basisdata tabel user.

*Sequence diagram login* terlihat pada gambar 3.27.



**Gambar 3.27** *Sequence Diagram Login User*

#### 3.2.2.1.4 Activity Diagram

Dalam pemodelan dinamis suatu sistem, terjadi langkah-langkah berurutan(*sequential*) serta konkuren, dengan *activity diagram* sistem dapat dimodelkan dengan aliran-aliran dari objek dalam pergerakannya dari suatu state ke state lainnya dalam suatu aliran kendali. *Activity diagram* merupakan diagram yang digunakan untuk memodelkan aspek-aspek dinamis dari suatu sistem.

Biasanya *activity diagram* tidak menampilkan secara detail urutan proses, namun hanya memberikan gambaran umum bagaimana urutan prosesnya. *Activity diagram* juga berguna untuk menggambarkan perilaku paralel atau menjelaskan bagaimana perilaku dalam berbagai use case berinteraksi.

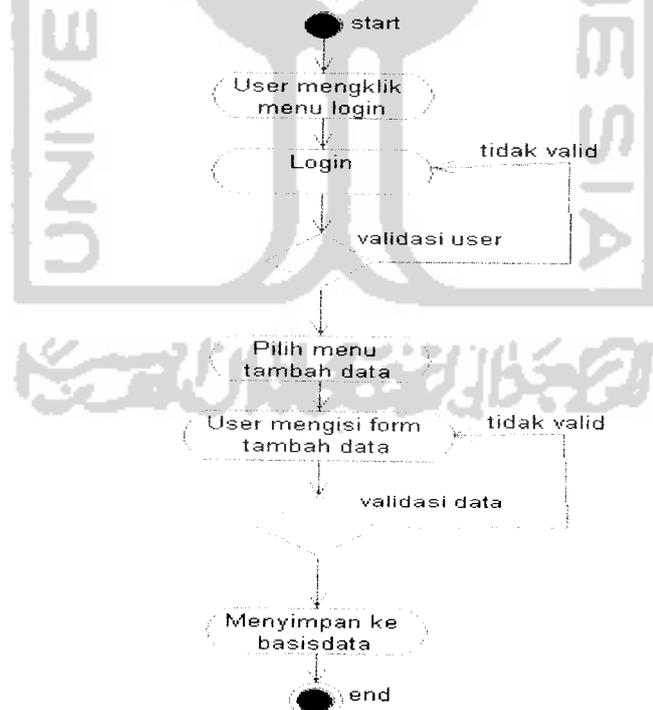
Berikut akan dijelaskan bagaimana gambaran umum atas aktifitas user admin terhadap sistem ini yang mencakup aktifitas tambah, ubah, dan hapus data.

### 3.2.2.1.4.1 Acitivity Diagram Tambah Data

Aktifitas user admin dalam penambahan data baru kedalam basisdata, diterangkan urutan aktifitasnya adalah :

- a. User admin memilih menu *login*, dan kemudian sistem akan memeriksa apakah username dan *password* yang dimasukan benar,
- b. Setelah berhasil *login*, user memilih menu tambah data,
- c. User mengisi form untuk tambah data, kemudian sistem akan memeriksa apakah data yang diisikan sudah lengkap dan benar,
- d. Sistem menyimpan data yang berhasil diisikan ke basisdata,
- e. Sistem mengakhiri proses tambah data yang dilakukan user.

*Activity diagram* untuk tambah data dijelaskan pada gambar 3.28.



**Gambar 3.28** *Activity Diagram* Tambah Data

### 3.2.2.1.4.2 Activity Diagram Ubah Data

Aktifitas user admin dalam pengubahan data yang sudah ada, diterangkan urutan aktifitasnya adalah :

- a. User admin memilih menu *login*, dan kemudian sistem akan memeriksa apakah username dan *password* yang dimasukan benar,
- b. Setelah berhasil *login*, user memilih menu ubah data,
- c. Sistem menampilkan semua data yang sudah ada,
- d. User memilih data yang akan diubah,
- e. User mengubah nilai data yang ada dari form, kemudian sistem akan memeriksa apakah data yang diubah sudah lengkap dan benar,
- f. Sistem menyimpan data yang berhasil diubah ke basisdata,
- g. Sistem mengakhiri proses ubah data yang dilakukan user.

*Activity diagram* untuk ubah data dijelaskan pada gambar 3.29.

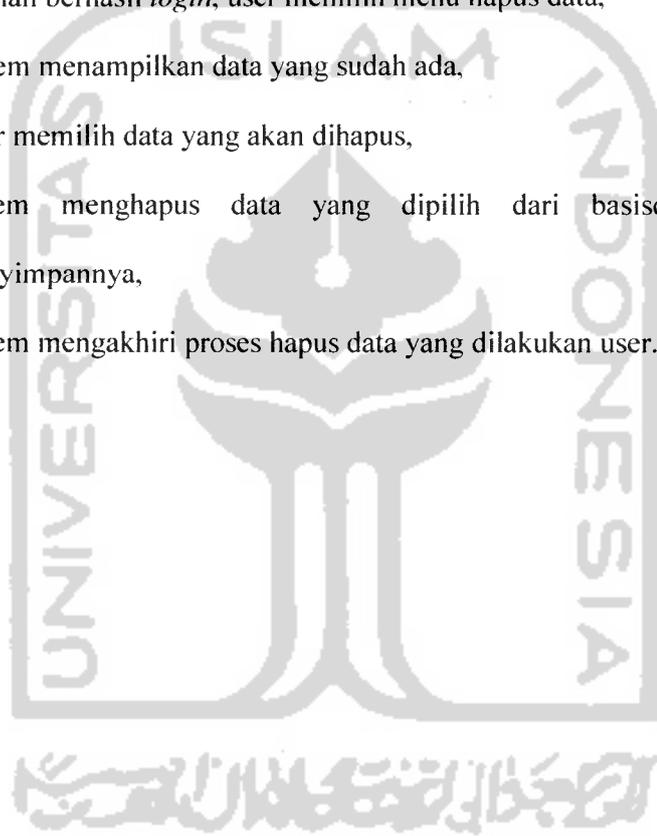


**Gambar 3.29** *Activity Diagram* Ubah Data

### 3.2.2.1.4.3 Activity Diagram Hapus Data

Aktifitas user admin dalam penghapusan data dalam basisdata, diterangkan urutan aktifitasnya adalah :

- a. User admin memilih menu *login*, dan kemudian sistem akan memeriksa apakah username dan *password* yang dimasukan benar,
- b. Setelah berhasil *login*, user memilih menu hapus data,
- c. Sistem menampilkan data yang sudah ada,
- d. User memilih data yang akan dihapus,
- e. Sistem menghapus data yang dipilih dari basisdata, kemudian menyimpannya,
- f. Sistem mengakhiri proses hapus data yang dilakukan user.



*Activity diagram* untuk hapus data dijelaskan pada gambar 3.30.



**Gambar 3.30** *Activity Diagram* Hapus Data

### 3.2.2.2 Perancangan Basis Data

Basis data termasuk salah satu komponen penting dalam sebuah sistem berbasis komputer. Pada penelitian ini digunakan basis data relational. Sistem ini terdiri dari beberapa tabel, yaitu :

#### 3.2.2.2.1 Struktur Tabel

##### 1) Tabel Mahasiswa

Tabel mahasiswa digunakan untuk menyimpan data-data induk mahasiswa.

Struktur dari tabel data mahasiswa dapat dilihat pada tabel 3.1

**Tabel 3.1** Tabel mahasiswa

Nama Field	Tipe Data	Panjang	Keterangan
<u>Nim</u>	Integer	10	No Induk mahasiswa (PK)
Nama	Varchar	50	Nama mahasiswa
Tempat lahir	Varchar	50	Tempat lahir mahasiswa
Tgl lahir	Date		Tanggal lahir mahasiswa
Kelamin	Varchar	1	Jenis kelamin mahasiswa
Alamat_YK	Varchar	150	Alamat yogyakarta mahasiswa
Alamat asal	Varchar	150	Alamat asal mahasiswa
Email	Varchar	50	Email mahasiswa
Telepon	Varchar	50	Nomor telepon mahasiswa
Nama ayah	Varchar	100	Nama ayah mahasiswa
Pekerjaan ayah	Varchar	100	Pekerjaan ayah mahasiswa
Nama ibu	Varchar	100	Nama ibu mahasiswa
Pekerjaan ibu	Varchar	100	Pekerjaan ibu mahasiswa
Alamat ortu	Varchar	100	Alamat orangtua mahasiswa

##### 2) Tabel Mahasiswa\_otentikasi

Tabel mahasiswa\_otentikasi digunakan untuk menyimpan data autentikasi mahasiswa dan status keanggotaan mahasiswa. Struktur dari tabel mahasiswa\_otentikasi dapat dilihat pada tabel 3.2.

**Tabel 3.2** Tabel mahasiswa\_ autentikasi

Nama Field	Type Data	Panjang	Keterangan
Nim	Varchar	10	Nim mahasiswa (FK)
Password	Varchar	100	Password mahasiswa
Status	Integer	1	Status keanggotaan mahasiswa

**3) Tabel koneksi\_data**

Tabel koneksi\_data digunakan untuk menampung data-data settingan koneksi pada sistem, seperti : *port* yang digunakan, data BPS (*bit per second*), *data\_bits*, *parity*, *stop\_bits*, dan *flow\_control*. Struktur dari tabel koneksi\_data dapat dilihat pada tabel 3.3.

**Tabel 3.3** Tabel koneksi\_data

Nama Field	Type Data	Panjang	Keterangan
<u>Nama</u>	Varchar	50	Nama koneksi (PK)
Port	Varchar	30	Port yang digunakan
Bps	Integer	10	Bits per second
Data_bits	Integer	3	Data_bits yang digunakan
Parity	Varchar	30	Parity yang dipakai
Stop_bits	Float		Stop_bits
Flow_control	Varchar	30	Flow_control yang digunakan

**4) Tabel Koneksi\_log**

Tabel koneksi\_log digunakan untuk menyimpan data-data log koneksi yang pernah dipakai pada sistem. Struktur dari tabel koneksi\_log dapat dilihat pada tabel 3.4.

**Tabel 3.4** Tabel koneksi\_log

Nama Field	Type Data	Panjang	Keterangan
<u>Id_log</u>	Integer	10	No id log koneksi (PK)
Nama	Varchar	50	Nama koneksi (FK)
Waktu	Time	30	Waktu melakukan koneksi
Status	Integer	1	Status koneksi

### 5) Tabel Sms\_inbox

Tabel sms\_inbox digunakan untuk menampung data-data sms yang masuk pada sistem. Struktur dari tabel sms\_inbox dapat dilihat pada tabel 3.5.

**Tabel 3.5** Tabel sms\_inbox

Nama Field	Tipe Data	Panjang	Keterangan
<u>Id_inbox</u>	Integer	50	Id tabel inbox (PK)
No_telp	Varchar	30	No telepon pengirim pesan
Pesan	Varchar	160	Isi pesan
Waktu	Datetime		Waktu terima pesan

### 6) Tabel Sms\_outbox

Tabel sms\_outbox digunakan untuk menampung data-data sms keluar yang dikirimkan dari ke user pengguna. Struktur dari tabel sms\_outbx dapat dilihat pada tabel 3.6.

