

**RANCANG BANGUN APLIKASI *SMART CARD* UNTUK
INFORMASI MAHASISWA DAN PEMBAYARAN PARKIR
MAHASISWA
(Studi Kasus ITN Malang)**

SKRIPSI



**Disusun oleh :
YOHANES VIANEY RICKY NUGROHO
12.18.142**

**PROGRAM STUDI TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2016**

LEMBAR PERSETUJUAN

**RANCANG BANGUN APLIKASI *SMART CARD* UNTUK
INFORMASI MAHASISWA DAN PEMBAYARAN PARKIR
MAHASISWA**

SKRIPSI

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

Disusun Oleh :

YOHANES VIANEY RICKY NUGROHO

12.18.142

Diperiksa dan Disetujui,

Dosen Pembimbing I

Dosen Pembimbing II

Suryo Adi Wibowo, ST, MT.

NIP.P 1031000438

Febriana Santi W, S.Kom., M.Kom.

NIP.P. 1031000425

Mengetahui,

Program Studi Teknik Informatika S-1

Ketua

Joseph Dedy Irawan, ST, MT

NIP. 197404162005011002

**PROGRAM STUDI TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG**

2016

MILIK
BERPUSTAKAAN
ITN MALANG

LEMBAR KEASLIAN
PERNYATAAN KEASLIAN SKRIPSI

Sebagai mahasiswa Program Studi Teknik Informatika S-1 Fakultas Teknologi Industri Institut Teknologi Nasional Malang, yang bertanda tangan di bawah ini, saya:

Nama : Yohanes Vianey Ricky N

NIM : 12.18.142

Menyatakan dengan sesungguhnya bahwa skripsi saya dengan judul "*Rancang Bangun Aplikasi Smart Card Untuk Informasi Mahasiswa Dan Pembayaran Parkir Mahasiswa*" merupakan karya asli dan bukan merupakan duplikat dan mengutip seluruhnya karya orang lain. Apabila di kemudian hari, karya asli saya disinyalir bukan merupakan karya asli saya, maka saya akan bersedia menerima segala konsekuensi apa pun yang diberikan Program Studi Teknik Informatika S-1 Fakultas Teknologi Industri Institut Teknologi Nasional Malang.

Demikian surat pernyataan ini saya buat dengan sebenar-benarnya.

Malang, 25 Januari 2015

Yang membuat pernyataan



Yohanes Vianey Ricky N

NIM. 1218142

RANCANG BANGUN APLIKASI *SMART CARD* UNTUK INFORMASI MAHASISWA DAN PEMBAYARAN PARKIR MAHASISWA

Yohanes Vianey Ricky N (1218142)
Program Studi Teknik Informatika S-1
Institut Teknologi Nasional Malang
Email : vianeyricky@gmail.com

Dosca Pembimbing: 1. Suryo Adi Wibowo, ST, MT.
2. Febriana Santi W, S.Kom., M.Kom

Abstrak

Smart Card menjadi salah satu pilihan untuk memenuhi kebutuhan efisiensi bertransaksi. Tidak hanya transaksi dalam jumlah besar namun biaya kecil seperti retribusi parkir juga menjadi salah satu transaksi yang membutuhkan efisiensi. *Smart Card* merupakan sebuah teknologi yang tergolong unggulan yang banyak digunakan oleh pemerintah untuk menjadikannya sebagai jalan keluar sebuah masalah. Seperti halnya kartu sehat, kartu Indonesia pintar dll.

Institut Teknologi Nasional Malang merupakan salah satu institut yang mana memiliki jumlah mahasiswa yang cukup besar. Berada di pusat kota tak jarang ketika pagi hari terjadi kemacetan yang cukup panjang karena banyaknya mahasiswa yang memasuki area parkir yang letaknya tepat di pinggir jalan. Karenanya, teknologi *Smart Card* menjadi pilihan penulis untuk mengatasi masalah tersebut yang diharapkan dapat menambah efisiensi waktu bagi mahasiswa dan memberi kemudahan baik bagi mahasiswa sebagai pengguna serta petugas parkir sebagai admin. Selain untuk pembayaran retribusi parkir *Smart Card* juga berfungsi sebagai kartu mahasiswa yang di dalamnya terdapat informasi umum mengenai mahasiswa yang berguna bagi dosen, karyawan atau mahasiswa untuk pendataan mahasiswa secara cepat dan efektif karena dapat meminimalisir kesalahan penulisan informasi bila dilakukan secara manual.

Dalam Aplikasi ini menggunakan Reader dengan tipe ACS38 yang mana sangat kompatibel dengan 4 versi windows yang banyak digunakan saat ini, seperti Windows Xp, Windows 7, Windows 8.1 serta Windows 10. Selain itu Aplikasi ini dapat digunakan dengan mudah oleh para petugas parkir sehingga dapat menambah efisiensi waktu dalam pembayaran parkir.

Kata Kunci : *Smart Card, Parkir, Efisiensi*

KATA PENGANTAR

Puji Syukur Kehadirat Tuhan Yang Maha Esa, karena atas kasih dan karuniaNya yang telah diberikan selama ini sehingga penulis dapat menyelesaikan skripsi ini dengan Judul **“Rancang Bangun Aplikasi *Smart Card* Untuk Informasi Mahasiswa Dan Pembayaran Parkir Mahasiswa”**.

Penyusunan skripsi ini merupakan salah satu persyaratan untuk menyelesaikan program pendidikan Strata Satu (S-1) Teknik Informatika, Fakultas Teknologi Industri di Institut Teknologi Nasional Malang.

Pada kesempatan ini penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Bapak Dr. Ir. Lalu Mulyadi, MTA selaku Rektor Institut Teknologi Nasional Malang.
2. Bapak Ir. Anang Subardi, MT selaku Dekan Fakultas Teknologi Industri Institut Teknologi Nasional Malang.
3. Bapak Joseph Dedy Irawan, ST, MT selaku Ketua Program Studi Teknik Informatika S-1 Institut Teknologi Nasional Malang.
4. Bapak Suryo Adi Wibowo, ST, MT selaku Dosen pembimbing I.
5. Ibu Febriana Santi W, S.Kom., M.Kom selaku Dosen pembimbing II.
6. Bapak dan Ibu Dosen Teknik Informatika S-1 selaku pengamat dan penguji.
7. Kedua Orang Tua tercinta yaitu Bapak Yohanes Iswahyudi dan Ibu Tri Endang beserta saudara dan keluarga besar.
8. Keluarga besar Sanggar Blit'z ITN Malang beserta Pembina Bapak Heksa Galuh.
9. Semua teman-teman seperjuangan Teknik Informatika yang tidak bisa disebutkan satu-persatu yang telah memberikan semangat, dukungan, saran dan bantuan.
10. Serta semua pihak yang telah membantu dalam penyelesaian skripsi ini.

Penulis menyadari bahwa skripsi masih jauh dari sempurna, oleh karena itu penulis mengharapkan kritik dan saran dari pembaca. Sehingga skripsi ini bisa bermanfaat bagi para pembaca sekalian.

Malang, 2015

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PERSETUJUAN	ii
LEMBAR KEASLIAN	iii
ABSTRAK	iv
KATA PENGANTAR	v
DAFTAR ISI	vi
DAFTAR GAMBAR	ix
DAFTAR TABEL	xii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan	2
1.5 Metode Penelitian	2
1.6 Sistematika Penulisan	3
BAB II LANDASAN TEORI	5
2.1 Tinjauan Singkat ITN Malang	5
2.2 <i>Smart Card</i>	5
2.3 <i>Smart Card Reader</i>	6
BAB III ANALISA DAN PERANCANGAN SISTEM	11
3.1 Desain Sistem	11
3.2 <i>Flowchart</i>	11
3.2.1 Proses Pembayaran Parkir	11
3.2.2 Pelugas Parkir	12
3.2.3 <i>Admin</i>	13
3.2.4 <i>Super Admin</i>	14
3.3 Struktur Menu	15
3.4 Perancangan <i>Interface</i> (Antar Muka)	17
3.4.1 <i>Form Home</i>	17
3.4.2 <i>Form Login Super Admin</i>	18
3.4.3 <i>Form Login Admin</i>	18

3.4.4	<i>Form Login Petugas Parkir</i>	19
3.4.5	<i>Form Login Data Mahasiswa</i>	19
3.4.6	<i>Form Petugas Parkir</i>	20
3.4.7	<i>Form Admin</i>	20
3.4.8	<i>Form Data Parkir</i>	21
3.4.9	<i>Form Isi Ulang Saldo</i>	21
3.5	Perancangan <i>Database</i>	22
3.5.1	Tabel	22
3.5.2	Relasi Tabel	23
BAB IV IMPLEMENTASI DAN PENGUJIAN		24
4.1	Implementasi	24
4.1.1	<i>Form Utama Aplikasi</i>	24
4.1.2	<i>Form Login Super Admin</i>	25
4.1.3	<i>Form Login Admin</i>	25
4.1.4	<i>Form Login Petugas Parkir</i>	26
4.1.5	<i>Form Data Mahasiswa</i>	26
4.1.6	<i>Form Data Parkir</i>	27
4.1.7	<i>Form Isi Ulang Saldo</i>	27
4.2	Konfigurasi Reader ACS38	28
4.2.1	<i>Import Library Smart Card Tool Set Pro v3.4</i>	28
4.2.2	Konfigurasi <i>Library</i>	28
4.2.3	Konfigurasi <i>Smart Card 7816</i>	30
4.2.4	Tahap Pengolahan Data <i>Smart Card</i>	30
4.2.4.1	<i>Source Code</i> Konversi	30
4.2.4.2	Pembacaan dan Penulisan Data Pada <i>Smart Card</i>	31
4.3	Pengujian	32
4.3.1	Pengujian <i>Login dan Logout</i>	32
4.3.1.1	Pengujian <i>Form Super Admin Login</i>	32
4.3.1.2	Pengujian <i>Form Admin Login</i>	34
4.3.1.3	Pengujian <i>Form Petugas Parkir Login</i>	36
4.3.2	Pengujian CRUD (Create Read Update Delete)	38
4.3.2.1	Pengujian <i>Form Admin</i>	38

4.3.2.2 Pengujian <i>Form</i> Petugas Parkir -----	41
4.3.2.3 Pengujian <i>Form</i> Data Mahasiswa -----	43
4.3.2.4 Pengujian <i>Form</i> Data Parkir -----	46
4.3.2.5 Pengujian <i>Form</i> Isi Ulang Saldo -----	48
4.3.3 Pengujian <i>Reader</i> ACS38 -----	50
4.3.4 Pengujian <i>Smart Card</i> ISO7816 -----	51
4.3.5 Tabel Pengujian Fungsional -----	52
4.3.6 Pengujian <i>Reader</i> ACS38 Dengan Tabel Fungsional -----	53
4.3.7 Pengujian <i>Smart Card</i> Dengan Tabel Fungsional -----	54
4.3.8 Pengujian <i>User</i> -----	54
4.3.9 Pengujian Aplikasi -----	55
BAB V PENUTUP -----	56
5.1 Kesimpulan -----	56
5.2 Saran -----	56
DAFTAR PUSTAKA -----	57
LAMPIRAN -----	58

DAFTAR GAMBAR

Gambar 2.1 Institut Teknologi Nasional Malang	5
Gambar 2.2 <i>Smart Card</i>	5
Gambar 2.3 <i>Smart Card Reader</i>	6
Gambar 2.4 Standar PIN <i>Contact Smart Card</i>	7
Gambar 2.5 Bentuk <i>Contact Smart Card</i>	7
Gambar 2.6 Bentuk <i>Contactless Smart Card</i>	8
Gambar 2.7 Karakter Fisik dari Smart Card Berdasar Standar iso 7816	10
Gambar 3.1 Blok Diagram Sistem	11
Gambar 3.2 <i>Flowchart</i> Pembayaran Parkir	12
Gambar 3.3 <i>Flowchart</i> Petugas	13
Gambar 3.4 <i>Flowchart</i> Admin	14
Gambar 3.5 <i>Flowchart</i> Super Admin	15
Gambar 3.6 Struktur Menu Super Admin	16
Gambar 3.7 Struktur Menu Admin	16
Gambar 3.8 Struktur Menu Petugas Parkir	17
Gambar 3.9 <i>Form Home</i>	17
Gambar 3.10 Struktur Menu Admin	18
Gambar 3.11 <i>Form</i> Login Admin	18
Gambar 3.12 <i>Form</i> Login Petugas Parkir	19
Gambar 3.13 <i>Form</i> Penambahan Data Mahasiswa	19
Gambar 3.14 <i>Form</i> Penambahan Petugas	20
Gambar 3.15 <i>Form</i> Penambahan Admin	20
Gambar 3.16 <i>Form</i> Aplikasi Parkir	21
Gambar 3.17 <i>Form</i> Isi ulang saldo	21
Gambar 3.18 Relasi Tabel	23
Gambar 4.1 Tampilan Utama <i>Home</i>	24
Gambar 4.2 Tampilan <i>Form Login Super Admin</i>	25
Gambar 4.3 Tampilan <i>Form Login Admin</i>	25
Gambar 4.4 Tampilan <i>Form Login</i> Petugas Parkir	26

Gambar 4.5 Tampilan <i>Form</i> Data Mahasiswa -----	26
Gambar 4.6 Tampilan <i>Form</i> Data Parkir -----	27
Gambar 4.7 Tampilan <i>Form</i> Isi Ulang Saldo -----	27
Gambar 4.8 Import <i>library smart card tool set pro</i> -----	28
Gambar 4.9 Pengaturan <i>Library ActiveX Smart Card Tool Set Pro</i> -----	29
Gambar 4.10 Konfigurasi <i>reader</i> -----	29
Gambar 4.11 Konfigurasi <i>smart card</i> pada <i>Event OnCardDetected</i> -----	30
Gambar 4.12 <i>SourceCode</i> pada <i>ActiveX</i> -----	30
Gambar 4.13 <i>SourceCode Hex to Integer</i> -----	31
Gambar 4.14 <i>SourceCode Integer to Hex</i> -----	31
Gambar 4.15 Tampilan <i>Login</i> Awal -----	31
Gambar 4.16 Tampilan <i>Login</i> Berhasil -----	31
Gambar 4.17 Tampilan <i>Login</i> Gagal -----	31
Gambar 4.18 Tampilan Awal Setelah <i>Login</i> -----	32
Gambar 4.19 Tampilan <i>Logout Super Admin</i> -----	32
Gambar 4.20 Tampilan <i>Login</i> Awal -----	33
Gambar 4.21 Tampilan <i>Login</i> Berhasil -----	33
Gambar 4.22 Tampilan <i>Login</i> Gagal -----	34
Gambar 4.23 Tampilan Awal Setelah <i>Login</i> -----	34
Gambar 4.24 Tampilan <i>Logout Super Admin</i> -----	34
Gambar 4.25 Tampilan <i>Login</i> Awal -----	35
Gambar 4.26 Tampilan <i>Login</i> Berhasil -----	35
Gambar 4.27 Tampilan <i>Login</i> Gagal -----	36
Gambar 4.28 Tampilan Awal Setelah <i>Login</i> -----	36
Gambar 4.29 Tampilan <i>Logout Super Admin</i> -----	37
Gambar 4.30 Tampilan Awal <i>Form Admin</i> -----	37
Gambar 4.31 Tampilan Data Berhasil Dimasukkan-----	38
Gambar 4.32 Tampilan Data Berhasil Diedit -----	38
Gambar 4.33 Tampilan <i>Prompt</i> Hapus Data -----	39
Gambar 4.34 Tampilan Data Berhasil Dihapus-----	39

Gambar 4.35 Tampilan Awal <i>Form</i> Petugas Parkir-----	40
Gambar 4.36 Tampilan Data Berhasil Dimasukkan-----	40
Gambar 4.37 Tampilan Data Berhasil <i>Diedit</i> -----	41
Gambar 4.38 Tampilan <i>Prompt</i> Hapus Data -----	41
Gambar 4.39 Tampilan Data Berhasil Dihapus -----	42
Gambar 4.40 Tampilan Awal <i>Form</i> Data Mahasiswa -----	42
Gambar 4.41 Tampilan Data Berhasil Dimasukkan-----	43
Gambar 4.42 Tampilan Data Berhasil <i>Diedit</i> -----	43
Gambar 4.43 Tampilan <i>Prompt</i> Hapus Data -----	44
Gambar 4.44 Tampilan Data Berhasil Dihapus-----	44
Gambar 4.45 Tampilan Awal <i>Form</i> Data Parkir-----	45
Gambar 4.46 Proses Memasukkan Kartu (Masuk Parkir)-----	45
Gambar 4.47 Pencatatan Nim, Nomor Kendaraan Tanggal Masuk dan Jam Masuk (<i>History</i> Terbawah) -----	46
Gambar 4.48 Proses Memasukkan Kartu (Keluar Parkir)-----	46
Gambar 4.49 Pencatatan tanggal keluar, jam keluar serta pengurangan saldo (<i>History</i> Terbawah)-----	46
Gambar 4.50 <i>Prompt Delete Log</i> -----	47
Gambar 4.51 Tampilan Awal <i>Form</i> Isi Ulang Saldo-----	47
Gambar 4.52 Pemilihan Nim dan Jumlah Saldo-----	48
Gambar 4.53 Pengisian Saldo Berhasil -----	48
Gambar 4.54 Proses Memasukkan <i>Reader</i> Kedalam USB 3.0 -----	49
Gambar 4.55 Koneksi <i>Reader</i> Berhasil -----	49
Gambar 4.56 Diskoneksi <i>Reader</i> Berhasil -----	50
Gambar 4.57 Deteksi Kartu Berhasil-----	50

DAFTAR TABEL

Tabel 3.1. Tabel Mahasiswa	22
Tabel 3.2. Tabel Data Parkir	22
Tabel 3.3. Tabel Saldo	23
Tabel 4.1 Perintah APDU <i>Smart Card</i>	31
Tabel 4.2 Tabel Pengujian Fungsional	52
Tabel 4.3 Tabel Fungsional <i>Reader ACS38</i>	54
Tabel 4.4 Tabel Fungsional <i>Smart Card</i>	54
Tabel 4.5 Tabel Pengujian <i>User</i>	55
Tabel 4.6 Tabel Pengujian Aplikasi	55

BAB I

PENDAHULUAN

1.1 Latar Belakang

Di era modernisasi ini efisiensi menjadi salah satu hal yang diutamakan dan dapat dikatakan menjadi kebutuhan primer bagi sebagian orang, tak terkecuali mahasiswa. Sebagai seorang mahasiswa waktu menjadi sebuah barang yang berharga dan tidak dapat tergantikan. Apalagi saat ini kemudahan dalam bertransaksi sudah menjadi tuntutan.

Smart Card menjadi salah satu pilihan untuk memenuhi kebutuhan efisiensi bertransaksi. Tidak hanya transaksi dalam jumlah besar namun biaya kecil seperti retribusi parkir juga menjadi salah satu transaksi yang membutuhkan efisiensi. *Smart Card* merupakan sebuah teknologi yang tergolong unggulan yang banyak digunakan oleh pemerintah untuk menjadikannya sebagai jalan keluar sebuah masalah. Seperti halnya kartu sehat, kartu Indonesia pintar dll. (Swastika dkk, 2010)

Di Indonesia sendiri sudah terdapat 2 daerah yang sudah menerapkan kartu pintar atau *Smart Card* salah satunya adalah Bandung. Seperti yang dikutip dari **BANDUNG, KOMPAS.com** (Senin, 14 Desember 2015 | 15:13 WIB) — Pemerintah Kota Bandung bersama Bank Indonesia Perwakilan Provinsi Jawa Barat dan sejumlah bank berkolaborasi dalam peluncuran kartu pintar, Bandung *SmartCard*. Kepala Perwakilan Kantor Bank Indonesia Provinsi Jawa Barat Rosmaya mengatakan, warga Bandung bisa mendapatkan kartu tersebut dengan membelinya di gerai-gerai perbankan yang melakukan kerja sama dan PT Telekomunikasi Indonesia sebagai penerbit dengan harga Rp 20.000, belum termasuk saldo. (Ramdhani,2015)

Tidak hanya bandung Jamsostek Jakarta juga menggunakan *Smart Card* seperti yang ditulis pada **JAKARTA, KOMPAS.com** (Kamis, 12 Desember 2013 | 19:35 WIB) - Menjelang bertransformasi menjadi Badan Penyelenggara Jaminan Sosial (BPJS) Ketenagakerjaan, PT Jamsostek (Persero) memperkenalkan *smart card* sebagai pengganti kartu peserta Jamsostek. (Setiawan, 2013)

Institut Teknologi Nasional Malang merupakan salah satu institut yang mana memiliki jumlah mahasiswa yang cukup besar. Berada di pusat kota tak jarang

ketika pagi hari terjadi kemacetan yang cukup panjang karena banyaknya mahasiswa yang memasuki area parkir yang letaknya tepat di pinggir jalan.

Karenanya, teknologi *Smart Card* menjadi pilihan penulis untuk mengatasi masalah tersebut yang diharapkan dapat menambah efisiensi waktu bagi mahasiswa dan memberi kemudahan baik bagi mahasiswa sebagai pengguna serta petugas parkir sebagai *admin*.

Selain untuk pembayaran retribusi parkir *Smart Card* juga berfungsi sebagai kartu mahasiswa yang di dalamnya terdapat informasi umum mengenai mahasiswa yang berguna bagi dosen, karyawan atau mahasiswa untuk pendataan mahasiswa secara cepat dan efektif karena dapat meminimalisir kesalahan penulisan informasi bila dilakukan secara manual.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dikemukakan sebelumnya, maka perumusan masalah yang dapat dibuat adalah sebagai berikut :

1. Bagaimana cara membuat aplikasi *Smart Card* untuk pembayaran retribusi parkir di Institut Teknologi Nasional Malang ?
2. Bagaimana cara membuat database yang terkoneksi dengan ACS38 sebagai *reader* ?

1.3 Tujuan

Adapun tujuan yang ingin dicapai dari pembuatan aplikasi ini adalah:

1. Merancang aplikasi *Smart Card* untuk mahasiswa dalam pembayaran retribusi parkir.
2. Membuat sistem *database* yang terkoneksi dengan ACS38 sebagai *reader*.

1.4 Batasan Masalah

Dalam penyusunan skripsi agar menjadi sistematis dan mudah dimengerti, maka akan diterapkan beberapa batasan masalah. Batasan-batasan masalah itu antara lain :

1. Aplikasi *Smart Card* hanya digunakan sebagai alat pembayaran.
2. Informasi yang terdapat pada *Smart Card* merupakan data mahasiswa berupa nama, alamat, kontak, jurusan dan nim.

1.5 Metode Penelitian

Adapun Metode Penelitian yang digunakan adalah sebagai berikut:

A. Studi Literatur

Pada tahap ini dipelajari literatur dan perencanaan serta konsep awal untuk membentuk program yang akan dibuat yaitu didapat dari referensi buku, internet, maupun sumber-sumber yang lain.

