

**APLIKASI SISTEM PEMINJAMAN BUKU PERPUSTAKAAN
MENGUNAKAN TEKNOLOGI SMART CARD DAN
SMS GATEWAY**

SKRIPSI



Disusun oleh :

**Nur Toha Efendi
NIM. 0812531**

MILIK
PERPUSTAKAAN
ITN MALANG

**JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2012**

3045

REPUBLICAN LEADERSHIP MOVEMENTS
NATIONAL LEADERSHIP MOVEMENT
NATIONAL LEADERSHIP MOVEMENT
NATIONAL LEADERSHIP MOVEMENT

THE LEADERSHIP
MOVEMENTS
MOVEMENTS

3045

THE LEADERSHIP

MOVEMENTS
MOVEMENTS
MOVEMENTS

LEMBAR PERSETUJUAN

APLIKASI SISTEM PEMINJAMAN BUKU PERPUSTAKAAN MENGUNAKAN TEKNOLOGI SMART CARD DAN SMS GATEWAY

SKRIPSI

*Disusun dan Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh
Gelar Sarjana Teknik Komputer Strata Satu (S-1)*

Disusun oleh :

NUR TOHA EFENDI

NIM. 0812531

Mengetahui,

Ketua Program Studi Teknik Elektro S-1

Ir. Yusuf Ismail Nakhoda, MT

NIP. Y. 1018800189

Diperiksa dan Disetujui

Dosen Pembimbing I

Dosen Pembimbing II

Dr. Eng. Aryanto Soetedjo, ST, MT

NIP. P. 1030800417

M. Ibrahim Ashari, ST, MT

NIP. P. 1030100358

**JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG**

SURAT PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini :

Nama : Nur Toha Efendi
NIM : 0812531
Program Studi : Teknik Elektro S-1
Konsentrasi : Teknik Komputer

Dengan ini menyatakan bahwa Skripsi yang saya buat adalah hasil karya sendiri, tidak merupakan plagiasi dari karya orang lain. Dalam Skripsi ini tidak memuat karya orang lain, kecuali dicantumkan sumbernya sesuai dengan ketentuan yang berlaku.

Demikian surat pernyataan ini saya buat, dan apabila di kemudian hari ada pelanggaran atas surat pernyataan ini, saya bersedia menerima sanksinya.

Malang, 30 Agustus 2012

Yang membuat Pernyataan,



Nur Toha Efendi
NIM. 0812531

ABSTRAK

APLIKASI SISTEM PEMINJAMAN BUKU PERPUSTAKAAN MENGUNAKAN TEKNOLOGI SMART CARD DAN SMS GATEWAY

Nur Toha Efendi, NIM 0812531

Dosen Pembimbing: Dr. Eng. Aryuanto S, ST, MT dan M. Ibrahim Ashari, ST, MT

Kemajuan teknologi khususnya teknologi informasi yang semakin hari mengalami perkembangan yang sangat pesat menyebabkan pemanfaatan dan penggunaan teknologi informasi sudah hampir memasuki seluruh aspek kehidupan dan aktifitas manusia. Manajemen perpustakaan yang kian berkembang dinilai perlu memanfaatkan kehadiran teknologi informasi tersebut. Sistem peminjaman buku perpustakaan yang dibuat adalah sebuah sistem aplikasi yang digunakan untuk melakukan transaksi peminjaman buku perpustakaan. Dalam sistem ini menggunakan kartu *smart card* yang akan diisi data identitas mahasiswa. Sehingga nantinya kartu *smart card* ini akan menggantikan formulir peminjaman buku perpustakaan yang sah, dan sistem ini juga menggunakan teknologi informasi yaitu *SMS Gateway* yang mana berfungsi sebagai informasi peringatan batas peminjaman buku dan digunakan sebagai media untuk memperpanjang batas peminjaman buku. Dengan tujuan untuk memberikan kemudahan layanan yang efisien, cepat, dan tepat guna.

Hasil pengujian yang dilakukan pada sistem aplikasi peminjaman buku ini menunjukkan bahwa sistem peminjaman buku ini dapat digunakan untuk menggantikan sistem lama. Dimana sistem ini kartu *smart card* dapat digunakan sebagai kartu peminjaman buku yang sah, tanpa harus lagi menulis formulir peminjaman buku kembali, dan dari hasil pengujian sistem mengirimkan pesan pemberitahuan dapat ditunjukkan bahwa pemrosesan setiap pesan dari sistem ke anggota berjalan cukup baik, dengan waktu rata-rata 10 detik per sms yang terkirim ke nomor telepon anggota. Dan hasil pengujian pepanjangan melalui SMS menunjukkan bahwa sistem dapat berjalan dengan baik juga, dengan waktu rata-rata 20 detik per sms mulai dari anggota mengirimkan pesan ke sistem sampai sistem mengirim balik pesan perpanjangan secara otomatis.

Kata Kunci : Komputer, *Smart Card* dan *SMS Gateway*.

KATA PENGANTAR

Puji Syukur ke hadirat Tuhan Yang Maha Kuasa atas segala limpahan berkat dan rahmat-Nya sehingga laporan skripsi berjudul Sistem Aplikasi Peminjaman Buku Perpustakaan menggunakan Teknologi *Smart Card* dan *SMS Gateway* dapat terselesaikan.

Penulisan skripsi ini dibuat untuk memenuhi salah satu syarat dalam memperoleh gelar sarjana teknik. Ucapan terima kasih yang sebesar-besarnya penulis sampaikan pada:

1. Bapak Ir. Soeparno Djiwo, MT selaku Rektor Institut Teknologi Nasional Malang.
2. Bapak Ir. Sidik Noertjahjono, MT selaku Dekan Fakultas Teknologi Industri Institut Teknologi Nasional Malang.
3. Bapak Ir. Yusuf Ismail Nakhoda, MT selaku Ketua Program Studi Teknik Elektro S-1
4. Bapak Dr. Eng. Aryunto Soetedjo, ST, MT selaku Dosen Pembimbing I.
5. Bapak M. Ibrahim Ashari, ST, MT selaku Dosen Pembimbing II.
6. Ayah Ibu tercinta dan semua keluarga besar yang selama ini selalu mendukung dan mendoakan penulis.
7. Semua pihak yang telah membantu dalam penulisan dan penyusunan skripsi ini.

Penulis menyadari bahwa penulisan skripsi ini masih jauh dari sempurna, untuk itu kritik dan saran dari pembaca sangat penulis harapkan untuk perbaikan skripsi ini.

Malang, Agustus 2012

Penulis

DAFTAR ISI

Lembar Persetujuan	i
Surat Pernyataan Orisinalitas	ii
Abstrak	iii
Kata Pengantar	iv
Daftar Isi	v
Daftar Gambar	viii
Daftar Tabel	x
BAB I PENDAHULUAN	
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan dan Manfaat Penelitian.....	2
1.4.1 Tujuan.....	2
1.4.2 Manfaat.....	3
1.5 Metodologi Penelitian.....	3
1.6 Sistematika Penulisan.....	4
BAB II LANDASAN TEORI	
2.1 Perpustakaan.....	5
2.1.1 Pengertian Perpustakaan.....	5
2.1.2 Tujuan Perpustakaan.....	6
2.1.3 Fungsi Perpustakaan.....	6
2.1.4 Layanan Perpustakaan.....	8
2.2 Smart Card.....	8
2.2.1 Pengertian Smart Card.....	8
2.2.2 Jenis Memori pada Smart Card.....	9
2.2.3 Tipe-Tipe Smart Card.....	10
2.2.4 Keuntungan menggunakan Smart Card.....	10
2.2.5 Komunikasi Smart Card.....	12
2.2.5.1 Model Komunikasi Smart Card.....	13
2.2.5.2 Format APDU.....	14

2.3 Smart Card Reader	14
2.4 SMS Gateway.....	15
2.4.1 Element Utama Arsitektur SMS.....	17
2.4.1.1 Short Message Entity (SME)	17
2.4.1.2 SMS Service Centre (SMSC).....	17
2.4.1.3 Base Station Subsystem (BSS)	18
2.4.2 Keuntungan SMS Gateway	18
2.5 AT Command	18
2.6 Borland Delphi	21
2.7 Microsoft Acces.....	23

BAB III ANALISA DAN PERANCANGAN SISTEM

3.1 Anlisa Sistem.....	24
3.1.1 Identifikasi Masalah	24
3.2 Gambaran Umum Sistem	24
3.3 Perancangan Sistem.....	27
3.3.1 Perancangan Sistem Smart Card	27
3.3.1.1 Smart Card yang digunakan.....	28
3.3.1.2 Smart Card Reader yang digunakan	29
3.3.2 Perancangan Sistem SMS Gateway.....	29
3.4 Flowchart Sistem.....	31
3.4.1 Flowchart Admin.....	31
3.4.2 Flowchart Pendaftaran.....	33
3.4.3 Flowchart Transaksi Peminjaman	34
3.4.4 Flowchart Pengembalian Buku	35
3.5 Perencanaan dan Pembuatan Perangkat Lunak	35
3.5.1 Desain Halaman Login.....	35
3.5.2 Desain Halaman Menu Utama	36
3.5.3 Desain Halaman Menu Tambah Buku	36
3.5.4 Desain Halaman Menu Pendaftaran Anggota	37
3.5.5 Desain Halaman Menu Transaksi.....	37
3.5.6 Desain Halaman Menu Laporan.....	38
3.5.7 Desain program Komunikasi Serial	39

BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1	Kebutuhan Sistem.....	40
4.2	Pengujian Hardware	41
4.2.1	Pengujian Smart Card	41
4.2.1.1	Langkah Pengujian.....	41
4.2.1.2	Hasil Pengujian	42
4.2.2	Pengujian Modem GSM Wavecom	44
4.2.2.1	Langkah Pengujian.....	44
4.2.2.2	Hasil Pengujian	45
4.3	Pengujian perangkat Lunak	46
4.3.1	Tampilan Utama Aplication	46
4.3.1.1	Menu Utama Program.....	48
4.3.1.2	Pengujian Program Tambah Buku.....	49
4.3.1.3	Pengujian Program Daftar Anggota.....	50
4.3.1.4	Pengujian Program Menu Transaksi.....	53
4.3.1.5	Pengujian Pesan Peringatan Batas Peminjaman	56
4.3.1.6	Pengujian Anggota Mengirim Pesan Perpanjangan.....	57
4.3.1.7	Pengujian Sistem Mengirim Pesan Balasan Perpanjangan..	57
4.3.1.7	Pengujian Pesan Sistem Menolak Perpanjangan	58
4.4	Hasil Pengujian.....	59
4.4.1	Hasil Pengujian Baca kartu Smart Card	59
4.4.2	Hasil Pengujian Sistem Mengirim Pesan Pemberitahuan.....	59
4.4.3	Hasil Pengujian Perpanjangan melalui SMS	60
4.4.4	Hasil Pengujian Aplikasi	61

BAB V PENUTUP

5.1	Kesimpulan.....	63
5.2	Saran	63

DAFTAR PUSTAKA	64
-----------------------------	----

LAMPIRAN

DAFTAR GAMBAR

Gambar 2.1 Bentuk <i>Smart Card</i>	8
Gambar 2.2 Delapan titik kontak pada smart card	9
Gambar 2.3 Perbedaan antara kartu <i>contact</i> (a) dan kartu <i>contactless</i> (b)	12
Gambar 2.4 Komunikasi Smart Card	14
Gambar 2.5 <i>Smart Card Reader</i> tipe ACR30	15
Gambar 2.6 IDE Borland Delphi7	22
Gambar 3.1 Diagram <i>Desain Sistem</i>	26
Gambar 3.2 Perancangan <i>Smart Card</i>	27
Gambar 3.3 Kartu <i>Smart Card</i> SLE4442	29
Gambar 3.4 <i>Smart Card Reader</i> ACR30	29
Gambar 3.5 Cara kerja SMS gateway secara umum	30
Gambar 3.6 Modem Wavecom M1306B Q2403A USB	31
Gambar 3.7 <i>Flowchart</i> Admin	32
Gambar 3.8 <i>Flowchart</i> Pendaftaran	33
Gambar 3.9 <i>Flowchart</i> Transaksi Peminjaman	34
Gambar 3.10 <i>Flowchart</i> Pengembalian Buku	35
Gambar 3.11 Desain Login	35
Gambar 3.12 Desain Menu Halaman Utama	36
Gambar 3.13 Desain Menu Tambah Buku	36
Gambar 3.14 Desain Menu Pendafran Anggota	37
Gambar 3.15 Desain Menu Transaksi	38
Gambar 3.16 Desain Menu Laporan	38
Gambar 3.17 Komunikasi serial antara aplikasi dengan modem dan reader	39
Gambar 4.1 <i>Conection Name</i>	41
Gambar 4.2 Pilih com port dimana <i>smart card reader</i> terhubung dengan PC	42
Gambar 4.3 Mengubah <i>Bits per second reader</i>	42
Gambar 4.4 Hasil pengujian <i>reader</i> dapat membaca kartu <i>smart card</i>	43
Gambar 4.5 Hasil pengujian status <i>reader</i> terputus dengan kartu	43
Gambar 4.6 <i>Conection Name</i>	44
Gambar 4.7 Pilih port com dimana modem terhubung dengan PC	45
Gambar 4.8 Mengubah <i>Bits per second modem</i>	45

Gambar 4.9 Hasil pengujian koneksi Modem GSM.....	46
Gambar 4.10 Tampilan Utama Login.....	47
Gambar 4.11 Login Sukses.....	47
Gambar 4.12 Login Gagal	47
Gambar 4.13 Tampilan Program Menu Tool	48
Gambar 4.14 Program Menu Aplikasi.....	48
Gambar 4.15 Program Menu Laporan.....	48
Gambar 4.16 Tampilan Menu Tambah Data Buku	49
Gambar 4.17 Tampilan menginputkan Data Buku baru.....	49
Gambar 4.18 Tampilan pesan dialog informasi Data Buku Tersimpan	49
Gambar 4.19 Tampilan Data Buku Tersimpan.....	50
Gambar 4.20 Tampilan Pesan Dialog Informasi salah menginputkan No. Register Buku	50
Gambar 4.21 Tampilan Menu Pilihan Daftar Anggota Baru.....	51
Gambar 4.22 Tampilan Aplikasi Daftar Anggota Baru.....	51
Gambar 4.23 Tampilan mengelola Daftar Anggota Baru.....	51
Gambar 4.24 Tampilan mengupload photo anggota.....	52
Gambar 4.25 Tampilan Pesan Dialog Informasi Data Anggota Baru Tersimpan	52
Gambar 4.26 Tampilan Data Anggota Baru Berhasil Disimpan.....	52
Gambar 4.27 Tampilan Pesan Dialog Informasi menginputkan No. Angggota yang sama.....	53
Gambar 4.28 Tampilan Pesan Dialog Informasi menginputkan No. Telp yang sama ...	53
Gambar 4.29 Tampilan sistem menampilkan data identitas anggota	54
Gambar 4.30 Tampilan memasukkan data transaksi peminjaman buku	54
Gambar 4.31 Tampilan Pesan Dialog Informasi Data Transaksi Tersimpan	55
Gambar 4.32 Tampilan Data transaksi tersimpan pada sistem.....	55
Gambar 4.33 Tampilan Pesan Dialog Informasi salah menginputkan No. Register Buku	55
Gambar 4.34 Tampilan pesan pemberitahuan batas peminjaman buku ke No. Telp 1 ..	56
Gambar 4.35 Tampilan pesan peringatan batas peminjaman buku ke No. Telp 2	56
Gambar 4.36 Tampilan Mengirim Pesan Perpanjangan.....	57
Gambar 4.37 Tampilan sistem mengirimkan pesan balasan perpanjangan.....	58
Gambar 4.38 Tampilan pesan sistem menolak perpanjangan	58

DAFTAR TABEL

Tabel 2.1 Tabel Control AT Comm.....	19
Tabel 2.2 Tabel fungsi AT Comm secara umum.....	20
Tabel 2.3 Tabel AT Command untuk SMS.....	20
Tabel 4.1 Pengujian Reader baca kartu smart card	43
Tabel 4.2 Hasil pengujian reader terputus dengan kartu	44
Tabel 4.3 Hasil Pengujian Baca kartu Smart Card.....	59
Tabel 4.4 Pengujian Sistem Mengirim Pesan Pemberitahuan.....	59
Tabel 4.5 Pengujian Perpanjangan melalui SMS	60
Tabel 4.6 Hasil Pengujian Sistem Aplikasi secara umum.....	61

BAB I PENDAHULUAN

1.1 Latar Belakang

Kemajuan teknologi khususnya teknologi informasi yang semakin hari mengalami perkembangan yang sangat pesat menyebabkan pemanfaatan dan penggunaan teknologi informasi sudah hampir memasuki seluruh aspek kehidupan dan aktifitas manusia. Manajemen perpustakaan yang kian berkembang dinilai perlu memanfaatkan kehadiran teknologi informasi tersebut.

Sebagaimana diketahui, sistem peminjaman buku pada perpustakaan yang berjalan saat ini masih menggunakan sistem manual, dalam pengertian bahwa seorang mahasiswa jika ingin melakukan transaksi peminjaman buku, mahasiswa harus menulis identitas mereka seperti nama, nim, jurusan, angkatan, no seri buku dan judul buku yang akan dipinjam pada selembor formulir peminjaman secara manual yang disediakan oleh petugas perpustakaan, sistem tersebut tidak efisien bagi mahasiswa dan admin, dikarenakan seorang admin harus mengumpulkan formulir peminjaman satu persatu, dimana hal tersebut rentan menimbulkan resiko kehilangan salah satu formulir. Selain itu seorang mahasiswa sering sekali telat mengembalikan buku tepat pada tanggal yang sudah ditentukan. Untuk menghindari hal tersebut maka dibuatlah sistem atau aplikasi yang bertujuan untuk memberikan kemudahan layanan yang cepat, efisien dan tepat guna bagi peminjam dan seorang admin.

Pada pengimplementasian sistem ini menggunakan teknologi *Smart Card* dan *SMS Gateway*, dimana teknologi *Smart Card* menawarkan kemudahan dan keamanan penyimpanan data karena adanya mekanisme enkripsi data sebelum data tersebut disimpan didalam memori sehingga tingkat keamanannya dapat terjamin dan *Sms Gateway* digunakan untuk sistem informasinya yang bertujuan untuk mengirimkan pesan peringatan batas peminjaman buku. *SMS Gateway* sendiri merupakan layanan sms dua arah dimana pengirim pesan dapat menerima pesan balasan dari penerima pesan. Sehingga penerima dapat memperpanjang masa peminjaman buku secara langsung tanpa perlu harus ke perpustakaan. Disini penulis menggunakan dua teknologi

sekaligus karena dalam sistem ini diharapkan mampu menjadi suatu aplikasi yang memberikan kemudahan layanan yang efisien dan tepat guna.

1.2 Rumusan Masalah

Dari latar belakang masalah yang telah penulis paparkan, maka penulis mencoba untuk mengembangkan sebuah sistem peminjaman buku perpustakaan. Adapun beberapa permasalahan yang ada yaitu :

1. Bagaimana membuat sistem aplikasi peminjaman buku perpustakaan menggunakan teknologi *Smart Card* dan *SMS Gateway*.
2. Bagaimana kinerja sistem aplikasi tersebut yang bertujuan untuk memberikan kemudahan layanan yang efisien dan tepat guna.

1.3 Batasan Masalah

Untuk mencapai tujuan penyelesaian tugas akhir ini secara maksimal, maka diperlukan batasan masalah yang diharapkan agar permasalahan tidak meluas dan tetap fokus pada tujuan utama. Adapun batasan-batasan masalah pada skripsi ini yaitu :

1. Aplikasi menitik beratkan dalam transaksi peminjaman buku pada perpustakaan.
2. Transaksi dapat dilakukan sebatas identitas dan nomor handphone yang terdaftar.
3. Tidak membahas database buku pada perpustakaan.
4. Jumlah buku yang dapat dipinjam maksimal dua buku.
5. Perpanjangan peminjaman buku melalui sms hanya dapat dilakukan sekali.
6. Ruang lingkup yang digunakan dalam sistem ini yaitu Teknik Elektro ITN Malang.

1.4 Tujuan dan Manfaat Penelitian

1.4.1 Tujuan

Adapun tujuan penulisan skripsi ini adalah dapat membuat suatu sistem aplikasi peminjaman buku perpustakaan menggunakan teknologi *Smart Card* dan dapat merancang sistem untuk mengirimkan informasi sebagai pesan peringatan batas peminjaman buku dengan menggunakan teknologi *SMS Gateway*

1.4.2 Manfaat

Adapun manfaat penulisan skripsi ini adalah :

1. *Smart Card* ini akan menggantikan formulir peminjaman buku perpustakaan.
2. Mempermudah seorang mahasiswa untuk mengembalikan buku tepat waktu tanpa harus terkena denda.
3. Mahasiswa dapat memperpanjang masa peminjaman buku dengan memanfaatkan fasilitas SMS.
4. Dengan sistem peminjaman buku dengan penerapan teknologi terbaru yaitu *Smart Card* dan *SMS Gateway* diharapkan mampu memberikan kemudahan layanan yang cepat, efisien dan tepat guna.

1.5 Metodologi Penelitian

Adapun metode penelitian yang digunakan dalam pembuatan perangkat lunak ini adalah sebagai berikut:

1. Studi literatur

Pengumpulan data yang dilakukan dengan mencari bahan-bahan kepustakaan, referensi, dan dari survei lapangan dari berbagai sumber sebagai landasan teori yang ada hubungannya dengan permasalahan yang dijadikan objek penelitian.

2. Analisa Kebutuhan Aplikasi

Data dan informasi yang telah diperoleh akan dianalisa agar dihasilkan kerangka global yang bertujuan untuk mendefinisikan kebutuhan sistem dimana nantinya akan digunakan sebagai acuan perancangan sistem.

3. Perancangan dan Implementasi

Berdasarkan data dan informasi yang telah diperoleh serta analisa kebutuhan untuk membangun sistem ini, akan dibuat rancangan kerangka global yang menggambarkan mekanisme dari sistem yang akan dibuat dan diimplementasikan ke dalam sistem.

4. Eksperimen dan Evaluasi

Pada tahap ini, sistem yang telah selesai dibuat akan diuji coba, yaitu pengujian berdasarkan fungsionalitas program, dan akan dilakukan koreksi dan penyempurnaan program jika diperlukan.

5. Penulisan laporan tugas akhir.

1.6 Sistematika Penulisan

Untuk mempermudah dan memahami pembahasan penulisan skripsi ini, maka sistematika penulisan disusun sebagai berikut :

BAB I : PENDAHULUAN

Bab ini merupakan bagian pendahuluan dimana akan tercakup secara umum mengenai latar belakang penulisan laporan, ruang lingkup karya tulis skripsi ini, tujuan dan manfaat yang mau dicapai, metodologi yang dipakai dalam penyusunan laporan dan sistematika penulisan yang digunakan.

BAB II : LANDASAN TEORI

Bab ini berisi tentang teori - teori yang mendukung dan berhubungan dengan judul penulisan skripsi

BAB III : ANALISA DAN PERANCANGAN SISTEM

Bab ini berisi mengenai analisa kebutuhan sistem baik software maupun hardware yang di perlukan untuk membuat kerangka global yang menggambarkan mekanisme dari sistem yang akan di buat.

BAB IV : IMPLEMENTASI DAN PENGUJIAN SISTEM

Bab ini berisi tentang implementasi dari perancangan sistem yang telah dibuat serta pengujian terhadap sistem tersebut.

BAB V : PENUTUP

Merupakan bab terakhir yang memuat intisari dari hasil pembahasan yang berisikan kesimpulan dan saran yang dapat digunakan sebagai pertimbangan untuk pengembangan penulisan selanjutnya.

BAB II LANDASAN TEORI

Dalam pembuatan aplikasi sistem peminjaman buku perpustakaan menggunakan teknologi *Smart card* dan *SMS Gateway* ini, mengacu pada beberapa dasar teori yang mendukung sistem kerja dari aplikasi. Adapun dasar teori yang didalam perancangan aplikasi ini adalah sebagai berikut.

2.1 Perpustakaan

2.1.1 Pengertian Perpustakaan

Perpustakaan diartikan sebuah ruangan atau gedung, yang digunakan untuk menyimpan buku dan terbitan lainnya, juga sebagai pusat sumber informasi yang diatur menurut sistem dan aturan yang baku, dan didayagunakan untuk keperluan pendidikan dan penelitian.^[11]

Perpustakaan adalah suatu ruangan, bagian dari gedung/bangunan, atau gedung itu sendiri, yang berisi buku-buku koleksi, yang disusun dan diatur sedemikian rupa sehingga mudah dicari dan dipergunakan apabila sewaktu-waktu diperlukan untuk pembaca.^[13]

Perpustakaan merupakan salah satu sarana pembelajaran yang dapat menjadi sebuah kekuatan untuk mencerdaskan bangsa. Perpustakaan mempunyai peranan penting sebagai jembatan menuju penguasaan ilmu pengetahuan yang sekaligus menjadi tempat rekreasi yang menyenangkan dan menyegarkan. Perpustakaan memberi kontribusi penting bagi terbukanya informasi tentang ilmu pengetahuan^[2]

Untuk mewujudkan kondisi perpustakaan sesuai dengan fungsi dan peranannya maka perpustakaan harus dirubah sistem operasionalnya dari perpustakaan manual atau tradisional menjadi perpustakaan yang berbasis pada teknologi informasi dan komunikasi Dengan menerapkan teknologi informasi dan komunikasi diharapkan setiap perpustakaan secara bertahap dapat mengejar ketinggalannya dari perpustakaan-perpustakaan yang lebih maju dan lebih modern serta dapat mengoptimalkan fungsi perpustakaan bagi masyarakat.

2.1.2 Tujuan Perpustakaan

Perpustakaan secara umum bertujuan untuk menciptakan masyarakat terpelajar dan terdidik, terbiasa membaca, berbudaya tinggi serta mendorong terciptanya pendidikan sepanjang hayat, dan juga melakukan layanan informasi kepada masyarakat.

Tujuan penyelenggaraan perpustakaan adalah:^[7]

1. Memenuhi keperluan informasi masyarakat perguruan tinggi, lazimnya staf pengajar dan mahasiswa sering pula mencakup tenaga administrasi perguruan tinggi.
2. Menyediakan bahan pustaka rujukan (referensi) pada semua tingkat akademis, artinya mulai dari mahasiswa tahun pertama hingga mahasiswa program pascasarjana dan pengajar.
3. Menyediakan ruang belajar untuk pemakai perpustakaan.
4. Menyediakan jasa peminjaman yang tepat guna bagi berbagai jenis pemakai.
5. Menyediakan jenis informasi aktif yang tidak hanya terbatas pada lingkungan perguruan tetapi juga lembaga induknya.

Berdasarkan uraian di atas, dapat dilihat bahwa tujuan dari perpustakaan tidak lain sebagai penyedia jasa pelayanan informasi meliputi pengumpulan, pelestarian, pengolahan sehingga dapat dimanfaatkan pengguna sebagai wujud dukungan, melancar serta mempertinggi kualitas pelaksanaan program kegiatan perguruan tinggi.

