



**SISTEM INFORMASI PERPARKIRAN KENDARAAN BERMOTOR
RODA DUA DI PROGRAM STUDI SISTEM INFORMASI UNIVERSITAS
JEMBER MENGGUNAKAN TEKNOLOGI RFID (*Radio Frequency
Identification*)**

SKRIPSI

Oleh

Anwar Syarifuddin Fajri

NIM 102410101001

PROGRAM STUDI SISTEM INFORMASI

UNIVERSITAS JEMBER

2015



**SISTEM INFORMASI PERPARKIRAN KENDARAAN BERMOTOR
RODA DUA DI PROGRAM STUDI SISTEM INFORMASI UNIVERSITAS
JEMBER MENGGUNAKAN TEKNOLOGI RFID (*Radio Frequency
Identification*)**

SKRIPSI

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat
untuk menyelesaikan Program Studi Sistem Informasi (S1)
dan mencapai gelar Sarjana Sistem Informasi

Oleh

Anwar Syarifuddin Fajri

NIM 102410101001

**PROGRAM STUDI SISTEM INFORMASI
UNIVERSITAS JEMBER**

2015

PERSEMBAHAN

Puji syukur terhadap kehadiran ALLAH SWT atas segala nikmat hidup dan kesempatan untuk menimba ilmu. Sehingga saya dapat menyelesaikan skripsi yang berjudul “SISTEM INFORMASI PERPARKIRAN KENDARAAN BERMOTOR RODA DUA DI PROGRAM STUDI SISTEM INFORMASI UNIVERSITAS JEMBER MENGGUNAKAN TEKNOLOGI RFID (*Radio Frequency Identification*)”. Skripsi ini disusun sebagai salah satu persyaratan untuk mencapai gelar sarjana di Program Studi Informasi.

Skripsi ini saya persembahkan kepada kedua orang tua saya terima kasih atas seluruh limpahan kasih sayang selama ini dan yang tidak pernah lelah memberikan motivasi dan juga yang telah memberikan kepercayaan penuh kepada saya selama proses pengerjaan dan dukungan (dana atau moril yang baik). Tanpa adanya beliau mungkin skripsi ini tidak akan selesai.

Skripsi ini saya dedikasikan kepada diri saya sendiri yang telah banyak meluangkan waktu bermain demi menyelesaikan skripsi ini. Karena hampir setiap hari selama beberapa bulan ini saya habiskan dengan mempelajari banyak referensi – referensi.

Skripsi ini saya persembahkan juga kepada dosen-dosen pembimbing saya yang tidak pernah protes walaupun saya ajak konsultasi berkali-kali, dan selalu memotivasi agar segera lulus dengan harapan yang baik-baik. Terima kasih dosen-dosen pembimbingku.

Skripsi ini saya persembahkan juga kepada orang-orang terdekat saya. Pacar, sahabat, dan rekan. Pacar yang jarang memotivasi dan malah sering mengajak jalan-jalan, sungguh pacar yang baik ☺. Sahabat-sahabat baik yang sudah lulus maupun belum lulus, kalian luar biasa.

Skripsi ini juga saya persembahkan kepada teman – teman Sistem Informasi 2010 “ZerOne” senasib, seperjuangan dan sepenanggungan, terima kasih atas gelak tawa dan solidaritas yang luar biasa sehingga membuat hari-hari semasa kuliah menjadi lebih berarti. Semoga tak ada lagi duka di dada tapi suka dan bahagia juga tawa dan canda dan semoga tali silaturahmi kita tetap terjaga.

Skripsi ini juga saya persembahkan kepada taretan – taretan JONG MADURA (Paguyuban Mahasiswa Madura Di Jember) yang selalu menemani saya selama ada di jember dan juga selalu memberi semangat dan jalan kepada saya untuk menjadi seorang yang lebih maju dan mandiri.

Semoga Allah SWT membalas jasa dan budi kalian semua dikemudian hari dan memberikan kemudahan dalam segala urusan,, aaammiinn ☺



MOTO

“ Sesungguhnya bersama kesulitan ada kemudahan. Maka apabila engkau telah selesai (dari suatu urusan), tetaplah bekerja keras (untuk urusan lain), dan hanya kepada Tuhanmulah engkau berharap”

(QS. Al-Insyirah : 6-8)*)

“Sebaik-baik kalian islamnya adalah yang paling baik akhlaq jika mereka menuntut ilmu.” (HR. Ahmad)**)

Man Jadda Wa Jadda, Man Shabara Zhafira

(Barang siapa bersungguh-sungguh akan sukses dan barang siapa bersabar akan beruntung.)***)

*) Departemen Agama Republik Indonesia. 1984. *Al Qur'an dan Terjemahannya*. Semarang: CV.Asy-Syifa'.

***) Zainuddin, A. 1430 H. *Apakah Anda Termasuk Sebaik-baik Manusia?*. Dammam KSA: IslamicCultural Center.

****) Fuadi, A. 2010. *Ranah Tiga Warna*. Jakarta: PT. Gramedia.

PERNYATAAN

Saya yang bertanda tangan dibawah ini :

Nama : Anwar Syarifuddin Fajri

NIM : 102410101001

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Sistem Informasi Perparkiran Kendaraan Bermotor Roda Dua di Program Studi Sistem Informasi Universitas Jember Menggunakan Teknologi RFID (*Radio Frequency Identification*)” adalah benar-benar karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada institusi mana pun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa ada tekanan dan paksaan dari pihak mana pun serta bersedia mendapat sanksi akademik jika ternyata dikemudian hari pernyataan ini tidak benar.

Jember,

Yang menyatakan,

Anwar Syarifuddin Fajri

NIM 102410101001

SKRIPSI

**SISTEM INFORMASI PERPARKIRAN KENDARAAN BERMOTOR
RODA DUA DI PROGRAM STUDI SISTEM INFORMASI UNIVERSITAS
JEMBER MENGGUNAKAN TEKNOLOGI RFID (*Radio Frequency
Identification*)**

Oleh

Anwar Syarifuddin Fajri

NIM 102410101001

Menyetujui

Dosen Pembimbing Utama

Dosen Pembimbing Pendamping

Prof. Drs. Slamin, M.Comp.Sc., Ph.D
NIP. 196704201992011001

Yanuar Nurdiansyah ST, M.Cs.
NIP. 198201012010121004

PENGESAHAN

Skripsi berjudul “Sistem Informasi Perparkiran Kendaraan Bermotor Roda Dua di Program Studi Sistem Informasi Universitas Jember Menggunakan Teknologi RFID (*Radio Frequency Identification*)” telah diuji dan disahkan pada :

Hari, tanggal :

Tempat : Program Studi Sistem Informasi.

Tim Penguji :

Dosen Penguji I

Dosen Penguji II

Dr. Saiful Bukhori, ST., M.Kom
NIP. 196811131994121001

Nelly Oktavia A, S.Si., MT
NIP. 198410242009122008

Mengesahkan

Dekan

Prof. Drs. Slamini, M.Comp.Sc., Ph.D
NIP. 196704201992011001

Anwar Syarifuddin Fajri

Program Studi Sistem Informasi, Universitas Jember

ABSTRAK

Sistem perparkiran kendaraan roda dua pada saat ini khususnya di area parkir Universitas Jember banyak menggunakan sistem perparkiran manual yaitu dengan cara melihat Nomor kendaraan dengan STNK setiap kendaraan yang hendak keluar dari area parkir dan sistem tersebut banyak memiliki kelemahan terutama dari segi keamanan dan efektifitas waktu yang digunakan. Dan pada saat ini kemajuan teknologi sangat pesat, salah satunya adalah RFID. RFID adalah sebuah metode identifikasi dengan menggunakan sarana yang disebut dengan label RFID atau transponder (tag) untuk menyimpan dan mengambil data jarak jauh. Oleh sebab itu sistem parkir kendaraan roda dua di area parkir Program Study Sistem informasi memilih menggunakan sistem Parkiran menggunakan teknologi RFID karena memiliki beberapa keunggulan dan mampu mengatasi kelemahan sistem parkir yang ada sebelumnya. Tujuan penelitian ini adalah membuat suatu sistem perparkiran kendaraan roda dua yang nantinya secara otomatis dapat mendeteksi semua data pengguna area parkir baik data kendaraan dan juga data pemilik kendaraan. Hasil dari penelitian ini adalah suatu sistem parkir kendaraan roda dua yang nantinya mampu menyimpan data pemilik dan kendaraan secara otomatis ke dalam komputer dengan menggunakan RFID dan Smart card sebagai readernya.

Keyword : Sistem Parkiran Roda Dua, RFID, Smart card

RINGKASAN

SISTEM INFORMASI PERPARKIRAN KENDARAAN BERMOTOR RODA DUA di PROGRAM STUDI SISTEM INFORMASI UNIVERSITAS JEMBER MENGGUNAKAN TEKNOLOGI RFID (*Radio Frequency Identification*). Anwar Syarifuddin Fajri; 102410101001; 2015; 96 Halaman; Program Studi Informasi Universitas Jember.

Tempat parkir merupakan salah satu komponen penting dalam memberikan suatu kenyamanan dan ketenangan kepada pengguna fasilitas umum karena parkir adalah suatu keadaan tidak bergerak suatu kendaraan yang bersifat sementara (Direktur Jenderal Perhubungan Darat, 1996). Keamanan, kemudahan dan kenyamanan adalah faktor utama yang diharapkan oleh pengguna fasilitas parkir tersebut.

Semakin berkembangnya teknologi saat ini segala sesuatu yang awalnya manual sekarang diganti menjadi otomatis termasuk di antaranya adalah parkir. Salah satu fasilitas parkir yang masih menggunakan cara manual adalah fasilitas perparkiran di Program Studi Informasi Universitas Negeri Jember. Sistem yang diterapkan saat ini adalah dengan cara pengecekan satu persatu nomor kendaraan dengan menggunakan STNK (Surat Tanda Nomor Kendaraan) . Cara tersebut sangat tidak efisien karena diharuskan petugas melakukan pengecekan satu persatu setiap kendaraan yang keluar – masuk area parkir program Studi Informasi Universitas Jember selain itu dengan metode seperti itu sering kali terjadi antrian kendaraan yang cukup panjang padahal kendaraan yang sering keluar – masuk program Studi Informasi Universitas Jember merupakan suatukendaraan yang sama yaitu karyawan dan mahasiswa (hasil studi lapangan di area kampus Program Studi Informasi).

Teknologi yang dapat menjawab dari segala macam persoalan yang ada dan menjawab dari kelemahan metode – metode di atas adalah menggunakan teknologi RFID.RFID adalah sebuah metode identifikasi dengan menggunakan sarana yang disebut dengan label RFID atau transponder (tag) untuk menyimpan dan mengambil data jarak jauh.

Sistem parkir yang menggunakan teknologi RFID, petugas tidak perlu lagi mengecek satu persatu secara berulang ulang setiap kendaraan yang keluar masuk karena sudah dilakukan otomasi oleh komputer. Keuntungan menggunakan RFID daripada metode lain yaitu memungkinkan data dapat dibaca secara otomatis tanpa memperhatikan garis arah bacaan, melewati bahan *non-conductor* seperti buku, majalah, naskah dan barang lainnya dengan kecepatan akses beberapa ratus *tag* setiap detik pada jarak ± 100 meter. Menurut Want (2004) RFID bukan hanya sekedar pengganti barcode. RFID juga memberikan berbagai macam fitur yang tidak diberikan barcode seperti keamanan, cara pembacaan yang tidak harus langsung. Want mengatakan bahwa nantinya RFID akan di gunakan secara global sebagai metode identifikasi otomatis.

Sistem informasi parkir menggunakan teknologi RFID ini di buat untuk mempermudah, mempercepat dan memberikan kenyamanan serta keamanan bagi pemilik kendaraan yang parkir di area parkir Program Studi Informasi Universitas Jember dan petugas parkir itu sendiri karena semuanya bersifat otomatis sehingga petugas parkir tidak lagi mengecek kendaraan yang masuk ataupun keluar secara satu per satu.

Hasil dari penelitian disimpulkan bahwa sistem parkir menggunakan teknologi RFID sangat tepat digunakan untuk menggantikan sistem parkir sebelumnya yang bersifat manual karena banyak kelebihan yang terdapat pada RFID.

PRAKATA

Puji syukur ke hadirat Allah SWT. atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Sistem Informasi Perparkiran Kendaraan Bermotor Roda Dua di Program Studi Sistem Informasi Universitas Jember Menggunakan Teknologi RFID (*Radio Frequency Identification*)”.

Penyusunan skripsi ini tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada :

1. Kedua orang tua saya Ibu/Bapak Aminatus Subaidah / Fathorrozi S.Ag sekeluarga yang telah memberikan dukungan penuh fasilitas, materi, dan doa demi terselesaikannya skripsi ini;
2. Prof. Drs. Slammin, M.Comp.Sc., Ph.D dan Bapak Yanuar Nurdiansyah S.T., M.Cs., selaku Dosen Pembimbing Utama dan Dosen Pembimbing Anggota yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi ini;
3. Novidha Satya Ningtyas yang selalu memberikan semangat dan meluangkan banyak waktunya;
4. Kakakku Mohammad Fahmi Rosyidi dan Adekku Nuril LailyAlfiani;
5. Sahabat dan semua rekan seperjuangan ZerOne PSSI Universitas Jember;
6. Taretan Jong Madura
7. Dan semua pihak yang tidak dapat disebutkan satu per satu.

Penulis juga menerima segala kritik dan saran dari semua pihak demi kesempurnaan skripsi ini. Akhirnya penulis berharap, semoga skripsi ini dapat bermanfaat.