B. Pengumpulan Data dan Analisis

Pada tahap ini adalah proses pengumpulan data yang dibutuhkan untuk pembuatan program, serta melakukan analisa atau pengamatan pada data yang sudah terkumpul untuk selanjutnya diolah lebih lanjut.

C. Analisis dan Perancangan System

Setelah selesai pada tahap pengumpulan data dan analisis maka tahap selanjutnya adalah melakukan analisa dan perancangan sistem. Pada tahap ini adalah proses perancangan dari system yang akan dibuat untuk selanjutnya akan diproses lebih lanjut.

D. Pembuatan Program

Setelah tahap perancangan sistem maka tahap selanjutnya adalah pembuatan program. Pada tahap ini sistem yang sebelumnya telah dibuat akan diterapkan pada program yang akan dibuat. Pembuatan program ini menggunakan pemrograman PHP dan menggunakan *certainty factor* sebagai metode penalaran pada program ini.

E. Uji Coba Program

Setelah program selesai dibuat maka dilakukan pengujian program untuk mengetahui apakah program tersebut telah bekerja dengan benar dan sesuai dengan sistem yang dibuat.

F. Pembuatan Kesimpulan

Pada tahap akhir ini adalah pembuatan kesimpulan atau ringkasan dari skripsi ini dan kesimpulan tentang program yang telah dibuat.

1.6 Sistematika Penulisan

Untuk mempermudah memahami pembahasan pada penulisan skripsi ini, maka sistematika penulisan diperoleh sebagai berikut :

- BAB I** : Pendahuluan
Berisi latar belakang, perumusan masalah, batasan masalah, tujuan penelitian, metode penelitian, dan sistematia penulisan.
- BAB II** : Landasan Teori
-

Berisi dasar teori mengenai permasalahan yang berhubungan dengan penelitian ini.

BAB III : Analisa dan Perancangan Sistem

Berisi mengenai perancangan sistem pakar diagnosis penyakit pada gigi berbasis web yang akan di buat.

BAB VI : Implementasi dan Pengujian

Berisi implementasi terhadap proses sistem pakar diagnosis penyakit pada gigi berbasis web yang akan di buat, serta melakukan pengujian terhadap aplikasi tersebut.

BAB V : Kesimpulan dan Saran

Berisi kesimpulan dari hasil penelitian yang dilakukan dan saran yang dapat digunakan untuk bahan pengembangan penelitian berikutnya.

BAB II

LANDASAN TEORI

2.1 Tinjauan Singkat Institut Teknologi Nasional Malang



Gambar 2.1 Institut Teknologi Nasional Malang

Di kota Malang terdapat cukup banyak universitas maupun perguruan tinggi, salah satunya adalah Institut Teknologi Nasional Malang. Institut Teknologi Nasional Malang atau ITN Malang merupakan sebuah Institut yang memiliki cukup banyak mahasiswa. Seperti hasil survey penulis (*Parkir-01/survey/pen/12/2015*) pada lampiran 1 tidak kurang dari 500 mahasiswa perhari keluar masuk area parkir di dalamnya, sehingga pada jam – jam tertentu membuat antrian yang cukup panjang saat jam padat di kampus ini. Selain itu dengan jumlah mahasiswa yang cukup banyak, bukan tidak mungkin adanya kesalahan masukan data mengenai nomor kendaraan bila dilakukan secara manual.

2.2 *Smart Card*



Gambar 2.2 *Smart Card*

Smart Card memiliki chip kecil keemasan pada kartu, saat dibaca oleh *reader*, chip tersebut melakukan kontak dengan konektor yang dapat membaca informasi dari chip, dan dapat menuliskan informasi kembali ke dalam *chip*. Pada *Smart Card*, beberapa *standard ISO* telah dikeluarkan untuk mendefinisikan bentuk fisik, posisi,

karakteristik, protokol, *format* perintah yang dikirim dan respon yang dikembalikan, ketahanan kartu, hingga fungsinya. Kartu ini sendiri tidak memiliki baterai sebagai sumber tenaga, karena energi yang dibutuhkan akan dihasilkan oleh *card reader*, yang digunakan sebagai media komunikasi antara *Smart Card* dan *host* (misalnya komputer). Aplikasi yang melakukan proses dapat di letakkan pada host / komputer, bersamaan dengan *database* atau *tools* yang diperlukan oleh aplikasi. (Swastika dkk, 2010)

2.3 *Smart Card reader*



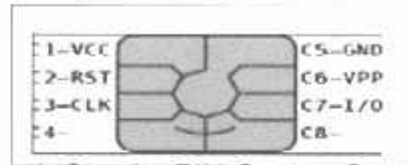
Gambar 2.3 *Smart Card Reader*

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*). (Swastika dkk, 2010)

A. *Contact Smart card*

Contact smart card memiliki sebuah chip emas yang berukuran sekitar 0,5 inci di bagian depan, tidak seperti kartu kredit yang memiliki magnetic strip di bagian belakang. *Contact Smart card* membutuhkan aplikasi *smart card reader* untuk membaca dan menulis data dari dan ke dalam *chip* tersebut. (Swastika dkk, 2010)

Standar *PIN* koneksi berdasarkan *ISO7816* seperti pada Gambar 2.4 dan Gambar 2.5:



Gambar 2.4. Standar *PIN* *Contact Smart Card*



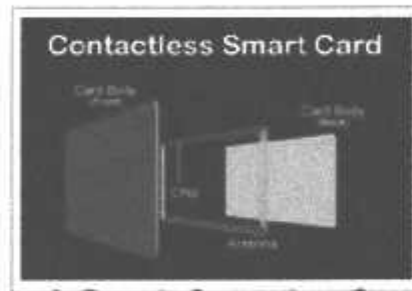
Gambar 2.5. Bentuk *Contact Smart Card*

B. *Contactless Smart card*

Contactless Smart card tampak seperti kartu kredit plastik dengan *chip computer* dan *antenna coil* di dalamnya seperti pada Gambar 2.6. *Contactless smart card* dapat ditulis dan dibaca hanya dengan didekatkan pada antenna luar. *Contactless Smart card* digunakan bila membutuhkan transaksi yang harus diproses dengan cepat. Dua kategori tambahan lainnya merupakan turunan dari kedua tipe yang telah dijelaskan sebelumnya. Kedua kategori tersebut adalah *Combi Card* dan *Hybrid Card*. *Hybrid card* memiliki dua chips, masing-masing merepresentasikan antarmuka *contact* dan *contactless*. Kedua chip tersebut tidak saling berhubungan, tetapi untuk sebagian besar aplikasi digunakan secara bersamaan dalam melayani kebutuhan *consumer* dan *card issuer*. Berbeda dengan *Hybrid Card*, *Combi card* hanya memiliki sebuah *chip* yang merepresentasikan antarmuka *contact* dan *contactless*. (Swastika dkk, 2010)

Chip yang digunakan pada kedua jenis kategori tersebut di atas dapat dikategorikan ke dalam dua bagian, yaitu: *chip microprocessor* dan *chip memory*.

Chip memory dapat dilihat sebagai *floppy disk* kecil dengan layanan keamanan (optional). *Memory card* dapat menyimpan 103 hingga 16000 bits data. *Memory card* lebih murah bila dibandingkan dengan *chip microprocessor*, hanya saja fasilitas keamanan yang dimiliki pun lebih sedikit. Keamanan *memory card* bergantung pada keamanan yang diberikan *card reader* saat pemrosesan data. *Chip microprocessor* dapat menambahkan, menghapus, ataupun memanipulasi informasi yang tersimpan dalam memory. *Chip microprocessor* dapat dianggap sebagai miniature computer dengan input/output port, sistem operasi, dan hard disk. (Swastika dkk, 2010)



Gambar 2.6. Bentuk *Contactless Smart Card*

Standar Internasional untuk protokol yang paling banyak digunakan dalam teknologi *smart card* ini adalah *ISO 7816*. Meskipun demikian, terdapat pula beberapa standar lain yang digunakan. *Smart card* menyimpan dan memproses informasi melalui rangkaian elektronik yang ada di dalam silikon yang dalam substrat plastik dari *body*-nya. Terdapat dua jenis *smart card* yang paling banyak digunakan yaitu *intelligent smart card* yang terdiri atas sebuah mikroprosesor dan mampu membaca, menulis, dan menghitung, seperti halnya sebuah mikrokomputer. Serta *memory card*, tidak memiliki mikroprosesor dan hanya digunakan untuk menyimpan informasi saja. *memory card* menggunakan *security logic* untuk mengatur akses dari memori. Meskipun demikian, sesungguhnya terdapat lima buah jenis *smart card* yang ada saat ini yaitu :

1. *Memory Card*

Merupakan sebuah alat (card) yang berfungsi sebagai tempat penyimpanan data digital (seperti gambar, audio dan video) pada sebuah gadget seperti kamera digital, PDA dan *Handphone*.

2. *Processor Card*

Prosesor Card adalah komponen yang mengeksekusi perhitungan kompleks yang memungkinkan komputer untuk bisa digunakan menjelajah internet, memutar lagu di iTunes, dan menjalankan sistem operasi.

3. *Electronic Purse Card*

Merupakan jenis *SmartCard* yang digunakan sebagai alat transaksi baik online maupun debit yang biasanya digunakan bank – bank untuk melakukan transaksi.

4. *Security Card*

Security Card digunakan untuk kebutuhan – kebutuhan yang memerlukan keamanan seperti *login* akun tertentu , maupun keperluan *security* lainnya.

5. *Java Card*

Java Card utamanya digunakan untuk aplikasi-aplikasi pada *smart card*, yang sederhana wujudnya seperti SIM card pada *handphone*.

Semua *smart card* berisi tiga tipe memori yaitu :

1. *Persistent Non-Mutable Memory*

Persistent Non-Mutable Memory atau biasa disebut ROM (*Read Only-Memory*) merupakan jenis memori yang tidak memungkinkan penggunaanya untuk mengubah data yang ada didalamnya.

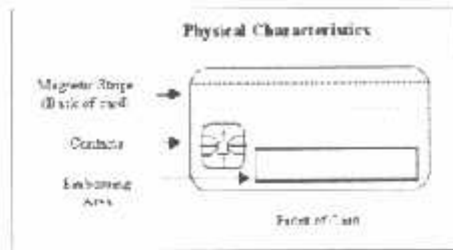
2. *Persistent Mutable Memory*

Persistent Mutable Memory atau EEPROM (*Electronical Erasable Programmable Read Only Memory*), seperti ROM, dapat menjaga isi data walaupun daya tidak ada. Perbedaannya adalah isi memori ini dapat dimodifikasi selama penggunaan kartu yang normal.

3. *Non-Persistent Mutable Memory*

Non-Persistent Mutable Memory atau RAM (*Random Acces Memory*) digunakan sebagai ruang bekerja yang bertempo (*temporary*) untuk menyimpan dan memodifikasi data. RAM adalah memori *nonpersistent*, sehingga isi informasi tidak dijaga ketika daya tidak ada. ROM, EEPROM, dan RAM merupakan memori yang paling banyak digunakan untuk ketiga tipe memori tersebut .

Karakter fisik dari *smart card* berdasar standar ISO 7816 seperti pada Gambar 2.7:



Gambar 2.7. Karakter fisik dari *smart card* berdasar standar ISO 7816.

Normalnya, sebuah *smart card* tidak memiliki *power supply*, *display*, atau *keyboard*. *Smart card* berinteraksi dengan dunia luar menggunakan *serial communication interface* via delapan *contact point* yang ada. Sebuah *smart card* dimasukkan ke dalam Card Acceptance Device (CAD), yang memungkinkan untuk berhubungan dengan komputer lain. Bentuk lain dari penggunaan *Card Acceptance Device* adalah terminal, *reader*, dan *IFD (interface device)*. Perangkat-perangkat tersebut menyediakan berbagai fungsi dasar yang sama, diantaranya untuk *supply smart card* dengan *power* dan untuk menghasilkan koneksi untuk pertukaran data.

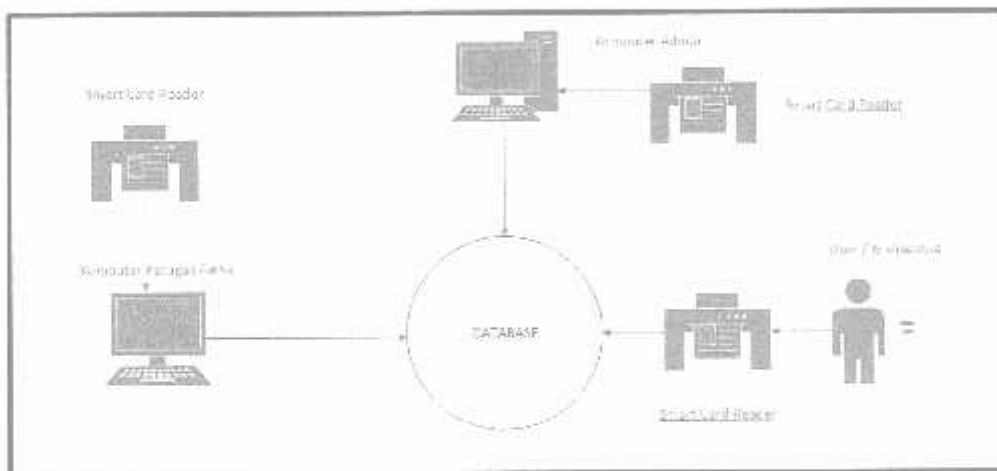
Ketika dua buah komputer berkomunikasi satu dengan yang lain, akan terjadi pertukaran paket data yang disusun berdasarkan sekumpulan protokol tertentu. Demikian pula *smart card*, berkomunikasi dengan dunia luar menggunakan paket data tersendiri yang dinamakan *APDU (Application Protocol Data Units)*. *APDU* berisi baik perintah-perintah maupun pesan respon. Dalam *smart card* tersebut, digunakan model *masterslave* dimana *smart card* sendiri berperan secara pasif sebagai *slave*. *smart card* selalu menunggu perintah *APDU* dari sebuah terminal. Selanjutnya perintah tersebut dieksekusi dan dikembalikan ke terminal sebagai respon *APDU*. Perintah *APDU* dan respon *APDU* tersebut dipertukarkan secara antara *card* dan terminal. Untuk membangun sebuah aplikasi *smart card* diperlukan beberapa komponen yaitu : *smart card reader*, *software* untuk berkomunikasi dengan *reader* dan *card* dipasang pada *reader*; dan *smart card* dan *smart card hardware*. (Swastika dkk, 2010)

BAB III

ANALISA DAN PERANCANGAN SISTEM

3.1 Desain sistem

Pada Gambar 3.1 merupakan blok diagram sistem yang akan dibuat dimana aplikasi komputer *admin* digunakan untuk pendataan, pengisian saldo, dan pengecekan data, sedangkan *Smart Card* digunakan untuk penyimpanan data mahasiswa beserta saldo dan *Smart Card Reader* digunakan untuk membaca data yang terdapat pada *Smart Card*.



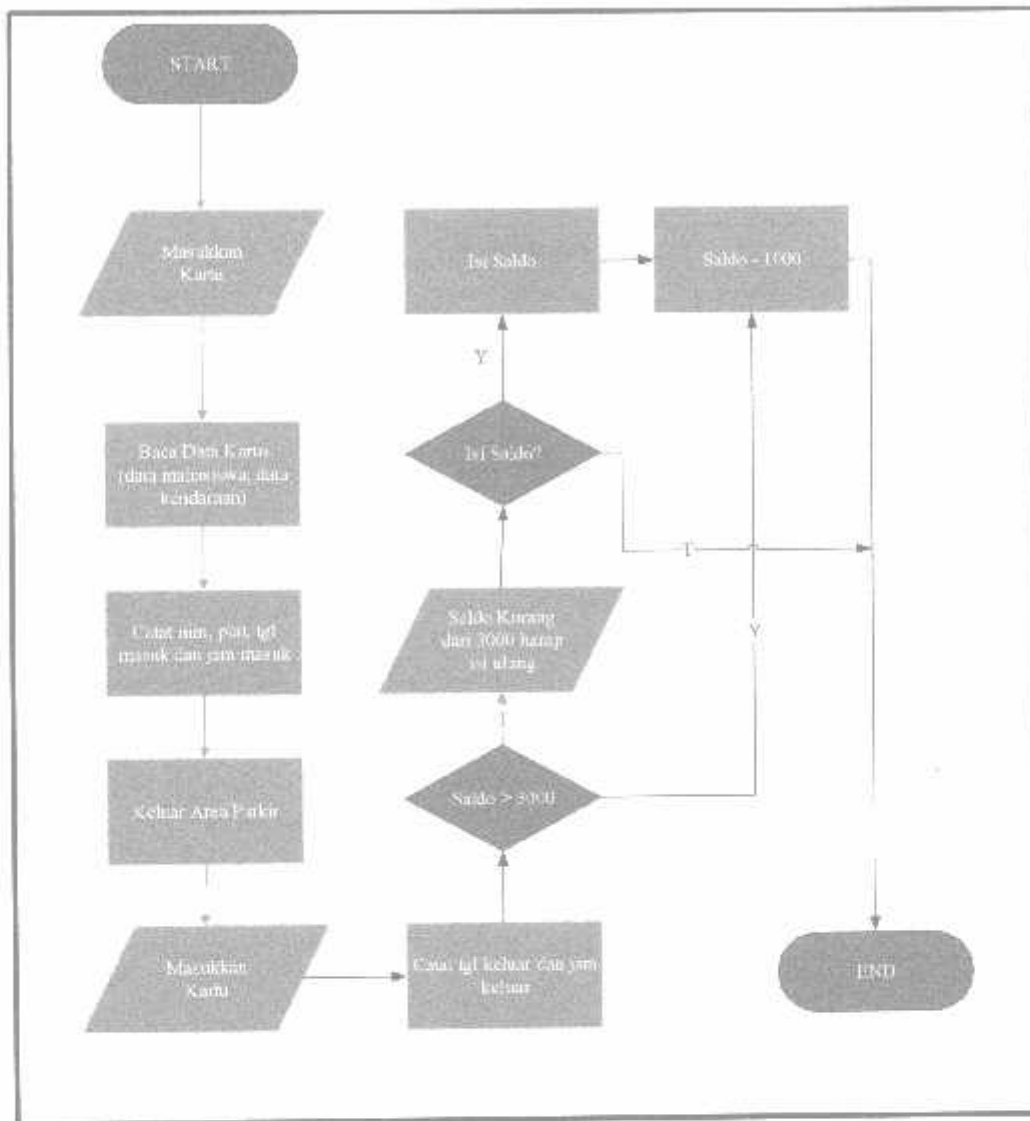
Gambar 3.1 Blok Diagram Sistem

3.2 Flowchart

Flowchart adalah bentuk Gambar/diagram yang mempunyai aliran satu atau dua arah secara *sekuensial*. *Flowchart* digunakan untuk merepresentasikan maupun mendesain program. Oleh karena itu *flowchart* harus bisa merepresentasikan komponen-komponen dalam bahasa pemrograman. (Adelia,2011)

3.2.1 Proses Pembayaran Parkir

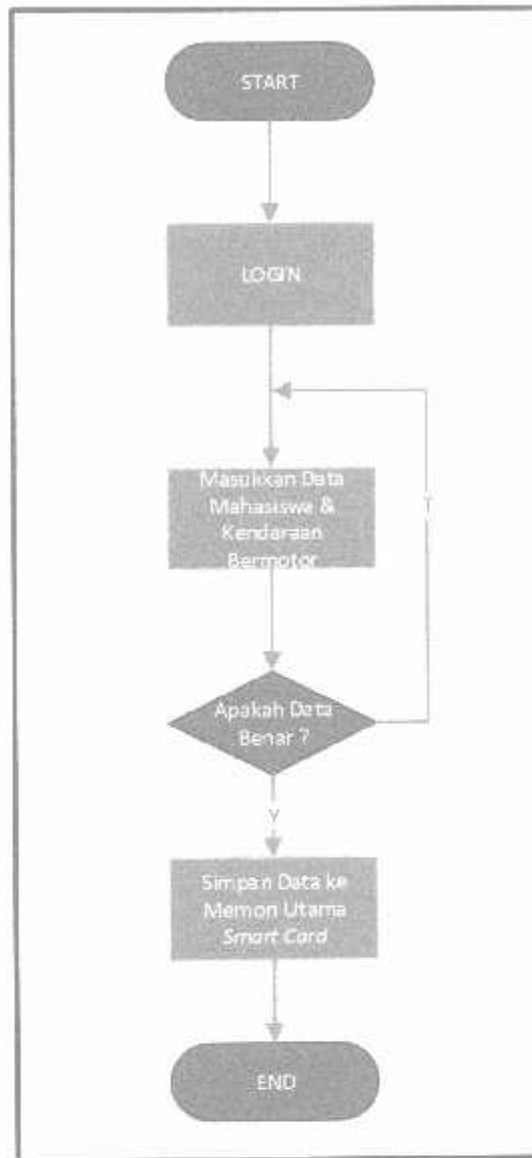
Untuk proses pembayaran parkir ketika masuk area parkir petugas memasukkan kartu milik mahasiswa kemudian pembacaan data kartu oleh program dan mencatat nim, plat nomor, tanggal masuk serta jam masuk area parkir. Pada saat keluar area parkir kartu di *inputkan* kembali kemudian dibaca oleh program dicatat tanggal keluar serta jam keluar dan saldo dikurangi 1000 rupiah. Berikut *flowchart* yang ditunjukkan pada Gambar 3.2.



Gambar 3.2 Flowchart Pembayaran Parkir

3.2.2 Petugas Parkir

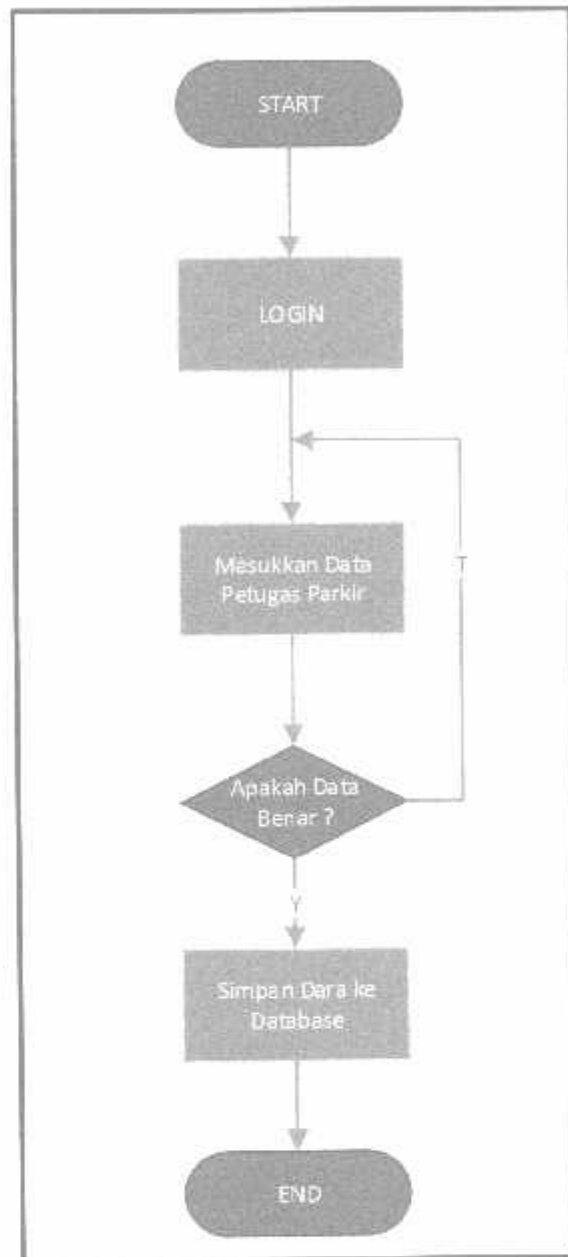
Pada Gambar 3.3 Ini menjelaskan tentang sistem petugas parkir. Dimana petugas parkir melakukan fungsi *login* yang digunakan untuk menginputkan data mahasiswa serta kendaraan bermotor yang dipakai .