2.1.3 Fungsi Perpustakaan

Selain mempunyai tujuan yang jelas sebuah perpustakaan juga harus memiliki fungsi tentunya. Begitu juga halnya dengan perpustakaan perguruan tinggi. Berdasarkan Buku Pedoman Perpustakaan Perguruan Tinggi, fungsi perpustakaan perguruan tinggi dapat ditinjau dari berbagai segi yaitu: ^[10]

1. Fungsi Edukasi

Perpustakaan merupakan sumber belajar bagi sivitas akademika, oleh karena itu koleksi yang mendukung pencapaian tujuan pembelajaran, pengorganisasian bahan pembelajaran setiap program studi, koleksi tentang strategi belajar mengajar dan materi pendukung evaluasi pembelajaran

2. Fungsi Informasi

Perpustakaan merupakan sumber informasi yang mudah diakses oleh pencari dan pengguna informasi.

3. Fungsi Riset

Perpustakaan merupakan fungsi bahan-bahan riset dan sekunder yang paling mutakhir sebagai bahan untuk melakukan penelitian dan pengkajian ilmu pengetahuan teknologi dan seri koleksi pendukung penelitian di perpustakaan perguruan tinggi mutlak dimiliki, karena tugas perguruan tinggi adalah menghasilkan karya-karya penelitian yang dapat diaplikasikan untuk kepentingan pembangunan masyarakat dalam berbagai bidang.

4. Fungsi Rekreasi

Perpustakaan harus menyediakan koleksi rekreatif yang bermakna untuk membangun dan mengembangkan kreatifitas, minat dan daya inovasi pengguna perpustakaan.

5. Fungsi Publikasi

Perpustakaan selayaknya juga membantu melakukan publikasi karya yang dihasilkan oleh karya perguruan tingginya sivitas akademik dan non akademik.

6. Fungsi Deposit

Perpustakaan menjadi pusat deposit untuk seluruh karya dan pengetahuan.

7. Fungsi Interpretasi

Perpustakaan sudah seharusnya melakukan kajian dan memberikan nilai tambah terhadap sumber-sumber informasi yang dimilikinya untuk membantu pengguna dalam melakukan Tri dharmanya.

2.1.4 Layanan Perpustakaan

Beberapa Jenis Layanan Perpustakaan secara umum adalah sebagai berikut.^[9]

1. Layanan peminjaman bahan pustaka (layanan sirkulasi).

Layanan sirkulasi atau layanan peminjaman dan pengembalian bahan pustaka adalah satu kegiatan di perpustakaan yang melayani peminjaman dan pengembalian buku. Kegiatan sirkulasi dapat dilaksanakan sesudah buku-buku selesai diproses dengan lengkap dengan label-labelnya seperti kartu buku, kartu tanggal kembali, dan *call number* pada punggung buku.

2. Layanan referensi

Koleksi ini tidak boleh dibawa pulang oleh pengunjung perpustakaan dan hanya untuk dibaca ditempat.

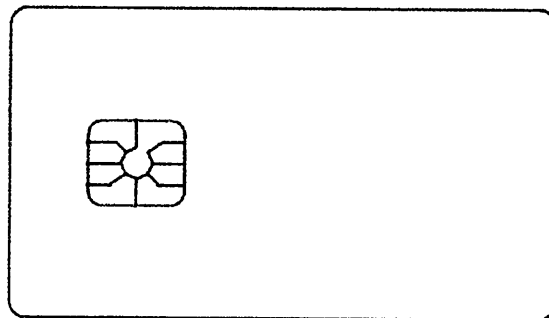
3. Layanan ruang baca

Layanan ruang baca adalah layanan yang diberikan oleh perpustakaan berupa tempat layanan untuk melakukan kegiatan membaca di perpustakaan.

2.2 Smart Card

2.2.1 Pengertian *Smart card*

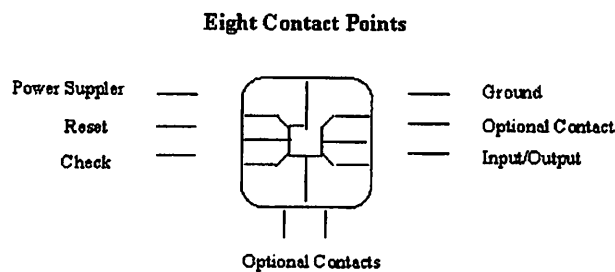
Smart card atau sering juga disebut *ICC (Integrated Circuit Card)* adalah kartu plastik yang berukuran sama dengan kartu kredit yang di dalamnya terdapat *chip* silikon yang disebut *microcontroller*. *Chip* merupakan rangkaian terintegrasi (*integrated circuit*) yang terdiri dari prosesor dan memori. *Chip*, seperti layaknya CPU (*Central Processing Unit*) di komputer, bertugas melaksanakan perintah dan menyediakan *power* ke *smart card*.^[8]



Gambar 2.1 Bentuk *Smart Card*

Smart card berbeda dengan *Magnetic Card* yang juga cukup banyak digunakan dalam aplikasi kartu absensi di kantor-kantor dan lain sebagainya, namun memiliki beberapa kelebihan dalam hal kehandalan, kemampuan menyimpan informasi yang ratusan kali lebih banyak, lebih sulit untuk dipalsukan. *Smart card* cukup mudah untuk diprogram, sehingga memungkinkan untuk dikembangkan lebih lanjut di sisi aplikasinya

Pada *smart card* terdapat wilayah kontak sebanyak 8 titik yang berfungsi untuk mengalirkan tegangan dan juga sebagai antarmuka masukan dan keluaran (I/O – *input output*). *Smart card* yang saat ini banyak beredar dan digunakan untuk baik kebutuhan perkantoran, bisnis maupun industri dibuat berdasarkan standar ISO/IEC 7816 ^[17]



Gambar 2.2 Delapan titik kontak pada smart card

2.2.2 Jenis Memori pada *Smart Card*

Secara umum ada 3 jenis memori yang digunakan, yaitu^[8]:

1. ROM (*Read Only Memory*), berfungsi untuk menyimpan program utama dan sifatnya permanen.
2. RAM (*Random Access Memory*), berfungsi untuk menyimpan data sementara ketika proses sedang berjalan atau hasil penghitungan selama mengeksekusi perintah.
3. EEPROM (*Electrically Erasable Programmable Read Only Memory*), berfungsi untuk menyimpan program dan data yang sewaktu-waktu bisa diubah. Seperti halnya *hard disk* pada komputer.

2.2.3 Tipe-Tipe *Smart Card*

Secara komersial, industri membuat *smart card* dalam beberapa tipe, yaitu^[8]:

1. ***Memory card***. *Smart card* tipe ini tidak mempunyai *processor* atau sistem keamanan yang canggih melainkan hanya perlindungan fisik (karena *smart card* bersifat *tamper proof*). *Smart card* ini merupakan tipe pertama yang dikenal orang dan digunakan pertama kali untuk kartu telepon. Tipe kartu ini menyimpan data yang telah di-*preload* oleh manufakturnya, kemudian mesin pembaca akan mengurangi isi variabel yang disimpannya
2. ***Memory protected cards***. *Smart card* tipe ini mempunyai sistem keamanan yang lebih canggih daripada *memory cards*, misalnya mekanisme *password* untuk mengakses *smart card*.
3. ***Microprocessor cards***. *Smart card* tipe ini mempunyai *processor* sehingga dapat melakukan komputasi walaupun terbatas. Keterbatasannya ada pada ukuran ROM yang dimiliki dan fungsi aritmatika yang masih sederhana. Kemampuannya antara lain mengorganisasikan berkas (*file*) yang dilindungi dengan *password*.
4. ***Java cards***. *Smart card* ini dilengkapi dengan *Java Virtual Machine* sedemikian hingga dapat dimasukkan berbagai program ke dalamnya.
5. ***Public key cards***. *Smart card* ini mendukung *public key cryptography* (kriptografi asimetris) sehingga proses enkripsi/dekripsi dapat dilakukan secara internal dan dapat menyimpan *key*.

2.2.4 Keuntungan menggunakan *Smart Card*

Keuntungan menggunakan kartu *smart card* sebagai berikut^[4]:

1. Lebih handal daripada *kartu magnetik* (kartu magnetik)

Kehandalan dari *smart card* disebabkan oleh proteksi terhadap keamanan data yang disimpan. Keamanannya tidak hanya tergantung pada chip, namun juga keseluruhan sistem termasuk aplikasi serta proses pembuatan dari *smart card* itu sendiri. Chip menjamin keamanan data yang disimpan di dalam *smart card* disebabkan adanya mekanisme enkripsi sehingga tidak mudah dibaca oleh pihak yang tidak berwenang.

2. Lebih banyak menyimpan informasi daripada *kartu magnetik*.

Kapasitas memori dari *smart card* lebih besar dibanding *kartu magnetik*. *Kartu magnetic* hanya memiliki memori sebesar 140 byte yang hanya cukup untuk menyimpan kode PIN dan data untuk *login* ke dalam *server-based system*. Oleh karena itu, transaksi lebih banyak dilakukan secara *on-line*. Sedangkan *smart card* mempunyai ukuran memory bermacam-macam, misalnya dari 1 Kbyte (CP1 dari ASE (Alladin Smartcard Environment)), 2 Kbyte (CC1 dari ASE(Alladin Smartcard Environment)), 22 Kbyte (JavaCard) dan 31 Kbyte(MSC0402 dari Motorola). Selain berisi informasi, *smart card* juga berisi sistem operasi yang mengendalikan seluruh proses yang terjadi di *smart card*.

3. Lebih sulit untuk ditiru daripada *kartu magnetik*

Kartu magnetik mempunyai pita magnetik pada permukaannya. Peng-*copy-an* terhadap *kartu magnetik* dilakukan dengan meng-*copy* pita magnetik tersebut ke *kartu* lain. Pada *smart card* peng-*copy-an* terhadap *kartu* sulit dilakukan, ini disebabkan karena setiap *kartu* memiliki nomor seri yang unik, tidak ada 2 buah *kartu* yang memiliki nomor seri yang sama

4. Dapat digunakan kembali

Setelah nilai yang tertulis di dalam *smart card*, misalnya jumlah pulsa/uang habis, *smart card* dapat di'isi' ulang dengan menuliskan nilai tertentu ke dalamnya. Ini bisa dilakukan selama kondisi *smart card* masih baik, misalnya tidak terdapat kerusakan pada chip. Berbeda dengan *kartu magnetik*, setelah nilai yang ada di dalamnya habis, maka *kartu* tersebut tidak dapat digunakan kembali.

5. Dapat melakukan banyak fungsi di berbagai area industry

Keistimewaan yang dimiliki oleh *smart card*, yaitu dalam hal kapasitas simpan dan kemampuan untuk melakukan proses, *smart card* menawarkan skema *multi-function*, yaitu satu *kartu* untuk berbagai layanan. *Smart card* banyak dimanfaatkan misalnya di sektor telekomunikasi, misalnya SIM card pada layanan GSM. SIM selain sebagai *kartu telepon* dengan sistem *Pre-paid*

juga akan dikembangkan layanan untuk kredit, jadi semacam ATM pribadi. Di samping itu *smart card* telah dimanfaatkan di sektor lain, seperti sektor keuangan, transportasi, dan kesehatan.

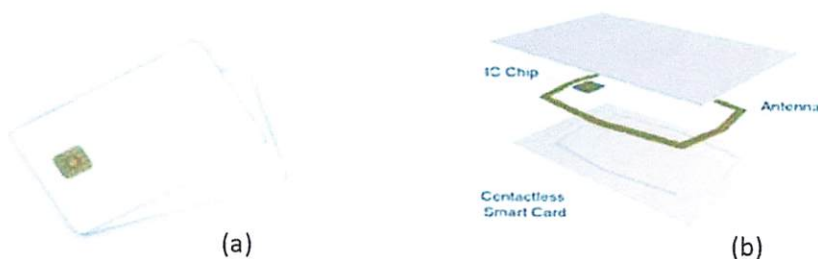
6. Selalu mengalami evolusi (sesuai dengan perkembangan chip komputer dan memori).

Smart card mempunyai standar mikroprosesor 8-bit, namun saat ini mulai dikembangkan mikroprosesor 32-bit yang mempunyai keuntungan, yaitu memungkinkan melakukan pemrograman dengan menggunakan bahasa tingkat tinggi dan meningkatkan kekuatan komputasi untuk fungsi matematika yang kompleks yang tidak mungkin dilakukan pada mikroprosesor 8-bit. Peningkatan kekuatan komputasi ini akan mempercepat jalannya program dan waktu transaksi.

2.2.5 Komunikasi Smart Card

Smart card berinteraksi dengan dunia luar dengan menggunakan antarmuka komunikasi serial melalui 8 titik kontak. Ukuran dan letak dari kontak tersebut didefinisikan di dalam ISO7816, bagian kedua. Aplikasi komunikasi dengan *reader* (yang kemudian akan berkomunikasi dengan *smart card*) menggunakan protokol yang standar, yaitu protocol *International Standard Organization (ISO) 7816*.^[8]

Smart card dapat berkomunikasi dengan *reader* dengan 2 cara, yaitu dengan **kartu contact** dan **kartu contactless**.^[5]



Gambar 2.3 Perbedaan antara kartu *contact* (a) dan kartu *contactless* (b)

- **Kartu *contact*** harus dimasukkan ke dalam *Card Acceptance Device* (CAD), lalu berkomunikasi dengan dunia luar melalui antarmuka komunikasi serial lewat 8 titik kontak (*contact points*).
- **Kartu *contactless*** tidak perlu dimasukkan ke dalam CAD. Kartu ini berkomunikasi dengan dunia luar melalui antena yang terselubung di dalam kartu. Daya disediakan oleh baterai internal atau dapat juga dikumpulkan oleh antena. Kartu *contactless* mengirimkan data ke CAD melalui gelombang elektromagnetik.

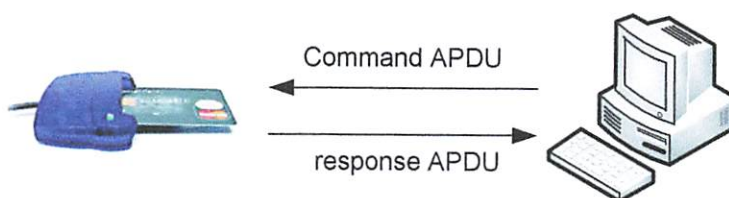
Karena mikrosirkuit pada kartu *contactless* disegel di dalam kartu, kartu *contactless* dapat mengatasi keterbatasan kartu *contact*, yaitu tidak ada *contact* yang harus rusak karena seringnya penggunaan kartu, kartu tidak perlu dimasukkan dengan hati-hati ke dalam CAD, dan kartu tidak harus sesuai standar agar dapat masuk ke dalam slot CAD.

Namun, kartu *contactless* punya kekurangan. Kartu *contactless* harus dalam jangkauan yang sudah ditentukan untuk pertukaran data dengan CAD. Karena kartu bisa keluar dengan cepat dari jangkauannya, maka durasi transaksi menjadi pendek, sehingga data yang dapat dikirimkan terbatas. Ada kemungkinan suatu transaksi dapat mengambil alih kartu atau data yang dikirim disadap tanpa disadari pemegang kartu. Saat ini, kartu *contactless* lebih mahal daripada kartu *contact*. Tugas akhir ini menggunakan kartu *contact*, karena lebih umum digunakan.

2.2.5.1 Model Komunikasi Smart Card

Komunikasi antara kartu dan *host* melewati jalan *half-duplexed*, data dapat dikirim dari *host* ke kartu atau dari kartu ke *host*, akan tetapi tidak bisa kedua-duanya sekaligus.^[5]

Ketika dua komputer berkomunikasi satu sama lain, mereka menukarkan paket data yang dibangun mengikuti protokol, misalnya protokol TCP/IP. Begitu juga, *smart card* berkomunikasi dengan komputer dengan menggunakan data pakatnya, yang disebut *Application Protocol Data Units* (APDUs). Sebuah APDU berisi pesan perintah saja atau pesan respon saja.^[5]



Gambar 2.4 Komunikasi Smart Card

Dalam dunia kartu, model *master-slave* (tuan-budak) diterapkan. *Smart card* selalu memainkan peran pasif (*slave/budak*), menunggu perintah APDU dari *host*. *Smart card* kemudian mengeksekusi instruksi yang dispesifikkan dalam perintah dan menjawab ke *host* dengan APDU respon. APDU perintah dan APDU respon dipertukarkan secara bergantian antara kartu dan *host*.

2.2.5.2 Format APDU

APDU merupakan unit dasar untuk pertukaran paket di dalam *smartcard*. Komunikasi antara kartu dengan *reader* dilakukan dengan APDU[ISO7816-95]. APDU dinyatakan sebagai data paket yang berisi perintah lengkap atau respon yang lengkap dari kartu. Untuk menyediakan fungsionalitas seperti ini, APDU mendefinisikan struktur yang didefinisikan dalam beberapa dokumen ISO 7816.^[5]

Pesan APDU menurut ISO 7816 terdiri dari dua struktur, yaitu satu digunakan oleh aplikasi *host* pada sisi CAD untuk mengirim perintah ke kartu, yang lain digunakan oleh kartu untuk mengirim respon kembali ke aplikasi *host*. Yang pertama disebut *Command APDU* (C-APDU) dan yang kedua disebut *response APDU* (R-APDU). *Command APDU* selalu dipasangkan dengan *response APDU*.

2.3 Smart Card Reader

Smart card reader atau biasa disebut *reader* adalah antarmuka yang menghubungkan komunikasi antara *smart card* dengan PC. Istilah lain yang digunakan untuk *smart card reader* adalah terminal, *CAD* (*Card Acceptance Device*) dan IFD (*interface device/perangkat antarmuka*). *Smart card* tidak berarti tanpa adanya *smart card reader*, yang berfungsi sebagai perantara komunikasi antara *smart card* dengan peralatan

lain seperti komputer. Komputer membaca atau menulis data melalui *smart card reader*, kemudian *smart card reader* mengubah perintah membaca/menulis tersebut ke dalam bahasa yang dimengerti *smart card*. *reader* membangun hubungan dengan mikroprosesor *smart card* melalui kontak elektrik pada permukaan *smart card*. Melalui hubungan elektrik ini, *reader* menyediakan *power* ke *smart card* dan membangun hubungan pertukaran data. *Reader* dapat berhubungan dengan PC biasanya melalui *port keyboard*, *port serial* atau *port PCMCIA*. Namun pada saat ini kebanyakan *reader* berhubungan dengan PC melalui *port USB (Universal Serial Bus)*.^[14]



Gambar 2.5 *Smart Card Reader* tipe ACR30

2.4 SMS Gateway

SMS Gateway adalah suatu platform yang menyediakan mekanisme untuk menghantar dan menerima SMS dari peralatan mobile (HP, PDA phone, dan lain-lain) yang menggunakan *keyword* tertentu. SMS Gateway adalah komunikasi SMS dua arah. 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 dapat dimanfaatkan untuk keperluan lain seperti melakukan polling, transaksi dengan sebuah sistem, pemantauan, dan sebagainya.^[15]

Gateway adalah aplikasi SMS dimana pesan yang di terima dan dikirimkan menggunakan bantuan Gateway Device terintegrasi dengan database server yang dapat mendistribusikan pesan SMS secara otomatis. SMS Gateway merupakan aplikasi berbasis

komputer, sehingga dapat di otomatisasi. Dapat menyimpan data dalam jumlah yang banyak karena disimpan di sebuah hardisk server.

Sistem ini memiliki kemampuan menerima dan atau menyampaikan informasi lewat SMS dan mengelola informasi tersebut dari dan atau ke dalam bank data (database). Sistem ini juga memiliki kemudahan dalam mengatur aturan proses bisnis yang diinginkan seperti : bagaimana mengelola data SMS yang diterima, melakukan pencarian informasi, dan menyimpan informasi. Pengaturan tersebut dapat dengan mudah dilakukan oleh Administrator lewat alat bantu konfigurasi aplikasi yang telah ada. Sistem ini juga menyediakan beberapa cara penyampaian informasi lewat SMS yang dapat dipilih sesuai kebutuhan.

Beberapa fitur umum penyampaian informasi lewat SMS

a. SMS Interaktif

Pengguna HP meminta atau memberi informasi dengan mengirim SMS ke nomor Modem GSM. Setelah data sms dari Modem GSM diproses oleh PC Server SMS, Modem GSM akan mengirim SMS jawaban berupa informasi yang tepat ke HP Pengguna.

b. SMS Scheduler

PC Server SMS memonitor database dan penjadwalan (schedule) dan secara periodik mengirim informasi lewat SMS ke HP Pengguna yang terdaftar.

c. SMS Alert

PC Server SMS memonitor kejadian-kejadian yang telah ditentukan oleh Administrator, dan mengirim pemberitahuan kejadian lewat SMS ke HP pengguna yang terdaftar

Selain digunakan untuk pengiriman pesan antara mobile phone, teknologi SMS juga memungkinkan pengiriman pesan antara mobile phone dengan PC. Agar hal ini dapat dilakukan maka user memerlukan suatu teknologi khusus yang mampu menjadi perantara antara kedua alat tersebut. Prinsip kerja dari SMS Gateway ini kita dapat melakukan pengaturan SMS dengan PC yang dihubungkan dengan jalur GSM dari mobile phone tersebut dengan teknologi ini kita dapat mengembangkan SMS bukan hanya berkaitan

dengan pengiriman dan penerimaan SMS namun juga bisa diaplikasikan ke beberapa teknologi lainnya.

Ketika sebuah pesan dikirimkan oleh aplikasi komputer, pertama kali pesan tersebut akan dikirimkan kepada modem GSM atau disebut juga SME (Short Message Entity) , dan kemudian modem GSM akan mentransmisikan pesan tersebut kepada SMSC (Short Message Service Center) atau penyedia layanan GSM melalui jalur nirkabel BSS (Base Station System). Di sisi lain, pesan yang diterima oleh modem GSM akan disimpan di dalam memori atau di dalam SIM card. Setelah itu, modem GSM akan memberikan peringatan kepada komputer. Program pada komputer akan menerima peringatan tersebut dan kemudian membaca, memori yang tepat serta menghapus pesan dari phone untuk menyiapkan ruang bagi pesan berikutnya.

2.4.1 Elemen utama arsitektur SMS

2.4.1.1 Short Message Entity (SME)

Short Message Entity (SME) adalah elemen yang dapat mengirim atau menerima pesan singkat. SME dapat berupa software aplikasi pada mobile handset, dapat juga berupa perangkat faksimile, perangkat telex, remote internet server, dll. Sebuah SME dapat berupa server yang terkoneksi dengan SMS centre secara langsung atau melalui gateway. Dikenal juga External SME (ESME) yang merepresentasikan sebuah WAP proxy/server, Email Gateway atau Voice Mail server^[16]

2.4.1.2 SMS Service Centre (SMSC)

SMS Service Centre (SMSC) memegang peran kunci dalam arsitektur SMS. Fungsi utama SMSC adalah menyampaikan pesan singkat antara SME dengan MS, juga menyimpan dan meneruskan pesan singkat (menyimpan pesan jika penerima SME tidak tersedia). SMSC dapat terintegrasi sebagai bagian dari mobile network atau sebagai entitas network independen.^[16]

2.4.1.3 Base Station Subsystem (BSS)

Base Station Subsystem (BSS) atau yang biasa dikenal sebagai radio subsystem adalah penyedia dan pengatur transmisi radio dari system selular. Fungsi utama dari BSS adalah menghubungkan antara MS (*Mobile Station*) dengan NSS (*Network Switching Subsystem*). Interface antara MS dengan subsistem lain dari GSM juga diatur melalui BSS.^[16]

2.4.2 Keuntungan SMS Gateway

SMS Gateway merupakan pintu gerbang bagi penyebaran informasi dengan menggunakan SMS. *SMS Gateway* dapat menyebarkan pesan ke ratusan nomor secara otomatis dan cepat yang langsung terhubung dengan database nomor-nomor ponsel saja tanpa harus mengetik ratusan nomor dan pesan di ponsel karena semua nomor akan diambil secara otomatis dari database tersebut sehingga dapat menghemat waktu. Selain itu, kebutuhan untuk *SMS Gateway* juga tidak terlalu berlebihan karena hanya menggunakan sebuah PC dengan menggunakan sebuah ponsel atau modem, kabel data, kartu GSM, dan sebuah program yang dapat dibangun sendiri sebagai pengirim pesan. *SMS Gateway* dapat mengustomisasi pesan-pesan yang ingin dikirim. Dengan menggunakan program tambahan yang dapat dibuat sendiri, pengirim pesan dapat lebih fleksibel dalam mengirim berita karena biasanya pesan yang ingin dikirim berbeda-beda untuk masing-masing penerimanya.^[15]

2.5 AT Command

AT command adalah perintah yang digunakan dalam komunikasi dengan serial port. Dengan AT Command kita dapat mengetahui vendor dari handphone, mengetahui kekuatan sinyal, membaca pesan SMS, mengirim SMS, menghapus pesan dan masih banyak lagi MSCComm menyediakan komunikasi data melalui serial port (juga mendukung USB dan infrared) dari suatu aplikasi. Sehingga dapat berkomunikasi dengan ponsel atau modem GSM yang terpasang pada serial port computer.^[16]

Bahasa yang dikenali oleh modem GSM disebut dengan *AT command*. AT merupakan kependekan dari Attention. AT command dapat digunakan antara lain untuk menginstruksikan perintah-perintah sabagai berikut:^[16]

1. Mengirim dan menerima pesan SMS, atau fax.
2. Mendapatkan informasi mengenai device, misalnya nama manufaktur, nomor IMEI, dan lain-lain.
3. Mendapatkan status device, misalnya status aktifitas, status registrasi network, kekuatan sinyal, ataupun status baterai.
4. Penulisan dan pencarian phonebook.
5. Dari sisi keamanan, Anda dapat mengaktifkan fasilitas lock dan mengubah password.
6. Menyimpan dan mengembalikan konfigurasi.

Tidak semua device mengimplementasikan seluruh *AT command*, pada umumnya modem GSM lebih mendukung banyak *AT command* dibandingkan ponsel biasa *AT command* disupport dengan menggunakan HyperTerminal yang disediakan oleh Microsoft Windows yang terletak pada menu *All Programs-Accessories, Communications-Hyperterminal*. Sebelum digunakan terlebih dahulu pastikan driver telah terinstal agar system operasi mengenali device yang akan dipakai kemudian masukkan kartu GSM pada modem GSM. Kemudian koneksikan pada port komputer, supaya program *HyperTerminal* dapat dipanggil.^[16]

Untuk dapat melakukan test koneksi dengan modem GSM, cukup ketikkan perintah AT diikuti dengan penekanan tombol enter yang akan direspon dengan OK jika tidak terdapat masalah

Di bawah ini menunjukkan *AT Command* yang menunjukkan bagaimana cara menggunakan perintah *AT HyperTerminal* dan program *Microsoft Windows* untuk melakukan perintah yang harus dimasukkan dalam *HyperTerminal*.