Jember, Agustus 2015

Penulis

DAFTAR ISI

	Halaman
HALAMAN JUDUL	ii
HALAMAN PERSEMBAHAN	iii
HALAMAN MOTO	v
HALAMAN PERNYATAAN	vi
HALAMAN PEMBIMBINGAN	vii
HALAMAN PENGESAHAN	viii
ABSTRAK	ix
RINGKASAN	x
PRAKATA	xii
DAFTAR ISI	xiii
DAFTAR GAMBAR	xvi
DAFTAR TABEL	xvii
DAFTAR LAMPIRAN	xviii
BAB 1. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	4
1.3 Tujuan Penelitian	4
1.4 Batasan Masalah	4
BAB 2. TINJAUAN PUSTAKA	5
2.1 RFID (<i>Radio Frequency Identification</i>)	5
2.2 Sejarah RFID	6
2.3 Cara kerja RFID	7
2.4 Perbedaan RFID dengan Barcode Optik	7
2.5 Tag RFID	9
2.6 Reader RFID	10
2.7 Frekuensi RFID	11
2.8 Kategori Sistem RFID	12
2.9 <i>Smartcard</i>	13

2.10	Tipe-tipe <i>Smartcard</i>	14
2.11	Cara Kerja <i>Smartcard</i>	16
2.12	Komponen <i>Smartcard</i>	17
BAB 3.	METODOLOGI PENELITIAN	21
3.1	Jenis penelitian	21
3.2	Tahapan penelitian	21
3.3	Objek Penelitian	21
3.4	Teknik Pengumpulan Data	22
3.5	Tahap Perancangan	22
3.6	Implementasi	25
3.7	Perawatan (<i>maintenance</i>)	26
BAB 4.	PERANCANGAN SISTEM PERPARKIRAN	
	KENDARAAN	27
4.1	Deskripsi umum	27
4.2	Analisis Kelemahan dan Kekurangan Sistem Perparkiran Awal	27
4.3	Perancangan Sistem Perparkiran Kendaraan Roda Dua Baru	28
4.4	<i>Requiere ment Spesification</i>	29
4.4.1	Kebutuhan Fungsional	29
4.4.2	Kebutuhan Non Fungsional	29
4.5	Perancangan Sistem	30
4.5.1	<i>WorkFlow</i>	30
4.5.2	<i>Business Process</i>	32
4.5.3	Data Flow Diagram (DFD).....	33
4.5.4	Kamus Data	37
4.5.5	<i>Entity Relationship Diagram (ERD)</i>	38
4.6.	Pengujian Sistem	39
4.6.1	Pengujian White Box	39
4.6.2	Pengujian Black Box	43

BAB 5. HASIL DAN PEMBAHASAN	44
5.1 Implementasi Sistem	44
5.2 Hasil Sistem yang telah dibuat	44
5.2.1 Data Pengunjung	45
5.2.2 Daftar Baru	51
5.2.3 Rekap Pengunjung	63
BAB 6. PENUTUP	68
6.1 Kesimpulan	68
6.2 Saran	68
DAFTAR PUSTAKA	69
LAMPIRAN	71

DAFTAR GAMBAR

	Halaman
2.1 Kartu reader acces RFID	6
2.2 Komponen sistem kerja RFID	7
2.3 <i>Tag</i> RFID	9
2.4 Bentuk <i>smartcard</i>	14
2.5 Delapan titik kontak <i>smartcard</i>	15
2.6 Komponen <i>smartcard</i>	17
3.1 <i>Flow chart</i> tahapan penelitian	21
3.2 Tahapan model <i>waterfall</i>	23
4.1 <i>Workflow</i> input data pengunjung sistem parkir menggunakan teknologi RFID.....	30
4.2 <i>Workflow</i> Daftar baru	31
4.3 <i>Workflow</i> rekap data pengunjung	32
4.4 <i>Business process</i> Sistem Parkiran Kendaraan Bermotor roda dua	33
4.5 DFD level 0.....	34
4.6 DFD level 1	35
4.7 DFD level 2.....	36
4.8 <i>Entity Relationship Diagram</i> Sitem Parkiran Kendaraan Bermotor roda dua	38
4.9 Listing program data parkir	42
4.10 Grafik alir function	42
5.1 Screenshot halaman data pengunjung parkir	45
5.2 Screenshot halaman data pengunjung	51
5.3 <i>Screenshot</i> halaman daftar baru	52
5.4 Halaman daftar baru pengguna	53
5.5 <i>Screenshot</i> halaman ubah dan hapus pengguna	54
5.6 <i>Screenshot</i> halaman rekap pengunjung	63

DAFTAR TABEL

	Halaman
2.1 Perbedaan RFID dengan Barcode	8
2.2 Perbandingan <i>Tag</i> aktif dan <i>Tag</i> Pasif	10
4.1 Kamus data pengunjung	37
5.1 <i>Source code</i> data pengunjung	46
5.2 <i>Listing</i> Program Daftar Baru	54
5.3 <i>Source code</i> Rekap Pengunjung	63



DAFTAR LAMPIRAN

	Halaman
A. <i>Screenshot</i> Program	71
A1. <i>User Interface</i> Data Pengunjung	71
A2. <i>User interface</i> Daftar Baru	73
A3. Rekap Pengunjung	76
B. Pengujian Sistem.....	77
B1. Pengujian Sistem	77
B2. Pengujian Pada Menu Data Parkir keluar dan Masuk	78
B3. Pengujian pada Refresh Table	80
B4. Pengujian Pada Tombol Tambah user	82
B5. Pengujian Pada tombol Edit User	83
B6. Pengujian Pada tombol Delete User	85
B7. Pengujian Pada tombol Reset pada fitur Daftar Baru User	86
B8. Pengujian Pada Daftar Baru mengisi form Table	87
B9. Pengujian Pada Fitur Daftar Baru pada tombol Cari	89
B10. Pengujian Pada Fitur Rekap Data pada tombol Cari	91
B11. Pengujian Refresh Table1	93
B12. Pengujian Refresh Table2	95

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Era globalisasi saat ini teknologi berkembang sangat cepat. Dalam kehidupan sehari-hari teknologi sering kali digunakan untuk mempermudah pekerjaan seseorang karena dengan adanya teknologi semua pekerjaan yang dulunya manual menjadi serba otomatis sehingga dapat mempermudah dan membantu manusia dalam melakukan suatu pekerjaan. Dengan adanya teknologi segala macam pekerjaan manusia menjadi lebih mudah dan cepat. Teknologi merupakan kumpulan alat, aturan dan prosedur yang merupakan penerapan pengetahuan ilmiah terhadap suatu pekerjaan tertentu dalam cara yang memungkinkan pengulangan (Capra, 2004). Menurut kamus besar Bahasa Indonesia (1990) definisi dari teknologi adalah metode ilmiah untuk mencapai tujuan praktis ilmu pengetahuan terapan.

Tempat parkir merupakan salah satu komponen penting dalam memberikan suatu kenyamanan dan ketenangan kepada pengguna fasilitas umum karena parkir adalah suatu keadaan tidak bergerak suatu kendaraan yang bersifat sementara (Direktur Jenderal Perhubungan Darat, 1996). Keamanan, kemudahan dan kenyamanan adalah faktor utama yang diharapkan oleh pengguna fasilitas parkir tersebut.

Semakin berkembangnya teknologi saat ini segala sesuatu yang awalnya manual sekarang diganti menjadi otomatis termasuk di antaranya adalah parkir. Salah satu fasilitas parkir yang masih menggunakan cara manual adalah fasilitas parkir di Program Studi Informasi Universitas Jember. Sistem yang diterapkan saat ini adalah dengan cara pengecekan satu persatu nomor kendaraan dengan menggunakan STNK (Surat Tanda Nomor Kendaraan). Cara tersebut sangat tidak efisien karena diharuskan petugas melakukan pengecekan satu persatu setiap kendaraan yang keluar – masuk area parkir program Studi Informasi Universitas Jember selain itu dengan metode seperti itu sering kali terjadi antrian kendaraan yang cukup panjang padahal kendaraan yang sering keluar – masuk program Studi Informasi Universitas Jember merupakan suatu

kendaraan yang sama yaitu karyawan dan mahasiswa (hasil studi lapangan di area kampus Program Studi Informasi).

Alternatif yang dapat dipilih untuk menggantikan metode parkir yang secara manual atau yang dengan cara pengecekan secara satu persatu nomor kendaraan menggunakan STNK seperti yang diterapkan di Program Studi Sistem Informasi tersebut diantaranya adalah metode karcis yang di cetak, metode stiker, kartu pintar (smart card) berupa barcode dan kartu elektronik RFID. Metode karcis yang di cetak memiliki kelemahan yaitu petugas harus melakukan pengetikan berulang – ulang setiap ada kendaraan yang akan keluar masuk sistem sementara kendaraan yang keluar masuk sistem pada umumnya adalah kendaraan yang sama yang melakukan aktifitas setiap hari yaitu mahasiswa dan karyawan. Selain itu metode ini tidak dapat merekam informasi pengguna sehingga bila terjadi sesuatu kasus pencurian tidak dapat segera ditangani. Oleh karena itu metode ini tidak dipilih. Aturan metode parkir menggunakan stiker mengharuskan setiap kendaraan yang akan memasuki sistem harus memiliki stiker. Metode stiker memiliki kelemahan yaitu kendaraan yang keluar masuk sistem adalah kendaraan yang tetap. Bila ada penambahan kendaraan serta terjadi kasus pembajakan kendaraan seperti saat Penerimaan Mahasiswa Baru akan terjadi kesulitan dalam menangani kendaraan yang keluar masuk. Selain itu stiker sangat mudah untuk ditiru dan mudah terlepas sehingga dapat di manfaatkan oleh pihak yang tidak bertanggung jawab sebagai tindak kriminal. Metode kartu barcode juga memiliki kelemahan yaitu hanya mampu membaca label dengan laser optik atau teknologi image dan juga membutuhkan kepresian yang sangat tinggi dan harus terjadi kontak langsung antara kartu dengan pembaca (Maryono, 2005).

Teknologi yang dapat menjawab dari segala macam persoalan yang ada dan menjawab dari kelemahan metode – metode di atas adalah menggunakan teknologi RFID. Sejarah perkembangan RFID dimulai sejak tahun 1920, tetapi berkembang menjadi IFF *transponder* pada tahun 1939 yang waktu itu berfungsi sebagai alat identifikasi pesawat musuh, dipakai oleh militer Inggris pada perang dunia II (Henia, 2006:1). RFID adalah sebuah metode identifikasi dengan menggunakan sarana yang disebut dengan label RFID atau transponder (tag)

untuk menyimpan dan mengambil data jarak jauh. Label atau transponder (tag) adalah sebuah benda yang bisa dipasang atau dimasukkan di dalam sebuah produk, hewan atau bahkan manusia dengan tujuan untuk identifikasi menggunakan gelombang radio (Maryono, 2005). Menurut Want (2004) RFID bukan hanya sekedar pengganti barcode. RFID juga memberikan berbagai macam fitur yang tidak diberikan barcode seperti keamanan, cara pembacaan yang tidak harus langsung. Want mengatakan bahwa nantinya RFID akan di gunakan secara global sebagai metode identifikasi otomatis.

Sistem parkir yang menggunakan teknologi RFID, petugas tidak perlu lagi mengecek satu persatu secara berulang ulang setiap kendaraan yang keluar masuk karena sudah dilakukan otomasi oleh komputer. Keuntungan menggunakan RFID daripada metode lain yaitu memungkinkan data dapat dibaca secara otomatis tanpa memperhatikan garis arah bacaan, melewati bahan *non-conductor* seperti buku, majalah, naskah dan barang lainnya dengan kecepatan akses beberapa ratus *tag* setiap detik pada jarak ± 100 meter. *Tag* RFID terbuat dari *microchip* berbahan dasar silikon yang memiliki kemampuan fungsi identifikasi sederhana yang disatukan dalam satu desain (Supandri, 2004). Selain itu keunggulan dari suatu sistem yang menggunakan teknologi RFID adalah dari segi waktu, biaya, pengawasan, keamanan, dan kenyamanan bagi pengguna fasilitas tersebut.

Berdasarkan ulasan dan pendapat para ahli tentang RFID yang dilakukan oleh peneliti sebelumnya, dapat disimpulkan bahwa penggunaan RFID tersebut cocok diterapkan dalam sistem parkir Program Studi Informasi karena memiliki banyak keunggulan dibandingkan dengan metode – metode lainnya. Selain itu dengan menggunakan teknologi RFID akan meningkatkan rasa kenyamanan dan keamanan bagi para pengguna fasilitas parkir tersebut. Oleh karena itu, penulis akan melakukan penelitian tentang Sistem Informasi Perparkiran Kendaraan Bermotor di Program Studi Informasi Menggunakan Teknologi RFID sehingga pengelolaan parkir menjadi lebih efisien dan aman. Harapannya dengan adanya sistem tersebut akan berdampak positif dan mampu

meningkatkan keamanan dan kenyamanan bagi pengguna fasilitas parkir di Program Studi Informasi Universitas Jember.

1.2 Perumusan Masalah

1. Bagaimana cara merancang dan membangun sistem perparkiran kendaraan jenis roda dua menggunakan teknologi RFID di Program Studi Informasi Universitas Jember?
2. Bagaimana cara memberikan fasilitas parkir parkir dalam hal keamanan kendaraan dan kenyamanan bagi pengguna dan petugas parkir?

1.3 Tujuan Penelitian

1. Merancang dan membangun sistem parkir baru untuk jenis kendaraan roda dua di Program Studi Informasi Universitas Jember menggunakan teknologi RFID (*Radio Frequency Identification*).
2. Sistem informasi perparkiran kendaraan roda dua di Program Studi Sistem Informasi Universitas Jember untuk memberikan keamanan dan kenyamanan bagi pengguna dan petugas parkir.

1.4 Batasan Masalah

Adapun batasan masalah pada sistem informasi parkir ini adalah sebagai berikut:

1. Sistem ini dibuat untuk mengelola kendaraan parkir jenis roda dua di Program Studi Sistem Informasi Universitas Jember.
2. Sistem parkir ini hanya diimplementasikan di Program Studi Sistem Informasi.
3. Sistem ini dijalankan menggunakan *smart card* yang berfungsi sebagai alat identifikasi otomatis yang nantinya dibaca oleh *reader* RFID.

BAB 2. TINJAUAN PUSTAKA

2.1 RFID (Radio Frequency Identification)

RFID adalah suatu teknologi identifikasi berbasis gelombang dimana metode identifikasinya menggunakan sarana yang disebut label RFID atau transponder (tag) untuk menyimpan dan mengambil data jarak jauh. Teknologi ini mampu mengidentifikasi berbagai objek secara simultan tanpa diperlukan kontak langsung (atau dalam jarak pendek) (Finkenzeller, 2003).

Menurut (Kenzeller, 1999) RFID adalah teknologi identifikasi yang fleksibel, mudah digunakan dan sangat cocok untuk operasi otomatis. RFID mengkombinasikan keunggulan yang tidak tersedia pada teknologi identifikasi yang lain. RFID dapat disediakan dalam *device* yang hanya dapat dibaca saja (*Read Only*) atau dapat dibaca dan ditulis (*Read/Write*), tidak memerlukan kontak langsung maupun jalur cahaya untuk dapat beroperasi, dapat berfungsi pada berbagai variasi kondisi lingkungan, dan menyediakan tingkat integritas data yang tinggi. Sebagai tambahan, karena teknologi ini sulit untuk dipalsukan, maka RFID dapat menyediakan tingkat keamanan yang tinggi.

Teknologi RFID merupakan bagian dari RF (Radio Frekuensi) yang digunakan media sebagai media identifikasi secara wireless yang terdiri dari dua komponen (Karigianes, 2007) yaitu

- a. RFID tag yang terdiri dari sebuah device yang kecil yang tertanam dalam sebuah benda seperti label, smartcard dan lainnya yang memiliki identifikasi yang unik dan memori yang dapat di tulis.
- b. RFID reader merupakan sebuah device yang dapat berkomunikasi tanpa kontak langsung dengan suatu tag untuk mengidentifikasikannya apabila terhubung dalam satu asosiasi data (Karigianes, 2007)



Gambar 2.1 Kartu reader acces RFID (Sumber : Karigianes, 2007)

2.2 Sejarah RFID

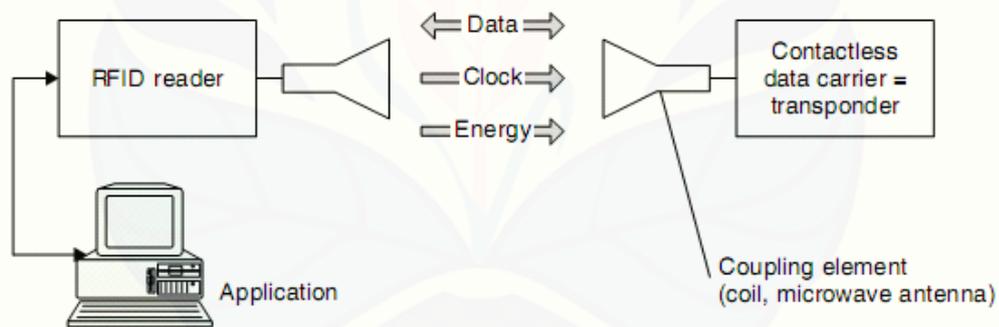
Sejarah perkembangan RFID dimulai sejak tahun 1920, tetapi berkembang menjadi IFF *transponder* pada tahun 1939. Yang waktu itu berfungsi sebagai alat identifikasi pesawat musuh, dipakai oleh militer Inggris pada perang dunia II. Sejak tahun 1945 beberapa orang berfikir bahwa perangkat pertama RFID ditemukan oleh Leon Theremin sebagai suatu *tool spionase* untuk pemerintah rusia (Henlia, 2006).