Gambar 3.3 *Flowchart* Petugas

3.2.3 Admin

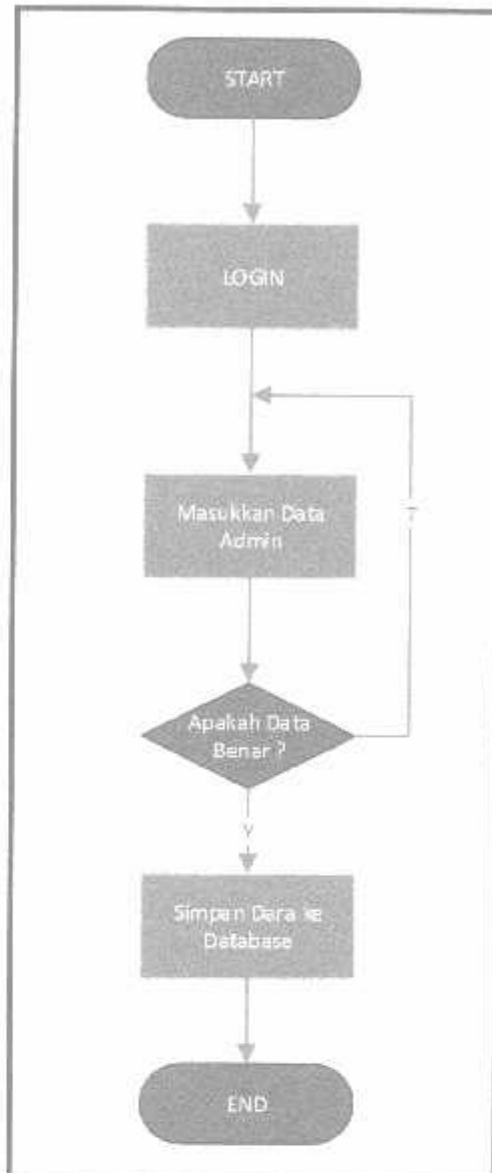
Pada Gambar 3.4 Ini menjelaskan tentang sistem *admin* dimana *Admin* memberi hak akses kepada petugas parkir guna kepentingan *Login* aplikasi, *Admin* menyertakan nama, *username*, serta *password* yang nantinya digunakan petugas parkir untuk *login*. *Admin* melakukan *login* kemudian memasukkan data petugas jika benar maka langsung disimpan pada *database*.



Gambar 3.4 Flowchart Admin

3.2.4 Super Admin

Super Admin bertugas untuk memberi hak akses kepada *Admin* guna fungsi *login* pada aplikasi, *Super Admin* menyertakan nama, *username* serta *password* yang nantinya digunakan *admin* untuk *login* pada aplikasi. *Super Admin* melakukan *login* kemudian memasukkan data *Admin* jika benar maka langsung disimpan pada *database*. Seperti yang ditunjukkan pada Gambar 3.5.



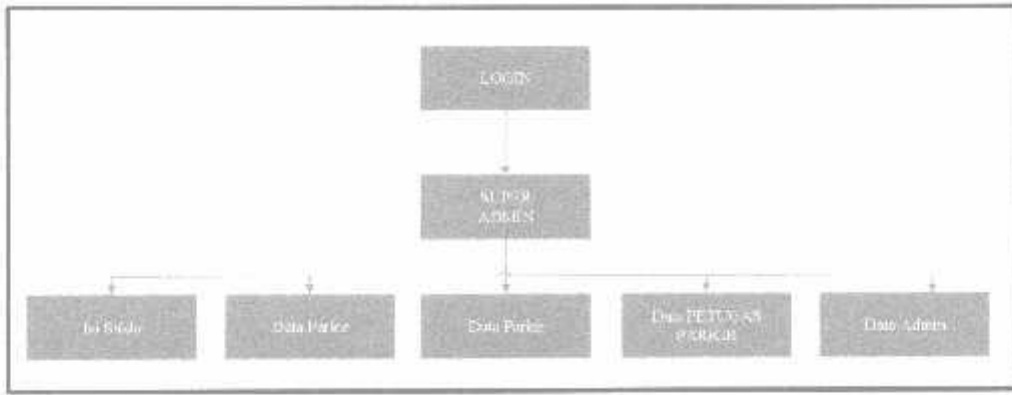
Gambar 3.5 Flowchart Super Admin

3.3 Struktur Menu

Dalam aplikasi ini terdapat menu-menu pada tiap hak aksesnya (*Super Admin*, *Admin*, *Petugas Parkir*).

1. Struktur Menu *Super Admin*

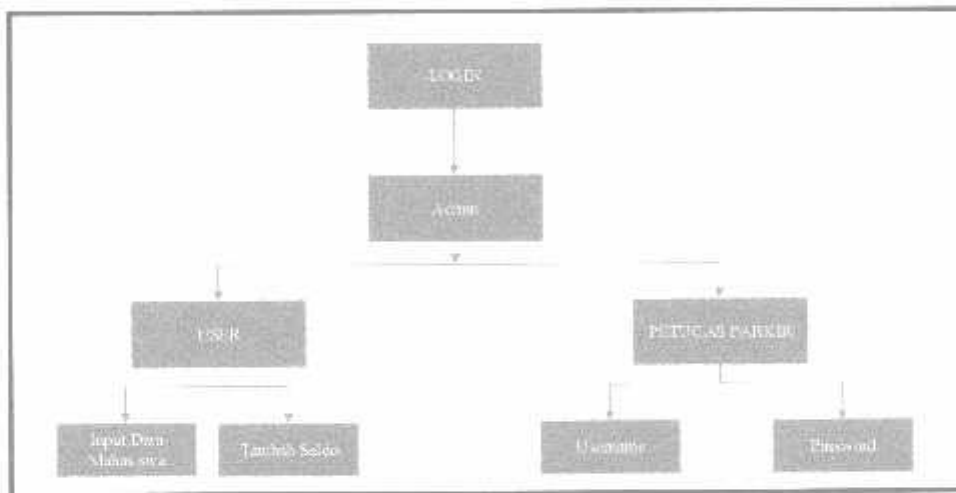
Seluruh Menu akan muncul jika masuk sebagai *Super Admin*. Yakni menu Isi Saldo, Data Parkir, Data Mahasiswa Data Petugas Parkir serta Data *Admin* seperti pada Gambar 3.6.



Gambar 3.6 Struktur Menu *Super Admin*

2. Struktur Menu *Admin*

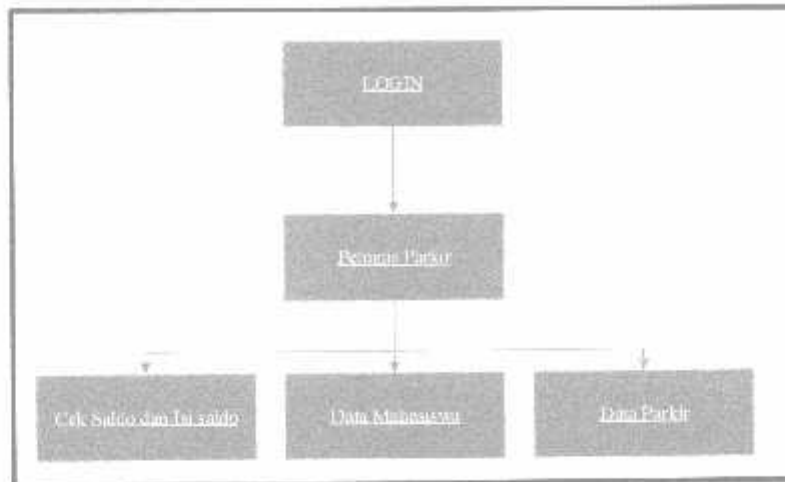
Terdapat beberapa menu jika masuk sebagai *Admin*. Diantaranya menu Isi Saldo, Data Parkir, Data Mahasiswa, Data Petugas Parkir seperti pada Gambar 3.7.



Gambar 3.7 Struktur Menu *Admin*

3. Struktur Menu Petugas Parkir

Terdapat beberapa menu jika masuk sebagai Petugas Parkir. Yakni menu Isi Saldo, Data Parkir serta Data Mahasiswa seperti pada Gambar 3.8.



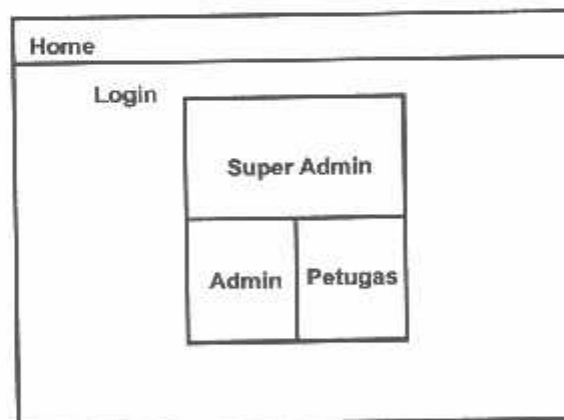
Gambar 3.8 Struktur Menu Petugas Parkir

3.4 Perancangan *Interface* (Antar muka)

Dalam pembuatan sistem aplikasi ini penulis akan memberikan gambaran antar muka aplikasi ke dalam sebuah gambar *prototype* desain antar muka pada tiap hak akses yang berinteraksi aktif dengan sistem aplikasi, adalah sebagai berikut:

3.4.1 *Form Home*

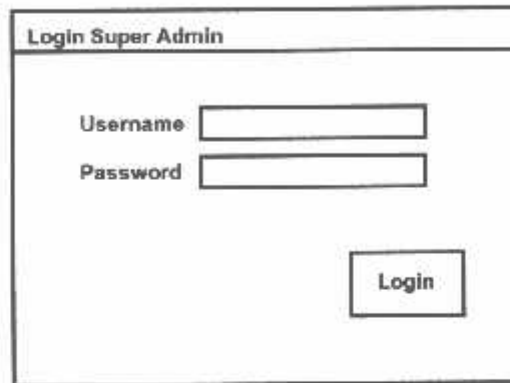
Merupakan tampilan awal ketika aplikasi pertama kali dibuka secara garis besar *form home* merupakan fitur untuk *login* dari berbagai tiga hak akses yang ada seperti *login Super Admin*, *Admin* serta petugas parkir yang dapat diakses dengan tombol. Seperti yang dilihat pada Gambar 3.9.



Gambar 3.9 *Form Home*.

3.4.2 Form Login SUPER ADMIN

Tampilan *Form Login Super Admin* berisi *username* dan *password* yang di atur secara *default* untuk satu orang yang berguna untuk mengakses seluruh fitur yang ada dalam aplikasi serta memberi hak akses pada *admin* secara khusus dan akses aplikasi secara umum dapat dilihat pada Gambar 3.10.

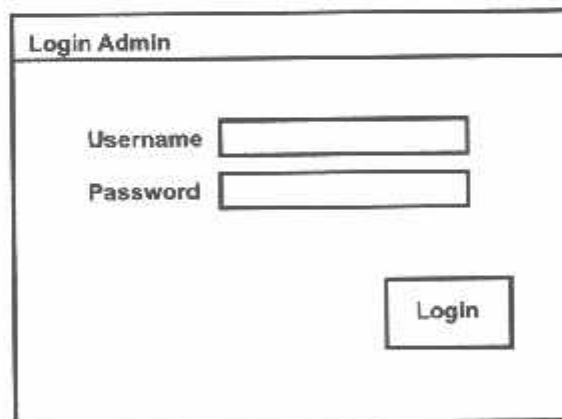


The diagram shows a rectangular box representing the 'Login Super Admin' form. At the top left, the text 'Login Super Admin' is written. Below this, there are two input fields: the first is labeled 'Username' and the second is labeled 'Password'. To the right of these fields is a rectangular button labeled 'Login'.

Gambar 3.10 Struktur Menu Admin

3.4.3 Form Login Admin

Hampir sama seperti *form Super Admin* tampilan *Form Login Admin* berisi *username* dan *password* yang nantinya akan di *direct* ke *form* petugas parkir untuk penambahan user baru. Untuk perancangan *form* dapat dilihat pada Gambar 3.11.



The diagram shows a rectangular box representing the 'Login Admin' form. At the top left, the text 'Login Admin' is written. Below this, there are two input fields: the first is labeled 'Username' and the second is labeled 'Password'. To the right of these fields is a rectangular button labeled 'Login'.

Gambar 3.11 Form Login Admin

3.4.4 Form Login Petugas Parkir

Pada *form login* Petugas Parkir berisi *username* dan *password* yang digunakan untuk *login* oleh petugas parkir guna mengakses *form* data mahasiswa. Tampilan *Form* Petugas Parkir dapat dilihat pada Gambar 3.12.

The image shows a rectangular window titled "Petugas Login". Inside the window, there are two text input fields. The first is labeled "Username" and the second is labeled "Password". Below these fields, centered towards the right, is a button labeled "Login".

Gambar 3.12 *Form* Login Petugas Parkir

3.4.5 Form Data Mahasiswa

Form Data Mahasiswa berfungsi menambah informasi mahasiswa seperti nim, nama, alamat, TTL, Fakultas, Jurusan, Angkatan serta plat nomor kendaraan yang digunakan berikut merupakan tampilan perancangan antar muka *form* data mahasiswa dapat dilihat pada Gambar 3.13.

The image shows a rectangular window titled "Data Mahasiswa". At the top, there is a large, empty rectangular box. Below this box, there are two columns of input fields. The left column contains fields for "Nim", "Nama", "Alamat", and "TTL". The right column contains fields for "Fakultas", "Jurusan", "Angkatan", and "No. Telp.". At the bottom of the form, there are five buttons: "Clear", "Tambah", "Simpan", "Edit", and "Hapus".

Gambar 3.13 *Form* Penambahan Data Mahasiswa

3.4.6 Form Petugas Parkir

Form yang berfungsi menambah *username* dan *password* untuk petugas parkir berisi Id, nama petugas, user serta *password* yang nantinya digunakan petugas untuk melakukan fungsi *login* dapat dilihat pada Gambar 3.14.

The image shows a web form titled "Data Petugas Parkir". It features four text input fields labeled "ID", "Nama", "Username", and "Password" stacked vertically. Below these fields are five buttons: "Clear", "Tambah", "Simpan", "Edit", and "Hapus". To the right of the input fields is a large, empty rectangular box, likely intended for a profile picture or additional information.

Gambar 3.14 Form Penambahan Petugas

3.4.7 Form Admin

Form yang berfungsi menambah *username* dan *password* untuk admin berisi Id, nama petugas, user serta *password* yang nantinya digunakan admin untuk melakukan fungsi *login* dapat dilihat pada Gambar 3.15.

The image shows a web form titled "Data Admin". It features four text input fields labeled "ID", "Nama", "Username", and "Password" stacked vertically. Below these fields are five buttons: "Clear", "Tambah", "Simpan", "Edit", and "Hapus". To the right of the input fields is a large, empty rectangular box, likely intended for a profile picture or additional information.

Gambar 3.15 Form Penambahan Admin

3.4.8 Form Data Parkir

Form yang berfungsi untuk aplikasi keluar masuk parkir mahasiswa yang nantinya berisi nim, nomor kendaraan, jam masuk, jam keluar, tanggal masuk serta tanggal keluar kendaraan untuk perancangan antar muka dapat dilihat pada Gambar 3.16.

The image shows a software interface window titled "Data Parkir". At the top, there is a header bar with the text "Data Parkir". Below the header is a large, empty rectangular box, likely intended for displaying a list of parking records. At the bottom of the window, there are four buttons: "Connect" and "Disconnect" are positioned on the left side, and "Delete" is on the right side.

Gambar 3.16 Form Aplikasi Parkir

3.4.9 Form Isi Ulang Saldo

Form yang berfungsi menambah saldo untuk kartu mahasiswa yang berfungsi sebagai pembayaran parkir yang bersifat prabayar. Berikut tampilan form isi ulang saldo dapat dilihat pada Gambar 3.17.

The image shows a software interface window titled "Isi Saldo". At the top, there is a header bar with the text "Isi Saldo". Below the header is a large, empty rectangular box. Underneath this box, there are three input fields with labels to their left: "Nim", "Sisa Saldo", and "Masukkan Saldo". To the right of the "Masukkan Saldo" field is a "Reload" button.

Gambar 3.17 Form Isi ulang

3.5 Perancangan Database

Perancangan *database* dibuat untuk menampung berbagai jenis – jenis data yang dibutuhkan dalam aplikasi parkir ini sehingga dapat digunakan sebagai bahan penyimpanan data.

3.5.1 Tabel

Terdapat beberapa tabel dalam *database* yang digunakan pada aplikasi ini. Meliputi tabel data mahasiswa, tabel data parkir serta tabel saldo.

A. Tabel Data Mahasiswa

Dibawah ini merupakan data – data yang berguna sebagai informasi mahasiswa yang meliputi nim, nama, alamat, tempat tanggal lahir, fakultas, jurusan, angkatan, nomor telpon, nomor kendaraan serta saldo yang dapat dilihat pada Tabel 3.1.

Tabel 3.1. Tabel Mahasiswa

<i>Field</i>	<i>Type</i>	<i>Size</i>
Nim	<i>Char</i>	20
Nama	<i>Char</i>	40
Alamat	<i>Char</i>	40
TTL	<i>Char</i>	20
Fakultas	<i>Char</i>	20
Jurusan	<i>Char</i>	20
Angkatan	<i>Char</i>	10
Nomor_telepon	<i>Char</i>	20
Nomor_Kendaraan	<i>Char</i>	20
Saldo	<i>Char</i>	20

B. Tabel Data Parkiran

Dibawah ini merupakan data – data yang berguna sebagai data aplikasi saat parkir meliputi nim, nomor kendaraan, tanggal masuk, jam masuk, tanggal keluar, jam keluar, saldo yang dapat dilihat pada Tabel 3.2.

Tabel 3.2. Tabel Data Parkir

<i>Field</i>	<i>Type</i>	<i>Size</i>
Nim	<i>Char</i>	20
Nomor_kendaraan	<i>Char</i>	10
Tanggal_masuk	<i>Date</i>	20
Jam_masuk	<i>Date</i>	20

Tanggal_keluar	<i>Date</i>	20
Jam_keluar	<i>Date</i>	20
Saldo	<i>Char</i>	10

C. Tabel Saldo

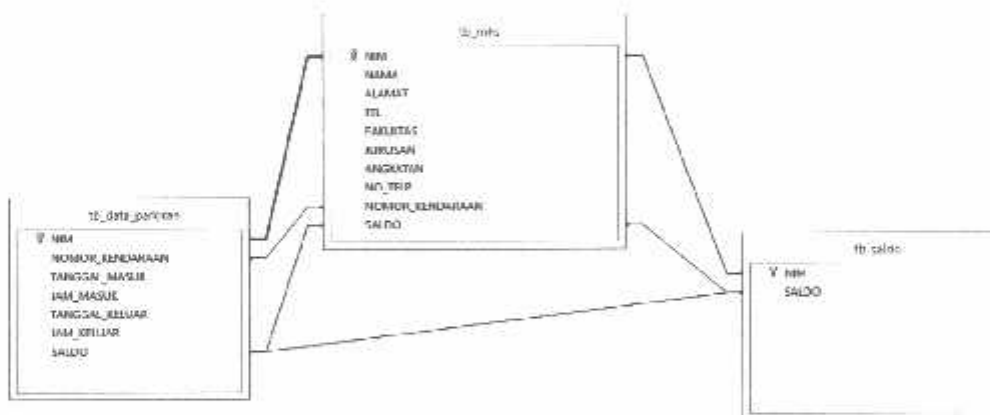
Dibawah ini merupakan tabel saldo yang nantinya digunakan sebagai saldo isi ulang dalam aplikasi yang dapat dilihat pada Tabel 3.1..

Tabel 3.3. Tabel Saldo

<i>Field</i>	<i>Type</i>	<i>Size</i>
Nim	<i>Char</i>	20
Saldo	<i>Char</i>	20

3.5.2 Relasi Tabel

Relasi tabel merupakan hubungan antar tabel dimana dalam satu tabel membutuhkan data dari tabel lainnya dalam aplikasi terdapat tiga buah tabel yang memiliki relasi yakni tabel mahasiswa, tabel data parkir serta tabel saldo. Berikut merupakan tabel relasi yang dapat dilihat pada Gambar 3.18.



Gambar 3.18 Relasi Tabel

BAB IV

IMPLEMENTASI DAN PENGUJIAN

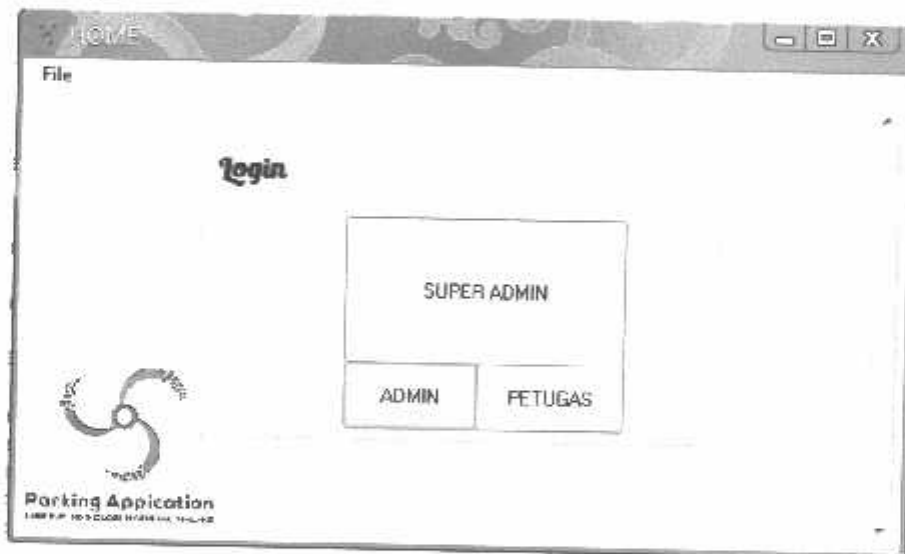
Dalam bab ini akan dijelaskan mengenai hasil uji coba perangkat *Smart Card*. Setelah itu hasil uji coba yang telah dilakukan, akan dianalisa apakah rancangan ini dapat memenuhi tujuan yang akan dicapai seperti yang dipaparkan pada Bab I.