Tabel 2.1 Tabel Control AT Comm

AT Command	Fungsi
ATD	Melakukan dial
ATH	Hang up panggilan yang masuk
ATA	Menjawab panggilan yang masuk

Tabel 2.2 Tabel fungsi AT Comm secara umum

AT Command	Fungsi
AT+CGMI	Mengambil informasi manufaktur
AT+CGMM	Mengambil informasi model
AT+CGMR	Mengambil informasi revisi
AT+CGSN	Mengambil informasi serial number device
AT+CSCS	Memilih set karakter

Tabel 2.3 Tabel AT Command untuk SMS

AT command	Fungsi
AT+CPMS	Menentukan penyimpanan pesan
AT+CMGF	Format pesan
AT+CSCA	Nomor service center
AT+CNMI	Pengaktifkan indikasi pesan baru
AT+CMGL	Daftar pesan
AT+CMGR	Membaca pesan
AT+CMGS	Mengirim pesan
AT+CMGD	Menghapus pesan
AT+CMMS	Mengirimkan lebih banyak pesan
+CMTI	Notifikasi delivery pesan
+CDSI	Notifikasi status report

Untuk AT *command* yang berawalan dengan AT (misalnya: AT+CMGS, AT+CMGD) diketikkan dengan diikuti parameter-parameter yang dibutuhkan, sedangkan command tanpa awalan AT (seperti +CMT, +CDSI) merupakan result yang dikeluarkan oleh device. Jika komputer tidak memiliki dokumentasi yang cukup mengenai AT command yang didukung oleh device yang digunakan AT command dapat dicoba melalui *Hyperteminal*. Disini *hyperteminal* berfungsi mengembalikan pesan kesalahan jika AT command yang diketikkan tidak dimengerti oleh *device*.

Saat sebuah AT *command* tidak menampilkan pesan kesalahan, perlu dicoba lebih jauh dengan mengetikkan parameter-parameter yang mungkin digunakan dalam aplikasi.

Hal ini penting mengingat AT *command* juga bisa berbeda parameter antara device satu dengan device lainnya. Terdapat dua tipe AT *command*, yaitu *basic command* dan *extended command*. *Basic command* adalah AT *command* yang tidak menggunakan tanda

+, misalnya ATD, ATH, dan ATA. Sebaliknya, AT *command* yang menggunakan tanda + merupakan *extended command*. Setiap *extended command* memiliki parameter atau perintah test yaitu =?, yang akan mengembalikan informasi mengenai *command* tersebut.

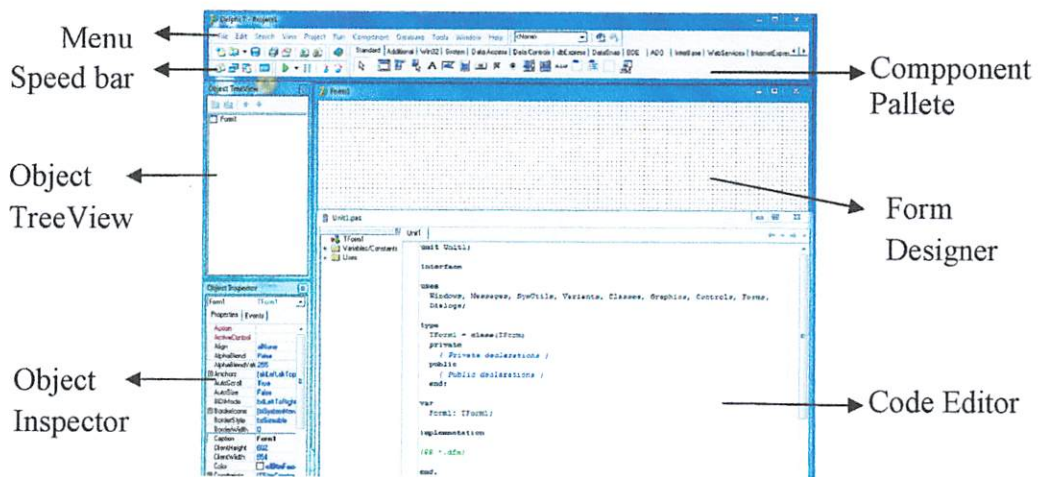
2.6 Borland Delphi

Borland Delphi merupakan suatu bahasa pemrograman yang memberikan berbagai fasilitas pembuatan aplikasi visual. Keunggulan bahasa pemrograman ini terletak pada produktivitas, kualitas, pengembangan perangkat lunak, kecepatan kompilasi, pola design yang menarik serta diperkuat dengan pemrograman yang terstruktur. Keunggulan lain dari Delphi adalah dapat digunakan untuk merancang program aplikasi yang memiliki tampilan seperti program aplikasi lain yang berbasis Windows.^[1]

Khusus untuk pemrograman database, Borland Delphi menyediakan fasilitas objek yang kuat dan lengkap yang memudahkan programmer dalam membuat program. Format database yang dimiliki Delphi adalah format database Paradox, dBase, MS. Access, ODBC, SyBASE, dan lain lain.^[1]

Delphi memiliki tampilan khusus yang didukung suatu lingkup kerja komponen Delphi untuk membangun suatu aplikasi dengan menggunakan Visual Component Library (VCL).

Pada pembuatan Desain sistem aplikasi peminjaman ini menggunakan Borland Delphi 7, untuk melihat bentuk project Borland Delphi 7 bisa dilihat pada gambar dibawah ini



Gambar 2.6 IDE Borland Delphi7

Keterangan pada gambar 2.6 sebagai berikut:

1. **Main Menu** : merupakan penunjuk ke sluruh aktifitas yang disediakan oleh Delphi
2. **Toolbar/Speedbar** : meruapakn icon yang dirancang untuk lebih memudahkan menjangkau fasilitas yang ada pada delphi.
3. **Component Palette** : berisi kumpulan ikon yang melambangkan komponen-komponen yang terdapat pada *VCL (Visual Component Library)*.
4. **Form Designer** : Merupakan suatu objek yang dapat dipakai sebagai tempat untuk merancang program aplikasi. Form berbentuk sebuah meja kerja yang dapat diisi dengan komponen-komponen yang diambil dari Component Palette
5. **Code Editor** : merupakan tempat di mana Anda dapat menuliskan kode program. Pada bagian ini Anda dapat menuliskan pernyataan-pernyataan dalam Object Pascal
6. **Object Inspector** : Digunakan untuk mengubah properti atau karakteristik dari sebuah komponen. Object Inspector terdiri dari dua tab, yaitu **Properties** dan **Events**
7. **Object TreeView** menampilkan diagram pohon dari komponen-komponen yang bersifat visual maupun nonvisual yang telah terdapat dalam form, data module, atau frame. Object TreeView juga menampilkan hubungan logika antar komponen

2.7 Microsoft Access

Microsoft Access adalah sebuah program aplikasi basis data komputer relasional yang ditujukan untuk perusahaan kecil hingga menengah. Aplikasi ini merupakan anggota dari beberapa aplikasi *Microsoft Office*, selain tentunya *Microsoft Word*, *Microsoft Excel*, dan *Microsoft PowerPoint*. Aplikasi ini menggunakan mesin basis data *Microsoft Jet Database Engine*, dan juga menggunakan tampilan grafis yang sudah dikenal pengguna sehingga memudahkan pengguna.^[6]

Microsoft Access dapat menggunakan data yang disimpan di dalam format *Microsoft Access*, *Microsoft Jet Database Engine*, *Microsoft SQL Server*, *Oracle Database*, atau semua kontainer basis data yang mendukung standar ODBC. *Access* juga mendukung teknik-teknik pemrograman berorientasi objek, tetapi tidak dapat digolongkan ke dalam perangkat bantu pemrograman berorientasi objek.^[6]

Kelebihan menggunakan program aplikasi basis data *Microsoft Access* sebagai berikut:

1. Sudah tersedia atau ter-install disetiap komputer auditor
2. Pengguna sudah terlatih untuk menggunakannya dalam menyelesaikan pekerjaannya sehari - hari jadi diperlukan waktu pembelajaran yang lebih sedikit
3. Memiliki fungsi atau kemampuan yang sangat besar karena tidak dirancang hanya untuk audit software, sehingga secara kasar bisa dikatakan bahwa semua fungsi atau kemampuan audit software bisa dikerjakan oleh *Access*. Sehingga kita bisa menambah, merancang sendiri fungsi atau kemampuan tanpa harus menunggu update software.
4. Salah satu keunggulan *Microsoft Access* dilihat dari perspektif programmer adalah kompatibilitasnya dengan bahasa pemrograman *Structured Query Language (SQL)*.

BAB III

ANALISA DAN PERANCANGAN SISTEM

3.1 Analisa Sistem

Analisa ditunjukkan untuk memberikan gambaran secara umum tentang aplikasi dan memberikan solusi terhadap permasalahan yang dihadapi. Dalam analisa masalah dan penyelesaian dari masalah yang dihadapi.

3.1.1 Identifikasi Masalah

Tahap identifikasi masalah merupakan tahapan paling awal untuk melakukan perancangan dan pembuatan aplikasi. Tahapan ini di gunakan untuk melakukan observasi atau penelusuran permasalahan untuk mendapatkan permasalahan umum dari permasalahan yang di hadapi. Di dalam tahapan ini juga di lakukan perumusan permasalahan yaitu merumuskan atau menetapkan permasalahan yang di hadapi, sehingga lebih fokus untuk mencari dan memecahkan permasalahan yang ada.

Pada skripsi ini permasalahan yang dihadapi ada pada sistem peminjaman buku perpustakaan, yaitu masih menggunakan sistem peminjaman manual, dari sistem peminjaman tersebut ada beberapa kelemahan yaitu mahasiswa atau peminjam harus mengisi formulir peminjaman buku, dimana sistem tersebut tidak efisien bagi mahasiswa dan admin, serta seorang mahasiswa sering melupakan batas peminjaman buku tersebut dari kelemahan tersebut maka timbul pemikiran untuk membuat sistem peminjaman buku yang diharapkan bisa mengatasi kelemahan tersebut. Sistem peminjaman buku perpustakaan yang akan dibuat yaitu dengan memanfaatkan teknologi terbaru yaitu menggunakan teknologi *Smart Card* dan *SMS Gateway*.

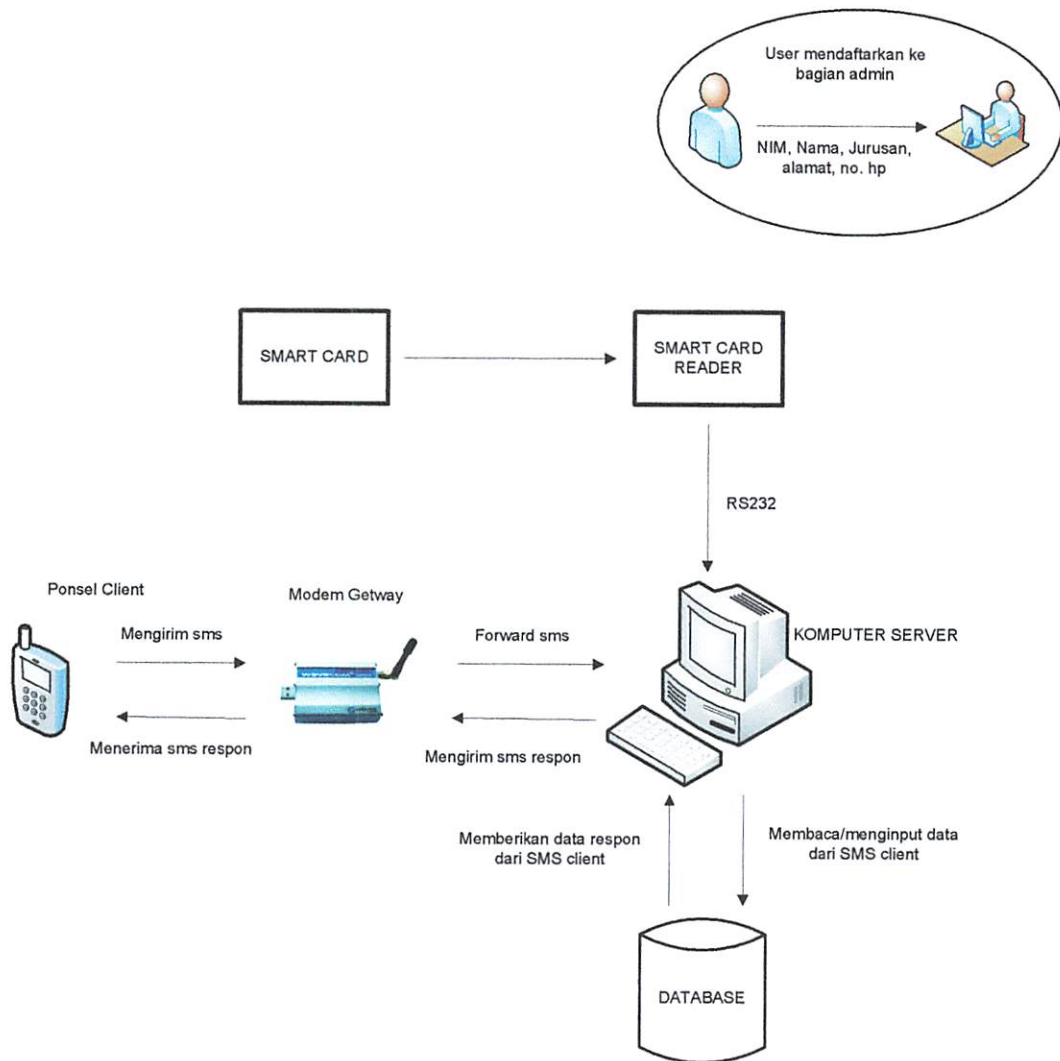
3.2 Gambaran Umum Sistem

Sistem peminjaman buku perpustakaan adalah sebuah aplikasi yang digunakan untuk melakukan transaksi peminjaman buku perpustakaan. Dalam sistem ini akan menggunakan kartu *smart card* yang akan diisi data identitas mahasiswa. Sehingga nantinya kartu *smart card* ini akan menggantikan formulir peminjaman buku perpustakaan yang sah, dan sistem ini juga menggunakan teknologi informasi yaitu *SMS Gateway* yang

mana berfungsi sebagai informasi peringatan batas peminjaman buku dan digunakan sebagai media untuk memperpanjang batas peminjaman buku. Dengan tujuan untuk memberikan kemudahan layanan yang efisien, cepat, dan tepat guna.

Secara lebih rincinya akan dijelaskan sebagai berikut. Mahasiswa sebelumnya harus mendaftarkan diri terlebih dahulu ke bagian admin, identitas mahasiswa tersebut akan tersimpan pada kartu *smart card* dan sistem aplikasi, yang mana nantinya kartu tersebut digunakan untuk transaksi peminjaman buku. Jika mahasiswa ingin melakukan peminjaman buku, Mahasiswa hanya memasukkan kartu *smart card* di depan *smart card reader*, selanjutnya *smart card* akan langsung dibaca oleh *card reader*. Setelah *smart card* dibaca oleh *card reader*, informasi mengenai identitas yang tersimpan dalam *smart card* akan segera ditampilkan pada sistem aplikasi. Selanjutnya seorang admin memasukkan nomor register buku yang akan di pinjam oleh mahasiswa dan memasukkan batas pengembalian buku. Setelah selesai, data akan tersimpan pada database sistem dan akan tersimpan juga pada kartu *smart card* mahasiswa sebagai bukti peminjaman yang sah. Selanjutnya sistem akan mengirimkan pesan secara otomatis kepada mahasiswa sebagai pemberitahuan batas peminjaman buku, sistem akan mengirim pesan sehari sebelum tanggal jatuh tempo peminjaman buku. Disamping itu sistem ini dapat menerima pesan dari para mahasiswa untuk melakukan perpanjangan peminjaman buku, hal ini dapat memudahkan seorang anggota untuk langsung memperpanjang masa pinjaman buku tanpa harus datang langsung ke ruang perpustakaan.

Berikut desain gambaran umum sistem peminjaman buku menggunakan teknologi *Smart Card* dan *SMS Gateway* yang ditunjukkan pada gambar 3.1 berikut ini.



Gambar 3.1 Diagram *Desain Sistem*

Dari gambar 3.1 dapat dijelaskan secara umum dan fungsi dari masing-masing blok sebagai berikut :

1. **Smart card** berfungsi sebagai kartu anggota yang digunakan untuk menyimpan identitas mahasiswa dan untuk menyimpan data pinjaman buku sebagai bukti peminjaman yang sah
2. **Smart card reader** atau biasa disebut *reader* adalah antarmuka yang menghubungkan komunikasi antara *smart card* dengan PC.
3. **Komputer Server** berfungsi sebagai pengolah data atau sistem yang melakukan proses sesuai dengan request yang dilakukan.

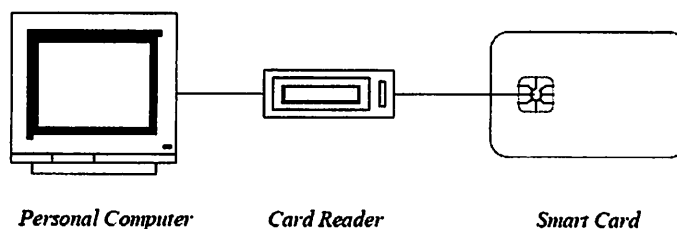
4. **Modem Gateway** berfungsi sebagai penghubung antara komputer server dan ponsel client. Modem gateway melakukan penerimaan pesan dari client dan menginput pesan tersebut kedalam komputer server untuk diproses dan kemudian melakukan pengiriman pesan sesuai dengan pesan yang diperintahkan oleh komputer server ke ponsel client.
5. **Ponsel Client** menerima pesan yang berisi informasi yang dikirim oleh sistem.
6. **Database** *database* ini merupakan media yang menyimpan seluruh data, baik untuk dikirim ke modem Gateway ataupun untuk keperluan aplikasi. Sistem ini menggunakan *Microsoft Access* sebagai *database* sistemnya.

3.3 Perancangan Sistem

3.3.1 Perancangan Sistem Smart Card

Rancangan sistem *smart card* terbagi atas dua bagian besar yaitu pengiriman data antara aplikasi dan *smart card* dan pembacaan data dari *smart card* ke aplikasi. Bagian ini menguraikan bagaimana data identitas mahasiswa, dan identitas buku disimpan dalam database dan pertukaran pesan yang terjadi antara aplikasi dan *smart card*.

Pada sistem *smart card* tersusun atas kartu dan *smart card reader* yang bertujuan untuk merubah mengubah perintah membaca atau menulis tersebut ke dalam bahasa yang dimengerti *smartcard*.



Gambar 3.2 Perancangan *Smart Card*

Penjelasan dari masing-masing blok pada gambar 3.2 adalah sebagai berikut:

1. *Personal Computer* (PC)

PC adalah pusat pengontrol pada aplikasi ini meliputi sistem registrasi, data buku, data pengguna sistem, data pengguna teregistrasi dan data transaksi.

PC dapat memonitor operasi yang dilakukan oleh sistem, memberi perintah pada *smart card reader* untuk membaca dan menuliskan data pada kartu *smart card*. PC terhubung melalui kabel serial dengan *smart card reader*.

2. *Smart Card Reader*

Smart Card Reader adalah suatu alat untuk pengisian data ke *chip* yang terdapat pada kartu *smart card* sekaligus untuk membaca data yang terdapat pada *chip smart card*.

3. *Smart Card*

Kartu plastik yang berukuran sama dengan kartu kredit yang di dalamnya terdapat *chip* silikon yang disebut *microcontroller*. *Chip* merupakan rangkaian terintegrasi (*integrated circuit*) yang terdiri dari prosesor dan memori.

3.3.1.1 Smart card yang digunakan

Kartu *Smart card* yang digunakan dalam sistem ini menggunakan kartu *smart card* tipe SLE4442, *Smart card* SLE4442 merupakan *smart card* jenis *memory protected card* di mana *smart card* jenis ini mempunyai kode pengaman untuk mengontrol akses penulisan maupun pembacaan ke dalam memori. SLE4442 memiliki 256 byte memori. Kartu kontak juga memiliki EEPROM berfungsi untuk menyimpan program dan data yang sewaktu-waktu bisa diubah. Pada saat *power* aktif, seluruh memori dapat dibaca. Setelah memasukkan kartu ke dalam aplikasi sistem semua proses penulisan maupun penghapusan data pada memori dapat dilakukan. Sedangkan untuk komunikasi dengan *reader* dilakukan melalui kontak langsung dengan permukaan titik kontakanya. Berikut spesifikasi kartu SLE4442

- Produsen : Siemens Semiconductor Group, Jerman
- *Type* : *Memory Protected Card* SLE4442
- Jenis memori : *EEPROM 256-Byte*
- *Protection memory* : 178.55 Kbytes
- *Answer-To-Reset* : ISO 7816-3
- Protokol : *2-Wire*
- *Programming time* : 2,5 ms per byte



Gambar 3.3 Kartu *Smart Card* SLE4442

3.3.1.2 Smart Card Reader yang digunakan

Smart card reader yang digunakan adalah tipe reader ACR30, ACR30 merupakan produk ACS. Berikut spesifikasi Reader :

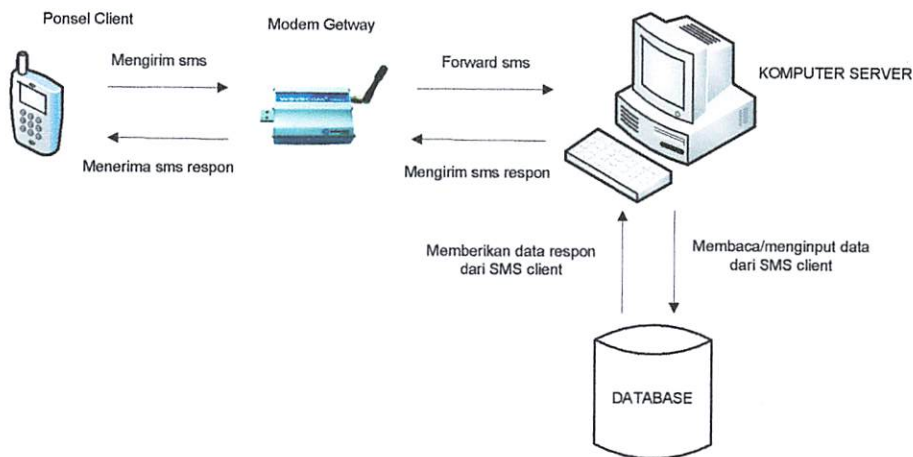
- *Type* : **ACR30 Smart Card Reader RS232**
- Antarmuka : RS232
- Mendukung smart card yang sesuai dengan standar ISO-7816-1/2/3.
- Mendukung memory card dengan fungsi write protect (SLE4432/4442 dan SLE4418/4428).
- Mendukung sistem operasi Windows 98/ME/NT/2K/XP, Linux, dan MAC OSX.



Gambar 3.4 *Smart Card Reader* ACR30

3.3.2 Perancangan Sistem SMS Gateway

Secara umum alat yang digunakan untuk *SMS Gateway* terdiri dari komputer server, modem gateway, dan ponsel client, berikut cara kerja dari sistem *SMS Gateway* ditunjukkan pada gambar 3.5



Gambar 3.5 Cara kerja SMS gateway secara umum

Dari gambar 3.5 dapat dijelaskan secara umum dan fungsi dari masing-masing blok sebagai berikut :

1. Komputer Server

Komputer Server berfungsi sebagai pengolah data atau yang melakukan proses sesuai dengan request yang dilakukan oleh modem gateway. Dimana setiap pesan yang masuk maupun keluar akan disimpan secara otomatis oleh komputer server kedalam sebuah database dan melakukan respon terhadap request perintah yang diterima oleh modem gateway.

2. Modem Gateway

Modem Gateway berfungsi sebagai penghubung antara komputer server dan ponsel client. Modem gateway melakukan penerimaan pesan dari client dan menginput pesan tersebut kedalam komputer server untuk diproses dan kemudian melakukan pengiriman pesan sesuai dengan pesan yang diperintahkan oleh komputer server ke ponsel client.

3. Ponsel Client

Telepon seluler pengguna menerima SMS yang berisi informasi yang dikirim oleh sistem.

Modem yang digunakan pada sistem peminjaman buku ini adalah

- Type : Modem Wavecom M1306B Q2403A USB
- Antarmuka : USB



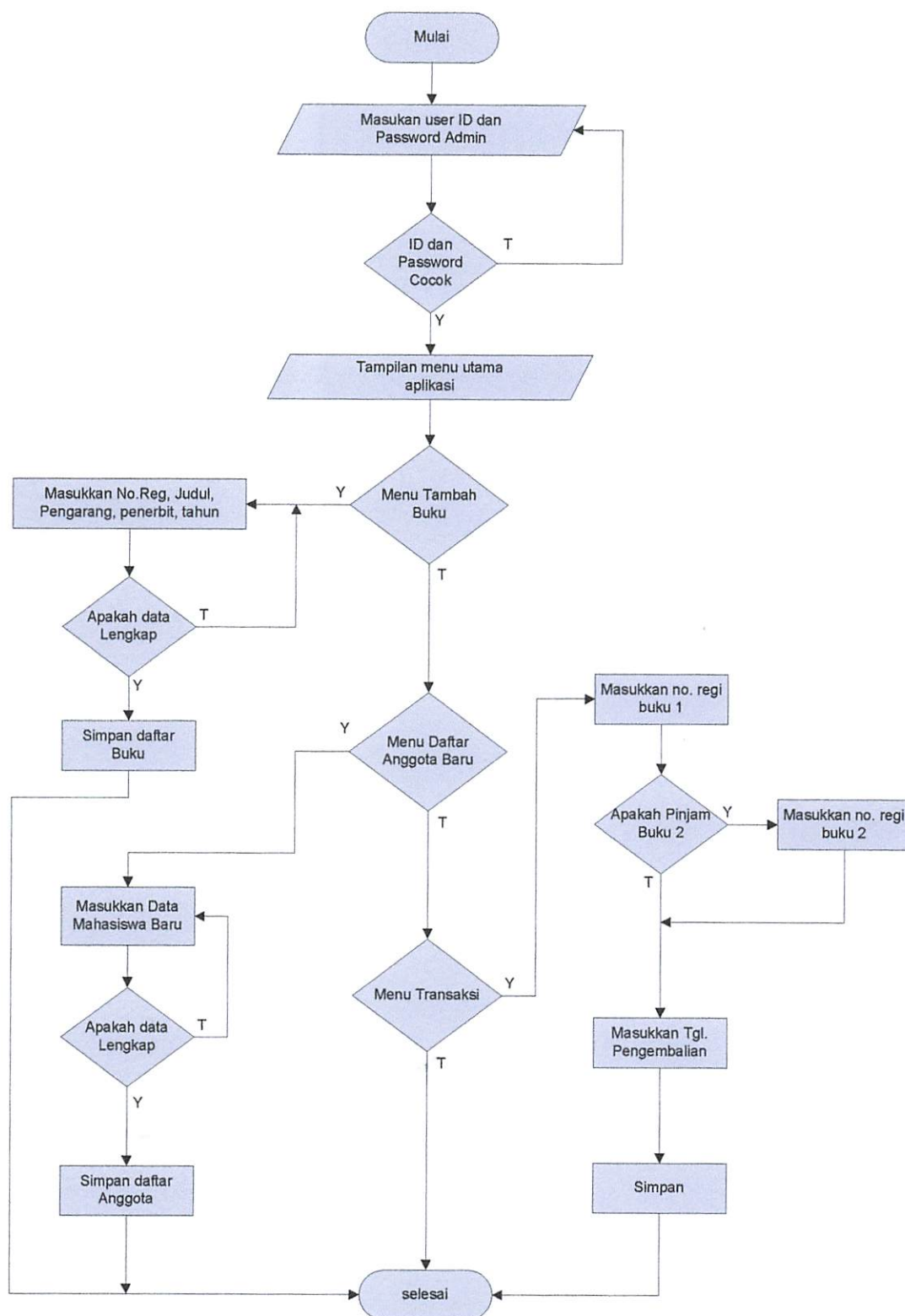
Gambar 3.6 Modem Wavecom M1306B Q2403A USB

3.4 Flowchart Sistem

Untuk sistem aplikasi peminjaman buku perpustakaan ini terdapat beberapa *flowchart*, seperti *flowchart admin*, *flowchart pendaftaran*, *flowchart transaksi peminjaman* dan *flowchart pengembalian buku*. Untuk penjelasannya lebih lanjut dibahas sebagai berikut.