Alat dari Mario Cardullo adalah nenek moyang pertama dari RFID modern, karena merupakan sebuah *transponder* radio pasif dengan memori. Perangkat pasif pertama yang didukung oleh sinyal yang menginterograsi, dan telah didemonstrasikan pada tahun 1971 di New York Port Authority. Alat ini terdiri dari sebuah *transponder* dengan 16 bit memori yang digunakan sebagai *toll device*. Cardullo mematenkan dasar penggunaan frekuensi radio, suara dan cahaya sebagai transmisi.

2.3 Cara kerja RFID

Pembaca RFID (RFID reader) adalah merupakan penghubung antara software aplikasi dengan antenna yang meradiasikan gelombang radio tag (transponder) RFID (Fangan, 2004)

Identifikasi objek atau data pada teknologi RFID dilakukan dengan cara mencocokkan data yang tersimpan dalam memori tag transponder dengan data yang dikirimkan oleh reader. RFID dibentuk oleh komponen utama tag reader dan antenna tag dapat menggunakan daya aktif (tag aktif) atau tidak (tag pasif) serta diletakkan pada objek yang akan diidentifikasi. Pada tag pasif sinyal dikirimkan oleh reader melalui gelombang elektromagnetik, kemudian tag akan merespon dan mengirimkan data/informasi di dalamnya. Reader juga memiliki kemampuan untuk melakukan perubahan data pada tag selain membaca dan mengambil data informasi yang tersimpan dalam tag. Sedangkan antenna pada System RFID berpengaruh terhadap jarak jangkauan pembacaan atau identifikasi objek. Teknologi RFID menggunakan frekuensi antara 30kHz hingga 3GHz.



Gambar 2.2 Komponen sistem kerja RFIDD (sumber : RFID Handbook, Klaus Finkenzeller)

2.4 Perbedaan RFID dengan Barcode Optik

Menurut Ahson (2008) keunggulan RFID dibandingkan dengan barcode yaitu :

- a. Identifikasi yang unik, artinya sebuah barcode mengidentifikasi tipe obyek tempat ia dicetak, misalnya “sebatang coklat merek ABC dengan kadar 70% dan berat 100 gram”. Sebuah tag RFID selangkah lebih maju dengan

mengemisikan sebuah nomor seri unik di antara jutaan obyek yang identik, sehingga ia dapat mengidentifikasi “sebatang coklat merek ABC dengan kadar 70% dan berat 100 gram sengan nomor seri 1234567”. *Identifier* yang unik dalam RFID dapat berperan sebagai pointer terhadap entri basis data yang menyimpan banyak histor transaksi untuk item – item individu.

- b. Otomasi, artinya sebuah barcode memerlukan kontak line-ofsight dengan reader yaitu dengan cara scan optik, dan tentu aja peletakan fisik harus tepat dari obyek yang di scan.

Jadi menurut Ahson bahwa keunggulan dari RFID di bandingkan dengan barcode adalah Identifier yang unik dalam RFID dapat berperan sebagai pointer terhadap entri basis data yang menyimpan banyak histori transaksi untuk item – item individu dan RFID dapat dibaca tanpa kontak line-ofsight dan tanpa penepatan yang presisi. Reader RFID dapat melakukan scan terhadap tag RFID sebanyak ratusan perdetik

Menurut Garfinkel (2005) perbedaan mendasar antara RFID dengan barcode dapat terlihat pada tabel berikut ini :

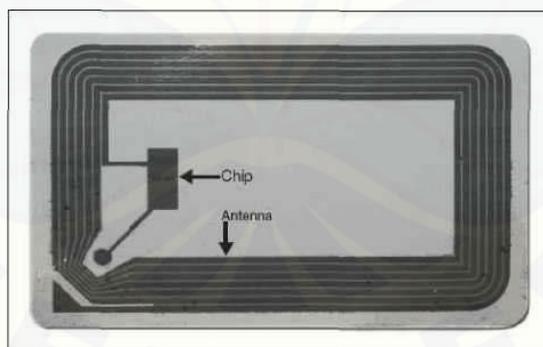
Tabel 2.1 Perbedaan RFID dengan Barcode

RFID	Barcode
Dapat ditempel dan tersembunyi, tidak memerlukan pandangan langsung	Harus dengan pandangan langsung
Dapat dibaca meskipun terhalang benda kecuali benda logam	Tidak dapat dibaca jika terhalang
Dapat diprogram/ entri ulang dalam keadaan bergerak	Tidak dapat di program / entri ulang dalam keadaan bergerak
Dapat diterapkan dalam lingkungan yang keras, seperti di luar rumah, sekitar bahan kimia dan kelembaban	Harus ditempatkan ditempat yang terlindungi agar tidak merusak kode
Tag RFID berisikan 1MB memori (1miliar karakter) bahkan fraksi terkecil dari 64 bits	Jumlah informasi terbatas sekitar 20 karakter

2.5 Tag RFID

Menurut Brown (2005) bahwa "Sistem RFID merupakan suatu sistem identifikasi otomatis yang bertujuan untuk memungkinkan data ditransmisikan oleh peralatan portable yang disebut tag. Kemudian tag tersebut dibaca oleh suatu reader RFID dan diproses menurut kebutuhan dari aplikasi tertentu. Data yang ditransmisikan oleh tag dapat menyediakan informasi identifikasi atau lokasi, atau hal-hal khusus tentang produk-produk ber-tag, seperti harga, warna, tanggal pembelian dan lain-lain".

Sebuah tag RFID atau transponder, terdiri atas sebuah microchip dan sebuah antena, (seperti terlihat pada Gambar.3). Microchip itu sendiri dapat berukuran sekecil butiran pasir, sekitar 0.4 mm. Chip tersebut menyimpan nomor seri yang unik atau informasi lainnya tergantung kepada tipe memorinya. Tipe memori itu sendiri dapat *read-only*, *read-write*, atau *write-once-read-many*. Antena yang terpasang pada mikrochip mengirimkan informasi ke reader RFID. Biasanya rentang pembacaan diindikasikan dengan besarnya antena. Antena yang lebih besar mengindikasikan rentang pembacaan yang lebih jauh. Tag tersebut terpasang atau tertanam dalam obyek yang akan diidentifikasi. Tag dapat di-scan dengan reader RFID bergerak maupun stasioner.



Gambar 2.3 Tag RFID (Sumber: Brown, 2005)

Tag RFID adalah alat yang dibuat dari IC dan antena yang terintegrasi di dalamnya, yang memiliki memori sehingga tag dapat digunakan untuk menyimpan data. Memori pada tag dibagi menjadi beberapa sel. Ada beberapa sel yang digunakan untuk menyimpan data *read only*, misalnya nomor seri yang unik

untuk disimpan saat sebuah tag diproduksi. Selain itu, ada beberapa sel lain yang dapat ditulis dan dibaca secara berulang (Weinsten, 2005).

Tag RFID ada berbagai macam namun secara garis besar dibagi menjadi dua yaitu tag aktif dan tag pasif. Perbandingan *Tag* aktif dan *Tag* Pasif terdapat pada tabel 2.2

Tabel 2.2 Perbandingan *Tag* aktif dan *Tag* Pasif

No	<i>Tag aktif</i>	<i>Tag pasif</i>
1	Adanya daya sendiri	Daya didapat dari reader
2	Jarak baca lebih jauh	Jarak baca dekat
3	Masa hidup <i>tag</i> tergantung baterai	Masa hidup <i>tag</i> lebih panjang
4	Bentuk lebih besar	Bentuk kecil
5	Harga mahal	Harga murah

Sumber : Bearl, 2005

2.6 Reader RFID

Menurut Milles (2005) bahwa Untuk berfungsinya sistem RFID, maka diperlukan sebuah reader atau alat scanning yang dapat membaca tag dengan benar dan mengkomunikasikan hasilnya ke suatu database yang ada. Sebuah reader menggunakan antenanya sendiri untuk berkomunikasi dengan tag. Ketika reader memancarkan gelombang radio, seluruh tag yang dirancang pada frekuensi tersebut serta berada pada rentang bacanya akan memberikan respon.

Sebuah pembaca RFID harus menyelesaikan dua buah tugas, yaitu:

- a. Menerima perintah dari software aplikasi
- b. Berkomunikasi dengan tag RFID

Pembaca RFID adalah merupakan penghubung antara software aplikasi dengan antena yang akan meradiasikan gelombang radio ke tag RFID. Akibatnya data dapat berpindah secara wireless ke tag RFID yang berada berdekatan dengan antena.

2.7 Frekuensi RFID

Pemilihan frekuensi radio merupakan kunci karakteristik operasi sistem RFID. Frekuensi menentukan kecepatan komunikasi dan jarak baca terhadap *tag*. Tingginya frekuensi mengindikasikan jauhnya jarak baca. Jika frekuensi yang lebih tinggi, jarak baca pun menjadi lebih jauh.

Menurut Pandian (2009) bahwa ada empat frekuensi utama yang digunakan oleh sistem RFID:

- a. Band LF berkisar dari 125 KHz hingga 1134 KHz. Band ini paling sesuai untuk penggunaan jarak pendek (*short-range*) seperti item anti pencurian, identifikasi hewan dan sistem kunci mobil.
- b. Band HF beroperasi pada 13,56 MHz. Frekuensi ini memungkinkan akurasi yang lebih baik dalam jarak tiga kaki dan karena itu dapat mereduksi resiko kesalahan pembacaan tag. Sebagai konsekuensinya band ini lebih cocok untuk pembacaan pada tingkat item (*item-level reading*). Tag pasif dengan frekuensi 13,56 MHz dapat dibaca dengan laju 10 s.d 100 tag perdetik pada jarak tiga kaki atau kurang. Tag RFID HF digunakan untuk pelacakan bagasi pesawat terbang, perpustakaan toko buku, kontrol akses gedung, pelacakan item pakaian, parkir.
- c. Tag dengan band UHF beroperasi di sekitar 900 MHz dan dapat dibaca dari jarak yang lebih jauh dari tag HF, berkisar dari 3 hingga 15 kaki. Tag ini lebih sensitif terhadap faktor-faktor lingkungan daripada tag-tag yang beroperasi pada frekuensi lainnya. Band 900 MHz muncul sebagai band yang lebih disukai untuk aplikasi rantai supply disebabkan laju dan rentang bacanya. Tag UHF pasif dapat dibaca dengan laju sekitar 100 hingga 1.000 tag perdetik. Tag ini umumnya digunakan pada pelacakan kontainer, truk, trailer, terminal peti kemas, serta telah diadopsi oleh peritel besar dan Departemen Pertahanan Amerika Serikat.
- d. Tag yang beroperasi pada frekuensi gelombang mikro, biasanya 2.45 dan 5.8 GHz, mengalami lebih banyak pantulan gelombang radio dari obyek-obyek di dekatnya yang dapat mengganggu kemampuan reader untuk berkomunikasi dengan tag. Tag RFID gelombang mikro biasanya digunakan untuk manajemen

rantai supply.

Dari kesimpulan yang didapat dari pendapat pandian (2009) mengenai frekuensi yang terdapat pada RFID adalah Band LF berkisar dari 125 KHz hingga 134 KHz, Band HF beroperasi pada 13.56 MHz, Tag dengan band UHF beroperasi di sekitar 900 MHz dan Tag yang beroperasi pada frekuensi gelombang mikro, biasanya 2.45 dan 5.8 GHz.

2.8 Kategori Sistem RFID

Menurut Miles (2008) bahwa "Secara kasar sistem-sistem RFID dapat dikelompokkan menjadi empat kategori sebagai berikut :

- a. Sistem Electronic Article Surveillance (EAS) : Umumnya digunakan pada toko-toko untuk menyensor ada tidaknya suatu item. Produk-produk diberi tag dan reader berantena besar ditempatkan di masing-masing pintu keluar toko untuk mendeteksi pengambilan item secara tidak sah.
- b. Sistem Portable Data Capture : dicirikan oleh penggunaan reader RFID yang portabel yang memungkinkan sistem ini digunakan dalam setting yang bervariasi.
- c. Sistem Networked : dicirikan oleh posisi reader yang tetap yang terhubung secara langsung ke suatu sistem manajemen informasi terpusat, sementara transponder berada pada orang atau item-item yang dapat dipindahkan.
- d. Sistem Positioning : Digunakan untuk identifikasi lokasi item-item atau kendaraan.

Kesimpulan yang diambil dari Miles (2008) mengenai pembagian sistem RFID adalah pertama Sistem Electronic Article Surveillance , kedua Sistem Portable Data Capture, sistem networked dan Sistem positioning.

Hasil penelitian yang dilakukan oleh Johan Wijaya Kusuma,dkk dengan judul PENGAMAN RUMAH MENGGUNAKAN KUNCI ELEKTRONIK BERBASIS RFID yang didalamnya menyebutkan bahwa menggunakan RFID diharapkan dapat diaplikasikan di setiap rumah demi meningkatkan keamanan

tanpa mengorbankan kenyamanan seluruh anggota keluarga dalam mengakses rumahnya sendiri.

Proses pengamanan berbasis *RFID* khususnya sebagai pembuka pintu belum banyak dikembangkan, sehingga merupakan peluang bagi kita untuk membuka mata pencaharian dibidang ini dan tentu saja selain mendapatkan keuntungan kita juga menyediakan lapangan kerja sehingga akan memperkecil angka pengangguran yang ada. Kunci rumah dengan *RFID* akan terus dikembangkan melalui penelitian dan uji coba untuk diaplikasikan diberbagai peralatan yang masih menggunakan kunci mekanik biasa sebagai pembukanya. Dengan menggunakan *RFID*, kita dapat menghemat penggunaan logam dalam proses pembuatan kunci dan secara tidak langsung mengurangi efek global warming serta mendukung program dunia yaitu *GO GREEN*.

2.9 Smartcard

Smartcard adalah kartu plastik yang berukuran sama dengan kartu kredit yang di dalamnya terdapat *chip* silikon yang disebut *microcontroller*. *Chip* merupakan *integrated circuit* yang terdiri dari prosesor dan memori. *Chip*, seperti layaknya CPU (*Central Processing Unit*) di komputer, bertugas melaksanakan perintah dan menyediakan *power* ke *smartcard*. *Smartcard* merupakan pengembangan dari kartu magnetis, namun berbeda dengan kartu magnetis yang hanya dipakai sebagai tempat penyimpanan data, *smartcard* mempunyai kemampuan untuk memproses dan menginterpretasikan data, serta menyimpan data tersebut secara aman dan juga akan sulit untuk memalsukan *smartcard* (Hanifah, 2010).

Kartu pintar ditemukan dan dipatenkan pada tahun 1970-an. Ada beberapa perselisihan tentang siapa “penemu” asli kartu ini termasuk Jurgen Dethloff dari Jerman, Arimura dari Jepang, dan Roland Moreno dari Perancis. Penggunaan massal pertama kali dari kartu ini adalah untuk pembayaran telepon di Perancis pada tahun 1983.

Kartu pintar sentuh memiliki chip keemasan dengan diameter sekitar setengah inci. Pada saat dimasukan ke Handphone atau Modem yang dapat membaca informasi dari chip dan menuliskannya kembali.