4.1 Implementasi

Merupakan penerapan dari aplikasi *Smart Card* berisi tampilan tampilan *form* beserta konfigurasi *Smart Card* yang ada dalam aplikasi parkir ini.

4.1.1 Form utama Aplikasi

Halaman ini merupakan halaman yang pertama kali tampil pada saat membuka aplikasi dengan tiga tombol yang berguna sebagai fungsi *login* pada *Super Admin*, *Admin* serta *Petugas parkir*. Berikut tampilan *form Home* pada Gambar 4.1.



Gambar 4.1 Tampilan Utama *Home*

4.1.2 Form Login Super Admin

Halaman ini berisi *form login* untuk *Super Admin username* serta *password* sudah diatur default dari program sehingga tidak dapat diubah lagi. Tampilan *form login Super Admin* dapat dilihat pada Gambar 4.2.



Gambar 4.2 Tampilan *Form Login Super Admin*

4.1.3 Form Login Admin

Halaman ini berisi *form login* untuk *admin username* serta *password* yang digunakan telah didaftarkan oleh *Super Admin* sehingga hanya *username* dalam *database* yang dapat melakukan fungsi *login*. Berikut tampilan *form login admin* pada Gambar 4.3.



Gambar 4.3 Tampilan *Form Login Admin*

4.1.4 Form Login Petugas Parkir

Halaman ini berisi *form login* untuk petugas parkir *username* serta *password* yang digunakan telah didaftarkan oleh *Admin* sehingga hanya *username* dalam *database* yang dapat melakukan fungsi *login*.

Gambar 4.4 Tampilan *Form Login* Petugas Parkir

4.1.5 Form Data Mahasiswa

Form Data mahasiswa berisi informasi mahasiswa serta data kendaraan seperti nim, nama, alamat, tempat tanggal lahir, fakultas, jurusan, angkatan, nomor telepon serta nomor kendaraan yang dipakai dalam aplikasi ini juga dapat merubah default saldo awal saat pertama kali kartu dibuat. Tampilan *form* data mahasiswa dapat dilihat pada Gambar 4.5.

Gambar 4.5 Tampilan *Form* Data Mahasiswa

4.1.6 Form Data Parkir

Form Data mahasiswa berisi *log* atau *history* keluar masuknya kendaraan bermotor. Dalam *form* ini dicatat nim, nomor kendaraan, tanggal masuk, jam masuk, tanggal keluar dan jam keluar serta jumlah saldo terakhir. Tampilan *form* data parkir dapat dilihat pada Gambar 4.6.

13-Januari-2016 23:27:49						
NIM	NO. MOP KENDARAAN	TANGGAL MASUK	JAM MASUK	TANGGAL KELUAR	JAM KELUAR	SALDO
1218142 N3664L		25 November 2015	06:17:00	25 November 2015	06:17:06	4000
1218142 N3064L		25 November 2015	06:17:11	25 November 2015	06:17:10	3000
1218142 N3668		15 Desember 2015	11:04:38	15 Desember 2015	11:04:42	4000
1218142 N0000K		15 Desember 2015	11:04:58	15 Desember 2015	11:04:58	3000
1218142 N888		04 January 2016	11:49:47	04 January 2016	11:48:51	13000
1218142 N3664L		04 January 2016	11:59:38	04 January 2016	12:00:03	18000
1218142 N3664L		06 January 2016	10:05:42	06 January 2016	10:05:47	13000
1218142 N3668		11 January 2016	14:04:23	11 January 2016	14:04:26	12000
1218142 N3264L		12 January 2016	15:23:13	12 January 2016	15:23:18	10000
1218142 N3264L		13 January 2016	08:52:57	13 January 2016	08:52:54	3000
1218142 N3664L		13 January 2016	09:03:41	13 January 2016	09:03:44	6000

Gambar 4.6 Tampilan *Form* Data Parkir

4.1.7 Form Isi Ulang Saldo

Form Isi Ulang Saldo digunakan untuk pengisian ulang saldo kartu guna pembayaran parkir dalam *form* ini terdapat nim serta jumlah saldo akhir yang nantinya ditambahkan jumlah saldo yang sudah dipaketkan dalam kelipatan 10.000 rupiah. Tampilan *form* isi ulang saldo dapat dilihat pada Gambar 4.7.

NIM	SALDO
1218142	18000

NIM _____

SISA SALDO _____

MASUKKAN SALDO _____

Reload

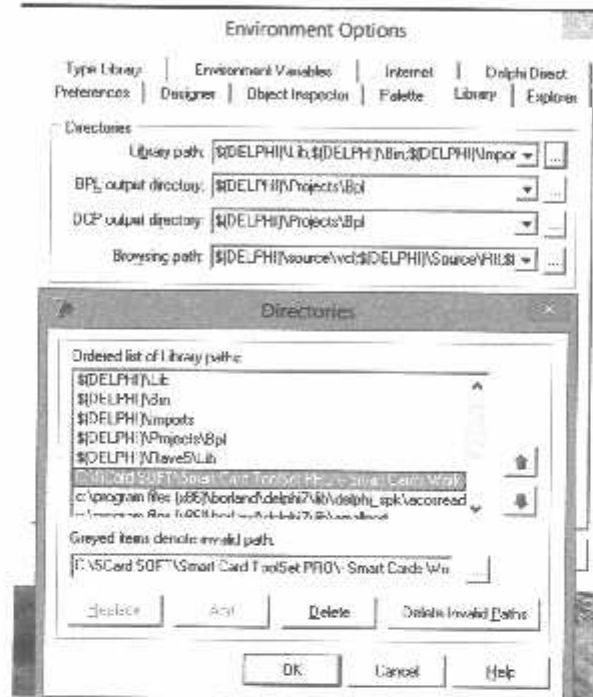
Gambar 4.7 Tampilan *Form* Isi Ulang Saldo

4.2 Konfigurasi Reader ACS 38

Merupakan cara atau langkah – langkah dalam mengkonfigurasi *reader ACS 38* dengan komputer menggunakan aplikasi Delphi7.

4.2.1 Import Library Smart Card Tool Set Pro v3.4

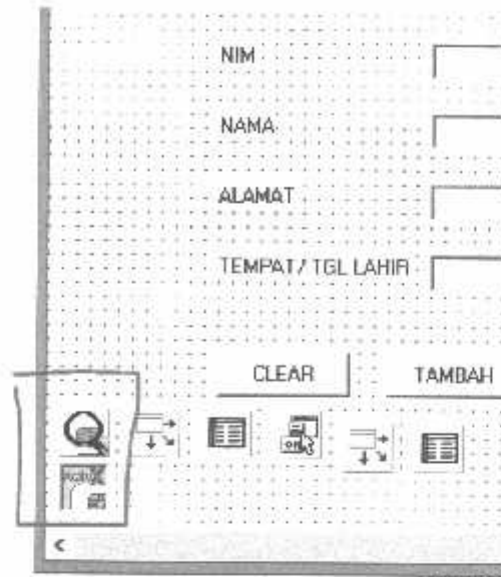
Konfigurasi *reader ACS38* melalui Delphi 7 dengan *plugin library Smart Card Tool Set Pro V3.4* . Langkah awal adalah dengan *import library* dengan memilih tab *tool* -> *environment option* -> *library* -> *library path* dan pilih *smart card tool set pro library* seperti gambar 4.8.



Gambar 4.8 Import *library smart card tool set pro*

4.2.2 Konfigurasi Library

Setelah melakukan *Import Library* kemudian masukkan *ActiveX* dengan cara *drag icon SmartCard Toolset* pada *bar ActiveX* dan masukkan pada *form* yang menggunakan fungsi – fungsi yang berhubungan dengan *Smart Card*. Pengaturan *Library ActiveX Smart Card Tool Set Pro* dalam aplikasi seperti gambar 4.9.



Gambar 4.9 Pengaturan *Library ActiveX Smart Card Tool Set Pro*

Untuk mengaktifkan *library* ini diperlukan *source code* yang harus dimasukkan pada tombol *connect* untuk koneksi *reader* adalah seperti pada gambar 4.10.

```

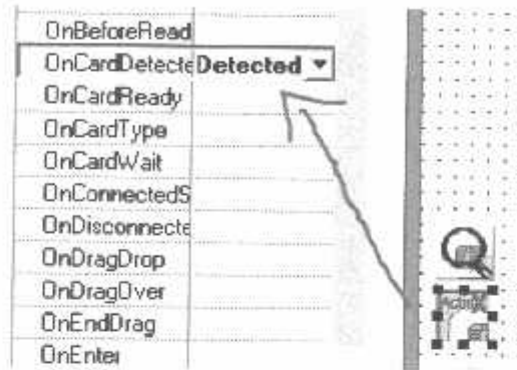
C#
File Log Out
connect_act Disconnect Reader
mahasiswa.DAS
form_saldo data mahasiswa
instanci:
• SmartCardToolSetSERVER_ActiveX1.ConnectSCARD;
• con:~SmartCardToolSetSERVER_ActiveX1.IsConnectedSCARD;
• Opened:=True;
• if con then
begin
• ShowMessage('Koneksi Berhasil');
• Connected:=True;
• ShowMessage('Masukkan Kartu Untuk Disconnect');
• bt_tambah.Enabled:=True;
• bt_edit.Enabled:=True;
• bt_hapus.Enabled:=True;
• bt_saldo_awal.Enabled:=True;
• dggrid_mhs.Visible:=True;
• dggrid_mhs.Enabled:=True;
• bt_connect.Enabled:=False;
• dggrid_mhs.Visible:=True;
end
else
begin
• ShowMessage('Koneksi Gagal , Masukkan Reader');
• Connected:=false;
end;
end:
end:
7/3, 1 Modified instal Code/Diagram/

```

Gambar 4.10 Konfigurasi *reader*

4.2.3 Konfigurasi *Smart Card* 7816

Setelah konfigurasi *reader* berhasil masuk pada tahapan konfigurasi *smart card*. Pada event *ActiveX Smart Card Toolset* klik *onCardDetected* dan masukkan *source code* seperti Gambar 4.11 dan Gambar 4.12



Gambar 4.11 Konfigurasi *smart card* pada *Event OnCardDetected*

```
begin
koneksi:=SmartCardToolSetSERVER_ActiveX1.IsConnectedSCARD;
```

Gambar 4.12 *SourceCode* pada *ActiveX*

4.2.4 Tahap Pengolahan Data *Smart Card*

Setelah tahapan konfigurasi *Smart Card* maka *source code* dimasukkan untuk proses pembacaan dan penulisan pada kartu. Pada *Smart Card* digunakan angka *hexadecimal* untuk berkomunikasi baik untuk pemberian perintah maupun *reply answer* dari kartu.

4.2.4.1 *Source Code* Konversi

Dibawah ini merupakan *Source Code* konversi dari bilangan hexa baik dari hexa ke integer maupun integer ke bilangan hexa, seperti yang terlihat pada Gambar 4.13 dan Gambar 4.14.

```
function hexaToInt(a : string) : Integer;
begin
    if (a <> '') and (a[1] <> '#') then
        result := hexToInt64('#' + a);
    else
        result := hexToInt64(a);
    end;
end;
```

Gambar 4.13 *SourceCode Hex to Integer*

```

*
*      awal:=CardReply;
*      Isi:=HexToBin(awal,8);
*      hasil:=BinToHex(Isi);
*

```

Gambar 4.14 *Source Code Integer to Hex*

4.2.4.2 Pembacaan dan Penulisan Data Pada *Smart Card*

Baik proses pembacaan dan penulisan data pada *Smart Card* dilakukan dengan beberapa tahapan yakni verifikasi PIN yang terdapat pada kartu (*Verify*), pemilihan alamat data pada kartu (*Select*) selanjutnya perintah disesuaikan sesuai kebutuhan baik pembacaan data (*Read*) maupun penulisan data (*Write*). Seperti yang sudah dijelaskan pada BAB II, perintah yang digunakan berupa APDU (*Application Protocol Data Unit*) di dalam APDU terdapat beberapa komponen yakni cla (*Class*), ins (*Instruction*), p1-p2 (*Parameter*), lc (*Length Data In*), Datafield, le (*Length Parameter*). Berikut adalah perintah atau *command* yang digunakan pada masing – masing tahapan seperti yang terdapat pada Tabel 4.1.

Tabel 4.1. Perintah APDU *Smart Card*

No	Fungsi	Cla	Ins	P1	P2	Data Field	Lc	le
1	<i>Verify</i>	80	20	07	00	41 43 4F 53 54 45 53 54	08	
2	<i>Select</i>	80	A4	00	00	FF 02	02	
3	<i>Read</i>	80	B2	00	00		00	
4	<i>Write</i>	80	D2	00	00	00 12 96 5E	04	

Dari tabel diatas terdapat Cla (*Class*) dimana 80 merupakan *class* yang dipakai oleh *Smart Card* ISO-7816 untuk berkomunikasi. Ins (*Instruction*) merupakan kode – kode pada setiap perintah yang digunakan, P1 dan P2 sebagai *Parameter* merupakan *header* dari data, *Data Field* di perintah *verify* merupakan data PIN dari kartu, sedangkan pada perintah *select* digunakan untuk memasukkan alamat *file* pada kartu untuk *write* digunakan untuk memasukkan data yang semuanya dimasukkan dalam bentuk bilangan hexa. Lc (*Data Field Length*) adalah seberapa panjang data yang dimasukkan dalam *Data Field*. I.e (*Parameter Length*) sebagai panjang dari parameter.

4.3 Pengujian

Pengujian dilakukan untuk mengetahui apakah setiap *form* pada aplikasi yang dibangun berjalan dengan baik yang diuji menggunakan *Windows 8.1* dengan Bahasa pemrograman *Delphi7*.

4.3.1 Pengujian *Login dan Logout*

Pengujian dilakukan pada setiap *form* yang menggunakan fungsi *login* di dalamnya. *Form – form* yang dimaksud meliputi *Form login Super admin*, *Form Login Admin* serta *Form Login Petugas Parkir*.

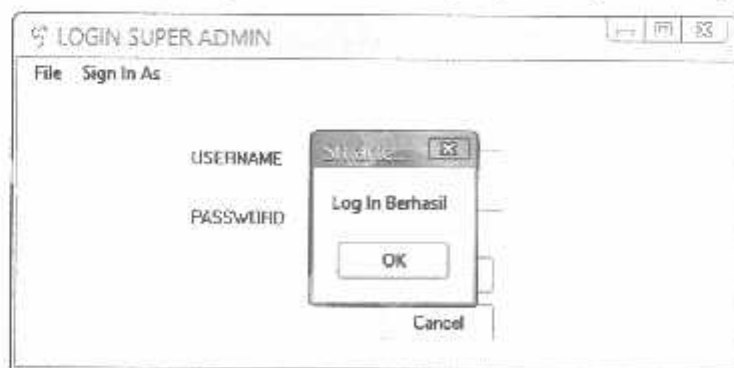
4.3.1.1 Pengujian *Form Super Admin Login*

Form Super Admin memiliki fungsi *login* didalamnya dengan *Username* dan *Password* yang telah di atur secara *default* dengan “*super_ricky*” sebagai *username* serta “*1994*” sebagai *passwordnya*. Berikut merupakan tampilan awal *login Super Admin* dapat dilihat pada Gambar 4.15.



Gambar 4.15 Tampilan *Login Awal*

Selanjutnya proses *login* dengan menekan tombol *Sign In* pada *Form Login Super Admin*. Seperti pada Gambar 4.16 yang menunjukkan *Login berhasil*.



Gambar 4.16 Tampilan *Login Berhasil*

Dibawah ini merupakan pengujian *login* dengan *username* dan *password* yang salah. Hasil pengujian dapat dilihat pada Gambar 4.17.



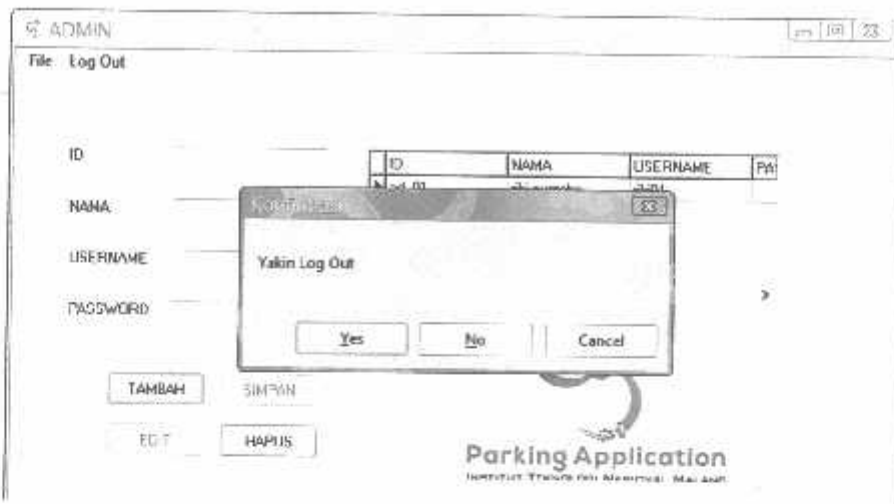
Gambar 4.17 Tampilan *Login* Gagal

Setelah berhasil *Login* kemudian *Super Admin* akan langsung membuka *Form Admin* sebagai tampilan awal guna menambahkan hak akses *Admin* pada aplikasi seperti yang ditunjukkan pada Gambar 4.18.



Gambar 4.18 Tampilan Awal Setelah *Login*

Untuk fungsi *logout* pada aplikasi dapat diakses pada *MenuTab* pada setiap *form* yang ada setelah *login*. Berikut adalah pengujian *logout* yang diuji pada *form Admin* seperti pada Gambar 4.19.



Gambar 4.19 Tampilan *Logout Super Admin*

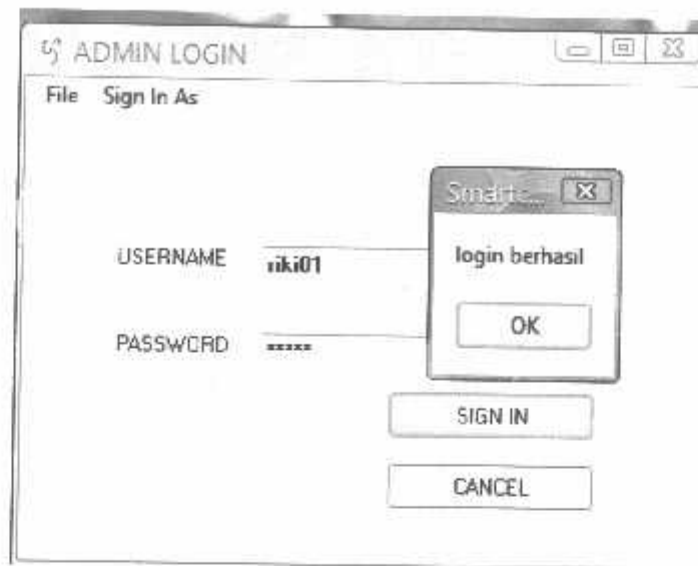
4.3.1.2 Pengujian *Form Admin Login*

Form Admin memiliki fungsi *login* didalamnya dengan *Username* dan *Password* yang telah didaftarkan oleh *Super Admin* sebelumnya. Berikut merupakan tampilan awal *login Admin* dapat dilihat pada Gambar 4.20.



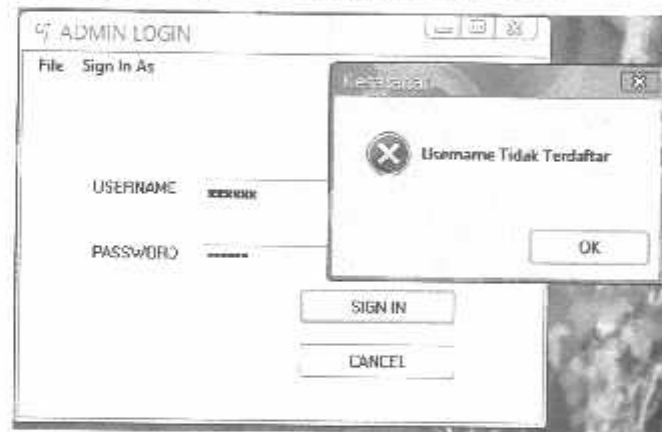
Gambar 4.20 Tampilan *Login Awal*

Selanjutnya proses *login* dengan menekan tombol *Sign In* pada *Form Login Admin*. Seperti pada Gambar 4.21 yang menunjukkan *Login* berhasil.



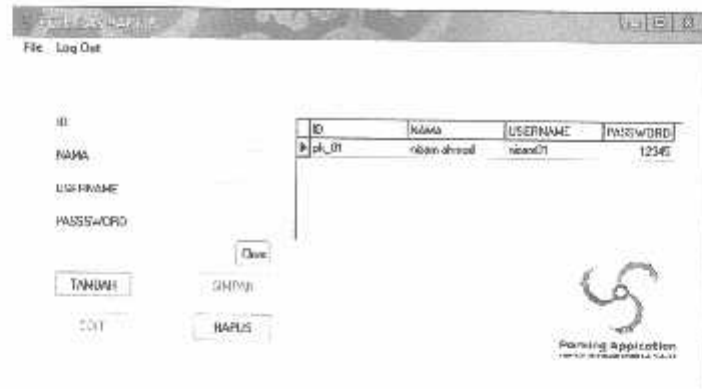
Gambar 4.21 Tampilan *Login* Berhasil

Dibawah ini merupakan pengujian *login* dengan *username* dan *password* yang salah atau dengan *username* dan *password* yang tidak terdaftar dalam *database*. Hasil pengujian dapat dilihat pada Gambar 4.22.



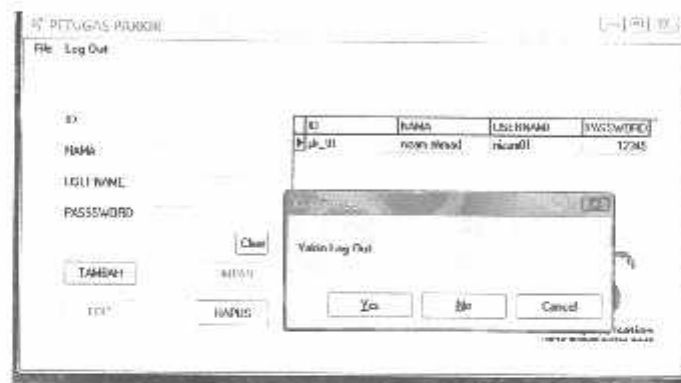
Gambar 4.22 Tampilan *Login* Gagal

Setelah berhasil *Login* kemudian *Admin* akan langsung membuka *Form* *Petugas Parkir* sebagai tampilan awal guna menambahkan hak akses *Petugas Parkir* pada aplikasi seperti yang ditunjukkan pada Gambar 4.23.



Gambar 4.23 Tampilan Awal Setelah *Login*

Untuk fungsi logout pada aplikasi dapat diakses pada Menu Tab pada setiap *form* yang ada setelah *login*. Berikut adalah pengujian logout yang diuji pada *form* Petugas Parkir seperti pada Gambar 4.24.



