

SKRIPSI

PENGEMBANGAN APLIKASI SISTEM INFORMASI MANAGEMENT PARKIR MODERN BERBASIS CLIENT SERVER



Disusun Oleh
ARY ARTANTO
05.12.600

JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2010

REPORT

THE REPORT OF THE COMMITTEE ON THE NATIONAL DEFENSE ESTABLISHMENT
CONCERNING THE NATIONAL DEFENSE ESTABLISHMENT

THE NATIONAL DEFENSE ESTABLISHMENT
CONCERNING THE NATIONAL DEFENSE ESTABLISHMENT
CONCERNING THE NATIONAL DEFENSE ESTABLISHMENT

THE NATIONAL DEFENSE ESTABLISHMENT
CONCERNING THE NATIONAL DEFENSE ESTABLISHMENT
CONCERNING THE NATIONAL DEFENSE ESTABLISHMENT
CONCERNING THE NATIONAL DEFENSE ESTABLISHMENT
CONCERNING THE NATIONAL DEFENSE ESTABLISHMENT

LEMBAR PERSETUJUAN

**PENGEMBANGAN APLIKASI SISTEM INFORMASI MANAGEMENT
PARKIR MODERN BERBASIS CLIENT SERVER**

SKRIPSI

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



Disusun Oleh :

ARY ARTANTO

05.12.600

Diperiksa dan Disetujui

Mengetahui

Ketua Jurusan Teknik Elektro S-1

Dosen Pembimbing

Ir. Yusuf Ismail Nakhoda, MT

NIP.Y. 1018800189

I Komang Somawirata, ST, MT

NIP.Y. 1030100361



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

2010

ABSTRAKSI

PENGEMBANGAN APLIKASI SISTEM INFORMASI *MANAGEMENT* PARKIR *MODERN* BERBASIS *CLIENT SERVER*

Ary Artanto

05.12.600

Email : ary_artanto@ymail.co.id

Jurusan Teknik Elektro

Konsentrasi Teknik Komputer & Informatika S-1

Fakultas Teknologi Industri

Institut Teknologi Nasional Malang

Dosen Pembimbing : I Komang Somawirata, ST, MT

Dalam era globalisasi dengan tingkat perkembangan yang semakin maju dan modern, informasi telah menjadi salah satu faktor yang turut mendukung pertumbuhan dan perkembangan suatu bidang usaha maupun organisasi. Tak terkecuali bidang perparkiran, lahan parkir merupakan lahan yang berpotensi menghasilkan banyak uang atau pendapatan. Sistem lama yang melakukan pencatatan dengan karcis biasa tanpa adanya penyimpanan data yang rapi menyebabkan penyimpangan dalam hal pendapatan, keterbatasan tempat parkir, pengguna parkir yang meningkat dan juga bagi pemilik stan yang menyewa di plaza sering kali direpotkan dengan membayar karcis setiap masuk plaza, padahal mereka mempunyai intensitas yang sangat sering untuk ke plaza. Dengan alasan ini, maka bagaimana mencari cara agar bisa mengelola parkir dengan baik.

Dalam sistem informasi *management* parkir ini mempunyai fungsi untuk membantu pengelola plaza dalam mengelola operasional parkir. Terdapat 2 macam jenis parkir didalam sistem informasi ini yaitu parkir umum dan parkir member. Untuk parkir umum, karena keterbatasan tempat parkir dan pengunjung yang meningkat maka biaya parkir umum didapat dari lama waktu parkir dari pengguna parkir. Sedangkan parkir member ditujukan bagi pemilik stan dengan sistem pemberlakukan pembayaran parkir per bulan dan dapat diketahui masa aktif dari member.

Dengan dibuatnya sistem informasi ini diharapkan dapat memudahkan dalam mengelola parkir sehingga dapat meningkatkan kinerja, efisiensi dalam operasional parkir dan mengurangi kebocoran pendapatan parkir.

Kata Kunci : Parkir, Sistem Informasi.

KATA PENGANTAR



Alhamdulillah, puji syukur kehadiranMu Ya Allah yang telah memberikan Rahmat dan HidayahNya, sehingga penulis dapat menyelesaikan skripsi dengan judul “Pengembangan Aplikasi Sistem Informasi *Management Parkir Modern* Berbasis *Client Server*” dengan lancar. Skripsi merupakan persyaratan kelulusan Studi di Jurusan Teknik Elektro S-1 Konsentrasi Teknik Komputer dan Informatika ITN Malang dan untuk mencapai gelar Sarjana Teknik.

Keberhasilan penyelesaian laporan skripsi ini tidak lepas dari dukungan dan bantuan berbagai pihak. Untuk itu penyusun menyampaikan terima kasih kepada :

1. Bapak Prof. Dr. Ir. Abraham Lomi, MSEE selaku Rektor Institut Teknologi Nasional Malang.
2. Bapak Ir. Sidik Noetjahjono, MT selaku Dekan Fakultas Teknologi Industri Institut Teknologi Nasional Malang
3. Bapak Ir. F. Yudi Limpraptono, MT selaku Ketua Jurusan Teknik Elektro S-1.
4. Bapak Yusuf Ismail Nahkoda ST, MT selaku Sekretaris Jurusan Teknik Elektro S-1.
5. Bapak I Komang Somawirata, ST, MT selaku Dosen Pembimbing.
6. Ayah dan Ibu serta saudara-saudara kami yang selalu memberikan do'a restu, dorongan dan semangat.
7. Teman-teman dan semua yang telah membantu dalam penyelesaian penyusunan skripsi ini.

Penulis telah berusaha semaksimal mungkin dan menyadari sepenuhnya akan keterbatasan pengetahuan dalam menyelesaikan laporan ini. Untuk itu penyusun mengharapkan saran dan kritik yang membangun dari pembaca demi kesempurnaan laporan ini.

Harapan penulis semoga laporan skripsi ini memberikan manfaat bagi perkembangan ilmu pengetahuan bangsa dan negara.

Malang, Juli 2010

Penulis

DAFTAR ISI

LEMBAR PERSETUJUAN	i
ABSTRAKSI	ii
KATA PENGANTAR	iii
DAFTAR ISI	v
DAFTAR GAMBAR	x
DAFTAR TABEL	xv
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Tujuan	2
1.4. Batasan Masalah	2
1.5. Metodologi Penulisan	3
1.6. Sistematika Penulisan	4
BAB II LANDASAN TEORI	6
2.1. Konsep Dasar Sistem	6
2.1.1 Pengertian Sistem	6
2.1.2 Karakteristik Sistem	6
2.2. Konsep Dasar Informasi	8
2.2.1 Pengertian Informasi.....	8
2.2.2 Kualitas Informasi	9
2.3 Konsep Dasar Sistem Informasi	10

2.3.1 Pengertian Sistem Informasi.....	10
2.3.2 Komponen-komponen Sistem Informasi.....	12
2.4 Perancangan Sistem	13
2.4.1 Pengertian Perancangan Sistem	13
2.4.2 Diagram Aliran Data	14
2.4.3 Pemodelan Data.....	16
2.5 <i>Client Server</i>	17
2.6 Delphi	18
2.6.1 Kegunaan Delphi	19
2.6.2 Keunggulan Delphi.....	19
2.7 SQL Server 2000	19
BAB III PERANCANGAN DAN DESAIN SISTEM	23
3.1 Analisa Sistem	24
3.1.1 Deskripsi Sistem.....	24
3.1.2 Sistem Yang di Pakai Saat Ini	24
3.1.3.Pengguna Sistem.....	24
3.1.4.Spesifikasi Sistem.....	25
3.2 Perancangan Sistem	27
3.2.1 Data Flow Diagram (DFD).....	27
3.2.2 Desain Basis Data	31
3.2.2.1 Relasi Antar Tabel	31
3.2.2.2 Struktur Tabel Yang Digunakan.....	33

3.2.3 Desain Antarmuka Aplikasi.....	39
3.2.3.1 Desain Form Login.....	40
3.2.3.2 Desain Menu Aplikasi	40
3.2.3.3 Form Karyawan	41
3.2.3.4 Form Tarif Parkir Umum.....	41
3.2.3.5 Form Tarif Parkir Member	42
3.2.3.6 Form Kapasitas Parkir	42
3.2.3.7 Form Member	43
3.2.3.8 Form Aktifasi member.....	43
3.2.3.9 Desain Form Hak Akses	44
3.2.3.10 Desain Form <i>Update</i> Password	44
3.2.3.11 Desain Form Parkir Masuk Motor.....	45
3.2.3.12 Desain Form Parkir Masuk Mobil	45
3.2.3.13 Desain Form Parkir Keluar Motor.....	46
3.2.3.14 Desain Form Parkir Keluar Mobil	47
3.2.3.15 Desain Form Pengeluaran.....	48
3.2.3.16 Desain Form Filter laporan.....	49
3.2.3.17 Desain Laporan.....	49
3.2.3.18 Desain Karcis Masuk.....	51
3.2.3.19 Desain Karcis Keluar	52

BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM	53
4.1. Implementasi Sistem	53
4.2. Pengujian Aplikasi.....	53
4.2.1. Pengujian Form Login	54
4.2.2. Pengujian Form Menu Utama	55
4.2.3 Pengujian Form Karyawan	56
4.2.4 Pengujian Form <i>User Access</i>	58
4.2.5 Pengujian Form <i>Update Password</i>	60
4.2.6 Pengujian Form Tarif Parkir umum.....	61
4.2.7 Pengujian Form Tarif Parkir member.....	62
4.2.8 Pengujian Form Kapasitas Parkir	63
4.2.9 Pengujian Form Member	64
4.2.10 Pengujian Form Aktifasi Member	66
4.2.11 Pengujian Form Parkir Masuk Motor	67
4.2.12 Pengujian Form Parkir Masuk Mobil	71
4.2.13 Pengujian Form Parkir Keluar Motor	74
4.2.14 Pengujian Form Parkir Keluar Mobil	76
4.2.15 Pengujian Form Pengeluaran.....	79
4.2.16 Pengujian Form Laporan Karyawan.....	80
4.2.17 Pengujian Form Laporan Member.....	81
4.2.18 Pengujian Form Laporan Aktif Member	83
4.2.19 Pengujian Form Laporan Parkir Umum	84
4.2.20 Pengujian Form Laporan Parkir member	86

4.2.21 Pengujian Form Laporan Pendapatan.....	88
4.2.22 Pengujian Form Laporan Pengeluaran	89
4.2.23 Pengujian Form Laporan Laba/Rugi	91
4.2.24 Pengujian Karcis Masuk.....	92
4.2.25 Pengujian Karcis Keluar	93
BAB V PENUTUP	94
5.1 Kesimpulan	94
5.2 Saran	94
DAFTAR PUSTAKA	95
LAMPIRAN-LAMPIRAN	

DAFTAR GAMBAR

BAB II LANDASAN TEORI

Gambar 2.1. Desain Sistem model <i>Client-Server</i>	11
--	----

BAB III PERANCANGAN DAN DESAIN SISTEM

Gambar 3-1. Desain Sistem	26
Gambar 3-2. DFD Level 0	27
Gambar 3-3. DFD Level 1	28
Gambar 3-4. DFD Level 2 Pengolahan Data Parkir Masuk	30
Gambar 3-5. DFD Level 2 Pengolahan Data Parkir Keluar	30
Gambar 3-6. CDM	31
Gambar 3-7. PDM	32
Gambar 3-8. Form Login	40
Gambar 3-9. Desain Menu Aplikasi	40
Gambar 3-10. Form Karyawan	41
Gambar 3-11. Form Tarif Parkir Umum	41
Gambar 3-12. Form Tarif Parkir Member	42
Gambar 3-13. Form Kapasitas	42
Gambar 3-14. Form Member	43
Gambar 3-15. Form Aktifasi Member	43
Gambar 3-16. Form Setting hak Akses	44
Gambar 3-17. Form Update Password	44

Gambar 3-18. Form Parkir Masuk Motor.....	45
Gambar 3-19. Form Parkir Masuk Mobil.....	46
Gambar 3-20. Form Parkir Keluar Motor.....	47
Gambar 3-21. Form Parkir Keluar Mobil.....	48
Gambar 3-22. Form Pengeluaran.....	48
Gambar 3-23. Form Filter Laporan	49
Gambar 3-24. Desain Laporan.....	50
Gambar 3-25. Rancangan Karcis Masuk.....	52
Gambar 3-26. Rancangan Karcis Masuk.....	52

BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM

Gambar 4-1. Form Login	54
Gambar 4-2. Form Pesan Login Salah.....	54
Gambar 4-3. Form Menu Utama	55
Gambar 4-4. Entri Data Karyawan.....	56
Gambar 4-5. Data Karyawan Tersimpan	56
Gambar 4-6. Data Karyawan Telah di <i>Update</i>	57
Gambar 4-7. Pesan Data Tidak Lengkap	57
Gambar 4-8. Entri <i>User Access</i>	58
Gambar 4-9. Data telah di <i>Update</i>	59
Gambar 4-10. Data Telah di <i>Delete</i>	59
Gambar 4-11. Pesan Kesalahan input NIP.....	60
Gambar 4-12. Entri <i>Update Password</i>	60
Gambar 4-13. Entri Tarif Parkir Umum	61

Gambar 4-14. Pesan Kesalahan <i>Input</i> Tarif Umum	61
Gambar 4-15. Entri Tarif Parkir Member.....	62
Gambar 4-16. Pesan Kesalahan <i>Input</i> Tarif Member.....	62
Gambar 4-17. Entri Kapasitas Parkir	63
Gambar 4-18. Pesan Kesalahan <i>Input</i> Kapasitas	63
Gambar 4-19. Entri Data Member.....	64
Gambar 4-20. Data Member Telah DiSimpan.....	64
Gambar 4-21. Data Member Telah Di <i>Update</i>	65
Gambar 4-22. Pesan Data Tidak Lengkap	65
Gambar 4-23. Entri Data Aktifasi Member	66
Gambar 4-24. Entri Data Aktif Member	66
Gambar 4-25. Data Aktif Member Setelah diPerpanjang.....	67
Gambar 4-26. Pesan Kesalahan <i>Input</i> ID Member	67
Gambar 4-27. Entri Data Parkir Masuk Umum Motor	68
Gambar 4-28. Pesan Peringatan Kendaraan	68
Gambar 4-29. Data Parkir Masuk Umum Motor di Tabel T_Parkir	68
Gambar 4-30. Data Parkir Masuk Umum Motor di Tabel T_Parkir_Umum	69
Gambar 4-31. Entri Data Parkir Masuk Member Motor	69
Gambar 4-32. Data Parkir Masuk Member Motor di Tabel T_Parkir.....	70
Gambar 4-33. Data Parkir Masuk Member Motor di T_Parkir_Member..	70
Gambar 4-34. Pesan Peringatan Masa Aktif Member Motor	70
Gambar 4-35. Entri Data Parkir Masuk Umum Mobil.....	71
Gambar 4-36. Pesan Peringatan Kendaraan	72

Gambar 4-37.	Data Parkir Masuk Umum Mobil di Tabel T_Parkir	72
Gambar 4-38.	Data Parkir Masuk Umum Mobil di T_Parkir_Umum	72
Gambar 4-39.	Entri Data Parkir Masuk Member Mobil.....	73
Gambar 4-40.	Data Parkir Masuk Member Mobil di Tabel T_Parkir.....	73
Gambar 4-41.	Data Parkir Masuk Member Mobil diT_Parkir_Member....	73
Gambar 4-42.	Pesan Peringatan Masa Aktif Member Mobil.....	73
Gambar 4-43.	Entri Data Parkir Keluar Umum Motor	74
Gambar 4-44.	Data Parkir Keluar Motor di Tabel T_Parkir	75
Gambar 4-45.	Data Parkir Keluar Motor di Tabel T_Parkir_Umum	75
Gambar 4-46.	Entri Data Parkir Keluar Member Motor	75
Gambar 4-47.	Data Parkir Keluar Member Motor di Tabel T_Parkir.....	76
Gambar 4-48.	Pesan Peringatan Kendaraan Motor Telah Keluar	76
Gambar 4-49.	Entri Data Parkir Keluar Umum Mobil.....	77
Gambar 4-50.	Data Parkir Keluar Mobil di Tabel T_Parkir	77
Gambar 4-51.	Data Parkir Keluar Mobil di Tabel T_Parkir_Umum	77
Gambar 4-52.	Entri Data Parkir Keluar Member Mobil.....	78
Gambar 4-53.	Data Parkir Keluar Member Mobil di Tabel T_Parkir.....	78
Gambar 4-54.	Pesan Peringatan Kendaraan Mobil Telah Keluar	78
Gambar 4-55.	Entri Data Pengeluaran Parkir.....	79
Gambar 4-56.	Data Pengeluaran Parkir Telah Tersimpan.....	79
Gambar 4-57.	Filter Laporan Karyawan	80
Gambar 4-58.	Hasil Laporan Karyawan.....	81
Gambar 4-59.	Filter Laporan Member.....	81

Gambar 4-60. Hasil Laporan Member.....	82
Gambar 4-61. Filter Laporan Aktif Member.....	83
Gambar 4-62. Laporan Aktif Member.....	84
Gambar 4-63. Filter Laporan Parkir Umum.....	85
Gambar 4-64. Laporan Parkir Umum.....	86
Gambar 4-65. Filter Laporan Parkir Member.....	87
Gambar 4-66. Laporan Parkir Member.....	88
Gambar 4-67. Filter Laporan Pendapatan.....	88
Gambar 4-68. Laporan Pendapatan.....	89
Gambar 4-69. Filter Laporan Pengeluaran.....	90
Gambar 4-70. Laporan Pengeluaran.....	90
Gambar 4-71. Filter Laporan Laba/Rugi.....	91
Gambar 4-72. Laporan Laba/Rugi.....	92
Gambar 4-73. Karcis Masuk.....	93
Gambar 4-74. Karcis Keluar.....	93

DAFTAR TABEL

BAB II LANDASAN TEORI

2-1. Simbol DFD	15
-----------------------	----

BAB III PERANCANGAN DAN DESAIN SISTEM

3-1. Tabel Karyawan	33
3-2. Tabel Tarif.....	33
3-3. Tabel User_Acc	34
3-4. Tabel Kapasitas	34
3-5. Tabel Member	35
3-6. Tabel Tarif_Member	35
3-7. Tabel Aktif_Member	36
3-8. Tabel Parkir.....	36
3-9. Tabel Parkir_Umum	37
3-10. Tabel Parkir_Member	37
3-11. Tabel Pengeluaran	38
3-12. Tabel Pengeluaran_Detil.....	38
3-13. Tabel Pendapatan	39
3-14. Tabel Laba/Rugi.....	39

BAB I

PENDAHULUAN

1.1 LATAR BELAKANG

Dalam era seperti saat ini, begitu banyak sektor kehidupan yang tidak terlepas dari peran serta dan penggunaan teknologi komputer, khususnya dalam lingkup bidang pekerjaan. Semakin hari kemajuan teknologi komputer baik dibidang piranti lunak maupun perangkat keras berkembang sangat pesat. Solusi untuk bidang kerja apapun akan ada cara untuk dapat dilakukan melalui media komputer termasuk dalam bidang perparkiran.

Pada sistem sebelumnya, pengelola Plaza menerapkan dengan cara melakukan pencatatan kendaraan yang masuk dengan menggunakan karcis biasa yang dilakukan oleh penjaga pintu masuk. Ketika pengguna parkir keluar, menunjukkan karcis terlebih dahulu, kemudian penjaga pintu keluar melakukan pengecekan. Dan juga keterbatasan tempat parkir, pengunjung yang meningkat dan pemilik stan atau *counter* yang direpotkan dengan membayar karcis setiap masuk plaza, padahal mereka mempunyai intensitas yang sangat sering untuk ke plaza.

Dengan menggunakan sistem komputerisasi, pencarian nomer kendaraan akan lebih cepat berdasarkan data yang dimasukkan setiap transaksinya. Dengan banyaknya transaksi yang dilakukan tiap harinya maka semua transaksi kendaraan yang masuk ataupun keluar dapat terpantau dengan laporan tiap periodenya sehingga diketahui berapa pendapatan parkir secara rinci.

Masalah ini diharapkan dapat teratasi dengan adanya Sistem Informasi manajemen parkir yang mampu meningkatkan kinerja dalam operasional parkir.

1.2 RUMUSAN MASALAH

Bagaimana membangun aplikasi Sistem Informasi manajemen Parkir yang dapat mempermudah mengelola parkir sehingga dapat meningkatkan kinerja dan efisiensi dalam pengolahan data parkir.

1.3 TUJUAN

Tujuan dari penulisan skripsi ini adalah untuk mengembangkan sistem informasi manajemen parkir yang dapat membantu kelancaran dalam pengelolaan parkir.

1.4 BATASAN MASALAH

Dalam laporan akhir “Pengembangan Aplikasi Sistem Informasi *Management Parkir Modern Berbasis Client Server*”, penulis akan memberikan batasan-batasan masalah agar tidak terjadi penyimpangan maksud dan tujuan utama penyusunan skripsi ini.

1. Aplikasi sistem informasi parkir ini diterapkan di “Plaza Taman Bontang”.
2. Sistem informasi yang dibahas hanya mencakup transaksi keluar masuk kendaraan yang terdiri dari pencatatan data kendaraan mobil dan motor.
3. Program-program yang digunakan dalam perancangan dan pembuatan aplikasi ini ialah :

- Borland Delphi 7.
- SQL Server 2000.
- Crystal Report 8.5

1.5 METODOLOGI PENELITIAN

Metode penelitian penulisan yang dipakai dalam pembuatan skripsi ini, antara lain :

- **Studi Pustaka / Literatur**

Pengumpulan data ini dilakukan dengan cara mencari bahan-bahan kepustakaan sebagai landasan teori yang ada hubungannya dengan permasalahan yang dijadikan objek penelitian.

- **Perancangan Sistem**

Perancangan sistem yang dilakukan dengan membuat alur sistem untuk mengetahui alur jalannya aplikasi yang dibuat.

- **Desain Aplikasi**

Untuk menghasilkan aplikasi yang baik desain dibuat secara sederhana agar mudah digunakan dengan tampilan yang sedikit menarik.

- **Pengujian Aplikasi dari Sistem yang dibuat**

Eksperimen yang dilakukan untuk mengetahui kadar atau kesulitan yang mungkin akan terjadi dalam perancangan *software*.

1.6 SISTEMATIKA PENULISAN

Penulisan skripsi ini terbagi dalam lima bab dengan sistematika sebagai berikut :

BAB I : PENDAHULUAN

Pada bab ini dibahas tentang latar belakang permasalahan, rumusan masalah, batasan masalah, metodologi dan sistematika penulisan dari aplikasi yang dibuat.

BAB II : LANDASAN TEORI

Berisi tentang landasan teori yang mendukung dalam perencanaan dan pembuatan aplikasi.

BAB III : PERANCANGAN SISTEM

Dalam bab ini membahas tentang perancangan sistem dan program mencakup desain sistem. Desain modul dan desain *software*.

BAB IV : IMPLEMENTASI DAN PENGUJIAN SISTEM

Berisi tentang implementasi *software* terhadap sistem yang telah dirancang dan berikut pengujian serta analisa dari perancangan sistem yang telah dibuat. Pengujian aplikasi dilakukan untuk mengetahui permasalahan-permasalahan yang mungkin muncul. Analisa dilakukan untuk menyempurnakan aplikasi yang dibuat

jika terdapat kekurangan atau kesalahan pada jalannya program maupun dari hasil yang diperoleh.

BAB V : KESIMPULAN DAN SARAN

Berisi tentang kesimpulan yang dapat diambil berdasarkan hasil uraian pada bab-bab sebelumnya dan saran serta petunjuk pengembangan dan penyempurnaan hasil skripsi ini.

BAB II

LANDASAN TEORI

2.1 Konsep Dasar Sistem

2.1.1 Pengertian Sistem

Terdapat dua kelompok didalam mendefinisikan sistem,yaitu yang menekankan pada prosedurnya dan yang akan menekankan pada komponen atau elemennya. Pendekatan sistem yang menekankan pada prosedur mendefinisikan bahwa sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu. Pendekatan sistem yang menekankan pada elemen atau komponennya mendefinisikan bahwa “Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu”.

2.1.2 Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat yang tertentu yaitu:

a) Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi atau saling bekerja sama untuk membentuk suatu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu sub sistem atau bagian-bagian dari sistem.

b) Batas Sistem

Batas sistem (*Boundary*) merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas suatu sistem menunjukkan ruang lingkup dari sistem tersebut.

c) Lingkungan Luar Sistem

Lingkungan luar sistem (*Environment*) adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem bersifat menguntungkan dan juga dapat bersifat merugikan sistem tersebut.

d) Penghubung Sistem

Penghubung (*Interface*) sistem adalah merupakan media penghubung antara suatu subsistem dengan subsistem lainnya. Keluaran (*output*) dari satu subsistem akan menjadi masukan (*input*) untuk subsistem yang lainnya dengan melalui penghubung.

e) Masukan Sistem

Masukan (*Input*) adalah energi yang dimasukkan kedalam sistem. Dapat berupa masukan perawatan (*Maintenance Input*) dan masukan sinyal (*Signal Input*).

f) Keluaran Sistem

Keluaran (*Output*) adalah hasil dari energi yang diolah dan di klasifikasikan menjadi keluaran yang berguna dan sisa

pembuangan. Keluaran dapat merupakan masukan untuk subsistem yang lain atau kepada supra *system*.

g) Pengolah Sistem

Suatu sistem dapat mempunyai suatu bagian pengolahan yang akan merubah masukan menjadi keluaran.

h) Sasaran Sistem

Suatu sistem mempunyai tujuan atau sasaran, sasaran dari sistem sangat menentukan masukan yang dibutuhkan sistem dan keluaran yang dihasilkan sistem.

2.2 Konsep Dasar Informasi

Dalam konsep dasar informasi akan diuraikan mengenai pengertian dan kualitas dari informasi.

2.2.1 Pengertian Informasi

Informasi sangat penting dalam suatu organisasi terutama bagi manajemen dalam mengambil keputusan. Apakah sebenarnya informasi sehingga sangat penting bagi organisasi? “Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya”.

Sumber dari informasi adalah data. Data merupakan bentuk yang masih mentah yang belum menjelaskan secara rinci, sehingga data perlu diolah lebih lanjut untuk menghasilkan suatu informasi.

2.2.2 Kualitas Informasi

Kualitas dari suatu informasi tergantung tiga hal yaitu informasi harus akurat (*accurate*), tepat pada waktunya (*timeliness*) dan relevan (*relevance*).

a) Akurat (*Accurate*)

Informasi harus bebas dari kesalahan dan tidak bias atau menyesatkan. Akurat juga berarti bahwa informasi harus jelas mencerminkan maksudnya. Informasi harus akurat karena dari sumber informasi sampai ke penerima informasi mungkin banyak mengalami gangguan (*noise*) yang dapat mengubah atau merusak informasi tersebut.

b) Tepat Pada Waktunya (*Timeliness*)

Informasi yang sampai pada si penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi. Karena informasi merupakan landasan di dalam pengambilan keputusan. Bila pengambilan keputusan terlambat maka dapat berakibat fatal bagi organisasi. Dewasa ini informasi mahal karena harus cepat dikirim dan didapatkan sehingga memerlukan teknologi mutakhir untuk mendapatkan, mengolah dan mengirimnya.

c) Relevan (*Relevance*)

Informasi tersebut mempunyai manfaat untuk pemakaiannya. Relevansi informasi untuk setiap orang, satu dengan yang lain adalah berbeda.

2.3 Konsep Dasar Sistem Informasi

Informasi dapat diperoleh dari suatu sistem informasi (*Information System*).

2.3.1 Pengertian Sistem Informasi

Dalam artian umum, sistem informasi merupakan interrelasi antara beberapa komponen yang menyimpan, mengambil, menerima, memproses, dan mendistribusikan informasi untuk mendukung kegiatan pengambil keputusan, dan mengontrol sebuah organisasi. Sedangkan dalam arti khusus, sistem informasi adalah perangkat lunak yang digunakan untuk meningkatkan kinerja sebuah sistem dalam menangani informasi atau data.

Ada dua macam komponen dalam sistem informasi, yaitu :

1. Komponen dasar

- a. *Input* : meliputi elemen yang di-*capture*, dirakit yang masuk ke sistem untuk diproses.
- b. Proses : meliputi proses transformasi yang mengubah *input* menjadi *output*.
- c. *Output* : meliputi transfer elemen yang dihasilkan oleh proses ke tujuan.

2. Komponen tambahan

- a. *Feedback* : data yang menyatakan performansi sistem.
- b. *Control* : meliputi *monitoring* dan evaluasi *feedback* untuk menentukan apakah sistem mencapai tujuan. Fungsinya adalah membuat penyesuaian untuk *input* sistem dan pemrosesan

komponen untuk memastikan sistem menghasilkan *output* yang tepat.

Dalam sistem informasi, ada tiga tahap yang harus dipenuhi, yaitu input, proses dan output, dimana output memungkinkan suatu umpan balik yang dapat merubah atau memodifikasi suatu input. Di sinilah suatu sistem informasi berperan sebagai pengambil keputusan dalam menangani suatu permasalahan.

Di dalam pengembangan sistem informasi, dikenal istilah *SDLC* (*System Development Life Cycle*) atau “Siklus Hidup Pengembangan Sistem”, yang terdiri dari enam tahap :

1. Perencanaan sistem
2. Analisis sistem
3. Rancangan sistem general atau konseptual
4. Evaluasi dan pemilihan sistem
5. Rancangan sistem terinci atau fungsional
6. Implementasi sistem

Empat tahap pertama disebut tahap *FRONT-END*, dua yang terakhir disebut tahap *BACK-END*. Setelah sistem baru dikembangkan dan dikonversikan ke operasi, selanjutnya menuju ke tahap pemeliharaan sistem yang berlangsung beberapa tahun, 10 sampai 20 tahun atau lebih lama. Jika sistem ini tidak lagi efisien dan efektif untuk tahap dipelihara, maka tidak dilanjutkan dan sistem baru dikembangkan untuk menggantikannya, *SDLC* mulai dari awal lagi.

2.3.2 Komponen-komponen Sistem Informasi

Sistem informasi terdiri dari komponen-komponen yang disebut dengan istilah blok bangunan (*building block*) yaitu :

1. Blok Masukan (*Input Block*)

Input mewakili data yang masuk ke dalam sistem informasi. *Input* disini termasuk metode-metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.

2. Blok Model (*Model Block*)

Blok ini terdiri dari kombinasi prosedur, logika dan model matematik yang akan memanipulasi data *input* dan data yang tersimpan di basis data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang diinginkan.