Standar ISO/IEC 7816 dan ISO/IEC mendefinisikan :

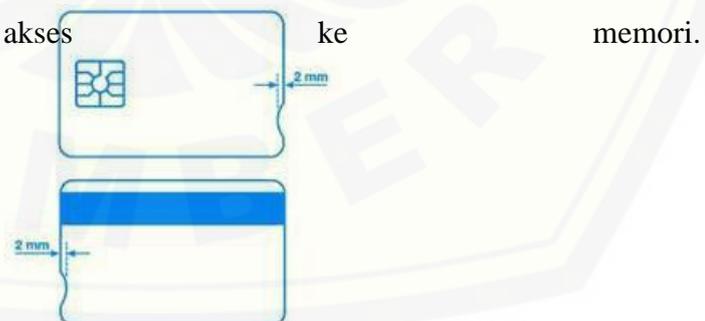
- a. Bentuk fisik
- b. Posisi dan bentuk penghubung elektronik
- c. Sifat kelistrikan
- d. Protokol komunikasi
- e. Ketahanan kartu
- f. Fungsi

Smart card dapat dikelompokan berdasarkan :

1. *Function* yang merupakan perbedaan paling mendasar antara memori card dan microprocessor card.
2. *Acces mechanism* yaitu *contact dan contactless*.
3. *Physical characteristic*, dilihat dari ukuran dan bentuk.

2.10 Tipe – tipe smartcard

Menurut sariasih (1999) ada 2 tipe smartcard, yaitu smartcard yang mempunyai mikroprosesor dan menawarkan kemampuan membaca, menulis dan melakukan penghitungan, seperti mikrokomputer kecil. Yang kedua adalah smartcard memori yang tidak mempunyai mikroprosesor dan digunakan hanya untuk tempat menyimpan. Smartcard memori menggunakan security logic untuk mengatur akses ke memori.

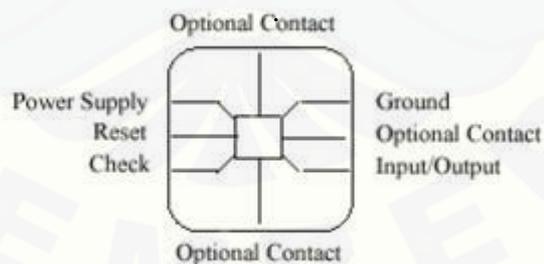


Gambar 2.4 . Bentuk *smartcard*

Secara komersial, industri membuat smartcard dalam beberapa varian, yaitu:

- a. **Memory cards.** *Smartcard* jenis ini hanya berfungsi untuk menyimpan data, tidak mempunyai processor atau sistem keamanan yang canggih melainkan hanya perlindungan fisik (karena smartcard bersifat tamper proof).
- b. **Memory protected cards.** *Smartcard* jenis ini mempunyai sistem keamanan yang lebih canggih daripada memory cards, misalnya mekanisme password untuk mengakses smartcard.
- c. **Microprocessor cards.** *Smartcard* jenis ini mempunyai processor sehingga dapat melakukan komputasi walaupun terbatas. Kemampuannya antara lain mengorganisasikan berkas (file) yang dilindungi dengan password.
- d. **Java cards.** *Smartcard* ini dilengkapi dengan Java Virtual Machine sedemikian hingga dapat dimasukkan berbagai program ke dalamnya.
- e. **Public key cards.** *Smartcard* ini mendukung public key cryptography (kriptografi asimetris) sehingga proses enkripsi/dekripsi dapat dilakukan secara internal dan dapat menyimpan key.

Pada umumnya, smartcard tidak berisi power supply, display atau keyboard. Smartcard berinteraksi dengan dunia luar dengan menggunakan antarmuka komunikasi serial melalui 8 titik kontak. Ukuran dan letak dari kontak tersebut didefinisikan didalam ISO 7816, bagian kedua. Gambar berikut menunjukkan kontak di dalam smartcard.



Gambar 2.5 Delapan titik kontak *smartcard*

2.11 Cara Kerja *Smartcard*

Aplikasi berkomunikasi dengan *reader* (yang kemudian akan berkomunikasi dengan *smartcard*) menggunakan protokol yang standar, yaitu protokol *International Standard Organization (ISO) 7816*. *Smartcard* merupakan *personal hardware* yang harus berkomunikasi dengan perangkat lainnya untuk mengakses perangkat *display* atau jaringan komputer.

Smartcard dapat berkomunikasi dengan *reader* dengan 2 cara, yaitu :

- a. *contact smartcard* - koneksi dibuat ketika *reader* bersentuhan dengan chip yang ada di *smartcard*.
- b. *contactless smartcard* – dapat berkomunikasi melalui antena, mengurangi keperluan untuk memasukkan dan mengambil *smartcard*. Dengan *contactless*, yang harus dilakukan hanya mendekatkan *smartcard* ke *reader*, dan selanjutnya *smartcard* akan berkomunikasi. *Contactless smartcard* dapat digunakan di dalam aplikasi dimana pemasukan/penarikan *smartcard* tidak praktis dan pertimbangan kecepatan.

Di sisi lain aplikasi ini juga melakukan otentikasi terhadap pemakai. Otentikasi dilakukan terhadap *smartcard* dengan cara mengetahui apakah *smartcard* tersebut asli, dalam arti :

- a. apakah *smartcard* memang berasal dari perusahaan pemberi layanan aplikasi tersebut.
- b. Beberapa perusahaan diasumsikan telah menerapkan skenario yang serupa, yaitu memberikan layanan dengan otentikasi menggunakan *smartcard*. tiap perusahaan akan menggunakan kode yang unik untuk menandai *smartcard* yang dikeluarkannya sehingga hanya *smartcard* yang dikeluarkan oleh perusahaan, misal A yang dapat mengakses layanan yang diberikan oleh perusahaan A. Hal ini juga dimaksudkan untuk mencegah sembarang *smartcard* dapat menggunakan layanan yang diberikan.
- c. apakah *smartcard* tersebut tidak ditiru atau digandakan. Data yang disimpan di dalam *smartcard* dapat ditiru jika tidak mendapat proteksi yang layak. Jika suatu ketika seorang *attacker* dapat membaca isi *smartcard*, dia bisa meng-copy isi (data yang disimpan) *smartcard*

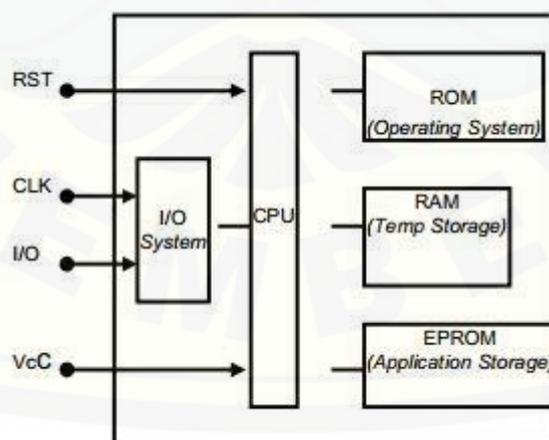
kesembarang *smartcard* sehingga *smartcard* hasil *copy* tersebut dapat digunakan seperti *smartcard* yang asli. Hal ini juga bisa dilakukan oleh si pemegang *smartcard* yang 'nakal' dengan menggandakan *smartcard* yang dimilikinya. Untuk itu pihak pemberi layanan perlu memeriksa apakah *smartcard* yang dipakai benar-benar asli.

- d. apakah pemakai benar-benar pemilik *smartcard* yang asli. *Smartcard* bisa hilang atau mungkin dicuri sehingga bisa digunakan oleh orang yang tidak berwenang. Hal ini tentu saja akan sangat merugikan pihak pemberi layanan maupun orang yang mempunyai *smartcard* tersebut. Oleh karena itu perlu mekanisme untuk mengetahui bahwa pemegang *smartcard* adalah orang yang benar-benar berhak.

Dalam rangka mengembangkan aplikasi berbasis *smartcard*, perlu beberapa perangkat: *smartcard reader*, perangkat lunak untuk berkomunikasi dengan *reader* maupun perangkat lunak yang berkomunikasi dengan kartu dan *smartcard*. *Reader* menyediakan *path* untuk aplikasi, untuk mengirim dan menerima *command* dari kartu (Sariasih, 1999).

2.12 Komponen *Smartcard*

Komponen *smartcard* dibagi menjadi berbagai komponen komponen dasar yaitu :



Gambar 2.6 Komponen *smart card*

a. *Carrier*

Material dasar yang digunakan untuk pembuatan *smartcard* adalah *polyvinyl chloride* (PVC) atau thermoplastic sejenis. Selain itu bahan yang dapat digunakan adalah *Acrylonitrile butadine styrene* (ABS) yang lebih tahan pada suhu tinggi, *Polycarbonate* yang digunakan untuk kartu mobile-telephone, dan *Polyethylene terephthalate* (PETP) yang banyak digunakan di Jepang karena fleksibel dan ringan.

b. *Chip*

Komponen utama kartu yaitu IC yang dipasang didalam kartu. Isi *chip* ini bisa terdiri dari memory, microprosesor atau PGA Chip.

c. *Microprocessor*

Kebanyakan *smartcard* mempunyai 8-bit microprosesor dengan desain Motorola 6805 atau intel 8051 dengan clock speed tertinggi 5 MHz. Tetapi sudah ada microprosesor 16-bit. RISC microcontroler banyak digunakan di *smartcard* untuk aplikasi yang perlu kecepatan proses daripada multifunction.

d. *Memory*

Memori memakan tempat terbesar di IC *smartcard*. Memori ini dibagi menjadi 5 area berdasarkan tipe semi konduktor memori yang dipakai.

e. *Coprocessor*

Bagian chip yang dibuat untuk melakukan operasi aritmatika dalam fungsi kriptografi seperti reinkripsi DES atau RSA

f. *Memory Management*

Digunakan untuk mengontrol memory dan menyediakan proteksi hardware dari akses yang tidak valid. Proteksi ini menggunakan metode hirarki data.

g. *I/O*

Microprosesor *smartcard* menggunakan single bidirectional serial input – output interface. Metode ini sesuai dengan standar ISO 7816-3 tentang protokol komunikasi.

h. *Contact*

Contact Card mempunyai kurang lebih delapan titik kontak. Posisi dan desainnya disesuaikan dengan ISO 7816-2. Walaupun begitu masih banyak orang terutama di Perancis, yang menggunakan desain posisi transisi (pojok kiri atas). Contact ini dibuat dari emas atau bahan berkonduksi lainnya. Kontak ini dihubungkan dengan chip dengan kabel yang sangat tipis.

i. *Antenna*

Contactless card menggunakan sinyal dengan frekuensi radio sebagai media transmisi I/O. Antena sehingga antena dipasang di kartu sebagai coil. Antena juga berfungsi mendapatkan energi dari frekuensi radio, selain baterai yang ada pada kartu. Sinyal yang digunakan mempunyai frekuensi 135 KHz atau 13,56 MHz. Jika menggunakan frekuensi yang rendah, energi yang diperlukan rendah dan bisa mencapai jangkauan 1m, tetapi kecepatan transfer data rendah. Sedangkan jika menggunakan frekuensi tinggi, maka akan memakan energi lebih tinggi dan kecepatan transfernya juga tinggi.

j. Mask

Mask adalah program permanen yang ada di ROM, sering disebut sebagai OS *smartcard*. Perbedaannya dengan OS PC pada umumnya adalah Mask melakukan fungsi aplikasi seperti mengurangi nilai, membandingkan tanda tangan digital dan pola yang ada (Sariasih, 1999).



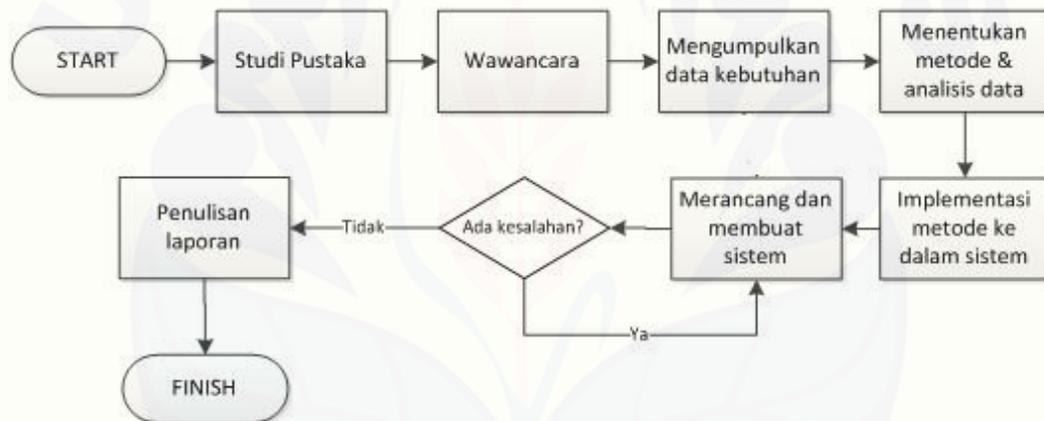
BAB 3. METODOLOGI PENELITIAN

3.1 Jenis Penelitian

Metode penelitian yang dilakukan yaitu dengan menggunakan metode deskripsif, karena mendiskripsikan dan menginterpretasikan sesuatu. Misalnya sistem yang dibuat saat ini adalah sistem parkir yang otomatis menggunakan teknologi RFID untuk menggantikan sistem parkir yang manual

3.2 Tahapan penelitian

Pada bagian ini akan di jelaskan tentang tahapan yang dilakukan agar penelitian dapat berjalan baik dan teratur. Tahapan penelitian digambarkan dengan *flow chart* diagram seperti pada gambar 3.1



Gambar 3.1 *Flow chart* tahapan penelitian

3.3 Objek Penelitian

Penelitian dilakukan di Program Studi Sistem Informasi Univeritas negeri Jember, pada penelitian ini penulis mengambil data tentang sistem parkir dan jumlah kendaraan yang sering melakukan aktifitas di area parkir Program Studi Sistem Informasi.

3.4 Teknik Pengumpulan Data

Teknik pengumpulan yang dilakukan oleh penulis antara lain :

1. Studi Kepustakaan (*Literature*)

Menurut Miller dkk (2010) tinjauan literatur adalah bagian penting dari setiap proyek penelitian yang baik, dan jika tidak dilakukan dengan benar, proyek dapat berakhir sia-sia dengan mereplikasi penelitian sebelumnya. Metode ini dilakukan untuk mendapatkan data *literature* tambahan dari buku acuan mengenai teknologi RFID sumber yang digunakan berupa buku, jurnal, karya ilmiah, dan situs-situs penunjang yang dapat membantu dalam penyelesaian laporan penelitian.