Gambar 4.24 Tampilan *Logout Super Admin*

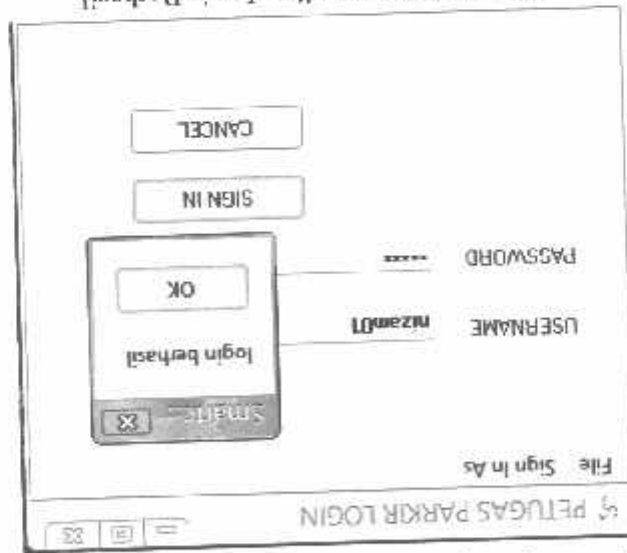
4.3.1.3 Pengujian *Form Petugas Parkir Login*

Form Petugas Parkir memiliki fungsi *login* didalamnya dengan *Username* dan *Password* yang telah didaftarkan oleh *Admin* sebelumnya. Berikut merupakan tampilan awal *login* Petugas Parkir dapat dilihat pada Gambar 4.25.



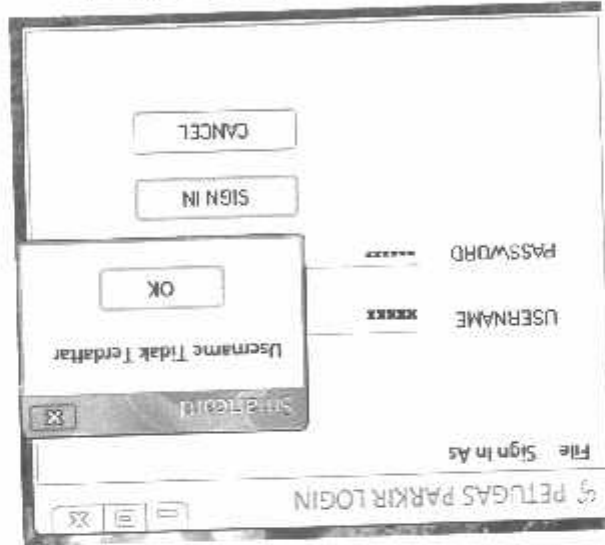
Gambar 4.25 Tampilan *Login Awal*

Selanjutnya proses login dengan menekan tombol *Sign In* pada *Form Login* Petugas Parkir. Seperti pada gambar 4.26 yang menunjukkan *Login* berhasil.



Gambar 4.26 Tampilan *Login* Berhasil

Dibawah ini merupakan pengujian login dengan *username* dan *password* yang salah atau dengan *username* dan *password* yang tidak terdaftar dalam *database*. Hasil pengujian dapat dilihat pada gambar 4.27.



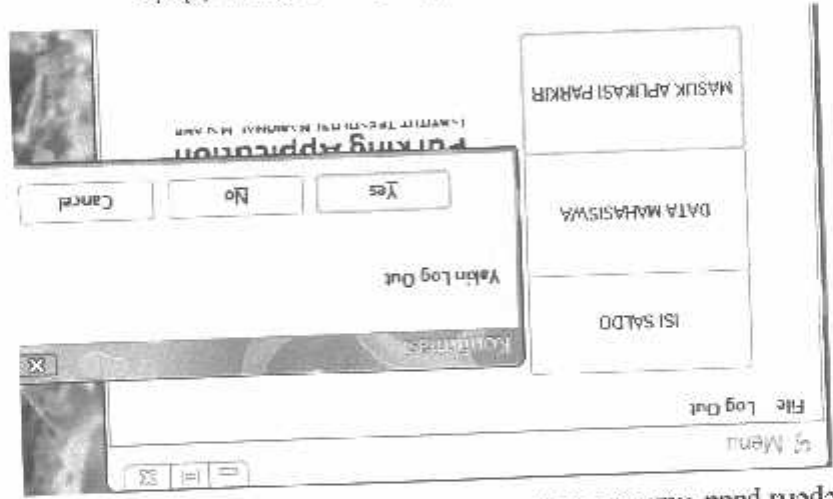
Gambar 4.27 Tampilan *Login* (gagal)

Setelah berhasil *Login* kemudian Petugas Parkir akan langsung membuka *Form Menu* yang berisi 3 menu seperti Isi Saldo, Data Mahasiswa dan Data Parkir seperti yang ditunjukkan pada Gambar 4.28.

4.3.2.1 Pengujian Form Admin
 Form Admin digunakan oleh Super Admin sebagai sarana untuk menambah hak akses pada admin. Terdapat Id, Nama, Username dan Password untuk di isi dalam form Admin seperti yang ditunjukkan pada Gambar 4.30.

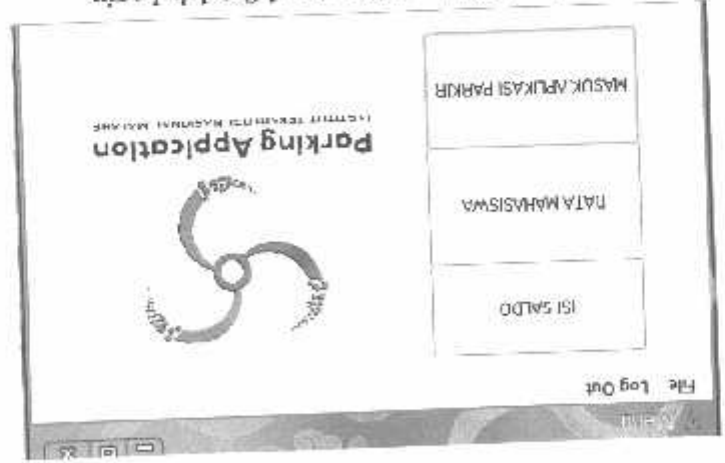
4.3.2 Pengujian CRUD (CREATE READ UPDATE DELETE)
 Pengujian dilakukan pada setiap form yang menggunakan fungsi CRUD di dalamnya. Form - form yang dimaksud meliputi Form Admin, Form Petugas Parkir, Form Data Mahasiswa, Form Data Parkir dan Form Isi Saldo.

Gambar 4.29 Tampilan Logout Super Admin



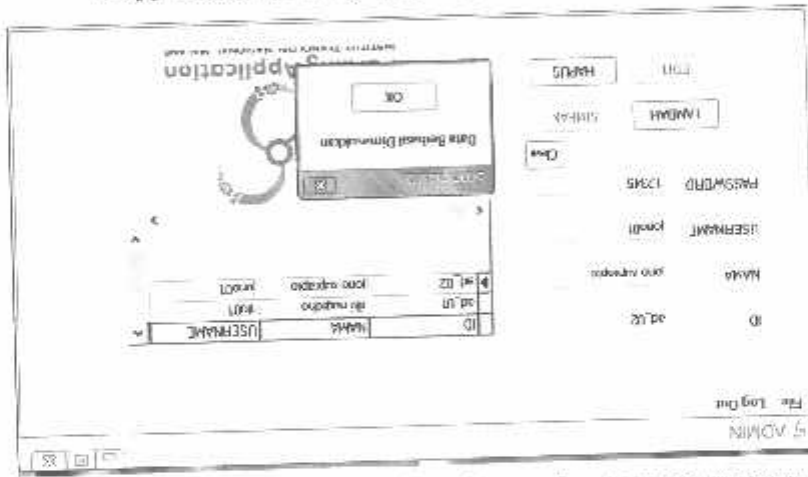
Untuk fungsi logout pada aplikasi dapat diakses pada Menu Tab pada setiap form yang ada setelah login. Berikut adalah pengujian logout yang diuji pada form Menu seperti pada Gambar 4.29.

Gambar 4.28 Tampilan Awal Setelah Login



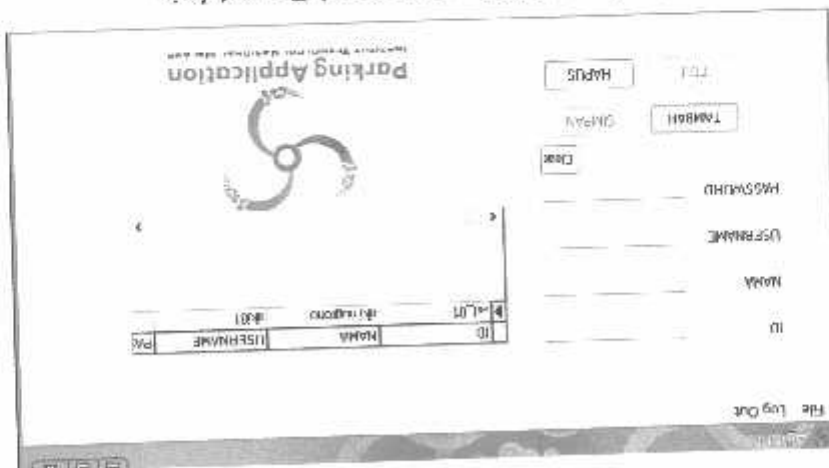
Titik pengujian *edit* data terlebih dahulu dipilih dengan klik label pada data yang ingin *diedit* setelah itu proses *editing data* yang dilakukan dengan mengubah nama "Jono suprapto" dengan "Jono sunandar" selanjutnya klik tombol *Edit*. Seperti yang ditunjukkan pada Gambar 4.32.

Gambar 4.31 Tampilan Data Berhasil Dimasukkan



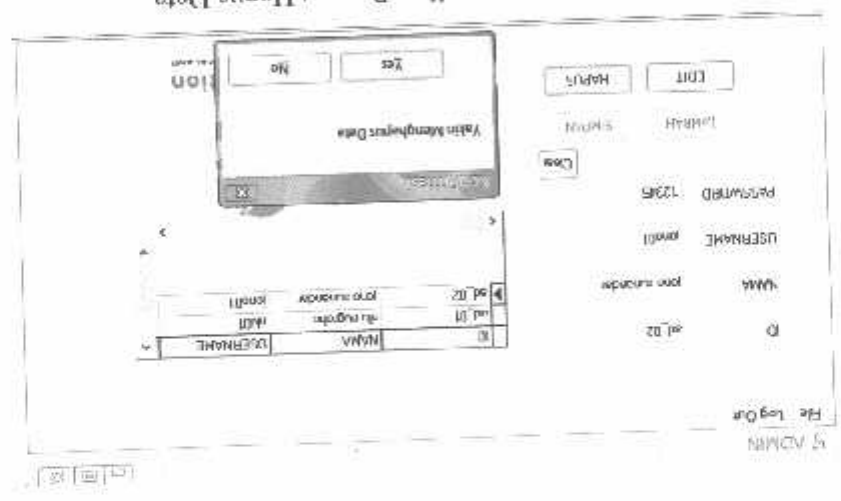
Selanjutnya dengan menekan tombol Tambah, *Super Admin* dapat menginputkan data – data yang di perlukan, setelah itu disimpan pada *database* dengan menekan tombol Simpan. Seperti pada Gambar 4.31.

Gambar 4.30 Tampilan Awal Form Admin

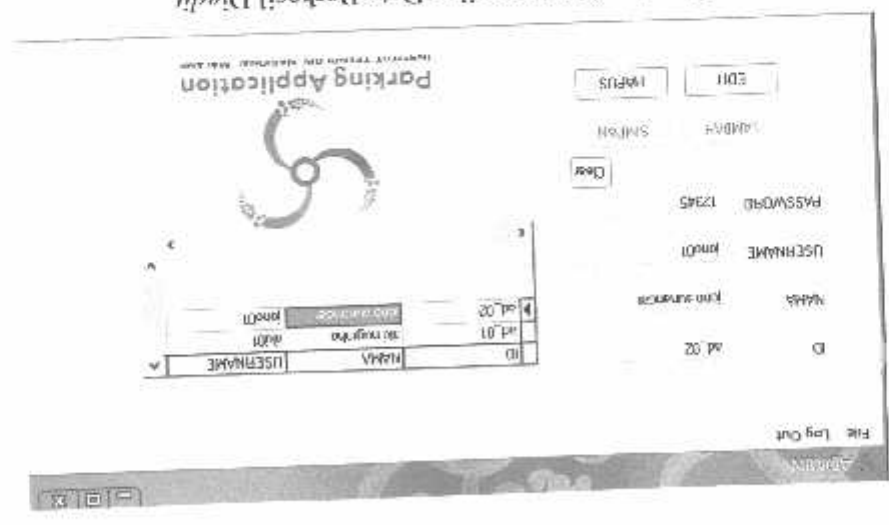


Hamptir sama seperti proses *edit* pengujian *delete* data dilakukan dengan memilih data terlebih dahulu kemudian klik tombol Hapus sehingga akan muncul *prompt* hapus data, kemudian klik *Yes* untuk menghapus data seperti pada Gambar 4.33 dan hasil hapus data pada Gambar 4.34.

Gambar 4.33 Tampilan *Prompt* Hapus Data

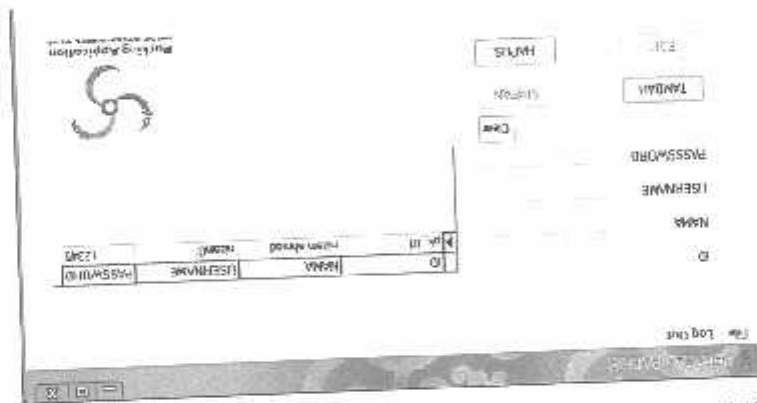


Gambar 4.32 Tampilan Data Berhasil *Dieedu*



Selanjutnya dengan menekan tombol Tambah, Admin dapat memasukkan data - data yang di perlukan, setelah itu disimpan pada *database* dengan menekan tombol Simpan. Seperti pada Gambar 4.36.

Gambar 4.35 Tampilan Awal Form Petugas Parkir

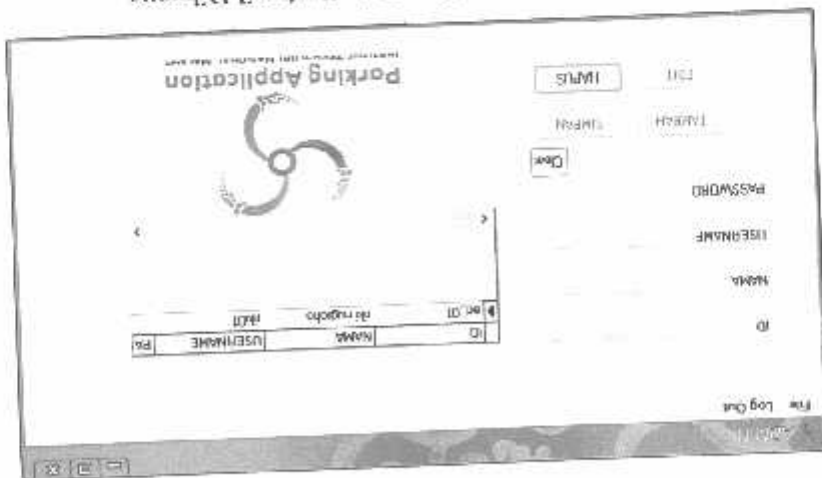


Gambar 4.35.

Form Petugas Parkir digunakan oleh Admin sebagai sarana untuk menambah hak akses pada Petugas Parkir. Terdapat Id, Nama, Username dan Password untuk di isi dalam form Petugas Parkir seperti yang ditunjukkan pada

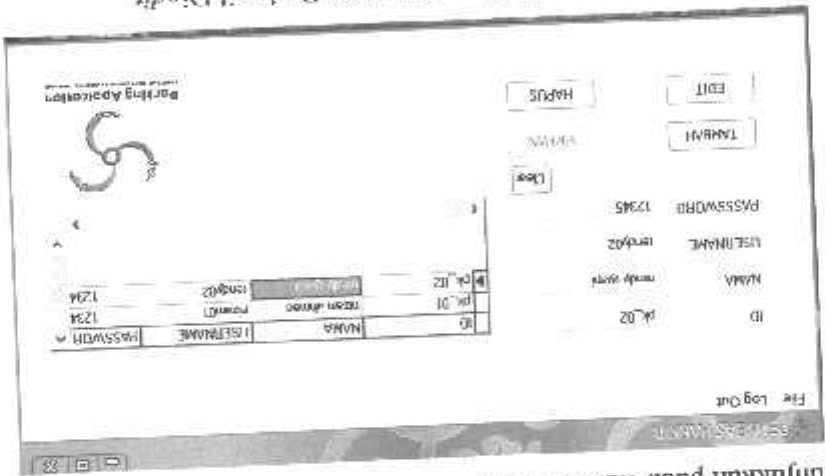
4.3.2.2 Pengujian Form Petugas Parkir

Gambar 4.34 Tampilan Data Berhasil Dihapus



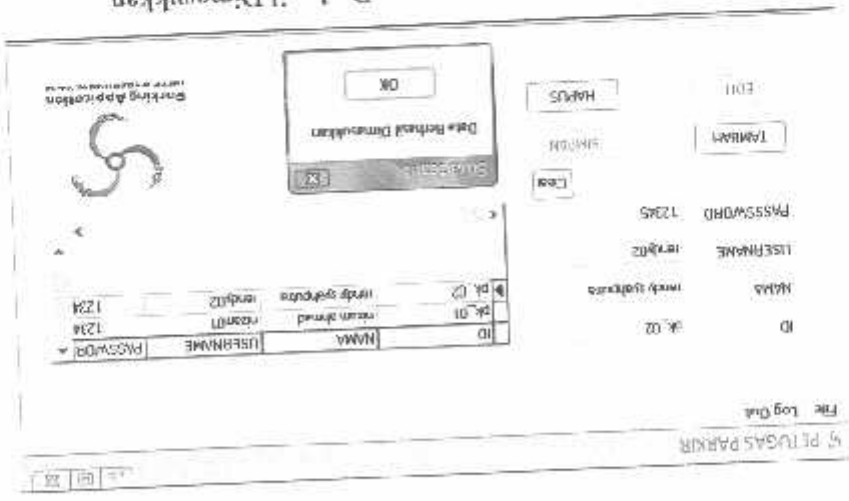
Hampir sama seperti proses *edit* pengujian *delete data* dilakukan dengan memilih data terlebih dahulu kemudian klik tombol Hapus sehingga akan muncul *prompt* hapus data, kemudian klik *Yes* untuk menghapus data seperti pada Gambar 4.38 dan hasil hapus data pada Gambar 4.39.

Gambar 4.37 Tampilan Data Berhasil Diedit



Untuk pengujian *edit* data terlebih dahulu dipilih dengan klik label pada data yang ingin *diedit* setelah itu proses *editing data* yang dilakukan dengan mengubah nama "rendy syahputra" dengan "rendy syapi" selanjutnya klik tombol *Edit*. Seperti yang ditunjukkan pada Gambar 4.37.

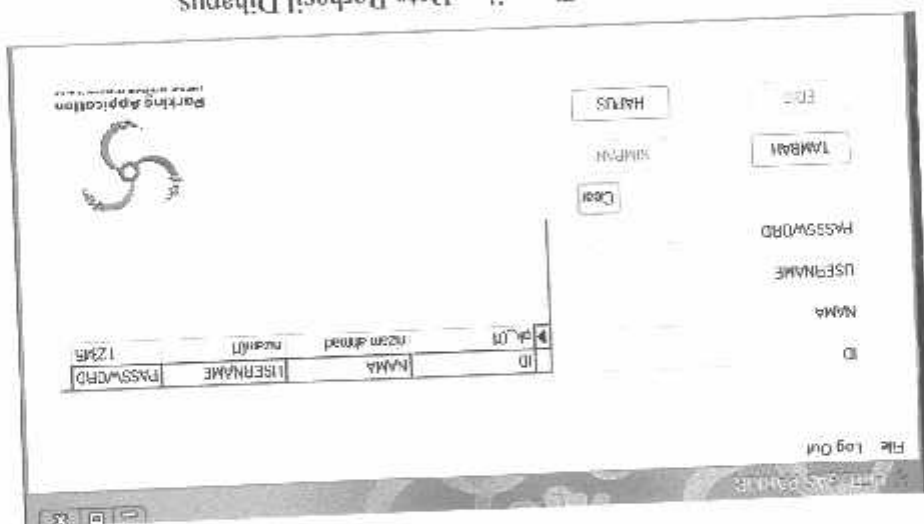
Gambar 4.36 Tampilan Data Berhasil Dimasukkan



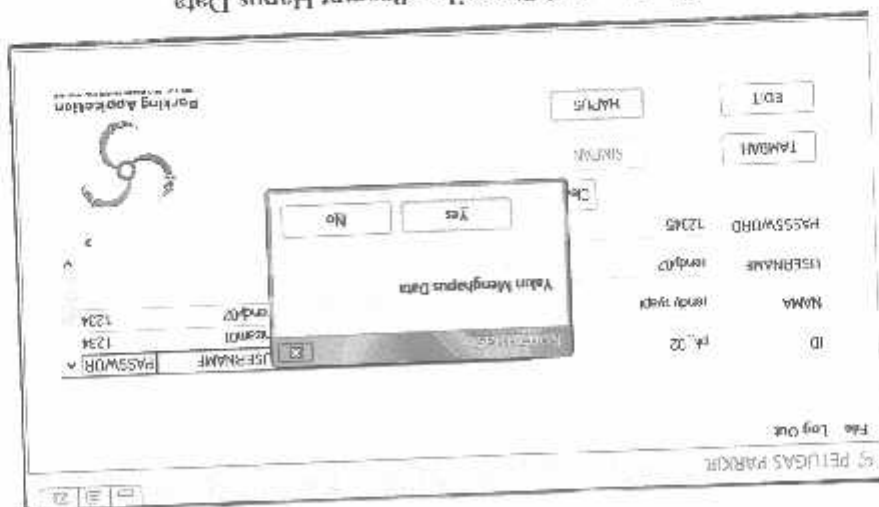
4.3.2.3 Pengujian Form Data Mahasiswa

Form Data Mahasiswa digunakan oleh Petugas Parkir sebagai sarana untuk menambah data mahasiswa dan kendaraan bermotor. Form Data mahasiswa berisi informasi mahasiswa serta data kendaraan seperti nim, nama, alamat, tempat tanggal lahir, fakultas, jurusan, angkatan, nomor telepon serta nomor kendaraan yang dipakai dalam aplikasi ini juga dapat diubah *default* saldo awal saat pertama kali kartu dibuat seperti yang ditunjukkan pada Gambar 4.40.