3. Blok Keluaran (*Output Block*)

Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

4. Blok Teknologi (*Technology Block*)

Teknologi merupakan kotak alat dalam sistem informasi. Teknologi digunakan untuk menerima *input*, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem secara keseluruhan. Teknologi terdiri dari 3 bagian utama yaitu teknisi (*humanware* atau *brainware*), perangkat lunak (*software*) dan perangkat keras (*hardware*).

5. Blok Basis Data (*Database Block*)

Basis data merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. Data perlu disimpan di dalam basis data untuk keperluan penyediaan informasi lebih lanjut.

6. Blok Kendali (*Controls Block*)

Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah ataupun bila terlanjur terjadi kesalahan-kesalahan dapat langsung cepat teratasi.

2.4 Perancangan Sistem

2.4.1 Pengertian Perancangan Sistem

Perancangan sistem dapat didefinisikan sebagai berikut: menurut **Robert J. Verzello** atau **Jhon Reuter III** “Perancangan Sistem adalah tahap setelah analisis dari siklus pengembangan sistem sebagai pendefinisian dari kebutuhan-kebutuhan fungsional dan persiapan untuk rancang bangun implementasi, menggambarkan bagaimana suatu sistem dibentuk”. Menurut **Jhon Burth** dan **Gary Grundnitski** “Perancangan sistem didefinisikan sebagai penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah kedalam satu kesatuan yang utuh dan berfungsi”. Menurut **George M. Scoot** “Perancangan sistem menentukan bagaimana suatu sistem akan menyelesaikan dari komponen-komponen perangkat lunak dan perangkat keras dari suatu sistem sehingga setelah instalasi dari sistem benar-benar memuat rancang bangun yang telah ditetapkan pada akhir tahap analisis sistem”.

2.4.2 Diagram Aliran Data

Diagram aliran data adalah representasi grafik dari sebuah sistem. DAD menggambarkan komponen-komponen sebuah sistem, aliran-aliran data di mana komponen-komponen tersebut, dan asal, tujuan, dan penyimpanan dari data tersebut. Kita dapat menggunakan DAD untuk dua hal utama, yaitu untuk membuat dokumentasi dari sistem informasi yang ada, atau untuk menyusun dokumentasi untuk sistem informasi yang baru.

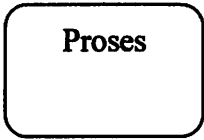
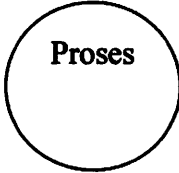



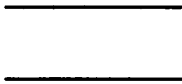


Kelebihan utama pendekatan aliran data, yaitu :

1. Kebebasan dari menjalankan implementasi teknis sistem.
2. Pemahaman lebih jauh mengenai keterkaitan satu sama lain dalam sistem dan subsistem.
3. Mengkomunikasikan pengetahuan sistem yang ada dengan pengguna melalui diagram aliran data.
4. Menganalisis sistem yang diajukan untuk menentukan apakah data-data dan proses yang diperlukan sudah ditetapkan.

Simbol-simbol yang digunakan dalam DFD dapat dilihat pada tabel

2.1 :

Tabel 2.1 Simbol DFD

Keterangan	Simbol Gene and Sarson	Simbol De Marco and Jourdan
Orang, unit yang mempergunakan atau melakukan transformasi data. Komponen fisik tidak diidentifikasi.		
Aliran data dengan arah khusus dari sumber ke tujuan		
Penyimpanan data atau tempat data direfer oleh proses	<p data-bbox="666 1183 806 1212">Data Store</p> 	<p data-bbox="966 1183 1106 1212">Data Store</p> 
Entitas eksternal, dapat berupa orang/unit terkait yang berinteraksi dengan sistem tetapi diluar sistem	<p data-bbox="666 1404 879 1433">Entitas eksternal</p> 	<p data-bbox="966 1404 1179 1433">Entitas eksternal</p> 

2.4.3 Pemodelan Data

Model data adalah sekumpulan cara / peralatan / *tool* untuk mendeskripsikan data-data, hubungannya satu sama lain, semantiknya, serta batasan konsistensi.

Ada dua model data, yaitu : *Entity Relationship Diagram* (ERD) dan model relasional. Keduanya menyediakan cara untuk mendeskripsikan perancangan basis data pada peringkat logika.

Model ERD atau *Conceptual Data Model* (CDM) adalah model yang dibuat berdasarkan anggapan bahwa dunia nyata terdiri dari koleksi obyek-obyek dasar yang dinamakan entitas (*entity*) serta hubungan (*relationship*) antara entitas-entitas itu.

Model Relasional atau *Physical Data Model* (PDM) adalah model yang menggunakan sejumlah tabel untuk menggambarkan data serta hubungan antara data-data tersebut. Setiap tabel mempunyai sejumlah kolom di mana setiap kolom memiliki nama yang unik.

Di dalam ERD atau CDM maupun PDM, relasi (hubungan) setiap entitas mempunyai derajat hubungan (kardinalitas) yang menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas yang lain. Relasi kardinalitas yang terjadi di antara dua himpunan entitas dapat berupa :

- a. 1 ke 1 (*one to one*), setiap entitas pada suatu himpunan entitas berhubungan paling banyak satu entitas pada himpunan entitas yang lain, begitu juga sebaliknya.
- b. 1 ke N (*one to many*), setiap entitas berhubungan dengan banyak entitas pada himpunan entitas yang lain, tetapi tidak sebaliknya.

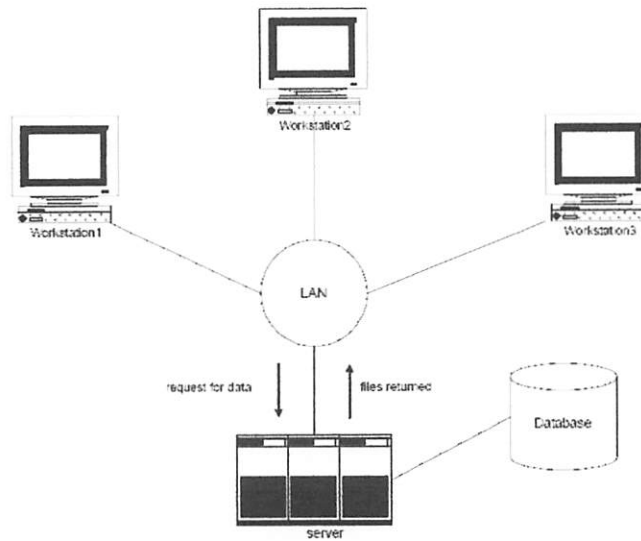
- c. N ke 1 (*many to one*), setiap entitas berhubungan dengan paling banyak satu entitas pada himpunan entitas yang lain, tetapi tidak sebaliknya.
- d. N ke N (*many to many*), setiap entitas pada suatu himpunan dapat berhubungan dengan entitas pada himpunan entitas yang lain, demikian sebaliknya.

2.5 *Client-server*

Client-server merupakan sebuah paradigma dalam teknologi informasi yang merujuk kepada cara untuk mendistribusikan aplikasi ke dalam dua pihak: pihak klien dan pihak server.

Dalam model klien/server, sebuah aplikasi dibagi menjadi dua bagian yang terpisah, tapi masih merupakan sebuah kesatuan yakni komponen klien dan komponen server. Komponen klien juga sering disebut sebagai *front-end*, sementara komponen server disebut sebagai *back-end*. Komponen klien dari aplikasi tersebut dijalankan dalam sebuah workstation dan menerima masukan data dari pengguna. Komponen klien tersebut akan menyiapkan data yang dimasukkan oleh pengguna dengan menggunakan teknologi pemrosesan tertentu dan mengirimkannya kepada komponen server yang dijalankan di atas mesin server, umumnya dalam bentuk *request* terhadap beberapa layanan yang dimiliki oleh server. Komponen server akan menerima request dari klien, dan langsung memprosesnya dan mengembalikan hasil pemrosesan tersebut kepada klien. Klien pun menerima informasi hasil pemrosesan data yang dilakukan server dan menampilkannya kepada pengguna, dengan menggunakan aplikasi yang berinteraksi dengan pengguna.

Desain model *client-server* dapat dilihat pada gambar 2-1.



Gambar 2-1 Desain Sistem Model *Client-Server*

2.6 Delphi

Delphi adalah suatu bahasa pemrograman (*development language*) yang digunakan untuk merancang suatu aplikasi program. Delphi adalah kompiler/penterjemah bahasa Pascal (awalnya dari Pascal) yang merupakan bahasa tingkat tinggi sekelas dengan basic, C. Bahasa pemrograman di delphi disebut bahasa *procedural* artinya bahasa/sintaknya mengikuti urutan tertentu/prosedur. Delphi termasuk keluarga visual sekelas Visual Basic, Visual C, artinya perintah-perintah untuk membuat objek dapat dilakukan secara visual. Delphi juga merupakan bahasa berorientasi objek. Saat anda pertama kali masuk ke Visual Basic maupun delphi, anda akan diperhadapkan pada sebuah form kosong yang akan dibuat secara otomatis. Form tersebut diberi nama Form1. Form ini merupakan tempat bekerja untuk membuat antarmuka pengguna (*user interface*).

2.6.1 Kegunaan Delphi

1. Untuk membuat aplikasi windows.
2. Untuk merancang aplikasi program berbasis grafis.
3. Untuk membuat program berbasis jaringan (*client/server*).
4. Untuk merancang program .net (berbasis internet).

2.6.2 Keunggulan Delphi

1. IDE (*Integrated Development Environment*) atau lingkungan pengembangan aplikasi sendiri adalah satu dari beberapa keunggulan delphi, didalamnya terdapat menu – menu yang memudahkan kita untuk membuat suatu proyek program.
2. Proses kompilasi cepat, pada saat aplikasi yang kita buat dijalankan pada delphi, maka secara otomatis akan dibaca sebagai sebuah program, tanpa dijalankan terpisah.
3. Mudah digunakan, source kode delphi yang merupakan turunan dari pascal, sehingga tidak perlu suatu penyesuaian lagi.
4. Bersifat multi purphase, artinya bahasa pemrograman delphi dapat digunakan untuk pengembangan aplikasi.

2.7 SQL Server 2000

SQL Server 2000 merupakan salah satu software RDBMS yang diperhitungkan pada saat ini. Perkembangan SQL Server 2000 dimulai pada waktu Microsoft pertama kali bekerja sama dengan Sybase pada tahun 1989 untuk membuat Sybase SQL Server di platform OS/2. Kemudian, seiring dengan kemajuan Microsoft memasarkan sistem

operasinya, pada versi 4.2 Microsoft membuat SQL Server untuk platform Windows NT. Sampai pada versi 6.0, Microsoft memutuskan kerjasamanya dengan Sybase. Versi 7.0, Microsoft membuat dari awal SQL Server dan merupakan versi pertama yang dapat digunakan di platform Windows 9.X

SQL Server 2000 mempunyai tujuh edisi, yaitu Standard Edition, Enterprise Edition, Personal Edition, Developer Edition, Windows CE Edition, Evaluation Edition dan Microsoft Desktop Engine. Dari kesemua versi yang ada, Enterprise Edition merupakan versi yang paling banyak digunakan dan mempunyai fasilitas terlengkap. Di dalam versi Enterprise Edition, tidak seperti vendor database lain, SQL Server 2000 menyediakan fasilitas Online Transaction Processing (OLTP) dan Online Analytical Processing (OLAP), juga Data Transformation Service (DTS) yang berguna untuk mengekspor-impor data. *SQL Server 2000* secara otomatis akan menginstall enam database utama, yaitu master, model, tempdb, pubs Northwind, dan msdb. Berikut penjelasan masing-masing database :

1. Master

Master database ialah gabungan dari tabel-tabel sistem yang mencatat instalasi *server* secara keseluruhan dan seluruh database yang dibuat secara subsekuen. Database master sangat kritis pada sistem Anda, jadi pastikan untuk membackup/mengkopinya. Operasi seperti pembuatan database lainnya, mengubah konfigurasi dan memodifikasi account login akan mengubah database master,

jadi setelah melakukan aktivitas tersebut, Anda harus *mbackup* database master.

2. Model

Model database ialah template database. Setiap kali Anda membuat database, SQL server membuat sebuah kopi dari model untuk membentuk basis dari database baru. Jika Anda menginginkan setiap database baru untuk memulai dengan obyek tertentu atau ijin, Anda dapat meletakkannya di model, dan seluruh database baru akan mengikutinya.

3. Tempdb

Tempdb ialah database tempat menampung data sementara untuk wadah kerja kita. Database ini akan dibuat setiap *SQL server* di start ulang. Seluruh pengguna/user mempunyai hak untuk membuat dan menggunakan tabel *private* dan *global* sementara yang disimpan di database *tempdb* (*tabel private dan global mempunyai awalan # dan ##*).

4. Pubs

Database ini ialah database contoh yang sering digunakan untuk panduan belajar *SQL server*. Anda boleh menghapusnya jika Anda tidak menginginkannya. Anda tidak perlu takut untuk membuat perubahan/modifikasi pada database pubs, karena Anda dapat *install* ulang database pubs dengan menjalankan scriptnya pada subdirektori *install*. Pada *SQL Query Analyzer*, buka file dengan nama *Instpubs.sql* dan eksekusi file script tersebut.

Pastikan bahwa tidak ada database *pubs* di *server* Anda dengan men-drop database tersebut.

5. Northwind

Database Northwind ialah contoh database yang awalnya dibangun untuk digunakan di Microsoft Access. Database Northwind dapat dibuat dengan cara yang sama seperti database *pubs*, yaitu dengan menginstal *instnwnd.sql* di subdirektori *install*.

6. Msdb

Database MSDB digunakan oleh *SQL Server Agent Service*, yang membentuk aktivitas terjadwal seperti *backup* dan tugas-tugas replikasi. Seluruh informasi di dalam *msdb* dapat diakses dari *SQL Server Enterprise Manager*. Jadi, Anda biasanya tidak harus mengakses data secara langsung.

BAB III

PERANCANGAN DAN DESAIN SISTEM

Pada bab ini dijelaskan mengenai analisis dan perancangan desain sistem aplikasi. Analisis sistem didefinisikan sebagai bagaimana memahami dan mengspesifikasi dengan detail apa yang harus dilakukan oleh sistem. Sementara perancangan sistem diartikan sebagai menjelaskan dengan detail bagaimana bagian-bagian dari sistem informasi diimplementasikan. Dengan demikian ,analisis dan perancangan sistem informasi bisa didefinisikan sebagai: proses organisasional kompleks dimana sistem informasi berbasis komputer diimplementasikan. Atau bisa diringkas sebagai berikut, Analisis:mendefinisikan masalah (*from requirements to specification*), Perancangan: memecahkan masalah (*from specification to implementation*).

Dalam merancang aplikasi pada proyek akhir ini terlebih dahulu dilakukan pembuatan desain proses,desain basis data,serta desain antar muka aplikasi. Desain proses berguna untuk mengintegrasikan semua proses yang terjadi dalam aplikasi yang dibuat. Desain basis data berguna untuk mengetahui data apa saja yang akan dibutuhkan dalam proses yang akan dikerjakan. Sedangkan perancangan antar muka interaksi antara pengguna dengan sistem aplikasi yang dibuat,sehingga pengguna dapat mengoperasikan aplikasi yang dibuat.

3.1 Analisis Sistem

3.1.1 Deskripsi Sistem

Sistem Informasi *Management* Parkir ini memiliki fungsi untuk mengelola operasional dari parkir. Data yang dikelola adalah data pegawai, tarif parkir, kapasitas parkir, data member dan data parkir kendaraan. Data-data tersebut kemudian diolah menjadi data pemasukan untuk bagian parkir. Besarnya pengeluaran dalam hal untuk operasional parkir juga dihitung yang kemudian akan menjadi data pengeluaran.

3.1.2 Sistem Yang di Pakai Saat Ini

Sistem yang ada pada saat ini sudah berbasis komputer, akan tetapi belum menggunakan basis data yang terstruktur. Aplikasi yang digunakan hanya sebatas aplikasi Microsoft Excel yang disediakan oleh pengelola plaza. Penggunaannya hanya untuk melakukan pendataan pendapatan parkir tiap harinya. Untuk pengolah data transaksi parkir masih menggunakan pencatatan dengan karcis biasa. Ini mengakibatkan tidak adanya penyimpanan data yang rapi sehingga menyebabkan terjadinya penyimpangan dalam pendapatan parkir. Hal ini menyebabkan sistem kurang efektif dan efisien dalam hal operasional parkir.

3.1.3 Pengguna Sistem

Pengguna sistem informasi *management* parkir ini, antara lain:

1. Manager yang bertugas untuk memonitor segala aktifitas operasional yang terjadi di perusahaan dan juga tentunya di bagian parkir dengan mempunyai akses untuk mengetahui semua laporan yang masuk.

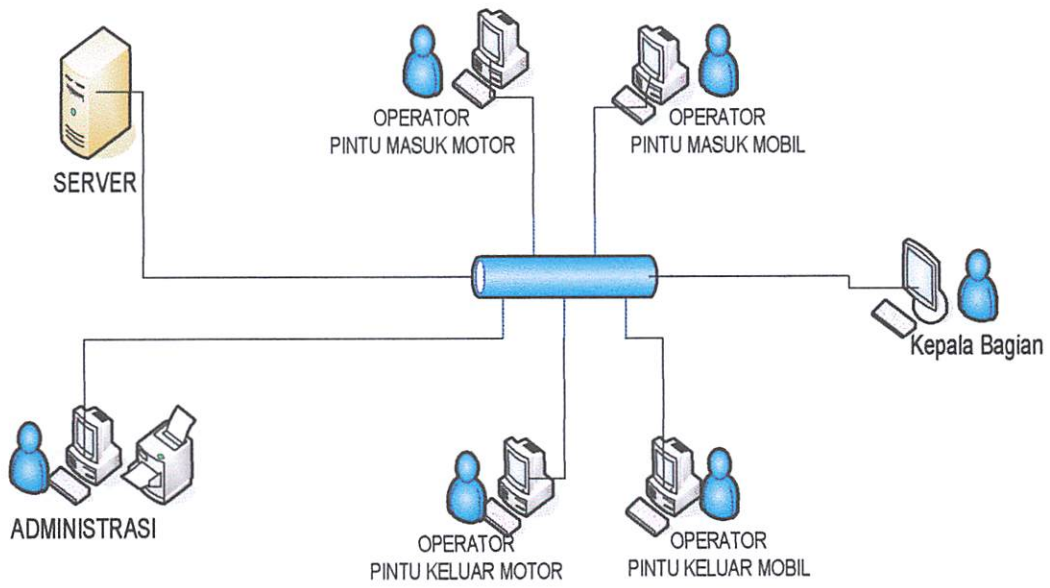
2. Kepala Bagian yang bertugas untuk men-*set* biaya parkir dan jumlah tempat parkir.
3. Operator masuk yang bertugas meng-*input* data kendaraan yang masuk ke areal parkir.
4. Operator keluar yang bertugas meng-*input* kendaraan yang keluar dari areal parkir.
5. Administrasi parkir yang bertugas meng-*input* data member, memperpanjang aktifasi member, pengeluaran yang dibutuhkan untuk operasional parkir serta mencek laporan parkir.

3.1.4 Spesifikasi Sistem

Sesuai dengan tujuan Sistem Informasi manajemen parkir maka spesifikasi sistem adalah sebagai berikut :

1. Sistem informasi dengan model *client - server*, dimana aplikasi *client* dapat mengakses basis data.
2. Sistem basisdata *server* menggunakan SQL Server 2000 dengan menggunakan satu buah *server*.
3. Sebuah modul pelaporan yang dapat memberikan data-data dan rekapitan yang akurat yang sangat berguna untuk pengambilan keputusan dan mengetahui laba atau rugi bagi pada bagian parkir.
4. Sistem informasi ini, diharapkan dapat memberikan kemudahan kepada pegawai untuk melakukan pencatatan kendaraan yang parkir sehingga proses lebih efektif dan efisien.

Gambar desain sistem dari sistem informasi management parkir dapat dilihat pada gambar 3.1.



Gambar 3.1 Desain Sistem

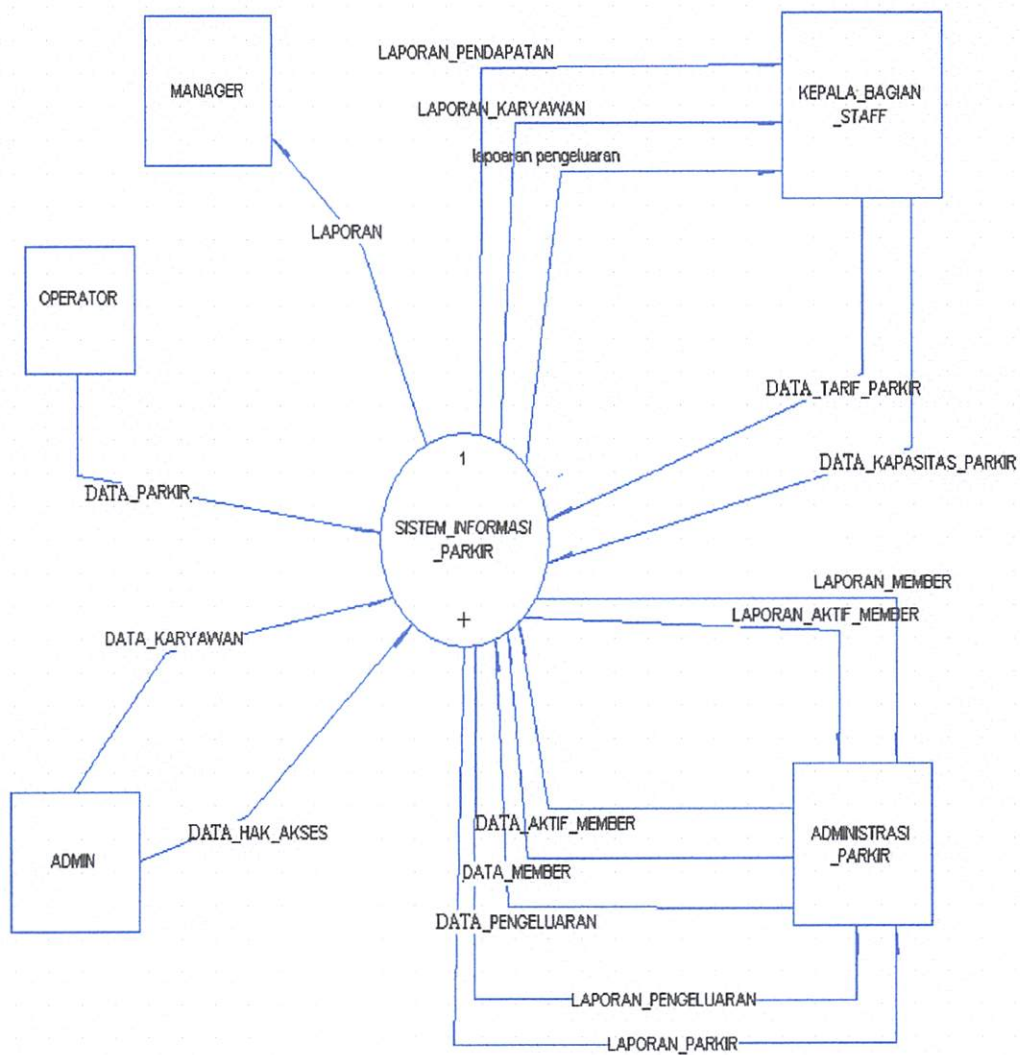
3.2 Perancangan Sistem

3.2.1. Data Flow Diagram (DFD)

1) DFD Level 0

DFD level 0 menunjukkan semua proses utama yang menyusun keseluruhan sistem.

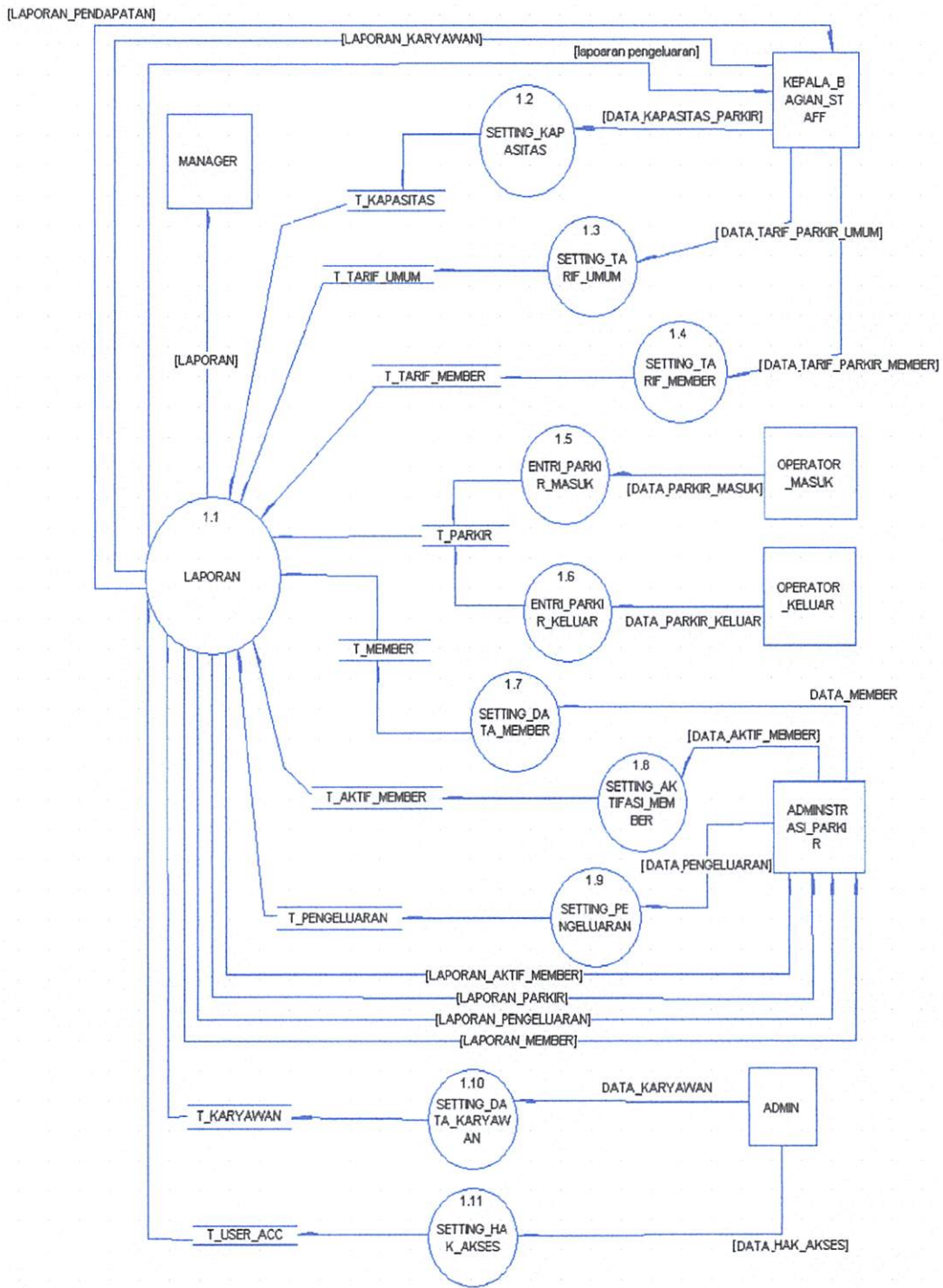
Gambar 3.2 merupakan tampilan dari DFD level-0:



Gambar 3.2 DFD Level 0 Sistem Informasi Parkir

2) DFD Level 1

Gambar 3.3 merupakan tampilan dari DFD level-1:

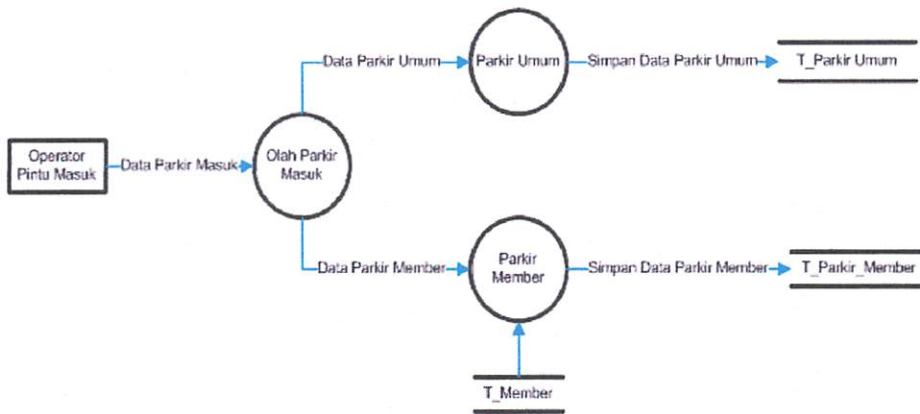


Gambar 3.3 DFD Level 1 Sistem Informasi Parkir

Seperti yang digambarkan pada DFD Level 1 diatas, terdapat sebelas anak proses sebagai berikut:

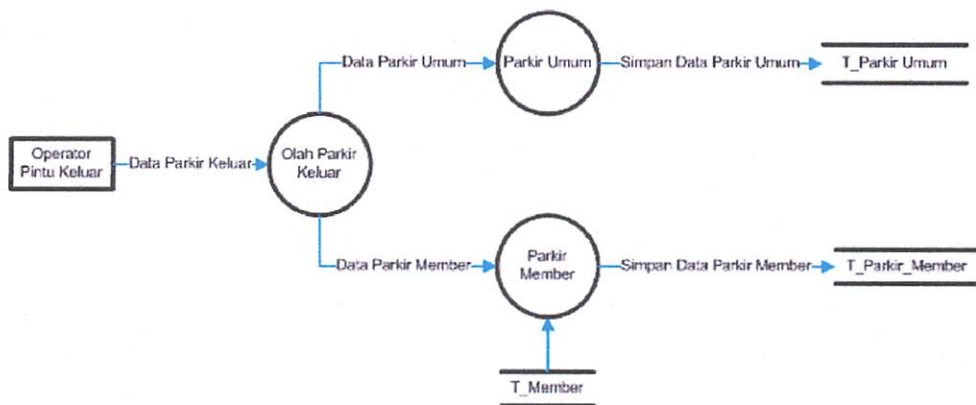
- a) Proses seting data karyawan berfungsi unrtuk mengolah data karyawan dari penambahan karyawan dan mengubah data karyawan.
- b) Proses setting kapasitas parkir berfungsi untuk mengolah data jumlah kapasitas parkir dari penambahan dan pengubahan data kapasitas parkir.
- c) Proses setting tarif parkir umum berfungsi untuk mengolah data tarif parkir umum dari penambahan dan pengubahan.
- d) Proses setting tarif parkir member berfungsi untuk mengolah data tarif parkir member dari penambahan dan pengubahan
- e) Proses olah data parkir masuk berfungsi untuk mengolah data parkir masuk dari penambahan kendaraan yang parkir.
- f) Proses olah data parkir keluar berfungsi untuk mengolah data parkir keluar dari penambahan dan pembayaran biaya parkir.
- g) Proses registrasi member baru berfungsi untuk mengolah data member dari penambahan dan pengubahan.
- h) Proses perpanjang aktif member berfungsi untuk mengolah data aktif member dari penambahan perpanjang aktifasi.
- i) Proses pengeluaran berfungsi untuk mengolah data pengeluaran dari penambahan,pengubahan dan penghapusan pengeluaran.
- j) Proses setting hak akses berfungsi untuk mengatur hak akses dari para pegawai.
- k) Proses laporan berfungsi untuk mengolah semua laporan.