**Tabel 3.6** Tabel sms\_outbox

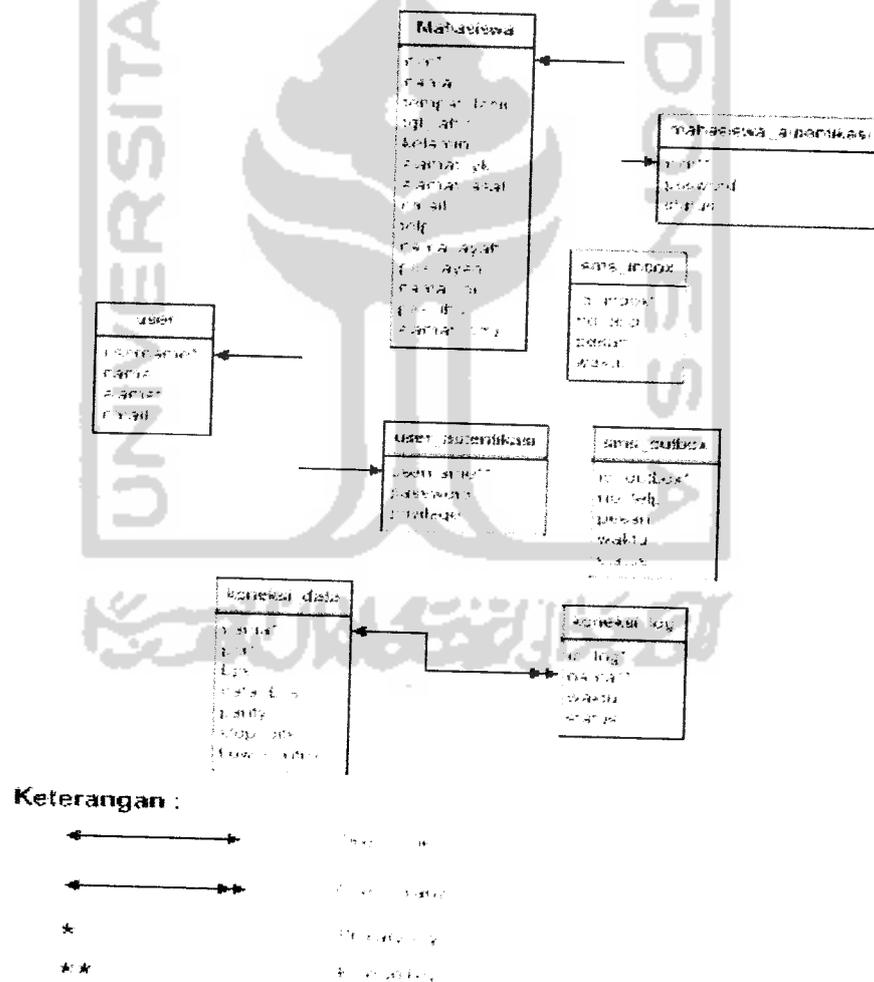
Nama Field	Tipe Data	Panjang	Keterangan
<u>Id_outbox</u>	Integer	50	Id tabel outbox (PK)
No_telp	Varchar	30	No telepon yang dituju
Pesan	Varchar	160	Isi pesan
Waktu	Datetime		Waktu pengiriman pesan
Status	Integer	1	Status pengiriman

### 7) Tabel User

Tabel user digunakan untuk menampung data-data user dalam sistem, yaitu : *Super admin, admin*, dan pegawai. Struktur dari tabel user ini dapat dilihat pada tabel 3.7.

### 3.2.2.2 Relasi Antar Tabel

Basis data merupakan kumpulan data yang berhubungan satu sama lain yang menghasilkan suatu informasi. Oleh karena itu, dari tabel-tabel diatas diperlukan adanya relasi menggunakan Diagram Relasi Entitas atau sering disebut ERD (*Entity Relationship Diagram*), agar informasi yang dihasilkan oleh basis data lebih akurat dan tidak terjadi kerangkapan data (*redudancy*). ERD untuk tabel-tabel diatas dapat dilihat pada gambar 3.31.



Gambar 3.31 Relasi Antar Tabel

### 3.2.2.3 Perancangan Antarmuka

Perancangan antarmuka untuk user mahasiswa berbasiskan format teks sms. Sedangkan, perancangan antarmuka *server* untuk sistem pengaksesan data pribadi mahasiswa berbasis SMS ini menggunakan NetBeans 5.5 sebagai teks editornya.

#### 3.2.2.3.1 Perancangan Halaman utama

Perancangan halaman utama untuk user mahasiswa berbasis format teks sms. Sedangkan, tampilan utama *server* sistem pengaksesan data pribadi mahasiswa berbasis SMS ini dibagi menjadi beberapa bagian, diantaranya : bagian banner, bagian *button* menu samping kiri, bagian panel informasi tengah, dan bagian footer yang berisi informasi tambahan. Rancangan *interface* untuk sistem pengaksesan data pribadi mahasiswa berbasis SMS ini dapat dilihat pada gambar 3.32.



**Gambar 3.32** Tampilan utama (*interface*) sistem

### 3.2.2.3.2 Perancangan *Input*

Masukan data (*Input*) terhadap sistem merupakan suatu langkah awal pemrosesan data dalam suatu sistem informasi, data tersebut berupa data-data mentah yang diperlukan sistem untuk diproses. Perancangan masukan untuk sistem ini dibagi menjadi 2 bagian, yaitu : perancangan format sms yang akan dipakai user mahasiswa dan perancangan masukan data pada *sms server* untuk mengambil data yang diisikan oleh seorang *user* pada sistem *server*.

#### 3.2.2.3.2.1 Perancangan format sms

Perancangan format sms merupakan tahapan ditentukannya format sms yang dikirimkan user terhadap sistem. Perancangan ini ditentukan agar ke-*valid* an data yang dikirimkan user dapat disimpan ke basisdata. Perancangan format sms untuk sistem pengaksesan data pribadi mahasiswa berbasis sms ini dibagi menjadi perancangan format sms panduan dan perancangan format sms blok kode perintah.

##### 1. Perancangan format sms panduan

Format sms panduan digunakan user mahasiswa untuk membantu mahasiswa apabila tidak mengetahui fasilitas-fasilitas yang ada pada sistem ini. Adapun perancangan format sms untuk panduan, sebagai berikut :

- a) *HELP* : format sms untuk mengetahui semua perintah yang tersedia.
- b) *REG?* : format sms untuk mengetahui panduan melakukan pendaftaran.
- c) *UP?* : format sms untuk mengetahui panduan merubah *password* lama dengan *password* baru, untuk user mahasiswa berbasis sms.

- d) IPK? : format sms untuk mengetahui panduan melihat ipk mahasiswa yang bersangkutan apabila sudah terdaftar.
- e) JK? : format sms untuk mengetahui panduan melihat jadwal kuliah yang ada.
- f) JU? : format sms untuk mengetahui panduan melihat jadwal ujian yang ada.
- g) ALAMATYK? : format sms untuk mengetahui panduan melihat alamat yogyakarta mahasiswa yang bersangkutan.
- h) ALAMATASAL? : format sms untuk mengetahui panduan melihat alamat asal mahasiswa yang bersangkutan.
- i) TELP? : format sms untuk mengetahui panduan melihat nomor telepon terbaru mahasiswa yang bersangkutan.
- j) UAYK? : format sms untuk mengetahui panduan mengubah alamat yogyakarta mahasiswa yang bersangkutan dengan alamat yogyakarta terbaru.
- k) UASAL? : format sms untuk mengetahui panduan mengubah alamat asal mahasiswa yang bersangkutan dengan alamat asal terbaru.
- l) UTELP? : format sms untuk mengetahui panduan mengubah nomor telepon mahasiswa yang bersangkutan dengan nomor telepon terbaru.



## 2. Perancangan format sms blok kode perintah

Format sms blok kode perintah digunakan user untuk berinteraksi dengan sistem melalui sms, dengan mengetahui blok-blok perintah yang ada dengan dipisahkan tanda (#). Adapun perancangan format sms untuk blok-blok kode perintah, sebagai berikut :

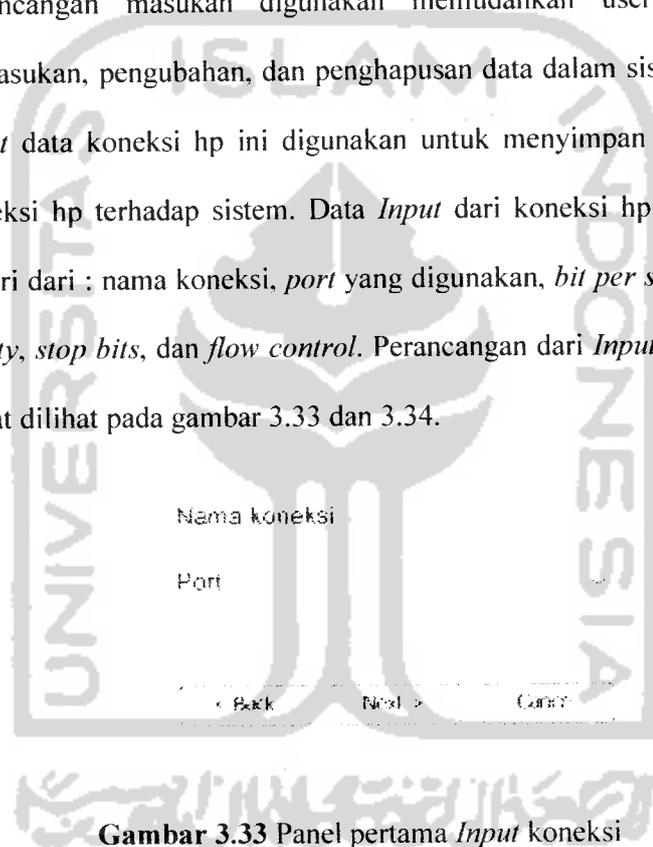
- a) Format sms untuk pendaftaran : REG#<NIM>
- b) Format sms untuk melihat alamat yogyakarta mahasiswa :  
ALAMATYK#<NIM>#<PASSWORD>
- c) Format sms untuk melihat alamat asal mahasiswa :  
ALAMATASAL#<NIM>#<PASSWORD>
- d) Format sms untuk melihat nomor telepon mahasiswa :  
TELP#<NIM>#<PASSWORD>
- e) Format sms untuk melihat ipk terakhir mahasiswa :  
IPK#<NIM>#<PASSWORD>
- f) Format sms untuk melihat jadwal kuliah yang ada : JK#<NAMA  
MATKUL>.
- g) Format sms untuk melihat jadwal ujian yang ada : JU#<NAMA MATKUL>
- h) Format sms untuk mengubah *password* : UP#<NIM>#<PASSWORD  
LAMA>#<PASSWORD BARU>
- i) Format sms untuk mengubah alamat yogyakarta mahasiswa :  
UAYK#<NIM>#<PASSWORD>#<ALAMAT YK TERBARU>
- j) Format sms untuk mengubah alamat asal mahasiswa :  
UASAL#<NIM>#<PASSWORD>#<ALAMAT ASAL TERBARU>
- k) Format sms untuk mengubah nomor telepon mahasiswa :  
UTELP#<NIM>#<PASSWORD>#<TELP TERBARU>.

### 3.2.2.3.2.2 Perancangan masukan data *sms server*

Perancangan masukan data *sms server* merupakan tahapan membuat rancangan *form Inputan* data yang akan digunakan untuk menangkap data yang dibutuhkan sistem untuk diproses pada *sms server*.

#### 1. Perancangan masukan data koneksi hp

Perancangan masukan digunakan memudahkan user dalam proses pemasukan, perubahan, dan penghapusan data dalam sistem. Antarmuka *Input* data koneksi hp ini digunakan untuk menyimpan data pengaturan koneksi hp terhadap sistem. Data *Input* dari koneksi hp terhadap sistem terdiri dari : nama koneksi, *port* yang digunakan, *bit per second*, *data bits*, *parity*, *stop bits*, dan *flow control*. Perancangan dari *Input* data koneksi hp dapat dilihat pada gambar 3.33 dan 3.34.



Gambar 3.33 Panel pertama *Input* koneksi



**Gambar 3.34** Panel kedua input koneksi

## **2. Perancangan masukan data mahasiswa**

Antarmuka input data mahasiswa ini digunakan untuk menyimpan data pribadi mahasiswa. Antarmuka input data mahasiswa ini terdiri dari beberapa data inputan, diantaranya : nim, nama, tempat dan tanggal lahir, jenis kelamin, alamat YK, alamat asal, email, telepon, nama ayah, pekerjaan ayah, nama ibu, pekerjaan ibu, dan alamat orang tua. Perancangan antarmuka untuk input data mahasiswa dapat dilihat pada gambar 3.35.

Data Mahasiswa    Data Keluarga

NIM

Nama

Tempat Lahir

Tanggal Lahir

Jenis Kelamin     Laki-laki  
 Perempuan

Alamat Asal

Alamat YK

Email

Telepon

**Gambar 3.35** Input data mahasiswa

### 3. Perancangan masukan data user

Antarmuka input data user ini digunakan untuk menyimpan data user. Input data user ini terdiri dari beberapa inputan data user yang digunakan untuk membedakan hak akses user, diantaranya : username, password, re-type password, nama user, hak akses, alamat, dan email. Perancangan antarmuka untuk input data user dapat dilihat pada gambar 3.36.

Username

Password

Re-type Password

---

Name

Hak akses

Alamat

Email

Done    Reset to default    Cancel

**Gambar 3.36** Input data user

#### 4. Perancangan masukan data indeks prestasi kumulatif

Antarmuka input data ipk ini digunakan untuk menyimpan data ipk mahasiswa. Input data ipk ini terdiri dari beberapa inputan, diantaranya : nim, nama, ip semester, dan ipk mahasiswa yang bersangkutan. Perancangan antarmuka untuk input data ipk dapat dilihat pada gambar 3.37.