Gambar 4.39 Tampilan Data Berhasil Dihapus

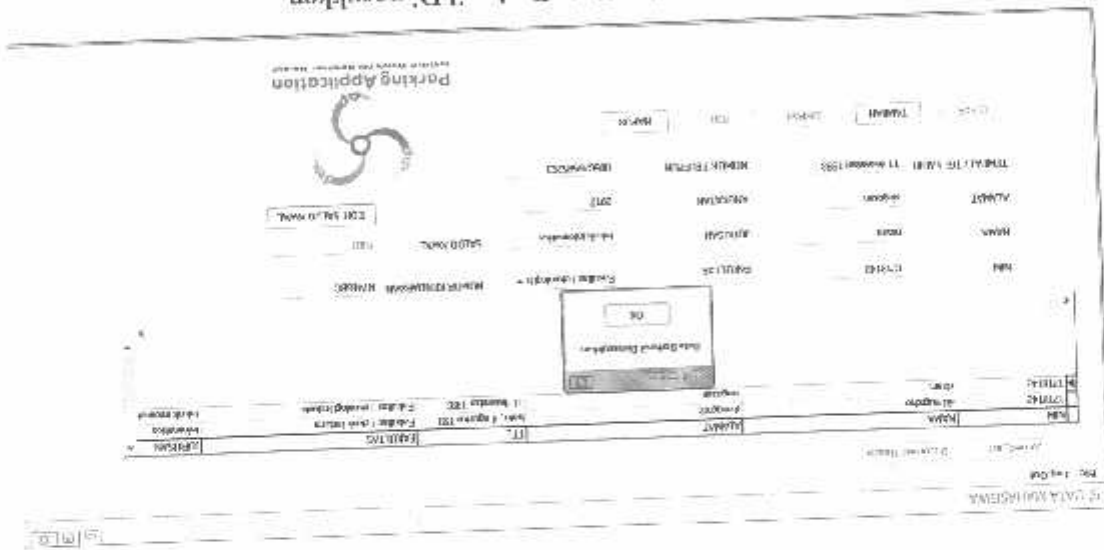


Gambar 4.38 Tampilan Prompt Hapus Data



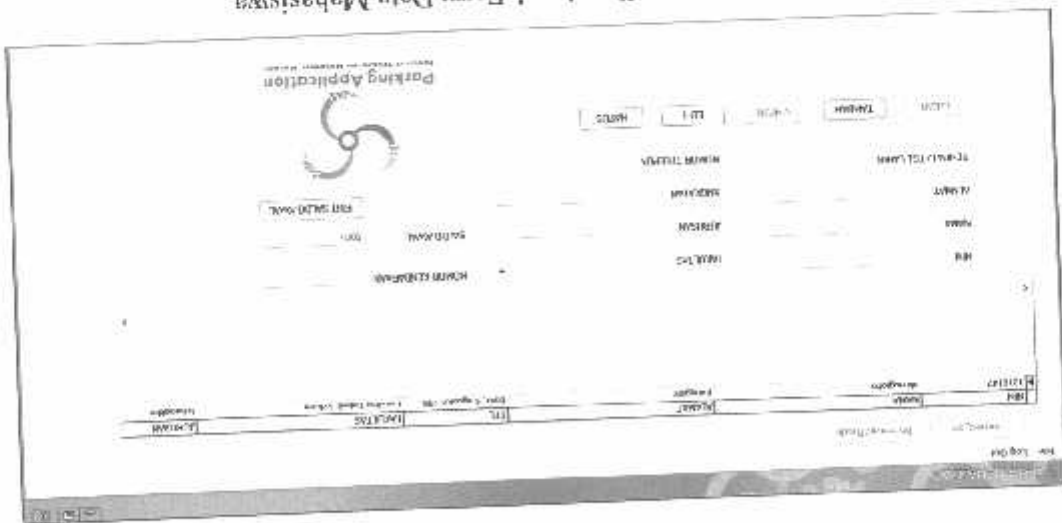
Untuk pengujian *edit* data terlebih dahulu dipilih dengan klik label pada data yang ingin *diedit* setelah itu proses *editing data* yang dilakukan dengan mengubah nama "nizam" dengan "nizam ahmad" selanjutnya klik tombol *Edit*. Seperti yang ditunjukkan pada Gambar 4.42.

Gambar 4.41 Tampilan Data Berhasil Dimasukkan

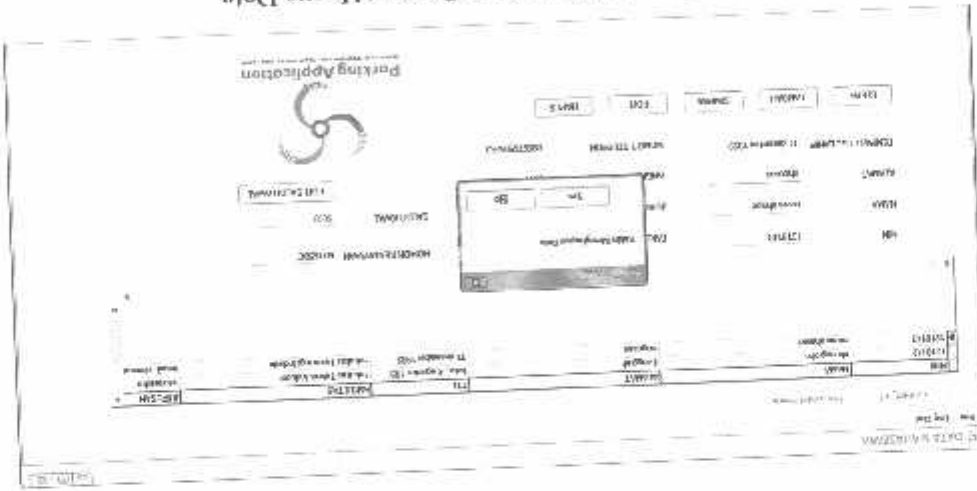


Selanjutnya dengan menekan tombol Tambah, Petugas Parkir dapat menginputkan data – data yang di perlukan, setelah itu disimpan pada *database* dengan menekan tombol Simpan. Seperti pada Gambar 4.41.

Gambar 4.40 Tampilan Awal Form Data Mahasiswa

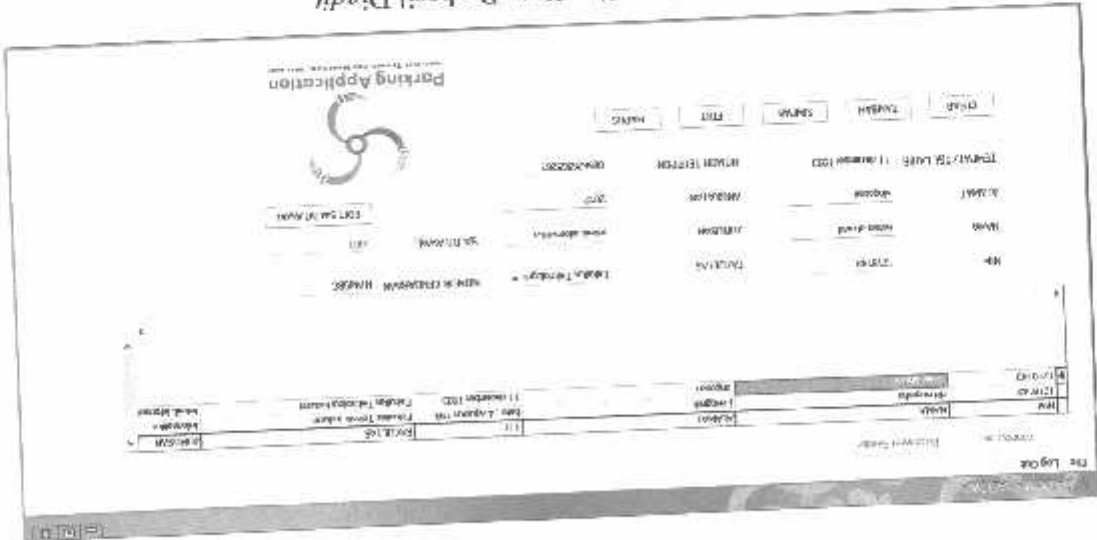


Gambar 4.43 Tampilan Prompt Hapus Data



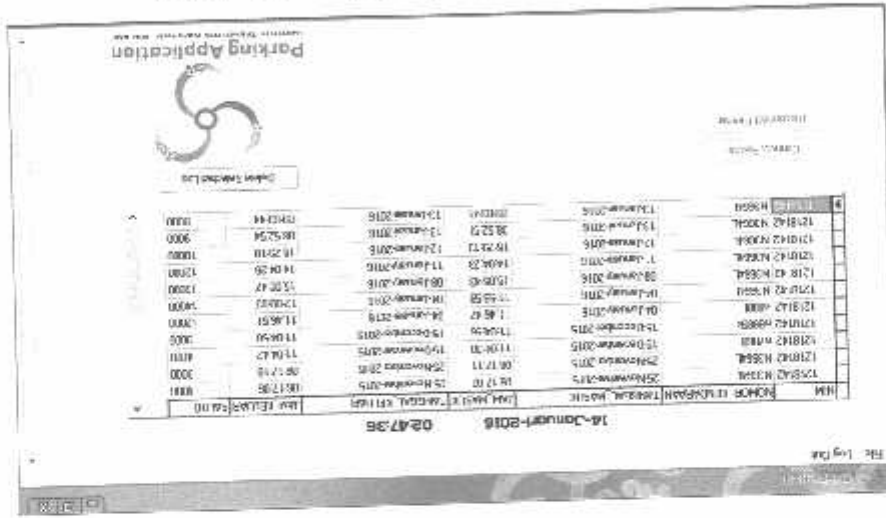
Hampir sama seperti proses *edit* pengujian *delete* data dilakukan dengan memilih data terlebih dahulu kemudian klik tombol Hapus sehingga akan muncul *prompt* hapus data, kemudian klik *Yes* untuk menghapus data seperti pada Gambar 4.43 dan hasil hapus data pada Gambar 4.44.

Gambar 4.42 Tampilan Data Berhasil Di~~edit~~



Pengujian *form* digunakan dengan menggunakan *smart card* ISO 7816 dan *reader ACS38*. Pertama masukkan kartu untuk mencatat nim, nomor kendaraan tanggal masuk dan jam masuk yang diasumsikan sebagai mahasiswa yang memasuki area parkir, selanjutnya masukkan kartu lagi dan mencatat tanggal keluar, jam keluar serta pengurangan saldo sejumlah 1.000 rupiah. Pengujian dapat dilihat pada (gambar 4.46, Gambar 4.47, Gambar 4.48 dan Gambar 4.49).

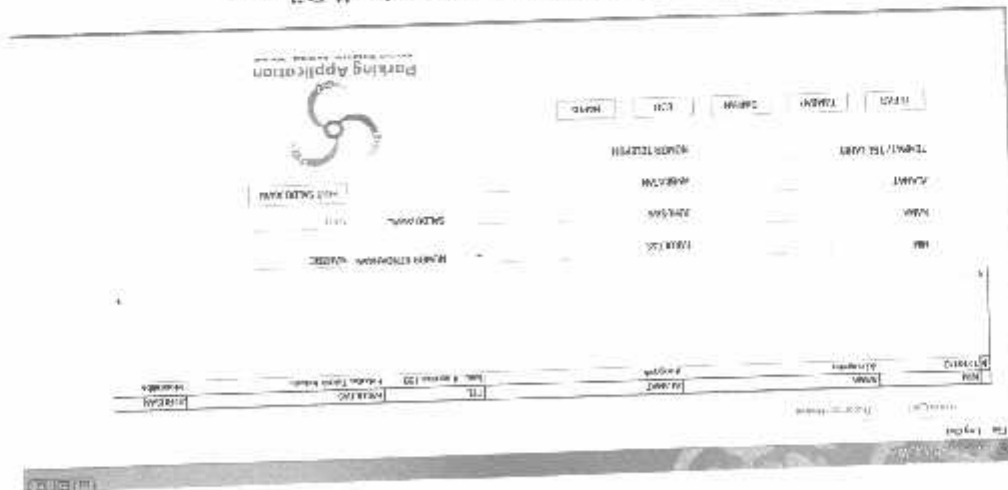
Gambar 4.45 Tampilan Awal Form Data Parkir



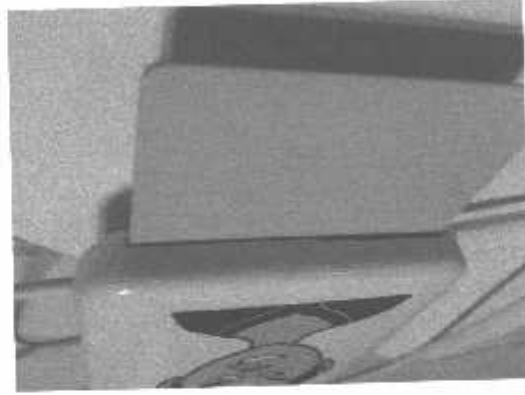
Form Data Parkir digunakan oleh petugas parkir sebagai sarana transaksi pembayaran parkir. Didalam *form* Data Parkir terdapat sebuah tabel untuk mencatat *history* parkir dimana terdapat nim, nomor kendaraan, tanggal masuk, jam masuk, tanggal keluar dan jam keluar seperti ditunjukkan pada Gambar 4.45.

4.3.2.4 Pengujian Form Data Parkir

Gambar 4.44 Tampilan Data Berhasil Dihapus



Gambar 4.48 Proses Memasukkan Kartu (Keluar Parkir)



(History Terbahas)

Gambar 4.47 Pencatatan Nimm, Nomor Kendaraan Tanggal Masuk dan Jam Masuk

Parking Application

No	NIM	Nomor Kendaraan	Tanggal Masuk	Jam Masuk	Status
1	12191210001	12191210001	12/12/2018	08:17:00	10000
2	12191210002	12191210002	12/12/2018	08:17:00	10000
3	12191210003	12191210003	12/12/2018	08:17:00	10000
4	12191210004	12191210004	12/12/2018	08:17:00	10000
5	12191210005	12191210005	12/12/2018	08:17:00	10000
6	12191210006	12191210006	12/12/2018	08:17:00	10000
7	12191210007	12191210007	12/12/2018	08:17:00	10000
8	12191210008	12191210008	12/12/2018	08:17:00	10000
9	12191210009	12191210009	12/12/2018	08:17:00	10000
10	12191210010	12191210010	12/12/2018	08:17:00	10000
11	12191210011	12191210011	12/12/2018	08:17:00	10000
12	12191210012	12191210012	12/12/2018	08:17:00	10000
13	12191210013	12191210013	12/12/2018	08:17:00	10000
14	12191210014	12191210014	12/12/2018	08:17:00	10000
15	12191210015	12191210015	12/12/2018	08:17:00	10000
16	12191210016	12191210016	12/12/2018	08:17:00	10000
17	12191210017	12191210017	12/12/2018	08:17:00	10000
18	12191210018	12191210018	12/12/2018	08:17:00	10000
19	12191210019	12191210019	12/12/2018	08:17:00	10000
20	12191210020	12191210020	12/12/2018	08:17:00	10000
21	12191210021	12191210021	12/12/2018	08:17:00	10000
22	12191210022	12191210022	12/12/2018	08:17:00	10000
23	12191210023	12191210023	12/12/2018	08:17:00	10000
24	12191210024	12191210024	12/12/2018	08:17:00	10000
25	12191210025	12191210025	12/12/2018	08:17:00	10000
26	12191210026	12191210026	12/12/2018	08:17:00	10000
27	12191210027	12191210027	12/12/2018	08:17:00	10000
28	12191210028	12191210028	12/12/2018	08:17:00	10000
29	12191210029	12191210029	12/12/2018	08:17:00	10000
30	12191210030	12191210030	12/12/2018	08:17:00	10000
31	12191210031	12191210031	12/12/2018	08:17:00	10000
32	12191210032	12191210032	12/12/2018	08:17:00	10000
33	12191210033	12191210033	12/12/2018	08:17:00	10000
34	12191210034	12191210034	12/12/2018	08:17:00	10000
35	12191210035	12191210035	12/12/2018	08:17:00	10000
36	12191210036	12191210036	12/12/2018	08:17:00	10000
37	12191210037	12191210037	12/12/2018	08:17:00	10000
38	12191210038	12191210038	12/12/2018	08:17:00	10000
39	12191210039	12191210039	12/12/2018	08:17:00	10000
40	12191210040	12191210040	12/12/2018	08:17:00	10000
41	12191210041	12191210041	12/12/2018	08:17:00	10000
42	12191210042	12191210042	12/12/2018	08:17:00	10000
43	12191210043	12191210043	12/12/2018	08:17:00	10000
44	12191210044	12191210044	12/12/2018	08:17:00	10000
45	12191210045	12191210045	12/12/2018	08:17:00	10000
46	12191210046	12191210046	12/12/2018	08:17:00	10000
47	12191210047	12191210047	12/12/2018	08:17:00	10000
48	12191210048	12191210048	12/12/2018	08:17:00	10000
49	12191210049	12191210049	12/12/2018	08:17:00	10000
50	12191210050	12191210050	12/12/2018	08:17:00	10000

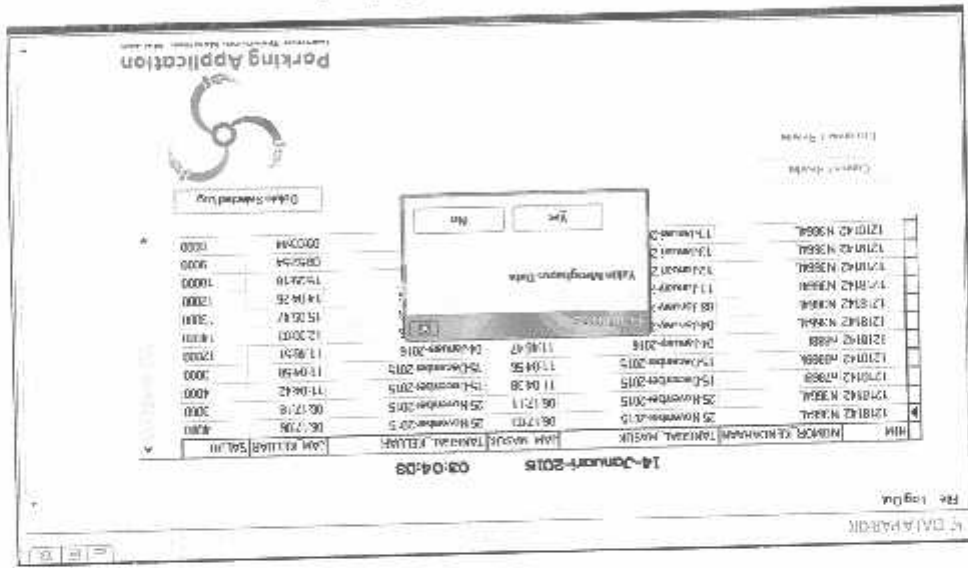
Gambar 4.46 Proses Memasukkan Kartu (Masuk Parkir)



4.3.2.5 Pengujian Form Isi Uang Saldo

Form Isi Saldo digunakan oleh petugas parkir untuk mengisi saldo pada kartu yang nantinya digunakan oleh mahasiswa untuk pembayaran parkir. Tampilan awal form dapat dilihat pada Gambar 4.51.

Gambar 4.50 Prompt Delete Log



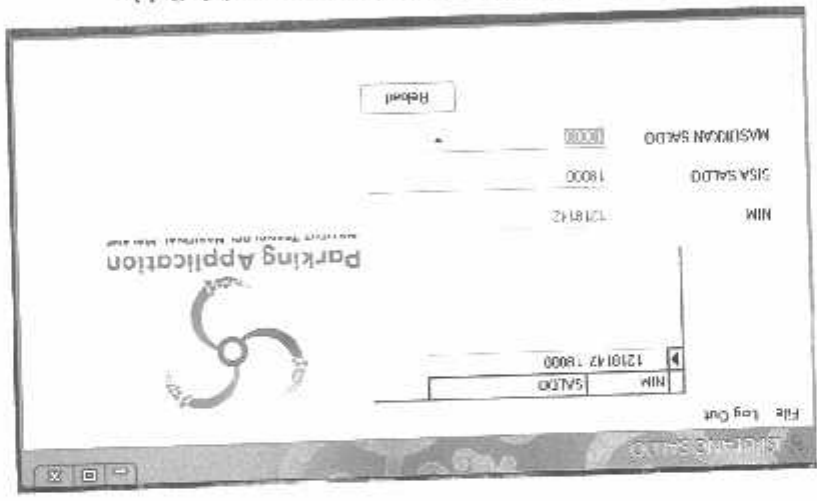
Gambar 4.48.

Selanjutnya adalah pengujian *Delete log* atau *history* pada *Form Data* Parkir. Dengan klik data yang ingin dihapus klik tombol *Delete Selected Log* kemudian akan muncul *prompt* pilih *Yes* untuk menghapus data seperti pada

Gambar 4.49 Pencatatan tanggal keluar, jam keluar serta pengurangan saldo

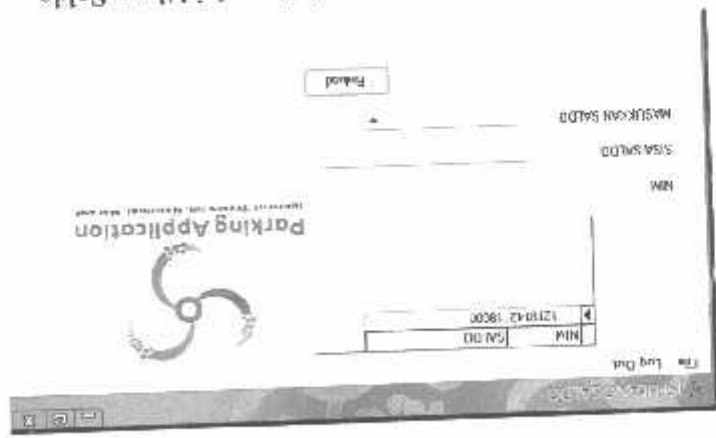


Gambar 4.52 Pemilihan NIM dan Jumlah Saldo



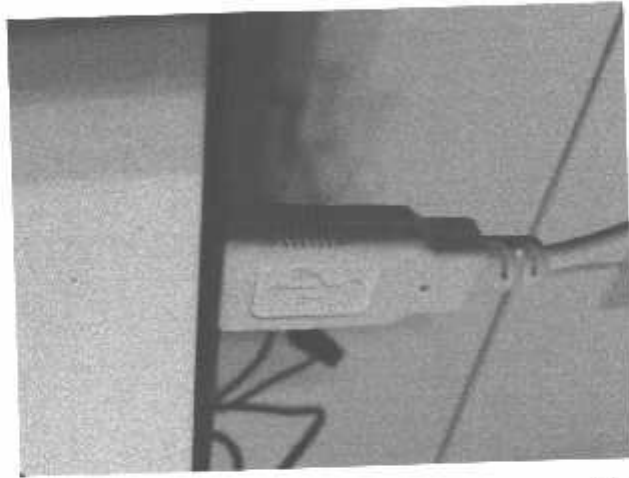
Pengujian *form* dilakukan dengan mengisi saldo yang dicari sesuai NIM. Langkah awal adalah dengan memilih data nim yang akan diisi saldo setelah itu pilih nominal dan klik tombol *Reload* seperti ditunjukkan pada Gambar 4.52 dan Gambar 4.53.