2. Wawancara (*Interview*)

Wawancara merupakan salah satu objek penelitian. Menurut Jogiyanto (2007) wawancara adalah Komunikasi dua arah untuk mendapatkan data dari responden. Wawancara dapat berupa wawancara personal, wawancara intersep dan wawancara telepon. Dalam penelitian ini penulis melakukan wawancara secara langsung pada karyawan dan mahasiswa tentang sistem perpustakaan yang ada saat ini

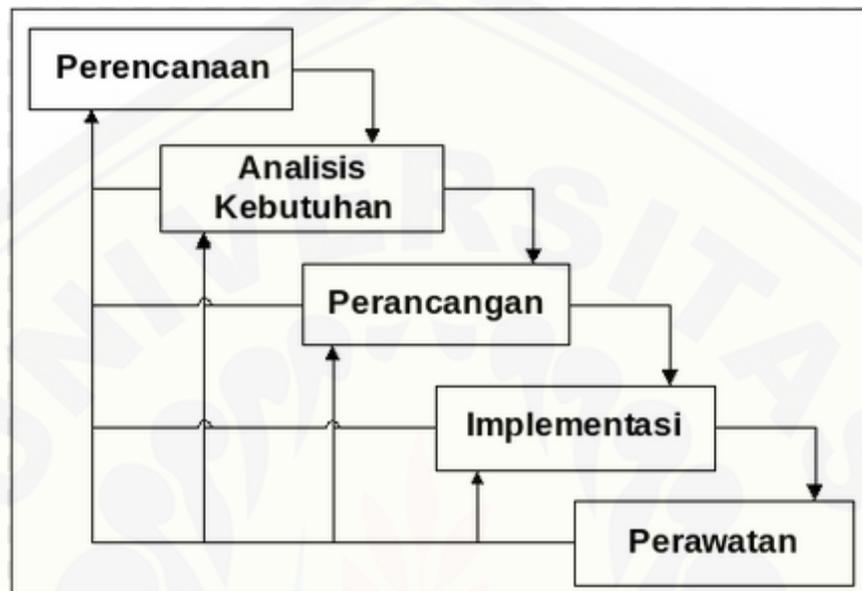
3. Pengamatan (*Observasi*)

Observasi adalah melakukan pengamatan secara langsung ke objek penelitian untuk melihat dari dekat kegiatan yang dilakukan (Riduwan, 2010). Observasi dilakukan untuk mengetahui sistem parkir yang di pakai saat ini.

3.5 Tahap Perancangan

Metode analisis yang digunakan dalam perancangan sistem menggunakan metode *Object Oriented* (Berorientasi Objek) yang merupakan salah satu teknik konsep yang digunakan di dalam bahasa pemrograman. Konsep ini digunakan karena memiliki kemudahan dalam pengembangan program aplikasi oleh programmer.

Tahap perancangan (software) sistem dilakukan dengan mengadopsi model *waterfall*. *Waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam membangun software (Pressman 2010). Tahapan pada model *Waterfall* dapat dilihat pada gambar 3.2



Gambar 3.2 Tahapan model *waterfall* (Sumber: Pressman, 2010)

Model *waterfall* memiliki beberapa tahapan yaitu :

1. Perencanaan

Perencanaan merupakan suatu analisa terhadap kebutuhan *software*, dan tahap untuk mengadakan pengumpulan data – data dengan melakukan dengan *customer*, maupun mengumpulkan data – data tambahan baik yang ada di jurnal, artikel, maupun internet

2. Analisis kebutuhan

Proses analisa kebutuhan merupakan proses dari perencanaan. Tahapan ini akan menghasilkan dokumen *user requirement* atau suatu data yang berhubungan dengan keinginan user dalam pembuatan software

3. Perancangan

Proses perancangan ini akan menerjemahkan syarat kebutuhan ke sebuah perancangan *software* yang dapat diperkirakan sebelum dibuat *coding*. Proses ini berfokus pada rancangan struktur data, arsitektur *software*, representasi

interface, dan detail (algoritma) prosedural. Tahapan ini akan menghasilkan dokumen yang disebut *software requirement*.

4. Implementasi

Implementasi merupakan proses membuat kode. *Coding* atau pengkodean merupakan penerjemahan desain dalam bahasa yang bisa dikenali oleh komputer. *Programmer* akan menerjemahkan transaksi yang diminta oleh *user*. Tahapan inilah yang merupakan tahapan secara nyata dalam mengerjakan suatu *software*, artinya penggunaan komputer akan dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai maka akan dilakukan *testing* terhadap sistem yang telah dibuat tadi. Tujuan *testing* adalah menemukan kesalahan-kesalahan terhadap sistem tersebut untuk kemudian bisa diperbaiki.

5. Perawatan

Tahapan ini bisa dikatakan final dalam pembuatan sebuah *software* atau sistem. Setelah melakukan analisis, desain dan pengkodean maka sistem yang sudah jadi akan digunakan oleh *user*. Kemudian *software* yang telah dibuat harus dilakukan pemeliharaan secara berkala.

Kelebihan dari model ini adalah selain karena pengaplikasian menggunakan model ini mudah, kelebihan dari model ini adalah ketika semua kebutuhan sistem dapat didefinisikan secara utuh, eksplisit, dan benar di awal proyek, maka *Software Engineering* (SE) dapat berjalan dengan baik dan tanpa masalah. Meskipun seringkali kebutuhan sistem tidak dapat didefinisikan se-eksplisit yang diinginkan, tetapi paling tidak, problem pada kebutuhan sistem di awal proyek lebih ekonomis dalam hal uang (lebih murah), usaha, dan waktu yang terbuang lebih sedikit jika dibandingkan problem yang muncul pada tahap-tahap selanjutnya.

Kekurangan yang utama dari model ini adalah kesulitan dalam mengakomodasi perubahan setelah proses dijalani. Fase sebelumnya harus lengkap dan selesai sebelum mengerjakan fase berikutnya. Masalah dengan *waterfall* :

1. Perubahan sulit dilakukan karena sifatnya yang kaku.
2. Karena sifat kakunya, model ini cocok ketika kebutuhan dikumpulkan secara

lengkap sehingga perubahan bisa ditekan sekecil mungkin. Tapi pada kenyataannya jarang sekali konsumen/pengguna yang bisa memberikan kebutuhan secara lengkap, perubahan kebutuhan adalah sesuatu yang wajar terjadi.

3. *Waterfall* pada umumnya digunakan untuk rekayasa sistem yang besar yaitu dengan proyek yang dikerjakan di beberapa tempat berbeda, dan dibagi menjadi beberapa bagian sub-proyek.

3.6 Implementasi

Pada tahap ini penulis melakukan penerapan sistem dan pengujian terhadap sistem yang sudah di analisa dan di desain oleh penulis. Pada pengujian di lakukan dengan dua metode yaitu *Black-Box testing* dan *White-Box testing*.

a. *Black-Box testing*

Menurut Pressman (2010) *Black-box testing* disebut juga bahaviora testing. Berfokus pada kebutuhan fungsional dari software. Teknik ini memungkinkan untuk mendapatkan beberapa kondisi input yang akan sepenuhnya melaksanakan kebutuhan fungsional dari program. Pengujian *Black-box* berusaha untuk menemukan error pada beberapa kategori sebagai berikut :

1. Tidak benar atau hilangnya fungsi
2. Kesalahan antarmuka (*interface*)
3. Kesalahan dalam struktur data atau akses *database* eksternal
4. Kesalahan perilaku dan kinerja
5. Kesalahan *inialisasi* dan *terminasi*

Pengujian yang dilakukan pada sistem ini bertujuan untuk memastikan apakah fungsi-fungsi maupun masukan dan keluaran (I/O) dalam sistem sudah sesuai dengan kebutuhan dan mampu menjalankan semua fitur yang sudah dirancang.

b. *White-Box testing*

Menurut Pressman (2010), *white-box testing* dapat pula disebut *glas-box testing*. *White box* merupakan pengujian struktur kontrol yang dilakukan oleh perancang sistem tersebut, yang meliputi antara lain :

1. Memeriksa semua *independent path* dalam modul telah dieksekusi setidaknya sekali.
2. Memeriksa semua *logical decisions* benar atau salah
3. Memeriksa semua *loop* dieksekusi atau tidak
4. Memeriksa internal data struktur untuk memastikan kevalidannya (*validity*)

White-box testing pada sistem ini akan dilakukan dengan menghitung *independent path*. Cara menghitung *independent path* adalah menggunakan cyclomatic complexity. *Cyclomatic complexity* merupakan sebuah pengukuran kuantitatif kompleksitas logis suatu program dari grafik alir yang dapat diperoleh dengan perhitungan :

$$V(G) = E - N + 2 \dots (1)$$

E = Jumlah edge atau garis grafik alir

N = Jumlah simpul grafik alir

3.7 Perawatan (*maintenance*)

Tahap perawatan dilakukan ketika sistem sudah dioperasikan. Pada tahapan ini dilakukan monitoring proses atau pengontrolan, evaluasi, dan perubahan (perbaikan) bila diperlukan

BAB 4. PERANCANGAN SISTEM PERPARKIRAN KENDARAAN

Pada bab ini akan diuraikan tentang penyusunan sistem perparkiran kendaraan roda dua di Program Studi Informasi Universitas Jember. Yang meliputi perancangan basis data, perancangan *interface* dan perancangan program aplikasi dengan teknologi *Radio Frequency Identification* (RFID) yang digunakan dalam penelitian ini.

4.1 Deskripsi umum

Deskripsi umum aplikasi yang akan dibangun dalam penelitian ini adalah aplikasi *PARKIRAN* yang dapat membantu mahasiswa, karyawan dan dosen untuk mendapatkan kenyamanan dan keamanan kendaraannya dalam memarkir di area parkir Program Studi Informasi Universitas Jember. Selain itu aplikasi ini juga untuk memudahkan petugas parkir untuk mengetahui jumlah volume kendaraan yang masuk dan nomer kendaraan yang dulunya menggunakan cara manual yaitu dengan cara melihat satu persatu STNK dari masing – masing kendaraan yang di parkir dalam area parkir Program Studi Informasi Universitas Jember. Aplikasi ini berupa aplikasi yang berbasis desktop yang hanya dapat dioperasikan oleh petugas parkir itu sendiri. Dengan adanya aplikasi ini nantinya petugas parkir tidak perlu melihat surat kendaraan secara manual karena nantinya plat nomor kendaraan akan tersimpan secara otomatis dengan menggunakan alat yang bernama *Radio Frequency Identification* (RFID). Sistem ini nantinya dijalankan menggunakan *smart card* jadi bagi kendaraan yang mau di parkir pemiliknya harus memiliki *smart card* terlebih dahulu. Karena setiap aktifitas yang dilakukan atau keluar masuknya kendaraan akan tersimpan secara otomatis kedalam sistem ini.

4.2 Analisis Kelemahan dan Kekurangan Sistem Perparkiran Awal

Kendala dan kelemahan yang terdapat dalam sistem perparkiran kendaraan dalam metode secara manual yaitu dengan cara melihat satu persatu setiap

kendaraan yang keluar masuk area parkir Program Studi Informasi Universitas Jember antara lain :

1. Sistem parkir rentan terhadap pencurian yang disebabkan kelalaian petugas parkir kendaraan yang lupa tidak melihat setiap kendaraan yang keluar masuk area parkir yang tidak membawa Surat Tanda Nomor Kendaraan (STNK)
2. Tidak adanya data yang tersimpan seperti kapan kendaraan masuk dan kapan kendaraan keluar. Karena pada metode melihat satu persatu nomor kendaraan tidak adanya penyimpanan dan duplikasi data.
3. Tidak semua kendaraan diperiksa nomor kendaraan oleh penjaga karena kendaraan ataupun pemiliknya sudah dikenal oleh petugas parkir sehingga ketertiban dan keamanan sangat rendah.
4. Keharusan petugas parkir untuk melihat satu persatu nomor kendaraan pengguna yang nantinya membutuhkan waktu yang lama sehingga terjadi antrian yang tidak terduga.

4.3 Perancangan Sistem Perparkiran Kendaraan Roda Dua Baru

Dari analisa yang didapat dari sistem parkiran yang lama dapat diketahui bahwa kelemahan dan kekurangan sistem tersebut dapat diatasi dengan menerapkan sistem perparkiran kendaraan roda dua yang menggunakan teknologi RFID (*Radio Frekuensi Identifikasi*). Keunggulan dari sistem ini adalah :

1. Tindak pencurian dapat diatasi karena setiap pemilik kendaraan harus memiliki *tag* yang berfungsi untuk memberikan identitas terhadap setiap kendaraan dan pemilik sehingga yang dapat keluar masuk hanya yang memiliki kartu itu saja.
2. Data kendaraan, data pengguna atau pemilik dan data kejadian keluar dan masuk kendaraan tersimpan dalam basis data. sehingga dapat meminimalisir tindak kejahatan terhadap pencurian kendaraan yang di parkir.
3. Sistem yang terotomatisasi akan memberikan kemudahan, kenyamanan dan keamanan bagi pengguna dan petugas jaga parkir itu sendiri.

4.4 Requirement Specification

Untuk mengetahui spesifikasi yang dibutuhkan, maka diperlukan analisis mengenai spesifikasi kebutuhan sistem. Spesifikasi kebutuhan sistem dibedakan menjadi dua, yaitu kebutuhan fungsional dan kebutuhan non fungsional. Kebutuhan fungsional adalah jenis kebutuhan yang berisikan proses – proses apa saja yang diberikan oleh sistem informasi tersebut. Sedangkan kebutuhan non fungsional adalah batasan layanan atau fungsi yang ditawarkan sistem.

4.4.1 Kebutuhan Fungsional

Kebutuhan fungsional merupakan fitur dasar yang harus dimiliki oleh perangkat lunak untuk menerima masukan data kemudian diproses hingga menghasilkan suatu keluaran. Sistem perparkiran kendaraan yang menggunakan teknologi RFID memiliki kebutuhan fungsional yaitu :

1. sistem dapat menginputkan identitas pengunjung
2. sistem dapat menginputkan data pengunjung
3. sistem dapat menampilkan rekap data pengunjung
4. sistem dapat membaca data input dari RFID

4.4.2 Kebutuhan Non Fungsional

Kebutuhan non fungsional bertujuan untuk memenuhi kebutuhan fungsional selain itu kebutuhan non fungsional merupakan fungsi layanan pada sistem yang tidak secara langsung terkait pada fungsi sistem. Adapun kebutuhan non fungsional dari sistem perparkiran ini adalah :

1. Tampilan menarik dan interaktif untuk memberikan kenyamanan pemakaian bagi *user* dalam mengakses sistem

2. Sistem mampu menampilkan hasil dalam waktu hitunan detik
3. Sistem hanya mampu di operasikan melalui sistem operasi Microsoft Windows.

4.5 Perancangan Sistem

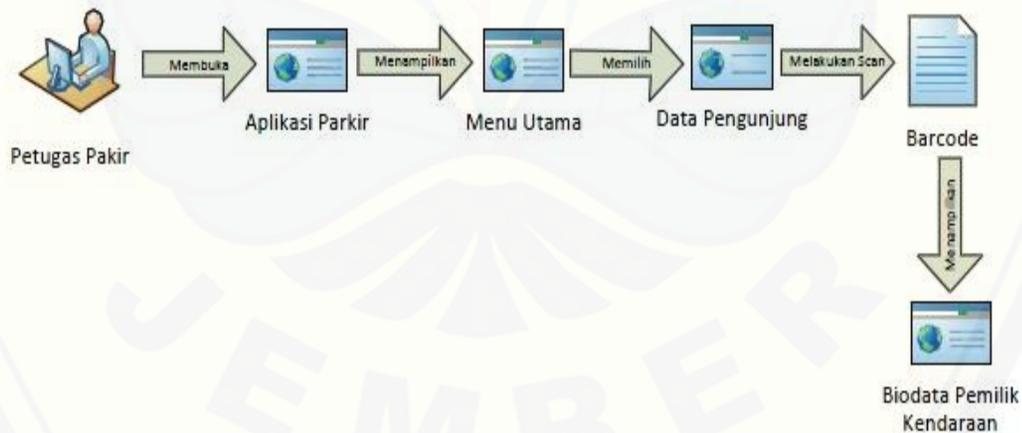
Tujuan utama dari perancangan sistem secara umum adalah memberikan gambaran tentang sistem yang akan di buat serta untuk memahami alur informasi dan proses yang ada dalam sistem.