NIM

IP Semester

IPK

Done    Reset to default    Cancel

**Gambar 3.37** Input data ipk

### 5. Perancangan masukan data jadwal kuliah

Antarmuka input data jadwal kuliah ini digunakan untuk menyimpan data jadwal kuliah. Input data jadwal kuliah ini terdiri dari beberapa inputan, diantaranya : nama mata kuliah, hari kuliah, jam kuliah, ruang kuliah, dan dosen pengampu mata kuliah tersebut. Perancangan antarmuka untuk input data jadwal kuliah dapat dilihat pada gambar 3.38

The image shows a web form for entering lecture schedule data. The form is titled 'ISLAM' and is part of the 'UNIVERSITAS INDONESIA' system. It contains five input fields: 'Nama mata kuliah', 'Hari kuliah', 'Jam kuliah', 'Ruang kuliah', and 'Dosen pengampu'. Below the fields are three buttons: 'Done', 'Reset to default', and 'Cancel'. The form is overlaid on a watermark of the Universitas Islam Indonesia logo.

Gambar 3.38 Input data jadwal kuliah

### 6. Perancangan masukan data jadwal ujian

Antarmuka input data jadwal ujian ini digunakan untuk menyimpan data jadwal ujian mahasiswa. Input data jadwal ujian ini terdiri dari beberapa inputan, diantaranya : nama mata kuliah, hari ujian, jam ujian, ruang ujian, dan dosen pengampu mata kuliah tersebut. Perancangan antarmuka untuk input data jadwal ujian dapat dilihat pada gambar 3.39.

Nama wataku jah	<
Hari ujian	<
Jam ujian	<
Ruang ujian	<

**Gambar 3.39** Input data jadual ujian

### 7. Perancangan ubah data sms masuk (*inbox*)

Antarmuka ubah data sms masuk ini digunakan untuk mengubah data sms masuk (*inbox*) yang sudah tersimpan. Ada beberapa data inputan pada antarmuka ini, diantaranya : pesan, nomor pengirim, dan waktu pengiriman. Perancangan antarmuka untuk ubah data sms masuk dapat dilihat pada gambar 3.40.

Pesan	<	>
Dari	<	>
Waktu	<	>

**Gambar 3.40** Ubah data sms masuk (*inbox*)

### 8. Perancangan ubah data sms keluar (*outbox*)

Antarmuka ubah data sms keluar ini digunakan untuk mengubah data sms keluar (*outbox*) yang sudah tersimpan. Ada beberapa data inputan pada

antarmuka ini, diantaranya : pesan, nomor pengirim, waktu pengiriman, dan status. Perancangan antarmuka untuk ubah data sms keluar dapat dilihat pada gambar 3.41.

Pesan

Nomor

Waktu

Status

Done Reset to default Cancel

**Gambar 3.41** Ubah data sms keluar (*outbox*)

#### 9. Perancangan ubah password user

Antarmuka ubah password digunakan oleh user baik administrator ataupun pegawai untuk mengubah passwordnya sendiri. Ada beberapa data inputan untuk antarmuka ubah password ini, diantaranya : password lama, password baru, dan re-type password. Perancangan antarmuka untuk ubah password user dapat dilihat pada gambar 3.42.

Password lama

Password baru

Re-type password

Done Reset to default Cancel

**Gambar 3.42** Ubah password user

### 3.3 Implementasi Perangkat Lunak

Tahap implementasi merupakan tahap penerapan sistem di kehidupan yang sebenarnya agar dapat berfungsi sesuai kebutuhan, sehingga dapat diketahui apakah sistem yang dibuat sesuai dengan perancangan yang dibuat sebelumnya. Pada subbab implementasi perangkat lunak ini akan dijelaskan bagaimana sistem ini bekerja, dengan memberikan contoh dari format sms yang akan dikirimkan oleh *user* mahasiswa dan tampilan form-form yang terdapat pada sms server sistem ini.

#### 3.3.1 Batasan Implementasi

Sistem pengaksesan data pribadi mahasiswa berbasis sms ini dalam implementasinya, dibatasi pada proses pemasukan, pengubahan, dan penghapusan data-data mahasiswa, user, jadwal kuliah, jadwal ujian, ipk, dan sms. Sedangkan, dari sisi user (mahasiswa) sistem ini hanya dibatasi pada format sms untuk pengaksesan data pribadi yang bisa diubah saja.

#### 3.3.2 Implementasi format sms

Implementasi format sms untuk user mahasiswa digunakan untuk berinteraksi dengan sistem yang berfungsi sesuai fungsi formatnya masing-masing.

1. Untuk mengetahui semua kode perintah yang tersedia pada sistem, mahasiswa mengetik sms : `HELP`
2. Untuk mengetahui panduan pendaftaran, mahasiswa mengetik sms : `REG?`
3. Untuk mengetahui panduan melihat ipk, mahasiswa mengetik sms : `IPK?`

4. Untuk mengetahui panduan melihat jadwal kuliah, mahasiswa mengetik sms : JK?
5. Untuk mengetahui panduan melihat jadwal ujian, mahasiswa mengetik sms : JU?
6. Untuk mengetahui panduan melihat alamat tinggal yogyakarta, mahasiswa mengetik sms : ALAMATYK?
7. Untuk mengetahui panduan melihat alamat tinggal asal, mahasiswa mengetik sms : ALAMATASAL?
8. Untuk mengetahui panduan melihat nomor telepon, mahasiswa mengetik sms : TELP?
9. Untuk mengetahui panduan mengubah data alamat tinggal yogyakarta, mahasiswa mengetik : UAYK?
10. Untuk mengetahui panduan mengubah data alamat tinggal asal, mahasiswa mengetik : UASAL?
11. Untuk mengetahui panduan mengubah data nomor telepon, mahasiswa mengetik : UTELP?
12. Untuk mengetahui panduan mengubah *password*, mahasiswa mengetik sms : UP?
13. Untuk mendaftar pertama kali sebagai pengguna fasilitas *sms server*, mahasiswa mengetik sms : REG#03523001
14. Untuk melihat ipk, mahasiswa mengetik sms : IPK#03523001#1234
15. Untuk melihat jadwal kuliah, mahasiswa mengetik sms : JK#ALPRO
16. Untuk melihat jadwal ujian, mahasiswa mengetik sms : JU#ALPRO

17. Untuk melihat nomor telepon, mahasiswa mengetik sms :

TELP#03523001#1234

18. Untuk melihat alamat tinggal yogyakarta, mahasiswa mengetik sms :

ALAMATYK#03523001#1234

19. Untuk melihat alamat tinggal asal, mahasiswa mengetik sms :

ALAMATASAL#03523001#1234

20. Untuk mengubah alamat tinggal yogyakarta, mahasiswa mengetik sms :

UAYK#03523001#1234#JAKAL KM 14

21. Untuk mengubah alamat tinggal asal, mahasiswa mengetik sms :

UASAL#03523001#1234#KUNINGAN

22. Untuk mengubah nomor telepon, mahasiswa mengetik sms :

UTEHP#03523001#1234#08563256859

23. Untuk mengubah *password* user, mahasiswa mengetik sms :

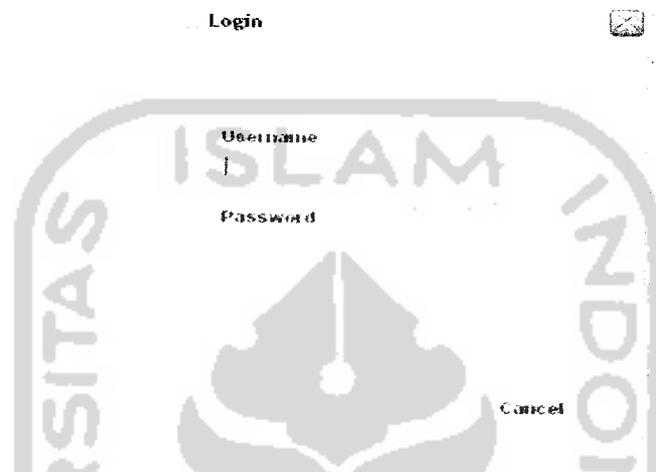
UP#1234#TESTTEST

### 3.3.3 Implementasi antarmuka

Implementasi antarmuka untuk *sms server* sistem pengaksesan data pribadi mahasiswa berbasis sms ini mempunyai beberapa tampilan dan form-form yang berfungsi sesuai fungsinya masing-masing. Tampilan dan form-form tersebut akan tampil sesuai dengan urutan tertentu, setelah pengguna melakukan salah satu proses, selanjutnya sistem akan bekerja sesuai yang ditentukan user.

### 3.3.3.1 Halaman login

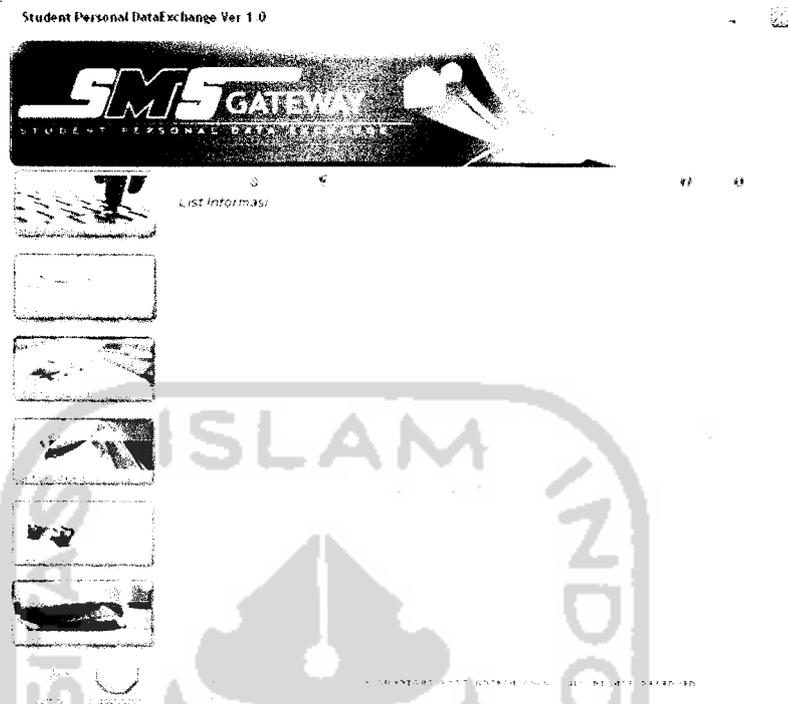
Halaman login digunakan user untuk masuk ke halamannya baik sebagai admin atau pegawai sesuai dengan fungsinya. Tampilan dari halaman login dapat dilihat pada gambar 3.43.



Gambar 3.43 Tampilan halaman login user

### 3.3.3.2 Halaman utama user

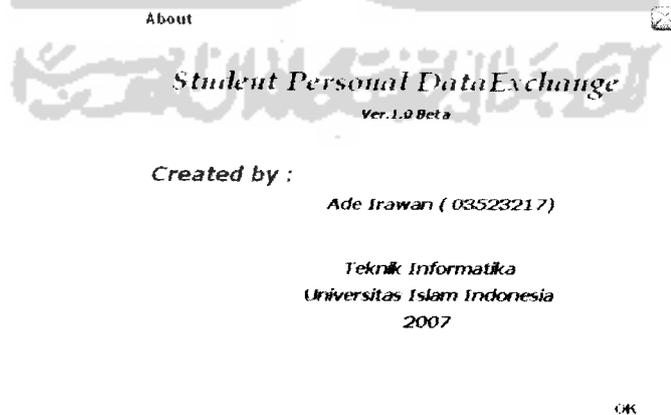
Halaman utama user merupakan tampilan pertama setelah berhasil login sebagai super admin, admin atau pegawai. Apabila login sebagai super admin, semua menu pengolahan akan bisa diakses, seperti : menu mahasiswa, koneksi, ipk, jadwal kuliah, jadwal ujian, user, dan olah sms. Apabila login sebagai admin, user ini tidak bisa melakukan pengolahan data user sedangkan menu lain bisa digunakan. Dan, apabila login sebagai pegawai menu koneksi, olah user, dan olah sms tidak akan bisa diakses. Tampilan dari halaman utama *user* dapat dilihat pada gambar 3.44.