3.4.1 Flowchart Admin

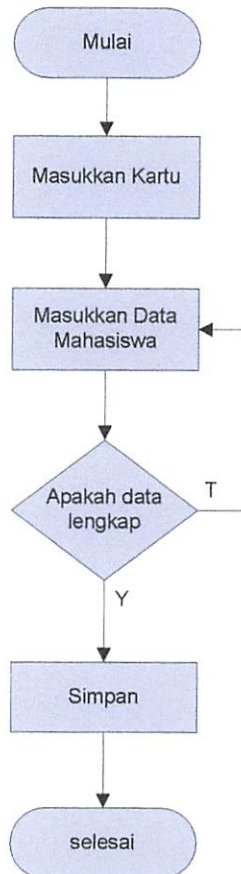
Pada *Flowchart* ini menjelaskan proses-proses yang dilakukan seorang admin pada sistem. Dalam *flowchart* ini dapat dijelaskan bahwa seorang admin dapat mengakses sistem seperti memasukkan data buku baru, mendaftarkan seorang anggota baru, dan melakukan transaksi peminjaman buku.



Gambar 3.7 FlowchartAdmin

3.4.2 Flowchart Pendaftaran

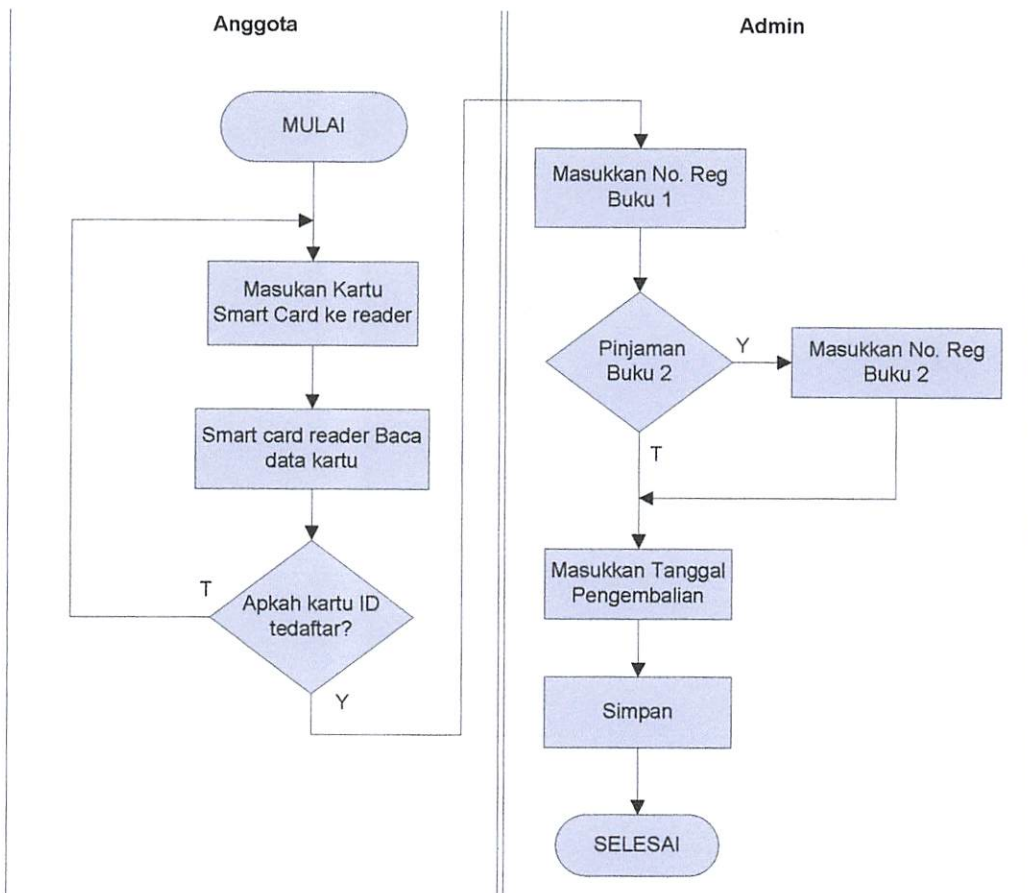
Pada *Flowchart* ini menjelaskan proses-proses yang dilakukan seorang admin untuk mendaftarkan identitas mahasiswa untuk disimpan pada kartu *smart card* dan database sistem.



Gambar 3.8 *Flowchart* Pendaftaran

3.4.3 Flowchart Transaksi Peminjaman

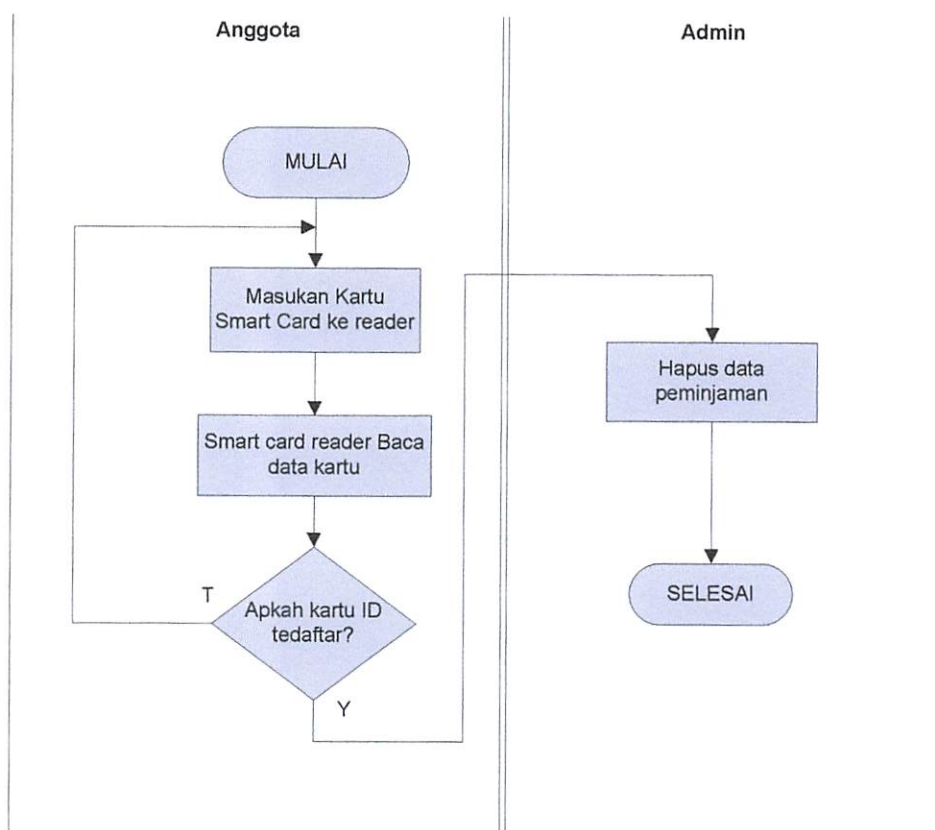
Pada *flowchart* ini menjelaskan proses-proses yang dilakukan untuk melakukan transaksi peminjaman buku dengan menggunakan kartu *smart card*.



Gambar 3.9 *Flowchart* Transaksi Peminjaman

3.4.4 Flowchart Pengembalian Buku

Pada *flowchart* ini menjelaskan proses-proses yang dilakukan untuk melakukan pengembalian buku dengan menggunakan kartu *smart card*.



Gambar 3.10 Flowchart Pengembalian Buku

3.5 Perencanaan dan Pembuatan Perangkat Lunak

Disini akan dibahas perancangan sistem aplikasi yang akan dibuat, dari hasil analisa sistem diatas disimpulkan untuk pembuatan sistem peminjaman buku perpustakaan menggunakan teknologi *smart card* dan *SMS gateway*. Untuk penjelasannya lebih lanjut dibahas sebagai berikut.

3.5.1 Desain Halaman Login

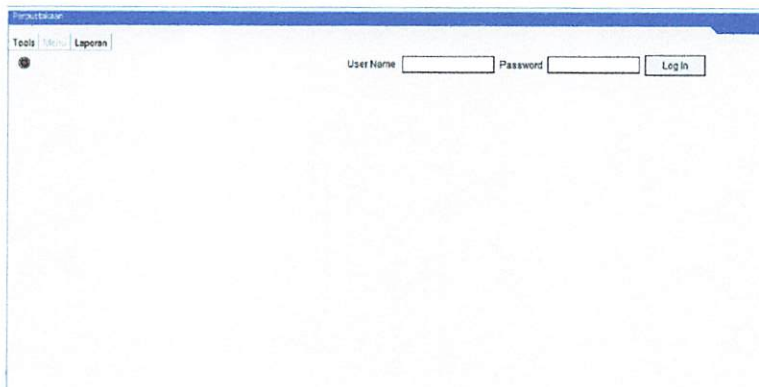
Pada halaman Login, seorang Admin diharuskan login terlebih dahulu untuk bisa menjalankan atau mengolah sistem. Untuk pembuatan desain login menggunakan *tEdit*, *Label*, *Button*, dan komponen yang digunakan *ADOTable (ADODB)* untuk pembentukan desainnya bisa dilihat pada gambar 3.11

User Name Password

Gambar 3.11 Desain Login

3.5.2 Desain Halaman Menu Utama

Pada halaman utama terdapat pilihan menu, yaitu menu *Tambah buku*, *daftar Anggita Baru*, *Tansaksi dan Laporan*, untuk pembentukan desainnya bisa dilihat pada gambar 3.12



Gambar 3.12 Desain Menu Halaman Utama

3.5.3 Desain Halaman Menu Tambah Buku

Pada halaman menu tambah buku, seorang admin dapat menabahkan data buku baru. Untuk pembuatan desain tambah buku menggunakan *tEdit*, *Label*, *Button* dan *DBgrid*, dan komponen yang digunakan *DataSource(DB)*, *ADOTable (ADODB)* untuk pembentukan desainnya bisa dilihat pada gambar 3.13

Gambar 3.13 Desain Menu Tambah Buku

3.5.4 Desain Halaman Menu Pendaftaran Anggota

Pada halaman menu pendaftaran anggota, seorang admin dapat memasukkan identitas mahasiswa, yang nantinya akan disimpan di database sistem. Untuk pembuatan desain menu pendaftaran anggota menggunakan *tEdit*, *Label*, *Button*, *DBgrid* dan *ComboBox* dan komponen yang digunakan *DataSource(DB)*, *OpenPictureDialog*, *SavePictureDialog*, *ADOTable (ADODB)* untuk pembentukan desainnya bisa dilihat pada gambar 3.14

The image shows a web form titled "Daftar Anggota Baru". On the left side, there are several input fields: "No Anggota" (with a dropdown arrow), "N I T A", "Nama", "Fakultas" (with a dropdown arrow), "Jurusan" (with a dropdown arrow), "Alamat Asal", "Alamat Sekarang", "No Telp 1", and "No Telp 2". To the right of these fields is a dashed rectangular box labeled "Upload Photo". Further to the right is a vertical column of four buttons: "Simpan", "Edit", "Hapus", and "Keluar". Below the input fields is a large empty rectangular area, likely intended for a data grid or table.

Gambar 3.14 Desain Menu Pendaftaran Anggota

3.5.5 Desain Halaman Menu Transaksi

Pada menu Transaksi, seorang admin hanya memasukkan No. Reg Buku1, No. Reg Buku 2, dan Batas Tanggal Peminjaman buku. Untuk pembuatan desain transaksi menggunakan *tEdit*, *Label*, *Button*, *DBgrid* *DateTimePicker* dan *ImageTr*, dan komponen yang digunakan *DataSource(DB)* untuk pembentukan desainnya bisa dilihat pada gambar 3.15

The screenshot shows a web form titled "Transaksi". It is organized into several sections:

- Member Information:** Fields for "No Anggota", "NIM", "Nama", "Fakultas", "Jurusan", "Alamat Asal", "Alamat Sekarang", and "No Telp 1".
- Registration and Dates:** Fields for "No Telp 2", "No Reg Buku 1", "No Reg Buku 2", "Tanggal Permisjaan", "Tanggal Pengembalian" (with a dropdown menu showing "16/06/2012"), "Buku 1:", and "Buku 2:".
- Photo Upload:** A dashed box labeled "Upload Foto" with a small camera icon.
- Action Buttons:** A vertical stack of buttons: "Simpan", "Edit", "Keluar", and "Batal".

Below the form fields is a large, empty rectangular area, likely intended for a data table or additional notes.

Gambar 3.15 Desain Menu Transaksi

3.5.6 Desain Halaman Menu Laporan

Pada menu Laporan, serorang admin dapat melihat data laporan peminjaman dan pengembalian, dan dapat mencetak data peminjaman buku. Untuk pembuatan desain menu laporan menggunakan *Label*, *Button*, *DBgrid* dan komponen yang digunakan *ADOQuery(ADODB)* untuk pembentukan desainnya bisa dilihat pada gambar 3.16

The screenshot shows a report page titled "LAPORAN PEMINJAMAN BUKU PERPUSTAKAAN". The page layout includes:

- Header:** A blue bar at the top with the text "LAPORAN PEMINJAMAN BUKU PERPUSTAKAAN" and a small printer icon on the right.
- Main Content:** A large, empty rectangular area, which is the intended location for the data table (DBgrid).
- Footer:** Two buttons at the bottom: "Exit" and "Cetak" (Print).

Gambar 3.16 Desain Menu Laporan

3.5.7 Desain Program Komunikasi Serial

Program komunikasi serial digunakan untuk menghubungkan program dengan *smart card reader* dan *Modem*. Komponen yang digunakan adalah *ComPort*.



Gambar 3.16 Komunikasi serial antara aplikasi dengan modem dan reader

BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM

Tahap pengujian sistem merupakan proses menjalankan aplikasi yang telah dirancang yang dapat dijalankan. Tahapan ini merupakan lanjutan dari proses perancangan sesuai dengan spesifikasi dan desain sistem. Untuk pengujian aplikasi sistem peminjaman buku ini yaitu dengan menjalankan sistem aplikasi.

4.1 Kebutuhan Sistem

Untuk mencapai hasil yang maksimal maka digunakan hardware dengan spesifikasi lebih dari apa yang direkomendasikan untuk software pada sistem ini. Dalam perancangan dan pembuatan aplikasi ini ada beberapa perangkat keras dan lunak komputer yang dibutuhkan antara lain :

1. Processor Intel Pentium Dual Core 2.0Ghz.
2. Memory 1024MB RAM.
3. Harddisk 250 GB
4. Microsoft Windows 7 Professional 32-bit sebagai sistem operasi.
5. Borland Delphi 7 untuk membuat aplikasi untuk interface antara *smart card* dan komputer serta modem untuk membuat tampilan bagi user di *server*.
6. 1 *Handphone, handphone* digunakan sebagai *SMS device* yang menerima dan mengirim SMS kepada admin.
7. Kartu AS Madura (Telkomsel) sebagai operator yang digunakan pada Modem Gateway.
8. Kabel Converter Serial to USB.
9. Kabel Converter PS2 to USB.

4.2 Pengujian Hardware

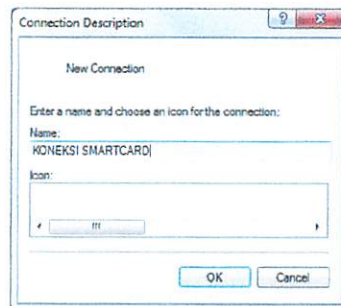
4.2.1 Pengujian Smart Card Reader

Pengujian smart card dilakukan dengan mencoba komunikasi dari reader ke komputer. Pengujian dilakukan dengan memanfaatkan fasilitas yang ada di sistem operasi *microsoft windows* yaitu *hyperterminal*.

4.2.1.1 Langkah pengujian

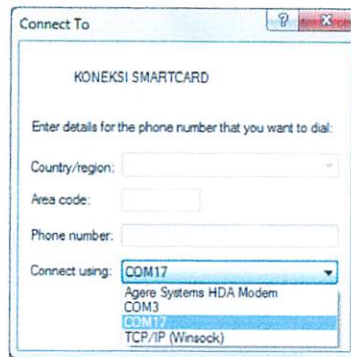
Sebelumnya untuk mengaktifkan atau menampilkan *Hyperterminal* di windows 7, caranya yaitu kita tinggal masuk ke komputer yang masih memakai OS Windows XP, kemudian Copy dua buah file yang berhubungan dengan hyper terminal yaitu **hypertrm.exe** dan **hypertrm.dll**.

- Untuk menjalankan Hyper Terminal, Eksekusi (dobel klik) file **hypertrm.exe**
- Setelah itu akan muncul dialog untuk mengisikan Connection Name seperti pada gambar 4.1



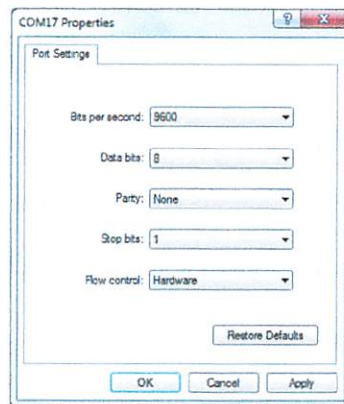
Gambar 4.1 Connection Name

- Masukan nama koneksinya misalkan KONEKSI SMARTCARD Klik OK
- Selanjutnya akan muncul dialog lagi Connect To, pada bagian Connet using pilih port com dimana modem terhubung dengan PC



Gambar 4.2 Pilih com port dimana *smart card reader* terhubung dengan PC

- Setelah itu akan klik tombol OK, maka akan muncul dialog lagi

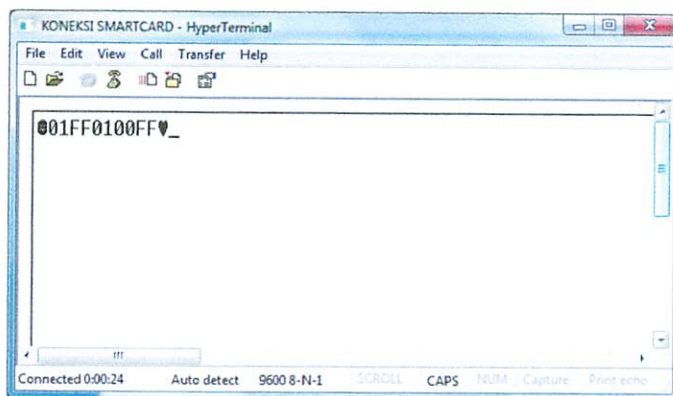


Gambar 4.3 Mengubah *Bits per second reader*

- Isikan *Bits per second* sesuai dengan Baudrate yang dipakai reader
- Selanjutnya akan muncul layar putih. Untuk pengetesan masukkan kartu kedalam *smart card reader*, lihat apa yang dihasilkan pada layar, apabila response yang dihasilkan berupa kode 01FF0100FF berarti *reader* dapat membaca kartu *smart card*

4.2.1.2 Hasil Pengujian

Tampilan pesan *command* untuk komunikasi port *smart card* pada program *HyperTerminal* ditunjukkan pada gambar 4.4



Gambar 4.4 Hasil pengujian *reader* dapat membaca kartu *smart card*

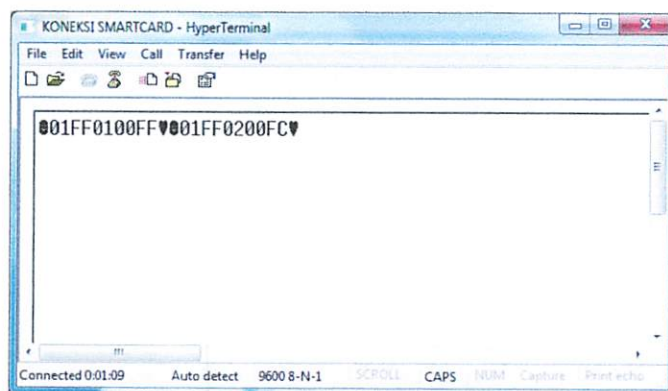
Dari hasil pengujian diatas didapatkan kode 01FF0100FF yang berarti *smart card reader* dapat membaca *kartu smart card*

Tabel 4.1 Pengujian Reader baca kartu smart card

Header	SW1	SW2	Data Length	Checksum
01 _H	FF _H	01 _H	00 _H	FF _H

Sumber : *Datasheet Refrence Manual ACR30*

Berikut tampilan status *command* koneksi *smart card* apabila kartu dilepas dari *smart card reader* dengan hasil pengujian didapatkan dengan kode 01FF0200FC



Gambar 4.5 Hasil pengujian status *reader* terputus dengan kartu

Tabel 4.2 Hasil pengujian reader terputus dengan kartu

Header	SW1	SW2	Data Length	Checksum
01 _H	FF _H	02 _H	00 _H	FC _H

Sumber : Datasheet Refrence Manual ACR30

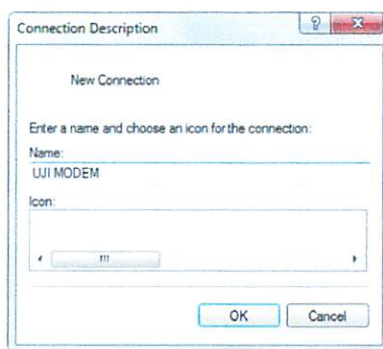
4.2.2 Pengujian Modem GSM Wavecom

Pengujian Modem GSM dilakukan dengan mencoba komunikasi dari modem ke komputer. Pengujian dilakukan dengan memanfaatkan fasilitas yang ada di sistem operasi *microsoft windows* yaitu *hyperterminal* yang nantinya dari Modem akan ditampilkan di *hyperterminal*.

4.2.2.1 Langkah pengujian

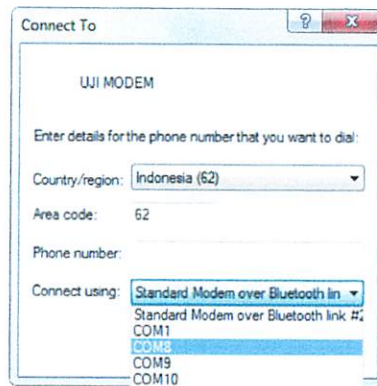
Sebelumnya untuk mengaktifkan atau menampilkan *Hyperterminal* di windows 7, caranya yaitu kita tinggal masuk ke komputer yang masih memakai OS Windows XP, kemudian Copy dua buah file yang berhubungan dengan hyper terminal yaitu **hypertrm.exe** dan **hypertrm.dll**.

- Untuk menjalankan Hyper Terminal, Eksekusi (dobel klik) file **hypertrm.exe**
- Setelah itu akan muncul dialog untuk mengisikan *Conection Name* seperti gambar 4.6



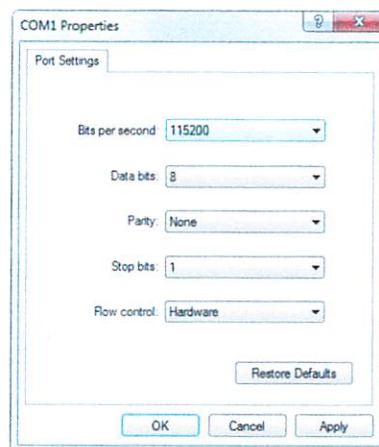
Gambar 4.6 *Conection Name*

- Masukkan nama koneksinya misalkan UJI MODEM Klik OK
- Selanjutnya akan muncul dialog lagi *Connect To*, pada bagian *Connet using* pilih port com dimana modem terhubung dengan PC



Gambar 4.7 Pilih port com dimana modem terhubung dengan PC

- Selanjutnya klik tombol OK, maka akan muncul dialog lagi

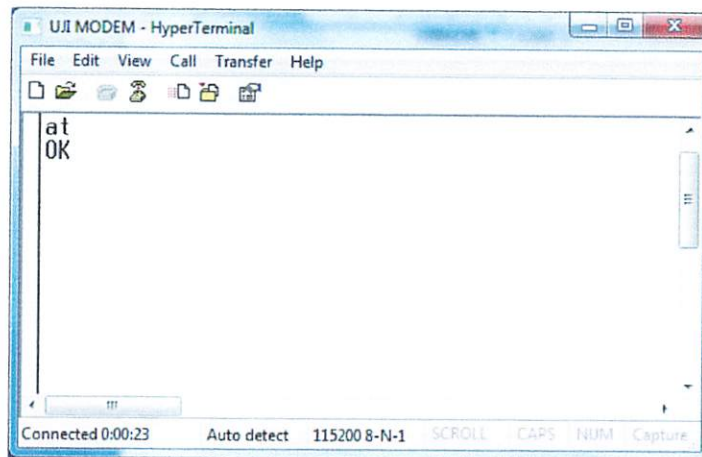


Gambar 4.8 Mengubah Bits per second modem

- Isikan *Bits per second* sesuai dengan Baudrate yang dipakai modem
- Selanjutnya akan muncul layar putih. Untuk pengetest ketikkan perintah AT bila response yang dihasilkan pada layar adalah OK, berarti anda telah sukses membuat koneksi ke modem.

4.2.2.2 Hasil Pengujian

Tampilan perintah *AT command* untuk komunikasi port pada program *HyperTerminal* ditunjukkan pada gambar 4.9



Gambar 4.9 Hasil pengujian koneksi Modem GSM

Pada gambar 4.9 menunjukkan bahwa pengujian koneksi antara modem GSM dengan komputer dapat terhubung dengan baik.

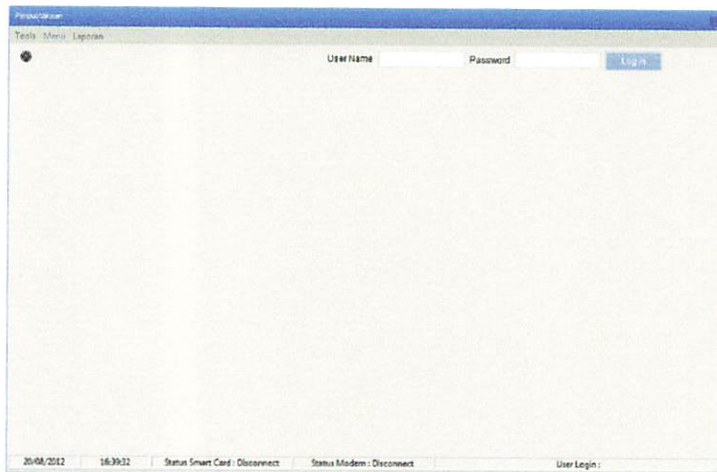
4.3 Pengujian Perangkat Lunak

Pengujian terhadap program yang telah selesai, dilakukan dengan cara melakukan uji coba terhadap proses yang ada secara satu persatu. Hal ini diperlukan untuk dapat melihat apakah program mampu berjalan dengan baik. Uji coba proses yang dilakukan akan difokuskan pada pengujian proses fungsionalitas aplikasi pada *notebook*.