3) DFD Level 2 Proses Pengolahan Data Parkir Masuk, dapat dilihat pada Gambar 3.4:



Gambar 3.4 DFD Level 2 Pengolahan Data Parkir Masuk

4) DFD Level 2 Proses Pengolahan Data Parkir Keluar, dapat dilihat pada Gambar 3.5:

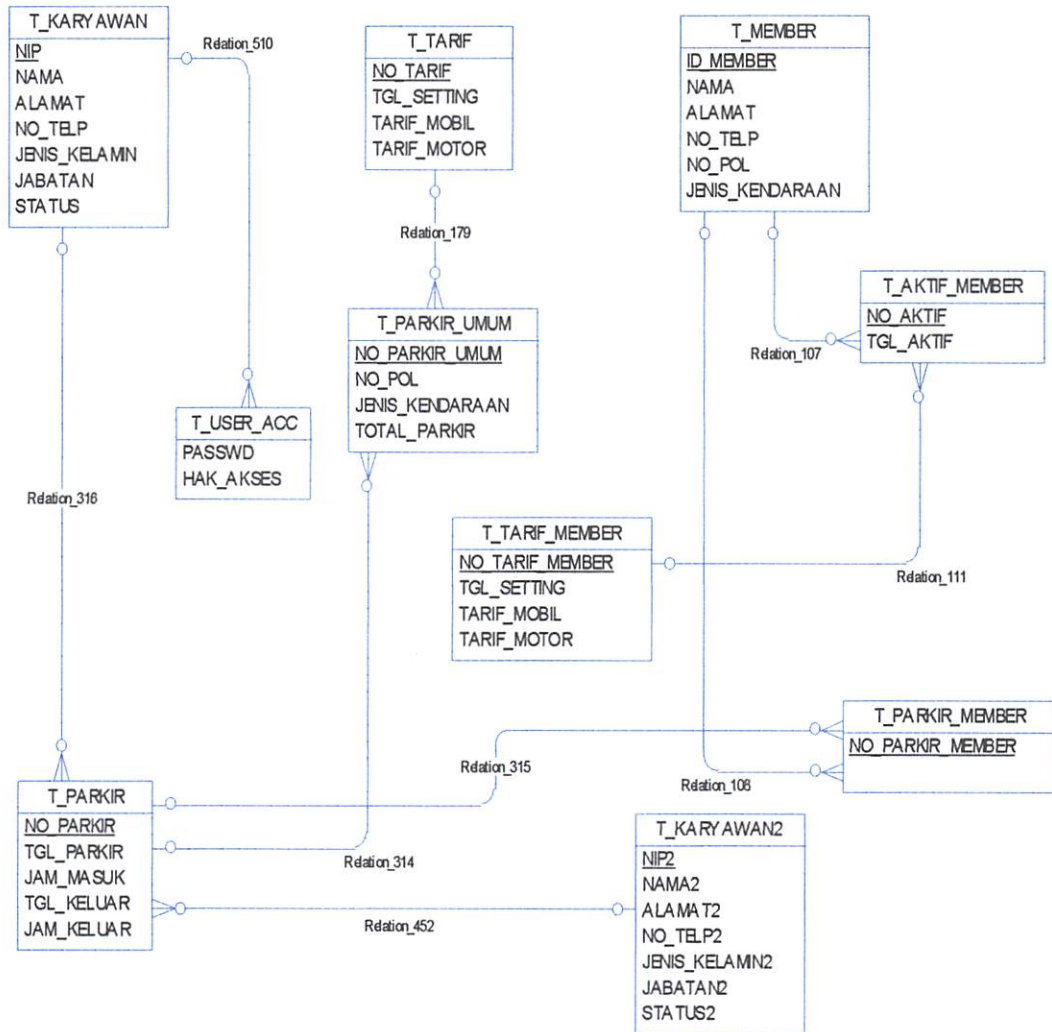


Gambar 3.5 DFD Level 2 Pengolahan Data Parkir Keluar

3.2.2. Desain Basis Data

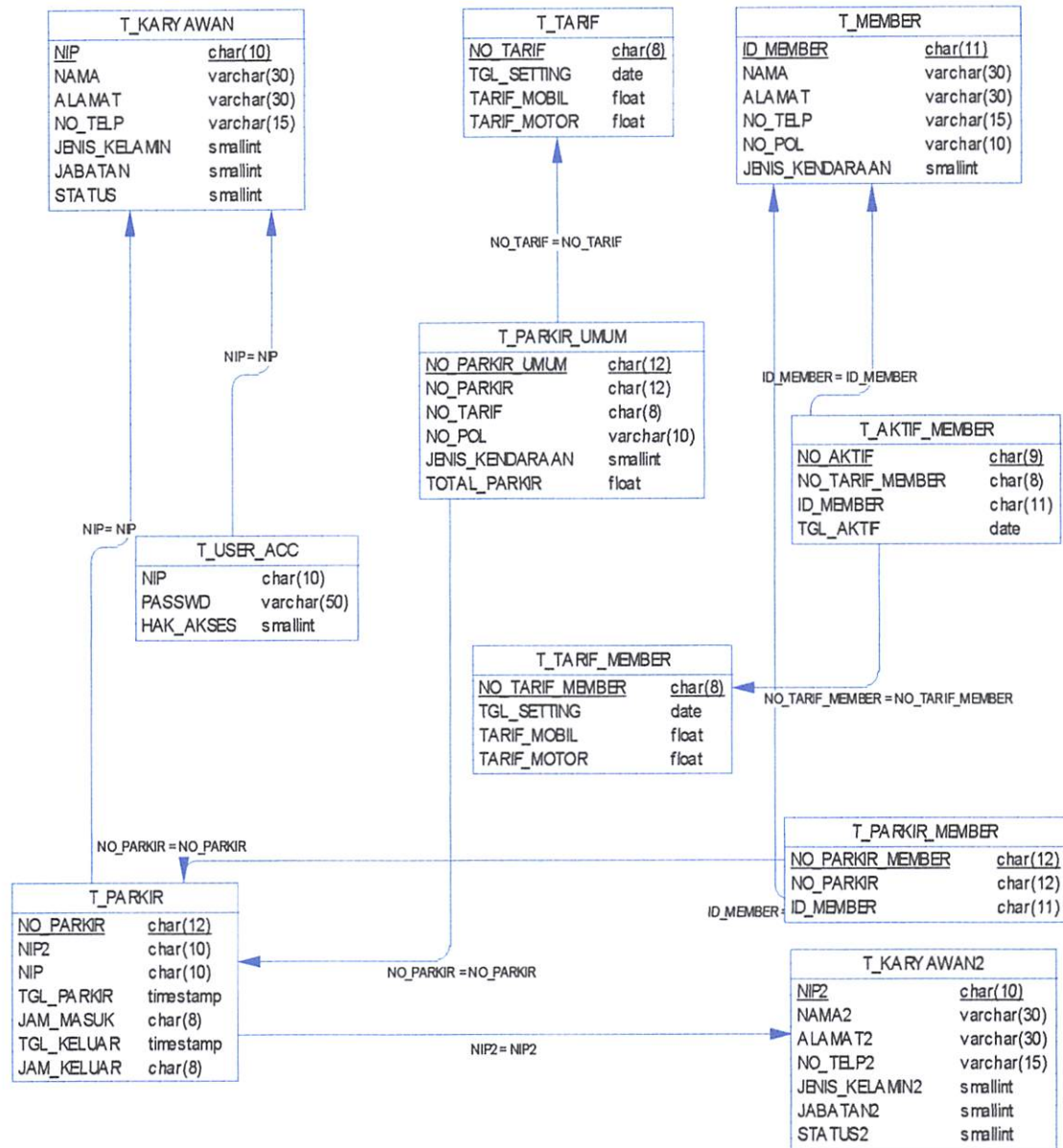
3.2.2.1 Relasi Antar Tabel

Relasi antar tabel dalam basis data sistem informasi manajemen parkir digambarkan dalam bentuk konsep atau *Conceptual Data Model* (CDM) seperti yang ditunjukkan dalam gambar 3.6 :



Gambar 3.6 CDM Data Sistem Informasi Parkir

Relasi tabel dalam basis data sistem informasi manajemen parkir juga di gambarkan dalam bentuk fisik atau *Physical Data Model (PDM)*, seperti yang ditunjukkan dalam gambar 3.7 :



Gambar 3.7 PDM Data Sistem Informasi Parkir


3.2.2.2 Struktur Tabel-Tabel yang digunakan

Berdasarkan CDM di atas, maka struktur tabel-tabel yang digunakan adalah:

1. Tabel T_KARYAWAN

Tabel karyawan ini digunakan untuk menyimpan data dari karyawan, seperti yang ditunjukkan pada tabel 3.1:


Tabel 3.1 Struktur Tabel Karyawan

No	Nama Kolom	Tipe Data	Primary Key
1	NIP	Char (10)	
2	NAMA	Varchar (30)	-
3	ALAMAT	Varchar (30)	-
4	NO_TELP	Varchar (15)	-
5	JENIS_KELAMIN	bit (1)	-
6	JABATAN	tinyint (1)	-
7	STATUS	Bit (1)	-

2. Tabel T_TARIF

Tabel Tarif ini digunakan untuk menyimpan data dari tarif parkir umum, seperti yang ditunjukkan pada tabel 3.2 :

Tabel 3.2 Struktur Tabel Tarif


No	Nama Kolom	Tipe Data	Primary Key
1	NO_TARIF	Char (8)	
2	TGL_SETTING	Datetime (8)	-

3	TARIF_MOBIL	Money (8)	-
4	TARIF_MOTOR	Money (8)	-

3. Tabel T_User_Acc

Tabel User_Acc ini digunakan untuk menyimpan password dari pengguna aplikasi, seperti yang ditunjukkan pada tabel 3.3 :


Tabel 3.3 Struktur Tabel User_Access

No	Nama Kolom	Tipe Data	Primary Key
1	NIP	Char(10)	
2	PASSWORD	Varchar (50)	-
3	HAK AKSES	Tinyint (1)	-

4. Tabel T_KAPASITAS

Tabel kapasitas ini digunakan untuk menyimpan data kapasitas dari areal parkir, seperti yang ditunjukkan pada tabel 3.4 :


Tabel 3.4 Struktur Tabel Kapasitas

No	Nama Kolom	Tipe Data	Primary Key
1	NO_KAPASITAS	Char(8)	
2	TGL_SETTING	Datetime (8)	-
3	KAPASITAS_MOBIL	Int(4)	-
4	KAPASITAS_MOTOR	Int(4)	-

5. Tabel T_MEMBER

Tabel member ini digunakan untuk menyimpan data dari member pengguna area parkir, seperti yang ditunjukkan pada tabel 3.5:


Tabel 3.5 Struktur Tabel member

No	Nama Kolom	Tipe Data	Primary Key
1	ID_MEMBER	Char(11)	
2	NAMA	Varchar (30)	-
3	ALAMAT	Varchar (30)	-
4	NO_TELP	Varchar (15)	-
5	NO_POL	varchar (10)	-
6	JENIS_KENDARAAN	Bit (1)	-
7	MERK	Varchar (30)	-
8	WARNA	Varchar (10)	-

6. Tabel T_TARIF_MEMBER

Tabel tarif member ini digunakan untuk menyimpan data tarif parkir yang diberlakukan untuk member, seperti yang ditunjukkan pada tabel 3.6 :


Tabel 3.6 Struktur Tabel Tarif_Member

No	Nama Kolom	Tipe Data	Primary Key
1	NO_TARIF_MEMBER	Char(8)	
2	TGL_SETTING	datetime(8)	-
3	TARIF_MOBIL	Money(8)	-
4	TARIF_MOTOR	Money(8)	-

7. Tabel T_AKTIF_MEMBER

Tabel aktif member ini digunakan untuk menyimpan data perpanjangan masa aktifasi member, seperti yang ditunjukkan pada tabel 3.7 :


Tabel 3.7 Struktur Tabel Parkir_Mobil

No	Nama Kolom	Tipe Data	Primary Key
1	NO_AKTIF	Char(11)	
2	TGL_AKTIF	datetime(8)	-
3	ID_MEMBER	Char(11)	-
4	NO_TARIF_MEMBER	Char(8)	-

8. Tabel T_PARKIR

Tabel parkir ini digunakan untuk menyimpan data kendaraan yang parkir, seperti yang ditunjukkan pada tabel 3.8 :


Tabel 3.8 Struktur Tabel Parkir

No	Nama Kolom	Tipe Data	Primary Key
1	NO_PARKIR	Char(12)	
2	TGL_PARKIR	datetime(8)	-
3	JAM_MASUK	Char(8)	-
4	TGL_KELUAR	datetime(8)	-
5	JAM_KELUAR	Char(8)	-
6	OPERATOR_MASUK	Char(10)	-
7	OPERATOR_KELUAR	char (10)	-

9. Tabel T_PARKIR_UMUM

Tabel parkir umum ini digunakan untuk menyimpan data kendaraan yang parkir sebagai pengguna umum, seperti yang ditunjukkan pada tabel 3.9 :


Tabel 3.9 Struktur Tabel Parkir_ Umum

No	Nama Kolom	Tipe Data	Primary Key
1	NO_PARKIR_UMUM	Char(12)	
2	NO_PARKIR	char(12)	-
3	NO_POL	varchar (10)	-
4	JENIS_KENDARAAN	Bit(1)	-
5	NO_TARIF	Char(8)	-
6	TOTAL_PARKIR	money (8)	-

10. Tabel T_PARKIR_MEMBER

Tabel parkir member ini digunakan untuk menyimpan data kendaraan yang parkir sebagai member, seperti yang ditunjukkan pada tabel 3.10 :

Tabel 3.10 Struktur Tabel Parkir_Member


No	Nama Kolom	Tipe Data	Primary Key
1	NO_PARKIR_MEMBER	Char(13)	
2	NO_PARKIR	Char(12)	-
3	ID_MEMBER	Char(11)	-

11. Tabel T_PENGELUARAN

Tabel pengeluaran ini digunakan untuk menyimpan data pengeluaran, yang digunakan untuk operasional parkir seperti yang ditunjukkan pada tabel 3.11

:


Tabel 3.11 Struktur Tabel Pengeluaran

No	Nama Kolom	Tipe Data	Primary Key
1	NO_PENGELUARAN	Char (9)	
2	TGL_PENGELUARAN	Datetime (8)	-
3	TOTAL_PENGELUARAN	Money (8)	-

12. Tabel T_PENGELUARAN_DETIL

Tabel pengeluaran detail ini digunakan untuk menyimpan data pengeluaran secara detail, seperti yang ditunjukkan pada tabel 3.12 :


Tabel 3.12 Struktur Tabel Pengeluaran_Detil

No	Nama Kolom	Tipe Data	Primary Key
1	NO_PENGELUARAN_D	Char (12)	
2	TGL_PENGELUARAN	Datetime (8)	-
3	KETERANGAN	Varchar(50)	-
4	TOTAL_PENGELUARAN	Money (8)	-

13. Tabel T_PENDAPATAN

Tabel pendapatan ini digunakan untuk menyimpan data pendapatan dari parkir, seperti yang digunakan pada tabel 3.13 :


Tabel 3.13 Struktur Tabel Pendapatan

No	Nama Kolom	Tipe Data	Primary Key
1	NO_PENDAPATAN	Char (9)	
2	TGL_PENDAPATAN	Datetime (8)	-
3	TOTAL_PARKIR_UMUM	Money (8)	-
4	TOTAL_PARKIR_MEMBER	Money (8)	-

14. Tabel T_LABA_RUGI

Tabel laba rugi ini digunakan untuk menyimpan data tentang laba atau rugi yang didapatkan, seperti yang ditunjukkan pada tabel 3.14 :

Tabel 3.14 Struktur Tabel Laba_Rugi

No	Nama Kolom	Tipe Data	Primary Key
1	NO_LABARUGI	Char (7)	
2	PENDAPATAN	Money (8)	-
3	PEGELUARAN	Money (8)	-

3.2.3. Desain Antarmuka Aplikasi

Sesuai dengan spesifikasi sistem di atas, sistem informasi ini diharapkan dapat dengan mudah dipakai oleh *user* yang mengoperasikan sistem ini. Untuk itu harus dibuat desain antarmuka yang mudah dipahami dan tidak terlalu rumit.

3.2.3.1. Desain Form Login

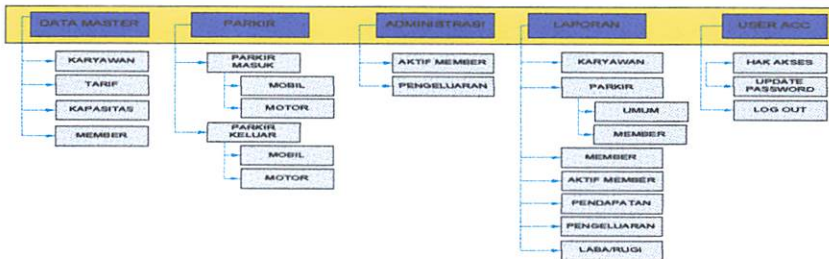
Form ini digunakan untuk *login* bagi *user* sesuai dengan hak aksesnya masing-masing, untuk bisa masuk ke dalam aplikasi sistem informasi. Gambar 3.8 merupakan desain form login.



Gambar 3.8 Form Login Pengguna Sistem Informasi

3.2.3.2. Desain Menu Aplikasi

Menu aplikasi terdiri dari berbagai menu yang terdapat di sistem informasi *management* parkir. Gambar 3.9 merupakan desain menu aplikasi.



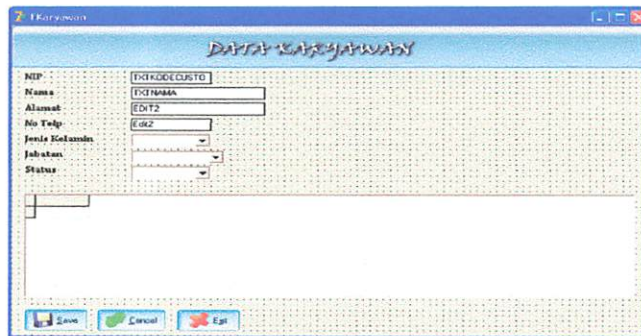
Gambar 3.9 Desain Menu Aplikasi Sistem Informasi Parkir

Spesifikasi menu dari masing-masing modul disajikan dalam desain menu-menu *pop-up* di dalam *childform* disusun secara vertikal pada bagian atas *childform* seperti gambar di atas.

Menu dan *childform* tersebut tidak bisa diakses semuanya, kecuali oleh pimpinan dan administrator yang akan diberi hak akses penuh. Karyawan di bawahnya, hanya akan dapat mengakses menu dengan hak akses yang diberikan, sesuai dengan tugas dan pekerjaannya masing-masing.

3.2.3.3. Desain Form Setting Karyawan

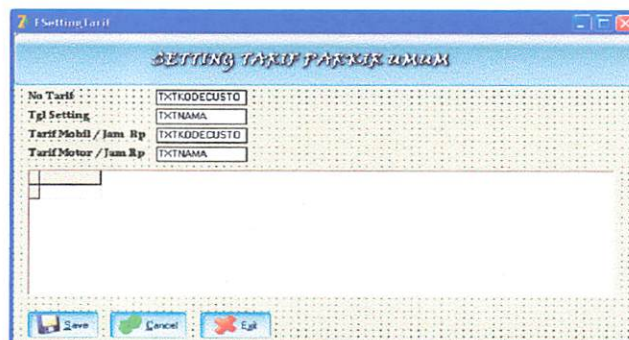
Rancangan ini memungkinkan administrator *men-set* identitas dari karyawan. Gambar 3.10 merupakan desain tampilan form karyawan:



Gambar 3.10 Rancangan Form Setting karyawan

3.2.3.4. Desain Form Setting Tarif Parkir Umum

Rancangan ini ditujukan bagi administrator untuk dapat *men-set* biaya parkir. Biaya parkir disetting harga per jam sesuai yang diinginkan. Gambar 3.11 merupakan desain tampilan form tarif parkir umum:



Gambar 3.11 Rancangan Form Setting Tarif Parkir Umum

3.2.3.5. Desain Form Setting Tarif Parkir Member

Rancangan ini ditujukan bagi administrator untuk dapat men-set biaya parkir yang diperuntukkan untuk member. Biaya parkir disetting harga per jam sesuai yang diinginkan. Gambar 3.12 merupakan desain tampilan tarif parkir member:

No Tarif Member	TXTKODECUSTO
Tgl Setting	TXTNAMA
Tarif Mobil / Bln Rp	TXTKODECUSTO
Tarif Motor / Bln Rp	TXTNAMA

Gambar 3.12 Rancangan Form Setting Tarif Parkir Member

3.2.3.6. Desain Form Setting Kapasitas Tempat Parkir

Rancangan ini dibuat untuk administrator agar bisa men-set jumlah tempat parkir yang tersedia dalam gedung. Gambar 3.13 merupakan desain tampilan form kapasitas parkir:

No Kapasitas	TXTKODECUSTO
Tgl Setting	TXTNAMA
Kapasitas Mobil	TXTKODECUSTO
Kapasitas Motor	TXTNAMA

Gambar 3.13 Rancangan Form Setting Jumlah Tempat Parkir

3.2.3.7. Desain Form Setting Member

Rancangan ini dibuat untuk administrasi parkir agar bisa men-set identitas dari member. Gambar 3.14 merupakan desain tampilan form member:

The screenshot shows a software window titled 'F Member' with a subtitle 'SETTING DATA MEMBER'. The form contains the following fields: ID MEMBER (TXTKODECUSTO), Nama (TXTNAMA), Alamat (EDIT2), No Telp (Edi2), No Pol (Edi2), Jenis Kendaraan (dropdown menu), Merk (Edi2), and Warna (Tul'wana). At the bottom, there are three buttons: Save, Cancel, and Exit.

Gambar 3.14 Rancangan Form Setting Member

3.2.3.8. Desain Form Aktifasi Member

Rancangan ini dibuat untuk administrasi parkir agar bisa men-set aktivasi member yang telah habis masa berlakunya. Gambar 3.15 merupakan desain tampilan form aktif member:

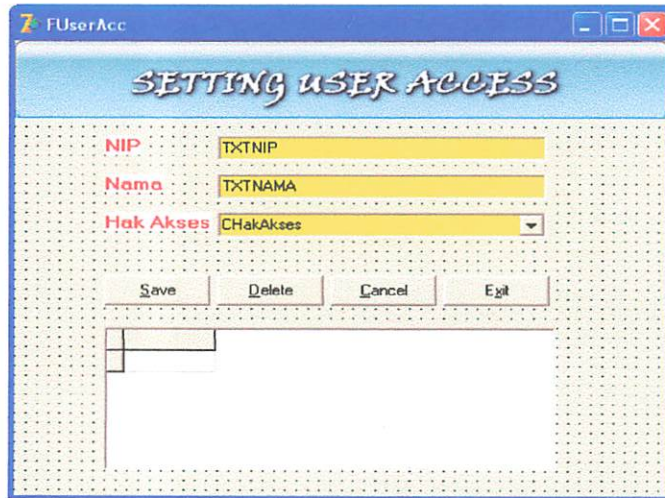
The screenshot shows a software window titled 'F AktifMember' with a subtitle 'SETTING AKTIF KEMBER'. The form contains the following fields: No Aktif (TXTKODECUSTO), Tanggal (TXTNAMA), ID Member (TXTKODECUSTO), Berlaku s/d (TXTNAMA), Nama Member (TXTKODECUSTOMER), NO POL (TXTKODECUSTO), JENIS KENDARAAN (TXTKODECUSTO), MERK (TXTKODECUSTOMER), and Total Harga Rp (TXTNAMA). At the bottom, there are three buttons: Save, Cancel, and Exit.

Gambar 3.15 Rancangan Form Setting Aktif Member

3.2.3.9. Desain Form Setting Hak Akses

Rancangan ini dibuat untuk mengatur hak akses dari setiap karyawan.

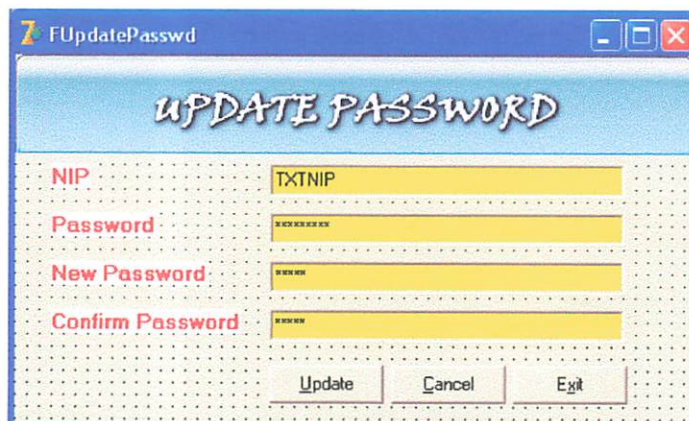
Gambar 3.16 merupakan desain tampilan form hak akses:

The image shows a software window titled 'FUserAcc' with a subtitle 'SETTING USER ACCESS'. The window has a blue header and a light blue background with a dotted pattern. It contains three input fields: 'NIP' with a yellow text box labeled 'TXTNIP', 'Name' with a yellow text box labeled 'TXTNAMA', and 'Hak Akses' with a yellow dropdown menu labeled 'CHakAkses'. Below these fields are four buttons: 'Save', 'Delete', 'Cancel', and 'Exit'. At the bottom, there is a large empty rectangular area.

Gambar 3.16 Rancangan Form Setting Hak Akses

3.2.3.10. Desain Form Update Password

Rancangan ini dibuat agar karyawan yang mengakses aplikasi ini bias mengubah passwordnya sesuai yang diinginkan. Gambar 3.17 merupakan desain tampilan form update password:

The image shows a software window titled 'FUpdatePasswd' with a subtitle 'UPDATE PASSWORD'. The window has a blue header and a light blue background with a dotted pattern. It contains four input fields: 'NIP' with a yellow text box labeled 'TXTNIP', 'Password' with a yellow text box containing masked characters, 'New Password' with a yellow text box containing masked characters, and 'Confirm Password' with a yellow text box containing masked characters. Below these fields are three buttons: 'Update', 'Cancel', and 'Exit'.

Gambar 3.17 Rancangan Form Update Password

3.2.3.11. Desain Form Parkir Masuk Motor

Rancangan form ini diperuntukkan bagi operator masuk untuk meng-*input* nomor plat kendaraan bagi pengguna parkir umum dan ID member yang terdaftar sebagai member yang memasuki areal parkir. Gambar 3.18 merupakan desain tampilan form parkir masuk motor:

The screenshot shows a software window titled "Parkir Masuk (MOTOR)" with a blue header. The header contains the logo "Plaza Taman" and "Bontang Kal-Tim". On the right side of the header, there is a digital display showing "00.00.00" and "Waktu Sekarang". Below the header, the form is divided into several sections. The top section has "No Parkir" and "Jenis Parkir" fields. Below that, there are two main sections: "PARKIR UMUM" and "PARKIR MEMBER". The "PARKIR UMUM" section has a "No Pol" field. The "PARKIR MEMBER" section has fields for "ID Member", "No Pol", "Nama", "Merk", "Warna", and "Tgl Exp Kartu". At the bottom left, there is a "Sisa Kapasitas" section with a digital display showing "PE 800!!!". Below that, there are fields for "No Karyawan" and "Nama Karyawan". At the bottom of the form, there are three buttons: "Save", "Cancel", and "Exit".

Gambar 3.18 Rancangan Form Parkir Masuk Motor

3.2.3.12. Desain Form Parkir Masuk Mobil

Rancangan form ini diperuntukkan bagi operator masuk mobil untuk meng-*input* nomor plat kendaraan bagi pengguna parkir umum dan ID member yang terdaftar sebagai member yang memasuki areal parkir. Gambar 3.19 merupakan desain tampilan form parkir masuk mobil:

The image shows a software window titled "Parkir Masuk (MOBIL)" with a blue header. The header contains the logo "Plaza Taman" and "Bontang Kal-Tim". On the right, there is a digital display showing "End 00.00.00" and "Waktu Sekarang 00.00.00". The main form area is divided into several sections:

- Top Left:** "No Parkir" with a text input field containing "TXTKODECUST".
- Top Center:** "PARKIR MASUK MOBIL" in large red letters.
- Top Right:** "Jenis Parkir" with a dropdown menu showing "CJenisParkir".
- Middle Left:** "PARKIR UMUM" section with "No Pol" and a text input field containing "TXTKODEC".
- Middle Right:** "PARKIR MEMBER" section with fields for "ID Member", "No Pol", "Nama", "Merk", "Warna", and "Tgl Exp Kartu", all containing "TXTKODECUSTO".
- Bottom Left:** "No Karyawan" and "Nama Karyawan" fields, both containing "TXTKODECUS".
- Bottom Center:** A digital display showing "PE800!!" and a parking fee of "00.00.00".
- Bottom Right:** "Save", "Cancel", and "Exit" buttons.