Gambar 4.51 Tampilan Awal Form Isi Ulang Saldo



Selanjutnya pada *form* yang menggunakan fungsi *SmartCard* klik tombol *Connect Reader*, kali ini pengujian dilakukan pada *Form Data Parkir* seperti yang ditunjukkan pada Gambar 4.55.

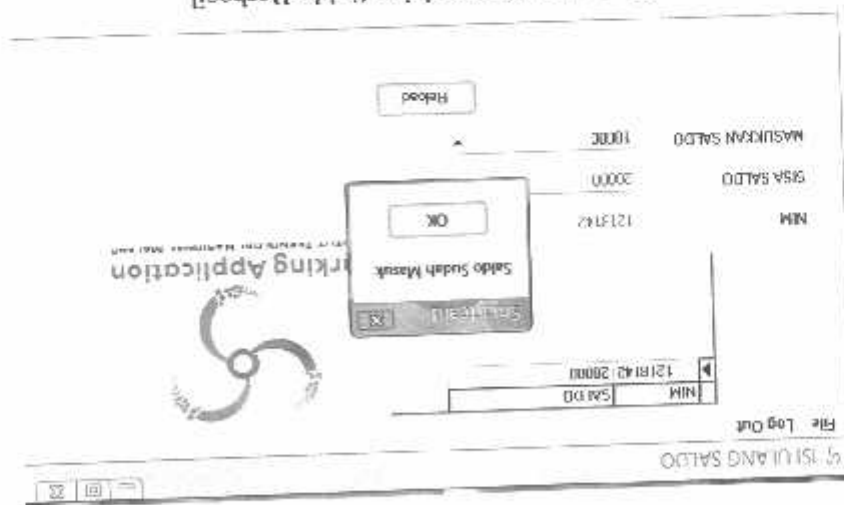
Gambar 4.54 Proses Memasukkan *Reader* Kedalam USB 3.0



Pengujian *Reader ACS38* dilakukan dengan menggunakan *Windows 8.1* dengan Bahasa pemrograman Delphi 7. Pengujian dibagi menjadi 2 tahap yakni tahap pengkoneksian dan tahap diskoneksi. Awal pengujian dilakukan dengan memasukkan *reader* ke USB 3.0 pada *personal computer* seperti yang dapat dilihat pada Gambar 4.54.

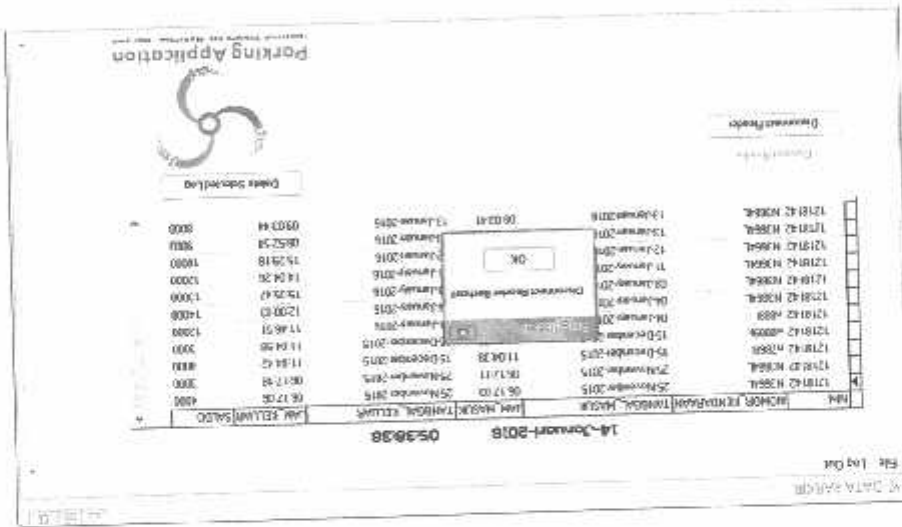
4.3.3 Pengujian *Reader ACS38*

Gambar 4.53 Pengisian Saldo Berhasil



4.3.4 Pengujian Smart Card ISO-7816
 Pengujian Smart Card ISO-7816 dilakukan dengan menggunakan Windows Parkir guna mengetahui apakah Smart Card dapat terdeteksi. Pengujian ditunjukkan pada Gambar 4.57.

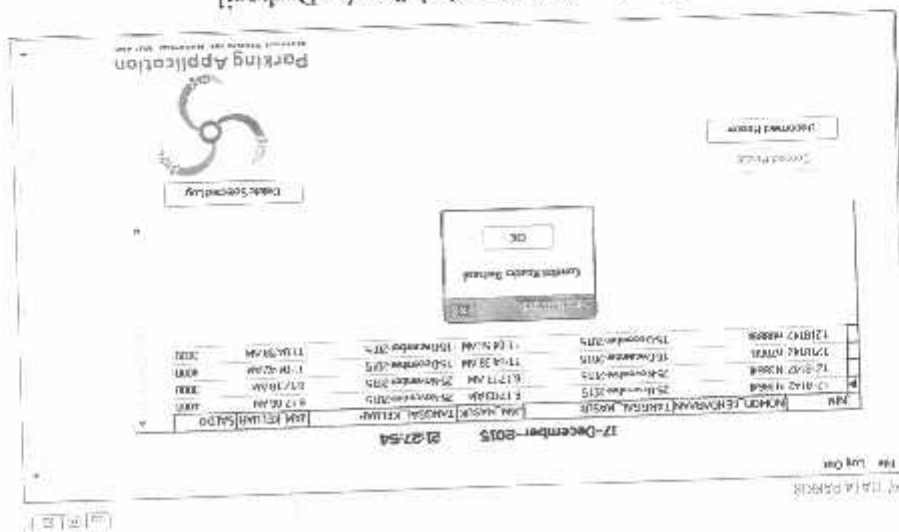
(gambar 4.56 Diskoneksi Reader Berhasil)



4.56.

Untuk melakukan proses koneksi klik tombol *Disconnect Reader*, pengujian dilakukan pada Form Data Parkir seperti yang ditunjukkan pada Gambar

(gambar 4.55 Koneksi Koneksi Berhasil)



No	Fungsi / Form		Berhasil		Hak Akses		
	Ya	Tidak	Super Admin	Admin	Petugas		
1	Form Login Super Admin		√				
	Login		√				
	Logout		√				
2	Form Login Admin		√				
	Login		√				
	Logout		√				
3	Form Login Petugas		√				
	Login		√				
	Logout		√				
4	Form Data Admin			√			
	Login		√				
	Logout		√				
	CRUD		√				

Tabel 4.2 Tabel Pengujian Fungsional

Dalam Tabel Pengujian Fungsional ini merupakan rangkuman hasil – hasil pengujian sebelumnya yang mana terdapat fungsi, form, hak akses serta berhasil tidaknya fungsi tersebut dijalankan pada form yang ada seperti yang dapat dilihat pada Tabel 4.2.

4.3.5 Tabel Pengujian Fungsional

(Gambar 4.57 Deteksi Kartu Berhasil)



4.3.6 Pengujian Reader ACS38 Dengan Tabel Fungsional

Pengujian Reader ACS38 dilakukan dengan 4 macam sistem operasi yang diujikan untuk mengetahui tingkat kompatibilitas dari reader ACS38. Berikut hasil pengujian pada Tabel 4.3.

Dari Tabel Fungsional diatas menunjukkan 20 fungsi pokok meliputi login, logout, CRUD, koneksi serta fungsi pembayaran dapat berjalan dengan baik secara keseluruhan.

				√	-	Read Smart Card
				√	-	Edit Data
8		√	√	√		Form Isi Saldo
				√	-	Read Smart Card
				√	-	Pembayaran
				√	-	Deteksi SmartCard
				√	-	Koneksi Reader
				√	-	Deleted Data
				√	-	Created data
7		√	√	√		Form Data Parkir
				√	-	Write Smart Card
				√	-	Read Smart Card
				√	-	Deteksi Smart Card
				√	-	Koneksi Reader
				√	-	Clear
				√	-	CRUD
6		√	√	√		Form Data Mahasiswa
				√	-	Clear
				√	-	CRUD
5		√	√	√		Form Data Petugas Parkir
				√	-	Clear

Pengujian *User* diterapkan kepada 4 pengguna yang meliputi masing-masing 2 orang petugas parkir di kampus 1 dan kampus 2 ITN Malang dengan rentan usia mulai dari 25 - 32 tahun. Pengujian dilakukan untuk mengetahui apakah

4.3.8 Pengujian *User*

Dari tabel fungsional *Smart Card* diatas menunjukkan bahwa *Smart Card* dengan tipe ISO - 7816 memiliki tingkat kompatibilitas yang sangat baik dengan 4 sistem operasi yang diuji coba sedangkan tipe ST:4442 sama sekali tidak kompatibel baik dengan *reader ACS38* maupun dengan sistem operasi yang diuji.

ISO 7816	Fungsi
√	Verify Smart Card
√	Select Data Smart Card
√	Read Data Smart Card
√	Write Data Smart Card

Tabel 4.4 Tabel Fungsional *Smart Card*

Pengujian *Smart Card* dilakukan dengan menguji beberapa fungsi diantaranya fungsi verifikasi kartu, pemilihan data dalam kartu, baca data kartu serta tulis data di dalam kartu. Berikut hasil pengujian pada Tabel 4.4.

4.3.7 Pengujian *Smart Card* Dengan Tabel Fungsional

diuji cobakan.

Dari tabel fungsional *Reader ACS38* diatas menunjukkan bahwa *Reader ACS38* memiliki tingkat kompatibilitas yang baik dengan 4 sistem operasi yang

Berhasil		Sistem Operasi
Ya	Tidak	
√	-	Windows XP
√	-	Windows 7
√	-	Windows 8.1
√	-	Windows 10

Tabel 4.3 Tabel Fungsional *Reader ACS38*

Dari hasil 4 pengujian diatas yang dilakukan terdapat selisih waktu 3,75 detik antara penggunaan aplikasi dengan tanpa aplikasi yang mana jika menggunakan aplikasi proses parkir bisa lebih cepat.

Kondisi	Waktu yang diperlukan				Rata - rata waktu
	Pengujian 1	Pengujian 2	Pengujian 3	Pengujian 4	
Manual Tanpa Aplikasi	5,28 detik	6,13 detik	4,57 detik	6,32 detik	5,57 detik
Menggunakan Aplikasi Smart Card	1,55 detik	2,03 detik	2,12 detik	1,55 detik	1,81 detik

Tabel 4.6 Tabel Pengujian Aplikasi

Pengujian Aplikasi dilakukan untuk mengetahui seberapa waktu yang didapat jika menggunakan ataupun tidak menggunakan aplikasi ini. Untuk hasil pengujian dapat dilihat pada Tabel 4.6.

4.3.9 Pengujian Aplikasi

Dari hasil pengujian diatas 3 dari 4 orang pengguna mengatakan aplikasi Smart Card ini mudah digunakan sedangkan 1 orang mengatakan sangat mudah digunakan.

Kriteria	Pengguna			
	Sangat Mudah	Mudah	Sulit	Sangat Sulit
Pengguna 1	-	√	-	-
Pengguna 2	√	-	-	-
Pengguna 3	-	√	-	-
Pengguna 4	-	√	-	-

Tabel 4.5 Tabel Pengujian User

pada Tabel 4.5. aplikasi dapat dengan mudah diakses dan digunakan dengan 4 pilihan kriteria yakni mudah, cukup mudah, sulit dan sangat sulit. Untuk hasil pengujian dapat dilihat

Dari beberapa simpulan yang telah diambil, maka dapat dikemukakan saran-saran yang akan sangat membantu untuk pengembangan *Smart Card*.

1. *Smart Card* yang digunakan saat ini merupakan tipe *Contact Smart Card* untuk pengembangan dapat digunakan *Contactless Smart Card* atau pun menggunakan RFID (Radio Frequency Identification).
2. Aplikasi saat ini hanya di khususkan untuk pembayaran parkir, untuk kedepannya *Smart Card* ini dapat dikembangkan untuk segala pembayaran seperti koperasi, kantin dll.

5.2. Saran

Berdasarkan hasil pengujian yang dilakukan pada aplikasi ini, maka dapat disimpulkan :

1. Dari hasil pengujian dapat diketahui aplikasi ini dapat berfungsi baik untuk deteksi *smart card* dan koneksi *Reader*, fungsi CRUD serta pembayaran parkir menggunakan *smart card* dengan tingkat keberhasilan mencapai 100%.
2. Dari tabel pengujian fungsional *Reader ACS38* tingkat kompatibilitas dengan 4 sistem operasi yang diuji, *Reader ACS38* menunjukkan tingkat kompatibilitas 100%.
3. Dari pengujian 2 buah *Smart Card* tipe ISO-7816 dan SLF4442 dapat disimpulkan tipe SLF4442 tidak kompatibel dengan *Reader ACS38* dengan 4 sistem operasi yang diujikan.
4. Berdasarkan label pengujian *user* 75% pengguna merasa mudah menggunakan aplikasi pembayaran parkir dengan *smart card* ini sedangkan 25% pengguna mengatakan sangat mudah menggunakannya. Dengan ini maka dapat disimpulkan bahwa aplikasi ini mudah digunakan oleh petugas parkir.
5. Berdasarkan pengujian yang dilakukan untuk mengetahui efisiensi waktu didapat bahwa dengan menggunakan aplikasi *smart card* terdapat efisiensi waktu sebesar $5,57/1,81 * 100\% = 9,47\%$.

BAB V PENUTUP

DAFTAR PUSTAKA

- Adella, Jimmy Seliawan, 2011, *Implementasi Customer Relationship Management (CRM) pada Sistem Reservasi Hotel berbasis Website dan Desktop*; Vol 6 No.2 2011, (September), 113-126
- Iidayah, Afti Nir, Wahyu Setia Budi, 2009, *Pembinaan Perangkat Lunak Berbasis Borland Delphi 7.0 Untuk Pengolahan Citra Radiografi Sinar-X*; Volume 26 2009, (November)
- Swastika, I Putu Agus, Siti Saibah Pua Lukka, Yanno Dwi Ananda, 2010, *Rancangan Bangun Aplikasi Smart Card Interface*; Vol 1 No.1 2010, (Desember), 32- 35
- Ramdhani, Dendhi, 14 desember 2015, *Bandung Smart Card Bisa Dipakai untuk Belanja dan Bayar Parkir*; *Bandung Smart Card Bisa Dipakai untuk Belanja dan Bayar Parkir*; *Bandung Smart Card Bisa Dipakai untuk Belanja dan Bayar Parkir* (9 Desember 2015)
- Setiawan, Sakina Rakhma Diah, 12desember 2013, *Jamsostek Ganti Kartu Peserta dengan "Smart Card"*; http://bisniskeuangan.kompas.com/read/2013/12/12/1935340/Jamsostek.Ganti.Kartu.Peserta.dengan.Smart.Card.7utm_campaign=related&utm_medium=bp-kompas&utm_source=news& (10 Desember 2015)

LAMPIRAN





BERITA ACARA UJIAN KOMPRESI
FAKULTAS TEKNOLOGI INDUSTRI

Nama : Yohanes Vianey Ricky Nugroho
NIM : 12.18.142
Program Studi : Teknik Informatika S-1
Judul Skripsi : Rancang Bangun Aplikasi *Smart Card* Untuk Informasi Mahasiswa Dan Pembayaran Parkir Mahasiswa

Dipertahankan dihadapan Majelis Pengungsi Skripsi Jenjang Strata Satu (S1)

pada:
Hari : Sabtu
Tanggal : 16 Januari 2016
Nilai : 80,34 (A)

Panitia Ujian Skripsi
Ketua Majelis Penguji

Joseph Dedy Irawan, ST. MT.
NIP. 197404162005011002

Dosen Penguji I

Joseph Dedy Irawan, ST. MT.
NIP. 197404162005011002

Dosen Penguji II

Abdul Wahid, M.Pd.I
NIP. 1031500507



FORMULIR PERBAIKAN SKRIPSI

Nama : Yohanes Vianey Ricky Nugroho
 NIM : 12.18.142
 Program Studi : Teknik Informatika S-1
 Judul Skripsi : Rancang Bangun Aplikasi *Smart Card* Untuk Informasi Mahasiswa Dan Pembayaran Parkir Mahasiswa

Tanggal	Pengujian	Uraian	Paraf
16 Januari 2016	I	- Password Admin Bisa Diganti - Password di Enkripsi - Baca Memory	<i>W</i>
16 Januari 2016	II	- Perumusan Masalah disesuaikan Tujuan	<i>W</i>

Anggota Penguji

Dosen Penguji I

Joseph Dedy Irawan, ST, MT.
NIP. 197404162005011002

Dosen Pembimbing I

Suryo Adi Wibowo, ST, MT.
NIP. P/1031000438

Mengetahui,

Dosen Penguji II

Abdul Wahid, M.Pd.I
NIP. 1031500507

Dosen Pembimbing II

Febriana Santi W, S.Kom., M.Kom.
NIP.P. 1031000425

Surat Bimbingan Skripsi



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. Sampingan Singaperbangsa No 2 Telp. (0341) 551431 (Hunting), Fax (0341) 552015 Malang 65145
Kampus II : Jl. Raya Karang, Km 2 Telp. (0341) 417536 Fax (0341) 417634 Malang

Malang, 23 Oktober 2015

Nomor : ITN-5931/INF/TA/2015

Lampiran : -

Perihal : Bimbingan Skripsi

Kepada :

Yth. Bpk/Ibu Saryo Adi Wibowo, ST, MT
Dosen Pembina Program Studi Teknik Informatika S-1
Institut Teknologi Nasional
Malang

Dengan Hormat,

Sesuai dengan permohonan dan persetujuan dalam proposal skripsi untuk mahasiswa :

Nama : YOHANES VIANEY RICKY NUGROHO
Nim : 1218142
Prodi : Teknik Informatika S-1
Fakultas : Teknologi Industri

Maka dengan ini pembimbingan kami sampaikan sepenuhnya kepada Saudara/i selama waktu 6 (enam) bulan, dihitung mulai tanggal :

23 Oktober 2015 S/D 23 Maret 2016

Sebagai satu syarat untuk mencermah Ujian Akhir Sarjana Teknik, Program Studi Teknik Informatika S-1. Demikian agar maklum dan atas perhatiannya kami sampaikan terima kasih.









Mengucapkan
Program Studi Teknik Informatika S-1
K e r i a,

Joseph Dedy Arwan, ST., MT.
NIP. 197404162005021002

Form 5-4a

BUKTI BIMBINGAN SKRIPSI

Nama : Yohanes Vianey Ricky Nugroho
 NIM : 12.18.142
 Masa Bimbingan : 23 Oktober 2015 s/d 23 Maret 2016
 Judul Skripsi : Rancang Bangun Aplikasi *Smart Card* Untuk Informasi Mahasiswa Dan Pembayaran Parkir Mahasiswa

No	Tanggal	Uraian	Paraf Pembimbing
1	18/11/2015	Interface GUI BAB II Library	
2	21/11/2015	BAB III, Flowchart	
3	23/11/2015	Input Data Kartu, Fungsional	
4	4/1/2016	Penjelasan Form	
5	11/1/2016	Konfigurasi	
6	13/1/2016	Sitasi	
7	14/1/2016	Desain Sistem	
8	15/1/2016	Tabel database, Relasi Tabel	

Malang, Januari 2016
 Dosen Pembimbing I

Survo Adi Wibowo, ST, MT
 NIP. P. 1031000438



Nomor : ITN-593/I/INF/TA/2015

Lampiran : -

Perihal : *Rimbungan Skripsi*

Kepada : Yth. Bpk/Ibu Rektora Saati W, S.Kom, M.Kom

Dosen Pembina Program Studi Teknik Informatika S-1

Institut Teknologi Nasional

Malang

Dengan Hormat,

Sesuai dengan permohonan dan persetujuan dalam proposal skripsi untuk mahasiswa :

Name : YOHANES VIANEY RICKY NUGROHO

Nim : 1218142

Prodi : Teknik Informatika S-1

Fakultas : Teknologi Industri

Maka dengan ini pembahasan kami serahkan sepenuhnya kepada Saudara/i selama waktu 6 (enam) bulan, terhitung mulai tanggal :

23 Oktober 2015 S/D 23 Maret 2016

Sebagai satu syarat untuk mencapai Ujian Akhir Sarjana Teknik, Program Studi Teknik Informatika S-1. Demikian agar maklum dan atas perhatiannya kami sampaikan terima kasih.