4.5.1 Workflow

Workflow adalah aliran kerja atau suatu informasi dari proses bisnis, baik secara keseluruhan maupun sebagian dimana dokumen atau informasi tugas tersebut diteruskan dari satu partisipan ke partisipan lain sesuai dengan prosedur atau ketentuan yang berlaku. *Workflow* digunakan untuk menentukan alur kerja dari sistem parkir kendaraan roda dua di Sistem Informasi Universitas Jember.

4.5.1.1 Workflow Data Pengunjung

Workflow Input data pengunjung menggambarkan alur kerja proses data pengunjung pada sistem parkir seperti pada Gambar 4.1

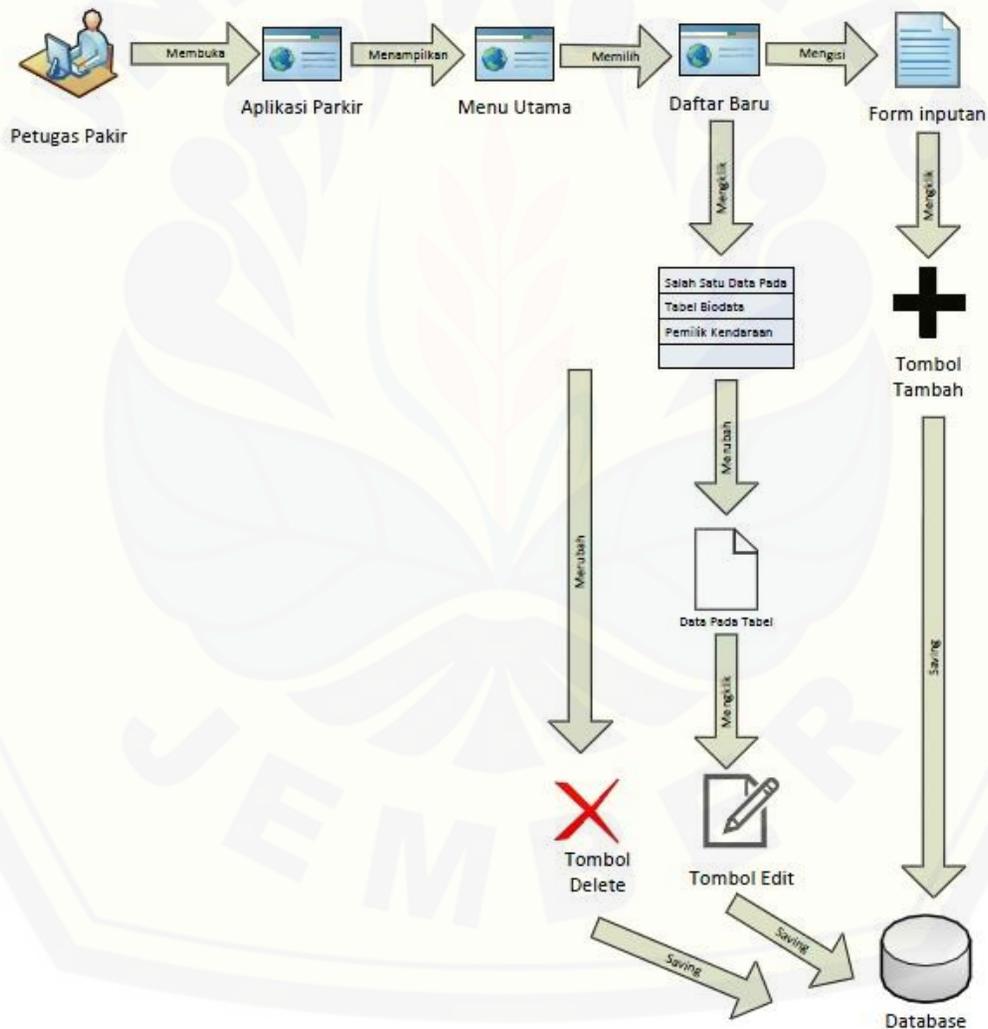


Gambar 4.1 *Workflow* input data pengunjung sistem parkir menggunakan teknologi RFID

Berdasarkan Gambar 4.1 dijelaskan bahwa petugas parkir harus membuka aplikasi sistem parkir kemudian memilih menu data pengunjung setelah itu *user* yang sudah terdaftar dan memiliki *smartcard* melakukan scanner dan nantinya akan muncul seluruh data pemilik kendaraan

4.5.1.2 Workflow Daftar Baru

Workflow daftar baru menggambarkan alur proses data dan cara melakukan proses daftar baru pada sistem parkir seperti pada Gambar 4.2

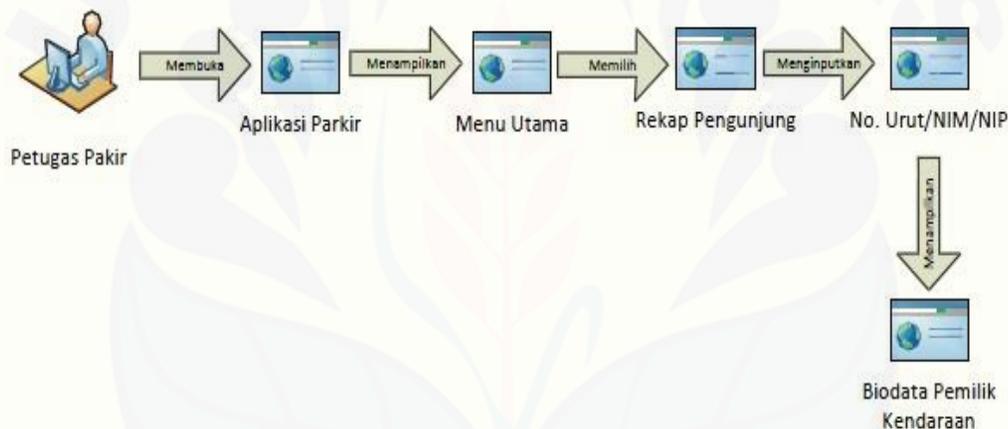


Gambar 4.2 *Workflow*daftar baru

Berdasarkan Gambar 4.2 dapat dilihat bahwa terdapat menu daftar baru. Petugas parkir dapat menambah *user* pada sistem, mengedit user, dan menghapus user pada sistem. Di fitur inilah petugas parkir dapat mengubah seluruh kelengkapan data user. Data user pada sistem ini adalah NIM/NIK/NIP, Nama, Jabatan, Angkatan, Merk kendaraan, Warna Kendaraan dan Nomor Polisi kendaraan. Dengan adanya fitur ini petugas parkir dapat dengan mudah memmanagement dan mengatur data user.

4.5.1.3 *Workflow* Rekap Data Pengunjung

Workflow Rekap Data Pengunjung menggambarkan alur proses Rekap Data Pengunjung pada Sistem Parkiran pada Gambar 4.3



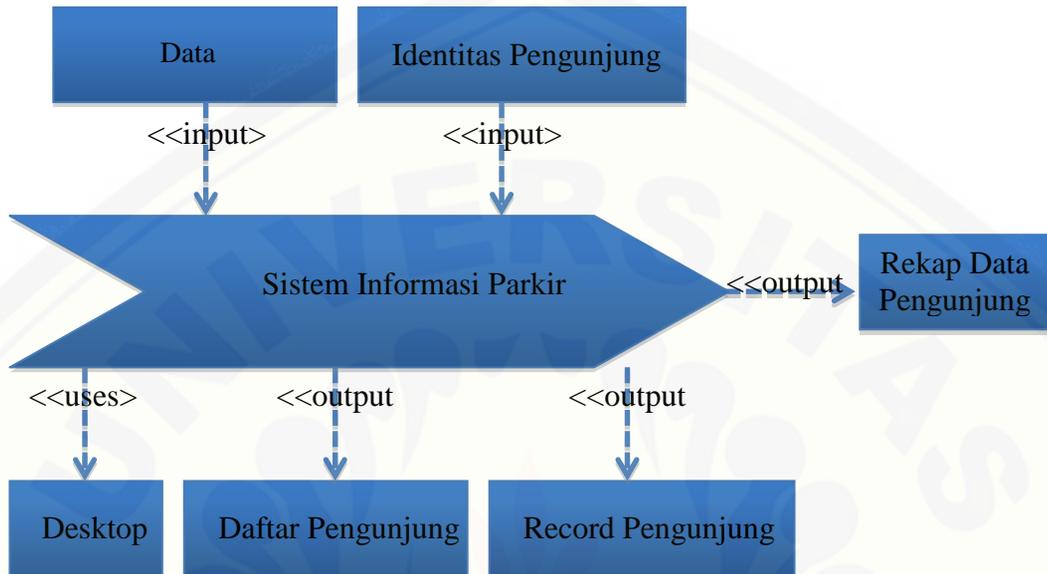
Gambar 4.3 *Workflow* rekap data pengunjung

Pada Gambar 4.3 dapat dilihat bahwa menu rekap pengunjung dapat menampilkan seluruh data yang masuk ataupun yang sudah keluar dari sistem.

4.5.2 *Business Process*

Pada Gambar 4.4 merupakan *business process* dari sistem parkir kendaraan roda dua di Program Studi Sistem Informasi Universitas Jember. *Business process* ini

menggambarkan komponen yang digunakan dalam pembuatan sistem, komponen tersebut kemudian di integrasikan untuk mencapai tujuan yang telah ditentukan.



Gambar 4.4 *Business process* Sistem Parkiran Kendaraan Bermotor roda dua

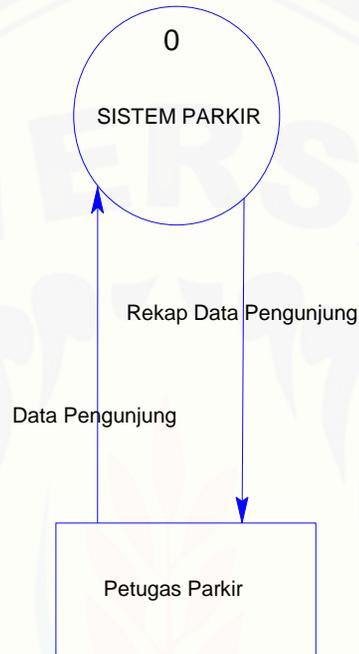
Berdasarkan Gambar 4.4 dapat dijelaskan bahwa yang menjadi input sistem adalah data pengunjung dan identitas pengunjung. Data pengunjung meliputi Nama, NIM, Jabatan, Angkatan sedangkan identitas pengunjung meliputi data kendaraan seperti merk kendaraan, plat nomor atau nomor polisi, dan warna kendaraan. Dan pada gambar 4.4 dijelaskan juga output dari sistem meliputi rekap data pengunjung, record data pengunjung, dan aplikasi berbasis desktop.

4.5.3 Data Flow Diagram (DFD)

Data Flow Diagram (DFD) adalah suatu model untuk menggambarkan asal data tujuan data serta proses apa saja yang terjadi pada suatu sistem. Dengan menggunakan DFD diharapkan dapat memudahkan *user* untuk mengerti bentuk aplikasi yang ditetapkan.

4.5.3.1 DFD level 0

DFD level 0 dapat dilihat dari gambar 4.5 dibawah ini. Pada DFD level 0 menggambarkan proses – proses dan aliran data pada sistem secara keseluruhan.

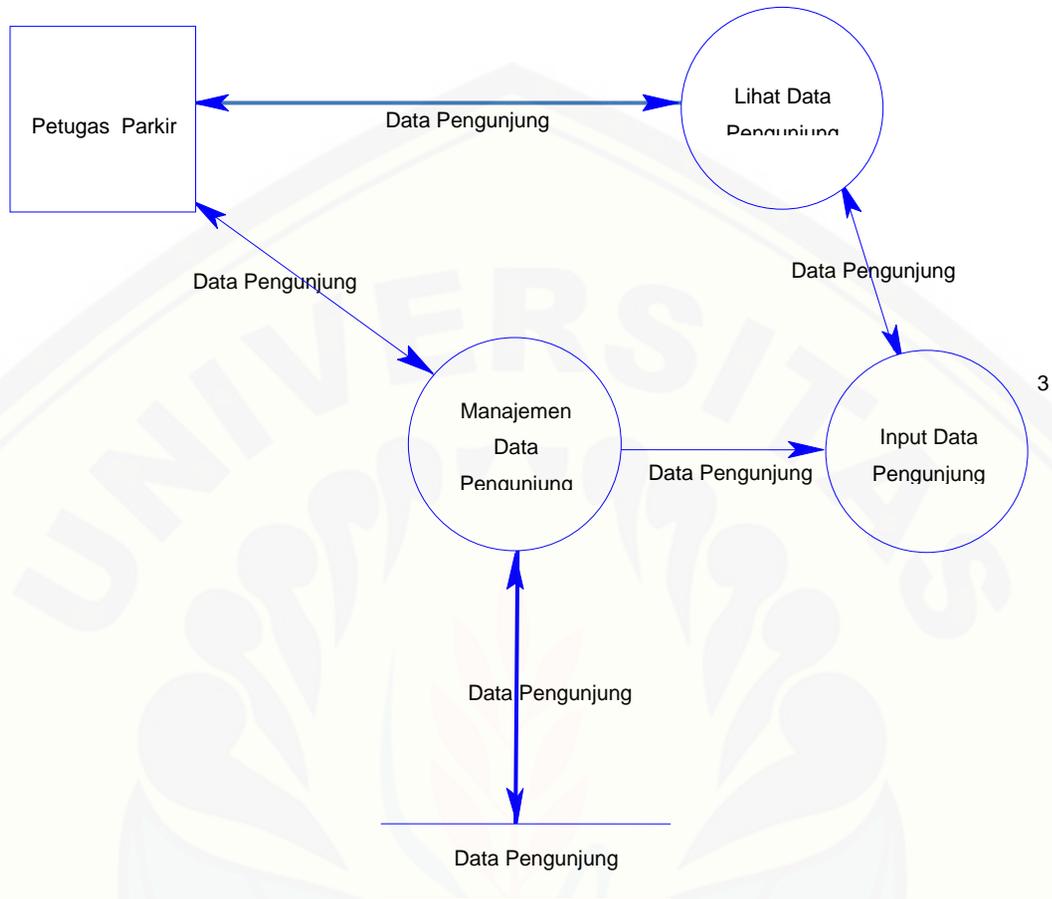


Gambar 4.5 DFD level 0

Pada gambar 4.5 dapat dijelaskan bahwa petugas parkir dapat mengakses seluruh fitur – fitur yang terdapat pada sistem mulai dari menambah user, mengubah data user, menghapus user dan mengedit data user. Petugas parkir juga dapat melihat seluruh data pengunjung.

4.5.3.2 DFD level 1

DFD level 1 dapat dilihat dari gambar 4.6 dibawah ini. Pada DFD level 1 di berikan gambaran yang lebih spesifik tentang sistem. DFD level 1 ini merupakan hasil dari pemecahan DFD level 0.