**Gambar 3.44** Tampilan halaman utama *user*

### 3.3.3.3 Halaman about

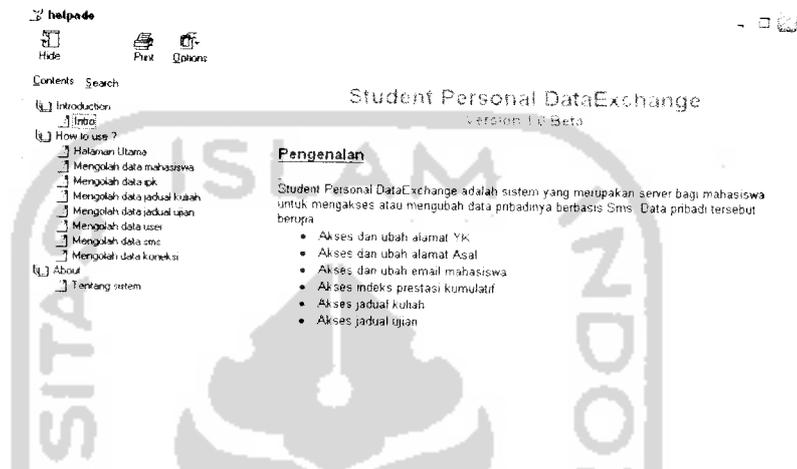
Halaman about ini berisi tentang penjelasan singkat sistem yang dibuat serta informasi tambahan dari sistem. Tampilan dari halaman about dapat dilihat pada gambar 3.45.



**Gambar 3.45** Tampilan halaman about

### 3.3.3.4 Halaman help

Halaman help merupakan halaman petunjuk atau cara-cara penggunaan sistem dengan tujuan agar memudahkan user dalam menggunakan sistem ini. Tampilan dari halaman help ini dapat dilihat pada gambar 3.46.



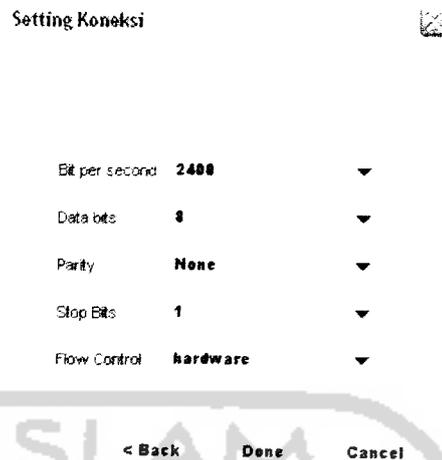
Gambar 3.46 Tampilan halaman help

### 3.3.3.5 Input data koneksi

Halaman ini digunakan untuk menambah setting data koneksi baru yang diinputkan admin. Halaman ini terdiri dari beberapa inputan, diantaranya : nama koneksi, port yang digunakan, *bit per second*, *data bits*, *parity*, *stop bits*, dan *flow control*. Tampilan dari halaman input data koneksi ini dapat dilihat pada gambar 3.47 untuk panel pertama dan gambar 3.48.



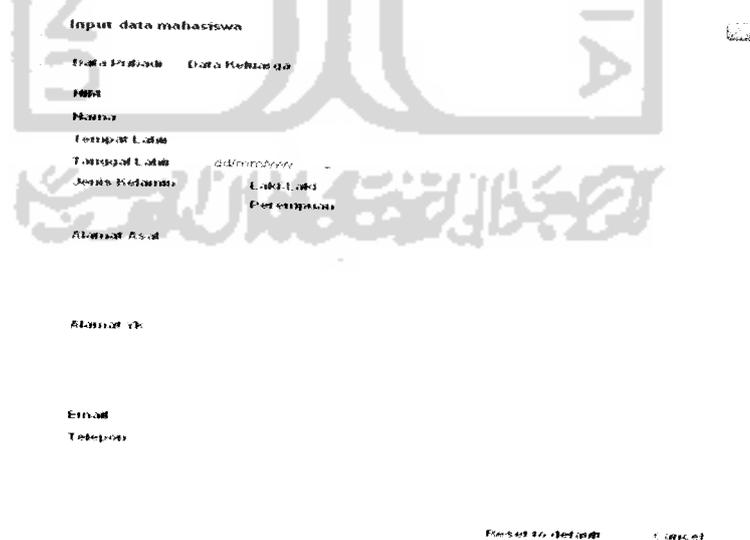
Gambar 3.47 Tampilan halaman input data koneksi panel pertama



**Gambar 3.48** Tampilan halaman input data koneksi panel kedua

### 3.3.3.6 Input data mahasiswa

Halaman ini digunakan untuk menambah data mahasiswa baru. Halaman ini terdiri dari beberapa inputan, diantaranya : nim, nama, tempat dan tanggal lahir, jenis kelamin, alamat YK, alamat asal, email, telepon, nama ayah, pekerjaan ayah, nama ibu, pekerjaan ibu, dan alamat orang tua. Tampilan dari halaman input data mahasiswa ini dapat dilihat pada gambar 3.49.



**Gambar 3.49** Tampilan halaman input data mahasiswa

### 3.3.3.7 Input data ipk

Halaman ini digunakan untuk menambah data ipk mahasiswa terbaru. Halaman ini terdiri dari beberapa inputan, diantaranya : nim, nama, ip semester, dan ipk. Tampilan dari halaman input data ipk dapat dilihat pada gambar 3.50.

Gambar 3.50 Tampilan halaman input data ipk

### 3.3.3.8 Input data jadwal kuliah

Halaman ini digunakan untuk menambah data jadwal kuliah mahasiswa. Halaman ini terdiri dari beberapa inputan, diantaranya : nama mata kuliah, hari kuliah, jam kuliah, ruang kuliah, dan dosen pengampu mata kuliah. Tampilan dari halaman input data jadwal kuliah dapat dilihat pada gambar 3.51.

Gambar 3.51 Tampilan halaman input data jadwal kuliah

### 3.3.3.9 Input data jadual ujian

Halaman ini digunakan untuk menambah data jadual ujian mahasiswa. Halamannya ini terdiri dari beberapa inputan, diantaranya : nama mata kuliah, hari ujian, jam ujian, ruang ujian, dan dosen pengampu mata kuliah. Tampilan dari halaman input data jadual ujian dapat dilihat pada gambar 3.52.

Data Jadual Ujian

Input Jadual Ujian

Nama Mata Kuliah [ pilih mata kuliah ]

Hari Ujian [ Pilih ]

Jam Ujian [ Pilih ]

Ruang Ujian [ Pilih ]

Done Reset to Default Cancel

Gambar 3.52 Tampilan halaman input data jadual ujian

### 3.3.3.10 Olah data sms

Halaman ini digunakan oleh admin untuk mengolah data sms yang masuk dan keluar, baik ubah atau hapus sms tersebut. Tampilan dari halaman olah data sms dapat dilihat pada gambar 3.53.

SMS INBOX	SMS OUTBOX	No	No telep	Pesan	Waktu
		1	629562979925	naaa.ccc	2007-08-30 16:49:36

Gambar 3.53 Tampilan halaman olah data sms

### 3.3.3.11 Halaman ubah password user

Halaman ini digunakan user untuk mengubah password untuk login ke sistem. Tampilan dari halaman ubah password dapat dilihat pada gambar 3.54.

Data User

User

Password Lama

Password Baru

Re-type Password

User Data

Nama adeirawan

Hak Akses

Alamat jawa km14

Email ade@yahoo.com

Done Reset Cancel

Data SMS

Gambar 3.54 Tampilan halaman ubah password user

### 3.3.3.12 Halaman ubah data sms

Halaman ini digunakan oleh admin untuk mengubah data sms masuk ataupun keluar. Tampilan dari halaman ubah data sms dapat dilihat pada gambar 3.55.

Data SMS

Pesan naasdfsdf

Dari 629562973926

Waktu 2007-08-30 16:40:36

Done Reset to default Cancel

Gambar 3.55 Tampilan halaman ubah data sms

### 3.3.4 Implementasi prosedural

Implementasi prosedural ini merupakan penerapan rancangan yang telah dibuat menjadi kedalam bentuk program (*sourcecode*). Pada sistem pengaksesan data pribadi mahasiswa berbasis sms ini, menitik beratkan pada proses penentuan *valid* tidaknya format per blok sms yang diterima sistem dan proses pengolahan sms yang diterima apabila format sms yan dikirimkan benar. Pada implementasi prosedural ini penyusun hanya akan menjelaskan tentang *method* apa yang digunakan untuk kedua proses diatas dalam sistem pengaksesan data pribadi mahasiswa berbasis sms.

a) Method *formatValidation()*

Method ini terletak pada *class sms* yang digunakan untuk memeriksa apakah format sms yang dikirimkan user sudah benar sesuai dengan aturan blok sms yang sudah ditentukan, untuk bisa diproses ke tahap selanjutnya menggunakan method *messageProcessing()*. Source code untuk method *formatValidation()* dapat dilihat pada lampiran B.

b) Method *messageProcessing()*

Method ini juga terletak pada *class sms* yang digunakan untuk memproses sms yang sudah terseleksi menggunakan method *formatValidation()*. Apabila format sudah sesuai, tetapi isian sms yang dilakukan user tidak *valid* maka akan dikirimkan sms pesan kesalahan kepada user. Sedangkan, jika format sudah sesuai dan isian sms sudah *valid*, maka sms akan diproses dan disimpan dalam tabel *inbox* sistem, dan kemudian diberi

balasan sesuai apa yang user mahasiswa butuhkan. Source code untuk method *messageProcessing()* dapat dilihat pada lampiran B.



## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1 Pengujian Program

Pengujian program merupakan tahap analisis terhadap sistem pengaksesan data pribadi mahasiswa berbasis sms. Pengujian ini dilakukan secara benar dan kompleks agar kekurangan-kekurangan yang ada dapat diketahui dan kemudian dapat diperbaiki agar kekurangan tersebut dapat dihilangkan. Pengujian terhadap sistem ini bertujuan agar mendapatkan hasil sebuah sistem yang akurat dan berdaya guna.

Pengujian program ini dapat dilakukan dengan mengirimkan sms kepada *sms server* agar dapat diketahui balasan apa yang dikirimkan sistem, sedangkan dari sisi *server* pengujian dilakukan dengan cara mengisi form-form yang telah dijelaskan sebelumnya, yaitu dengan mengisi data baik dengan data yang valid ataupun tidak valid, agar kesalahan pada sistem dapat diketahui.

#### 4.2 Analisis Kinerja Sistem

##### 4.2.1 Penanganan kesalahan

Sistem pengaksesan data pribadi mahasiswa berbasis sms ini cukup bersifat komunikatif terhadap user. Karena jika user mahasiswa mengirimkan sms yang tidak sesuai format atau data yang dikirimkan tidak valid, maka sistem akan membalas pesan kesalahan sesuai dengan format sms yang ada. Sedangkan dari sisi admin

apabila melakukan kesalahan input data, maka sistem akan memberi pesan kesalahan berupa dialog box.

#### 4.2.1.1 Penanganan kesalahan user mahasiswa

Penanganan sistem terhadap kesalahan yang dilakukan oleh user mahasiswa yaitu berupa valid atau tidaknya sms yang dikirimkan user. Sms tersebut digunakan untuk melakukan interaksi user mahasiswa dengan sistem sesuai kebutuhan. Adapun beberapa tipe kesalahan tersebut, diantaranya :

##### a. Format tidak dikenal

Penanganan kesalahan ini terjadi jika user mengirimkan format sms yang tidak dikenal atau sembarang. Sms tersebut adalah :

User kirim sms : "asdfasdf"

Sistem kirim sms : "Perintah tidak dikenal. Ketik HELP untuk mengetahui semua kode perintah."

##### b. User sudah terdaftar

Penanganan kesalahan ini terjadi apabila user melakukan pendaftaran lebih dari satu kali, atau status user sudah aktif. Sms tersebut adalah :

User kirim sms : "REG#03523001"

Sistem kirim sms : "Anda telah terdaftar. Gunakan fasilitas UPDATE data via HP."

##### c. Nim tidak dikenal

Penanganan kesalahan ini terjadi apabila user memasukkan nim yang salah dalam melakukan pendaftaran. Sms tersebut adalah :

User kirim sms : "REG#12343445"

Sistem kirim sms : "NIM tdk dikenal, ulangi dengan mengisikan NIM yg benar."

**d. Matakuliah tidak dikenal**

Penanganan kesalahan ini terjadi apabila user memasukan matakuliah yang tidak terdapat dalam sistem ini, diasumsikan mahasiswa ingin melihat jadwal kuliah atau jadwal ujian. Sms tersebut adalah :

User kirim sms : "JU#matematika"

Sistem kirim sms : "Nama mata kuliah tidak dikenal."