Gambar 3.19 Rancangan Form Parkir Masuk Mobil

3.2.3.13. Desain Form Parkir Keluar Motor

Rancangan form ini diperuntukkan bagi operator keluar motor untuk meng-*input* kendaraan motor pada saat keluar dari areal parkir. Form ini juga memberi informasi pada operator keluar mengenai biaya parkir yang harus dikutip dari kendaraan yang bersangkutan. Gambar 3.20 merupakan desain tampilan form parkir keluar motor:

Gambar 3.20 Rancangan Form Parkir Keluar Motor

3.2.3.14. Desain Form Parkir Keluar Mobil

Rancangan form ini diperuntukkan bagi operator keluar untuk meng-*input* kendaraan mobil pada saat keluar dari areal parkir. Form ini juga memberi informasi pada operator keluar mengenai biaya parkir yang harus dikutip dari kendaraan yang bersangkutan. Gambar 3.21 merupakan desain tampilan form parkir keluar mobil:

Parkir Keluar (MOBIL)

Plaza Taman
Bandung Kal-Tim

Hari: 00.00.00
Waktu Sekarang: 00.00.00

No Pol:

PARKIR KELUAR MOBIL

DATA MEMBER

No Parkir:

Tgl Parkir:

Jam Masuk:

Jam Keluar:

Lama Parkir: hari

Total Parkir: Rp

ID Member:

Nama:

Merk:

Warna:

No Karyawan:

Nama Karyawan:

Buttons: Save, Cancel, Exit

Gambar 3.21 Rancangan Form Parkir Keluar Mobil

3.2.3.15. Desain Form Pengeluaran Parkir

Rancangan ini dibuat untuk administrasi parkir agar bisa meng-inputkan pengeluaran yang digunakan untuk operasional parkir. Gambar 3.22 merupakan desain tampilan form pengeluaran:

Pengeluaran

DATA PENGELUARAN

No Pengeluaran: Tanggal:

Keterangan: Total:

Biaya Rp:

Buttons: Save, Cancel, Exit, Delete

Gambar 3.22 Rancangan Form Setting Pengeluaran Parkir

3.2.3.16. Desain Form Filter Laporan

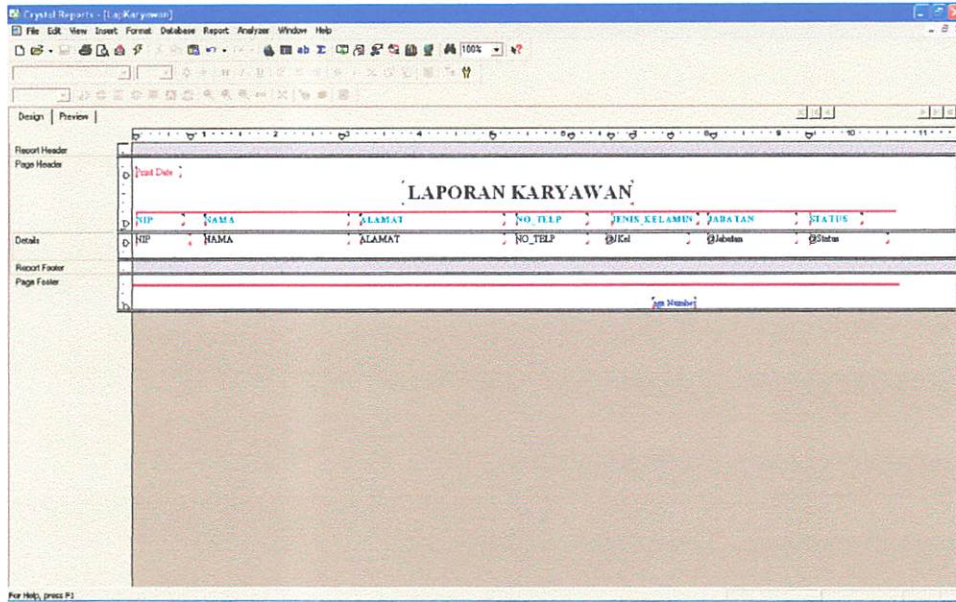
Rancangan ini berfungsi untuk mem-filter laporan yang ingin ditampilkan. Tapilan laporan langsung bisa di cetak. Gambar 3.23 merupakan desain tampilan form filter laporan:



Gambar 3.23 Rancangan Form Filter Laporan

3.2.3.17. Desain Laporan

Laporan bisa dicetak langsung dari form laporan yang terdapat tombol cetak. Untuk tampilan laporan pada halaman laporan, dibuat dengan menggunakan Crystal Reports. Gambar 3.24 merupakan desain tampilan laporan pada Crystal Reports:



Gambar 3.24 Rancangan Laporan

Sistem pelaporan pada sistem informasi parkir ini terdiri atas beberapa laporan, yaitu:

1. Laporan Pegawai

Laporan ini dibuat untuk mengetahui Data-data Pegawai yang bertugas di bagian parkir.

2. Laporan Member

Laporan ini dibuat untuk mengetahui Data-data member yang telah terdaftar.

3. Laporan Aktif Member

Laporan ini dibuat untuk mengetahui masa aktif dari member.

4. Laporan Parkir Umum

Laporan ini dibuat untuk mengetahui Data-data transaksi parkir umum yang terjadi.

5. Laporan Parkir Member

Laporan ini dibuat untuk mengetahui Data-data transaksi parkir member yang terjadi.

6. Laporan Pendapatan

Laporan ini dibuat untuk mengetahui pendapatan parkir.

7. Laporan Pengeluaran

Laporan ini dibuat untuk mengetahui Data-data pengeluaran yang digunakan untuk operasional parkir.

8. Laporan Laba/Rugi

Laporan ini dibuat untuk mengetahui keuntungan atau kerugian yang didapat dari parkir.

3.2.3.18.Desain Karcis Masuk

Rancangan ini berfungsi untuk menampilkan data dari kendaraan yang akan masuk parkir. Data yang ditampilkan pada karcis masuk ini adalah no parkir, tanggal parkir, jam masuk, nomor kendaraan, jenis kendaraan yang parkir dan operator masuk. Gambar 3.25 merupakan desain tampilan karcis masuk:

1		2		3		4		5		6		7	
<i>Plaza Caman</i> <i>Benteng Kalimantan Timur</i>													
KARCISPARKIR MASUK													
No Parkir	:	NO_PARKIR	:										
Tgl Parkir	:	TGL_PARKIR	:										
Jam Masuk	:	JAM_MASUK	:										
No Pol	:	NO_POL	:										
Jenis Parkir	:	@Jenis_Kendaraan	:										
Operator	:	NAMA	:										

Gambar 3.25 Rancangan Karcis Masuk

3.2.3.19. Desain Karcis Keluar

Rancangan ini berfungsi untuk menampilkan data dari kendaraan yang akan keluar parkir. Data yang ditampilkan pada karcis keluar ini adalah no parkir, tanggal parkir, jam masuk, tanggal keluar, jam keluar, nomor kendaraan, jenis kendaraan yang parkir, total biaya parkir dan operator keluar. Gambar 3.26 merupakan desain tampilan karcis keluar:

1		2		3		4		5		6		7		8		9		10	
<i>Plaza Caman</i> <i>Benteng Kalimantan Timur</i>																			
KARCISPARKIR KELUAR																			
No Parkir	:	NO_PARKIR	:																
Tgl Parkir	:	TGL_PARKIR	:	Tgl Keluar	:	TGL_KELUAR	:												
Jam Masuk	:	JAM_MASUK	:	Jam Keluar	:	JAM_KELUAR	:												
No Pol	:	NO_POL	:																
Jenis Parkir	:	@Jenis_Kendar	:	Total Parkir	:	T_L_PARKIR	:												
Operator	:	NAMA	:																

Gambar 3.26 Rancangan Karcis Keluar

BAB IV

IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Implementasi Sistem

Tahap implementasi pengembangan perangkat lunak merupakan proses perubahan spesifikasi sistem menjadi sistem yang dapat dijalankan. Tahap ini merupakan lanjutan dari proses perancangan, yaitu proses pemrograman perangkat lunak sesuai dengan spesifikasi dan desain sistem.

Aplikasi sistem informasi parkir ini menggunakan basis data SQL server yang berfungsi sebagai media penyimpanan data atau informasi yang terkumpul, yang terdiri dari beberapa tabel yang saling berhubungan. Sedangkan untuk menjembatani antara informasi yang akan dibuat dengan basis data yang ada, digunakan bahasa pemrograman Delphi 7.

Ada beberapa menu pada aplikasi ini. Diantaranya adalah login, menu utama, setting pegawai, setting tarif, setting kapasitas, setting member, setting user acces, aktifasi member, parkir masuk, parkir keluar, setting pengeluaran dan pelaporan.

4.2 Pengujian Aplikasi

Pada tahap pengujian, penulis akan mencoba membandingkan kesesuaian antara *input* dari *user* dengan kebutuhan *input* aplikasi. Pengujian akan dilakukan dengan memasukkan *input* yang dianggap sesuai dengan kebutuhan dan *input* yang tidak sesuai dengan kebutuhan dari aplikasi “ Sistem

Informasi *Management Parkir Modern Berbasis Client Server*". Hal ini dilakukan untuk menganalisis kinerja perangkat lunak yang telah dibuat. Hasil analisis ini akan sangat bermanfaat dalam pengembangan aplikasi ini dikemudian hari.

4.2.1 Pengujian Form Login

Pada proses login ini digunakan oleh semua pegawai untuk dapat mengakses aplikasi ini, dengan meng-input NIP dan *Password* dari pegawai. Seperti yang tampak pada gambar 4.1.



Gambar 4.1 Form Login

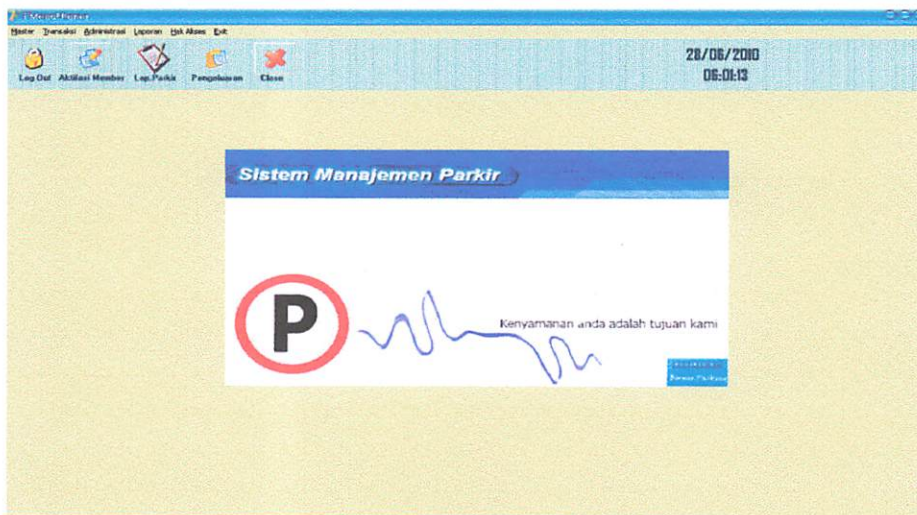
Kemudian NIP dan password akan di cocokkan terlebih dahulu. Apabila input yang dilakukan sesuai, maka sistem akan melanjutkan ke menu utama. Jika input yang dimasukkan tidak sesuai, akan keluar pesan seperti di gambar 4.2.



Gambar 4.2 Pesan Login Salah

4.2.2 Pengujian Form Menu Utama Aplikasi

Pengujian yang dilakukan pada form menu utama ini adalah untuk mengetahui apakah menu-menu yang digunakan oleh user telah sesuai dengan hakakses pengguna aplikasi. Menu –menu ini akan aktif setelah *user* melakukan *login* ke aplikasi ini. Gambar 4.3 merupakan tampilan dari menu utama.



Gambar 4.3 Menu Utama

4.2.3 Pengujian Form Setting Karyawan

Pengujian yang dilakukan pada form setting karyawan meliputi *textedit*, *combobox* dan *button* yang tersedia, apakah sudah berjalan sesuai dengan prosedur atau belum berfungsi dengan optimal. Seperti yang ditunjukkan pada gambar 4.4.

The screenshot shows a window titled "Setting Karyawan" with a header "DATA KARYAWAN". It contains several input fields: NIP, Nama (filled with "IHGAN"), Alamat (filled with "JL ENDE 99"), No Telp (filled with "11223344"), Jenis Kelamin (dropdown menu with "PRIA" selected), Jabatan (dropdown menu with "OPERATOR" selected), and Status (dropdown menu with "GIA" selected). Below the fields is a table with the following data:

NIP	Nama	Alamat	No Telp	Jenis Kelamin
123456789	HARIS	JL KANJURHAN 456	12315654	PRIA
1234567890	SANTI	JL MERDEKA 986	710789	WANITA
222222222	SAASA	JL PANCASILA	09677	WANITA
333333333	RICKI	JL SURABAYA	443355	PRIA
444444444	ERIC	KEBAGUSAN 56784	777	PRIA
555555555	ARY ARTANTO	JL MALANG 45	341341	PRIA

At the bottom of the form are three buttons: "Save", "Cancel", and "Exit".

Gambar 4.4 Entri Data Karyawan

Pada *button save* mempunyai fungsi untuk menyimpan data yang telah diinputkan pada *textedit*. Jika data berhasil disimpan maka akan muncul diDBGrid. Seperti yang ditunjukkan pada gambar 4.5.

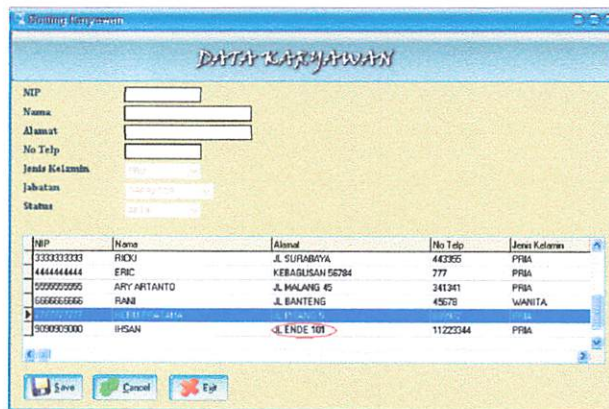
The screenshot shows the same "Setting Karyawan" window. The input fields are now empty. The table below shows the updated data after saving:

NIP	Nama	Alamat	No Telp	Jenis Kelamin
333333333	RICKI	JL SURABAYA	443355	PRIA
444444444	ERIC	JL MALANG 45	341341	PRIA
555555555	ARY ARTANTO	JL MALANG 45	341341	PRIA
666666666	RAH	JL BAYTENG	45678	WANITA
777777777	HERLI PRATAMA	JL PESANG 5	889977	PRIA
999999999	IHSAN	JL ENDE 99	11223344	PRIA

The "Save" button is highlighted in green, indicating it was just clicked. The "Cancel" and "Exit" buttons are also visible at the bottom.

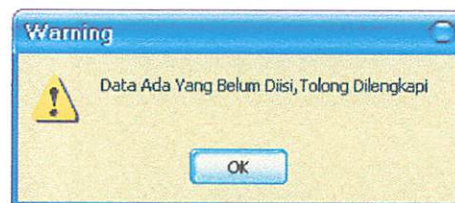
Gambar 4.5 Data Karyawan Tersimpan

Selain itu pada form ini dapat meng-*update* data karyawan. Cara yang dilakukan dengan meng-*inputkan* NIP di *textedit* NIP . Kemudian dengan menekan *enter* maka data karyawan akan ditampilkan di *textedit*. Sehingga dapat merubah data yang diinginkan. Pada kondisi ini *button save* akan berubah menjadi *button update* dengan menekan *button* tersebut maka data akan secara otomatis akan berubah. Seperti yang ditunjukkan pada gambar 4.6.



Gambar 4.6 Data yang telah di *update*

Jika terdapat data yang belum diisi, maka akan muncul pesan peringatan seperti gambar 4.7:

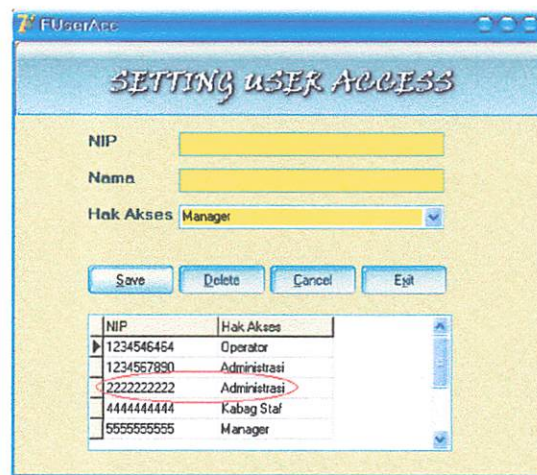


Gambar 4.7 Pesan Data Tidak Lengkap

4.2.4 Pengujian Form *User Access*

. Pengujian yang dilakukan pada form *user access* meliputi *textedit*, *combo box* dan *button* yang tersedia, apakah sudah berjalan sesuai dengan prosedur atau belum berfungsi dengan optimal.

Untuk memberikan hak akses kepada pengguna aplikasi dengan menginputkan NIP pada *textedit* untuk NIP kemudian dengan menekan enter maka muncul nama dari karyawan. Kemudian dapat memilih hak akses yang diberikan. Dengan menekan *button Save* maka data akan tersimpan. Apabila data telah tersimpan maka akan muncul di DBGrid. Gambar 4.8 merupakan tampilan dari form tersebut:

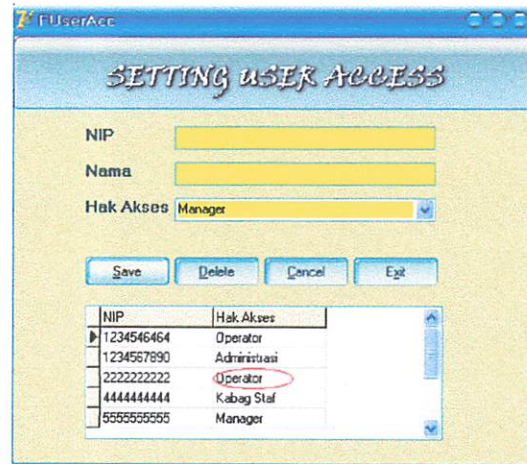


Gambar 4.8 Entri *User Access*

Untuk memperbarui atau merubah hak akses karyawan, sama dengan cara menyimpan tetapi pada saat *textedit* NIP di tekan *enter* maka *button save* berubah menjadi *button update*. Dengan menekan *button update* maka hak akses akan

berubah. Jika berhasil, maka data yang ditampilkan di DBGrid akan berubah.

Gambar 4.9 merupakan tampilan dari form tersebut:



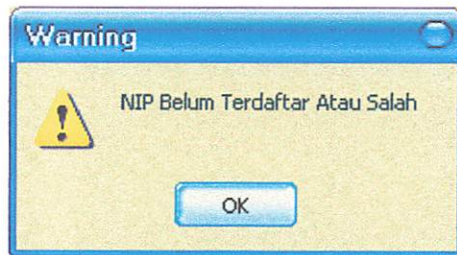
Gambar 4.9 Data Telah di *Update*

Untuk menghapus hak akses karyawan, prosedur sama dengan menyimpan dan meng-*update* tetapi yang membedakan yaitu dengan menekan tombol *delete*. Jika berhasil maka data di DBGrid akan terhapus. Gambar 4.10 merupakan tampilan dari form tersebut:



Gambar 4.10 Data Telah di *Delete*

Jika pengguna salah meng-inputkan NIP maka akan muncul pesan peringatan seperti gambar pada 4.11:



Gambar 4.11 Pesan Kesalahan *Input* NIP

4.2.5 Pengujian Form *Update Password*

Pengujian yang dilakukan pada form *Update Password* meliputi *textedit* dan *button* yang tersedia, apakah sudah berjalan sesuai dengan prosedur atau belum berfungsi dengan optimal. Jika ingin membuat *password* yaitu dengan cara menginputkan NIP pada *textedit* untuk NIP dan mengisikan *password* yang diinginkan. Gambar 4.12 merupakan tampilan dari form tersebut:



Gambar 4.12 Entri *Update Password*

4.2.6 Pengujian Form Setting Tarif Parkir Umum

Pengujian yang dilakukan pada form Setting Tarif Parkir Umum meliputi *textedit* dan *button* yang tersedia, apakah sudah berjalan sesuai dengan prosedur atau belum berfungsi dengan optimal.

Untuk mengatur tarif parkir umum, dengan meng-*inputkan* biaya tarif pada *textedit* tarif mobil dan tarif motor. Dengan menekan *button Save* maka data akan tersimpan. Apabila data berhasil tersimpan maka akan muncul di DBGrid. Gambar 4.13 merupakan tampilan dari form tersebut:

No Tarif	Tgl Setting	Tarif Mobil	Tarif Motor
T-100324	24/06/2010	4000	2000
T-100313	13/03/2010	2000	1000

Gambar 4.13 Entri Setting Tarif Parkir Umum

Untuk meng-*input* jumlah tarif dari parkir mobil dan motor harus menggunakan nominal angka. Jika ada kesalahan *input*, akan muncul pesan seperti pada gambar 4.14:



Gambar 4.14 Pesan Kesalahan *Input* Tarif Umum

4.2.7 Pengujian Form Setting Tarif Parkir Member

Pengujian yang dilakukan pada form Setting Tarif Parkir Member meliputi *textedit* dan *button* yang tersedia, apakah sudah berjalan sesuai dengan prosedur atau belum berfungsi dengan optimal.

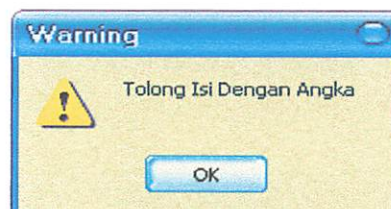
Untuk mengatur tarif parkir member, dengan meng-*inputkan* biaya tarif pada *textedit* tarif mobil dan tarif motor. Dengan menekan *button Save* maka data akan tersimpan. Apabila data berhasil tersimpan maka akan muncul di DBGrid.

Gambar 4.15 merupakan tampilan dari form tersebut:

No Tarif	Tgl Setting	Tarif Mobil	Tarif Motor
T-100524	24/05/2010	100000	50000
T-100315	15/03/2010	200000	100000

Gambar 4.15 Entri Setting Tarif Parkir Member

Untuk meng-*inputkan* jumlah tarif dari parkir member mobil dan motor harus menggunakan nominal angka. Jika ada kesalahan *input*, akan muncul pesan seperti pada gambar 4.16:



Gambar 4.16 Pesan Kesalahan *Input* Tarif Member

4.2.8 Pengujian Form Setting Kapasitas Tempat Parkir

Pengujian yang dilakukan pada form Setting Kapasitas Tempat Parkir meliputi *textedit* dan *button* yang tersedia, apakah sudah berjalan sesuai dengan prosedur atau belum berfungsi dengan optimal.

Untuk mengatur kapasitas parkir ,dengan meng-*inputkan* kapasitas parkir pada *textedit* kapasitas mobil dan kapsitas motor. Dengan menekan *button Save* maka data akan tersimpan. Apabila data berhasil tersimpan maka akan muncul di DBGrid. Gambar 4.17 merupakan tampilan dari form tersebut:

No Kapasitas	Tgl Setting	Kapasitas Mobil	Kapasitas Motor
K-100624	24/06/2010	200	100
K-100618	18/06/2010	5	10
K-100317	17/03/2010	40	90

Gambar 4.17 Entri Kapasitas Parkir

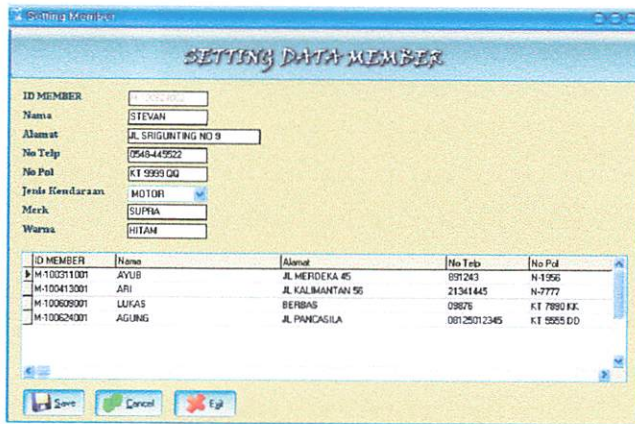
Untuk meng-*inputkan* jumlah kapasitas dari parkir mobil dan motor harus menggunakan nominal angka. Jika ada kesalahan *inputan*,akan muncul pesan seperti pada gambar 4.18:



Gambar 4.18 Pesan Kesalahan *Input* Kapasitas

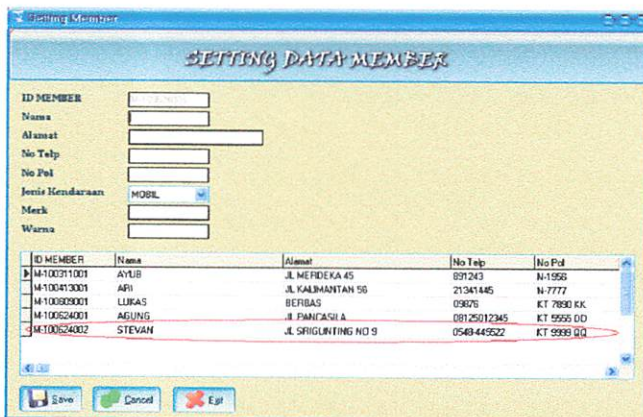
4.2.9 Pengujian Form Seting Member

Pengujian yang dilakukan pada form Seting Member meliputi *textedit*, *combobox* dan *button* yang tersedia, apakah sudah berjalan sesuai dengan prosedur atau belum berfungsi dengan optimal. Gambar 4.19 merupakan tampilan dari form tersebut:



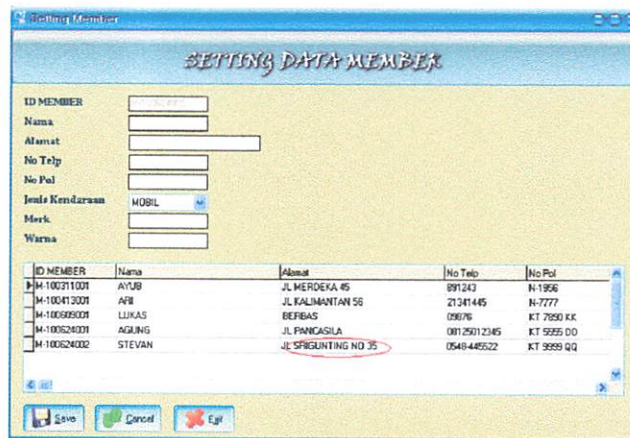
Gambar 4.19 Entri Data Member

Pada *button save* mempunyai fungsi untuk menyimpan data yang telah diisikan pada *textedit*. jika data berhasil disimpan maka akan muncul diDBGrid. Gambar 4.20 merupakan tampilan dari form tersebut:



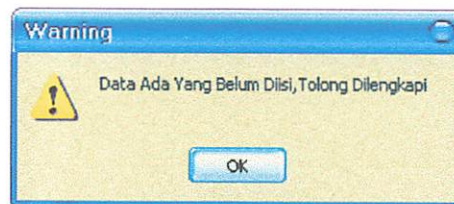
Gambar 4.20 Data Member Telah Tersimpan

Selain itu pada form ini dapat meng-update data Member. Cara yang dilakukan dengan meng-inputkan NIP di *textedit* NIP. Kemudian dengan menekan *enter* maka data karyawan akan ditampilkan di *textedit*. Sehingga dapat merubah data yang diinginkan. Pada kondisi ini *button save* akan berubah menjadi *button update* dengan menekan *button* tersebut maka data akan secara otomatis akan berubah. Gambar 4.21 merupakan tampilan dari form tersebut:



Gambar 4.21 Data Member Telah di *update*

Jika terdapat data yang belum diisi, maka akan muncul pesan peringatan seperti pada gambar 4.22:



Gambar 4.22 Pesan Data Tidak Lengkap

4.2.10 Pengujian Form Aktifasi Member

Pengujian yang dilakukan pada form Aktifasi Member meliputi *textedit* dan *button* yang tersedia, apakah sudah berjalan sesuai dengan prosedur atau belum berfungsi dengan optimal. Gambar 4.23 merupakan tampilan dari form tersebut:



Gambar 4.23 Entri Data Aktifasi Member

Pada *button save* mempunyai fungsi untuk menyimpan data yang telah diisikan pada *textedit*. Dari hasil pengujian aplikasi sudah berfungsi dengan baik, dapat dilihat pada gambar 4.24:

	NO AKTIF	TGL AKTIF	ID MEMBER	NO TARIF MEMBE
▶	A-100413001	13/04/2010	M-100311001	T-100315
	A-100413002	13/04/2010	M-100413001	T-100315
	A-100609001	09/06/2010	M-100609001	T-100315
*				

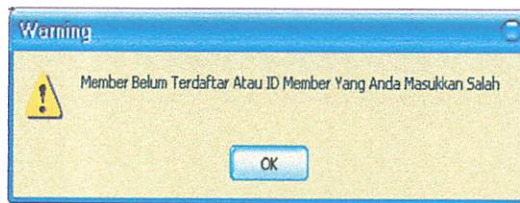
Gambar 4.24 Entri Data Aktif Member

Untuk memperbarui masa aktif dari member, sama dengan cara menyimpan tetapi pada saat *textedit* ID Member di tekan *enter* maka *button save* berubah menjadi *button update*. Dengan menekan *button update* maka masa aktif dari member akan berubah. Dari hasil pengujian aplikasi sudah berfungsi dengan baik, dapat dilihat dari pada gambar 4.25:

	NO AKTIF	TGL AKTIF	ID MEMBER	NO TARIF MEMBE
▶	A-100413001	13/04/2010	M-100311001	T-100315
	A-100413002	13/04/2010	M-100413001	T-100315
	A-100609001	09/06/2010	M-100609001	T-100315
	A-100625001	25/06/2010	M-100311001	T-100624
*				

Gambar 4.25 Data Aktif Member Setelah di Perpanjang

Apabila pada saat salah memasukkan ID member, maka akan muncul pesan peringatan seperti gambar 4.26:

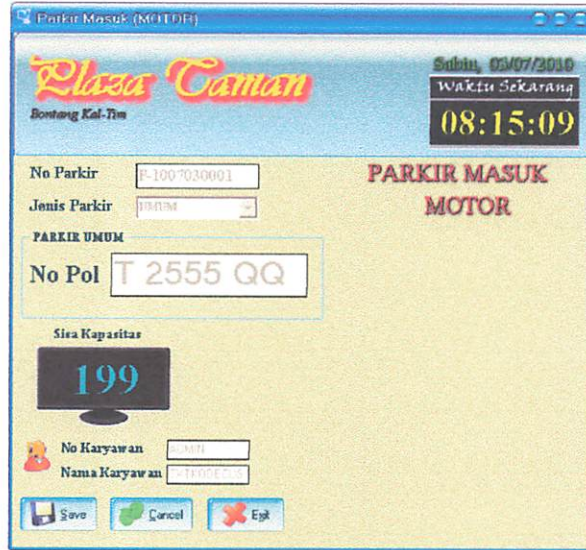


Gambar 4.26 Pesan Kesalahan *Input* ID Member

4.2.11 Pengujian Form Parkir Masuk Motor

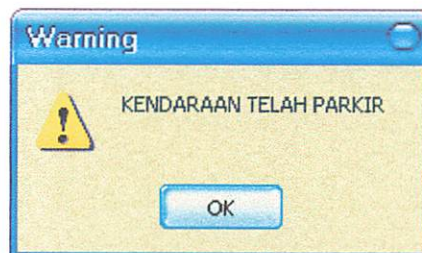
Pengujian yang dilakukan pada form parkir masuk motor meliputi *textedit*, *combobox* dan *button* yang tersedia, apakah sudah berjalan sesuai dengan prosedur atau belum berfungsi dengan optimal. Di dalam form ini terdapat 2 macam, yaitu parkir umum dan parkir member.