Pada perangkat lunak aplikasi sistem peminjaman yang dibuat, pengembangan aplikasi menggunakan editor Borland Delphi IDE kemudian di-*compile* dan dijalankan dalam *notebook*.

4.3.1 Tampilan Utama *Appllication*

Tampilan utama pada aplikasi ini terdapat fitur Login yang sebelumnya seorang admin harus memasukkan nama dan password sebelum melihat menu utama dari aplikasi ini, Berikut tampilan utama seperti ditunjukkan pada gambar 4.10



Gambar 4.10 Tampilan Utama Login

Pada saat memasukan User Name dan password berhasil dialog akan menampilkan informasi “User Name dan Password OK”. Untuk melihat pesan dialog tersebut bisa dilihat pada gambar 4.11



Gambar 4.11 Login Sukses

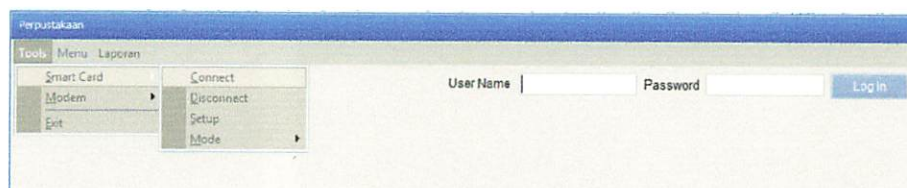
Jika pada saat memasukkan Nama dan Password salah, maka dialog akan menampilkan informasi “User Name dan Password Salah”. Untuk melihat pesan dialog tersebut bias dilihat pada gambar 4.12



Gambar 4.12 Login Gagal

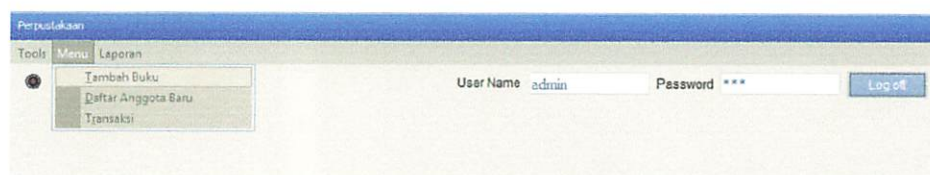
4.3.1.1 Menu Utama Program

Menu utama pada aplikasi Sistem Peminjaman buku ini terdapat menu-menu pilihan yang dapat dipilih oleh admin, seperti menu *Tool*, menu ini digunakan untuk mengkoneksikan hardware yang akan dihubungkan dengan aplikasi, didalam menu *Tool* ini juga terdapat pengaturan com port yang digunakan *smart card reader* maupun *modem*.



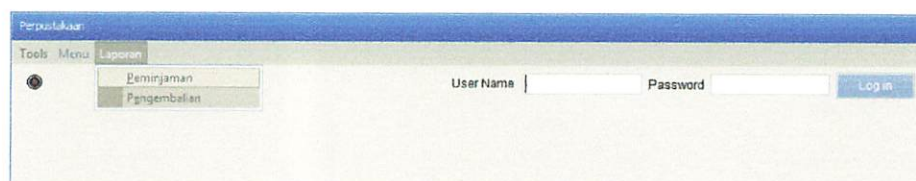
Gambar 4.13 Tampilan Program Menu Tool

Didalam menu utama program terdapat pilihan *Menu*, dimana dalam menu ini terdapat beberapa menu pilihan seperti menu *Tambah Buku*, *Daftar Anggota Baru* dan menu *Transaksi*.



Gambar 4.14 Program Menu Aplikasi

Dan didalam menu utama program terdapat juga pilihan *Laporan*, menu Laporan ini digunakan untuk melihat data peminjaman dan pengembalian buku dan dapat mencetak data peminjaman buku pada perpustakaan.

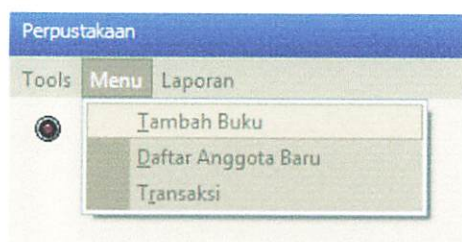


Gambar 4.15 Program Menu Laporan

4.3.1.2 Pengujian Program Tambah Buku

Pada menu *Tambah Buku* ini seorang admin dapat menambahkan data buku baru yang ingin di masukkan ke dalam sistem. Dalam menu ini terdapat field-field seperti No. Registeri, Judul Buku, Pengarang, Penerbit, dan Tahun.

Untuk menampilkan menu tambah buku admin memilih menu *Tambah Buku* pada menu *Edit*. Untuk melihat aplikasi pengujiannya bisa dilihat pada gambar 4.16



Gambar 4.16 Tampilan Menu Tambah Data Buku

Selanjutnya admin dapat menginputkan data buku baru pada sistem. Untuk melihat aplikasi pengujiannya bisa dilihat pada gambar 4.17



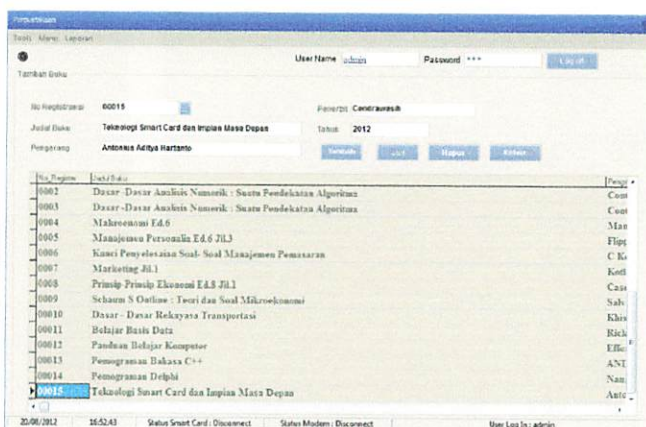
Gambar 4.17 Tampilan menginputkan Data Buku baru

Setelah selesai menginputkan data buku tersebut, kemudian menekan tombol **Simpan** dan dialog akan menampilkan pesan informasi "*Data Buku Tersimpan*". Untuk melihat pesan dialog tersebut bisa dilihat pada gambar 4.18



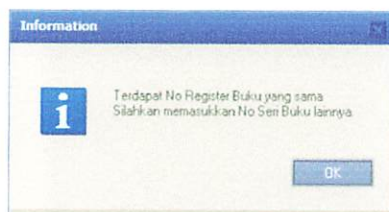
Gambar 4.18 Tampilan pesan dialog informasi Data Buku Tersimpan

Selanjutnya secara langsung data akan tersimpan pada sistem. Untuk melihat aplikasi pengujiannya bisa dilihat pada gambar 4.19



Gambar 4.19 Tampilan Data Buku Tersimpan

Jika dalam menginputkan nomor register buku yang sama, maka dialog akan menampilkan informasi “*Terdapat No. Register Buku yang sama. Silahkan memasukkan No. Seri Buku lainnya*”. Untuk melihat pesan dialog tersebut bisa dilihat pada gambar 4.20

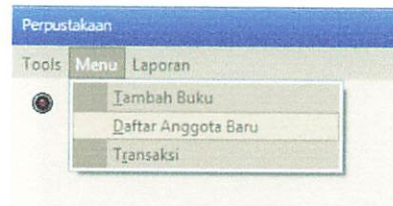


Gambar 4.20 Tampilan Pesan Dialog Informasi salah menginputkan No. Register Buku

4.3.1.3 Pengujian Program Daftar Anggota

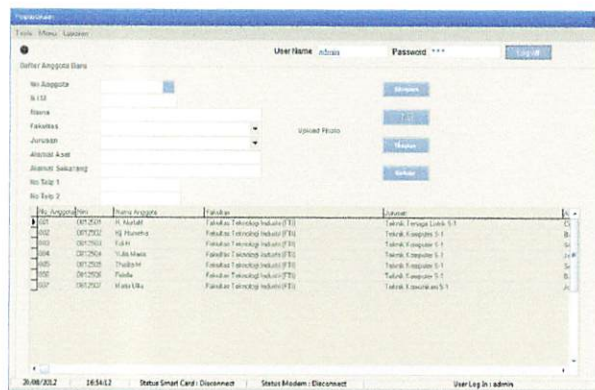
Pada menu *Daftar Anggota* ini, seorang admin dapat menginputkan data mahasiswa ke kartu. Pada menu ini admin dapat menginputkan data seperti No. Anggota, NIM, Nama, Fakultas, Jurusan, Alamat Asal, Alamat Sekarang No. Telepon, dan memasukkan photo mahasiswa. Identitas mahasiswa tersebut akan diinputkan pada kartu *smart card* dan disimpan pada database sistem. Disinilah bagaimana program dapat menulis data pada kartu *smart card*.

Sebelum menginputkan data ke kartu terlebih dahulu masukkan kartu ke slot *smart card reader*. Untuk menampilkan menu tambah buku admin memilih menu **Tambah buku** pada menu **Edit**. Untuk melihat aplikasi pengujianya bisa dilihat pada gambar 4.21



Gambar 4.21 Tampilan Menu Pilihan Daftar Anggota Baru

Setelah memilih Menu Daftar Anggota Baru, maka muncul program menu daftar anggota baru. Untuk melihat tampilan program tersebut bisa dilihat pada gambar 4.22



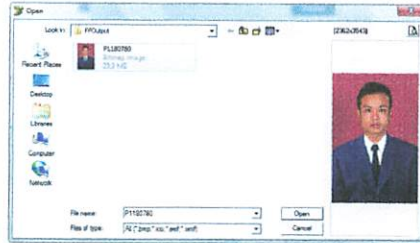
Gambar 4.22 Tampilan Aplikasi Daftar Anggota Baru

Selanjutnya admin dapat menginputkan Daftar Anggota Baru pada sistem. Untuk melihat aplikasi pengujianya bisa dilihat pada gambar 4.23.



Gambar 4.23 Tampilan mengelola Daftar Anggota Baru

Untuk mengupload photo anggota, sorang admin menekan double klik pada tulisan **“Upload Photo”** Untuk melihat aplikasi pengujiannya bisa dilihat pada gambar 4.24



Gambar 4.24 Tampilan mengupload photo anggota

Setelah selesai menginputkan Daftar Anggota Baru tersebut, kemudian menekan tombol **Simpan** dan dialog akan menampilkan pesan informasi **“Data Anggota Baru Tersimpan”**. Untuk melihat pesan dialog tersebut bisa dilihat pada gambar 4.25



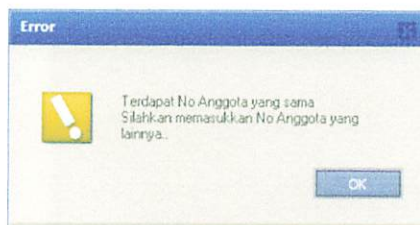
Gambar 4.25 Tampilan Pesan Dialog Informasi Data Anggota Baru Tersimpan

Selanjutnya secara otomatis data akan tersimpan pada sistem dan kartu *smart card*. Untuk melihat aplikasi pengujiannya bisa dilihat pada gambar 4.26



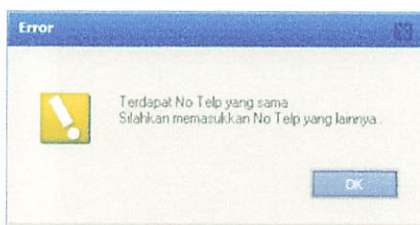
Gambar 4.26 Tampilan Data Anggota Baru Berhasil Disimpan.

Jika dalam menginputkan nomor anggota yang sama, maka dialog akan menampilkan informasi “*Terdapat No. Anggota yang sama. Silahkan memasukkan No. Seri Buku lainnya*”. Untuk melihat pesan dialog tersebut bisa dilihat pada gambar 4.27



Gambar 4.27 Tampilan Pesan Dialog Informasi menginputkan No. Anggota yang sama

Dan apabila dalam menginputkan nomor telepon yang sama, maka dialog akan menampilkan informasi “*Terdapat No. Telp yang sama. Silahkan memasukkan No. Telp yang lainnya*”. Untuk melihat pesan dialog tersebut bisa dilihat pada gambar 4.28



Gambar 4.28 Tampilan Pesan Dialog Informasi menginputkan No. Telp yang sama

4.3.1.4 Pengujian Program Menu Transaksi

Pada Menu *Transaksi* ini, program akan langsung menampilkan data yang tersimpan pada smart card, yang selanjutnya seorang admin hanya menginputkan *No. Registrasi buku pertama*, *No. Registrasi Buku ke dua*, dan *Tanggal pengembalian buku*, yang akan dipinjam seorang anggota atau mahasiswa yang bersangkutan. Pada pengujian ini disinilah program dapat membaca data dari kartu smart card, dan selanjutnya data baru dapat tersimpan kembali ke dalam kartu *smart card*.

Sebelum menginputkan data ke kartu terlebih dahulu masukkan kartu ke slot *smart card reader*. Selanjutnya sistem secara otomatis akan memunculkan identitas anggota yang

telah tersimpan pada kartu smart card. Untuk melihat tampilan program tersebut bisa dilihat pada gambar 4.29



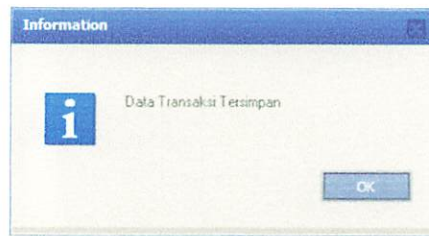
Gambar 4.29 Tampilan sistem menampilkan data identitas anggota

Berikutnya seorang admin memasukkan *No. Registrasi buku pertama*, *No. Reg Buku ke dua*, dan *Tanggal pengembalian buku*. Untuk melihat aplikasi pengujiannya tersebut bisa dilihat pada gambar 4.30



Gambar 4.30 Tampilan memasukkan data transaksi peminjaman buku

Setelah selesai menginputkan data transaksi tersebut, kemudian tekan tombol **Simpan** dan dialog akan menampilkan pesan informasi "*Data Transaksi Tersimpan*". Untuk melihat pesan dialog tersebut bisa dilihat pada gambar 4.31



Gambar 4.31 Tampilan Pesan Dialog Informasi Data Transaksi Tersimpan

Selanjutnya secara langsung data akan tersimpan pada sistem. Untuk melihat aplikasi pengujiannya bisa dilihat pada gambar 4.32



Gambar 4.32 Tampilan Data transaksi tersimpan pada sistem.

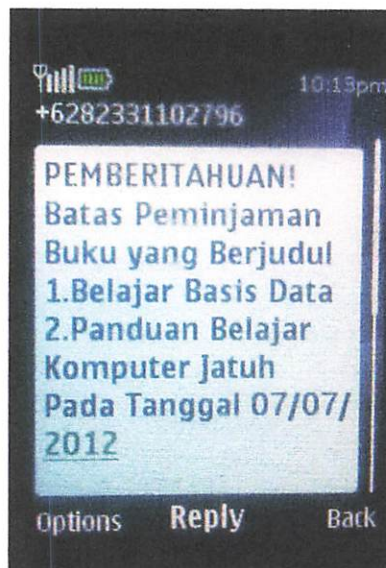
Dan apabila dalam menginputkan No. Register Buku salah, maka dialog akan menampilkan informasi “*No Register Buku Tidak Ditemukan atau Tidak Terdaftar*”. Untuk melihat pesan dialog tersebut bisa dilihat pada gambar 4.33



Gambar 4.33 Tampilan Pesan Dialog Informasi salah menginputkan No. Register Buku

4.3.1.5 Pengujian Pesan Pemberitahuan Batas Peminjaman

Pada pengujian ini sistem akan mengirimkan pesan pemberitahuan batas peminjaman buku secara otomatis satu hari sebelum jatuh tempo hari peminjaman buku kepada anggota. Sistem akan mengirimkan pesan berupa “**PEMBERITAHUAN! Batas Peminjaman Buku yang Berjudul 1.** (berisi judul buku pertama), **2.** (berisi Judul buku ke dua), **Jatuh Pada Tanggal dd/MM/yyyy** (berisi tanggal batas peminjaman buku) Untuk melihat pengujiannya bisa dilihat pada gambar 4.34 dan gambar 4.35



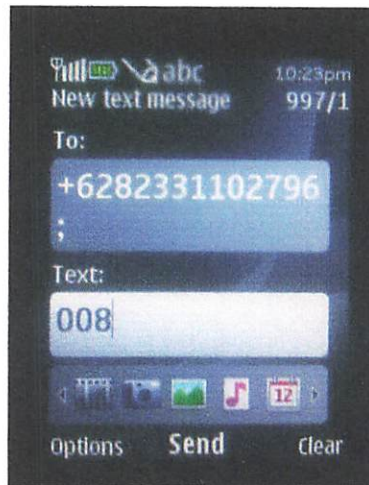
Gambar 4.34 Tampilan pesan pemberitahuan batas peminjaman buku ke No. Telp 1



Gambar 4.35 Tampilan pesan peringatan batas peminjaman buku ke No. Telp 2

4.3.1.6 Pengujian Anggota Mengirim Pesan Perpanjangan

Pada pengujian ini, seorang anggota dapat memperpanjang masa peminjaman buku dengan cara membalas pesan dari sistem, dengan cara mengetik “No. Anggota“ kirim ke nomor sistem. Disini sistem menggunakan nomor 082331102796. Sebagai contoh ketik **008** kirim **082331102796**. Untuk melihat pengujiannya bisa dilihat pada gambar 4.36



Gambar 4.36 Tampilan Mengirim Pesan Perpanjangan

4.3.1.7 Pengujian Sistem Mengirim Pesan Balasan Perpanjangan

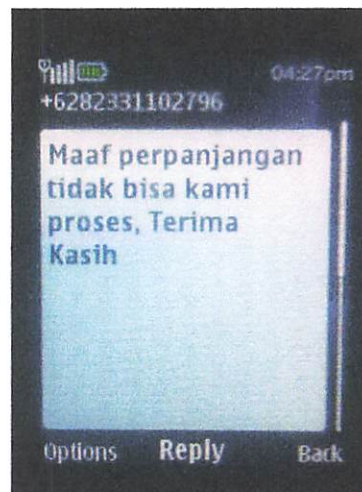
Pada pengujian ini sistem menerima pesan dari seorang anggota, yang selanjutnya sistem akan membalas pemberitahuan bahwa Peminjaman buku telah berhasil diproses. Berikut pesan yang akan dikirimkan sistem “*Peminjaman Telah di Perpanjang sampai dengan Tanggal dd/mm/yyyy (sistem akan menambah 7 hari masa perpanjangan), Terimakasih*”. Untuk melihat pengujiannya bisa dilihat pada gambar 4.37



Gambar 4.37 Tampilan sistem mengirimkan pesan balasan perpanjangan

4.3.1.8 Pengujian Pesan Sitem Menolak Perpajakan

Pada pengujian ini sistem akan menolak balasan dari anggota yang mengirimkan pesan perpanjangan untuk ke dua kalinya. Sistem akan mengirimkan pesan *“Maaf perpanjangan tidak bisa kami proses, Terimakasih”*. Untuk melihat pengujiannya bisa dilihat pada gambar 4.38



Gambar 4.38 Tampilan memasukkan data transaksi peminjaman buku

4.4 Hasil Pengujian

4.4.1 Hasil Pengujian Baca Kartu Smart Card

Pengujian ini dilakukan untuk mengetahui kondisi kartu pada saat kartu dimasukkan dalam *smart card reader*. Untuk melihat hasil pengujian dapat dilihat pada tabel 4.3

Tabel 4.3 Hasil Pengujian Baca kartu Smart Card

Pengujian Baca Kartu Smart Card											
Anggota	Uji Coba Ke-										Keterangan
	1	2	3	4	5	6	7	8	9	10	
Anggota 1	√	√	√	√	√	√	√	√	√	√	Sukses
Anggota 2	√	√	√	√	√	√	√	√	√	√	Sukses

Dari hasil pengujian dapat disimpulkan bahwa tingkat keberhasilan tiap pembacaan kartu oleh *smart card reader* selalu berhasil, hal tersebut dikarenakan *chip* yang berada pada kartu *smart card* dalam kondisi baik.

4.4.2 Hasil Pengujian Sistem Mengirim Pesan Pemberitahuan

Pada pengujian ini dilakukan untuk mengetahui berapa lama pesan dapat terkirim ke handphone anggota. Pengujian ini dilakukan dengan menggunakan 1 buah modem gateway, dimana mulai dari sistem mengirimkan pesan secara otomatis sampai pesan tersebut sampai ke anggota.. Untuk melihat hasil pengujian dapat dilihat pada tabel 4.4

Tabel 4.4 Pengujian Sistem Mengirim Pesan Pemberitahuan

NO	Sistem Kirim Pesan Otomatis		Terima Pesan dari Sistem		Keterangan
	Tanggal	Waktu	Tanggal	Waktu	
1	21/7/2012	07 : 57 : 10	21/7/2012	07 : 57 : 20	Sukses
2	22/7/2012	08 : 03 : 20	22/7/2012	08 : 03 : 30	Sukses
3	23/7/2012	10 : 10 : 35	23/7/2012	10 : 10 : 50	Sukses
4	24/7/2012	12 : 30 : 15	24/7/2012	12 : 30 : 30	Sukses
5	25/7/2012	15 : 40 : 45	25/7/2012	15 : 40 : 55	Sukses

Dari hasil pengujian tabel 4.4 ditunjukkan bahwa pemrosesan setiap pesan dari sistem ke anggota berjalan cukup baik, dengan waktu rata-rata 10 detik per sms yang terkirim ke nomor telepon anggota.

4.4.3 Hasil Pengujian Perpanjangan melalui SMS

Pada pengujian ini dilakukan untuk mengetahui berapa lama pesan balasan dari sistem dapat diterima oleh anggota. Pada uji coba sms dilakukan beberapa kali pengujian yakni 1 hingga 5 sms menggunakan 1 buah modem GSM, mulai dari penerimaan pesan dari pelanggan hingga pesan tersebut dibalas secara otomatis oleh sistem menuju nomor telepon anggota.

Tabel 4.5 Pengujian Perpanjangan melalui SMS

NO	Kirim Pesan		Terima Pesan Balasan dari Sistem		Keterangan
	Tanggal	Waktu	Tanggal	Waktu	
1	21/7/2012	07 : 57 : 25	21/7/2012	07 : 57 : 45	Sukses
2	22/7/2012	08 : 03 : 40	22/7/2012	08 : 04 : 00	Sukses
3	23/7/2012	10 : 10 : 55	23/7/2012	10 : 11 : 10	Sukses
4	24/7/2012	12 : 30 : 35	24/7/2012	12 : 30 : 55	Sukses
5	25/7/2012	15 : 41 : 00	25/7/2012	15 : 41 : 20	Sukses

Dari hasil pengujian tabel 4.5 ditunjukkan bahwa pemrosesan setiap pesan dari anggota ke sistem berjalan cukup baik, dan sistem mengirimkan pesan balasan perpanjangan berjalan dengan baik juga, dengan waktu rata-rata 20 detik per sms dari mulai anggota mengirimkan pesan ke sistem sampai sistem mengirim balik pesan perpanjangan secara otomatis.

4.4.4 Hasil Pengujian Aplikasi

Dari beberapa pengujian yang dilakukan pada sistem aplikasi peminjaman buku tersebut, maka diperoleh hasil secara umum yang ditunjukkan pada tabel 4.6

Tabel 4.6 Hasil *Pengujian Sistem Aplikasi secara umum*

NO	Pengujian	Hasil yang diharapkan	Hasil Pengamatan	Kesimpulan
1	Smart Card Reader	Dapat menghubungkan komunikasi antara smart card dengan PC	Sistem dapat berhubungan dengan kartu smart card	Sukses
2	Modem GSM	Dapat berkomunikasi dengan Komputer	Sistem dapat berkomunikasi dengan client	Sukses
3	Login Masukan Nama dan Password (Nama Benar salah password benar)	- Menampilkan pesan sukses - Dapat Login ke menu Utama	Sistem menampilkan pesan sukses dan dapat Login ke menu Utama	Sukses
4	Login Masukan Nama dan Password (Nama salah salah password salah)	- Menampilkan pesan error - Tidak dapat Login ke menu Utama	Sistem menampilkan pesan error dan tidak dapat Login ke menu Utama	Sukses
5	Tambah buku	Dapat menyimpan data ke sistem	Sistem dapat menyimpan data buku baru	Sukses
6	Daftar Anggota Baru	- Dapat menulis data id pada kartu smart card - Dapat menyimpan data pada sistem	Sistem dapat menulis data dan menyimpan pada database sistem dan kartu smart card.	Sukses
7	Transaksi peminjaman	- Dapat menampilkan data dari kartu - Data dapat ditulis kembali	Sistem dapat menampilkan data secara langsung dan dapat menulis atau menyimpan data baru pada sistem	Sukses
8	Kirim pesan	Dapat mengirimkan pesan secara otomatis	Sistem dapat mengirimkan pesan secara otomatis	Sukses

9	Terima pesan	Dapat terima pesan	Sistem menerima pesan perpanjangan	Sukses
10	Sistem balas pesan	Dapat mengirim pesan balasan perpanjangan	Sistem dapat mengirim pesan perpanjangan	Sukses
11	Sistem Menolak perpanjangan	Dapat menolak pesan perpanjangan ke dua melalui SMS	Sistem menolak kiriman pesan perpanjangan untuk kiriman pesan yang kedua	Sukses

Hasil pengujian menunjukkan bahwa sistem peminjaman buku ini dapat digunakan untuk menggantikan sistem lama. Dimana sistem ini kartu *smart card* dapat digunakan sebagai kartu peminjaman buku yang sah, tanpa harus lagi menulis fomulir peminjaman buku kembali, dan sistem dapat mengirim pesan pemberitahuan batas peminjaman buku, yang bertujuan anggota dapat mengembalikan peminjaman buku secara tepat waktu, tanpa harus terkena denda karena keterlambatan pengembalian buku.