**e. Password salah**

Penanganan kesalahan ini terjadi apabila user memasukan password yang salah dalam melakukan interaksi dengan sistem apabila statusnya sudah aktif atau mahasiswa yang bersangkutan belum terdaftar. Sms tersebut adalah :

User kirim sms : "ALAMATYK#03523001#asdfasdf"

Sistem kirim sms : "Password salah (case sensitive) atau belum terdaftar. Untuk daftar ketik REG#<NIM>."

**f. Password baru salah**

Penanganan kesalahan ini terjadi apabila user yang sudah aktif melakukan ubah password dengan memasukan password baru tidak sesuai dengan aturan yaitu minimal 4 digit dan maksimal 10 digit. Sms tersebut adalah :

User kirim sms : "UP#03523001#1234#ade"

Sistem kirim sms : "Password baru min 4 digit dan maks 10 digit."



#### g. Ubah alamat salah

Penanganan kesalahan ini terjadi apabila user yang sudah aktif melakukan ubah alamat dengan memasukan alamat yang baru melebihi batas 50 karakter.

Sms tersebut adalah :

User kirim sms : " uayk#03523022#1234#jl kaliurang km 14,5  
lodadi rt02/05 no 34 samping kampus,  
sleman DIY, 55584"

Sistem kirim sms : "Alamat YK baru maksimal 50 karakter."

#### h. Ubah telepon salah

Penanganan kesalahan ini terjadi apabila user yang sudah aktif melakukan ubah telepon dengan memasukan telepon yang baru tidak sesuai format. Sms

tersebut adalah :

User kirim sms : "UTELP#03523001#1234#asdfsadf"

Sistem kirim sms : "Telepon harus diisi dengan angka & tdk lebih  
dari 20 digit."

#### 4.2.1.2 Penanganan kesalahan user admin

User admin bertugas untuk memasukkan, mengubah, dan menghapus data yang dibutuhkan sistem. Penanganan kesalahan untuk ketiga proses diatas, diantaranya :

**a) Penanganan kesalahan login**

Penanganan kesalahan ini terjadi apabila admin memasukkan username dan password yang salah pada proses login. *Messagebox* yang keluar setelah melakukan kesalahan login terdapat pada gambar 4.1.



**Gambar 4.1** *Dialog box* kesalahan login

**b) Penanganan kesalahan pemilihan data**

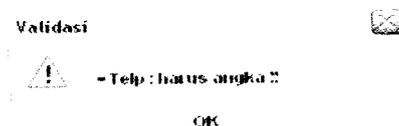
Penanganan kesalahan ini digunakan apabila admin melakukan proses ubah atau hapus data tanpa memilih terlebih dahulu data mana yang akan dihapus atau diubah. *Messagebox* yang keluar setelah melakukan kesalahan pemilihan data terdapat pada gambar 4.2.



**Gambar 4.2** *Dialog box* kesalahan pemilihan data

**c) Penanganan kesalahan input data**

Penanganan kesalahan ini terjadi apabila admin melakukan kesalahan input tipe data, contoh : tipe data telepon yang diisi bukan dengan bilangan. *Messagebox* yang keluar setelah melakukan kesalahan input data terdapat pada gambar 4.3.



**Gambar 4.3** *Dialog box* kesalahan input data

**d) Penanganan kesalahan input data kosong**

Penanganan kesalahan ini terjadi apabila admin belum memasukan data pada salah satu *field* yang harus diisi data. *Messagebox* yang keluar setelah melakukan kesalahan input data kosong terdapat pada gambar 4.4.



**Gambar 4.4** *Dialog box* kesalahan input data kosong

**e) Penanganan kesalahan panjang karakter**

Penanganan kesalahan ini terjadi apabila admin melakukan kesalahan panjang karakter, contoh : *field* yang diisi lebih dari panjang 50 *digit*. *Messagebox* yang keluar setelah melakukan kesalahan panjang karakter terdapat pada gambar 4.5.



**Gambar 4.5** *Dialog box* kesalahan panjang karakter

## 4.2.2 Pengujian dan analisis

Pada tahap pengujian dan analisis ini, dilakukan perbandingan antara kebenaran masukan dan kesesuaian data pada sistem, data yang terdapat pada sistem dapat dilihat pada lampiran A. Pengujian dan analisis program ini juga dibagi menjadi 2 yaitu : pengujian untuk user mahasiswa dan user admin.

### 4.2.2.1 Pengujian untuk user mahasiswa

Pengujian untuk user mahasiswa ini digunakan untuk menguji kebenaran sms yang dikirimkan user pada sistem, diantaranya :

#### a) Masukan Panduan

Jika user membutuhkan informasi panduan yang tersedia pada sistem. Urutannya sebagai berikut :

User kirim sms : "HELP"

Sistem kirim sms : " Untuk bantuan, kode perintah yang tersedia :  
REG?, IPK?, JU?, JK?, ALAMATYK?, ALAMATASAL?, TELP  
?, UAYK?, UASAL?, UTELP?, UP?."

Apabila user membutuhkan panduan untuk salah satu kode perintah yang tersedia, urutannya sebagai berikut :

User kirim sms : "UAYK?"

Sistem kirim sms : "Kode perintah Untuk ubah alamat Yogya, ketik  
UAYK#<nim>#<password>#<alamatykbbaru>.Contoh  
: UAYK#03523001#1234#sleman."

**b) Masukan pendaftaran**

Untuk menguji masukan pendaftaran, user melakukan sms ke server pertama kali untuk mendapatkan balasan pendaftaran berhasil. Urutannya sebagai berikut :

User kirim sms : "REG#03523027"

Sistem kirim sms : " Selamat Datang, Anda berhasil terdaftar.  
Password anda 1234, silahkan ubah password  
demi keamanan, ketik UP? untuk bantuan atau  
HELP untuk mengetahui semua kode perintah."

**c) Masukan lihat ipk**

Untuk menguji masukan lihat ipk, diasumsikan mahasiswa sudah aktif terdaftar. Mahasiswa akan dikirimkan balasan sms ipk terakhir mahasiswa yang bersangkutan. Urutannya sebagai berikut :

User kirim sms : "IPK#03523027#1234"

Sistem kirim sms : " Halo Hanif , IP Semester terakhir anda = 3  
dan IP Kumulatif anda = 3."

**d) Masukan lihat jadwal kuliah**

Untuk menguji masukan lihat jadwal kuliah, mahasiswa diasumsikan sudah mengetahui matakuliah apa saja yang bisa dilihat menggunakan sistem ini. Mahasiswa akan dikirimkan balasan sms jadwal kuliah yang ada. Urutannya sebagai berikut :

User kirim sms : "JK#Alpro1"

Sistem kirim sms : " Hari = Senin , Jam 07:00 - 08:40 , di Ruang  
= 302, dengan Dosen Pengajar = Riki,ST."

**e) Masukan lihat jadual ujian**

Untuk menguji masukan lihat jadual ujian, mahasiswa diasumsikan sudah mengetahui matakuliah apa saja yang bisa dilihat menggunakan sistem ini. Mahasiswa akan dikirimkan balasan sms jadual ujian yang ada. Urutannya sebagai berikut :

User kirim sms : "JU#Alpro1"

Sistem kirim sms : " Hari = Selasa , Jam 08:00 - 09:00 , di Ruang  
= 303."

**f) Masukan lihat alamat yogyakarta**

Untuk menguji masukan lihat alamat yogyakarta, diasumsikan mahasiswa sudah aktif terdaftar. Mahasiswa akan dikirimkan balasan sms alamat yogyakarta yang terakhir diisikan mahasiswa yang bersangkutan. Urutannya sebagai berikut :

User kirim sms : "ALAMATYK#03523027#1234"

Sistem kirim sms : " Halo Hanif , Alamat yk anda = Condong  
catur."

**g) Masukan lihat alamat asal**

Untuk menguji masukan lihat alamat asal, diasumsikan mahasiswa sudah aktif terdaftar. Mahasiswa akan dikirimkan balasan sms alamat asal yang terakhir diisikan mahasiswa yang bersangkutan. Urutannya sebagai berikut :

User kirim sms : "ALAMATASAL#03523027#1234"

Sistem kirim sms : " Halo Hanif , Alamat asal anda = Pati."

#### **h) Masukan lihat nomor telepon**

Untuk menguji masukan lihat nomor telepon, diasumsikan mahasiswa sudah aktif terdaftar. Mahasiswa akan dikirimkan balasan sms nomor telepon yang terakhir diisikan mahasiswa yang bersangkutan. Urutannya sebagai berikut :

User kirim sms : "TELP#03523027#1234"

Sistem kirim sms : " Halo Hanif , Telepon anda sekarang =  
0815345675."

#### **i) Masukan ubah alamat yogyakarta**

Untuk menguji masukan ubah alamat yogyakarta, diasumsikan mahasiswa sudah aktif terdaftar. Ubah alamat ini berfungsi untuk mengisikan alamat yogyakarta user mahasiswa terbaru tiap semesternya. Urutannya sebagai berikut :

User kirim sms : "UAYK#03523027#1234#Kentungan, jakal km 5,  
yogyakarta."

Sistem kirim sms : " Ubah alamat YK berhasil diubah."

#### **j) Masukan ubah alamat asal**

Untuk menguji masukan ubah alamat asal, diasumsikan mahasiswa sudah aktif terdaftar. Ubah alamat asal ini berfungsi untuk mengisikan alamat asal user mahasiswa terbaru tiap semesternya. Urutannya sebagai berikut :

User kirim sms : "UASAL#03523027#1234#Tambun, perumahan papanmas, Bekasi Timur."

Sistem kirim sms : " Ubah alamat asal berhasil diubah."

#### k) Masukan ubah nomor telepon

Untuk menguji masukan ubah nomor telepon, diasumsikan mahasiswa sudah aktif terdaftar. Ubah nomor telepon ini berfungsi untuk mengisikan nomor telepon user mahasiswa terbaru tiap semesternya. Urutannya sebagai berikut :

User kirim sms : "UTELP#03523027#1234#08562963625."

Sistem kirim sms : "Telepon berhasil diubah."

#### l) Masukan ubah *password*

Untuk menguji masukan ubah *password*, diasumsikan mahasiswa sudah aktif terdaftar dan memiliki *password* lama. Ubah *password* ini digunakan mahasiswa untuk mengubah *password default* nya pada saat pertama terdaftar, atau untuk mengubah *password* lamanya dengan *password* terbaru demi keamanan. Urutannya sebagai berikut :

User kirim sms : "UP#03523027#1234#adeirawan."

Sistem kirim sms : "Password berhasil diubah."

#### 4.2.2.2 Pengujian untuk user admin

Pengujian untuk user admin ini digunakan untuk menguji kebenaran masukan data yang dimasukan user pada sistem *sms server*, pengujian tersebut diantaranya :

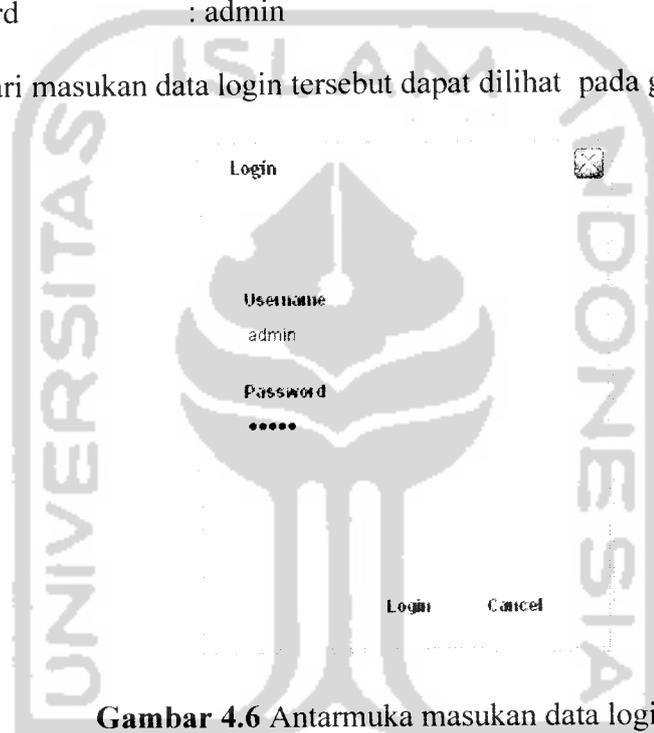
**a) Masukan data login**

Dalam pengujian masukan data login, akan diberikan contoh *form* login yang diisikan dengan data sesuai kebutuhan sistem, untuk menguji output yang dihasilkan.