Untuk pengujian parkir masuk umum, dengan memilih item umum pada *combobox* jenis parkir. Kemudian menginputkan nomor dari kendaraan yang akan masuk, setelah itu dengan menekan *button save* maka data kendaraan yang akan parkir telah tersimpan. Gambar 4.27 merupakan tampilan dari form tersebut:



Gambar 4.27 Entri Data Parkir Masuk Umum Motor

Jika ternyata kendaraan telah berada di area parkir dan ada nomor kendaraan yang sama akan masuk, maka muncul pesan peringatan seperti gambar 4.28:



Gambar 4.28 Pesan Peringatan Kendaraan

Dari hasil pengujian aplikasi sudah berfungsi dengan baik. Data tersimpan pada table parkir, dapat dilihat pada gambar 4.29:

NO PARKIR	TGL PARKIR	JAM MASUK	TGL KELUAR	JAM KELUAR	OPERATOR MASUK	OPERATOR KELUAR
P-1007020006	02/07/2010	16:36:30	03/07/2010	06:44:37	ADMIN	ADMIN
P-1007020007	02/07/2010	16:37:16	03/07/2010	06:42:28	ADMIN	ADMIN
P-1007020008	02/07/2010	16:43:39	02/07/2010		ADMIN	
P-1007030001	03/07/2010	08:15:13	03/07/2010		ADMIN	

Gambar 4.29 Data Parkir Masuk Umum Motor di Tabel T_Parkir

Data tersimpan pada table parkir umum, dapat dilihat pada gambar 4.30:

NO PARKIR UMUM	NO PARKIR	NO POL	JENIS KENDARAAN	NO TARIF	TOTAL PARKIR
PU-1007020003	P-1007020003	KT 4000 QQ	1	T-100624	1000
PU-1007020004	P-1007020004	KT 78 PP	0	T-100624	2000
PU-1007020005	P-1007020005	KT 78 PP	0	T-100624	2000
PU-1007020006	P-1007020006	KT 6543 K	0		0
PU-1007030001	P-1007030001	KT 2555 QQ	1		0

Gambar 4.30 Data Parkir Masuk Umum Motor di Tabel T_Parkir_Umum

Untuk pengujian parkir masuk member motor, dengan memilih item member pada *combobox* jenis parkir . Kemudian meng-*inputkan* ID Member dari pengguna parkir yang akan masuk, setelah itu dengan menekan *enter* maka data dari member akan ditampilkan apakah sesuai dengan kendaraannya dan masa aktif member akan diketahui. Setelah itu dengan menekan *button save* maka data kendaran yang akan parkir telah tersimpan. Gambar 4.31 merupakan tampilan dari form tersebut:

Gambar 4.31 Entri Data Parkir Masuk Member Motor

Dari hasil pengujian aplikasi sudah berfungsi dengan baik. Data tersimpan pada table parkir, dapat dilihat pada gambar 4.32:

NO PARKIR	TGL PARKIR	JAM MASUK	TGL KELUAR	JAM KELUAR	OPERATOR MASUK	OPERATOR KELUAR
P-1007020007	02/07/2010	16:37:16	03/07/2010	06:42:28	ADMIN	ADMIN
P-1007020006	02/07/2010	16:43:39	02/07/2010		ADMIN	
P-1007030001	03/07/2010	08:15:13	03/07/2010		ADMIN	
P-1007030002	03/07/2010	08:28:37	03/07/2010		ADMIN	

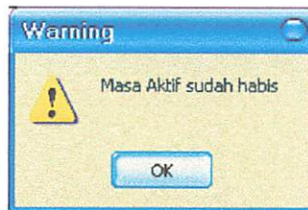
Gambar 4.32 Data Parkir Masuk Member Motor di Tabel T_Parkir

Data tersimpan pada table parkir Member, dapat dilihat pada gambar 4.33:

NO PARKIR MEMB	NO PARKIR	ID MEMBER
PM-1007020001	P-1007020006	M-100624002
PM-1007020002	P-1007020007	M-100311001
PM-1007030001	P-1007030002	M-100624002
*		

Gambar 4.33 Data Parkir Masuk Member Motor di Tabel T_Parkir_Member

Jika pada saat member akan masuk saat masa aktif dari member telah habis maka akan muncul pesan seperti pada gambar 4.34 :

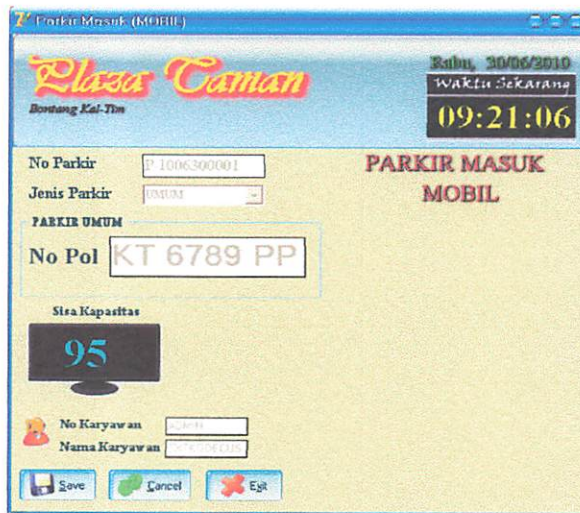


Gambar 4.34 Pesan Peringatan Masa Aktif Member Motor

4.2.12 Pengujian Form Parkir Masuk Mobil

Pengujian yang dilakukan pada form Parkir Masuk mobil meliputi *textedit*, *combobox* dan *button* yang tersedia, apakah sudah berjalan sesuai dengan prosedur atau belum berfungsi dengan optimal. Di dalam form ini terdapat 2 macam , yaitu parkir umum dan parkir member.

Untuk pengujian parkir masuk umum, dengan memilih item umum pada *combobox* jenis parkir. Kemudian menginputkan nomor dari kendaraan yang akan masuk, setelah itu dengan menekan *button save* maka data kendaraan yang akan parkir telah tersimpan. Gambar 4.35 merupakan tampilan dari form tersebut:



Gambar 4.35 Entri Data Parkir Masuk Umum Mobil

Jika ternyata kendaraan telah berada di area parkir dan ada nomor kendaraan yang sama akan masuk, maka muncul pesan peringatan seperti gambar pada gambar 4.36 :



Gambar 4.36 Pesan Peringatan Kendaraan

Dari hasil pengujian aplikasi sudah berfungsi dengan baik, dapat dilihat dari gambar dibawah 4.37 yang berarti bahwa data dapat tersimpan:

NO PARKIR	TGL PARKIR	JAM MASUK	TGL KELUAR	JAM KELUAR	OPERATOR MASUK	OPERATOR KELUAR
P-1006250003	25/06/2010	07:09:33	25/06/2010	07:09:48	ADMIN	ADMIN
P-1006250004	25/06/2010	07:10:16	25/06/2010	07:10:28	ADMIN	ADMIN
P-11006300001	30/06/2010	09:21:07			ADMIN	

Gambar 4.37 Data Parkir Masuk Umum Mobil di Tabel T_Parkir

Data tersimpan pada table parkir umum, dapat dilihat pada gambar 4.33:

NO PARKIR ULMUM	NO PARKIR	NO POL	JENIS KENDARAAN	NO TARIF	TOTAL PARKIR
PU-1006250003	P-1006250004	KT 5000 DD	1	T-100624	1000
PU-1006300001	P-1006300001	KT 6789 PP	0		

Gambar 4.38 Data Parkir Masuk Umum Mobil di Tabel T_Parkir_Umum

Untuk pengujian parkir masuk member mobil, dengan memilih item member pada *combobox* jenis parkir . Kemudian meng-inputkan ID Member dari pengguna parkir yang akan masuk, setelah itu dengan menekan *enter* maka data dari member akan ditampilkan apakah sesuai dengan kendaraannya dan masa aktif member akan diketahui. Setelah itu dengan menekan *button save* maka data kendaran yang akan parkir telah tersimpan. Gambar 4.39 merupakan tampilan dari form tersebut:



Gambar 4.39 Entri Data Parkir Masuk Member Mobil

Dari hasil pengujian aplikasi sudah berfungsi dengan baik, dapat dilihat dari gambar 4.40 yang berarti bahwa data dapat tersimpan:

NO PARKIR	TGL PARKIR	JAM MASUK	TGL KELUAR	JAM KELUAR	OPERATOR MASUK	OPERATOR KELUAR
P-1006250004	25/06/2010	07:10:16	25/06/2010	07:10:28	ADMIN	ADMIN
P-1006300001	30/06/2010	09:21:07	30/06/2010	09:22:55	ADMIN	ADMIN
P-1006300002	30/06/2010	10:26:49			ADMIN	

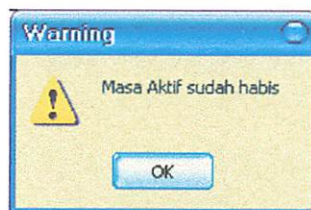
Gambar 4.40 Data Parkir Masuk Member Mobil di Tabel T_Parkir

Data tersimpan pada table parkir Member, dapat dilihat pada gambar 4.41:

NO PARKIR MEMB	NO PARKIR	ID MEMBER
PM-1003240001	P-1003240003	M-100311001
PM-1006090001	P-1006090002	M-100609001
PM-1006140001	P-1006140001	M-100609001
PM-1006180001	P-1006180003	M-100609001
PM-1006300001	P-1006300002	M-100311001

Gambar 4.41 Data Parkir Masuk Member Mobil di Tabel T_Parkir_Member

Jika pada saat member akan masuk saat masa aktif dari member telah habis maka akan muncul pesan sebagai berikut:



Gambar 4.42 Pesan Peringatan Masa Aktif Member Mobil

4.2.13 Pengujian Form Parkir Keluar Motor

Pengujian yang dilakukan pada form Parkir keluar motor meliputi *textedit*, dan *button* yang tersedia, apakah sudah berjalan sesuai dengan prosedur atau belum berfungsi dengan optimal. Di dalam form ini terdapat 2 macam , yaitu parkir umum dan parkir member.

Untuk pengujian parkir keluar umum motor, dengan meng-*inputkan* nomor dari kendaraan yang akan keluar, setelah itu dengan menekan *enter* maka data kendaraan yang parkir akan ditampilkan, seperti: No parkir, tanggal parkir, jam masuk, tanggal keluar, jam keluar, lama durasi parkir serta biaya yang dikenakan kepada pengguna parkir. Gambar 4.43 merupakan tampilan dari form tersebut:

The screenshot shows a software window titled "Parkir Keluar (MOTOR)". The window has a header with the logo "Plaza Taman" and "Bontang Kal-Tim". The date and time are displayed as "Sabtu, 03/07/2010" and "08:36:36". The main form area is titled "PARKIR KELUAR MOTOR". It contains several input fields: "No Pol" with the value "KT 2555 QQ", "No Parkir", "Tgl Parkir", "Jam Keluar", "Jam Masuk", "Lama Parkir" (with a dropdown menu for "hari"), and "Total Parkir" (Rp 1000). There is also a "DATA MEMBER" section with fields for "ID Member", "Nama", "Merk", "Warna", "No Karyawan", and "Nama Karyawan". At the bottom of the form are three buttons: "Save", "Cancel", and "Exit".

Gambar 4.43 Entri Data Parkir Keluar Umum Motor

Dari hasil pengujian, aplikasi sudah berfungsi dengan baik dapat dilihat dari gambar 4.44 yang berarti bahwa data dapat tersimpan:

NO PARKIR	TGL PARKIR	JAM MASUK	TGL KELUAR	JAM KELUAR	OPERATOR MASUK	OPERATOR KELUAR
P-1007020007	02/07/2010	16:37:16	03/07/2010	06:42:28	ADMIN	ADMIN
P-1007020008	02/07/2010	16:43:39	02/07/2010		ADMIN	
P-1007030001	03/07/2010	08:15:13	03/07/2010	08:36:34	ADMIN	ADMIN
P-1007030002	03/07/2010	08:28:37	03/07/2010		ADMIN	

Gambar 4.44 Data Parkir Keluar Motor di Tabel T_Parkir

Data tersimpan pada table parkir umum, dapat dilihat pada gambar 4.45:

NO PARKIR UMUM	NO PARKIR	NO POL	JENIS KENDARAAN	NO TARIF	TOTAL PARKIR
PU-1007020005	P-1007020005	KT 78 PP	0	T-100624	2000
PU-1007020006	P-1007020008	KT 6543 K	0		0
PU-1007030001	P-1007030001	KT 2555 QQ	1	T-100624	1000

Gambar 4.45 Data Parkir Keluar Motor di Tabel T_Parkir_Umum

Untuk pengujian parkir keluar member motor sama dengan parkir keluar umum motor, hanya saja setelah menekan *enter* maka data kendaraan dari member akan ditampilkan apakah sesuai dengan kendaraanya. Serta menampilkan No parkir, tanggal parkir, jam masuk, tanggal keluar, jam keluar, dan lama durasi parkir. .

Gambar 4.46 merupakan tampilan dari form tersebut:

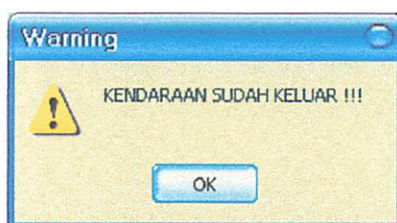
Gambar 4.46 Entri Data Parkir Keluar Member Motor

Dari hasil pengujian aplikasi sudah berfungsi dengan baik,dapat dilihat dari gambar 4.47 yang berarti bahwa data dapat tersimpan:

NO PARKIR	TGL PARKIR	JAM MASUK	TGL KELUAR	JAM KELUAR	OPERATOR_MASUK	OPERATOR_KELUAR
P-1007020007	02/07/2010	16:37:16	03/07/2010	06:42:28	ADMIN	ADMIN
P-1007020008	02/07/2010	16:43:39	02/07/2010		ADMIN	
P-1007030001	03/07/2010	08:15:13	03/07/2010	08:36:34	ADMIN	ADMIN
P-1007030002	03/07/2010	08:28:37	03/07/2010	08:43:07	ADMIN	ADMIN

Gambar 4.47 Data Parkir Keluar Member Motor di Tabel T_Parkir

Jika ternyata kendaraan telah keluar dari area parkir dan ada nomor kendaraan yang sama akan keluar,maka muncul pesan peringatan seperti pada gambar 4.48:



Gambar 4.48 Pesan Peringatan Kendaraan Motor Telah Keluar

4.2.14 Pengujian Form Parkir Keluar Mobil

Pengujian yang dilakukan pada form Parkir keluar mobil meliputi *textedit*, dan *button* yang tersedia,apakah sudah berjalan sesuai dengan prosedur atau belum berfungsi dengan optimal. Di dalam form ini terdapat 2 macam , yaitu parkir umum dan parkir member.

Untuk pengujian parkir keluar umum mobil, dengan meng-*inputkan* nomor dari kendaraan yang akan keluar,setelah itu dengan menekan *enter* maka data kendaraan yang parkir akan ditampilkan, seperti: No parkir, tanggal parkir, jam masuk,

tanggal keluar, jam keluar, lama durasi parkir serta biaya yang dikenakan kepada pengguna parkir. Gambar 4.49 merupakan tampilan dari form tersebut:

Gambar 4.49 Entri Data Parkir Keluar Umum Mobil

Dari hasil pengujian, aplikasi sudah berfungsi dengan baik dapat dilihat dari gambar 4.50 yang berarti bahwa data dapat tersimpan:

NO PARKIR	TGL PARKIR	JAM MASUK	TGL KELUAR	JAM KELUAR	OPERATOR MASUK	OPERATOR KELUAR
P-1006250003	25/06/2010	07:09:33	25/06/2010	07:09:48	ADMIN	ADMIN
P-1006250004	25/06/2010	07:10:16	25/06/2010	07:10:28	ADMIN	ADMIN
P-1006300001	30/06/2010	09:21:07	30/06/2010	09:22:55	ADMIN	ADMIN

Gambar 4.50 Data Parkir Keluar Mobil di Tabel T_Parkir

Data tersimpan pada tabel parkir Member, dapat dilihat pada gambar 4.51:

NO PARKIR UMUM	NO PARKIR	NO POL	JENIS KENDARAAN	NO TARIF	TOTAL PARKIR
PL-1006253003	P-1006250004	KT 5000 DD	1	T-100624	1000
PL-1006303001	P-1006300001	KT 6789 PP	0	T-100624	2000

Gambar 4.51 Data Parkir Keluar Mobil di Tabel T_Parkir_Umum

Untuk pengujian parkir keluar member mobil sama dengan parkir keluar umum mobil, hanya saja setelah menekan *enter* maka data kendaran dari member akan ditampilkan apakah sesuai dengan kendaraanya. Serta menampilkan No parkir,

tanggal parkir, jam masuk, tanggal keluar, jam keluar, dan lama durasi parkir.

Gambar 4.52 merupakan tampilan dari form tersebut:

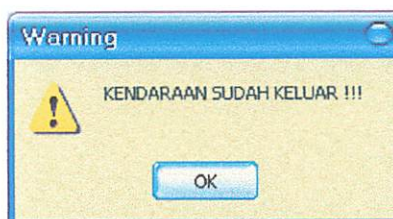
Gambar 4.52 Entri Data Parkir Keluar Member Mobil

Dari hasil pengujian aplikasi sudah berfungsi dengan baik,dapat dilihat dari gambar 4.53 yang berarti bahwa data dapat tersimpan:

NO PARKIR	TGL PARKIR	JAM MASUK	TGL KELUAR	JAM KELUAR	OPERATOR MASUK	OPERATOR KELUAR
P-1006250004	25/06/2010	07:10:16	25/06/2010	07:10:28	ADMIN	ADMIN
P-1006300001	30/06/2010	09:21:07	30/06/2010	09:22:55	ADMIN	ADMIN
P-1006300002	30/06/2010	10:26:49	30/06/2010	10:31:53	ADMIN	ADMIN

Gambar 4.53 Data Parkir Keluar Member Mobil di Tabel T_Parkir

Jika ternyata kendaraan telah keluar dari area parkir dan ada nomor kendaraan yang sama akan keluar,maka muncul pesan peringatan seperti pada gambar 4.54:



Gambar 4.54 Pesan Peringatan Kendaraan Mobil Telah Keluar

4.2.15 Pengujian Form Pengeluaran Parkir

Pengujian yang dilakukan pada form Pengeluaran Parkir meliputi *textedit*, dan *button* yang tersedia, apakah sudah berjalan sesuai dengan prosedur atau belum berfungsi dengan optimal. Gambar 4.55 merupakan tampilan dari form tersebut:

The screenshot shows a window titled "Setting Pengeluaran" with a header "DATA PENGELUARAN". It contains several input fields and a table. At the top, there are fields for "No Pengeluaran" (value: 100000) and "Tanggal" (value: 15/05/2010). Below is a table with three columns: "No", "Keterangan", and "Biaya (Rp)". The table has one row with "1", "TERTAS", and "100000". Below the table, there are fields for "Keterangan" (value: TINTA), "Biaya Rp", and "Total" (value: 100000). There are also buttons for "Save", "Cancel", "Exit", and "Delete".

No	Keterangan	Biaya (Rp)
1	TERTAS	100000

Gambar 4.55 Entri Data Pengeluaran Parkir

Pada *button save* mempunyai fungsi untuk menyimpan data yang telah diisikan pada *textedit*. Jika data berhasil disimpan maka akan muncul diDBGrid. Gambar 4.56 merupakan tampilan dari form tersebut:

The screenshot shows the same window as in Gambar 4.55, but now with two entries in the table. The second entry has "2" in the "No" column, "TINTA" in the "Keterangan" column, and "100000" in the "Biaya (Rp)" column. The "Total" field now shows "200000".

No	Keterangan	Biaya (Rp)
1	TERTAS	100000
2	TINTA	100000

Gambar 4.56 Data Pengeluaran Parkir Telah Tersimpan

4.2.16 Pengujian Form Laporan Karyawan

Pengujian yang dilakukan pada form laporan karyawan meliputi *ceck box*, *combo box*, *textedit*, dan *button* yang tersedia, apakah sudah berjalan sesuai dengan prosedur atau belum berfungsi dengan optimal. Gambar 4.57 merupakan tampilan dari form tersebut:



Gambar 4.57 Filter Laporan Karyawan

Terdapat dua Kriteria yang akan dipilih. Kriteria pertama berisi tentang item-item yang dimiliki karyawan, yaitu: NIP, nama, alamat, no. telephone, jenis kelamin, dan jabatan. Kemudian pada *textedit* diisikan sesuai dengan item yang dipilih. Kriteria kedua berisi tentang status karyawan, masih aktif bekerja atau sudah tidak bekerja lagi.

Setelah dipilih sesuai dengan kriteria maka akan muncul laporan yang diinginkan. Gambar 4.58 merupakan tampilan dari laporan tersebut:

NIP	NAMA	ALAMAT	NO_TELP	JENIS_KFLAMIN	JABATAN	STATUS
123456789	HARDI	JL KAHURUHAN 456	12315654	FRIA	OPERATOR	AKTIF
1234567890	SANTI	JL MERDEKA 346	710769	WANITA	ADMINISTRASI	AKTIF
222222222	SASA	JL PANGASILA	088677	WANITA	ADMINISTRASI	AKTIF
333333333	RICKI	JL SURABAYA	443355	FRIA	OPERATOR	AKTIF
444444444	ERIC	KEBAOUSAN 56784	777	FRIA	OPERATOR	AKTIF
454545454	DEDE	JUGRIK	55665566	FRIA	OPERATOR	AKTIF
555555555	ARY ARTANTO	JL MALANG 45	341341	FRIA	MANAGER	AKTIF
565656565	AAR	DAFAD	435345	FRIA	OPERATOR	AKTIF
666666666	RANI	JL BANTENO	45678	WANITA	OPERATOR	AKTIF
777777777	HERU PRATAMA	JL PISANO 5	889977	FRIA	OPERATOR	AKTIF

Gambar 4.58 Hasil Laporan Karyawan

4.2.17 Pengujian Form Laporan Member

Pengujian yang dilakukan pada form laporan member meliputi *check box*, *combo box*, *textedit*, dan *button* yang tersedia, apakah sudah berjalan sesuai dengan prosedur atau belum berfungsi dengan optimal. Gambar 4.59 merupakan tampilan dari form tersebut:

Gambar 4.59 Filter Laporan Member

Terdapat dua kriteria yang akan dipilih. Kriteria pertama berisi tentang item-item yang dimiliki member, yaitu: ID member, nama, alamat, no. telephone, nomor polisi, merk dan warna. Kemudian pada *textedit* diisikan sesuai dengan item yang dipilih. Kriteria kedua berisi tentang jenis kendaraan, yaitu motor dan mobil.

Setelah dipilih sesuai dengan kriteria maka akan muncul laporan yang diinginkan. Gambar 4.60 merupakan tampilan dari laporan tersebut:

Pusat Cerman
Jl. MULAWARAN NO 01
SUNTAHO UTARA
SUNTAHO

LAPORAN MEMBER
Tanggal Print: 16/08/2019

ID MEMBER	NAMA	ALAMAT	NO TELP	NO. POL	JENIS KENDARAAN	MERKE	WARNA
M-100311001	AYUB	JL MERDEKA 48	891243	N-1956	MOBIL	HONDA RZR	BLACK
M-100413001	ARI	JL KALIMANTAN 56	21341445	N-7777	MOBIL	HONDA CIVIC	KUNING
M-100609001	LUKAS	BERBAS	09876	KT 7890 KK	MOBIL	PEROZA	HITAM
M-100624001	AGUNG	JL PANCASILA	08123012345	KT 5555 DD	MOTOR	HONDA CBR	HITAM
M-100624002	STEPAN	JL SRIJOUNTING NO 35	0548-445522	KT 9999 QQ	MOTOR	SUPRA	HITAM
M-100702001	YUDI	JL BELAWAN	7788661	KT 7777 AA	MOTOR	KAWASAKI	HITAM
M-100816001	ILHAM	JL JAKSA	123456	N 123 M	MOTOR	MIO	PUTIH

Gambar 4.60 Hasil Laporan Member

4.2.18 Pengujian Form Laporan Aktif Member

Pengujian yang dilakukan pada form laporan aktif member meliputi *checkbox*, *combo box*, *textedit*, dan *button* yang tersedia, apakah sudah berjalan sesuai dengan prosedur atau belum berfungsi dengan optimal. Gambar 4.61 merupakan tampilan dari form tersebut:



Gambar 4.61 Filter Laporan Aktif Member

Terdapat tiga kriteria yang akan dipilih. Kriteria pertama berisi tentang item-item yang dimiliki member, yaitu: NO aktif, ID member, nama dan nomor polisi. Kemudian pada *textedit* diisikan sesuai dengan item yang dipilih. Kriteria kedua berisi tentang jenis kendaraan, yaitu motor dan mobil. Kriteria ketiga berisi tentang periode pada saat pengaktifan masa berlaku member.

Setelah dipilih sesuai dengan kriteria maka akan muncul laporan yang diinginkan. Gambar 4.62 merupakan tampilan dari laporan tersebut:

1 of 1 100% Total: 6 100% 6 of 6

Pusat Aman
Optimaly 3C Calculator Center

JL. MULAWANAN NO 01
 BONTANG UTARA
 BONTANG

LAPORAN AKTIF MEMBER

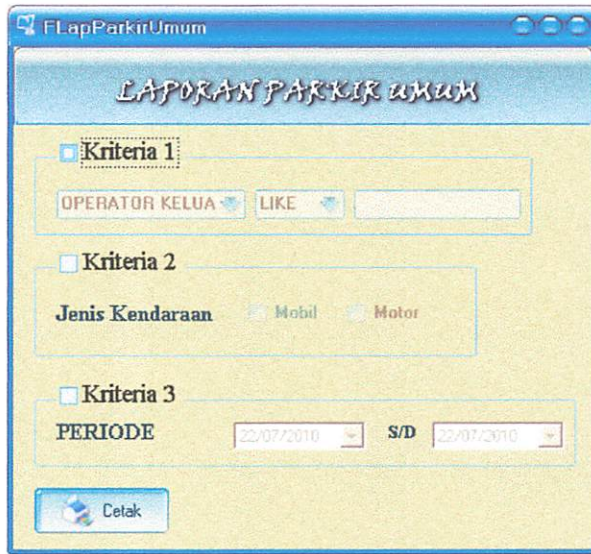
Tanggal Print: 16/08/2010

NO AKTIF	TGL AKTIF	ID MEMBER	NAMA	NOPOL	JENIS KENDARAAN	BIAYA
A-100413001	1304/2010	M-100311001	AYUB	N-1956	MOBIL	Rp 200.000
A-100413002	1304/2010	M-100413001	ARI	N-1777	MOBIL	Rp 200.000
A-100609001	09/06/2010	M-100609001	LUKAS	KT 7890 EK	MOBIL	Rp 200.000
A-100625001	25/06/2010	M-100311001	AYUB	N-1956	MOBIL	Rp 30.000
A-100630001	30/06/2010	M-100624002	STEVEN	KT 9999 QQ	MOTOR	Rp 20.000
A-100815001	15/08/2010	M-100311001	AYUB	N-1956	MOBIL	Rp 30.000
TOTAL						Rp 728.000

Gambar 4.62 Laporan Aktif Member

4.2.19 Pengujian Form Laporan Parkir Umum

Pengujian yang dilakukan pada form laporan parkir umum meliputi *checkbox*, *combo box*, *textedit*, dan *button* yang tersedia, apakah sudah berjalan sesuai dengan prosedur atau belum berfungsi dengan optimal. Gambar 4.63 merupakan tampilan dari form tersebut:



Gambar 4.63 Filter Laporan Parkir Umum

Terdapat tiga kriteria yang akan dipilih. Kriteria pertama berisi tentang item-item yang berhubungan dengan aktifitas parkir , yaitu: Operator keluar dan nomor polisi. Kemudian pada *textedit* diisikan sesuai dengan item yang dipilih. Kriteria kedua berisi tentang jenis kendaraan,yaitu motor dan mobil. Kriteria ketiga berisi tentang periode yang ingin ditampilkan menjadi laporan.

Setelah dipilih sesuai dengan kriteria maka akan muncul laporan yang diinginkan. Gambar 4.64 merupakan tampilan dari laporan tersebut:

NO PARKIR	TGL MASUK	JAM MASUK	TGL KELUAR	JAM KELUAR	NO POL	JENIS KENDARAAN	OPERATOR MASUK	OPERATOR KELUAR	TOTAL PARKIR
FU-1006150006	18/08/2010	15:15:47	14/08/2010	16:22:04	KK 8876	MOBIL	ADMIN	ADMIN	Rp 6.000
FU-1007020006	02/07/2010	16:43:39	14/08/2010	16:31:47	KT 6543 E	MOBIL	ADMIN	ADMIN	Rp 6.000
FU-1007180001	18/07/2010	15:23:62	14/08/2010	16:31:31	KT 6563 PP	MOBIL	ADMIN	ADMIN	Rp 6.000
FU-1007250001	25/07/2010	05:34:63	14/08/2010	16:31:20	KT 442 Q	MOBIL	1224546-864	ADMIN	Rp 6.000
FU-1007260001	26/07/2010	16:56:32	14/08/2010	16:31:10	KT 7777 FF	MOBIL	ADMIN	ADMIN	Rp 6.000
FU-1007260002	26/07/2010	16:57:00	14/08/2010	16:30:58	KT 888 MK	MOBIL	ADMIN	ADMIN	Rp 6.000
FU-1008350003	05/08/2010	17:16:50	14/08/2010	16:30:40	KT 7 C	MOBIL	ADMIN	ADMIN	Rp 6.000
FU-1008140001	14/08/2010	00:19:69	14/08/2010	16:30:27	KT 8 Y	MOBIL	ADMIN	ADMIN	Rp 6.000
FU-1008140002	14/08/2010	00:24:17	14/08/2010	16:30:16	KT 81 J	MOBIL	ADMIN	ADMIN	Rp 6.000
FU-1008140003	14/08/2010	16:28:27	14/08/2010	16:30:07	KT 5 TT	MOBIL	ADMIN	ADMIN	Rp 2.000
FU-1008140004	14/08/2010	16:34:37	14/08/2010		KT 5 R	MOBIL	ADMIN	ADMIN	Rp -
FU-1008150001	15/08/2010	09:12:42	15/08/2010		KT 7 E	MOBIL	ADMIN	ADMIN	Rp -
FU-1008150002	15/08/2010	09:15:20	15/08/2010		KT 56 G	MOBIL	ADMIN	ADMIN	Rp -
FU-1008150003	15/08/2010	09:15:34	15/08/2010		KT 85 N	MOBIL	ADMIN	ADMIN	Rp -
FU-1008150004	15/08/2010	09:15:51	15/08/2010		KT 9 J	MOBIL	ADMIN	ADMIN	Rp -
FU-1008150005	15/08/2010	09:15:59	15/08/2010		KT 4 O	MOBIL	ADMIN	ADMIN	Rp -
FU-1008150006	15/08/2010	09:16:10	15/08/2010		KT 34 LL	MOBIL	ADMIN	ADMIN	Rp -
FU-1008150007	15/08/2010	09:16:21	15/08/2010		KT 12 P	MOBIL	ADMIN	ADMIN	Rp -
Total									Rp 56.000

Gambar 4.64 Laporan Parkir Umum

4.2.20 Pengujian Form Laporan Parkir Member

Pengujian yang dilakukan pada form laporan parkir member meliputi *checkbox*, *combo box*, *textedit*, dan *button* yang tersedia, apakah sudah berjalan sesuai dengan prosedur atau belum berfungsi dengan optimal. Gambar 4.65 merupakan tampilan dari form tersebut:



Gambar 4.65 Filter Laporan Parkir Member

Terdapat tiga kriteria yang akan dipilih. Kriteria pertama berisi tentang item-item yang berhubungan dengan aktifitas parkir , yaitu: Operator keluar dan ID member. Kemudian pada *textedit* diisikan sesuai dengan item yang dipilih. Kriteria kedua berisi tentang jenis kendaraan,yaitu: motor dan mobil. Kriteria ketiga berisi tentang periode yang ingin ditampilkan menjadi laporan.