BAB V PENUTUP

5.1 Kesimpulan

Dari perencanaan, pembuatan dan pengujian Aplikasi Sistem Peminjaman Buku Perpustakaan Menggunakan Terkonologi Smart Card dan SMS Gateway maka dapat diambil beberapa kesimpulan sebagai berikut :

1. Dengan adanya proses transaksi dengan menggunakan *smart card* para pelaku transaksi dimudahkan dengan tanpa harus menulis formulir peminjaman buku.
2. Proses transaksi dapat dilakukan dengan mudah, dengan memasukkan kartu ke *smart card reader*, sistem secara otomatis akan menampilkan data identitas anggota, seorang admin hanya memasukkan nomor register buku dan tanggal pengembalian buku.
3. Dengan menggunakan layanan teknologi *SMS Gateway* seorang mahasiswa dapat mengembalikan buku pinjaman tepat pada waktu yang telah ditentukan.
4. Sistem memudahkan seorang anggota untuk memperpanjang peminjaman buku dengan memanfaatkan teknologi *SMS Gateway* tanpa harus datang langsung ke ruang perpustakaan.
5. Dari hasil pengujian dapat disimpulkan bahwa tingkat keberhasilan tiap pembacaan kartu oleh *smart card reader* selalu berhasil, hal tersebut dikarenakan *chip* yang berada pada kartu *smart card* dalam kondisi baik.
6. Dari hasil pengujian sistem mengirimkan pesan pemberitahuan dapat ditunjukkan bahwa pemrosesan setiap pesan dari sistem ke anggota berjalan cukup baik, dengan waktu rata-rata 10 detik per sms yang terkirim ke nomor telepon anggota.
7. Dari hasil pengujian pepanjangan melalui sms ditunjukkan bahwa pemrosesan setiap pesan dari anggota ke sistem berjalan cukup baik, dan sistem mengirimkan pesan balasan perpanjangan berjalan dengan baik juga, dengan waktu rata-rata 20 detik per sms dari mulai anggota mengirimkan pesan ke sistem sampai sistem mengirim balik pesan perpanjangan secara otomatis.

8. Dari beberapa pengujian yang dilakukan pada sistem aplikasi peminjaman buku ini menunjukkan bahwa sistem peminjaman buku ini dapat digunakan untuk menggantikan sistem lama. Dimana sistem ini kartu *smart card* dapat digunakan sebagai kartu peminjaman buku yang sah, tanpa harus lagi menulis fomulir peminjaman buku kembali, dan sistem dapat mengirim pesan pemberitahuan batas peminjaman buku, yang bertujuan anggota dapat mengembalikan peminjaman buku secara tepat waktu, tanpa harus terkena denda karena keterlambatan pengembalian buku

5.2 Saran

Implementasi menggunakan teknologi *Smart Card* dan *SMS Gateway* pada Aplikasi Sistem Peminjaman buku ini dapat dikembangkan lebih jauh lagi karena dalam pembuatannya masih jauh dari sempurna dan masih banyak kekurangan. Adapun saran yang dapat dikemukakan agar aplikasi ini bisa berfungsi dengan lebih optimal adalah pengembangan aplikasi *smart card* tidak hanya terbatas pada aplikasi *smart card* sebagai kartu identitas saja, tetapi juga dapat dikembangkan untuk aplikasi lainnya yang membutuhkan kemudahan dalam tukar-menukar informasi dan transaksi dengan tingkat keamanan yang tinggi.

DAFTAR PUSTAKA

- [1] Andi. Seri Panduan Pemograman : Pemograman Borland Delphi 7. C.V.Andi Offset. Jogyakarta
- [2] Arif, Syamsul. 2008. "Manajemen Pengelolaan Perpustakaan: Perpustakaan Berbasis Teknologi Informasi dan Komunikasi". [http://syamsularif. Web.ugm.ac.id/?p=4](http://syamsularif.Web.ugm.ac.id/?p=4).
- [3] Berbagi Informasi Smart Card, <http://kadiginfo.Wordpress.com/2010/10/05/smart-card/>, Tanggal Akses 4 April 2012
- [4] Cahyo, Bambang. 2003. *Tinjauan Smart Card untuk Pengamanan Database Berbasis Komputer*. ITB. Bandung <http://budi.insan.co.id/courses/ec7010/dikmenjur/bambang-report.doc>
- [5] Kiswati, Kuni. 2011. *Komunikasi Pada Smart Card*. <http://kkiswati.wordpress.com/2011/02/11/bab-2-smart-card-bagian-2/>, Tanggal akses 27 Juni 2012
- [6] Microsoft Acces, http://id.wikipedia.org/wiki/Microsoft_Access, tanggal akses 28 Juni 2012
- [7] Peran Pustakawan, <http://sungadi-sungadi.blogspot.com/2010/12/peran-pustakawan-pada-perpustakaan.html>, tanggal akses 25 September 2012
- [8] Pamungkas, Dedy. Aplikasi Smart Card Sebagai Kartu Prabayar Internet. Universitas Diponegoro, Semarang http://eprints.undip.ac.id/25775/1/ML2_F099589.pdf
- [9] Pengenalan Perpustakaan <http://repository.usu.ac.id/bitstream/123456789/20057/4/Chapter%20II.pdf>. Tanggal akses 7 Juli 2012
- [10] *Perpustakaan Perguruan Tinggi : Buku Pedoman*. 2004. Edisi 3. Jakarta : Depdiknas
- [11] Sulisty, Basuki. 1991. *Pengantar Ilmu Perpustakaan*. Gramedia Pustaka Utama. Jakarta
- [12] Sulisty, Basuki. 1993. *Pengantar Ilmu Perpustakaan*. Gramedia Pustaka Utama. Jakarta
- [13] Sutarno NS. 2003. *Perpustakaan dan Masyarakat*. Yayasan Obor Indonesia, Jakarta

- [14] Smart Card dan Card Reader, <http://www.scribd.com/doc/53173250/ML2F099589>, Tanggal Akses 4 April 2012
- [15] SMS Gateway, http://id.wikipedia.org/wiki/SMS_Gateway, Tanggal Akses 4 April 2012
- [16] SMS Gateway BAB I, <http://www.scribd.com/doc/51083638/BAB-I-sls>. Tanggal Akses 10 Juli 2012
- [17] Teknologi SmartCard dan Impian di Masa Depan, onno.vlsm.org/v11/ref-ind-1/physical/SmartCardDream.rtf, Tanggal akses 7 April 2012



LAMPIRAN



PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI

Nama : Nur Toha Efendi
Nim : 0812531
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Komputer
Judul : **Aplikasi Sistem Peminjaman Buku Perpustakaan Menggunakan
Teknologi Smart Card dan SMS Gateway**

Dipertahankan dihadapan Majelis Penguji Skripsi Jenjang Strata Satu (S-1)

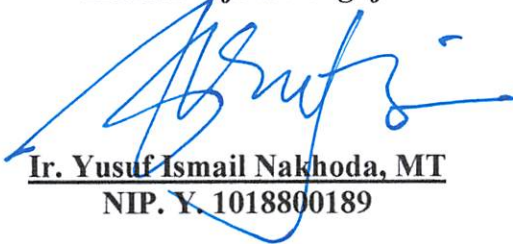
Pada Hari : Sabtu

Tanggal : 4 Agustus 2012

Dengan Nilai : 85,25 (A) *rd*

Panitia Ujian Skripsi

Ketua Majelis Penguji



Ir. Yusuf Ismail Nakhoda, MT
NIP. Y. 1018800189

Sekretaris Majelis Penguji



Dr. Eng. Aryuanto Soetedjo, ST, MT
NIP. P. 1030800417

Anggota Penguji

Penguji I



Bambang Prio Hartono, ST, MT
NIP. Y. 1028400082

Penguji II



Bima Aulia Firmandani, ST
1121



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
 FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
 PROGRAM PASCASARJANA MAGISTER TEKNIK

PT. BNI (PERSERO) MALANG
 BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
 Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

FORMULIR PERBAIKAN SKRIPSI

Dalam pelaksanaan ujian skripsi jenjang Strata 1 Jurusan Teknik Elektro Konsentrasi Teknik Komputer , maka perlu adanya perbaikan skripsi untuk mahasiswa :

Nama : Nur Toha Efendi
 Nim : 0812531
 Jurusan : Teknik Elektro S-1
 Konsentrasi : Teknik Komputer
 Masa Bimbingan : Semester Genap 2011-2012
 Judul : **Aplikasi Sistem Peminjaman Buku Perpustakaan Menggunakan Teknologi Smart Card dan SMS Gateway**

No	Penguji	Tanggal	Uraian	Paraf
1.	Penguji I	4 Agustus 2012	1. Abstrak disempurnakan 2. Kesimpulan	
2.	Penguji II	4 Agustus 2012	1. Flowchart 3.7 hal 32 2. Notifikasi program SMS perpanjangan 3. Laporan dibedakan - yang belum mengembalikan - yang sudah mengembalikan - tambah menu - per minggu - per bulan - per tahun 4. Kesimpulan	

Disetujui:

Dosen Penguji I

Dosen Penguji II

Bambang Prio Hartono, ST, MT
 NIP. Y. 1028400082

Bima Aulia Firmandani, ST
 1121

Mengetahui:

Dosen Pembimbing I

Dosen Pembimbing II

Dr. Eng. Aryuanto S, ST, MT
 NIP. P. 1030800417

M. Ibrahim Ashari, ST, MT
 NIP. P. 1030100358



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO
Kampus II : Jl. Raya Karanglo Km.2 Telp (0341) 417636

FORMULIR BIMBINGAN SKRIPSI

Nama : Nur Toha Efendi
Nim : 0812531
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Komputer
Masa Bimbingan : Semester Genap 2011-2012
Judul : **Aplikasi Sistem Peminjaman Buku Perpustakaan
Menggunakan Teknologi Smart Card dan SMS Gateway**

No.	Tanggal	Uraian	Paraf Pembimbing
1	12-07-2012	ACC BAB I, Revisi BAB II. Penambahan teori tentang perpustakaan, cantumkan nomor refrensi buku.	
2	13-07-2012	ACC BAB II, BAB III Revisi tata tulis.	
3	14-07-2012	ACC Bab III, BAB IV dan BAB V	
4	16-07-2012	ACC Seminar Hasil	
5	01-08-2012	ACC Ujian Kompre	

Malang,
Dosen Pembimbing I

Dr. Eng. Aryuanto S, ST, MT
NIP. P. 1030800417



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO
Kampus II : Jl. Raya Karanglo Km.2 Telp (0341) 417636

FORMULIR BIMBINGAN SKRIPSI

Nama : Nur Toha Efendi
Nim : 0812531
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Komputer
Masa Bimbingan : Semester Genap 2011-2012
Judul : Aplikasi Sistem Peminjaman Buku Perpustakaan
Menggunakan Teknologi Smart Card dan SMS Gateway

No.	Tanggal	Uraian	Paraf Pembimbing
1	16-07-2012	ACC BAB 1 Revisi BAB II. Point 2.1.5 nomer 1 sd 6 diberikan alinia. Penulisan keterangan gbr dan tabel yg jelas. Misal, seperti ditunjukkan pada gambar 2.5 dsb.	
2	17-07-2012	ACC BAB II. Revisi BAB III. Penulisan kalimat "seperti pada gambar dibawah ini, hendaknya dirubah menjadi seperti pada gambar 3.14. Begitu juga yg lain.	
3	24-07-2012	ACC BAB III, ACC BAB IV, ACC BAB V	

Malang,
Dosen Pembimbing II

M. Ibrahim Ashari, ST, MT
NIP. P. 1030100358



PERMOHONAN PERSETUJUAN SKRIPSI

Yang betanda tangan dibawah ini :

Nama : NUR TOHA EFENDI
 N I M : 08.12.531
 Semester : VII
 Fakultas : Teknologi Industri
 Jurusan : Teknik Elektro S-1
 Konsentrasi : ~~TEKNIK ELEKTRONIKA~~
 TEKNIK ENERGI LISTRIK
 TEKNIK KOMPUTER DAN INFORMATIKA
 TEKNIK KOMPUTER
 TEKNIK TELEKOMUNIKASI
 Alamat : Jl. Pemuda No 1. Desa Kewin. Kec. Ngajunoh. Kab Banyuwangi

Dengan ini kami mengajukan permohonan untuk mendapatkan persetujuan untuk membuat **SKRIPSI Tingkat Sarjana**. Untuk melengkapi permohonan tersebut, bersama kami lampirkan persyaratan-persyaratan yang harus dipenuhi.

Adapun persyaratan-persyaratan pengambilan **SKRIPSI** adalah sebagai berikut :

1. Telah melaksanakan semua praktikum sesuai dengan konsentrasinya (.....)
2. Telah lulus dan menyerahkan Laporan Praktek Kerja (.....)
3. Telah lulus seluruh mata kuliah keahlian (MKB) sesuai konsentrasinya (.....)
4. Telah menempuh mata kuliah ≥ 134 sks dengan IPK ≥ 2 dan tidak ada nilai E (.....)
5. Telah mengikuti secara aktif kegiatan seminar skripsi yang diadakan Jurusan (.....)
6. Memenuhi persyaratan administrasi (.....)

Demikian permohonan ini untuk mendapatkan penyelesaian lebih lanjut dan atas perhatiannya kami ucapkan terima kasih.

Telah diteliti kebenaran data tersebut diatas
 Recording Teknik Elektro

(*J. Ismail Nakhoda*)
 (.....)

Malang,201

Pemohon
Nur Toha Efendi
 (.....)

Disetujui
 Ketua Jurusan Teknik Elektro

J. Ismail Nakhoda
 Ir. Yusuf Ismail Nakhoda, MT
 NIP. Y. 1018800189

Mengetahui
 Dosen Wali

[Signature]
 (.....)

Catatan :

Bagi mahasiswa yang telah memenuhi persyaratan mengambil SKRIPSI agar membuat proposal dan mendapat persetujuan dari Ketua Jurusan/Sekretaris Jurusan T. Elektro S-1

1. *IP 409.5 / 138 - 3.53.*
2.
3.



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Nomor Surat : ITN-204/EL-FTI/2012
Lampiran : -
Perihal : BIMBINGAN SKRIPSI

Kepada : Yth. Bapak/Ibu **Dr. Eng. Aryuanto Soetedjo, ST, MT**
Dosen Teknik Elektro S-1
ITN MALANG

Dengan Hormat

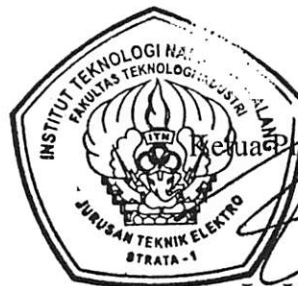
Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi untuk mahasiswa :

Nama : **NUR TOHA EFENDI**
Nim : **0812531**
Fakultas : **Teknologi Industri**
Program Studi : **Teknik Elektro S-1**
Konsentrasi : **Teknik Komputer**

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya kepada Saudara/i selama masa waktu :

" Semester Genap Tahun Akademik 2011 - 2012 "

Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.



Mengetahui

Ketua Program Studi Teknik Elektro S-1

Ir. Yusuf Ismail Nakhoda, MT

NIP.Y. 1018800189



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

BNI (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Nomor Surat : ITN-204/EL-FTI/2012

Lampiran : -

Perihal : BIMBINGAN SKRIPSI

Kepada : Yth. Bapak/Ibu **M. Ibrahim Ashari, ST, MT**
Dosen Teknik Elektro S-1
ITN MALANG

Dengan Hormat

Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi untuk mahasiswa :

Nama : **NUR TOHA EFENDI**
Nim : **0812531**
Fakultas : **Teknologi Industri**
Program Studi : **Teknik Elektro S-1**
Konsentrasi : **Teknik Komputer**

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya kepada Saudara/i selama masa waktu :

" Semester Genap Tahun Akademik 2011 - 2012 "

Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.



Mengetahui

Ketua Program Studi Teknik Elektro S-1

Ir. Yusuf Ismail Nakhoda, MT

NIP.Y. 1018800189



PROGRAM STUDI TEKNIK ELEKTRO S-1

FAKULTAS TEKNOLOGI INDUSTRI

INSTITUT TEKNOLOGI NASIONAL MALANG

Kampus II : Jl. Raya Karanglo Km. 2 Telp. (0341) 417636 Malang

jumlah : 1 (satu) berkas
Pembimbing Skripsi

kepada : Yth. Bapak/Ibu Dr. Eng. Aryuanto Soetedjo, ST, MT
Dosen Teknik Elektro S-1
ITN Malang

Yang bertanda tangan dibawah

Nama : **NUR TOHA EFENDI**
Nim : **0812531**
Jurusan : **Teknik Elektro S-1**
Konsentrasi : **Teknik Komputer**

Dengan ini mengajukan permohonan, kiranya Bapak/Ibu bersedia menjadi Dosen Pembimbing untuk penyusunan Skripsi dengan judul :

"APLIKASI SISTEM PEMINJAMAN BUKU PERPUSTAKAAN MENGGUNAKAN TEKNOLOGI SMART CARD DAN SMS GATEWAY "

Demikian permohonan kami buat dan atas kesediaan Bapak kami ucapkan terima kasih.

Mengetahui

Ketua Program Studi Teknik Elektro S-1


Ir. Yusuf Ismail Nakhoda, MT

NIP. X. 1018800189

Hormat Kami


NUR TOHA EFENDI

NIM. 0812531



PROGRAM STUDI TEKNIK ELEKTRO S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
Kampus II : Jl. Raya Karanglo Km. 2 Telp. (0341) 417636 Malang

mpiran : 1 (satu) berkas
Pembimbing Skripsi

pada : Yth. Bapak/Ibu M. Ibrahim Ashari, ST, MT
Dosen Teknik Elektro S-1
ITN Malang

Yang bertanda tangan dibawah

Nama : **NUR TOHA EFENDI**
Nim : **0812531**
Jurusan : **Teknik Elektro S-1**
Konsentrasi : **Teknik Komputer**

Dengan ini mengajukan permohonan, kiranya Bapak/Ibu bersedia menjadi Dosen Pembimbing untuk penyusunan Skripsi dengan judul :

"APLIKASI SISTEM PEMINJAMAN BUKU PERPUSTAKAAN MENGGUNAKAN TEKNOLOGI SMART CARD DAN SMS GATEWAY "

Demikian permohonan kami buat dan atas kesediaan Bapak kami ucapkan terima kasih.

Mengetahui

Ketua Program Studi Teknik Elektro S-1

Ir. Yusuf Ismail Nakhoda, MT

NIP. Y. 1018800189

Hormat Kami

NUR TOHA EFENDI

NIM. 0812531



PROGRAM STUDI TEKNIK ELEKTRO S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
Kampus II : Jl. Raya Karanglo Km. 2 Telp. (0341) 417636 Malang

PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI

Sesuai permohonan dari mahasiswa/i :

Nama : **NUR TOHA EFENDI**
Nim : **0812531**
Semester : **VIII (Delapan)**
Jurusan : **Teknik Elektro S-1**
Konsentrasi : **Teknik Komputer**

Dengan ini menyatakan bersedia/tidak bersedia*) Membimbing skripsi dari mahasiswa tersebut, dengan judul :

" APLIKASI SISTEM PEMINJAMAN BUKU PERPUSTAKAAN MENGGUNAKAN TEKNOLOGI SMART CARD DAN SMS GATEWAY "

Demikian surat pernyataan ini kami buat agar dapat dipergunakan seperlunya.

Hormat Kami

Dr. Eng. Aryuanto Soetedjo, ST, MT

NIP.P. 1030800417

*) Coret yang tidak perlu



PROGRAM STUDI TEKNIK ELEKTRO S-1

FAKULTAS TEKNOLOGI INDUSTRI

INSTITUT TEKNOLOGI NASIONAL MALANG

Kampus II : Jl. Raya Karanglo Km. 2 Telp. (0341) 417636 Malang

PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI

Sesuai permohonan dari mahasiswa/i :

Nama : **NUR TOHA EFENDI**

Nim : **0812531**

Semester : **VIII (Delapan)**

Jurusan : **Teknik Elektro S-1**

Konsentrasi : **Teknik Komputer**

Dengan ini menyatakan bersedia/tidak bersedia*) Membimbing skripsi dari mahasiswa tersebut, dengan judul :

" APLIKASI SISTEM PEMINJAMAN BUKU PERPUSTAKAAN MENGGUNAKAN TEKNOLOGI SMART CARD DAN SMS GATEWAY "

Demikian surat pernyataan ini kami buat agar dapat dipergunakan seperlunya.

Hormat Kami

M. Ibrahim Ashari, ST, MT

NIP.P. 1030100358

Catatan :

Setelah disetujui agar formulir ini Diserahkan mahasiswa/i yang bersangkutan kepada jurusan untuk diproses lebih lanjut

*) Coret yang tidak perlu

program Perpus;

uses

Forms,
Unit1 in 'Unit1.pas' {Main},
Unit2 in 'Unit2.pas' {Form2},
Unit3 in 'Unit3.pas' {FormReport},

{SR *.res}

begin

Application.Initialize;
Application.CreateForm(TMain, Main);
Application.CreateForm(TForm2, Form2);
Application.CreateForm(TFormReport, FormReport);
Application.Run;
end.

unit **Unit1**;

interface

uses

Windows, Messages, SysUtils, Variants, Classes,
Graphics, Controls, Forms,
Dialogs, Menus, SUIMainMenu, CPort, SUIMgr,
ExtCtrls, SUIForm,
SUIImagePanel, StdCtrls, Mask, SUIEdit, SUIButton,
DB, ADODB, SUIDlg,
SUIGroupBox, Grids, DBGrids, SUIDBCtrls,
SUIComboBox, ComCtrls, ExtDlgs,
SUIListBox, CPortCtl, SUIStatusBar;

type

TMain = class(TForm)
 suiForm1: TsuiForm;
 Theme: TsuiFileTheme;
 suiMainMenu1: TsuiMainMenu;
 ools1: TMenuItem;
 ComPortSC: TComPort;
 ComPortModem: TComPort;
 ButtonLogin: TsuiButton;
 EditUsername: TsuiEdit;
 EditPass: TsuiMaskEdit;
 Label1: TLabel;
 Label2: TLabel;
 ADOConnection1: TADOConnection;
 TableLogin: TADOTable;
 SMessageDialog: TsuiMessageDialog;
 Edit1: TMenuItem;
 ambahBuku1: TMenuItem;
 GroupTambahBuku: TsuiGroupBox;
 Label3: TLabel;
 Label4: TLabel;
 Label5: TLabel;
 Label6: TLabel;
 EditTNoReg: TsuiEdit;
 EditTJudulBuku: TsuiEdit;
 EditTPengarang: TsuiEdit;

EditTPenerbit: TsuiEdit;
ButtonTambah: TsuiButton;
ButtonEdit: TsuiButton;
ButtonBatal: TsuiButton;
DBGridBuku: TsuiDBGrid;
DataSourceBuku: TDataSource;
TableBuku: TADOTable;
TableAnggota: TADOTable;
GroupBoxTambahAnggota: TsuiGroupBox;
Label8: TLabel;
Label9: TLabel;
Label10: TLabel;
DataSourceAnggota: TDataSource;
DBGridAnggota: TsuiDBGrid;
Label11: TLabel;
Label12: TLabel;
Label13: TLabel;
Label14: TLabel;
EditDNoangt: TsuiEdit;
EditDNim: TsuiEdit;
EditDNama: TsuiEdit;
EditDALmtAsal: TsuiEdit;
EditDALmtSekarang: TsuiEdit;
EditDNoTlp: TsuiEdit;
Label15: TLabel;
ComboBoxDFakultas: TsuiComboBox;
ComboBoxDJurusan: TsuiComboBox;
ButtonDaftarSave: TsuiButton;
ButtonDaftarEdit: TsuiButton;
ButtonDaftarBatal: TsuiButton;
DaftarAnggotaBar1: TMenuItem;
GroupBoxTr: TsuiGroupBox;
Label16: TLabel;
Label17: TLabel;
Label18: TLabel;
Label19: TLabel;
Label20: TLabel;
Label21: TLabel;
Label22: TLabel;
Label23: TLabel;
Label24: TLabel;
Label25: TLabel;
Label26: TLabel;
EditTrNoAnggota: TsuiEdit;
EditTrNIM: TsuiEdit;
EditTrNama: TsuiEdit;
EditTrFak: TsuiEdit;
EditTrJurusan: TsuiEdit;
EditTrAlmtAsal: TsuiEdit;
EditTrAlmtSekarang: TsuiEdit;
EditTrNotlp: TsuiEdit;
EditTrNoReg1: TsuiEdit;
DateTimePickerPengembalian: TDateTimePicker;
TableTr: TADOTable;
DataSourceTr: TDataSource;
DBGridTr: TsuiDBGrid;
ButtonTrSave: TsuiButton;
ButtonTrEdit: TsuiButton;
ButtonTrBatal: TsuiButton;

```

ButtonTrHapus: TsuiButton;
ImageTr: TImage;
ImageDaftarAnggota: TImage;
OpenPictureDialog: TOpenPictureDialog;
SavePictureDialog: TSavePictureDialog;
ransaksi1: TMenuItem;
SmartCard1: TMenuItem;
Modem1: TMenuItem;
N1: TMenuItem;
Exit1: TMenuItem;
Connect1: TMenuItem;
Disconnect1: TMenuItem;
Setup1: TMenuItem;
Connect2: TMenuItem;
Disconnect2: TMenuItem;
Setup2: TMenuItem;
Label3x3Tr: TLabel;
Label3x3Anggota: TLabel;
Mode1: TMenuItem;
Baca1: TMenuItem;
ulis1: TMenuItem;
Edit2: TEdit;
TimerSC: TTimer;
ComLedCard: TComLed;
TimerCek: TTimer;
Label29: TLabel;
Label30: TLabel;
TimerMRead: TTimer;
TimerDateTime: TTimer;
Label31: TLabel;
StatusBar: TsuiStatusBar;
EditTrtg|Peminj: TsuiEdit;
Label7: TLabel;
EditTtahun: TsuiEdit;
ButtonTsch: TsuiButton;
ButtonDSch: TsuiButton;
Label27: TLabel;
EditTrNoReg2: TsuiEdit;
LabelBuku1: TLabel;
LabelBuku2: TLabel;
Labeltrbuku1: TLabel;
Labeltrbuku2: TLabel;
Button1: TButton;
ButtonHapus: TsuiButton;
ButtonAnggotaHapus: TsuiButton;
Label28: TLabel;
EditDNotlp2: TsuiEdit;
EditTrNoTlp2: TsuiEdit;
Label32: TLabel;
TimerSmsRead: TTimer;
DataSourceReport: TDataSource;
TableReport: TADOTable;
Laporan1: TMenuItem;
Peminjaman1: TMenuItem;
Pengembalian1: TMenuItem;
PopupMenu1: TPopupMenu;
Debug1: TMenuItem;
procedure FormCreate(Sender: TObject);
procedure ButtonLoginClick(Sender: TObject);