Username : admin

Password : admin

Hasil dari masukan data login tersebut dapat dilihat pada gambar 4.6



**Gambar 4.6** Antarmuka masukan data login

**b) Masukan data mahasiswa**

Dalam pengujian masukan data mahasiswa, akan diberikan contoh *form* mahasiswa yang diisikan dengan data sesuai kebutuhan sistem, untuk menguji output yang dihasilkan.

NIM : 03523026  
Nama : Ade irawan  
Tempat Lahir : Kuningan  
Tanggal Lahir : 2-Juni-1985  
Jenis Kelamin : Laki-laki  
Alamat Asal : Bekasi  
Alamat YK : Sleman  
Email : ade@yahoo.com  
Telepon : 08562973925  
Nama Ayah : Budi  
Pekerjaan Ayah : Wiraswasta  
Nama Ibu : Nani  
Pekerjaan Ibu : Ibu rumah tangga  
Alamat Ortu : Bekasi

*Form* mahasiswa yang diisi data tersebut dapat dilihat pada gambar 4.7



**Data Mahasiswa**

**Data Pribadi**      **Data Keluarga**

**NIM**

**Nama**      Ade Irawan

**Tempat Lahir**      Kuningan

**Tanggal Lahir**      0209/1985 ...

**Jenis Kelamin**       Laki-Laki  
 Perempuan

**Alamat Asal**      Bekasi

**Alamat Rk**      Sleman

**Email**      ade92@yahoo.com

**Telepon**      08562973926

Done      Reset to default      Cancel

**Gambar 4.7** Antarmuka masukan data mahasiswa

Hasil dari masukan data mahasiswa tersebut dapat dilihat pada gambar 4.8

List Informasi

No	Nim	Nama Mahasiswa	Alamat Yk	Alamat Asal	Telepon	view
1	03523027	Hanif	Condong catur	Pati	0815345675	
2	03523026	Ade Irawan	Sleman	Bekasi	08562973...	

**Gambar 4.8** Output masukan data mahasiswa

### c) Masukan data ipk

Dalam pengujian masukan data ipk, akan diberikan contoh *form* ipk yang diisi dengan data sesuai kebutuhan sistem, untuk menguji output yang dihasilkan.

NIM                   : 03523026

IPS                    : 3

IPK                    : 4

Form ipk yang diisikan data tersebut dapat dilihat pada gambar 4.9

The screenshot shows a form titled "Data ipk". It contains the following fields and values:

- Input Indeks Prestasi: (empty)
- NIM Mahasiswa: 03523026
- IP Semester: 3
- IPK: 4

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

**Gambar 4.9** Antarmuka masukan data ipk

Hasil dari masukan data ipk tersebut dapat dilihat pada gambar 4.10

List Informasi:

No	NIM	IPS	IPK
1	03523027	3	3
2	03523026	3	4

**Gambar 4.10** Output masukan data ipk

**d) Masukan data jadual kuliah**

Dalam pengujian masukan data jadual kuliah, akan diberikan contoh *form* jadual kuliah yang diisikan dengan data sesuai kebutuhan sistem, untuk menguji output yang dihasilkan.

Nama Matakuliah : Metnum

Hari Kuliah : Senin

Jam Kuliah : 12:30 – 14:20

Ruang Kuliah : 103

Dosen Pengampu : Ahmad, ST

Form jadual kuliah yang diisikan data tersebut dapat dilihat pada gambar 4.11

**Data Jadwal Kuliah**

Input Jadwal Kuliah

Nama Mata Kuliah: **Metnum**

Hari Kuliah: **Senin**

Jam Kuliah: **12:30 - 14:20**

Ruang Kuliah: **103**

Dosen Pengampu: **Ahmad, ST**

Done    Reset to Default    Cancel

**Gambar 4.11** Antarmuka masukan data jadwal kuliah

Hasil dari masukan data jadwal kuliah tersebut dapat dilihat pada gambar 4.12

List Informasi

No	Matkul	Hari	Jam	Ruang	Dosen
1	Stapro	Selasa	09:30 - 12:00	103	Budiyanto, ST
2	Metnum	Senin	12:30 - 14:20	103	Ahmad, ST

**Gambar 4.12** Output masukan data jadwal kuliah

**e) Masukan data jadwal ujian**

Dalam pengujian masukan data jadwal ujian, akan diberikan contoh *form* jadwal ujian yang diisi dengan data sesuai kebutuhan sistem, untuk menguji output yang dihasilkan.

Nama Ujian : Alpro 1

Hari Ujian : Selasa

Jam Ujian : 08:00 – 09:00

Ruang Ujian : 303

*Form* jadwal ujian yang diisi dengan data tersebut dapat dilihat pada gambar 4.13

Data Jadwal Ujian

Input Jadwal Ujian

Nama Mata Kuliah: Alpro 1

Hari Ujian: Selasa

Jam Ujian: 08:00 - 09:00

Ruang Ujian: 303

Daftar Reset to Default Cancel

**Gambar 4.13** Antarmuka masukan data jadwal ujian.

Hasil dari masukan data jadwal ujian tersebut dapat dilihat pada gambar 4.14

List Informasi:

No	Matakul	Hari	Jam	ruang
1	Metnum	Selasa	09:30 - 11:00	105
2	Alpro 1	Selasa	08:00 - 09:00	303

**Gambar 4.14** Output masukan data jadwal ujian

**f) Masukan data user**

Dalam pengujian masukan data user, akan diberikan contoh *form* user yang diisi dengan data sesuai kebutuhan sistem, untuk menguji output yang dihasilkan.

Username : Admin

Password : Admin

Re-type Password : Admin

Nama : Rudi

Hak akses : Admin

Alamat : Kentungan

Email : rudi@yahoo.com

Form user yang diisikan data tersebut dapat dilihat pada gambar 4.15

**Data User**

**User**

Username admin

Password .....

Re-type Password .....

**User Data**

Nama Rudi

Hak Akses Admin

Alamat Kertungan

Email rudi@yahoo.com

Done Reset Cancel

**Gambar 4.15** Antarmuka masukan data user

Hasil dari masukan data user tersebut dapat dilihat pada gambar 4.16

List Informasi:

No	Nama	Alamat	Email	Status
1	fajarsaptono	jakal	asdfsdf	Admin
2	hanifirini	jakal km15	dasdsd	Pegawai
3	Rudi	Kertungan	rudi@yahoo.com	Admin

**Gambar 4.16** Output masukan data user

#### g) Masukan data *setting* koneksi

Dalam pengujian masukan data *setting* koneksi, akan diberikan contoh *form setting* koneksi yang diisikan dengan data sesuai kebutuhan sistem, untuk menguji output yang dihasilkan.

Nama koneksi : Motorola

Pilih port : COM1

Bit per second : 2400

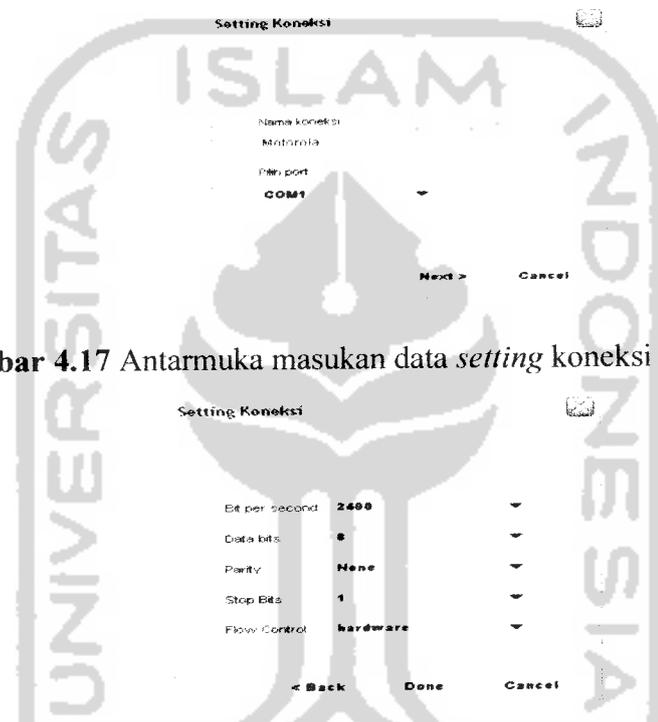
Data bits : 8

Parity : None

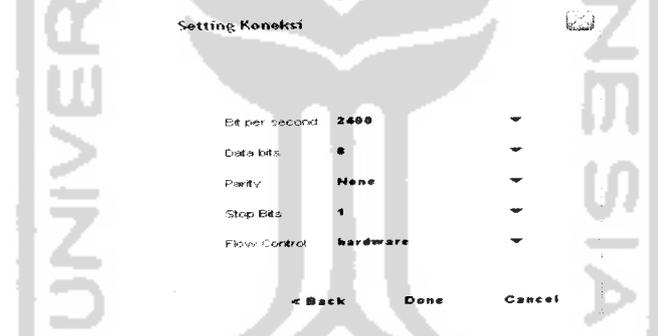
Stop bits : 1

Flow control : hardware

Form *setting* koneksi yang diisikan data tersebut dapat dilihat pada gambar 4.17 dan 4.18.



**Gambar 4.17** Antarmuka masukan data *setting* koneksi panel pertama



**Gambar 4.18** Antarmuka masukan data *setting* koneksi panel kedua

Hasil dari masukan data *setting* koneksi tersebut dapat dilihat pada gambar 4.19



**Gambar 4.19** *Output* masukan data *setting* koneksi

### 4.2.3 Hasil Analisis

Berdasarkan pengujian yang telah dilakukan diatas, maka dapat diambil kesimpulan atas kinerja dari sistem pengaksesan data pribadi mahasiswa berbasis sms ini menjadi hasil analisis yang telah didapat, yaitu :

1. Dari sisi user mahasiswa, sistem telah mampu mengenali format sms yang dikirimkan user. Sehingga apabila user tidak mengirimkan sms sesuai format, baik disengaja ataupun tidak maka sistem akan mengirimkan pesan kesalahan. Sedangkan dari sisi user admin sistem juga telah mampu menampilkan pesan kesalahan berupa *dialog box* apabila admin melakukan kesalahan baik kesalahan input atau kurang tepat dalam mengisi data.
2. Dilihat dari sisi *sms server* nya, sistem telah mampu menampilkan data-data yang berhasil diisikan oleh admin berupa tabel data sesuai yang diisikan pada form isian nya. Sedangkan dari sisi user, sistem telah mampu membalas sms sesuai kode perintah yang dikirimkan oleh user berupa pesan balasan yang tepat.

Berdasarkan pengujian dan analisis kinerja sistem baik berupa penanganan kesalahan ataupun pengujian sistem, dapat diperoleh beberapa kekurangan dan kelebihan sistem, diantaranya :

Kelebihan Sistem :

- a. Telepon atau modem yang digunakan untuk koneksi dengan sistem bersifat dinamis atau dapat diganti-ganti modem atau telepon untuk *server* nya.

- b. Format sms sudah dapat dibatasi tiap blok smsnya menggunakan tanda (#) sesuai dengan aturan yang telah ditentukan (tidak sembarang).

Kekurangan sistem :

- a. Data pribadi mahasiswa seperti ipk, jadwal kuliah, dan jadwal ujian yang diisikan belum bersifat dinamis
- b. Belum adanya fasilitas pencarian data pada sistem
- c. Belum adanya fasilitas cetak laporan sebagai bentuk pelaporan berkala kepada pihak yang bersangkutan.