Setelah dipilih sesuai dengan kriteria maka akan muncul laporan yang diinginkan. Gambar 4.66 merupakan tampilan dari laporan tersebut:

Plaza Taman
Jl. MULAWARAH NO 01
BOHARU UTARA
BOHARU

LAPORAN PARKIR MEMBER
Tanggal Print: 16/08/2010

NO PARKIR	TGL MASUK	JAM MASUK	TGL KELUAR	JAM KELUAR	ID MEMBER	OPERATOR_MASUK	OPERATOR_KELUAR
PM-100630001	30/06/2010	10:26:49	30/06/2010	00:10:31:53	M-100311001	ADMIN	ADMIN
PM-100630002	30/06/2010	11:44:58	01/07/2010	00:06:09:35	M-100624002	ADMIN	ADMIN
PM-100702001	02/07/2010	16:36:30	03/07/2010	00:06:44:37	M-100624002	ADMIN	ADMIN
PM-100702002	02/07/2010	16:37:16	03/07/2010	00:06:42:28	M-100311001	ADMIN	ADMIN
PM-100703001	03/07/2010	08:28:37	03/07/2010	00:08:43:07	M-100624002	ADMIN	ADMIN

Gambar 4.66 Laporan Parkir Member

4.2.21 Pengujian Form Laporan Pendapatan

Pengujian yang dilakukan pada form laporan parkir member meliputi *check box*, *combo box*, dan *button* yang tersedia, apakah sudah berjalan sesuai dengan prosedur atau belum berfungsi dengan optimal. Gambar 4.67 merupakan tampilan dari form tersebut:

FLapPendapatan

LAPORAN PENDAPATAN

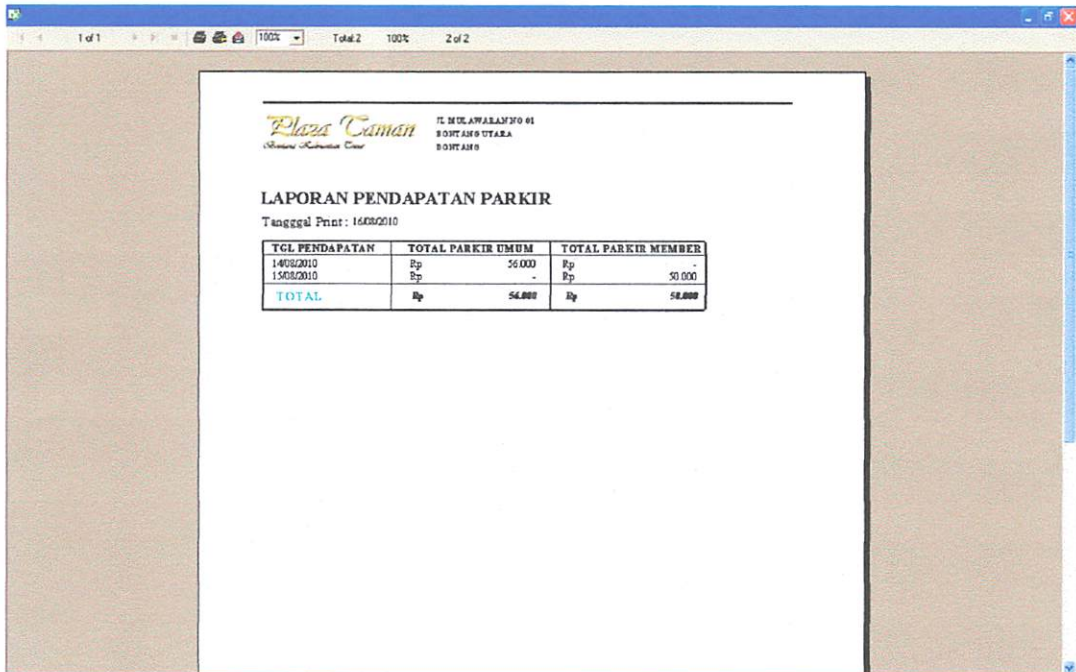
Kriteria

PERIODE S/D

Gambar 4.67 Filter Laporan Pendapatan

Terdapat kriteria yang akan dipilih yaitu periode dari pengeluaran yang akan ditampilkan menjadi laporan.

Setelah dipilih sesuai dengan kriteria maka akan muncul laporan yang diinginkan. Gambar 4.68 merupakan tampilan dari laporan tersebut:



The screenshot shows a web browser window displaying a report from Plaza Ciman. The report is titled "LAPORAN PENDAPATAN PARKIR" and is dated "Tanggal Print: 16/08/2010". It contains a table with the following data:

TGL PENDAPATAN	TOTAL PARKIR UMUM	TOTAL PARKIR MEMBER
14/08/2010	Rp 50.000	Rp -
15/08/2010	Rp -	Rp 50.000
TOTAL	Rp 54.000	Rp 50.000

Gambar 4.68 Laporan Parkir Member

4.2.22 Pengujian Form Laporan Pengeluaran

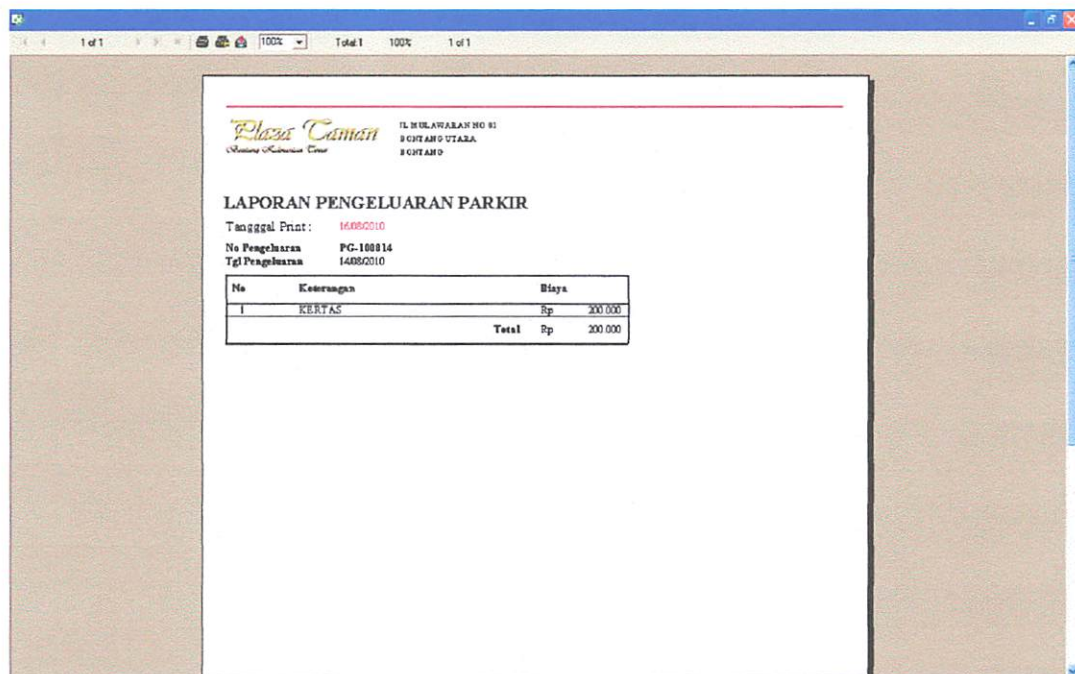
Pengujian yang dilakukan pada form laporan pengeluaran meliputi *ceck box*, *combo box*, dan *button* yang tersedia, apakah sudah berjalan sesuai dengan prosedur atau belum berfungsi dengan optimal. Gambar 4.69 merupakan t tampilan dari form tersebut:



Gambar 4.69 Filter Laporan Pengeluaran

Terdapat kriteria yang akan dipilih yaitu periode dari pengeluaran yang akan ditampilkan menjadi laporan.

Setelah dipilih sesuai dengan kriteria maka akan muncul laporan yang diinginkan. Gambar 4.70 merupakan tampilan dari laporan tersebut:



Gambar 4.70 Laporan Pengeluaran

4.2.23 Pengujian Form Laporan Laba/Rugi

Pengujian yang dilakukan pada form laporan laba/rugi meliputi *ceck box*, *combo box*, dan *button* yang tersedia, apakah sudah berjalan sesuai dengan prosedur atau belum berfungsi dengan optimal. Gambar 4.71 merupakan tampilan dari form tersebut:



	BULAN	TAHUN	S/D	BULAN	TAHUN
PERIODE	06	2010		07	2010

Gambar 4.71 Filter Laporan Laba/Rugi

Terdapat kriteria yang akan dipilih yaitu periode dari laba/rugi yang akan ditampilkan menjadi laporan.

Setelah dipilih sesuai dengan kriteria maka akan muncul laporan yang diinginkan. Gambar 4.72 merupakan tampilan dari laporan tersebut:

Plaza Taman
 Taman & Kolaborasi Ruang
 JL. MULAWARAN NO 01
 BOMBAH UTARA
 BOMBANG

LAPORAN LABA/RUGI
 Tanggal Print: 16/08/2010

NO LABARUGI	PENDAPATAN (Rp)	PENGELUARAN (Rp)	SALDO (Rp)
LB-1007	Rp 72.000	Rp -	Rp 72.000
LB-1008	Rp 112.000	Rp 200.000	Rp 88.000

Gambar 4.72 Laporan Laba/Rugi

4.2.24 Pengujian Karcis Masuk

Hasil pengujian yang telah dilakukan berhasil menampilkan data kendaraan yang akan parkir. Data yang ditampilkan pada karcis masuk ini adalah no parkir, tanggal parkir, jam masuk, nomor kendaraan, jenis kendaraan yang parkir dan operator masuk. Gambar 4.73 merupakan tampilan dari karcis masuk:

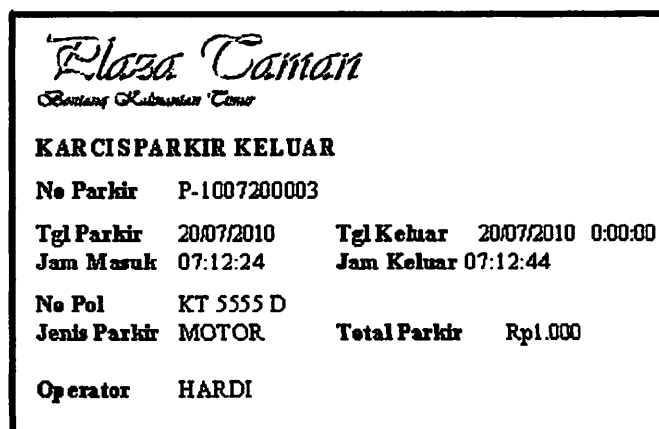


Gambar 4.73 Karcis Masuk

4.2.25 Pengujian Karcis Keluar

Hasil pengujian yang telah dilakukan berhasil menampilkan data kendaraan yang akan keluar dari area parkir. Data yang ditampilkan pada karcis keluar ini adalah no parkir, tanggal parkir, jam masuk, tanggal keluar, jam keluar, nomor kendaraan, jenis kendaraan yang parkir, total biaya parkir dan operator keluar.

Gambar 4.74 merupakan tampilan dari karcis masuk:



Gambar 4.74 Karcis Keluar

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan hasil pengujian sistem yang dilakukan, dapat diambil kesimpulan beberapa hal antara lain :

1. Sistem ini mampu memberi informasi mengenai jumlah tempat parkir yang masih tersedia dalam gedung.
2. Sistem ini mampu menghasilkan laporan parkir (per tanggal dan per bulan), laporan penerimaan dari parkir (per tanggal dan per bulan) dalam waktu yang relatif singkat sesuai dengan kebutuhan.
3. Proses pengecekan data maupun pencarian data bisa dilakukan dengan cepat dan mudah.

5.2. Saran

Mengingat berbagai keterbatasan yang dialami penulis terutama masalah pemikiran dan waktu, maka penulis menyarankan untuk pengembangan Sistem Informasi Manajemen Parkir Berbasis *Client Server* dimasa mendatang diantaranya sebagai berikut .:

1. Dapat mengontrol pintu masuk maupun keluar secara otomatis (*Bargate*).
2. Untuk pengambilan waktu diambil dari server sehingga kesesuaian waktu antara klien bisa berjalan sama.

DAFTAR PUSTAKA

1. Komputer Wahana, "*Membuat Kreatif dan Profesional dengan Delphi*", Elex Media Komputindo, Jakarta 2005.
2. Malik Jamaludin Jaja, "*Kumpulan Latihan Program Delphi*", Andi Offset, 2006.
3. Zakaria, Teddy markus dan Agus Priyono, "*Perancangan Antarmuka Untuk Interaksi Manusia Dan Komputer*". Bandung: Penerbit INFORMATIKA.
4. Indriyawan Eko, "*membangun Sistem andal Delphi*", Andi, 2008.
5. Madcoms., "*Pemrograman Borland Delphi 7*", Andi, Yogyakarta.
6. Kosasih Asep, "*Algoritma & Pemograman Dengan Bahasa Delphi 5.0*", YRAMA WIDYA, 2006.
7. Abdul Kadir, 2001, "*Dasar-dasar Pemograman Delphi, Jilid 6*", Andi Offset: Yogyakarta
8. mti.ugm.ac.id/~panji/dinus/rpl/DATA%20FLOW%20DIAGRAM%201.doc.
9. <http://www.delphi-id.org>.
10. <http://www.diskusiweb.com>.
11. [http://id.wikipedia.org/wiki/micro soft/SQL Server](http://id.wikipedia.org/wiki/micro_soft/SQL_Server)
12. <http://id.wikipedia.org/wiki/Klien-server>

LAMPIRAN



**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

NAMA : ARY ARTANTO
NIM : 05.12.600
JURUSAN : TEKNIK ELEKTRO S-1
KONSENTRASI : TEKNIK KOMPUTER DAN INFORMATIKA
JUDUL SKRIPSI : **PENGEMBANGAN APLIKASI SISTEM INFORMASI
MANAGEMENT PARKIR MODERN BERBASIS CLIENT
SERVER**

Dipertahankan dihadapan penguji skripsi jenjang program strata satu (S-1) pada :

Hari : Rabu
Tanggal : 18 Agustus 2010
Dengan nilai : 81,45 (A) *34*


PANITIA UJIAN SKRIPSI

**Mengetahui,
Ketua Majelis Penguji**



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

ANGGOTA PENGUJI

Dosen Penguji I


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

Dosen Penguji II


Sotyhady, ST
NIP.Y.1039700309



FORMULIR PERBAIKAN SKRIPSI

Dalam pelaksanaan Ujian Skripsi Jenjang Strata Satu (S-1) Jurusan Teknik Elektro Konsentrasi Teknik Komputer & Informatika, maka perlu adanya perbaikan skripsi untuk mahasiswa :

Nama : ARY ARTANTO
NIM : 05 12 600
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Komputer & Informatika
Masa Bimbingan : 30 Desember 2010 s/d 30 Juni 2010
Judul Skripsi : Pengembangan Aplikasi Sistem Informasi *Management Parkir Modern* Berbasis *Client Server*.

Penguji/Tanggal	Uraian	Paraf
Penguji I 18 Agustus 2010	Tambahkan kalimat pengantar pada gambar	

Mengetahui

PENGUJI I

(M. Ibrahim Ashari, ST.MT)
NIP.Y. 1030100358

Dosen Pembimbing

(I Komang Somawirata, ST, MT)
NIP.Y. 1030100361



FORMULIR BIMBINGAN SKRIPSI

Nama : Ary Artanto
NIM : 0512600
Masa Bimbingan : 30 Desember 2009 s/d 30 Juni 2010
Judul Skripsi : Pengembangan Aplikasi Sistem Informasi *Management* Parkir
Modern Berbasis Client Server

NO	Tanggal	Uraian	Paraf Pembimbing
1	19/01/2010	Konsultasi Skripsi.	
2	11/06/2010	Revisi Bab I,II dan III.	
3	15/06/2010	Acc Bab I,II dan III.	
4	29/06/2010	Revisi Bab IV dan V	
5	03/07/2010	Acc Bab IV	
6	06/07/2010	Acc Bab V	
7	06/07/2010	Acc Makalah Seminar Hasil	
8	06/07/2010	ACC	
9			
10			

Malang,

Juli 2010

Dosen Pembimbing

I Komang Somawirata, ST. MT.
NIP.Y. 1030100361

LAMPIRAN LISTING PROGRAM

Unit Service

```
unit UService;
interface
uses SysUtils, Udatabase, strutils;
type
  TKaryawan = Class
    NIP      :string;
    NAMA     :string;
    ALAMAT   :string;
    NO_TELP  :string;
    JENIS_KELAMIN:boolean;
    JABATAN  :byte;
    STATUS   :boolean;
  Public
    constructor Create; overload; virtual;
    Function Find_NIP(NIP1:string):boolean;
    Procedure
      Save_Data(NIP1,NAMA1,ALAMAT1,NO_TELP1,JENIS_KELAMIN1,JABATAN1,STAT
      US1:string);
    Procedure
      Update_Data(NIP1,NAMA1,ALAMAT1,NO_TELP1,JENIS_KELAMIN1,JABATAN1,STA
      TUS1:string);
  Private
    Sqlcmd:String;
  end;
  TMember = Class
    ID_MEMBER: string;
    NAMA     : string;
    ALAMAT   : string;
    NO_TELP  : string;
    NO_POL   : string;
    JENIS_KENDARAAN : boolean;
    MERK     : string;
    WARNA    : string;
  Public
    constructor Create; overload; virtual;
    Function Get_ID_Member:string;
    Function Find_ID_Member(ID_Member1:string):boolean;
    Procedure
      Save_Data(ID_MEMBER1,NAMA1,ALAMAT1,NO_TELP1,NO_POL1,JENIS_KENDAR
      AAN1,MERK1,WARNA1: string);
    Procedure
      Update_Data(ID_MEMBER1,NAMA1,ALAMAT1,NO_TELP1,NO_POL1,JENIS_KENDA
      RAAN1,MERK1,WARNA1: string);
  Private
    Sqlcmd:String;
  end;
  TTarif = Class
```



```

NO_TARIF : string;
TGL_SETTING : string;
TARIF_MOBIL : string;
TARIF_MOTOR : string;
Public
    constructor Create; overload; virtual;
    Function Get_NO_TARIF:string;
    Function Get_Last_No_Tarif: string;
    Function Get_Last_Tarif_Mobil:string;
    Function Get_Last_Tarif_Motor:string;
    Function Find_NO_TARIF(NO_TARIF1:string):boolean;
    Procedure
Save_Data(NO_TARIF1,TGL_SETTING1,TARIF_MOBIL1,TARIF_MOTOR1: string);
    Procedure
Update_Data(NO_TARIF1,TGL_SETTING1,TARIF_MOBIL1,TARIF_MOTOR1: string);

Private
    Sqlcmd:String;
end;
TTarifMember = Class
    NO_TARIF_MEMBER : string;
    TGL_SETTING : string;
    TARIF_MOBIL : string;
    TARIF_MOTOR : string;
Public
    constructor Create; overload; virtual;
    Function Get_NO_TARIF_MEMBER:string;
    Function Find_NO_TARIF_MEMBER(NO_TARIF_MEMBER1:string):boolean;
    Procedure
Save_Data(NO_TARIF_MEMBER1,TGL_SETTING1,TARIF_MOBIL1,TARIF_MOTOR1:
string);
    Procedure
Update_Data(NO_TARIF_MEMBER1,TGL_SETTING1,TARIF_MOBIL1,TARIF_MOTO
R1: string);
    Function Get_Last_Tarif_Mobil:string;
    Function Get_Last_Tarif_Motor:string;
    Function Get_Last_No_Tarif_Member:string;
Private
    Sqlcmd:String;
end;
TAktifMember = Class
    NO_AKTIF : string;
    TGL_AKTIF : string;
    ID_MEMBER : string;
    NO_TARIF_MEMBER : String;
Public
    constructor Create; overload; virtual;
    Function Get_NO_AKTIF_MEMBER:string;
    Function Get_Last_Aktif(ID_Member1:string):string;

```

```

Procedure
Save_Data(NO_AKTIF1,TGL_AKTIF1,ID_MEMBER1,NO_TARIF_MEMBER1: string);
//Procedure
Update_Data(NO_TARIF_MEMBER1,TGL_SETTING1,TARIF_MOBIL1,TARIF_MOTOR1: string);
Private
    Sqlcmd:String;
end;
TKapasitas = Class
    NO_KAPASITAS      : string;
    TGL_SETTING       : string;
    KAPASITAS_MOBIL   : integer;
    KAPASITAS_MOTOR   : integer;
Public
    constructor Create; overload; virtual;
    Function Get_NO_KAPASITAS:string;
    Function Find_NO_KAPASITAS(NO_KAPASITAS1:string):boolean;
    Procedure
Save_Data(NO_KAPASITAS1,TGL_SETTING1,KAPASITAS_MOBIL1,KAPASITAS_MOTOR1: string);
    Procedure
Update_Data(NO_KAPASITAS1,TGL_SETTING1,KAPASITAS_MOBIL1,KAPASITAS_MOTOR1: string);
    Function Get_Last_Kapasitas_Mobil:integer;
    Function Get_Last_Kapasitas_Motor:integer;
    Function Get_Last_No_Kapasitas:string;
    Function Get_Sisa_Kapasitas_Mobil:integer;
    Function Get_Sisa_Kapasitas_Motor:integer;
Private
    Sqlcmd:String;
end;
T_Parkir = Class
    NO_PARKIR      : String;
    TGL_PARKIR     : String;
    JAM_MASUK      : String;
    TGL_KELUAR     : String;
    JAM_KELUAR     : String;
    OPERATOR_MASUK : String;
    OPERATOR_KELUAR: String;
Public
    Function Get_No_Parkir: String;
    Function Find_Parkir(NO_PARKIR1:string):Boolean;
    Function Find_Parkir_MOBIL1(NO_POL1:string):Byte;
    Function Find_Parkir_MOTOR1(NO_POL1:string):Byte;
    Function Find_Parkir_MOBIL2(NO_POL1:string):Byte;
    Function Find_Parkir_MOTOR2(NO_POL1:string):Byte;
    Procedure
Save_Parkir_Masuk(NO_PARKIR1,TGL_PARKIR1,JAM_MASUK1,OPERATOR_MASUK1: String);

```

```

    Procedure
Save_Parkir_Keluar(NO_PARKIR1,TGL_KELUAR1,JAM_KELUAR1,OPERATOR_KEL
UAR1: String);
    Private
        Sqlcmd:String;
    end;
    T_Parkir_Umum= Class(T_Parkir)
        NO_PARKIR_UMUM : String;
        NO_PARKIR      : String;
        NO_POL         : String;
        JENIS_KENDARAAN: Boolean;
        NO_TARIF       : String;
        TOTAL_PARKIR   : String;
    Public
        Function Get_No_Parkir_Umum: String;
        constructor Create; overload; virtual;
        Function Find_Parkir_Umum(NO_PARKIR1:string):Boolean;
    Procedure
Save_Data(NO_PARKIR_UMUM1,NO_PARKIR1,NO_POL1,JENIS_KENDARAAN1,NO
_TARIF1,TOTAL_PARKIR1: String);
        Procedure Update_Data(NO_PARKIR1,NO_TARIF1,TOTAL_PARKIR1: String);
    Private
        Sqlcmd:String;
    end;
    T_Parkir_Member= Class(T_Parkir)
        NO_PARKIR_MEMBER :String;
        NO_PARKIR         :String;
        ID_MEMBER         :String;
    Public
        Function Get_No_Parkir_Member: String;
        Function Find_Parkir_Member(NO_PARKIR1:string):Boolean;
        Procedure Save_Data(NO_PARKIR_MEMBER1,NO_PARKIR1,ID_MEMBER1:
String);
        constructor Create; overload; virtual;
    Private
        Sqlcmd:String;
    end;
    T_Pengeluaran = Class
        No_Pengeluaran:string;
        Tgl_Pengeluaran:string;
        Total:string;
    Public
        Function Get_No_Pengeluaran: String;
        Function Find_No_Pengeluaran(No_Pengeluaran1:String):Boolean;
    Procedure
Save_Pengeluaran(NO_PENGELUARAN1,TGL_PENGELUARAN1,TOTAL1:string);
        Procedure Update_Pengeluaran(NO_PENGELUARAN1,TOTAL1:string);
        Procedure Delete_Pengeluaran(NO_PENGELUARAN1:String);
    Private
        Sqlcmd:String;

```

```

end;
T_Pengeluaran_D = Class(T_Pengeluaran)
    No_Pengeluaran_D:string;
    No_Pengeluaran:string;
    Keterangan:string;
    Biaya:string;
Public
    Function Get_No_Pengeluaran_D(NO_PENGELUARAN1:String;NO:integer): String;
    Procedure
Save_Pengeluaran_D(NO_PENGELUARAN_D1,NO_PENGELUARAN1,KETERANGAN
1,BIAYA1:string);
    Procedure Delete_Pengeluaran_D(NO_PENGELUARAN1:string);
end;
T_Pendapatan = Class
    NO_PENDAPATAN : String;
    TGL_PENDAPATAN : String;
    TOTAL_PARKIR_UMUM: String;
    TOTAL_PARKIR_MEMBER: String;
Public
    Function Get_No_Pendapatan: String;
    Function Find_No_Pendapatan(NO_PENDAPATAN1:String): Boolean;
    Procedure
Save_Pendapatan(NO_PENDAPATAN1,TGL_PENDAPATAN1,TOTAL_PENDAPATAN_
UMUM1,TOTAL_PENDAPATAN_MEMBER1:string);
    Procedure
Update_Pendapatan(NO_PENDAPATAN1,TGL_PENDAPATAN1,TOTAL_PENDAPATA
N_UMUM1,TOTAL_PENDAPATAN_MEMBER1:string);
Private
    Sqlcmd:String;
end;
T_LabaRugi = Class
    NO_LABARUGI : string;
    PENDAPATAN : string;
    PENGELUARAN : string;
Public
    Function Get_No_LabaRugi: String;
    Function Find_No_LabarRugi(NO_LABARUGI1:String): Boolean;
    Procedure
Save_LabaRugi(NO_LABARUGI1,PENDAPATAN1,PENGELUARAN1:string);
    Procedure
Update_LabaRugi(NO_LABARUGI1,PENDAPATAN1,PENGELUARAN1:string);
Private
    Sqlcmd:String;
end;
T_UserAcc = Class
    NIP : string;
    Passwd : String;
    Hak_Akses : string;
Public
    Function Find_UserID(NIP: string): Boolean;

```

```

end;
T_Pengeluaran_D = Class(T_Pengeluaran)
    No_Pengeluaran_D:string;
    No_Pengeluaran:string;
    Keterangan:string;
    Biaya:string;
Public
    Function Get_No_Pengeluaran_D(NO_PENGELUARAN1:String;NO:integer): String;
    Procedure
Save_Pengeluaran_D(NO_PENGELUARAN_D1,NO_PENGELUARAN1,KETERANGAN
1,BIAYA1:string);
    Procedure Delete_Pengeluaran_D(NO_PENGELUARAN1:string);
end;
T_Pendapatan = Class
    NO_PENDAPATAN : String;
    TGL_PENDAPATAN : String;
    TOTAL_PARKIR_UMUM: String;
    TOTAL_PARKIR_MEMBER: String;
Public
    Function Get_No_Pendapatan: String;
    Function Find_No_Pendapatan(NO_PENDAPATAN1:String): Boolean;
    Procedure
Save_Pendapatan(NO_PENDAPATAN1,TGL_PENDAPATAN1,TOTAL_PENDAPATAN_
UMUM1,TOTAL_PENDAPATAN_MEMBER1:string);
    Procedure
Update_Pendapatan(NO_PENDAPATAN1,TGL_PENDAPATAN1,TOTAL_PENDAPATA
N_UMUM1,TOTAL_PENDAPATAN_MEMBER1:string);
Private
    Sqlcmd:String;
end;
T_LabaRugi = Class
    NO_LABARUGI : string;
    PENDAPATAN : string;
    PENGELUARAN : string;
Public
    Function Get_No_LabaRugi: String;
    Function Find_No_LabarRugi(NO_LABARUGI1:String): Boolean;
    Procedure
Save_LabaRugi(NO_LABARUGI1,PENDAPATAN1,PENGELUARAN1:string);
    Procedure
Update_LabaRugi(NO_LABARUGI1,PENDAPATAN1,PENGELUARAN1:string);
Private
    Sqlcmd:String;
end;
T_UserAcc = Class
    NIP : string;
    Passwd : String;
    Hak_Akses : string;
Public
    Function Find_UserID(NIP: string): Boolean;

```

```

Sqlcmd:='UPDATE T_KARYAWAN SET NAMA='+ Quotedstr(NAMA1) +', '+
'ALAMAT='+ Quotedstr(ALAMAT1) +',NO_TELP='+ Quotedstr(NO_TELP1)
+',JENIS_KELAMIN='+ JENIS_KELAMIN1 +', '+
'JABATAN='+ JABATAN1 +',STATUS='+ STATUS1 + ' WHERE NIP= '+
Quotedstr(NIP1) +";
FDatabase.ADOConn.BeginTrans;
FDatabase.ADOConn.Execute(Sqlcmd);
FDatabase.ADOConn.CommitTrans;
end;