```

```

procedure ButtonBatalClick(Sender: TObject);
procedure ambahBuku1Click(Sender: TObject);
procedure ButtonTambahClick(Sender: TObject);
procedure DaftarAnggotaBar1Click(Sender: TObject);
procedure ButtonDaftarBatalClick(Sender: TObject);
procedure ransaksi1Click(Sender: TObject);
procedure ButtonTrBatalClick(Sender: TObject);
procedure Setup2Click(Sender: TObject);
procedure Setup1Click(Sender: TObject);
procedure Exit1Click(Sender: TObject);
procedure Label3x3TrDb1Click(Sender: TObject);
procedure Label3x3AnggotaDb1Click(Sender: TObject);
procedure ButtonDaftarSaveClick(Sender: TObject);
procedure ComPortSCRxChar(Sender: TObject; Count:
Integer);
function hextoint(input : string):longint;
procedure Baca1Click(Sender: TObject);
procedure ulis1Click(Sender: TObject);
procedure smartcardW;
procedure TimerSCTimer(Sender: TObject);
procedure Connect1Click(Sender: TObject);
procedure Disconnect1Click(Sender: TObject);
procedure TimerCekTimer(Sender: TObject);
procedure ComPortModemRxChar(Sender: TObject;
Count: Integer);
procedure Connect2Click(Sender: TObject);
procedure Disconnect2Click(Sender: TObject);
procedure TimerMReadTimer(Sender: TObject);
procedure TimerDateTimeTimer(Sender: TObject);
procedure mread;
procedure ButtonTrSaveClick(Sender: TObject);
procedure ButtonTschClick(Sender: TObject);
procedure ButtonEditClick(Sender: TObject);
procedure ButtonDaftarEditClick(Sender: TObject);
procedure ButtonDSchClick(Sender: TObject);
procedure EditUsernameKeyPress(Sender: TObject;
var Key: Char);
procedure EditPassKeyPress(Sender: TObject; var
Key: Char);
procedure FormShow(Sender: TObject);
procedure EditTrNoReg1KeyPress(Sender: TObject;
var Key: Char);
procedure EditTrNoReg2KeyPress(Sender: TObject;
var Key: Char);
procedure suiButton1Click(Sender: TObject);
procedure ButtonSmsSchClick(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure ButtonTrHapusClick(Sender: TObject);
procedure ButtonHapusClick(Sender: TObject);
procedure ButtonAnggotaHapusClick(Sender:
TObject);
procedure TimerSmsReadTimer(Sender: TObject);
procedure Pengembalian1Click(Sender: TObject);
procedure Peminjaman1Click(Sender: TObject);
procedure Debug1Click(Sender: TObject);
procedure ImageDaftarAnggotaDb1Click(Sender:
TObject);

```



```

private
  { Private declarations }
public
  { Public declarations }
end;

var
  Main: TMain;
  dir,nosim,crec,noreg,z,cmti,cmtino:String;
  hh,tt,nn:Integer;
  pict: array [1..2] of Boolean;
  smartmode:Boolean;
  smartstatus:Boolean;
  i,counter,int,j,ii,mcount,test: Integer;
  temps:array [1..5 , 1..10] of string;
  data:array [1..5 , 1..10] of Byte;
  sc:array [1..2 , 1..5] of string;
  buff:array [1..75] of string;
  buffm:array [1..100] of string;
  no:array [1..100] of string;
  modemmod,cmtimod:Boolean;
  stt:array [1..3] of Boolean;
  dedit:TDate;
implementation

uses Math, ADOInt, ComObj, Unit2, Unit3;

{$R *.dfm}

function TMain.hextoint(input : string):longint;
var
  c,i : longint;
  input1 : string;
begin
  c:=0;
  input1:= '';
  for i:=length(input) downto 1 do
input1:=input1+input[i]+'';
  input1:=uppercase(input1);
  for i:=1 to length(input1) do
  begin
    if (input1[i] in ['A'..'F'])and(i>1)then
      c:=((ord(input1[i])-ord('A')+10) shl (4*(i-
1)))or c
    else
    if (input1[i] in ['0'..'9'])and(i>1)then
      c:=((ord(input1[i])-ord('0')) shl (4*(i-1))) or c
    else
    if (input1[i] in ['A'..'F'])and(i=1)then
      c:=(ord(input1[i])-ord('A')+10) or c
    else
    if (input1[i] in ['0'..'9'])and(i=1)then
      c:=(ord(input1[i])-ord('0')) or c;
    end;
    result:=c;
  end;
end;
procedure TMain.mread;
begin
  stt[1]:=True;

```

```

  stt[2]:=True;
  stt[3]:=True;
  mcount:=1;
  Form2.ListBox1.Clear;
  modemmod:=True;
  for ii:=1 to 100 do
  begin
    buffm[ii]:= '';
    no[ii]:= '';
  end;
  ii:=1;
  crec:= '';
  nn:=1;
  if ComPortModem.Connected then
  begin
  // ComPortModem.SetRTS(True);
  if cmtimod then

ComPortModem.WriteStr('at+cmgr='+cmtino+#13)
  else

ComPortModem.WriteStr('at+cmgl="ALL"+#13);

  TimerMRead.Enabled:=True;
  end;
end;
procedure TMain.smartcardW;
begin
  data[1][10]:=0;
  temps[1][10]:='00202A';
  for counter:=1 to length(EditDNoangt.Text) do
  begin
temps[1][9]:=copy(EditDNoangt.Text,counter,1);
  int:=ord(temps[1][9][1]);
  temps[1][9]:=IntToHex(int,1);
  temps[1][10]:=temps[1][10]+temps[1][9];
  end;
  temps[1][10]:=temps[1][10]+'23';
  int:=length(temps[1][10]);
  int:=int div 2;
  temps[1][9]:= '';
  if int<=15 then temps[1][9]:='0';
  temps[1][9]:=temps[1][9]+IntToHex(int,1);
  temps[1][8]:='0191';
  temps[1][8]:=temps[1][8]+temps[1][9];
  temps[1][8]:=temps[1][8]+temps[1][10];
  int:=length(temps[1][8]);
  counter:=1;
  repeat
    temps[1][9]:=copy(temps[1][8],counter,2);
    data[1][9]:=hextoint(temps[1][9]);
    data[1][10]:=data[1][10] xor data[1][9];
    counter:=counter+2;
  until counter>int;
  temps[1][9]:=IntToHex(data[1][10],1);
  temps[1][8]:=temps[1][8]+temps[1][9];

```

```

sc[1][4]:=#02+temps[1][8]+#03;
data[1][3]:=1;

if not smartmode then
begin
  ComPortSC.WriteStr(SC[1][1]);
  Sleep(500);
  ComPortSC.WriteStr(SC[1][2]);
  Sleep(500);
  ComPortSC.WriteStr(SC[1][3]);
  Sleep(500);
  ComPortSC.WriteStr(SC[1][4]);
  Sleep(500);
  ComPortSC.WriteStr(SC[1][5]);
  Sleep(500);
end;
end;
procedure TMain.FormCreate(Sender: TObject);
begin
  Screen.Cursors[crsqlwait]:=Screen.Cursors[crdefault];
  test:=0;
  StatusBar.Panels.Items[2].Text:='Status Smart Card :
Disconnect';
  StatusBar.Panels.Items[3].Text:='Status Modem :
Disconnect';
  stt[1]:=False;
  stt[2]:=False;
  stt[3]:=False;
  smartstatus:=False;
  modemmod:=False;
  sc[1][1]:=#02+'0102010604'+#03;
  sc[1][2]:=#02+'01800081'+#03;
  sc[1][3]:=#02+'019203FFFFFFF'+#03;
  sc[1][5]:=#02+'01810080'+#03;
  sc[2][1]:=#02+'0102010604'+#03;
  sc[2][2]:=#02+'01800081'+#03;
  sc[2][3]:=#02+'019203FFFFFFF'+#03;
  sc[2][4]:=#02+'01900300101496'+#03;
  sc[2][5]:=#02+'01810080'+#03;
  counter:=1;
  buff[1]:='0';
  buff[2]:='1';
  buff[3]:='2';
  buff[4]:='3';
  buff[5]:='4';
  buff[6]:='5';
  buff[7]:='6';
  buff[8]:='7';
  buff[9]:='8';
  buff[10]:='9';
  buff[11]:='.';
  buff[12]:=',';
  buff[13]:='<';
  buff[14]:='=';
  buff[15]:='>';
  buff[16]:='?';
  buff[17]:='@';
  buff[18]:='A';
  buff[19]:='B';

  buff[20]:='C';
  buff[21]:='D';
  buff[22]:='E';
  buff[23]:='F';
  buff[24]:='G';
  buff[25]:='H';
  buff[26]:='I';
  buff[27]:='J';
  buff[28]:='K';
  buff[29]:='L';
  buff[30]:='M';
  buff[31]:='N';
  buff[32]:='O';
  buff[33]:='P';
  buff[34]:='Q';
  buff[35]:='R';
  buff[36]:='S';
  buff[37]:='T';
  buff[38]:='U';
  buff[39]:='V';
  buff[40]:='W';
  buff[41]:='X';
  buff[42]:='Y';
  buff[43]:='Z';
  buff[44]:='[';
  buff[45]:='\';
  buff[46]:=']';
  buff[47]:='^';
  buff[48]:='_';
  buff[49]:='`';
  buff[50]:='a';
  buff[51]:='b';
  buff[52]:='c';
  buff[53]:='d';
  buff[54]:='e';
  buff[55]:='f';
  buff[56]:='g';
  buff[57]:='h';
  buff[58]:='i';
  buff[59]:='j';
  buff[60]:='k';
  buff[61]:='l';
  buff[62]:='m';
  buff[63]:='n';
  buff[64]:='o';
  buff[65]:='p';
  buff[66]:='q';
  buff[67]:='r';
  buff[68]:='s';
  buff[69]:='t';
  buff[70]:='u';
  buff[71]:='v';
  buff[72]:='w';
  buff[73]:='x';
  buff[74]:='y';
  buff[75]:='z';

  ComLedCard.State:=lsOff;
  GroupBoxTr.Top:=88;

```

```

GroupBoxTr.Left:=0;
pict[1]:=False;
pict[2]:=False;
dir:=GetCurrentDir;
Theme.ThemeFile:=dir+'\skn.ssk';

ADOConnection1.ConnectionString:='Provider=Microso
ft.Jet.OLEDB.4.0;Data Source='+dir+'\Data.mdb;Persist
Security Info=False';
ADOConnection1.Connected:=True;
TableLogin.Connection:=ADOConnection1;
TableLogin.TableName:='Login';
TableLogin.Active:=True;
TableBuku.Connection:=ADOConnection1;
TableBuku.TableName:='Buku';
TableBuku.Active:=True;
Edit1.Enabled:=False;
DBGridBuku.Visible:=False;
DataSourceBuku.DataSet:=TableBuku;
DBGridBuku.DataSource:=DataSourceBuku;
TableAnggota.Connection:=ADOConnection1;
TableAnggota.TableName:='Anggota';
TableAnggota.Active:=True;
DataSourceAnggota.DataSet:=TableAnggota;
DBGridAnggota.DataSource:=DataSourceAnggota;
TableTr.Connection:=ADOConnection1;
TableTr.TableName:='Transaksi';
TableTr.Active:=True;
DataSourceTr.DataSet:=TableTr;
DBGridTr.DataSource:=DataSourceTr;
TableReport.Connection:=ADOConnection1;
TableReport.TableName:='Treport';
TableReport.Active:=True;
DataSourceReport.DataSet:=TableReport;
smartmode:=True;
Baca1.Checked:=True;
ulis1.Checked:=False;
GroupTambahBuku.Width:=1000;
GroupTambahBuku.Top:=88;
GroupTambahBuku.Left:=8;
GroupTambahBuku.Height:=545;
GroupBoxTambahAnggota.Width:=1000;
GroupBoxTambahAnggota.Top:=88;
GroupBoxTambahAnggota.Left:=8;
GroupBoxTambahAnggota.Height:=545;
GroupBoxTr.Width:=1000;
GroupBoxTr.Top:=88;
GroupBoxTr.Left:=8;
GroupBoxTr.Height:=545;
end;

procedure TMain.ButtonLoginClick(Sender: TObject);
begin
if (EditUsername.Text<>'') and (EditPass.Text<>'') and
(ButtonLogin.Caption='Log in')then
begin
TableLogin.Filtered:=False;
TableLogin.Filter:='Nama='+QuotedStr(EditUsername.

```

```

Text);
TableLogin.Filtered:=True;
if TableLogin.IsEmpty then
begin
SMessageDialog.Caption:='Error';
SMessageDialog.Text:='User Name dan Password
Salah';
SMessageDialog.IconType:=suiStop;
SMessageDialog.ShowModal;
end
else
begin
TableLogin.Filtered:=False;

TableLogin.Filter:='Password='+QuotedStr(EditPass.Text);
TableLogin.Filtered:=True;
if TableLogin.IsEmpty then
begin
SMessageDialog.Caption:='Error';
SMessageDialog.Text:='User Name dan Password
Salah';
SMessageDialog.IconType:=suiStop;
SMessageDialog.ShowModal;
end
else
begin
EditUsername.Enabled:=False;
EditPass.Enabled:=False;
StatusBar.Panels.Items[4].Text:='User Log In :
'+EditUsername.Text;
ButtonLogin.Caption:='Log off';
SMessageDialog.Caption:='Sukses';
SMessageDialog.Text:='User Name dan Password
OK!!!';
SMessageDialog.IconType:=suiInformation;
SMessageDialog.ShowModal;

cmtimod:=False;
tt:=0;
mread;
TimerSmsRead.Enabled:=True;
Edit1.Enabled:=True;
ools1.Enabled:=True;
Modem1.Enabled:=True;
SmartCard1.Enabled:=True;
end
end;
TableLogin.Filtered:=False;
end
else
begin
StatusBar.Panels.Items[4].Text:='User Log In :';
ButtonLogin.Caption:='Log in';
EditUsername.Clear;
EditUsername.Enabled:=True;
EditPass.Clear;
EditPass.Enabled:=True;
Edit1.Enabled:=False;
//ools1.Enabled:=False;

```

```

Modem1.Enabled:=False;
SmartCard1.Enabled:=False;
GroupTambahBuku.Visible:=False;
DBGridBuku.Visible:=False;
GroupBoxTr.Visible:=False;
end
end;

procedure TMain.ButtonBatalClick(Sender: TObject);
begin
GroupTambahBuku.Visible:=False;
DBGridBuku.Visible:=False;
EditTNoReg.Clear;
EditTJudulBuku.Clear;
EditTPengarang.Clear;
EditTPenerbit.Clear;
TableBuku.Filtered:=False;
end;

procedure TMain.ambahBuku1Click(Sender: TObject);
begin
GroupBoxTr.Visible:=False;
GroupTambahBuku.Visible:=True;
DBGridBuku.Visible:=False;
GroupBoxTambahAnggota.Visible:=False;
DBGridAnggota.Visible:=False;
end;

procedure TMain.ButtonTambahClick(Sender: TObject);
begin
if (EditTNoReg.Text<>') and (EditTJudulBuku.Text<>')
and (EditTPengarang.Text<>')
and (EditTPenerbit.Text<>') and (EditTtahun.Text<>')
then
begin
TableBuku.Filtered:=False;

TableBuku.Filter:='No_Register='+QuotedStr(EditTNoReg.Text);
TableBuku.Filtered:=True;
if TableBuku.IsEmpty then
begin
TableBuku.Insert;

TableBuku.FieldName('No_Register').AsString:=EditTNoReg.Text;
TableBuku.FieldName('JudulBuku').AsString:=EditTJudulBuku.Text;

TableBuku.FieldName('Pengarang').AsString:=EditTPengarang.Text;

TableBuku.FieldName('Penerbit').AsString:=EditTPenerbit.Text;

TableBuku.FieldName('Tahun').AsString:=EditTtahun.Text;
TableBuku.Post;
SMessageDialog.Caption:='Information';

```

```

SMessageDialog.Text:='Data Buku Tersimpan';
SMessageDialog.IconType:=suiInformation;
SMessageDialog.ShowModal;
end
else
begin
SMessageDialog.Caption:='Information';
SMessageDialog.Text:='Terdapat No Seri Buku yang sama'+#13+'Silahkan memasukkan No Seri Buku yang lainnya..';
SMessageDialog.IconType:=suiInformation;
SMessageDialog.ShowModal;
end;
TableBuku.Filtered:=False;
end
end;

procedure TMain.DaftarAnggotaBar1Click(Sender: TObject);
begin
GroupBoxTambahAnggota.Visible:=True;
DBGridAnggota.Visible:=True;
GroupTambahBuku.Visible:=False;
DBGridBuku.Visible:=False;
groupBoxTr.Visible:=False;
end;

procedure TMain.ButtonDaftarBatalClick(Sender: TObject);
begin
TableAnggota.Filtered:=false;
GroupBoxTambahAnggota.Visible:=False;
DBGridAnggota.Visible:=False;
end;

procedure TMain.ransaksi1Click(Sender: TObject);
begin
GroupBoxTambahAnggota.Visible:=False;
DBGridAnggota.Visible:=False;
GroupTambahBuku.Visible:=False;
DBGridBuku.Visible:=False;
GroupBoxTr.Visible:=True;
EditTrtgIPeminj.Text:=DateToStr(Now);
end;

procedure TMain.ButtonTrBatalClick(Sender: TObject);
begin
groupBoxTr.Visible:=False;
end;

procedure TMain.Setup2Click(Sender: TObject);
begin
ComPortModem.ShowSetupDialog;
end;

procedure TMain.Setup1Click(Sender: TObject);
begin
ComPortSC.ShowSetupDialog;
end;

```

```

procedure TMain.Exit1Click(Sender: TObject);
begin
if ButtonLogin.Caption='Log in' then
Application.Terminate
else
begin
SMessageDialog.Caption:='Error';
SMessageDialog.Text:='Silahkan Log Off terlebih
dahulu'+#13+'Sebelum anda keluar dari program ini..';
SMessageDialog.IconType:=suiWarning;
SMessageDialog.ShowModal;
end;
end;

procedure TMain.Label3x3TrDb1Click(Sender: TObject);
begin
if OpenPictureDialog.Execute then
begin

ImageTr.Picture.LoadFromFile(OpenPictureDialog.FileName);
Label3x3Tr.Visible:=False;
pict[2]:=True;
end
end;

procedure TMain.Label3x3AnggotaDb1Click(Sender:
TObject);
begin
if OpenPictureDialog.Execute then
begin
Label3x3Anggota.Visible:=False;

ImageDaftarAnggota.Picture.LoadFromFile(OpenPictur
eDialog.FileName);
pict[1]:=True;
end
end;

procedure TMain.ButtonDaftarSaveClick(Sender:
TObject);
label nxt;
begin
if (EditDNoangt.Text<>'')
and(EditDNim.Text<>'')and(EditDNama.Text<>'')

and(ComboBoxDFakultas.Text<>'')and(ComboBoxDJuru
san.Text<>'') and(EditDAImtAsal.Text<>'')
and(EditDAImtSekarang.Text<>'')
and(EditDNoTlp.Text<>'') and(pict[1]) or
(smartsstatus)then
begin
TableAnggota.Filtered:=False;

TableAnggota.Filter:='No_Anggota'+QuotedStr(EditDN
oangt.Text);
TableAnggota.Filtered:=True;
if TableAnggota.IsEmpty then

begin
TableAnggota.Filtered:=False;

begin
TableAnggota.Filter:='No_Telp_1'+QuotedStr(EditDNo
Tlp.Text);
TableAnggota.Filtered:=True;
if TableAnggota.IsEmpty then
begin
if EditDNotlp2.Text="" then goto nxt;
TableAnggota.Filtered:=False;

TableAnggota.Filter:='No_Telp_2'+QuotedStr(EditDNo
tlp2.Text);
TableAnggota.Filtered:=True;
if TableAnggota.IsEmpty then
begin
nxt:
TableAnggota.Insert;

TableAnggota.FieldByName('No_Anggota').AsString:=
EditDNoangt.Text;

TableAnggota.FieldByName('Nim').AsString:=EditDNim.
Text;
TableAnggota.FieldByName('Nama
Anggota').AsString:=EditDNama.Text;

TableAnggota.FieldByName('Fakultas').AsString:=Comb
oBoxDFakultas.Text;

TableAnggota.FieldByName('Jurusan').AsString:=Comb
oBoxDJurusan.Text;
TableAnggota.FieldByName('Alamat
Asal').AsString:=EditDAImtAsal.Text;
TableAnggota.FieldByName('Alamat
Sekarang').AsString:=EditDAImtSekarang.Text;

TableAnggota.FieldByName('No_Telp_1').AsString:=Edit
DNoTlp.Text;

TableAnggota.FieldByName('No_Telp_2').AsString:=Edit
DNotlp2.Text;
TableAnggota.Post; pict[1]:=False;

SavePictureDialog.FileName:=dir+'\Foto'+EditDNoangt
.Text+'.bmp';

ImageDaftarAnggota.Picture.SaveToFile(SavePictureDia
log.FileName);
ulis1.Click;
data[1][4]:=0;
smartsstatusW;
EditDNoangt.Clear;
EditDNim.Clear;
EditDNama.Clear;
ComboBoxDFakultas.Text:="";
ComboBoxDJurusan.Text:="";
EditDAImtAsal.Clear;
EditDAImtSekarang.Clear;

```

```

        EditDNoTlp.Clear;
        EditDNotlp2.Clear;
        Baca1.Click;
        SMessageDialog.Caption:='Information';
        SMessageDialog.Text:='Data Anggota Baru
Tersimpan';
        SMessageDialog.IconType:=suiInformation;
        SMessageDialog.ShowModal;
    end
else
    begin
        SMessageDialog.Caption:='Error';
        SMessageDialog.Text:='Terdapat No Telp
yang sama'+#13+'Silahkan memasukkan No Telp yang
lainnya..';
        SMessageDialog.IconType:=suiWarning;
        SMessageDialog.ShowModal;
    end
end
else
    begin
        SMessageDialog.Caption:='Error';
        SMessageDialog.Text:='Terdapat No Telp yang
sama'+#13+'Silahkan memasukkan No Telp yang
lainnya..';
        SMessageDialog.IconType:=suiWarning;
        SMessageDialog.ShowModal;
    end
end
else
    begin
        SMessageDialog.Caption:='Error';
        SMessageDialog.Text:='Terdapat No Anggota
yang sama'+#13+'Silahkan memasukkan No Anggota
yang lainnya..';
        SMessageDialog.IconType:=suiWarning;
        SMessageDialog.ShowModal;
    end;
end
TableAnggota.Filtered:=False;
end
end;

procedure TMain.ComPortSCRxChar(Sender: TObject;
Count: Integer);
begin
for i:=0 to Count do
begin
ComPortSC.Read(data[1][1],1);
temps[1][1]:=chr(data[1][1]);
temps[1][2]:=IntToStr(ord(temps[1][1][1]));
if temps[1][2]='2' then
begin
data[1][2]:=1;
temps[1][3]:='';
end
else
if (temps[1][2]='3') and (data[1][2]=1) then
begin
data[1][2]:=2;

```

```

if temps[1][3]='01FF0100FF' then
begin
StatusBar.Panels.Items[2].Text:='Status Card :
Inserted';
smartstatus:=True;
ComLedCard.State:=IsOn;
if smartmode then
begin
TimerSC.Enabled:=True;
sc[1][4]:=#02+'019003002014A6'+#03;
data[1][3]:=1;
end;
end
else
if temps[1][3]='01FF0200FC' then
begin
StatusBar.Panels.Items[2].Text:='Status Card :
Removed';
ComLedCard.State:=IsOff;
smartstatus:=False;
if GroupBoxTr.Visible=True then
begin
EditTrNoAnggota.Clear;
EditTrNIM.Clear;
EditTrNama.Clear;
EditTrFak.Clear;
EditTrJurusan.Clear;
EditTrAlmtAsal.Clear;
EditTrAlmtSekarang.Clear;
EditTrNotlp.Clear;
EditTrNoReg1.Clear;
EditTrNoReg2.Clear;
GroupBoxTr.Visible:=False;
end
end;
if length(temps[1][3])>= 50 then
begin
StatusBar.Panels.Items[2].Text:='Status Smart
Card : Reading';
temps[1][4]:='';
data[1][4]:=1;
repeat
temps[1][5]:=copy(temps[1][3],data[1][4],2);
data[1][4]:=data[1][4]+2;
until ((temps[1][5]='2A') or
(data[1][4]>length(temps[1][3])));
repeat
temps[1][5]:=copy(temps[1][3],data[1][4],2);
int:=hexToInt(temps[1][5]);
int:=int-47;
temps[1][4]:=temps[1][4]+buff[int];
data[1][4]:=data[1][4]+2;
until ((copy(temps[1][3],data[1][4],2)='23') or
(data[1][4]>length(temps[1][3])));
counter:=1;
data[1][4]:=length(temps[1][3]);
Edit2.Text:=temps[1][4];
TimerCek.Enabled:=True;

```



```

    end;

    end
    else
    if data[1][2]=1 then
temps[1][3]:=temps[1][3]+temps[1][1];
    Sleep(1);
    application.ProcessMessages;
    end;
end;

procedure TMain.Baca1Click(Sender: TObject);
begin
    smartmode:=True;
    Baca1.Checked:=True;
    ulis1.Checked:=False;
    StatusBar.Panels.Items[2].Text:='Status Smart Card :
Reading';
end;

procedure TMain.ulis1Click(Sender: TObject);
begin
    smartmode:=False;
    Baca1.Checked:=False;
    ulis1.Checked:=True;
end;

procedure TMain.TimerSCTimer(Sender: TObject);
begin
    if ComPortSC.Connected then
        begin
            StatusBar.Panels.Items[2].Text:='Status Smart Card :
Write';
            ComPortSC.WriteString(sc[1][data[1][3]]);
            inc(data[1][3]);
            if data[1][3]=6 then TimerSC.Enabled:=False;
                end
        end;
end;

procedure TMain.Connect1Click(Sender: TObject);
begin
    ComPortSC.Open;
    StatusBar.Panels.Items[2].Text:='Status Smart Card :
Connect';
end;

procedure TMain.Disconnect1Click(Sender: TObject);
begin
    ComPortSC.Close;
    StatusBar.Panels.Items[2].Text:='Status Smart Card :
Disconnect';
end;

procedure TMain.TimerCekTimer(Sender: TObject);
begin
    TimerCek.Enabled:=False;
    TableAnggota.Filtered:=False;
    TableAnggota.Filter:='No_Anggota'+QuotedStr(Edit2.
Text);

```

```

    TableAnggota.Filtered:=True;
    if TableAnggota.IsEmpty then
        begin
            SMessageDialog.Caption:='Error';
            SMessageDialog.Text:='No Anggota Tidak
Terdaftar..';
            SMessageDialog.IconType:=suiWarning;
            SMessageDialog.ShowModal;
        end
    else
        begin
            GroupBoxTr.Visible:=True;

            EditTrNoAnggota.Text:=TableAnggota.FieldValues['No_
Anggota'];
            EditTrNIM.Text:=TableAnggota.FieldValues['Nim'];
            EditTrNama.Text:=TableAnggota.FieldValues['Nama
Anggota'];

            EditTrFak.Text:=TableAnggota.FieldValues['Fakultas'];

            EditTrJurusan.Text:=TableAnggota.FieldValues['Jurusan
'];

            EditTrAlmtAsal.Text:=TableAnggota.FieldValues['Alama
t Asal'];

            EditTrAlmtSekarang.Text:=TableAnggota.FieldValues
['Alamat Sekarang'];

            EditTrNotlp.Text:=TableAnggota.FieldValues['No_Telp_
1'];

            EditTrNoTlp2.Text:=TableAnggota.FieldValues['No_Telp
_2'];

            OpenPictureDialog.FileName:=dir+'Foto\'+EditTrNoAn
ggota.Text+'.bmp';

            ImageTr.Picture.LoadFromFile(OpenPictureDialog.FileN
ame);
            Label3x3Tr.Visible:=False;
            EditTrtglPeminj.Text:=DateToStr(Now);
            TableTr.Filtered:=false;

            TableTr.Filter:='No_Anggota'+QuotedStr(Edit2.Text);
            TableTr.Filtered:=True;
            if TableTr.IsEmpty then
                TableTr.Filtered:=False
            else
                ButtonTrHapus.Visible:=True;
            end;
            TableAnggota.Filtered:=False;
        end;
end;

procedure TMain.ComPortModemRxChar(Sender:
TObject; Count: Integer);
var
    sm,ss:String;