Mengestahui
Program Studi Teknik Informatika-S-1
Kota,

Joseph Deby Irawan, ST., MT.
NIP : 197404162005021002

Form 5-4a

BUKTI BIMBINGAN SKRIPSI

Nama : Yohanes Vianey Ricky Nugroho
 NIM : 12.18.142
 Masa Bimbingan : 23 Oktober 2015 s/d 23 Maret 2016
 Judul Skripsi : Rancang Bangun Aplikasi *Smart Card* Untuk Informasi Mahasiswa Dan Pembayaran Parkir Mahasiswa

No	Tanggal	Uraian	Paraf Pembimbing
1	18/11/2015	Latar Belakang, Kesimpulan	hr
2	21/11/2015	Rumusan Masalah, Daftar Pustaka	hr
3	10/12/2015	GUI	hr
4	18/12/2015	Flowchart	hr
5	4/1/2016	Penjelasan Form	hr
6	23/11/2015	Acc Seminar Progress	hr
7	13/1/2016	Revisi Laporan	hr
8	14/1/2016	Acc Kompre	hr

Malang,
 Januari 2016

Dosen Pembimbing II



Febrina Santi W, S.Kom, M.Kom.
 NIP.P. 1031000425

RANCANG BANGUN APLIKASI SMART CARD UNTUK INFORMASI MAHASISWA

DAN PEMBAYARAN PARKIR MAHASISWA

Pengujian Parkir Kampus I

No	Pengujian	Nama Petugas	Lama Waktu Service
1	Manual	ALI	5,28
2	Manual	ALI	6,13
3	Manual	ALI	4,57
4	Manual	ALI	6,32

Petugas Parkir

Malang, 6 Januari 2016

RANCANG BANGUN APLIKASI SMART CARD UNTUK INFORMASI MAHASISWA

DAN PEMBAYARAN PARKIR MAHASISWA

Pengujian Parkir Kampus I

No	Pengujian	Nama Petugas	Lama Waktu Service
1	Aplikasi	ALI	1,66
2	Aplikasi	ALI	2,03
3	Aplikasi	ALI	2,12
4	Aplikasi	ALI	1,55

Petugas Parkir

Malang, 6 Januari 2016

Source Code Form Data Parkir

```
var
data parkir: Tdata_parkir;
koneksi: WordBool;
Ib: Boolean;
Implementation
uses mahasiswa, home, unit2, unit1, login_admin, login_petugas,
super,
admin, parkir, form_saldo, strutils;
$R +.dfm;
function hexToInt(s : string) : Int64;
begin
if (s <> '') and (s[1] <> '0') then
result := strtoint64('0' + s)
else
result := strtoint64(s);
end;
procedure Tdata_parkir.ExitClick(Sender: TObject);
begin
if (Application.MessageBox('Yakin Keluar Aplikasi',
'Kontirmast', MB_YESNOCANCEL) = ID_Yes) then
Application.Terminate;
end;
end;
procedure Tdata_parkir.DataMahasiswaClick(Sender: TObject);
begin
if bt_connect.Enabled then
begin
data_mhs.Show;
hide;
end
else
begin
showMessage('Harap Disconnect');
showMessage('Masukkan Kartu Lalu Tekan Disconnect');
end;
end;
end;
end;
procedure Tdata_parkir.LogOutClick(Sender: TObject);
begin
Application.MessageBox('Yakin
Kontirmast', MB_YESNO) = ID_Yes) then
Log
Out
end;
hide;
```

```

form_home.Show;
if admin_login_lb_flag_admin.Caption='admin' then
begin
admin_login_lb_flag_admin.Caption:=0;
end
else
end
if petugas_login_lb_flag_petugas.Caption='petugas' then
begin
petugas_login_lb_flag_petugas.Caption:=0;
end
else
begin
if super_admin_lb_super.Caption='super' then
super_admin_lb_super.Caption:=0;
end;
end;
end;
end;
end;

procedure Tdata_parkir.Timer1Timer(Sender: TObject);
begin
lb_tgl.Caption:=FormatDateTime('dd-mmm-yy',now);
lb_jam.Caption:=FormatDateTime('hh:nn:ss',now);
end;

procedure Tdata_parkir.connect_reader_execute(Sender: TObject);
begin
SmartCardToolSetSERVER_ActiveX1.ConnectSCARD;
koneksi:=SmartCardToolSetSERVER_ActiveX1.IsConnectedSCARD;
GetReadersList_act.Execute;
if koneksi then
begin
ShowMessage('Koneksi Reader Berhasil!');
dbgrid_data_parkir.Visible:=True;
bt_disconnect.Enabled:=False;
lb_tgl.Visible:=True;
lb_jam.Visible:=True;
bt_delete.Enabled:=True;
if dm_1.tb_mhs.FieldByName('NIM').AsString="" then
ShowMessage('Data Mahasiswa Kosong');
end;
bt_connect.Enabled:=False;
end;
end;

```

```

procedure Tdata_parkir.Disconnect_reader_execute(Sender: TObject);
begin
SmartCardToolsetSERVER_ActiveX1.IsConnectedSCARD;
SmartCardToolsetSERVER_ActiveX1.DisconnectSCARD;
ShowMessage('Pemutusan Koneksi Reader Berhasil');
bt_connect.Enabled:=True;
bt_disconnect.Enabled:=False;
end;

procedure Tdata_parkir.SmartCardToolsetSERVER_ActiveX1.CardDetected(
ASender: TObject; var ReaderName: WideString);
begin
bt_disconnect.Enabled:=True;
Timer2.Enabled:=True;
end;

procedure Tdata_parkir.bt_deleteClick(Sender: TObject);
begin
if (Application.MessageBox('Yakin Menghapus Data', 'Konfirmasi', MB_YESNO)=ID_Yes) then
begin
dm_2.tb_data_parkiran.Delete;
end
end;

procedure Tdata_parkir.SmartCardToolsetSERVER_ActiveX1.Exit(
Sender: TObject);
begin
SmartCardToolsetSERVER_ActiveX1.DisconnectSCARD;
Application.Terminate;
end;

procedure Tdata_parkir.SmartCardToolsetSERVER_ActiveX1.ConnectedSCARD(
Sender: TObject);
begin
dbgrid_data_parkir.Visible:=True;
bt_connect.Enabled:=False;
bt_disconnect.Enabled:=True;
bt_delete.Enabled:=True;
lb_tgl.Visible:=True;
lb_jam.Visible:=True;
end;

procedure Tdata_parkir.FormActivate(Sender: TObject);
begin
end;

```



```

end
else
begin
ShowMessage('Harap Disconnect');
ShowMessage('Masukkan Kartu Lalu Tekan Disconnect');
end;
end;
end;

procedure Tdata_parkir.isSaldo1Click(Sender: TObject);
begin
if bt_connect.Enabled then
begin
Hide;
saldo.Show;
end
else
begin
ShowMessage('Harap Disconnect');
ShowMessage('Masukkan Kartu Lalu Tekan Disconnect');
end;
end;
end;

procedure Tdata_parkir.Timer2Timer(Sender: TObject);
begin
if loading_card.Position=50 then
begin
loading_card.Position:=loading_card.Position+50;
end;
end;
end;

procedure Tdata_parkir.GetReadersList_actExecute(Sender: TObject);
var
r1:widestring;
r2:string;
end;
begin
timer2.Enabled:=false;
read_act.Execute;
loading_card.Position:=0;
end
else
begin
loading_card.Position:=100 then
end;
if loading_card.Position=100 then
begin
timer2.Enabled:=false;
read_act.Execute;
loading_card.Position:=0;
end
else
begin
loading_card.Position:=loading_card.Position+50;
end;
end;
end;
end;

```

Source Code Form Data Mahasiswa

```
var  
data_mhs: TData_mhs;  
Reader: TReader;  
Opened : boolean;  
Connected : boolean;  
Implementation  
uses Math, TypInfo, home, data, admin, unit1, unit2, login_admin,  
login_petugas, suger, form_saldo, strutils, combj;  
($R * dfm)  
function StringToHex(s: String): String;  
var I: Integer;  
begin  
Result := '';  
for I := 1 to length (s) do  
Result := Result + IntToHex(ord(s[I]), 2);  
end;  
function HexToString(H: String): String;  
var I: Integer;  
begin  
Result := '';  
for I := 1 to length (H) div 2 do  
Result := Result + Char(StrToInt('s'+Copy(H, (I-1)*2+1, 2)));  
end;  
function hexToInt(s: string) : Integer;  
begin  
if (s <> '' ) and (s[1] <> '$') then  
result := strToInt64('$'+ s )  
else  
result := strToInt64(s);  
end;  
procedure Tdata_mhs.bt_tambahClick(Sender: TObject);  
begin  
ed_nim.Enabled:=True;  
ed_nama.Enabled:=True;  
ed_alamat.Enabled:=True;  
ed_ttl.Enabled:=True;  
ed_fakultas.Enabled:=True;  
ed_jurusan.Enabled:=True;  
ed_angkasan.Enabled:=True;  
ed_no_telp.Enabled:=True;  
bt_tambah.Enabled:=False;  
bt_edit.Enabled:=False;  
bt_hapus.Enabled:=True;
```

```
ed_nim.Text:='',
ed_nama.Text:='',
ed_alamat.Text:='',
ed_ttl.Text:='',
ed_fakultas.Text:='',
ed_jurusan.Text:='',
ed_angkatan.Text:='',
ed_no_telp.Text:='',
end;

procedure Tdata_mhs.SmartCardToolset_COM_ServerCardDetected(
  ASender: TObject; var ReaderName: WideString)
begin
  ShowMessage('Kartu masuk');
  ed_nim.Enabled:=True;
  ed_nama.Enabled:=True;
  ed_alamat.Enabled:=True;
  ed_ttl.Enabled:=True;
  ed_fakultas.Enabled:=True;
  ed_jurusan.Enabled:=True;
  ed_angkatan.Enabled:=True;
  ed_no_telp.Enabled:=True;
  bt_tambah.Enabled:=True;
  bt_simpan.Enabled:=True;
  bt_edit.Enabled:=True;
  bt_hapus.Enabled:=True;
end;

procedure Tdata_mhs.SmartCardToolset_COM_ServerCardWalt(ASender:
  TObject; var ReaderName: WideString);
begin
  ShowMessage('Masukkan Kartu Dulu');
  ed_nim.Enabled:=False;
  ed_nama.Enabled:=False;
  ed_alamat.Enabled:=False;
  ed_ttl.Enabled:=False;
  ed_fakultas.Enabled:=False;
  ed_jurusan.Enabled:=False;
  ed_angkatan.Enabled:=False;
  ed_no_telp.Enabled:=False;
  bt_tambah.Enabled:=False;
  bt_simpan.Enabled:=False;
  bt_edit.Enabled:=False;
  bt_hapus.Enabled:=False;
end;

procedure Tdata_mhs.SmartCardToolset_COM_ServerLibBeforeOpening(
  ASender: TObject; var ReaderName: WideString);
```

```
begin
ed_nim.Enabled:=False;
ed_nama.Enabled:=False;
ed_alamat.Enabled:=False;
ed_tgl.Enabled:=False;
ed_fakultas.Enabled:=False;
ed_jurusan.Enabled:=False;
ed_angkatan.Enabled:=False;
ed_no_telp.Enabled:=False;
bt_tambah.Enabled:=False;
bt_simpan.Enabled:=False;
bt_edit.Enabled:=False;
bt_hapus.Enabled:=False;
end;

procedure Tdata_mhs.bt_simpanClick(sender: TObject);
begin
if ed_nim.GetTextLen=0 then
ShowMessage('Lengkap! data terlebi dahulu');
if (ed_nim.GetTextLen>0) then
begin
lb_mhs.Append;
tb_mhs.FieldName('NIM').AsString:=ed_nim.Text;
tb_mhs.FieldName('NAMA').AsString:=ed_nama.Text;
tb_mhs.FieldName('ALAMAT').AsString:=ed_alamat.Text;
tb_mhs.FieldName('TGL').AsString:=ed_tgl.Text;
tb_mhs.FieldName('FAKULTAS').AsString:=ed_fakultas.Text;
tb_mhs.FieldName('JURUSAN').AsString:=ed_jurusan.Text;
tb_mhs.FieldName('ANGKATAN').AsString:=ed_angkatan.Text;
tb_mhs.FieldName('NO_TELP').AsString:=ed_no_telp.Text;
lb_mhs.FieldName('SALDO').AsString:=ed_saldo.Text;
tb_mhs.FieldName('NOMOR_KENDARAAN').AsString:=ed_nomor_kendaraan
n.Text;
tb_mhs.Insert;
bt_tambah.Enabled:=True;
bt_simpan.Enabled:=False;
saldo.tb_saldo.Append;
saldo.tb_saldo.FieldName('NIM').AsString:=ed_nim.Text;
saldo.tb_saldo.FieldName('SALDO').AsString:=ed_saldo.Text;
saldo.tb_saldo.Insert;
tb_mhs.Next;
saldo.tb_saldo.Next;
ShowMessage('Data Berhasil Dimasukkan');
End;
ed_nim.Text:= '';
ed_nama.Text:= '';
ed_alamat.Text:= '';
ed_tgl.Text:= '';
ed_fakultas.Text:= '';
```

```
ed_jurusan.Text:='';
ed_angkatan.Text:='';
ed_no_telp.Text:='';
end;

procedure Tdata_mhs.bt_editclick(Sender: TObject);
begin
  tb_mhs.Edit;
  tb_mhs.FieldName('NIM').AsString:=ed_nim.Text;
  tb_mhs.FieldName('NAMA').AsString:=ed_nama.Text;
  tb_mhs.FieldName('ALAMAT').AsString:=ed_alamat.Text;
  tb_mhs.FieldName('TTL').AsString:=ed_ttl.Text;
  tb_mhs.FieldName('FAKULTAS').AsString:=ed_fakultas.Text;
  tb_mhs.FieldName('JURUSAN').AsString:=ed_jurusan.Text;
  tb_mhs.FieldName('ANGKATAN').AsString:=ed_angkatan.Text;
  tb_mhs.FieldName('NO_TELP').AsString:=ed_no_telp.Text;
  tb_mhs.FieldName('SALDO').AsString:=ed_saldo.Text;
  tb_mhs.FieldName('NOMOR_KENDARAAN').AsString:=ed_nomor_kendaraan
  n.Text;
  tb_mhs.Post;
  ed_nim.Text:='';
  ed_nama.Text:='';
  ed_alamat.Text:='';
  ed_ttl.Text:='';
  ed_fakultas.Text:='';
  ed_jurusan.Text:='';
  ed_angkatan.Text:='';
  ed_no_telp.Text:='';
  dbgrid_mhs.Enabled:=True;
  cd_nim.Enabled:=True;
  ed_nama.Enabled:=True;
  ed_alamat.Enabled:=True;
  ed_ttl.Enabled:=True;
  ed_fakultas.Enabled:=True;
  ed_jurusan.Enabled:=True;
  ed_angkatan.Enabled:=True;
  ed_no_telp.Enabled:=True;
end;

procedure Tdata_mhs.bt_hapusclick(Sender: TObject);
begin
  if Application.MessageBox('Yakin
  Data', 'Konfirmasi', MB_YESNO)=ID_Yes) then
    menghapus
  begin
    tb_mhs.Delete;
    saldo.tb_saldo.Delete;
    if (saldo.tb_saldo.FieldName('NIM').AsString='')
    or (tb_mhs.FieldName('SALDO').AsString='') then
      or
    end;
  end;
end;
```

```

begin
  ShowMessage('DATA KOSONG');
end;
end;
end;

ed_nim.Text:='';
ed_nama.Text:='';
ed_alamat.Text:='';
ed_rtl.Text:='';
ed_fakultas.Text:='';
ed_jurusan.Text:='';
ed_angkatan.Text:='';
ed_no_kolp.Text:='';
end;

procedure Tdata_mhs.Formativate(sender: TObject);
var
  con:String;
begin
  Timer1.Enabled:=True;
  sProgressbar1.Visible:=False;
  dgGrid_mhs.Visible:=False;
  IF super_admin_lb_super.Caption='0' then
  begin
    ADMIN1.Visible:=False;
  IF admin_login_lb_flag_admin.Caption='0' then
  begin
    ADMIN1.Visible:=False;
  IF admin_login_lb_flag_admin.Caption='admin' then
  begin
    ADMIN1.Visible:=True;
    PetugasParked1.Visible:=True;
  end;
  else
  begin
    ADMIN1.Visible:=False;
  end
  else
  begin
    ADMIN1.Visible:=True;
    PetugasParked1.Visible:=True;
  end;
  end;
  end;
  dm_1.tb_mhs.Refresh;
ed_nim.Enabled:=False;
ed_nama.Enabled:=False;
ed_alamat.Enabled:=False;
ed_rtl.Enabled:=False;
end;

```

```
procedure Tdata_mhs.dbgrid_mhsCellClick(Column: TColumn);
begin
  ed_nim.Enabled:=True;
  ed_nama.Enabled:=True;
  ed_alamat.Enabled:=True;
  ed_ttl.Enabled:=True;
  ed_fakultas.Enabled:=True;
  ed_jurusan.Enabled:=True;
  ed_angkatan.Enabled:=True;
  ed_no_telp.Enabled:=True;
  with tb_mhs do
    ed_nim.Text:=FieldValues['NIM'];
    ed_nama.Text:=FieldValues['NAMA'];
    ed_alamat.Text:=FieldValues['ALAMAT'];
    ed_ttl.Text:=FieldValues['TTL'];
    ed_fakultas.Text:=FieldValues['FAKULTAS'];
    ed_jurusan.Text:=FieldValues['JURUSAN'];
    ed_angkatan.Text:=FieldValues['ANGKATAN'];
    ed_no_telp.Text:=FieldValues['NO_TELP'];
  end;
  bt_clear.Enabled:=True;
  bt_edit.Enabled:=True;
end;

procedure Tdata_mhs.bt_clearClick(Sender: TObject);
begin
  ed_nim.Text:='';
  ed_nama.Text:='';
  ed_alamat.Text:='';
  ed_ttl.Text:='';
  ed_fakultas.Text:='';
  ed_jurusan.Text:='';
  ed_angkatan.Text:='';
  ed_no_telp.Text:='';
end;

procedure Tdata_mhs.bt_clearClick(Sender: TObject);
begin
  ed_nim.Text:='';
  ed_nama.Text:='';
  ed_alamat.Text:='';
  ed_ttl.Text:='';
  ed_fakultas.Text:='';
  ed_jurusan.Text:='';
  ed_angkatan.Text:='';
  ed_no_telp.Text:='';
end;

procedure Tdata_mhs.ed_nimClick(Sender: TObject);
begin
  if (ed_nim.GetTextLen>0) then
    bt_clear.Enabled:=True;
end;

procedure Tdata_mhs.btnExitClick(Sender: TObject);
begin
  if (Application.MessageBox('Yakin Keluar dari Aplikasi',
  'Konfirmasi',MB_YESNOCANCEL)=ID_Yes) then
    begin
      Application.Terminate;
    end;
end;
end;
```


Source Code Form Saldo

```
function StringToHex(s: String): String;
begin
  var I: Integer;
  Result:= '';
  for I := 1 to length (s) do
    Result:= Result+IntToHex(ord(S[I]),2);
  end;
end;

function HexToString(H: String): String;
begin
  var I: Integer;
  Result:= '';
  for I := 1 to length (H) div 2 do
    Result:= Result+Char(StrToInt('$'+Copy(H, (I-1)*2+1,2)));
  end;
end;

function hexToInt(s : string) : Int64;
begin
  if s <> '' and (s[1] <> '$') then
    result := strToInt64('$' + s )
  else
    result := strToInt64(s);
  end;
end;

procedure Saldo.FormActivate(Sender: TObject);
begin
  tb_saldo.Edit;
  tb_saldo.FieldByName('NIM').AsString:=dm_1.tb_mhs.FieldValues['NI
M'];
  tb_saldo.Post;
  if dm_2.tb_saldo.FieldValues['SALDO']='0' then
    begin
      showMessage('Saldo Anda 0 Rupiah Harap Lakukan Isi Ulang');
      ed_sisa.Text:=dm_2.tb_saldo.FieldValues['SALDO'];
    end;
  ed_sisa.Enabled:=False;
  if (super_admin.ed_username.Text='super_ricky')
  and (super_admin.ed_password.Text='1994') then
    begin
      DataAdmin1.Visible:=True;
    end;
  Hide;
end;
else
begin
  DataAdmin1.Visible:=False;
end;
if admin.Login.Lb flag_admin.Caption='admin' then
```

```
begin
  DataRetugas1.Visible:=True;
  Hide;
end
else
begin
  DataRetugas1.Visible:=False;
end;
end;
procedure Tsaldo.cb_nimclick(Sender: TObject);
begin
  ed_sisa.Text:=dm_1.lb_mhs.FieldValues['SALDO'];
end;
procedure Tsaldo.ReloadClick(Sender: TObject);
var a,b,c:Integer;
begin
  a:=StrToInt(ed_sisa.Text);
  b:=StrToInt(ed_saldo_new.Text);
  c:=a+b;
  ed_sisa.Text:=IntToStr(c);
  data_mhs.tb_mhs.Edit;
  data_mhs.tb_mhs.FieldByName('SALDO').AsString:=IntToStr(c);
  data_mhs.tb_mhs.Post;
  tb_saldo.Edit;
  tb_saldo.FieldByName('SALDO').AsString:=IntToStr(c);
  tb_saldo.Post;
  dm_2.tb_data_parkiran.Edit;
  dm_2.tb_data_parkiran.FieldByName('SALDO').AsString:=-IntToStr(c);
  dm_2.tb_data_parkiran.Post;
  dm_2.tb_data_parkiran.FieldByName('SALDO').AsString:=IntToStr(c) then
  begin
    ShowMessage('Saldo sudah Masuk');
    ed_saldo_new.Text:='';
  end;
end;
Reload.Enabled:=False;
end;
procedure Tsaldo.ParkirClick(Sender: TObject);
begin
  Hide;
  data_parkir.Show;
end;
procedure Tsaldo.ExitClick(Sender: TObject);
begin
end;
```

```

IF (Application.MessageBox('Yakin Keluar Dari Aplikasi?',
'Konfirmasi', MB_YESNOCANCEL) = ID_Yes) then
begin
Application.Terminate;
end;
end;
procedure Tsaldo.DataMahasiswaClick(Sender: TObject);
begin
Hides;
data_mhs.show;
end;
procedure Tsaldo.DataAdminClick(Sender: TObject);
begin
Hides;
admin_form.show;
end;
end;
procedure Tsaldo.logoutClick(Sender: TObject);
begin
log
begin
Application.MessageBox('Yakin
'Konfirmasi', MB_YESNOCANCEL) = ID_Yes) then
Hides;
form_home.show;
if admin_login.lb_flag_admin.Caption='admin' then
begin
admin_login.lb_flag_admin.Caption:='0';
end
else
if petugas_login.lb_flag_petugas.Caption='petugas' then
begin
petugas_login.lb_flag_petugas.Caption:='0';
end
else
if super_admin.lb_super.Caption='super' then
begin
super_admin.lb_super.Caption:='0';
end;
end;
end;
end;
end;
end;
procedure Tsaldo.DataGridCellClick(Column: TColumn);
begin
ed_gisa.Text:=lp_saldo.FieldValues['SALDO'];
ed_nim.Text:=lp_saldo.FieldValues['NIM'];
end;
procedure Tsaldo.ButtonClick(Sender: TObject);
begin
lp_saldo.Delete;
end;

```

```
end;
procedure Tsaldo.DataPetugas1Click(Sender: TObject);
begin
  Hide;
  petugas_login.Show;
end;
procedure Tsaldo.ed_saldo_newchange(Sender: TObject);
begin
  if ed_saldo_new.Text='' then
    Reload.Enabled:=False
  else
    Reload.Enabled:=True;
  end;
procedure Tsaldo.connect_actxecute(Sender: TObject);
begin
  smartcardtoolsetsrvr_ActiveX1.ConnectCARD;
  getreaderslist_act.Execute;
  bt_disconnect.Enabled:=True;
  bt_connect.Enabled:=False;
end;
procedure Tsaldo.disconnectExecute(Sender: TObject);
begin
  bt_connect.Enabled:=True;
  bt_disconnect.Enabled:=False;
end;
procedure Tsaldo.disconnectExecute(Sender: TObject);
begin
  bt_connect.Enabled:=True;
  smartcardtoolsetsrvr_ActiveX1.DisconnectCARD;
end;
procedure Tsaldo.getreaderslist_actxecute(Sender: TObject);
var
  r1:widestring;
  r1list:string;
begin
  try
    comboBox1.Clear;
    r1:=smartcardtoolsetsrvr_ActiveX1.getreaderslist;
  if r1>''
  then begin
    r1list:=StringReplace(r1,' ','#13#10',[rReplaceAll]);
    comboBox1.Items.Text:=r1list;
    comboBox1.ItemIndex:=0;
  end;
finally
  end;
end;
end;
end;
procedure Tsaldo.select_actxecute(Sender: TObject);
```