## BAB V

### SIMPULAN DAN SARAN

#### 5.1 Simpulan

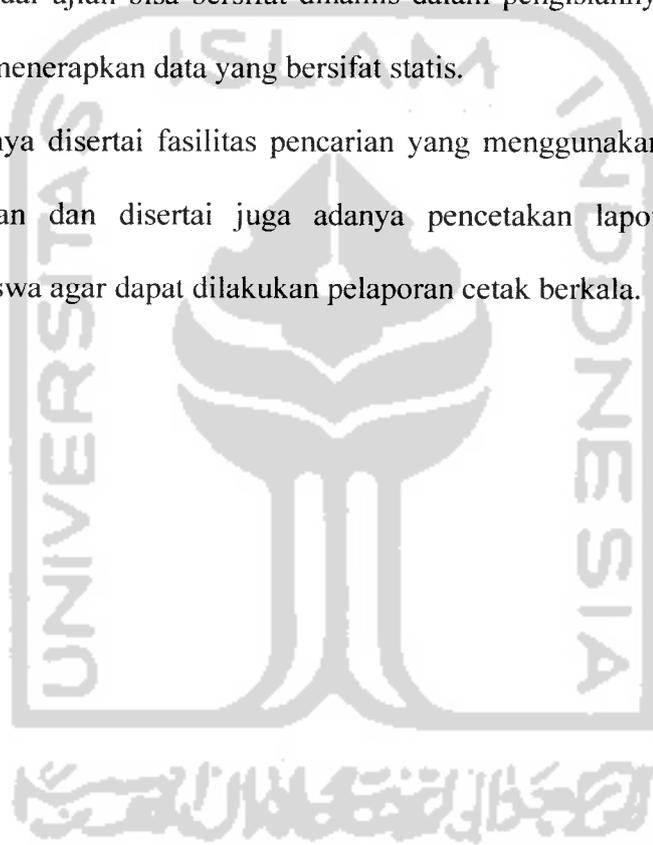
Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan, dapat disimpulkan bahwa :

1. Sistem pengaksesan data pribadi mahasiswa berbasis sms ini merupakan sistem yang mampu melakukan pengaksesan data pribadi mahasiswa seperti melihat alamat tinggal asal, alamat tinggal yogyakarta, nomor telepon dan sekaligus bisa mengubahnya melalui sms. Selain itu, sistem ini juga diberi fasilitas untuk melihat ipk mahasiswa yang bersangkutan, jadwal kuliah, serta jadwal ujian.
2. Teknologi *SMS Gateway* dapat diterapkan secara komunikatif pada sistem informasi ubah data pribadi mahasiswa. Sistem ini mempunyai data pribadi mahasiswa yang bersifat simulasi data menggunakan data statis, sehingga data yang dikeluarkan bersifat informasi data yang tersedia dalam sistem.
3. Sistem yang dibangun merupakan sistem informasi pengaksesan data pribadi mahasiswa berbasis sms yang ditekankan pada bagaimana cara menggunakan perintah-perintah *AT Command*, seperti membaca sms masuk, menghapus, dan sekaligus membalas sms tersebut secara otomatis.

## 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. Sebaiknya data yang berhubungan data akademis, seperti : ipk, jadwal kuliah, dan jadwal ujian bisa bersifat dinamis dalam pengisiannya, karena sistem ini masih menerapkan data yang bersifat statis.
2. Sebaiknya disertai fasilitas pencarian yang menggunakan salah satu metode pencarian dan disertai juga adanya pencetakan laporan data akademik mahasiswa agar dapat dilakukan pelaporan cetak berkala.



**DAFTAR PUSTAKA**

- [FAT99] Fathansyah. *Basis Data*. Bandung : Informatika Bandung, 1999.
- [HAR03] Hariyanto, Bambang. *Esensi-Esensi Bahasa Pemrograman Java*. Bandung : Informatika Bandung, 2003.
- [HAR06] Hartati, Sri. Suharto, Herry. Dan Wijono, Susilo. *Pemrograman GUI Swing Java Dengan Netbeans 5*. Yogyakarta : Andi, 2006.
- [KOM05a] Komputer, Wahana. *Pengembangan Aplikasi Sistem Informasi Akademik Berbasis SMS Dengan Java*. Jakarta : Salemba Infotek, 2005.
- [KOM05b] Komputer, Wahana. *Membuat Aplikasi Profesional Dengan Java*. Jakarta : PT. Elex Media Komputindo, 2005.
- [NUG05a] Nugroho, Adi. *Analisis dan Perancangan Sistem Informasi Dengan Metodologi Berorientasi Objek*. Bandung : Informatika Bandung, 2005.
- [NUG05b] Nugroho, Adi. *Rational Rose Untuk Pemodelan Berorientasi Objek*. Bandung : Informatika Bandung, 2005.
- [SIN04] Sinaga, Benyamin. *Pemrograman Berorientasi Objek Dengan Java*. Yogyakarta : Gava Media, 2005.

## LAMPIRAN A

Data Pribadi Mahasiswa yang terdapat pada *sms server* :

### 1. Data Mahasiswa

*List Informasi*

No	Nim	Nama Mahasiswa	Alamat Yk	Alamat Asal	Telepon	view
1	03523026	Ade Irawan	Sleman	Bekasi	08562973...	
2	03523027	Hanif	Condong catur	Pati	0815345675	
3	03523028	Bayu	Toraja, Jakal km 14	Ciamis, Bandung	08564578...	
4	03523029	Ayu	Jakal km 13,5	Purwodadi, Jateng	0813256589	
5	03523030	Mulyadi	Degolan RT 02/05	Papua	08995623...	
6	03523031	Siti	Banguntapan, Bantul	Jl. Anggajaya, Kebumen.	0856123445	
7	03523032	Slamet	Jl. Bunga, Yogyakarta	Jl. Bunga, Yogyakarta	0856789123	
8	03523033	Ramdhan	Seturan, Yogyakarta	Jl. Rengasdengklok, Ka...	0856369258	
9	03523034	Nurul	Lodadi, Jakal, Yogyakarta	Batujaya, Cikampek	0856159263	
10	03523035	Asok	Jakal km 14,5, Sleman	Bao-Bao, Kalimantan	0856357489	
11	03523036	Amal	Monjali, Yogyakarta	Kalideres, Jakbar	0813123596	
12	03523037	Agung	Jl., magelang km 14	Lemah abang, Cikarang	0852456789	
13	03523038	Nurhasanah	Janti, Yogyakarta	Jl. Manggis, Bogor Selat...	0817895236	
14	03523039	Terry	Monjali, Peum bunga m...	Lenteng agung, Jakarta...	0812348569	
15	03523040	Hasan	Jakal KM10, Sleman	Jl. Pulau seribu, Kalima...	0852695324	
16	03523041	Rahmat	Toraja, jakal km 14, 5, Y...	Jl. Pendawa lima kutow...	0859965231	
17	03523042	Danang	Jl. Malioboro Km 10. Per...	Jl. Kaca piring no 10, Ja...	0854765298	
18	03523043	Nia	Turi, rt02/06 desa jelita...	Turi, rt02/06 desa jelita...	0817598426	
19	03523044	Lisa	Jl. Mangkubumi, no 6, Y...	Jatiwaringin, Jakarta Ti...	0852158265	

### 2. Data IPK

*List Informasi*

No	NIM	IPS	IPK
1	03523026	3	4
2	03523027	3	3
3	03523028	4	4
4	03523029	3	4
5	03523030	3	4
6	03523031	3,25	3,3
7	03523032	2,75	3
8	03523033	3,75	3,7
9	03523034	2,5	3
10	03523035	2,75	3,1
11	03523036	3,5	3,6
12	03523037	3	3,2
13	03523038	3	3
14	03523039	3,7	3,6
15	03523040	3,7	3,5
16	03523041	2,5	3
17	03523042	3	3
18	03523043	3,5	3,2
19	03523044	3,7	3,5

### 3. Data Jadwal Kuliah

#### List Informasi

No	Matkul	Hari	jam	ruang	dosen
1	Agama Islam	Rabu	12:30 - 14:20	203	Chaerudin, ST.
2	Alpro1	Senin	07:00 - 08:40	302	Riki, ST
3	Alpro2	Kamis	07:00 - 08:40	301	Said, ST., MT.
4	Analog	Jumat	07:00 - 08:40	305	Adi, ST., MCS,
5	B Inggris	Rabu	09:30 - 12:00	202	Lia, ST
6	Grafkom	Jumat	12:30 - 14:20	101	Imam, ST.
7	JST	Jumat	09:30 - 12:00	304	Anang, ST
8	Kalkulus	Jumat	14:20 - 15:30	104	Ridwan, ST
9	Kewarganegaraan	Rabu	14:20 - 15:30	203	Soleh, ST.
10	Logmat	Rabu	15:30 - 17:00	205	Haris, ST., MCS
11	Metnum	Senin	12:30 - 14:20	103	Ahmad, ST
12	Metopei	Jumat	09:30 - 12:00	303	Indra, ST
13	Muamalah	Senin	15:30 - 17:00	302	Amal, ST
14	Pancasila	Selasa	12:30 - 14:20	105	Dian, ST
15	PBO	Jumat	07:00 - 08:40	101	Nuh, ST., MSI
16	PTI	Selasa	09:30 - 12:00	102	Lina, ST., MCS.
17	Simbada	Rabu	07:00 - 08:40	201	Indah, ST., M.St.
18	Sister	Selasa	07:00 - 08:40	101	Rini, ST
19	Stapro	Selasa	09:30 - 12:00	103	Budryanto, ST

### 4. Data Jadwal Ujian

#### List Informasi

No	Matkul	Hari	jam	ruang
1	Agama Islam	Kamis	15:30 - 17:00	205
2	Alpro 1	Selasa	08:00 - 09:00	303
3	Alpro2	Kamis	15:30 - 17:00	203
4	Analog	Senin	08:00 - 09:00	101
5	B Inggris	Kamis	09:30 - 11:00	304
6	Grafkom	Senin	12:30 - 14:20	104
7	JST	Kamis	12:30 - 14:20	205
8	Kalkulus	Senin	09:30 - 11:00	102
9	Kewarganegaraan	Kamis	08:00 - 09:00	303
10	Logmat	Kamis	12:30 - 14:20	301
11	Metnum	Selasa	09:30 - 11:00	105
12	Metopei	Senin	14:20 - 15:20	105
13	Muamalah	Rabu	09:30 - 11:00	104
14	Pancasila	Jumat	14:20 - 15:20	203
15	PBO	Senin	15:30 - 17:00	201
16	PTI	Jumat	09:30 - 11:00	202
17	Simbada	Kamis	08:00 - 09:00	305
18	Sister	Jumat	08:00 - 09:00	201
19	Stapro	Rabu	14:20 - 15:20	105

## LAMPIRAN B

Source code *formatValidation()* dan *messageProcessing()* yang terletak pada class *sms*.

### 1. Method *formatValidation()*

```

/* Method : untuk validasi format perintah
*/

private boolean formatValidation(String msg){// cek terhadap
format yang sesungguhnya.blok.

boolean stat=false;
String[] hashStr=msg.toUpperCase().split("#");
if(hashStr[0].equals("REG")){
    if(hashStr.length==2){
        stat=true;
    }else{
        confMsg="Format salah.Ketik REG? untuk bantuan, atau
mengetahui semua kode perintah ketik HELP";
    }
}else if(hashStr[0].equals("HELP")){
    if(hashStr.length==1){
        stat=true;
    }else{
        confMsg="Format salah.Ketik HELP untuk mengetahui
semua kode perintah";
    }
}else
if(hashStr[0].equals("REG?")||hashStr[0].equals("IPK?")||
hashStr[0].equals("JK?")||hashStr[0].equals("JU?")||
hashStr[0].equals("ALAMATYK?")||
hashStr[0].equals("ALAMATASAL?")||
hashStr[0].equals("TELP?")||hashStr[0].equals("UAYK?")||
hashStr[0].equals("UASAL?")||
hashStr[0].equals("UTELP?")||
hashStr[0].equals("UP?")){
    stat=true;
}else if(hashStr[0].equals("ALAMATYK")){
    if(hashStr.length==3){
        stat=true;
    }else{
        confMsg="Format salah. Ketik ALAMATYK? untuk bantuan
atau Untuk mengetahui semua kode perintah,
ketik HELP";
    }
}else if(hashStr[0].equals("ALAMATASAL")){
    if(hashStr.length==3){
        stat=true;
    }else{
        confMsg="Format salah. Ketik ALAMATASAL? untuk
bantuan atau Untuk mengetahui semua kode
perintah, ketik HELP";
    }
}

```

```

    }
}else if(hashStr[0].equals("TELP")){
    if(hashStr.length==3){
        stat=true;
    }else{
        confMsg="Format salah. Ketik TELP? untuk bantuan
                atau Untuk mengetahui semua kode perintah,
                ketik HELP";
    }
}
}else if(hashStr[0].equals("IPK")){
    if(hashStr.length==3){
        stat=true;
    }else{
        confMsg="Format salah. Ketik IPK? untuk bantuan atau
                Untuk mengetahui semua kode perintah, ketik
                HELP";
    }
}
}
}else if(hashStr[0].equals("JK")){
    if(hashStr.length==2){
        stat=true;
    }else{
        confMsg="Format salah. Ketik JK? untuk bantuan atau
                Untuk mengetahui semua kode perintah, ketik
                HELP";
    }
}
}else if(hashStr[0].equals("JU")){
    if(hashStr.length==2){
        stat=true;
    }else{
        confMsg="Format salah. Ketik JU? untuk bantuan atau
                Untuk mengetahui semua kode perintah, ketik
                HELP";
    }
}
}
}else if(hashStr[0].equals("UAYK")){
    if(hashStr.length==4){
        stat=true;
    }else{
        confMsg="Format salah. Ketik UAYK? untuk bantuan
                atau Untuk mengetahui semua kode perintah,
                ketik HELP";
    }
}
}
}else if(hashStr[0].equals("UASAL")){
    if(hashStr.length==4){
        stat=true;
    }else{
        confMsg="Format salah. Ketik UASAL? untuk bantuan
                atau Untuk mengetahui semua kode perintah,
                ketik HELP";
    }
}
}
}else if(hashStr[0].equals("UTELP")){//jangan terlalu
                                     dikekang
    if(hashStr.length==4){
        stat=true;
    }else{