```



```

begin
if modemmod then
begin
ComPortModem.ReadStr(sm,Count);
crec:=crec+sm;
end
else
begin
ComPortModem.ReadStr(sm,Count);
Form2.ListBox1.Items.Add(sm);
cmti:=cmti+sm;
if IntToStr(ord(cmti[length(cmti)-1]))='13' then
begin
cmtino:=cmti[length(cmti)-2];
cmtimod:=True;
tt:=0;
mread;
end
end
end;

procedure TMain.Connect2Click(Sender: TObject);
begin
ComPortModem.Open;
ComPortModem.SetRTS(True);
StatusBar.Panels.Items[3].Text:='Status Modem :
Connected';
end;

procedure TMain.Disconnect2Click(Sender: TObject);
begin
ComPortModem.Close;
StatusBar.Panels.Items[3].Text:='Status Modem :
Disconnect';
end;

procedure TMain.TimerMReadTimer(Sender: TObject);
var
fg,fs,cekno:String;
label tex;
begin
modemmod:=False;
TimerMRead.Enabled:=False;
if Length(crec)<20 then goto tex;
if test=2 then
begin
test:=3;
end;
for ii:=1 to length(crec) do
begin
Label30.Caption:=IntToStr(length(crec));
fg:=copy(crec,ii,1);
fs:=IntToStr(ord(fg[1]));
j:=StrToInt(fs);
if (j=34) and (stt[1]) then
stt[1]:=False
else
if (j=44) and(not stt[1]) and (stt[2]) then
stt[2]:=False
else
if (j=34) and (not stt[1]) and (not stt[2]) and (stt[3])
then
stt[3]:=False
else
if (not stt[1]) and (not stt[2]) and (not stt[3]) then
begin
nn:=ii+1;
Label30.Caption:= '';
nosim:=copy(crec,nn-15,1);
if nosim='' then nosim:=copy(crec,nn-16,1);
Label31.Caption:=copy(crec,nn-15,1);
if cmtimod then nosim:=cmtino;
repeat
Label30.Caption:=Label30.Caption+fg;
fg:=copy(crec,nn,1);
fs:=IntToStr(ord(fg[1]));
inc(nn);
until fg='';
repeat
fg:=copy(crec,nn,1);
fs:=IntToStr(ord(fg[1]));
j:=StrToInt(fs);
inc(nn);
until j=10;
Label29.Caption:= '';
repeat
if (j<>13) and (j<>10) then
Label29.Caption:=Label29.Caption+fg;
fg:=copy(crec,nn,1);
fs:=IntToStr(ord(fg[1]));
j:=StrToInt(fs);
inc(nn);
until j=13;
buffm[StrToInt(nosim)]:=Label29.Caption;
no[StrToInt(nosim)]:=Label30.Caption;

ComPortModem.WriteStr('at+cmgd='+nosim+#13);
inc(nn);
crec:=copy(crec,nn,length(crec));
stt[1]:=True;
stt[2]:=True;
stt[3]:=True;
if copy(no[strtoint(nosim)],1,1)='+' then
begin

z:=copy(no[strtoint(nosim)],4,length(no[strtoint(nosim)
]));
no[strtoint(nosim)]:= '0';
no[strtoint(nosim)]:=no[strtoint(nosim)]+z;
end;
TableTr.Filtered:=False;

TableTr.Filter:='No_Anggota'+QuotedStr(buffm[StrToI
nt(nosim)]);
TableTr.Filtered:=True;
if TableTr.IsEmpty then

else

```

```

begin
  z:=TableTr.FieldValues['No_Telp_1'];
  if z=no[StrToInt(nosim)] then
    begin
      cekno:=TableTr.FieldValues['Keterangan'];
      if cekno='N' then
        begin
          dedit:=StrToDate(TableTr.FieldValues['Tgl_Pengembalian']);
          dedit:=dedit+7;
          TableTr.Edit;

          TableTr.FieldByName('Tgl_Pengembalian').AsString:=DateToStr(dedit);

          TableTr.FieldByName('Keterangan').AsString:='P';
          TableTr.Post;
          if ComPortModem.Connected=True then
            begin
              Sleep(1000);

              ComPortModem.WriteStr('AT+CMGS='+z+#13);
              Sleep(100);
              ComPortModem.WriteStr('Peminjaman
Telah Di Perpanjang Sampai dengan Tanggal
'+DateToStr(dedit)+'Terima Kasih'+#26);
              Sleep(1000);
            end
          else
            begin
              Sleep(1000);

              ComPortModem.WriteStr('AT+CMGS='+z+#13);
              Sleep(100);
              comPortModem.WriteStr('Maaf
perpanjangan tidak bisa kami proses, Terima
Kasih'+#26);
              Sleep(1000);
            end
          end;
          TableTr.Filtered:=False;
          Form2.ListBox1.Items.Add('No Pengirim =
'+no[StrToInt(nosim)]+', '+buffm[StrToInt(nosim)];
          TimerMRead.Enabled:=True;
          inc(tt);
          Form2.Label6.Caption:=IntToStr(tt);
          goto tex;
        end
      else
        begin
          stt[1]:=True;
          stt[2]:=True;
          stt[3]:=True;
        end
      end;
    end
  if nn<=length(crec) then

```

```

    inc(nn);
    tex:
      TableTr.Filtered:=False;
      inc(test);
      end;

    procedure TMain.TimerDateTimeTimer(Sender:
    TObject);
    begin
      StatusBar.Panels[0].Text:=DateToStr(now);
      StatusBar.Panels[1].Text:=timeToStr(now);
      end;

    procedure TMain.ButtonTrSaveClick(Sender: TObject);
    begin
      if (EditTrNoAnggota.Text<>") and (EditTrNIM.Text<>")
      and (EditTrNama.Text<>")
      and (EditTrFak.Text<>") and (EditTrJurusan.Text<>")
      and (EditTrAlmtAsal.Text<>")
      and (EditTrAlmtSekarang.Text<>") and
      (EditTrNotlp.Text<>") and (EditTrNoReg1.Text<>")and
      (EditTrNoReg2.Text<>")
      and (EditTrtglPeminj.Text<>") and
      (EditTrtglPeminj.Text<>DateToStr(DateTimePickerPeng
      embalian.Date)) then
        begin
          TableBuku.Filtered:=False;

          TableBuku.Filter:='No_Register='+QuotedStr(EditTrNoReg1.
          Text);
          TableTr.Filtered:=True;
          if TableBuku.IsEmpty then
            begin
              SMessageDialog.Caption:='Error';
              SMessageDialog.Text:='No Seri Buku Tidak
              Terdaftar..';
              SMessageDialog.IconType:=suiWarning;
              SMessageDialog.ShowModal;
            end
          else
            begin
              TableTr.Insert;

              TableTr.FieldByName('No_Anggota').Text:=EditTrNoAn
              ggota.Text;
              TableTr.FieldByName('Nim').Text:=EditTrNIM.Text;
              TableTr.FieldByName('Nama
              Anggota').Text:=EditTrNama.Text;

              TableTr.FieldByName('Fakultas').Text:=EditTrFak.Text;

              TableTr.FieldByName('Jurusan').Text:=EditTrJurusan.Te
              xt;
              TableTr.FieldByName('Alamat
              Asal').Text:=EditTrAlmtAsal.Text;
              TableTr.FieldByName('Alamat
              Sekarang').Text:=EditTrAlmtSekarang.Text;

```

```

TableTr.FieldByName('No_Telp_1').Text:=EditTrNotlp.
Text;

TableTr.FieldByName('No_Telp_2').Text:=EditTrNoTlp2.
Text;
    TableTr.FieldByName('No Reg Buku
1').Text:=EditTrNoReg1.Text;
    TableTr.FieldByName('No Reg Buku
2').Text:=EditTrNoReg2.Text;
    TableTr.FieldByName('Judul Buku
1').Text:=Labeltrbuku1.Caption;
    TableTr.FieldByName('Judul Buku
2').Text:=Labeltrbuku2.Caption;
    TableTr.FieldByName('Tgl
Peminjaman').Text:=EditTrtglPeminj.Text;

TableTr.FieldByName('Tgl Pengembalian').Text:=DateT
oStr(DateTimePickerPengembalian.Date);
    TableTr.FieldByName('Keterangan').Text:='N';
    TableTr.Post;
    SMessageDialog.Caption:='Information';
    SMessageDialog.Text:='Data Transaksi Tersimpan';
    SMessageDialog.IconType:=suilInformation;
    SMessageDialog.ShowModal;
end;
TableBuku.Filtered:=False;
end
end;

procedure TMain.ButtonTschClick(Sender: TObject);
begin
    TableBuku.Filtered:=False;

TableBuku.Filter:='No_Register'+QuotedStr(EditTNoRe
g.Text);
    TableBuku.Filtered:=True;
    if TableBuku.IsEmpty then
        TableBuku.Filtered:=False
    else
        begin
            ButtonTambah.Enabled:=False;
            ButtonEdit.Enabled:=True;
            EditTJudulBuku.Text:=TableBuku.FieldValues['Judul
Buku'];

EditTPengarang.Text:=TableBuku.FieldValues['Pengara
ng'];

EditTPenerbit.Text:=TableBuku.FieldValues['Penerbit'];
noreg:=EditTNoReg.Text;
            if TableBuku.FieldValues['Tahun']=Null then
                EditTtahun.Text:=''
            else
                EditTtahun.Text:=TableBuku.FieldValues['Tahun'];
            end;
        end;
end;

procedure TMain.ButtonEditClick(Sender: TObject);
begin
    TableBuku.Edit;

TableBuku.FieldByName('No_Register').Text:=EditTNoR
eg.Text;
    TableBuku.FieldByName('Judul
Buku').Text:=EditTJudulBuku.Text;

TableBuku.FieldByName('Pengarang').Text:=EditTPenga
rang.Text;

TableBuku.FieldByName('Penerbit').Text:=EditTPenerbi
t.Text;

TableBuku.FieldByName('Tahun').Text:=EditTtahun.Text;
    TableBuku.Post;
    TableBuku.Filtered:=False;
    ButtonEdit.Enabled:=False;
    ButtonTambah.Enabled:=True;
    TableBuku.Filtered:=False;
end;

procedure TMain.ButtonDaftarEditClick(Sender:
TObject);
begin
    TableAnggota.Edit;

TableAnggota.FieldByName('No_Anggota').Text:=EditD
Noangt.Text;

TableAnggota.FieldByName('Nim').Text:=EditDNim.Text
;
    TableAnggota.FieldByName('Nama
Anggota').Text:=EditDNama.Text;

TableAnggota.FieldByName('Fakultas').Text:=ComboBo
xDfakultas.Text;

TableAnggota.FieldByName('Jurusan').Text:=ComboBox
DJurusan.Text;
    TableAnggota.FieldByName('Alamat
Asal').Text:=EditDAImltAsal.Text;
    TableAnggota.FieldByName('Alamat
Sekarang').Text:=EditDAImltSekarang.Text;

TableAnggota.FieldByName('No_Telp_1').Text:=EditDN
oTlp.Text;

TableAnggota.FieldByName('No_Telp_2').Text:=EditDN
otlp2.Text;
    TableAnggota.Post;
    TableAnggota.Filtered:=False;

SavePictureDialog.FileName:=dir+'\Foto'+EditDNoangt
.Text+'.bmp';

ImageDaftarAnggota.Picture.SaveToFile(SavePictureDia
log.FileName);

```

```

ButtonDaftarSave.Enabled:=True;
ButtonDaftarEdit.Enabled:=False;
EditDNoangt.Clear;
EditDNim.Clear;
EditDNama.Clear;
ComboBoxDFakultas.Clear;
ComboBoxDJurusan.Clear;
EditDALmtAsal.Clear;
EditDALmtSekarang.Clear;
EditDNoTlp.Clear;
EditDNotlp2.Clear;
SMessageDialog.Caption:='Information';
SMessageDialog.Text:='Data Anggota Baru
Tersimpan';
SMessageDialog.IconType:=suiInformation;
SMessageDialog.ShowModal;
end;

procedure TMain.ButtonDSchClick(Sender: TObject);
begin
    ButtonDaftarSave.Enabled:=False;
    TableAnggota.Filtered:=False;

TableAnggota.Filter:='No_Anggota='+QuotedStr(EditDN
oangt.Text);
    TableAnggota.Filtered:=True;
    if TableAnggota.IsEmpty then
        begin
            ButtonDaftarSave.Enabled:=true;
            TableAnggota.Filtered:=False;
        end
    else
        begin
            EditDNim.Text:=TableAnggota.FieldValues['Nim'];
            EditDNama.Text:=TableAnggota.FieldValues['Nama
Anggota'];

ComboBoxDFakultas.Text:=TableAnggota.FieldValues['F
akultas'];

ComboBoxDJurusan.Text:=TableAnggota.FieldValues['J
urusan'];

EditDALmtAsal.Text:=TableAnggota.FieldValues['Alamat
Asal'];

EditDALmtSekarang.Text:=TableAnggota.FieldValues['Al
amat Sekarang'];

EditDNoTlp.Text:=TableAnggota.FieldValues['No_Telp_
1'];

EditDNotlp2.Text:=TableAnggota.FieldValues['No_Telp
_2'];
            ButtonDaftarEdit.Enabled:=True;

OpenPictureDialog.FileName:=dir+'\Foto'+EditDNoangt
.Text+'.bmp';

```

```

ImageDaftarAnggota.Picture.LoadFromFile(OpenPicture
Dialog.FileName);
    Label3x3Anggota.Visible:=False;
end;
    ButtonDaftarSave.Enabled:=False;
end;

```

```

procedure TMain.EditUsernameKeyPress(Sender:
TObject; var Key: Char);
begin
    if Key=#13 then ButtonLogin.Click;
end;

```

```

procedure TMain.EditPassKeyPress(Sender: TObject;
var Key: Char);
begin
    if Key=#13 then ButtonLogin.Click;
end;

```

```

procedure TMain.FormShow(Sender: TObject);
begin
    EditUsername.SetFocus;
end;

```

```

procedure TMain.EditTrNoReg1KeyPress(Sender:
TObject; var Key: Char);
begin
    if Key=#13 then
        begin
            TableBuku.Filtered:=False;

```

```

TableBuku.Filter:='No_Register='+QuotedStr(EditTrNoR
eg1.Text);
    TableBuku.Filtered:=True;
    if TableBuku.IsEmpty then
        begin
            EditTrNoReg1.Clear;
            SMessageDialog.Caption:='Error';
            SMessageDialog.Text:='No Register Buku Tidak Di
Temukan atau Tidak Terdaftar...';
            SMessageDialog.IconType:=suiWarning;
            SMessageDialog.ShowModal;
        end
    else
        begin

```

```

Labelrbuku1.Caption:=TableBuku.FieldValues['Judul
Buku'];
        end;
        TableBuku.Filtered:=False;
end;
end;

```

```

procedure TMain.EditTrNoReg2KeyPress(Sender:
TObject; var Key: Char);
begin
    if Key=#13 then
        begin
            TableBuku.Filtered:=False;

```

```

TableBuku.Filter:= 'No_Register'+QuotedStr(EditTrNoReg2.Text);
TableBuku.Filtered:=True;
if TableBuku.IsEmpty then
begin
    EditTrNoReg2.Clear;
    SMessageDialog.Caption:='Error';
    SMessageDialog.Text:='No Register Buku Tidak Di
Temukan atau Tidak Terdaftar...';
    SMessageDialog.IconType:=suiWarning;
    SMessageDialog.ShowModal;
end
else
begin

Labeltrbuku2.Caption:=TableBuku.FieldValues['Judul
Buku'];
end;
TableBuku.Filtered:=False;
end;
end;
procedure TMain.suiButton1Click(Sender: TObject);
begin
    TableTr.Filtered:=False;

end;

procedure TMain.ButtonSmsSchClick(Sender: TObject);
begin
    TableTr.Filtered:=False;
    TableTr.Filtered:=True;
end;

procedure TMain.Button1Click(Sender: TObject);
begin
    cmntimod:=False;
    tt:=0;
    mread;
end;

procedure TMain.ButtonTrHapusClick(Sender:
TObject);
begin
    if (Application.MessageBox('Anda yakin akan
menghapus Data ini ', 'Konfirmasi
Penghapusan', MB_YESNO or
MB_ICONQUESTION)=IDNO) then
        Abort
    else
        begin
            TableReport.Insert;

TableReport.FieldByName('No_Anggota').Text:=TableTr
.FieldValues['No_Anggota'];

TableReport.FieldByName('Nama
Anggota').Text:=TableTr.FieldValues['Nama Anggota'];

TableReport.FieldByName('Fakultas').Text:=TableTr.Fiel
dValues['Fakultas'];

TableReport.FieldByName('Jurusan').Text:=TableTr.Fiel
dValues['jurusan'];

TableReport.FieldByName('No_Telp_1').Text:=TableTr.F
ieldValues['No_Telp_1'];
        if TableTr.FieldValues['No_telp_2']<>Null then

TableReport.FieldByName('No_Telp_2').Text:=TableTr.F
ieldValues['No_Telp_2'];
            TableReport.FieldByName('No Reg Buku
1').Text:=TableTr.FieldValues['No Reg Buku 1'];
            TableReport.FieldByName('No Reg Buku
2').Text:=TableTr.FieldValues['No Reg Buku 2'];
            TableReport.FieldByName('Judul Buku
1').Text:=TableTr.FieldValues['Judul Buku 1'];
            TableReport.FieldByName('Judul Buku
2').Text:=TableTr.FieldValues['Judul Buku 2'];
            TableReport.FieldByName('Tgl
Peminjaman').Text:=TableTr.FieldValues['Tgl
Peminjaman'];

TableReport.FieldByName('Tgl_Pengembalian').Text:=T
ableTr.FieldValues['Tgl_Pengembalian'];
            TableReport.Post;
            TableTr.Delete;
            TableTr.First;
            ButtonTrHapus.Visible:=False;
            TableTr.Filtered:=False;
        end
    end;

procedure TMain.ButtonHapusClick(Sender: TObject);
begin
    if (Application.MessageBox('Anda yakin akan
menghapus Data ini ', 'Konfirmasi
Penghapusan', MB_YESNO or
MB_ICONQUESTION)=IDNO) then
        Abort
    else
        begin
            TableBuku.Delete;
            TableBuku.First;
        end
    end;

procedure TMain.ButtonAnggotaHapusClick(Sender:
TObject);
begin
    if (Application.MessageBox('Anda yakin akan
menghapus Data ini ', 'Konfirmasi
Penghapusan', MB_YESNO or
MB_ICONQUESTION)=IDNO) then
        Abort

```

```

else
begin
  TableAnggota.Delete;
  TableAnggota.First;
end
end;

procedure TMain.TimerSmsReadTimer(Sender:
TObject);
begin
  TimerSmsRead.Enabled:=False;
  Form2.Button2.Click;
end;

procedure TMain.Pengembalian1Click(Sender:
TObject);
begin
  with FormReport do begin
    Label1.Caption:='LAPORAN PENGEMBALIAN BUKU
PERPUSTAKAAN';
    DBGridReport.DataSource:=DataSourceReport;
    ButtonCetak.Visible:=False;
    show;
  end;
end;

procedure TMain.Peminjaman1Click(Sender: TObject);
begin
  with FormReport do begin
    Label1.Caption:='LAPORAN PEMINJAMAN BUKU
PERPUSTAKAAN';
    DBGridReport.DataSource:=DataSourceTr;
    ButtonCetak.Visible:=True;
    show;
  end;
end;

procedure TMain.Debug1Click(Sender: TObject);
begin
  Form2.Show;
end;

procedure TMain.ImageDaftarAnggotaDbClick(Sender:
TObject);
begin
  if OpenPictureDialog.Execute then
  begin
    Label3x3Anggota.Visible:=False;

ImageDaftarAnggota.Picture.LoadFromFile(OpenPictur
eDialog.FileName);
    pict[1]:=True;
  end
end;

end.

```

```

unit Unit2;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes,
  Graphics, Controls, Forms,
  Dialogs, StdCtrls, Grids, ExtCtrls, Cport;

type
  TForm2 = class(TForm)
    Button1: TButton;
    Button2: TButton;
    StringGrid1: TStringGrid;
    Timer1: TTimer;
    Edit1: TEdit;
    ListBox1: TListBox;
    Label1: TLabel;
    Button3: TButton;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    ComboBox1: TComboBox;
    Label6: TLabel;
    Label7: TLabel;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form2: TForm2;

implementation

uses Unit1, DB;

{$R *.dfm}
var
  no,tt: integer;
  //item:array [1..4 , 1..10] of string;
  procedure TForm2.Button1Click(Sender: TObject);
  begin
    close;
  end;

  procedure TForm2.Button2Click(Sender: TObject);
  var
    i: integer;t,tx,m: TDate;n: TTimeStamp;

```



```

ck: boolean;
begin

n:=DateTimeToTimeStamp(now);n.Time:=0;m:=TimeSt
ampToDateTime(n);
listbox1.Clear;ck:=false;
button2.Caption:=datetimetostr(m);
with main do
begin
TableTr.First;i:=0;
stringgrid1.RowCount:=TableTr.RecordCount+1;
repeat

t:=strtodate(TableTr.FieldValues['Tgl_Pengembalian']);
tx:=m-t;
if (tx=-1) then
begin

stringgrid1.Cells[1,i+1]:=TableTr.FieldValues['No_Telp_
1'];
stringgrid1.Cells[2,i+1]:=TableTr.FieldValues['Judul
Buku 1'];
stringgrid1.Cells[3,i+1]:=TableTr.FieldValues['Judul
Buku 2'];

stringgrid1.Cells[4,i+1]:=TableTr.FieldValues['Tgl_Penge
mbalian'];
if TableTr.FieldValues['No_Telp_2']<>Null then

stringgrid1.Cells[5,i+1]:=TableTr.FieldValues['No_Telp_
2'];

ListBox1.Items.Add(StringGrid1.Cells[1,i+1]+' '+StringGr
id1.Cells[2,i+1]+' '+StringGrid1.Cells[3,i+1]+' '+StringGri
d1.Cells[4,i+1]+' '+StringGrid1.Cells[5,i+1]);
ck:=true;
end
else
begin
stringgrid1.Cells[5,i+1]:='-';
stringgrid1.Cells[1,i+1]:='-';
stringgrid1.Cells[2,i+1]:='-';
stringgrid1.Cells[3,i+1]:='-';
stringgrid1.Cells[4,i+1]:='-';
end;
TableTr.Next;inc(i);
until TableTr.Eof;
end;
if ck then begin no:=1;timer1.Enabled:=true;end;
end;

procedure TForm2.Timer1Timer(Sender: TObject);
begin
timer1.Enabled:=false;
edit1.Text:=stringgrid1.Cells[1,no];
if no=stringgrid1.RowCount then
begin
inc(no);
end
end

```

```

else
begin
edit1.Text:='selesai';
Button3.Click;
end
end;

procedure TForm2.Button3Click(Sender: TObject);
var
j,i,k,l,m:Integer;
s1:String;
begin
i:=0;
j:=ListBox1.Count;
repeat
Label1.Caption:=ListBox1.Items.Strings[i];
if Main.ComPortModem.Connected then
begin

Label2.Caption:="";Label3.Caption:="";Label4.Caption:="";
Label5.Caption:="";Label6.Caption:="";
l:=1;
s1:=Label1.Caption;
for k:=1 to length(s1) do
begin
if s1[k]=',' then
inc(l)
else
begin
case l of
1:Label2.Caption:=Label2.Caption+s1[k];
2:Label3.Caption:=Label3.Caption+s1[k];
3:Label4.Caption:=Label4.Caption+s1[k];
4:Label5.Caption:=Label5.Caption+s1[k];
5:Label6.Caption:=Label6.Caption+s1[k];
end
end
end;
Label7.Caption:=IntToStr(length(Label2.Caption));

Main.ComPortModem.WriteStr('AT+CMGS'+Label2.Ca
ption+#13);
sleep(100);

Main.ComPortModem.WriteStr('PEMBERITAHUAN!
Batas Peminjaman Buku yang Berjudul 1.'+
Label3.Caption+'2.'+Label4.Caption+' Jatuh Pada
Tanggal '+Label5.Caption+#26);
if Label6.Caption<>" then
begin
sleep(10000);

Main.ComPortModem.WriteStr('AT+CMGS'+Label6.Ca
ption+#13);
sleep(100);

Main.ComPortModem.WriteStr('PEMBERITAHUAN!
Batas Peminjaman Buku yang Berjudul 1.'+

```

```

Label3.Caption+'2.'+Label4.Caption+' Jatuh Pada
Tanggal '+Label5.Caption+#26);
    end
    end;
    sleep(10000);
    inc(i);
    until i>=j
end;

procedure TForm2.FormCreate(Sender: TObject);
begin
    EnumComPorts(combobox1.Items);
end;

end.

```

```

unit Unit3;

interface

uses
    Windows, Messages, SysUtils, Variants, Classes,
    Graphics, Controls, Forms,
    Dialogs, SUIButton, Grids, DBGrids, SUIDBCtrls,
    ExtCtrls, SUIForm,
    StdCtrls, QuickRpt, DB, ADODB, QRCtrls;

type
    TFormReport = class(TForm)
        suiForm1: TsuiForm;
        DBGridReport: TsuiDBGrid;
        ButtonExit: TsuiButton;
        Label1: TLabel;
        ButtonCetak: TsuiButton;
        ADOQuery1: TADOQuery;
        QuickRep1: TQuickRep;
        QRBand1: TQRBand;
        QRLabel1: TQRLabel;
        QRBand2: TQRBand;
        QRLabel3: TQRLabel;
        QRLabel4: TQRLabel;
        QRLabel5: TQRLabel;
        QRLabel6: TQRLabel;
        QRBand3: TQRBand;
        QRDBText1: TQRDBText;
        QRDBText2: TQRDBText;
        QRDBText3: TQRDBText;
        QRDBText4: TQRDBText;
        QRDBText5: TQRDBText;
        QRBand4: TQRBand;
        QRSysData1: TQRSysData;
        QRSysData2: TQRSysData;
        QRLabel8: TQRLabel;
        QRLabel9: TQRLabel;
        QRLabel10: TQRLabel;
        QRDBText6: TQRDBText;

```

```

QRDBText7: TQRDBText;
QRDBText8: TQRDBText;
QRLabel2: TQRLabel;
QRLabel11: TQRLabel;
QRLabel12: TQRLabel;
QRLabel14: TQRLabel;
QRLabel15: TQRLabel;
QRLabel7: TQRLabel;
    procedure ButtonExitClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure ButtonCetakClick(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;

var
    FormReport: TFormReport;

implementation
uses Unit1;
{$R *.dfm}

procedure TFormReport.ButtonExitClick(Sender:
TObject);
begin
    FormReport.Close;
end;

procedure TFormReport.FormCreate(Sender: TObject);
begin
    ADOQuery1.Connection:=Main.ADOConnection1;
    ADOQuery1.Active:=True;
end;

procedure TFormReport.ButtonCetakClick(Sender:
TObject);
begin
    with ADOQuery1 do begin
        Close;
        SQL.Clear;
        SQL.Add('select * from Transaksi');
        Open;
    end;
    QuickRep1.Preview;
end;

end.

```