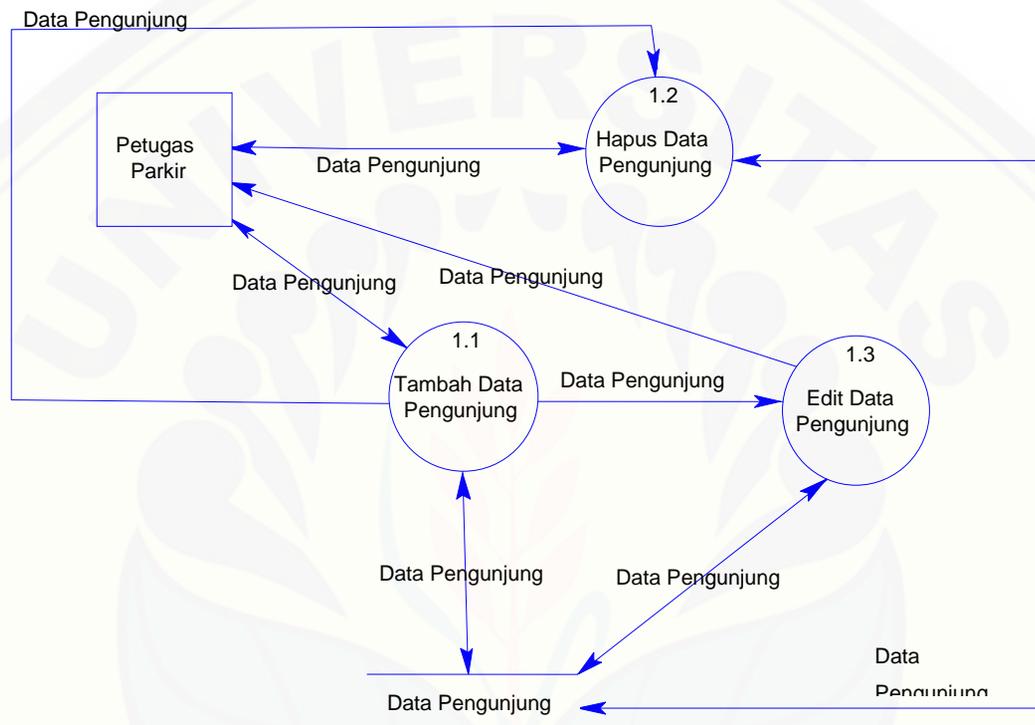


Gambar 4.6 DFD level 1

Pada gambar 4.6 dapat dijelaskan bahwa petugas parkir memiliki 3 fitur yaitu manajemen data pengunjung (petugas dapat menghapus data pengunjung, menambah pengunjung dan mengubah data pengunjung), lihat data pengunjung (petugas dapat melihat keseluruhan data pengunjung mulai dari Nama, NIM/NIK/NIP, angkatan dan jabatan) dan input data pengunjung.

4.5.3.3 DFD level 2

DFD level 2 dapat dilihat dari gambar 4.7 dibawah ini. Pada DFD level 2 di berikan gambaran yang lebih spesifik tentang sistem. DFD level 2 ini merupakan hasil dari pemecahan DFD level 1.



Gambar 4.7 DFD level 2

Pada Gambar 4.7 dapat dijelaskan bahwa petugas parkir dapat memiliki hak akses penuh terhadap sistem karena petugas parkir dapat menghapus data pengunjung, tambah data pengunjung dan edit data pengunjung.

4.5.4 Kamus Data

Tabel 4.1 Kamus data pengunjung

Nama Arus Data	Data Pengunjung
Alias	-
Format Data	Form input dan Data Digital
Arus Data	<ol style="list-style-type: none"> 1. Dari Petugas Parkir menuju Proses 1 2. Dari Proses 1 menuju Petugas Parkir 3. Dari Proses 1 menuju Storage Data Pengunjung 4. Dari Storage Data Pengunjung menuju Proses 1 5. Dari Proses 1 menuju Proses 3 6. Dari Petugas Parkir menuju Proses 1.1 7. Dari Proses 1.1 menuju Petugas Parkir 8. Dari Proses 1.1 menuju Storage Data Pengunjung 9. Dari Proses 1.1 menuju proses 1.2 10. Dari Proses 1.1 menuju proses 1.3 11. Dari Data Pengunjung menuju Proses 1.2 13. Dari Proses 1.2 menuju Data Pengunjung 14. Dari Petugas Parkir menuju Proses 1.3 17. Dari Proses 1.3 menuju Petugas Parkir 18. Dari Proses 1.3 menuju Storage Data Pengunjung 19. Dari Storage Data Pengunjung menuju Proses 1.3
Penjelasan	Data Pengunjung merupakan data mengenai pengunjung yang menggunakan fasilitas parkir.
Periode	Setiap kali dilakukan pengecekan pengguna jasa parkir
Volume	volume rata-rata tiap hari adalah 150 dan volume puncak adalah 200

Struktur Data	<ol style="list-style-type: none"> 1. user_id : int(11) 2. user_rfid : varchar(15) 3. user_nomer : varchar(20) 4. user_nama : varchar(25) 5. user_jabatan : enum('Dosen', 'Karyawan', 'Mahasiswa', 'Tamu') 6. user_angkatan : varchar(4) 7. user_merk : varchar(15) 8. user_warna : varchar(15) 9. user_nopol : varchar(15) 10. user_status : int(11)
----------------------	---

4.5.5 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) sistem parkir kendaraan bermotor roda dua di Program Study Informasi merupakan komponen dan struktur *database*. ERD sistem parkir kendaraan bermotor roda dua dapat dilihat pada gambar 4..8



Gambar 4.8 *Entity Relationship Diagram* Sitem Parkiran Kendaraan Bermotor roda dua

Berdasarkan gambar 4.8 dapat dijelaskan bahwa dalam sistem ini terdapat 2 entitas yaitu parkir.user dan parkir.record dengan relasi *one to many*. Tabel parkir.user bertindak sebagai *parent* dengan primary key user_id dan tabel.record sebagai *child*.

4.6 Pengujian Sistem

Pengujian sistem dilakukan setelah pengkodean sistem selesai, pengujian ini dilakukan untuk mengetahui apakah sesuai dengan kebutuhan *user*. Pengujian dilakukan dengan dua cara yaitu *white box* testing dan pengujian *black box* testing. Pengujian *white box* hanya 1 fitur saja dalam sistem ini, testing fitur lainnya dapat dilihat pada lampiran.

4.6.1 Pengujian *White Box*

White Box testing dapat juga disebut dengan *glas box* testing. *White box* merupakan pengujian struktur kontrol yang dilakukan oleh perancang sistem tersebut (Pressman, 2010).

4.6.1.1 Pengujian pada fitur Data Parkir

a. Listing Program

```

705
706 private void pdp_field_rfidKeyReleased(java.awt.event.KeyEvent evt) {
707     if (evt.getKeyCode() == 10) {}-1
708         String query = "SELECT user_nomer, user_nama, user_jabatan, user_angkatan, u.
709             + " FROM user
710             + " WHERE user_rfid = '" + pdp_field_rfid.getText() + "' AND user_st. }-2
711         try {
712             Statement st = kon.get_koneksi().createStatement();
713             ResultSet rs = st.executeQuery(query);
714
715             set_field(rs);
716             insert_parkir();
717             refresh_table(); }-4
718
719             pdp_field_rfid.setText("");
720             pdp_field_rfid.requestFocus();
721         } catch (SQLException ex) {}-5
722             Logger.getLogger(Parkir.class.getName()).log(Level.SEVERE, null, ex);}-6
723         }
724     }
725 }
726
727 private void set_field(ResultSet rs) {
728     try {}-6
729         while (rs.next()) {}-7
730             pdp_field_nim.setText(rs.getString("user_nomer"));
731             pdp_field_nama.setText(rs.getString("user_nama"));
732             pdp_field_jabatan.setText(rs.getString("user_jabatan"));
733             pdp_field_angkatan.setText(rs.getString("user_angkatan")); }-8
734             pdp_field_angkatan.setText(rs.getString("user_angkatan"));
735             pdp_field_merk.setText(rs.getString("user_merk"));
736             pdp_field_warna.setText(rs.getString("user_warna"));
737             pdp_label_no_polisi.setText(rs.getString("user_nopol"));
738         }
739     } catch (Exception e) {}-9
740         Logger.getLogger(Parkir.class.getName()).log(Level.SEVERE, null, e);}-10
741     }
742 }
743

```

```

743
744 private void insert_parkir() {
745     String query = "SELECT r.record_id"
746         + " FROM user u, record r"
747         + " WHERE user_rfid = '" + pdp_field_rfid.getText() + "' AND user"
748         + "         AND u.user_id = r.user_id AND DATE(masuk) = DATE(NOW())"
749
750     int baris = 0;
751     String id = "";
752
753     try { }-12
754         Statement st = kon.get_koneksi().createStatement(); }-13
755         ResultSet rs = st.executeQuery(query);
756
757         while (rs.next()) { }-14
758             id = rs.getString("record_id"); }-15
759             baris++;
760         }
761     } catch (Exception e) { }-16
762         System.out.println("pesan sistem 1 : " + e.getMessage()); }-17
763     }
764
765     if (baris == 0) { }-18
766         String query2 = "INSERT INTO record VALUES (NULL, (SELECT user_id FR" }-19
767
768         try { }-20
769             Statement st = kon.get_koneksi().createStatement(); }-21
770             st.executeUpdate(query2);
771         } catch (Exception e) { }-22
772             System.out.println("pesan sistem 2 : " + e.getMessage()); }-23
773     }
774 } else { }-24
775     String query2 = "UPDATE record SET keluar = NOW() WHERE record_id = " + id; }-25
776
777     try { }-26
778         Statement st = kon.get_koneksi().createStatement(); }-27
779         st.executeUpdate(query2);
780     } catch (Exception e) { }-28
781         System.out.println("pesan sistem 3 : " + e.getMessage()); }-29
782     }
783 }
784
785
786 private void refresh_table() {
787     try { }-30
788         String query = "SELECT user_nomer, user_nama, user_jabatan, user_nopol, TIME(masuk) AS masuk,
789             String[] judul = {"No", "NIM / NIK / NIP", "Nama", "Jabatan", "Plat Nomer", "IN", "OUT"};
790
791         Statement st = kon.get_koneksi().createStatement();
792         ResultSet rs = st.executeQuery(query);
793
794         DefaultTableModel model = new DefaultTableModel(null, judul);
795
796         int urut = 0;
797         while (rs.next()) { }-32
798             String user_nomer = rs.getString("user_nomer");
799             String user_nama = rs.getString("user_nama");
800             String user_jabatan = rs.getString("user_jabatan");
801             String user_nopol = rs.getString("user_nopol");
802             String masuk = rs.getString("masuk");
803             String keluar = rs.getString("keluar");
804
805             Object[] d = {++urut, user_nomer, user_nama, user_jabatan, user_nopol, masuk, keluar};
806
807             model.addRow(d);
808         }

```

```

809
810     table.setModel(model); }-34
811
812     if (table.getColumnModel().getColumnCount() > 0) {-35
813         table.getColumnModel().getColumn(0).setResizable(false);
814         table.getColumnModel().getColumn(0).setPreferredWidth(36);
815         table.getColumnModel().getColumn(1).setResizable(false);
816         table.getColumnModel().getColumn(1).setPreferredWidth(128);
817         table.getColumnModel().getColumn(2).setResizable(false);
818         table.getColumnModel().getColumn(2).setPreferredWidth(135);
819         table.getColumnModel().getColumn(3).setResizable(false);
820         table.getColumnModel().getColumn(3).setPreferredWidth(80);
821         table.getColumnModel().getColumn(4).setResizable(false);
822         table.getColumnModel().getColumn(4).setPreferredWidth(85);
823         table.getColumnModel().getColumn(5).setResizable(false);
824         table.getColumnModel().getColumn(5).setPreferredWidth(73);
825         table.getColumnModel().getColumn(6).setResizable(false);
826         table.getColumnModel().getColumn(6).setPreferredWidth(73);
827     }
828 } catch (Exception e) {-37
829     System.out.println("error : " + e.getMessage()); }-38
830 }
831 }
832

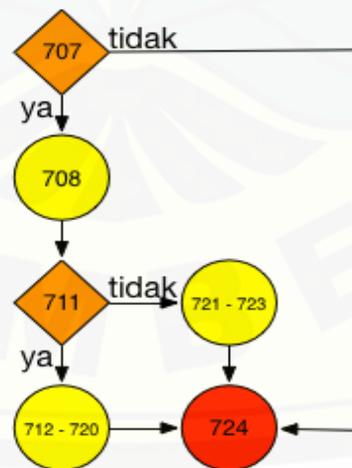
```

Gambar 4.9 Listing program data parkir

b. Kompleksitas Siklomatik

1. Fitur Data Parkir

a) Grafik Alir Function Record Data Parkir



Gambar 4.10 Grafik alir function

b) Penghitungan Kompleksitas siklomatik

Penghitungan kompleksitas siklomatik $V(G)$ untuk grafik alir G menggunakan rumus $V(G) = E - N + 2$. Dimana E adalah jumlah *edge* grafik alir dan N adalah jumlah *node* grafik alir.

$$V(G) = E - N + 2$$

$$V(G) = 7 - 6 + 2 = 3$$

4.6.2 Pengujian *Black Box*

Pengujian ini akan menguji aplikasi Parkiran menggunakan teknologi RFID yang dilakukan dalam bentuk tertulis. Pengujian *black box* ini dilakukan untuk menguji interface dari sistem, yang di sesuaikan dengan kebutuhan fungsional dan nonfungsional yang disesuaikan pada sistem yang telah dibuat. Pengujian ini dilakukan oleh *user* yang menggunakan sistem. *User* dalam sistem ini adalah para pengguna area parkir Program Studi Sistem Informasi Universitas Jember khususnya mahasiswa, karyawan dan dosen.

BAB 5. HASIL DAN PEMBAHASAN

Bab ini menguraikan dan menjelaskan tentang hasil dan pembahasan dari perancangan Sistem Informasi Perparkiran kendaraan roda dua di Program Studi Informasi Universitas Jember menggunakan teknologi *Radio Frequency Identification* (RFID).

5.1 Implementasi Sistem

Implementasi yang dimaksud dalam hal ini adalah mengkodekan sistem dengan menggunakan bahasa pemrograman JAVA NetBeans IDE 8.0 dan XAMPP dalam pembuatan *database*. Dilampirkan kode – kode program pada beberapa fitur yaitu fitur Data Pengunjung, Daftar Baru dan Rekap Data.

5.2 Hasil Sistem yang telah dibuat

Sistem informasi parkir kendaraan roda dua di program studi informasi dibuat dan dirancang menggunakan model waterfall seperti yang telah ditetapkan pada bab sebelumnya. Tahapan – tahapan yang dilakukan pada pembuatan sistem ini adalah mulai dari perencanaan, analisis kebutuhan, perancangan, implementasi dan perawatan. Model perancangan UML menggunakan konsep Prosedural. Pemodelan UML dilakukan mulai dari *Workflow*, *DFD level 0 sampai level 2*, *business proses*. Selain itu juga dibuat ERD (*Entity Relational diagram*) untuk menggambarkan hubungan – hubungan antar *entity* pada suatu *database*.

Tahap selanjutnya adalah pengujian sistem. Setelah sistem selesai dibuat, sistem diuji menggunakan sistem pengujian *black box* dan *white box*.

Sistem informasi parkir kendaraan roda dua yang telah dibuat nantinya mampu mendeteksi secara otomatis mulai dari biodata pemilik kendaraan dan data kendaraan sehingga lebih aman dan efisien dalam pengoperasiannya. Sistem ini berbasis desktop sehingga tidak memerlukan koneksi internet. Dalam sistem ini terdapat 3 fitur yaitu data pengunjung, daftar baru dan rekap pengunjung.