```

```

        confMsg="Format salah. Ketik UTELP? untuk bantuan
        atau Untuk mengetahui semua kode perintah,
        ketik HELP";
    }
} else if(hashStr[0].equals("UP")){//jangan terlalu dikekang
    if(hashStr.length==4){
        stat=true;
    } else{
        confMsg="Format salah. Ketik UP? untuk bantuan atau
        Untuk mengetahui semua kode perintah, ketik
        HELP";
    }
} else{
    confMsg="Perintah tidak dikenal. Ketik HELP untuk
    mengetahui semua kode perintah.";
}
return stat;
}

```

## 2. Method *messageProcessing()*

```

/* Method : untuk pengolahan SMS yang telah diseleksi formatnya.
*/
/***** untuk pengolahan pesan yang masuk. *****/

public void messageProcessing(String msg){
    if(formatValidation(msg)){
        String[] hashStr=msg.toUpperCase().split("#");
        if(hashStr[0].equals("REG")){
            String nim=msg.toUpperCase().split("#")[1];
            if(mhs.isMahasiswa(nim)){
                /***** Pertama kali REG *****/
                if(!user.isAlreadyActivatedMahasiswa(nim)){
                    user.setActivatedMahasiswa(nim);
                    confMsg="Selamat Datang, Anda berhasil
                    terdaftar. Password anda 1234, ubah
                    password dgn format:
                    UP#<nim>#<password>#<password baru>";
                } else{
                    confMsg="Anda telah terdaftar.Gunakan
                    fasilitas UPDATE data via HP";
                }
            } else{
                confMsg="NIM tdk dikenal,ulangi dengan
                mengisikan NIM yg benar.";
            }
        } else if(hashStr[0].equals("HELP")) ||
            hashStr[0].equals("REG?") ||
            hashStr[0].equals("IPK?") ||
            hashStr[0].equals("JU?") ||
            hashStr[0].equals("JK?") ||
            hashStr[0].equals("ALAMATYK?") ||
            hashStr[0].equals("ALAMATASAL?") ||

```

```

hashStr[0].equals("TELP?") ||
hashStr[0].equals("UAYK?") ||
hashStr[0].equals("UASAL?") ||
hashStr[0].equals("UTELP?") ||
hashStr[0].equals("UP?")
){
if(hashStr[0].equals("HELP")){
    confMsg="Untuk bantuan, kode perintah yang
    tersedia :
    REG?, IPK?, JU?, JK?, MATKUL?, ALAMATYK?, ALAM
    ATASAL?, TELP?, UAYK?, UASAL?, UTELP?, UP?";
}else if(hashStr[0].equals("REG?")){
    confMsg="Kode perintah Untuk daftar, ketik
    REG#<nim>.Contoh REG#03523001";
}else if(hashStr[0].equals("IPK?")){
    confMsg="Kode perintah Untuk lihat IPK, ketik
    IPK#<nim>#<password>.Contoh
    IPK#03523001#abcde";
}else if(hashStr[0].equals("JK?")){
    confMsg="Kode perintah Untuk lihat Jadwal
    kuliah, ketik JK#<nama matakul>.Contoh
    JK#alpro1, untuk melihat nama-nama
    matakul ketik MATKUL";
}else if(hashStr[0].equals("JU?")){
    confMsg="Kode perintah Untuk lihat Jadwal
    ujian, ketik JU#<nama mata kuliah>.Contoh
    JU#alpro1, untuk melihat nama-nama
    matakul ketik MATKUL";
}else if(hashStr[0].equals("ALAMATYK?")){
    confMsg="Kode perintah Untuk lihat alamat
    Jogja, ketik
    ALAMATYK#<nim>#<password>.Contoh
    ALAMATYK#03523001#abcde";
}else if(hashStr[0].equals("ALAMATASAL?")){
    confMsg="Kode perintah Untuk lihat alamat
    asal, ketik
    ALAMATASAL#<nim>#<password>.Contoh
    ALAMATASAL#03523001#abcde";
}else if(hashStr[0].equals("TELP?")){
    confMsg="Kode perintah Untuk lihat telp, ketik
    TELP#<nim>#<password>.Contoh
    TELP#03523001#abcde";
}else if(hashStr[0].equals("UAYK?")){
    confMsg="Kode perintah Untuk ubah alamat
    Yogya, ketik
    UAYK#<nim>#<password>#<alamatyk
    baru>.Contoh
    UAYK#03523001#abcde#sleman";
}else if(hashStr[0].equals("UASAL?")){
    confMsg="Kode perintah Untuk ubah alamat
    asal, ketik
    UASAL#<nim>#<password>#<alamatasal
    baru>.Contoh
    UASAL#03523001#abcde#bekasi";
}else if(hashStr[0].equals("UTELP?")){
    confMsg="Kode perintah Untuk ubah telp, ketik
    UTELP#<nim>#<password>#<telp

```

```

        baru>.Contoh
        UTELP#03523001#abcde#08564736";
    }else if(hashStr[0].equals("UP?")){
        confMsg="Kode perintah Untuk ubah password, ketik
        UP#<nim>#<password>#<password
        baru>.Contoh UP#03523001#abcde#fghij";
    }
}
}else if(hashStr[0].equals("UP")){
    String nim=msg.split("#")[1];
    String oPwd=msg.split("#")[2]; //password lama
    String nPwd=msg.split("#")[3]; // password baru
    if(user.loginUser(nim,oPwd,0)){
        if(vld.passwordValidation(nPwd)){ // jika valid
            disimpan.
            user.changePassword(nim,nPwd,0);
            confMsg="Password berhasil diubah";
        }else{
            confMsg="Password baru min 4 digit dan maks
            10 digit";
        }
    }
}
}else{
    confMsg="Password salah (case sensitive) atau
    belum terdaftar.Untuk daftar ketik
    REG#<NIM>";
}
}
}else if(hashStr[0].equals("ALAMATYK")){
    String nim=msg.split("#")[1];
    String pwd=msg.split("#")[2]; //password lama
    if(user.loginUser(nim,pwd,0)){
        String[] slcData=mhs.getSelectedData(nim);
        confMsg="Halo "+slcData[1]+" , Alamat yk anda =
        "+slcData[5]+"";
    }else{
        confMsg="Password salah (case sensitive) atau
        belum terdaftar.Untuk daftar, ketik
        REG#<NIM>";
    }
}
}
}else if(hashStr[0].equals("ALAMATASAL")){
    String nim=msg.split("#")[1];
    String pwd=msg.split("#")[2]; //password lama
    if(user.loginUser(nim,pwd,0)){
        String[] slcData=mhs.getSelectedData(nim);
        confMsg="Halo "+slcData[1]+" , Alamat asal anda
        = "+slcData[6]+"";
    }else{
        confMsg="Password salah (case sensitive) atau
        belum terdaftar.Untuk daftar, ketik
        REG#<NIM>";
    }
}
}
}else if(hashStr[0].equals("TELP")){
    String nim=msg.split("#")[1];
    String pwd=msg.split("#")[2]; //password lama
    if(user.loginUser(nim,pwd,0)){
        String[] slcData=mhs.getSelectedData(nim);
        confMsg="Halo "+slcData[1]+" , Telepon anda
        sekarang = "+slcData[8]+"";
    }else{

```

```

        confMsg="Password salah (case sensitive) atau
        belum terdaftar.Untuk daftar,ketik
        REG#<NIM>";
    }
}
}else if(hashStr[0].equals("IPK")){
    String nim=msg.split("#")[1];
    String pwd=msg.split("#")[2];//password lama
    if(user.loginUser(nim,pwd,0)){
        String[] slcData=mhs.getSelectedData(nim);
        confMsg="Halo "+slcData[1]+" , IP Semester
        terakhir anda = "+slcData[15]+" dan IP
        Kumulatif anda = "+slcData[16]+"";
    }else{
        confMsg="Password salah (case sensitive) atau
        belum terdaftar.Untuk daftar,ketik
        REG#<NIM>";
    }
}
}else if(hashStr[0].equals("JK")){
    try{
        String[] rs=jk.getSelectedJK(hashStr[1]);
        confMsg="Hari = "+rs[1]+", "+rs[2]+" , di Ruang
        = "+rs[3]+", dengan Dosen Pengajar =
        "+rs[4]+"";
    }catch (ArrayIndexOutOfBoundsException ex){
        confMsg="Nama mata kuliah tidak dikenal";
    }
}
}else if(hashStr[0].equals("JU")){
    try{
        String[] rs=ju.getSelectedJU(hashStr[1]);
        confMsg="Hari = "+rs[1]+" "+rs[2]+" , di Ruang
        = "+rs[3]+"";
    }catch (ArrayIndexOutOfBoundsException ex){
        confMsg="Nama mata kuliah tidak dikenal";
    }
}
}else if(hashStr[0].equals("UAYK")){
    String nim=hashStr[1];
    String pwd=msg.split("#")[2];//password lama
    if(user.loginUser(nim,pwd,0)){
        String almtBaru=msg.split("#")[3];
        if(vld.maxLen(almtBaru,50)){
            confMsg="Alamat YK baru maksimal 50
            karakter";
        }else{
            String[] rs=mhs.getSelectedData(nim);//
            result dari data
            base

            String[] dataSet=new
            String[]{"",rs[1],rs[2],rs[3],rs[4],almtBaru
            ,rs[6],rs[7],rs[8],
            rs[9],rs[10],rs[11],rs[12],rs[13],rs[0]};//
            untuk input data

            mhs.editData(dataSet);
            confMsg="Ubah alamat YK berhasil diubah";
        }
    }
}
}else{

```

```

        confMsg="Password salah (case sensitive) atau
        belum terdaftar.Untuk daftar,ketik
        REG#<NIM>";
    }
}
}else if(hashStr[0].equals("UASAL")){
    String nim=hashStr[1];
    String pwd=msg.split("#")[2];//password lama
    if(user.loginUser(nim,pwd,0)){
        String almtBaru=msg.split("#")[3];
        if(vld.maxLen(almtBaru,50)){
            confMsg="Alamat asal baru maksimal 50
            karakter";
        }else{
            String[] rs=mhs.getSelectedData(nim);//
            result dari data base
            String[] dataSet=new
            String[]{"",rs[1],rs[2],rs[3],rs[4],rs[5],al
            mtBaru,rs[7],rs[8],
            rs[9],rs[10],rs[11],rs[12],rs[13],rs[0]};//
            untuk input data
            mhs.editData(dataSet);
            confMsg="Ubah alamat asal berhasil diubah";
        }
    }
}else{
    confMsg="Password salah (case sensitive) atau
    belum terdaftar.Untuk daftar,ketik
    REG#<NIM>";
}
}
}else if(hashStr[0].equals("UTELP")){
    String nim=hashStr[1];
    String pwd=msg.split("#")[2];//password lama
    if(user.loginUser(nim,pwd,0)){
        String telp=hashStr[3];
        if(!vld.isNumeric(telp) || vld.maxLen(telp,20)){
            confMsg="Telepon harus diisi dengan angka &
            tdk lebih dari 20 digit";
        }else{
            String[] rs=mhs.getSelectedData(nim);//
            result dari data base
            String[] dataSet=new
            String[]{"",rs[1],rs[2],rs[3],rs[4],rs[5],rs
            [6],rs[7],telp,
            rs[9],rs[10],rs[11],rs[12],rs[13],rs[0]};//
            untuk input data
            mhs.editData(dataSet);
            confMsg="Telepon berhasil diubah";
        }
    }
}else{
    confMsg="Password salah (case sensitive) atau
    belum terdaftar.Untuk daftar,ketik
    REG#<NIM>";
}
}
}
savedInbox=true;//baru bisa disimpan dalam basis data.
}
}
}

```