```

```

{ TMember }

```

```

function TMember.Find_ID_Member(ID_Member1: string): boolean;

```

```

begin

```

```

    result:=false;

```

```

    Sqlcmd:='Select * From T_Member Where ID_Member = '+ Quotedstr(ID_Member1) +";

```

```

    with FDatabase do

```

```

        begin

```

```

            search.Close;

```

```

            search.SQL.Clear;

```

```

            search.SQL.Add(sqlcmd);

```

```

            search.Open;

```

```

            if not (search.IsEmpty) then

```

```

                begin

```

```

                    result:=True;

```

```

                    NAMA      := Search.FieldbyName('Nama').AsString;

```

```

                    ALAMAT    := Search.FieldbyName('Alamat').AsString;

```

```

                    NO_TELP   := Search.FieldbyName('No_Telp').AsString;

```

```

                    NO_POL    := Search.FieldbyName('No_Pol').AsString;

```

```

                    JENIS_KENDARAAN:= Search.FieldbyName('Jenis_Kendaraan').AsBoolean;

```

```

                    MERK      := Search.FieldbyName('Merk').AsString;

```

```

                    WARNA     := Search.FieldbyName('Warna').AsString;

```

```

                end;

```

```

            end;

```

```

end;

```

```

function TMember.Get_ID_Member: string;

```

```

Var

```

```

    Temp:string;

```

```

begin

```

```

    Temp:='M-' + Formatdatetime('yyMMdd',date) +'%';

```

```

    Sqlcmd:='Select Top 1 ID_MEMBER From T_MEMBER where ID_MEMBER LIKE '+
QUOTEDSTR(TEMP) +' ORDER BY ID_MEMBER desc ';

```

```

    with FDatabase do

```

```

        begin

```

```

            search.Close;

```

```

            search.SQL.Clear;

```

```

            search.SQL.Add(sqlcmd);

```

```

            search.Open;

```

```

            if not (search.IsEmpty) then

```



```

        Result:='M-' + Formatdatetime('yyMMdd',date) + copy('000',1,3-
length(inttostr(strtoint(RightStr(Search.Fieldbyname('ID_MEMBER').AsString,3)) +1))) +
inttostr(strtoint(Rightstr(Search.Fieldbyname('ID_MEMBER').AsString,3))+1)
    else
        Result:='M-' + Formatdatetime('yyMMdd',date) +'001';
    end;
end;

```

```

procedure TMember.Save_Data(ID_MEMBER1, NAMA1, ALAMAT1, NO_TELP1,
NO_POL1,
    JENIS_KENDARAAN1, MERK1, WARNA1: string);
begin
    Sqlcmd:= 'INSERT INTO T_MEMBER VALUES('+ Quotedstr(ID_MEMBER1) +','+
Quotedstr(NAMA1) +',' +
    "+ Quotedstr(ALAMAT1) +','+ Quotedstr(NO_TELP1) +','+ Quotedstr(NO_POL1) +','+
JENIS_KENDARAAN1 +',' +
    "+ Quotedstr(MERK1) +','+ Quotedstr(WARNA1) +')';
    FDatabase.ADOConn.BeginTrans;
    FDatabase.ADOConn.Execute(Sqlcmd);
    FDatabase.ADOConn.CommitTrans;
end;

```

```

procedure TMember.Update_Data(ID_MEMBER1, NAMA1, ALAMAT1, NO_TELP1,
NO_POL1, JENIS_KENDARAAN1, MERK1, WARNA1: string);
begin
    Sqlcmd:= 'UPDATE T_MEMBER SET NAMA='+ Quotedstr(NAMA1) +',' +
    'ALAMAT='+ Quotedstr(ALAMAT1) +','+ NO_TELP='+ Quotedstr(NO_TELP1)
+',NO_POL='+ Quotedstr(NO_POL1) +',' +
    'JENIS_KENDARAAN='+ JENIS_KENDARAAN1 +','+ MERK='+ Quotedstr(MERK1)
+',WARNA='+ Quotedstr(WARNA1) +' WHERE ID_MEMBER= '+
Quotedstr(ID_MEMBER1) +'';
    FDatabase.ADOConn.BeginTrans;
    FDatabase.ADOConn.Execute(Sqlcmd);
    FDatabase.ADOConn.CommitTrans;
end;

```

{ TTarif }

```
function TTarif.Find_NO_TARIF(NO_TARIF1: string): boolean;
```

```

begin
    result:=false;
    Sqlcmd:='Select * From T_TARIF Where NO_TARIF = '+ Quotedstr(NO_TARIF1) +'';
    with FDatabase do
    begin
        search.Close;
        search.SQL.Clear;
        search.SQL.Add(sqlcmd);
        search.Open;
        if not (search.IsEmpty) then
            begin

```

```

    result:=True;
    TGL_SETTING:= Search.FieldbyName('Tgl_Setting').AsString;
    TARIF_MOBIL:= Search.FieldbyName('Tarif_Mobil').AsString;
    TARIF_MOTOR := Search.FieldbyName('Tarif_Motor').AsString;
end;
end;
end;

function TTarif.Get_Last_No_Tarif: string;
Var
    Tanggal: string;
begin
    Tanggal:=Formatdatetime('yyyy/MM/dd',date);
    Sqlcmd:= 'Select Top 1 * From T_Tarif Where TGL_SETTING < '+ Quotedstr(Tanggal)
+' order by No_Tarif Desc';
    with FDatabase do
    begin
        search.Close;
        search.SQL.Clear;
        search.SQL.Add(sqlcmd);
        search.Open;
        if not (search.IsEmpty) then
            result:=Search.FieldbyName('No_Tarif').AsString
        else
            result:="";
    end;
end;

function TTarif.Get_Last_Tarif_Mobil: string;
Var
    Tanggal:string;
begin
    Tanggal:=Formatdatetime('yyyy/MM/dd',date);
    Sqlcmd:= 'Select Top 1 * From T_Tarif Where TGL_SETTING < '+ Quotedstr(Tanggal)
+' order by No_Tarif Desc';
    with FDatabase do
    begin
        search.Close;
        search.SQL.Clear;
        search.SQL.Add(sqlcmd);
        search.Open;
        if not (search.IsEmpty) then
            result:=Search.FieldbyName('Tarif_Mobil').AsString
        else
            result:="";
    end;
end;

function TTarif.Get_Last_Tarif_Motor: string;
Var

```

```

Tanggal:string;
begin
  Tanggal:=Formatdatetime('yyyy/MM/dd',date);
  Sqlcmd:='Select Top 1 * From T_Tarif Where TGL_SETTING < '+ Quotedstr(Tanggal)
+' order by No_Tarif Desc';
  with FDatabase do
  begin
    search.Close;
    search.SQL.Clear;
    search.SQL.Add(sqlcmd);
    search.Open;
    if not (search.IsEmpty) then
      result:=Search.FieldName('Tarif_Motor').AsString
    else
      result:='';
  end;
end;

```

```

function TTarif.Get_NO_TARIF: string;
Var
  Temp:string;
begin
  Temp:='T-' + Formatdatetime('yyMMdd',date);
  Result:=Temp;
end;

```

```

procedure TTarif.Save_Data(NO_TARIF1, TGL_SETTING1, TARIF_MOBIL1,
  TARIF_MOTOR1: string);
begin
  Sqlcmd:='INSERT INTO T_TARIF VALUES('+ Quotedstr(NO_TARIF1) +','+
Quotedstr(TGL_SETTING1) +',' +
  '+ TARIF_MOBIL1 +','+ TARIF_MOTOR1 +)';
  FDatabase.ADOConn.BeginTrans;
  FDatabase.ADOConn.Execute(Sqlcmd);
  FDatabase.ADOConn.CommitTrans;
end;

```

```

procedure TTarif.Update_Data(NO_TARIF1, TGL_SETTING1, TARIF_MOBIL1,
  TARIF_MOTOR1: string);
begin
  Sqlcmd:='UPDATE T_TARIF SET TARIF_MOBIL='+ TARIF_MOBIL1 +',' +
  'TARIF_MOTOR='+ TARIF_MOTOR1 +' WHERE NO_TARIF=' +
Quotedstr(NO_TARIF1) +";
  FDatabase.ADOConn.BeginTrans;
  FDatabase.ADOConn.Execute(Sqlcmd);
  FDatabase.ADOConn.CommitTrans;
end;

```

```

{ TTarifMember }

```

```

function TTarifMember.Find_NO_TARIF_MEMBER(NO_TARIF_MEMBER1: string):
boolean;
begin
    result:=false;
    Sqlcmd:='Select * From T_TARIF_MEMBER Where NO_TARIF_MEMBER = '+
Quotedstr(NO_TARIF_MEMBER1) +";
    with FDatabase do
    begin
        search.Close;
        search.SQL.Clear;
        search.SQL.Add(sqlcmd);
        search.Open;
        if not (search.IsEmpty) then
        begin
            result:=True;
            TGL_SETTING:= Search.FieldbyName('Tgl_Setting').AsString;
            TARIF_MOBIL:= Search.FieldbyName('Tarif_Mobil').AsString;
            TARIF_MOTOR := Search.FieldbyName('Tarif_Motor').AsString;
        end;
    end;
end;

```

```

function TTarifMember.Get_Last_No_Tarif_Member: string;
Var
    Tanggal: string;
begin
    Tanggal:=Formatdatetime('yyyy/MM/dd',date);
    Sqlcmd:= 'Select Top 1 * From T_Tarif_Member Where TGL_SETTING < '+
Quotedstr(Tanggal) +' order by No_Tarif_Member Desc';
    with FDatabase do
    begin
        search.Close;
        search.SQL.Clear;
        search.SQL.Add(sqlcmd);
        search.Open;
        if not (search.IsEmpty) then
            result:=Search.FieldbyName('No_Tarif_Member').AsString
        else
            result:="";
    end;
end;

```

```

function TTarifMember.Get_Last_Tarif_Mobil: string;
Var
    Tanggal:string;
begin
    Tanggal:=Formatdatetime('yyyy/MM/dd',date);
    Sqlcmd:= 'Select Top 1 * From T_Tarif_Member Where TGL_SETTING < '+
Quotedstr(Tanggal) +' order by No_Tarif_Member Desc';
    with FDatabase do

```

```

begin
  search.Close;
  search.SQL.Clear;
  search.SQL.Add(sqlcmd);
  search.Open;
  if not (search.IsEmpty) then
    result:=Search.FieldbyName('Tarif_Mobil').AsString
  else
    result:="";
end;
end;

function TTarifMember.Get_Last_Tarif_Motor: string;
Var
  Tanggal:string;
begin
  Tanggal:=Formatdatetime('yyyy/MM/dd',date);
  Sqlcmd:= 'Select Top 1 * From T_Tarif_Member Where TGL_SETTING < '+
Quotedstr(Tanggal) +' order by No_Tarif_Member Desc';
  with FDatabase do
  begin
    search.Close;
    search.SQL.Clear;
    search.SQL.Add(sqlcmd);
    search.Open;
    if not (search.IsEmpty) then
      result:=Search.FieldbyName('Tarif_Motor').AsString
    else
      result:="";
  end;
end;

function TTarifMember.Get_NO_TARIF_MEMBER: string;
Var
  Temp:string;
begin
  Temp:='T-' + Formatdatetime('yyMMdd',date);
  Result:=Temp;
end;

procedure TTarifMember.Save_Data(NO_TARIF_MEMBER1, TGL_SETTING1,
  TARIF_MOBIL1, TARIF_MOTOR1: string);
begin
  Sqlcmd:= 'INSERT INTO T_TARIF_MEMBER VALUES('+
Quotedstr(NO_TARIF_MEMBER1) +','+ Quotedstr(TGL_SETTING1) +',' +
"+ TARIF_MOBIL1 +','+ TARIF_MOTOR1 +)';
  FDatabase.ADOConn.BeginTrans;
  FDatabase.ADOConn.Execute(Sqlcmd);
  FDatabase.ADOConn.CommitTrans;
end;

```

```

procedure TTarifMember.Update_Data(NO_TARIF_MEMBER1, TGL_SETTING1,
TARIF_MOBIL1, TARIF_MOTOR1: string);
begin
  Sqlcmd:= 'UPDATE T_TARIF_MEMBER SET TARIF_MOBIL='+ TARIF_MOBIL1 +','
+
  'TARIF_MOTOR='+ TARIF_MOTOR1 + ' WHERE NO_TARIF_MEMBER= '+
  Quotedstr(NO_TARIF_MEMBER1) +";
  FDatabase.ADOConn.BeginTrans;
  FDatabase.ADOConn.Execute(Sqlcmd);
  FDatabase.ADOConn.CommitTrans;
end;

```

{ TAktifMember }

```

function TAktifMember.Get_Last_Aktif(ID_Member1: string): string;
begin
  Sqlcmd:='Select Top 1 * From T_Aktif_Member Where ID_Member= '+
  Quotedstr(ID_Member1) +' order by No_Aktif desc ';
  with FDatabase do
  begin
    search.Close;
    search.SQL.Clear;
    search.SQL.Add(sqlcmd);
    search.Open;
    if not (search.IsEmpty) then
      result:=Formatdatetime('dd/MM/yyyy',search.Fieldbyname('Tgl_Aktif').AsDateTime)
    else
      result:="";
  end;
end;

```

```

function TAktifMember.Get_NO_AKTIF_MEMBER: string;
Var
  Temp:string;
begin
  Temp:='A-' + Formatdatetime('yyMMdd',date) +%';
  Sqlcmd:='Select Top 1 NO_AKTIF From T_AKTIF_MEMBER where NO_AKTIF LIKE
'+ QUOTEDSTR(TEMP) +' ORDER BY NO_AKTIF desc ';
  with FDatabase do
  begin
    search.Close;
    search.SQL.Clear;
    search.SQL.Add(sqlcmd);
    search.Open;
    if not (search.IsEmpty) then
      Result:='A-' + Formatdatetime('yyMMdd',date) + copy('000',1,3-
length(inttostr(strtoint(RightStr(Search.Fieldbyname('NO_AKTIF').AsString,3)) +1))) +
inttostr(strtoint(Rightstr(Search.Fieldbyname('NO_AKTIF').AsString,3))+1)
    else

```



```

        Result:='A-' + Formatdatetime('yyMMdd',date) +'001';
    end;
end;

procedure TaktifMember.Save_Data(NO_AKTIF1, TGL_AKTIF1, ID_MEMBER1,
    NO_TARIF_MEMBER1: string);
begin
    Sqlcmd:= 'INSERT INTO T_AKTIF_MEMBER VALUES('+ Quotedstr(NO_AKTIF1)
+' '+ Quotedstr(TGL_AKTIF1) +', ' +
    "+ Quotedstr(ID_MEMBER1) +','+ Quotedstr(NO_TARIF_MEMBER1) +)';
    FDatabase.ADOConn.BeginTrans;
    FDatabase.ADOConn.Execute(Sqlcmd);
    FDatabase.ADOConn.CommitTrans;
end;

{ TKapasitas }
function TKapasitas.Find_NO_KAPASITAS(NO_KAPASITAS1: string): boolean;
begin
    result:=false;
    Sqlcmd:='Select * From T_KAPASITAS Where NO_KAPASITAS = '+
Quotedstr(NO_KAPASITAS1) +";
    with FDatabase do
        begin
            search.Close;
            search.SQL.Clear;
            search.SQL.Add(sqlcmd);
            search.Open;
            if not (search.IsEmpty) then
                begin
                    result:=True;
                    TGL_SETTING:= Search.FieldByName('Tgl_Setting').AsString;
                    KAPASITAS_MOBIL:= Search.FieldByName('Kapasitas_Mobil').AsInteger;
                    KAPASITAS_MOTOR := Search.FieldByName('Kapasitas_Motor').AsInteger;
                end;
            end;
        end;
end;

function TKapasitas.Get_Last_Kapasitas_Mobil: integer;
begin
    Sqlcmd:= 'Select Top 1 * From T_Kapasitas order by No_Kapasitas Desc';
    with FDatabase do
        begin
            search.Close;
            search.SQL.Clear;
            search.SQL.Add(sqlcmd);
            search.Open;
            if not (search.IsEmpty) then
                result:=Search.FieldByName('Kapasitas_Mobil').AsInteger
            else

```

```

        result:=0;
    end;
end;

function TKapasitas.Get_Last_Kapasitas_Motor: integer;
begin
    Sqlcmd:= 'Select Top 1 * From T_Kapasitas order by No_Kapasitas Desc';
    with FDatabase do
        begin
            search.Close;
            search.SQL.Clear;
            search.SQL.Add(sqlcmd);
            search.Open;
            if not (search.IsEmpty) then
                result:=Search.FieldbyName('Kapasitas_Motor').AsInteger
            else
                result:=0;
            end;
        end;
    end;
end;

```

```

function TKapasitas.Get_Last_No_Kapasitas: string;
Var
    Tanggal:string;
begin
    Tanggal:=Formatdatetime('yyyy/MM/dd',date);
    Sqlcmd:= 'Select Top 1 * From T_kapasitas Where Tgl_Setting < '+ Quotedstr(Tanggal) +'
order by No_Kapasitas Desc';
    with FDatabase do
        begin
            search.Close;
            search.SQL.Clear;
            search.SQL.Add(sqlcmd);
            search.Open;
            if not (search.IsEmpty) then
                result:=Search.FieldbyName('No_Kapasitas').AsString
            else
                result:="";
            end;
        end;
    end;
end;

```

```

function TKapasitas.Get_NO_KAPASITAS: string;
Var
    Temp:string;
begin
    Temp:='K-' + Formatdatetime('yyMMdd',date);
    Result:=Temp;
end;
function TKapasitas.Get_Sisa_Kapasitas_Mobil: integer;
Var
    Sisa_Kapasitas_Mobil:integer;

```

```

Total_Parkir_Mobil:integer;
begin
  Total_Parkir_Mobil:=0;
  Sisa_Kapasitas_Mobil:= Get_Last_Kapasitas_Mobil;
  Sqlcmd:= 'SELECT COUNT(T_PARKIR_UMUM.NO_PARKIR_UMUM) FROM
T_PARKIR INNER JOIN ' +
  'T_PARKIR_UMUM ON T_PARKIR.NO_PARKIR =
T_PARKIR_UMUM.NO_PARKIR ' +
  'WHERE (T_PARKIR.JAM KELUAR = '+ Quotedstr("")
+')AND(T_PARKIR_UMUM.JENIS_KENDARAAN = 0)';
  with FDatabase do
  begin
    search.Close;
    search.SQL.Clear;
    search.SQL.Add(sqlcmd);
    search.Open;
    if not (search.IsEmpty) then
    begin
      Total_Parkir_Mobil:= Search.Fields[0].AsInteger;
    end;
  end;
  Sqlcmd:= 'SELECT COUNT(T_PARKIR_MEMBER.NO_PARKIR_MEMBER) AS
Expr1 FROM T_PARKIR INNER JOIN ' +
  'T_PARKIR_MEMBER ON T_PARKIR.NO_PARKIR =
T_PARKIR_MEMBER.NO_PARKIR INNER JOIN ' +
  'T_MEMBER ON T_PARKIR_MEMBER.ID_MEMBER =
T_MEMBER.ID_MEMBER ' +
  'WHERE(T_MEMBER.JENIS_KENDARAAN = 0) AND
(T_PARKIR.JAM KELUAR = '+ Quotedstr("") +)';
  with FDatabase do
  begin
    search.Close;
    search.SQL.Clear;
    search.SQL.Add(sqlcmd);
    search.Open;
    if not (search.IsEmpty) then
    begin
      Total_Parkir_Mobil:=Total_Parkir_Mobil + Search.Fields[0].AsInteger;
    end;
  end;
  Sisa_Kapasitas_Mobil:=Sisa_Kapasitas_Mobil-Total_Parkir_Mobil;
  Result:=Sisa_Kapasitas_Mobil;
end;

function TKapasitas.Get_Sisa_Kapasitas_Motor: integer;
Var
  Sisa_Kapasitas_Mobil:integer;
  Total_Parkir_Mobil:integer;
begin
  Total_Parkir_Mobil:=0;

```

```

Sisa_Kapasitas_Mobil:= Get_Last_Kapasitas_Motor;
Sqlcmd:= 'SELECT COUNT(T_PARKIR_UMUM.NO_PARKIR_UMUM) FROM
T_PARKIR INNER JOIN ' +
    'T_PARKIR_UMUM ON T_PARKIR.NO_PARKIR =
T_PARKIR_UMUM.NO_PARKIR ' +
    'WHERE (T_PARKIR.JAM_KELUAR = '+ Quotedstr("")
+')AND(T_PARKIR_UMUM.JENIS_KENDARAAN = 1)';
with FDatabase do
begin
    search.Close;
    search.SQL.Clear;
    search.SQL.Add(sqlcmd);
    search.Open;
    if not (search.IsEmpty) then
    begin
        Total_Parkir_Mobil:= Search.Fields[0].AsInteger;
    end;
end;
Sqlcmd:= 'SELECT COUNT(T_PARKIR_MEMBER.NO_PARKIR_MEMBER) AS
Expr1 FROM T_PARKIR INNER JOIN ' +
    'T_PARKIR_MEMBER ON T_PARKIR.NO_PARKIR =
T_PARKIR_MEMBER.NO_PARKIR INNER JOIN ' +
    'T_MEMBER ON T_PARKIR_MEMBER.ID_MEMBER =
T_MEMBER.ID_MEMBER ' +
    'WHERE(T_MEMBER.JENIS_KENDARAAN = 1) AND
(T_PARKIR.JAM_KELUAR = '+ Quotedstr("") +)';
with FDatabase do
begin
    search.Close;
    search.SQL.Clear;
    search.SQL.Add(sqlcmd);
    search.Open;
    if not (search.IsEmpty) then
    begin
        Total_Parkir_Mobil:=Total_Parkir_Mobil + Search.Fields[0].AsInteger;
    end;
end;
Sisa_Kapasitas_Mobil:=Sisa_Kapasitas_Mobil-Total_Parkir_Mobil;
Result:=Sisa_Kapasitas_Mobil;
end;

```

```

procedure TKapasitas.Save_Data(NO_KAPASITAS1, TGL_SETTING1,
KAPASITAS_MOBIL1, KAPASITAS_MOTOR1: string);
begin
    Sqlcmd:= 'INSERT INTO T_KAPASITAS VALUES('+ Quotedstr(NO_KAPASITAS1)
+', '+ Quotedstr(TGL_SETTING1) +', ' +
    '+ KAPASITAS_MOBIL1 +', '+ KAPASITAS_MOTOR1 +)';
    FDatabase.ADOConn.BeginTrans;
    FDatabase.ADOConn.Execute(Sqlcmd);
    FDatabase.ADOConn.CommitTrans;

```

```

end;

procedure TKapasitas.Update_Data(NO_KAPASITAS1, TGL_SETTING1,
  KAPASITAS_MOBIL1, KAPASITAS_MOTOR1: string);
begin
  Sqlcmd:= 'UPDATE T_KAPASITAS SET KAPASITAS_MOBIL='+
  KAPASITAS_MOBIL1 +', ' +
  'KAPASITAS_MOTOR='+ KAPASITAS_MOTOR1 + ' WHERE NO_KAPASITAS=' +
  Quotedstr(NO_KAPASITAS1) +";
  FDatabase.ADOConn.BeginTrans;
  FDatabase.ADOConn.Execute(Sqlcmd);
  FDatabase.ADOConn.CommitTrans;
end;

```

```

{ T_Parkir }

```

```

function T_Parkir.Find_Parkir(NO_PARKIR1: string): Boolean;
begin
  result:=false;
  Sqlcmd:='Select * From T_PARKIR Where NO_PARKIR = '+ Quotedstr(NO_PARKIR1)
  +";
  with FDatabase do
  begin
    search.Close;
    search.SQL.Clear;
    search.SQL.Add(sqlcmd);
    search.Open;
    if not (search.IsEmpty) then
    begin
      result:=True;
      TGL_PARKIR := Search.FieldByName('Tgl_Parkir').AsString;
      JAM_MASUK := Search.FieldByName('Jam_Masuk').AsString;
      TGL_KELUAR := Search.FieldByName('Tgl_Keluar').AsString;
      JAM_KELUAR := Search.FieldByName('Jam_Keluar').AsString;
    end;
  end;
end;

```

```

function T_Parkir.Find_Parkir_MOBIL1(NO_POL1: string): Byte;
begin
  Sqlcmd:= 'SELECT TOP 1
  T_PARKIR.NO_PARKIR,T_PARKIR.TGL_PARKIR,T_PARKIR.JAM_MASUK,T_PARKIR.JAM_KELUAR '+
  'FROM T_PARKIR INNER JOIN T_PARKIR_UMUM ON T_PARKIR.NO_PARKIR
  = T_PARKIR_UMUM.NO_PARKIR '+
  'WHERE (T_PARKIR_UMUM.JENIS_KENDARAAN = 0) AND
  T_PARKIR_UMUM.NO_POL = '+ Quotedstr(No_Pol1) +'ORDER BY
  T_PARKIR.NO_PARKIR DESC';
  with FDatabase do
  begin
    search.Close;

```

```

search.SQL.Clear;
search.SQL.Add(sqlcmd);
search.Open;
if not (search.IsEmpty) then
begin
  if Trim(Search.Fields[3].AsString) ='' then
  begin
    Result:=1;
    NO_PARKIR := Search.Fields[0].AsString;
    TGL_PARKIR := Search.Fields[1].AsString;
    JAM_MASUK := Search.Fields[2].AsString;
  end
  else
    Result:=2;
  end
else
  Result:=0;
end;
end;

```

```

function T_Parkir.Find_Parkir_MOBIL2(NO_POL1: string): Byte;
begin
  Sqlcmd:= 'SELECT TOP 1
T_PARKIR.NO_PARKIR,T_PARKIR.TGL_PARKIR,T_PARKIR.JAM_MASUK,T_PARK
IR.JAM_KELUAR ' +
  'FROM T_PARKIR INNER JOIN T_PARKIR_MEMBER ON
T_PARKIR.NO_PARKIR = T_PARKIR_MEMBER.NO_PARKIR INNER JOIN ' +
  'T_MEMBER ON T_PARKIR_MEMBER.ID_MEMBER =
T_MEMBER.ID_MEMBER ' +
  'WHERE (T_MEMBER.JENIS_KENDARAAN = 0) AND T_MEMBER.NO_POL = '+
Quotedstr(No_Pol1) +'ORDER BY T_PARKIR.NO_PARKIR DESC';
  with FDatabase do
  begin
    search.Close;
    search.SQL.Clear;
    search.SQL.Add(sqlcmd);
    search.Open;
    if not (search.IsEmpty) then
    begin
      if Trim(Search.Fields[3].AsString) ='' then
      begin
        Result:=1;
        NO_PARKIR := Search.Fields[0].AsString;
        TGL_PARKIR := Search.Fields[1].AsString;
        JAM_MASUK := Search.Fields[2].AsString;
      end
      else
        Result:=2;
      end
    end
  else
    Result:=0;
  end;
end;

```



```

    Result:=0;
end;
end;
function T_Parkir.Find_Parkir_MOTOR1(NO_POL1: string): Byte;
begin
    Sqlcmd:= 'SELECT TOP 1
T_PARKIR.NO_PARKIR,T_PARKIR.TGL_PARKIR,T_PARKIR.JAM_MASUK,T_PARK
IR.JAM_KELUAR '+
    'FROM T_PARKIR INNER JOIN T_PARKIR_UMUM ON T_PARKIR.NO_PARKIR
= T_PARKIR_UMUM.NO_PARKIR '+
    'WHERE (T_PARKIR_UMUM.JENIS_KENDARAAN = 1) AND
T_PARKIR_UMUM.NO_POL = '+ Quotedstr(No_Pol1) +'ORDER BY
T_PARKIR.NO_PARKIR DESC';
    with FDatabase do
    begin
        search.Close;
        search.SQL.Clear;
        search.SQL.Add(sqlcmd);
        search.Open;
        if not (search.IsEmpty) then
        begin
            if Trim(Search.Fields[3].AsString) ='' then
            begin
                Result:=1;
                NO_PARKIR := Search.Fields[0].AsString;
                TGL_PARKIR := Search.Fields[1].AsString;
                JAM_MASUK := Search.Fields[2].AsString;
            end
            else
                Result:=2;
            end
            else
                Result:=0;
        end;
    end;
end;

```

```

function T_Parkir.Find_Parkir_MOTOR2(NO_POL1: string): Byte;
begin
    Sqlcmd:= 'SELECT TOP 1
T_PARKIR.NO_PARKIR,T_PARKIR.TGL_PARKIR,T_PARKIR.JAM_MASUK,T_PARK
IR.JAM_KELUAR '+
    'FROM T_PARKIR INNER JOIN T_PARKIR_MEMBER ON
T_PARKIR.NO_PARKIR = T_PARKIR_MEMBER.NO_PARKIR INNER JOIN '+
    'T_MEMBER ON T_PARKIR_MEMBER.ID_MEMBER =
T_MEMBER.ID_MEMBER '+
    'WHERE (T_MEMBER.JENIS_KENDARAAN = 1) AND T_MEMBER.NO_POL = '+
Quotedstr(No_Pol1) +'ORDER BY T_PARKIR.NO_PARKIR DESC';
    with FDatabase do
    begin
        search.Close;

```

```

search.SQL.Clear;
search.SQL.Add(sqlcmd);
search.Open;
if not (search.IsEmpty) then
begin
  if Trim(Search.Fields[3].AsString) = " then
  begin
    Result:=1;
    NO_PARKIR := Search.Fields[0].AsString;
    TGL_PARKIR := Search.Fields[1].AsString;
    JAM_MASUK := Search.Fields[2].AsString;
  end
  else
    Result:=2;
  end
else
  Result:=0;
end;
end;

function T_Parkir.Get_No_Parkir: String;
Var
  Temp:string;
begin
  Temp:='P-' + Formatdatetime('yyMMdd',date) + '%';
  Sqlcmd:='Select Top 1 No_Parkir From T_PARKIR where NO_PARKIR LIKE '+
QUOTEDSTR(TEMP) +' ORDER BY NO_PARKIR desc ';
  with FDatabase do
  begin
    search.Close;
    search.SQL.Clear;
    search.SQL.Add(sqlcmd);
    search.Open;
    if not (search.IsEmpty) then
      Result:='P-' + Formatdatetime('yyMMdd',date) + copy('0000',1,4-
length(inttostr(strtoint(RightStr(Search.Fieldbyname('NO_PARKIR').AsString,4)) +1))) +
inttostr(strtoint(Rightstr(Search.Fieldbyname('NO_PARKIR').AsString,4))+1)
    else
      Result:='P-' + Formatdatetime('yyMMdd',date) +'0001';
    end;
  end;
end;

procedure T_Parkir.Save_Parkir_Keluar(NO_PARKIR1,TGL_KELUAR1,
  JAM_KELUAR1,OPERATOR_KELUAR1: String);
begin
  Sqlcmd:= 'UPDATE T_PARKIR SET TGL_KELUAR = '+ Quotedstr(TGL_KELUAR1)
+',JAM_KELUAR = '+ Quotedstr(JAM_KELUAR1) +', ' +
  'OPERATOR_KELUAR = '+ Quotedstr(OPERATOR_KELUAR1) +' WHERE
NO_PARKIR = '+ Quotedstr(NO_PARKIR1) +' ';

```

```

FDatabase.ADOConn.BeginTrans;
FDatabase.ADOConn.Execute(Sqlcmd);
FDatabase.ADOConn.CommitTrans;
end;

```

```

procedure T_Parkir.Save_Parkir_Masuk(NO_PARKIR1, TGL_PARKIR1, JAM_MASUK1,
OPERATOR_MASUK1: String);
begin
  Sqlcmd:= 'INSERT INTO T_PARKIR VALUES('+ Quotedstr(NO_PARKIR1) +','+
Quotedstr(TGL_PARKIR1) +',' +
  "+ Quotedstr(JAM_MASUK1) +','+ Quotedstr(TGL_PARKIR1) +','+ Quotedstr(") +','+
Quotedstr(OPERATOR_MASUK1) +','+ Quotedstr(") +')';
  FDatabase.ADOConn.BeginTrans;
  FDatabase.ADOConn.Execute(Sqlcmd);
  FDatabase.ADOConn.CommitTrans;
end;

```

```

{ T_Parkir_Umum }
function T_Parkir_Umum.Find_Parkir_Umum(NO_PARKIR1: string): Boolean;
begin
  result:=false;
  Sqlcmd:='Select * From T_PARKIR_UMUM Where NO_PARKIR = '+
Quotedstr(NO_PARKIR1) +";
  with FDatabase do
  begin
    search.Close;
    search.SQL.Clear;
    search.SQL.Add(sqlcmd);
    search.Open;
    if not (search.IsEmpty) then
    begin
      result:=True;
      NO_PARKIR_UMUM := search.FieldbyName('NO_PARKIR_UMUM').AsString;
      NO_POL := search.FieldbyName('NO_POL').AsString;
      JENIS_KENDARAAN := search.FieldbyName('JENIS_KENDARAAN').AsBoolean;
      NO_TARIF := search.FieldbyName('NO_TARIF').AsString;
      TOTAL_PARKIR := search.FieldbyName('TOTAL_PARKIR').AsString;
    end;
  end;
end;
function T_Parkir_Umum.Get_No_Parkir_Umum: String;
Var
  Temp:string;
begin
  Temp:='PU-' + Formatdatetime('yyMMdd',date) +'%';
  Sqlcmd:='Select Top 1 No_Parkir_Umum From T_PARKIR_UMUM where
NO_PARKIR_UMUM LIKE '+ QUOTEDSTR(TEMP) +' ORDER BY
NO_PARKIR_UMUM desc ';
  with FDatabase do
  begin

```

```

search.Close;
search.SQL.Clear;
search.SQL.Add(sqlcmd);
search.Open;
if not (search.IsEmpty) then
    Result:='PU-' + Formatdatetime('yyMMdd',date) + copy('0000',1,4-
length(inttostr(strtoint(RightStr(Search.Fieldbyname('NO_PARKIR_UMUM').AsString,4))
+1))) +

inttostr(strtoint(Rightstr(Search.Fieldbyname('NO_PARKIR_UMUM').AsString,4))+1)
    else
    Result:='PU-' + Formatdatetime('yyMMdd',date) +'0001';
end;
end;

procedure T_Parkir_Umum.Save_Data(NO_PARKIR_UMUM1, NO_PARKIR1, NO_POL1,
    JENIS_KENDARAAN1,NO_TARIF1, TOTAL_PARKIR1: String);
begin
    Sqlcmd:= 'INSERT INTO T_PARKIR_UMUM VALUES(+
Quotedstr(NO_PARKIR_UMUM1) +','+ Quotedstr(NO_PARKIR1) +',' +
    "+ Quotedstr(NO_POL1) +','+ JENIS_KENDARAAN1 +','+ Quotedstr(NO_TARIF1)
+','+ TOTAL_PARKIR1 +)';
    FDatabase.ADOConn.BeginTrans;
    FDatabase.ADOConn.Execute(Sqlcmd);
    FDatabase.ADOConn.CommitTrans;
end;

procedure T_Parkir_Umum.Update_Data(NO_PARKIR1, NO_TARIF1,
    TOTAL_PARKIR1: String);
begin
    Sqlcmd:= 'UPDATE T_PARKIR_UMUM SET NO_TARIF= '+ Quotedstr(NO_TARIF1)
+',TOTAL_PARKIR= '+ TOTAL_PARKIR1 + ' ' +
    'WHERE NO_PARKIR= '+ Quotedstr(NO_PARKIR1) +";
    FDatabase.ADOConn.BeginTrans;
    FDatabase.ADOConn.Execute(Sqlcmd);
    FDatabase.ADOConn.CommitTrans;
end;

{ T_Parkir_Member }
function T_Parkir_Member.Find_Parkir_Member(NO_PARKIR1: string): Boolean;
begin
    result:=false;
    Sqlcmd:='Select * From T_PARKIR_MEMBER Where NO_PARKIR = '+
Quotedstr(NO_PARKIR1) +";
    with FDatabase do
    begin
        search.Close;
        search.SQL.Clear;
        search.SQL.Add(sqlcmd);
        search.Open;

```

```

    if not (search.IsEmpty) then
    begin
        result:=True;
        NO_PARKIR_MEMBER:=
search.FieldbyName('NO_PARKIR_MEMBER').AsString;
        ID_MEMBER := search.FieldbyName('ID_MEMBER').AsString;
    end;
end;
end;

function T_Parkir_Member.Get_No_Parkir_Member: String;
Var
    Temp:string;
begin
    Temp:='PM-' + Formatdatetime('yyMMdd',date) +'%';
    Sqlcmd:='Select Top 1 No_Parkir_Member From T_PARKIR_MEMBER where
NO_PARKIR_MEMBER LIKE '+ QUOTEDSTR(TEMP) +' ORDER BY
NO_PARKIR_MEMBER desc ';
    with FDatabase do
    begin
        search.Close;
        search.SQL.Clear;
        search.SQL.Add(sqlcmd);
        search.Open;
        if not (search.IsEmpty) then
            Result:='PM-' + Formatdatetime('yyMMdd',date) + copy('0000',1,4-
length(inttostr(strtoint(RightStr(Search.Fieldbyname('NO_PARKIR_MEMBER').AsString,4)
)+1))) +
inttostr(strtoint(Rightstr(Search.Fieldbyname('NO_PARKIR_MEMBER').AsString,4))+1)
else
            Result:='PM-' + Formatdatetime('yyMMdd',date) +'0001';
        end;
    end;

procedure T_Parkir_Member.Save_Data(NO_PARKIR_MEMBER1, NO_PARKIR1,
ID_MEMBER1: String);
begin
    Sqlcmd:= 'INSERT INTO T_PARKIR_MEMBER VALUES(+
Quotedstr(NO_PARKIR_MEMBER1) +','+ Quotedstr(NO_PARKIR1) +',' +
"+ Quotedstr(ID_MEMBER1) +)';
    FDatabase.ADOConn.BeginTrans;
    FDatabase.ADOConn.Execute(Sqlcmd);
    FDatabase.ADOConn.CommitTrans;
end;

{ T_Pengeluaran }
procedure T_Pengeluaran.Delete_Pengeluaran(NO_PENGELUARAN1:String);
begin

```

```

Sqlcmd:= 'DELETE T_PENGELUARAN WHERE NO_PENGELUARAN = '+
QUOTEDSTR(NO_PENGELUARAN1) +' ';
FDatabase.ADOConn.BeginTrans;
FDatabase.ADOConn.Execute(Sqlcmd);
FDatabase.ADOConn.CommitTrans;
end;

```

```

function T_Pengeluaran.Find_No_Pengeluaran(No_Pengeluaran1:string): Boolean;
begin
    result:=false;
    Sqlcmd:='Select * From T_PENGELUARAN Where NO_PENGELUARAN = '+
Quotedstr(No_Pengeluaran1) +";
    with FDatabase do
    begin
        search.Close;
        search.SQL.Clear;
        search.SQL.Add(sqlcmd);
        search.Open;
        if not (search.IsEmpty) then
        begin
            result:=True;
            TGL_PENGELUARAN:= search.FieldByName("TGL_PENGELUARAN").AsString;
            TOTAL      := search.FieldByName("TOTAL").AsString;
        end;
    end;
end;

```

```

function T_Pengeluaran.Get_No_Pengeluaran: String;
begin
    result:='PG-' + Formatdatetime('yyMMdd',date) +";
end;

```

```

procedure T_Pengeluaran.Save_Pengeluaran(NO_PENGELUARAN1,
TGL_PENGELUARAN1,
TOTAL1: string);
begin
    Sqlcmd:= 'INSERT INTO T_PENGELUARAN VALUES(+
Quotedstr(NO_PENGELUARAN1) +' '+ Quotedstr(TGL_PENGELUARAN1) +' , '+
"+ TOTAL1 +)';
    FDatabase.ADOConn.BeginTrans;
    FDatabase.ADOConn.Execute(Sqlcmd);
    FDatabase.ADOConn.CommitTrans;
end;

```

```

procedure T_Pengeluaran.Update_Pengeluaran(NO_PENGELUARAN1,
TOTAL1: string);
begin
    Sqlcmd:= 'UPDATE T_PENGELUARAN SET TOTAL= '+ TOTAL1 +' WHERE
NO_PENGELUARAN = '+ quotedstr(NO_PENGELUARAN1) +' ';
    FDatabase.ADOConn.BeginTrans;

```



```
FDatabase.ADOConn.Execute(Sqlcmd);
FDatabase.ADOConn.CommitTrans;
end;
```

```
{ T_Pengeluaran_D }
```

```
procedure T_Pengeluaran_D.Delete_Pengeluaran_D(NO_PENGELUARAN1: string);
begin
    Sqlcmd:= 'DELETE T_PENGELUARAN_D WHERE NO_PENGELUARAN = '+
    quotedstr(NO_PENGELUARAN1) + ' ';
    FDatabase.ADOConn.BeginTrans;
    FDatabase.ADOConn.Execute(Sqlcmd);
    FDatabase.ADOConn.CommitTrans;
end;
```

```
function T_Pengeluaran_D.Get_No_Pengeluaran_D(NO_PENGELUARAN1: String;
NO: integer): String;
begin
    Result:=NO_PENGELUARAN1 + copy('000',1,3-length(inttostr(No))) + inttostr(No);
end;
```

```
procedure T_Pengeluaran_D.Save_Pengeluaran_D(NO_PENGELUARAN_D1,
NO_PENGELUARAN1,KETERANGAN1, BIAYA1: string);
begin
    Sqlcmd:= 'INSERT INTO T_PENGELUARAN_D VALUES('+
    Quotedstr(NO_PENGELUARAN_D1) +', '+
    "+ Quotedstr(NO_PENGELUARAN1) +','+ Quotedstr(KETERANGAN1) +','+
    BIAYA1 +)';
    FDatabase.ADOConn.BeginTrans;
    FDatabase.ADOConn.Execute(Sqlcmd);
    FDatabase.ADOConn.CommitTrans;
end;
```

```
{ T_Pendapatan }
```

```
function T_Pendapatan.Find_No_Pendapatan(NO_PENDAPATAN1: String): Boolean;
begin
    result:=false;
    Sqlcmd:='Select * From T_PENDAPATAN Where NO_PENDAPATAN = '+
    Quotedstr(No_Pendapatan1) +";
    with FDatabase do
    begin
        search.Close;
        search.SQL.Clear;
        search.SQL.Add(sqlcmd);
        search.Open;
        if not (search.IsEmpty) then
        begin
            result:=True;
            TGL_PENDAPATAN:= search.FieldByName('TGL_PENDAPATAN').AsString;
            TOTAL_PARKIR_UMUM:=search.FieldByName('TOTAL_PARKIR_UMUM').AsString;
```

```

TOTAL_PARKIR_MEMBER:=search.FieldbyName('TOTAL_PARKIR_MEMBER').AsString;
end;
end;
end;

```

```

function T_Pendapatan.Get_No_Pendapatan: String;
begin

```

```

    result:= 'PD-' + Formatdatetime('yyMMdd',date) +";
end;

```

```

procedure T_Pendapatan.Save_Pendapatan(NO_PENDAPATAN1, TGL_PENDAPATAN1,
TOTAL_PENDAPATAN_UMUM1, TOTAL_PENDAPATAN_MEMBER1: string);
begin

```

```

    Sqlcmd:= 'INSERT INTO T_PENDAPATAN VALUES(+
Quotedstr(NO_PENDAPATAN1) +','+ Quotedstr(TGL_PENDAPATAN1) +','+
"+ TOTAL_PENDAPATAN_UMUM1 +','+ TOTAL_PENDAPATAN_MEMBER1
+');

```

```

    FDatabase.ADOConn.BeginTrans;
    FDatabase.ADOConn.Execute(Sqlcmd);
    FDatabase.ADOConn.CommitTrans;
end;

```

```

procedure T_Pendapatan.Update_Pendapatan(NO_PENDAPATAN1,
TGL_PENDAPATAN1,
TOTAL_PENDAPATAN_UMUM1, TOTAL_PENDAPATAN_MEMBER1: string);
begin

```

```

    Sqlcmd:= 'UPDATE T_PENDAPATAN SET
TOTAL_PARKIR_UMUM=TOTAL_PARKIR_UMUM + '+
TOTAL_PENDAPATAN_UMUM1 +','+
'TOTAL_PARKIR_MEMBER=TOTAL_PARKIR_MEMBER + '+
TOTAL_PENDAPATAN_MEMBER1 +','+
'WHERE NO_PENDAPATAN= '+ Quotedstr(NO_PENDAPATAN1) +";

```

```

    FDatabase.ADOConn.BeginTrans;
    FDatabase.ADOConn.Execute(Sqlcmd);
    FDatabase.ADOConn.CommitTrans;
end;

```

```

{ T_LabaRugi }

```

```

function T_LabaRugi.Find_No_LabarRugi(NO_LABARUGI1: String): Boolean;
begin

```

```

    result:=false;
    Sqlcmd:='Select * From T_LABARUGI Where NO_LABARUGI = '+
Quotedstr(No_LABARUGI1) +";

```

```

    with FDatabase do
    begin

```

```

        search.Close;
        search.SQL.Clear;
        search.SQL.Add(sqlcmd);
        search.Open;

```

```

if not (search.IsEmpty) then
begin
    result:=True;
    PENDAPATAN:=search.FieldByName('PENDAPATAN').AsString;
    PENGELUARAN:=search.FieldByName('PENGELUARAN').AsString;
end;
end;
end;

function T_LabaRugi.Get_No_LabaRugi: String;
begin
    Result:='LB-' + Formatdatetime('yyMM',date);
end;

procedure T_LabaRugi.Save_LabaRugi(NO_LABARUGI1, PENDAPATAN1,
    PENGELUARAN1: string);
begin
    Sqlcmd:= 'INSERT INTO T_LABARUGI VALUES('+ Quotedstr(NO_LABARUGI1)
+', '+ PENDAPATAN1 +', ' +
    "+ PENGELUARAN1 +)';
    FDatabase.ADOConn.BeginTrans;
    FDatabase.ADOConn.Execute(Sqlcmd);
    FDatabase.ADOConn.CommitTrans;
end;

procedure T_LabaRugi.Update_LabaRugi(NO_LABARUGI1, PENDAPATAN1,
    PENGELUARAN1: string);
begin
    Sqlcmd:= 'UPDATE T_LABARUGI SET PENDAPATAN=PENDAPATAN + '+
PENDAPATAN1 +', ' +
    'PENGELUARAN=PENGELUARAN + '+ PENGELUARAN1 + ' ' +
    'WHERE NO_LABARUGI= '+ Quotedstr(NO_LABARUGI1) +"';
    FDatabase.ADOConn.BeginTrans;
    FDatabase.ADOConn.Execute(Sqlcmd);
    FDatabase.ADOConn.CommitTrans;
end;

{ T_UserAcc }
procedure T_UserAcc.Delete_UserAcc(NIP1: string);
begin
    Sqlcmd:= 'Delete T_UserAcc Where NIP = '+ Quotedstr(NIP1) +";
    FDatabase.ADOConn.BeginTrans;
    FDatabase.ADOConn.Execute(Sqlcmd);
    FDatabase.ADOConn.CommitTrans;
end;

function T_UserAcc.Find_UserID(NIP: string): Boolean;
begin
    result:= False;
    Sqlcmd:= 'Select * From T_UserAcc Where NIP = '+ Quotedstr(NIP) + ";

```

```

FDatabase.search.Close;
FDatabase.search.SQL.Clear;
FDatabase.search.SQL.Add(sqlcmd);
FDatabase.search.Open;
if not (FDatabase.search.IsEmpty) then
begin
    result:= True;
    Hak_Akses:= FDatabase.Search.Fieldbyname('Hak_Akses').AsString;
end;
end;

function T_UserAcc.Find_UserID1(NIP1, Passwd1: String): Boolean;
begin
    Result:= False;
    Sqlcmd:= 'Select NIP ,PASSWORD,Hak_Akses From T_USERACC Where NIP = '+
Quotedstr(NIP1) +' and Passwd = '+ Quotedstr(Passwd1)+'";
    FDatabase.search.Close;
    FDatabase.search.SQL.Clear;
    FDatabase.search.SQL.Add(sqlcmd);
    FDatabase.search.Open;
    if not (FDatabase.search.IsEmpty) then
begin
    Hak_Akses:=FDatabase.search.FieldName('HAK_AKSES').AsString;
    Result:= True;
End;
end;

procedure T_UserAcc.Save_UserAcc(NIP1, Passwd1, Hak_Akses1: string);
begin
    Sqlcmd:= 'INSERT INTO T_USERACC VALUES('+ Quotedstr(NIP1) +','+
Quotedstr(PASSWD1) +',' +
"+ HAK_AKSES1 +')';
    FDatabase.ADOConn.BeginTrans;
    FDatabase.ADOConn.Execute(Sqlcmd);
    FDatabase.ADOConn.CommitTrans;
end;

procedure T_UserAcc.Update_Password(NIP1, Passwd1: string);
begin
    Sqlcmd:= 'UPDATE T_USERACC SET PASSWORD= '+ QUOTEDSTR(Passwd1) +' ' +
'WHERE NIP= '+ Quotedstr(NIP1) +";
    FDatabase.ADOConn.BeginTrans;
    FDatabase.ADOConn.Execute(Sqlcmd);
    FDatabase.ADOConn.CommitTrans;
end;

procedure T_UserAcc.Update_UserAcc(NIP1, Passwd1, Hak_Akses1: string);
begin
    Sqlcmd:= 'UPDATE T_USERACC SET PASSWORD= '+ QUOTEDSTR(PASSWD) +',' +
'HAK_AKSES= '+ HAK_AKSES1 +' ' +

```

```

    'WHERE NIP= '+ Quotedstr(NIP1) +";
FDatabase.ADOConn.BeginTrans;
FDatabase.ADOConn.Execute(Sqlcmd);
FDatabase.ADOConn.CommitTrans;
end;
end.

```

Form Karyawan

```

procedure TFKaryawan.TxtNIPKeyPress(Sender: TObject; var Key: Char);
begin
    if (length(txtNIP.Text))=10 then
        begin
            if key=#13 then
                begin
                    Karyawan.NIP:=TxtNIP.Text;
                    if Karyawan1.Find_NIP(Karyawan.NIP) then
                        begin
                            TxtNama.Text:=Karyawan1.NAMA;
                            TxtAlamat.Text:=Karyawan1.ALAMAT;
                            TxtNoTelp.Text:=Karyawan1.NO_TELP;
                            if Karyawan1.JENIS_KELAMIN then
                                CJKel.ItemIndex:=1
                            else
                                CJKel.ItemIndex:=0;
                            CJabatan.ItemIndex:=Karyawan1.JABATAN;
                            if Karyawan1.STATUS then
                                CStatus.ItemIndex:=1
                            else
                                CStatus.ItemIndex:=0;
                            cmdsave.Caption:='&Update';
                        end;
                    TxtNIP.Enabled:=False;
                    TxtNama.Enabled:=True;
                    TxtAlamat.Enabled:=True;
                    TxtNoTelp.Enabled:=True;
                    CJKel.Enabled:=True;
                    CJabatan.Enabled:=True;
                    CStatus.Enabled:=True;
                    CmdSave.Enabled:=True;
                end;
            end;
        end;
end;

```

Form Kapasitas

```

procedure TFSettingKapasitas.CmdSaveClick(Sender: TObject);
begin
    With Kapasitas do
        begin
            NO_KAPASITAS:= TxtNoKapasitas.Text;
            TGL_SETTING:=Formatdatetime('yyyy/MM/dd',strtodate(TxtTglSetting.Text));
        end;
    end;
end;

```

```
KAPASITAS_MOBIL:=TxtKapasitasMobil.Text;
KAPASITAS_MOTOR:=TxtKapasitasMotor.Text;
if cmdsave.Caption='&Save' Then
```

```
KAPASITAS1.Save_Data(NO_KAPASITAS,TGL_SETTING,KAPASITAS_MOBIL,KAPASITAS_MOTOR)
else
KAPASITAS1.Update_Data(NO_KAPASITAS,TGL_SETTING,KAPASITAS_MOBIL,KAPASITAS_MOTOR);
setting_Awal;
end;
end;
```

Form Tarif

```
procedure TFSettingTarif.Laporan_Tarif>Nama_Grid: TDBGrid);
Var
Sqlcmd:String;
begin
Sqlcmd:='Select NO_TARIF,TGL_SETTING,TARIF_MOBIL,TARIF_MOTOR From
T_Tarif Order by NO_TARIF DESC';
with FDatabase.Query do
begin
Close;
SQL.clear;
SQL.Add(sqlcmd);
Open;
end;
Nama_grid.DataSource:=FDatabase.DataSource1;
Nama_grid.Columns[0].Title.caption:='No Tarif';
Nama_grid.Columns[1].Title.caption:='Tgl Setting';
Nama_grid.Columns[2].Title.caption:='Tarif Mobil';
Nama_grid.Columns[3].Title.caption:='Tarif Motor';
Nama_grid.Columns[0].Width:=100;
Nama_grid.Columns[1].Width:=100;
Nama_grid.Columns[2].Width:=100;
Nama_grid.Columns[3].Width:=100;
end;
```

Form Member

```
procedure TFMember.Setting_Awal;
begin
TxtIDMember.Text:=Member1.Get_ID_Member;
TxtNama.Text:="";
TxtAlamat.Text:="";
TxtNoTelp.Text:="";
TxtNoPol.Text:="";
CJenisKend.Clear;
CJenisKend.AddItem('MOBIL',self);
CJenisKend.AddItem('MOTOR',self);
CJenisKend.ItemIndex:=0;
```

```

TxtMerk.Text:="";
TxtWarna.Text:="";
cmdsave.Caption:='&Save';
cmdsave.Enabled:=True;
TxtIDMember.Enabled:=False;
TxtNama.Enabled:=True;
TxtAlamat.Enabled:=True;
TxtNoTelp.Enabled:=True;
TxtNoPol.Enabled:=True;
CJenisKend.Enabled:=True;
TxtMerk.Text:="";
TxtWarna.Text:="";
TxtNama.SetFocus;
Laporan_Member(Lap_Member);
end;

```

Form Aktif Member

```

procedure TFAktifMember.CmdSaveClick(Sender: TObject);

```

```

begin
  with AktifMember do
    begin
      NO_AKTIF:= TxtNoAktif.Text;
      TGL_AKTIF:= Formatdatetime('yyyy/MM/dd',strtodate(TxtTgl.Text));
      ID_MEMBER:= TxtIDMember.Text;

```

```

AktifMember1.Save_Data(NO_AKTIF,TGL_AKTIF,ID_MEMBER,NO_TARIF_MEMBER
);

```

```

end;
with Pendapatan do
begin
  NO_PENDAPATAN:=Pendapatan1.Get_No_Pendapatan;
  TGL_PENDAPATAN:=AktifMember.TGL_AKTIF;
  TOTAL_PARKIR_MEMBER:=TxtTotalHarga.Text;
  if Pendapatan1.Find_No_Pendapatan(NO_PENDAPATAN) then

```

```

Pendapatan1.Update_Pendapatan(NO_PENDAPATAN,TGL_PENDAPATAN,'0',TOTAL_P
ARKIR_MEMBER)

```

```

  else
  Pendapatan1.Save_Pendapatan(NO_PENDAPATAN,TGL_PENDAPATAN,'0',TOTAL_PA
RKIR_MEMBER);

```

```

  end;
  setting_Awal;
end;

```

Form Pengeluaran

```

procedure TFPengeluaran.TxtKeteranganKeyPress(Sender: TObject;

```

```

  var Key: Char);
begin
  IF Key=#13 then
    begin

```



```

if (trim(TxtKeterangan.Text)="") and (No > 0) then
begin
  cmddelete.Enabled:=false;
  TxtKeterangan.Enabled:=false;
  cmdsave.Enabled:=True;
  cmdsave.SetFocus;
end
else
begin
  Item_Pengeluaran.Keterangan:=TxtKeterangan.Text;
  if not Find_Pengeluaran_InTable(Item_Pengeluaran.Keterangan) then
  begin
    inc(No);
    TxtTotal.Text:=Floattostr(Hitung_Total);
    IF No > TabelPengeluaran.RowCount-1 then
TabelPengeluaran.RowCount:=TabelPengeluaran.RowCount+1;
    Insert_Pengeluaran;
    TxtKeterangan.Enabled:=false;
    TxtBiaya.Enabled:=True;
    TxtBiaya.SetFocus;
  end
  else
  begin
    Txtbiaya.Text:=TabelPengeluaran.Cells[2,No];
    TabelPengeluaran.Cells[2,No]:="";
    TxtTotal.Text:=Floattostr(Hitung_Total);
    TxtBiaya.Enabled:=True;
    TxtBiaya.SetFocus;
  end;
end;
end;
end;
end;

```

Form Setting User Acc

```

procedure TFUserAcc.TxtNIPKeyPress(Sender: TObject; var Key: Char);
begin
  If Key= #13 Then
  begin
    UserAcc.NIP:= TxtNIP.Text;
    If Karyawan1.Find_NIP(UserAcc.NIP) Then
    begin
      if Karyawan1.STATUS then
      begin
        TxtNama.Text:= karyawan1.NAMA;
        If useracc1.Find_UserID(UserAcc.NIP) Then
        begin
          CHakAkses.ItemIndex:= strtoint(UserAcc.Hak_Akses);
          cmdSave.Caption:= 'UPDATE';
          cmddelete.Enabled:= True;
        end;
      end;
    end;
  end;
end;

```

```

        TxtNIP.Enabled:= False;
        cmdsave.Enabled:= True;
    end;
End
else
    begin
        MessageDlg('NIP Belum Terdaftar Atau Salah', mtWarning,[mbOk], 0);
    end;
End;
end;

```

Form Password

```

procedure TFUpdatePasswd.TxtPasswdKeyPress(Sender: TObject; var Key: Char);
begin
    If Key = #13 Then
        begin
            UserAcc.NIP:=TxtNIP.Text;
            UserAcc.Passwd:=TxtPasswd.Text;
            UserAcc.NIP:= TxtNIP.Text;
            If UserAcc1.Find_UserID1(UserAcc.NIP,UserAcc.Passwd) Then
                begin
                    TxtNIP.Enabled:= False;
                    TxtPasswd.Enabled:= False;
                    TxtNewPasswd.Enabled:= True;
                    TxtNewPasswd.SetFocus;
                end
            Else
                ShowMessage('NIP or PASSWORD SALAH!!!');
            key:=#0;
        end;
end;

```

Form Login

```

procedure TFLogin.CmdLoginClick(Sender: TObject);
begin
    FMenuutama.Setting_Awal;
    If UserAcc1.Find_UserID1(TxtNIP.Text, Txtpasswd.Text) Then
        begin
            FmENUUTAMA.Show;
            with FMenuUtama do
                begin
                    UserAcc.Hak_Akses:=strtoint(UserAcc1.Hak_Akses);
                    FMenuUtama.No_Karyawan_Login:=txtNIP.Text;
                    menu5.Enabled:=true;
                    HakAkses2.ENABLED:= True;
                    HakAkses3.ENABLED:= True;
                    case UserAcc.Hak_Akses of
                        0: begin
                            laporan1.Enabled:=true;
                            laporan2.Enabled:=true;

```

```

    laporan3.Enabled:=true;
    laporan4.Enabled:=true;
    laporan5.Enabled:=true;
    laporan6.Enabled:=true;
    laporan7.Enabled:=true;
    laporan8.Enabled:=true;
    SpeedButton2.Enabled:=False;
    SpeedButton3.Enabled:=False;
    SpeedButton4.Enabled:=False;
end;
1: Begin
    Transaksi1.Enabled:=True;
    parkirMasuk1.Enabled:=True;
    ParkirMasuk2.Enabled:=True;
    transaksi2.Enabled:=True;
    parkirkeluar1.Enabled:=True;
    Parkirkeluar2.Enabled:=True;
end;
2: begin;
    master4.Enabled:=true;
    admin1.Enabled:=true;
    admin2.Enabled:=true;
    laporan2.Enabled:=true;
    laporan3.Enabled:=true;
    laporan4.Enabled:=true;
    laporan5.Enabled:=true;
    laporan6.Enabled:=true;
    laporan7.Enabled:=true;
    laporan8.Enabled:=true;
    hakakses1.Enabled:=true;
end;
3: begin
    master1.Enabled:=true;
    master2.Enabled:=true;
    master3.Enabled:=true;
    tarif1.Enabled:=true;
    tarif2.Enabled:=true;
    //Menu2.Enabled:=False;
    //Menu3.Enabled:=False;
    menu4.Enabled:=true;
    laporan1.Enabled:=true;
    laporan6.Enabled:=true;
    laporan7.Enabled:=true;
    laporan8.Enabled:=true;
    //SpeedButton1.Enabled:=True;
    SpeedButton2.Enabled:=False;
    SpeedButton3.Enabled:=False;
    SpeedButton4.Enabled:=False;
    //SpeedButton5.Enabled:=True;
end

```