5.2.1 Data Pengunjung

Halaman data pengunjung hanya bisa dilihat oleh petugas parkir akan tetapi yang mampu menjalankan sistem ini untuk menghasilkan data pengunjung adalah *user* yang memiliki *tag* atau *smart card* yang telah di isi biodata data pengunjung dan kendaraan. Dalam fitur Data Pengunjung nantinya terdapat form NIM/NIK/NIP, Nama, Jabatan (Mahasiswa, Dosen, Karyawan dan Tamu), Angkatan, Merk Kendaraan, Warna Kendaraan dan juga menampilkan Nomor kendaraan pemilik. Dalam fitur ini juga terdapat field RFID dan tabel record. Halaman Data Pengunjung dapat dilihat pada gambar 5.1

No	NIM / NIK / NIP	Nama	Jabatan	Plat Nomer	Tanggal	IN	OUT
----	-----------------	------	---------	------------	---------	----	-----

Gambar 5.1 Screenshot halaman data pengunjung parkir

Halaman hasil dari data pengunjung setelah *user* melakukan scanner *smartcard* terhadap RFID dapat dilihat pada Gambar 5.2. pada halaman data pengunjung akan ditampilkan NIM/NIK/NIP, Nama, Jabatan (dosen, mahasiswa, karyawan dan tamu) angkatan, merk kendaraan, warna kendaraan, Nomor Polisi kendaraan, tanggal masuk serta jam masuk dan keluar setiap kendaraan yang parkir

di area parkir Program Studi Informasi. *Listing* program fungsi dari Data Pengunjung dapat dilihat pada tabel 5.1

Tabel 5.1 *Source code* data pengunjung

```
private void pdp_field_rfidKeyReleased(java.awt.event.KeyEvent evt) {
    if (evt.getKeyCode() == 10) {
        String query = "SELECT user_nomer, user_nama, user_jabatan,
user_angkatan, user_merk, user_warna, user_nopol"
            + " FROM user"
            + " WHERE user_rfid = '" + pdp_field_rfid.getText() + "' AND
user_status = 1";
        try {
            Statement st = kon.get_koneksi().createStatement();
            ResultSet rs = st.executeQuery(query);

            set_field(rs);
            insert_parkir();
            refresh_table();

            pdp_field_rfid.setText("");
            pdp_field_rfid.requestFocus();
        } catch (SQLException ex) {
            Logger.getLogger(Parkir.class.getName()).log(Level.SEVERE, null,
ex);
        }
    }
}

private void set_field(ResultSet rs) {
    try {
        while (rs.next()) {
            pdp_field_nim.setText(rs.getString("user_nomer"));
        }
    }
}
```

```
pdp_field_nama.setText(rs.getString("user_nama"));
pdp_field_jabatan.setText(rs.getString("user_jabatan"));
pdp_field_angkatan.setText(rs.getString("user_angkatan"));
pdp_field_angkatan.setText(rs.getString("user_angkatan"));
pdp_field_merk.setText(rs.getString("user_merk"));
pdp_field_warna.setText(rs.getString("user_warna"));
pdp_label_no_polisi.setText(rs.getString("user_nopol"));
    }
} catch (Exception e) {
    Logger.getLogger(Parkir.class.getName()).log(Level.SEVERE, null, e);
}
}

private void insert_parkir() {
    String query = "SELECT r.record_id"
        + " FROM user u, record r"
        + " WHERE user_rfid = '" + pdp_field_rfid.getText() + "' AND
user_status = 1"
        + "          AND u.user_id = r.user_id AND DATE(masuk) =
DATE(NOW()) AND keluar = '0000-00-00 00:00:00'";

    int baris = 0;
    String id = "";

    try {
        Statement st = kon.get_koneksi().createStatement();
        ResultSet rs = st.executeQuery(query);

        while (rs.next()) {
            id = rs.getString("record_id");
```

```
        baris++;
    }
} catch (Exception e) {
    System.out.println("pesan sistem 1 : " + e.getMessage());
}

if (baris == 0) {
    String query2 = "INSERT INTO record VALUES (NULL, (SELECT
user_id FROM user WHERE user_rfid = " + pdp_field_rfid.getText() + "
AND user_status = 1 LIMIT 1), NOW(), '0000-00-00 00:00:00')";

    try {
        Statement st = kon.get_koneksi().createStatement();
        st.executeUpdate(query2);
    } catch (Exception e) {
        System.out.println("pesan sistem 2 : " + e.getMessage());
    }
} else {
    String query2 = "UPDATE record SET keluar = NOW() WHERE
record_id = " + id;

    try {
        Statement st = kon.get_koneksi().createStatement();
        st.executeUpdate(query2);
    } catch (Exception e) {
        System.out.println("pesan sistem 3 : " + e.getMessage());
    }
}
}

private void refresh_table() {
```

```
try {
    String query = "SELECT user_nomer, user_nama, user_jabatan,
user_nopol, TIME(masuk) AS masuk, TIME(keluar) AS keluar FROM user u,
record r WHERE u.user_id = r.user_id AND DATE(masuk) = DATE(NOW())
ORDER BY record_id DESC";
    String[] judul = {"No", "NIM / NIK / NIP", "Nama", "Jabatan", "Plat
Nomer", "IN", "OUT"};
    Statement st = kon.get_koneksi().createStatement();
    ResultSet rs = st.executeQuery(query);

    DefaultTableModel model = new DefaultTableModel(null, judul);

    int urut = 0;
    while (rs.next()) {
        String user_nomer = rs.getString("user_nomer");
        String user_nama = rs.getString("user_nama");
        String user_jabatan = rs.getString("user_jabatan");
        String user_nopol = rs.getString("user_nopol");
        String masuk = rs.getString("masuk");
        String keluar = rs.getString("keluar");

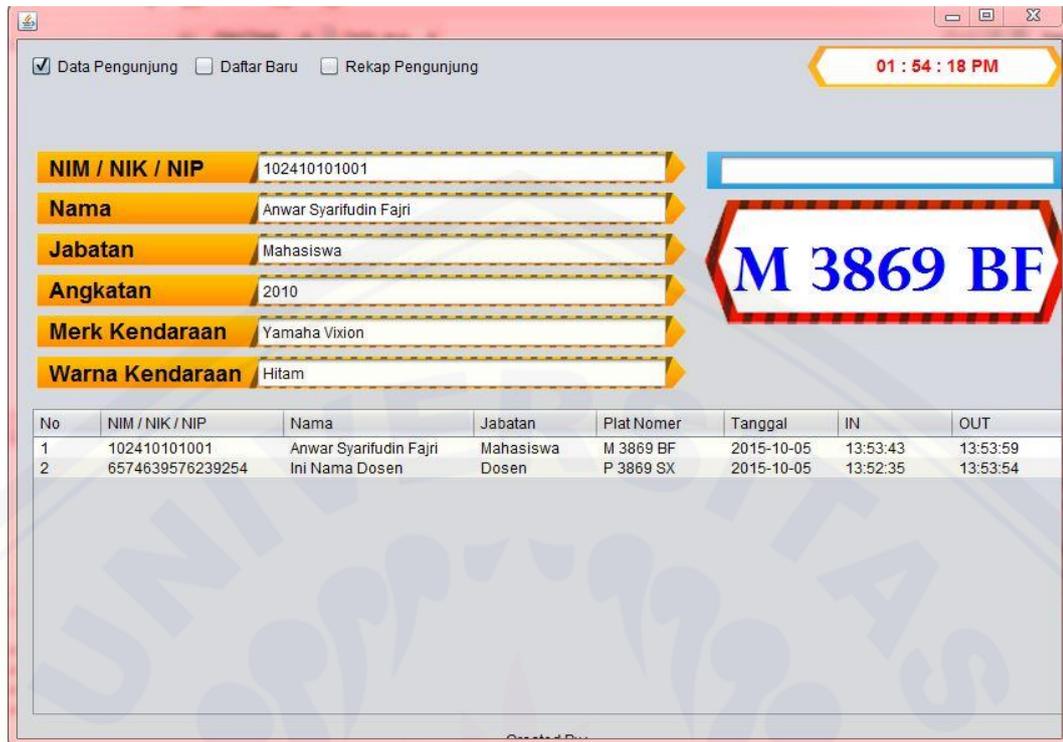
        Object[] d = {++urut, user_nomer, user_nama, user_jabatan,
user_nopol, masuk, keluar};

        model.addRow(d);
    }

    table.setModel(model);

    if (table.getColumnModel().getColumnCount() > 0) {
        table.getColumnModel().getColumn(0).setResizable(false);
    }
}
```

```
table.getColumnModel().getColumn(0).setPreferredWidth(36);
table.getColumnModel().getColumn(1).setResizable(false);
table.getColumnModel().getColumn(1).setPreferredWidth(128);
table.getColumnModel().getColumn(2).setResizable(false);
table.getColumnModel().getColumn(2).setPreferredWidth(135);
table.getColumnModel().getColumn(3).setResizable(false);
table.getColumnModel().getColumn(3).setPreferredWidth(80);
table.getColumnModel().getColumn(4).setResizable(false);
table.getColumnModel().getColumn(4).setPreferredWidth(85);
table.getColumnModel().getColumn(5).setResizable(false);
table.getColumnModel().getColumn(5).setPreferredWidth(73);
table.getColumnModel().getColumn(6).setResizable(false);
table.getColumnModel().getColumn(6).setPreferredWidth(73);
    }
} catch (Exception e) {
    System.out.println("error : " + e.getMessage());
}
}
```



The screenshot displays a web application interface for parking management. At the top, there are three checkboxes: "Data Pengunjung" (checked), "Daftar Baru", and "Rekap Pengunjung". A digital clock shows "01 : 54 : 18 PM". Below this, a series of yellow labels with white text are followed by input fields containing the following information:

- NIM / NIK / NIP**: 102410101001
- Nama**: Anwar Syarifudin Fajri
- Jabatan**: Mahasiswa
- Angkatan**: 2010
- Merk Kendaraan**: Yamaha Vixion
- Warna Kendaraan**: Hitam

To the right of these fields is a large blue and white license plate graphic that reads "M 3869 BF". Below the form fields is a table with the following data:

No	NIM / NIK / NIP	Nama	Jabatan	Plat Nomer	Tanggal	IN	OUT
1	102410101001	Anwar Syarifudin Fajri	Mahasiswa	M 3869 BF	2015-10-05	13:53:43	13:53:59
2	6574639576239254	Ini Nama Dosen	Dosen	P 3869 SX	2015-10-05	13:52:35	13:53:54

Gambar 5.2 Screenshot halaman data pengunjung

Pada gambar 5.2 dapat dijelaskan bahwa terdapat tampilan dari data pengunjung setelah *user* melakukan *scannersmartcard* terhadap rfid.

5.2.2 Daftar Baru

Fitur daftar baru hanya bisa diakses oleh petugas parkir. Pada fitur ini juga terdapat 3 fungsi yaitu tambah pengguna sistem, menghapus dan mengubah data pengguna sistem. Dalam fitur ini terdapat form yang wajib diisi secara lengkap oleh petugas parkir. Halaman Daftar Baru dapat dilihat pada gambar 5.3

The screenshot shows a web application interface for adding a new user. At the top, there are three checkboxes: "Data Pengunjung" (unchecked), "Daftar Baru" (checked), and "Rekap Pengunjung" (unchecked). A clock in the top right corner displays "06 : 36 : 30 PM". Below the header, there is a search bar and a "reset" button. The main form contains several input fields: "NIM / NIK / NIP", "Nama", "Jabatan" (with a dropdown menu open showing "Dosen", "Karyawan", "Mahasiswa", and "Tamu"), "Angkatan", "Merk Kendaraan", "Warna Kendaraan", and "Nomor Polisi". At the bottom, there is a table with columns for "No", "RFID", "NIM / NIK", "Jabatan", "Angkatan", "Plat Nomer", and "Status".

Gambar 5.3 Screenshot halaman daftar baru

Menu tambah pengguna berfungsi untuk menambah pengguna baru. Petugas parkir perlu menginputkan beberapa field yaitu NIM/NIK/NIP, nama, jabatan, angkatan, Merk Kendaraan, warna kendaraan dan Nomor polisi. Akan tetapi sebelum mengisi beberapa *field* tersebut petugas parkir harus menscan *smartcard* yang nantinya di berikan kepada pengguna baru. Tampilan menu tambah terdapat pada Gambar 5.4

No	RFID	NIM / NIK / NIP	Nama	Jabatan	Angkatan	Plat Nomer	Status
1	0000073002	19219319830110137	Andi	Dosen	-	P 1234 MZ	Aktif
2	0000072863	102410101001	Anwar Syarifuddin Fajri	Mahasiswa	2010	M 3869 BF	Aktif

Gambar 5.4 Halaman daftar baru pengguna

Tombol ubah dan hapus berfungsi untuk mengubah data pengguna yang sudah ada misalkan untuk mengganti identitas pengguna ataupun kendaraan sedangkan tombol hapus berfungsi untuk menghapus data yang sudah ada pada sistem misalnya kehilangan kartu *smartcard* atau pengguna sudah lulus. Untuk menjalankan fungsi ini adalah petugas parkir harus memilih data yang akan di hapus ataupun yang akan di ubah datanya namun sebelumnya petugas parkir harus menekan tombol reset terlebih dahulu. Tampilan halaman ubah dan hapus terdapat pada gambar 5.5

No	RFID	NIM / NIK / NIP	Nama	Jabatan	Angkatan	Plat Nomer	Status
1	0000073002	19219319830110137	Andi	Dosen	-	P 1234 MZ	Aktif
2	0000073002	131410101010	Ficki	Mahasiswa	2013	AD 5689 SU	Aktif
3	0000072863	102410101001	Anwar Syarifuddin Fajri	Mahasiswa	2010	M 3869 BF	Aktif

Gambar 5.5 Screenshot halaman ubah dan hapus pengguna
Listing program dari fitur Daftar Baru terdapat pada tabel 5.2.

Tabel 5.2 Listing Program Daftar Baru

```
private void pdb_tombol_addMouseClicked(java.awt.event.MouseEvent evt) {
    if (!pdb_edit_delete_status && cek_field()) {
        try {
            String query = "INSERT INTO user VALUES (NULL, '" +
                pdb_field_rfid.getText() + "', '" + pdb_field_nim.getText() + "', '" +
                pdb_field_nama.getText() + "', '" + pdb_field_jabatan.getSelectedItem() + "', '" +
                + pdb_field_angkatan.getText() + "', '" + pdb_field_merk.getText() + "', '" +
                pdb_field_warna.getText() + "', '" +
                pdb_field_no_polisi.getText().toUpperCase() + "', 1)";
            System.out.println(query);
            Statement st = kon.get_koneksi().createStatement();
            st.executeUpdate(query);
            reset_field();
        }
    }
}
```

```
        refresh_table1();
    } catch (Exception e) {
        System.out.println("error : " + e.getMessage());
    }
} else {
    JOptionPane.showMessageDialog(null, "Lengkapi data yang ada!",
"Warning", JOptionPane.WARNING_MESSAGE);
}
}
private void pdb_field_jabatanItemStateChanged(java.awt.event.ItemEvent
evt) {
    if (pdb_field_jabatan.getSelectedIndex() == 0) {
        pdb_field_angkatan.setText("-");
        pdb_field_angkatan.setEditable(false);
    } else if (pdb_field_jabatan.getSelectedIndex() == 1) {
        pdb_field_angkatan.setText("-");
        pdb_field_angkatan.setEditable(false);
    } else if (pdb_field_jabatan.getSelectedIndex() == 2) {
        pdb_field_angkatan.setText("");
        pdb_field_angkatan.setEditable(true);
    } else if (pdb_field_jabatan.getSelectedIndex() == 3) {
        pdb_field_angkatan.setText("-");
        pdb_field_angkatan.setEditable(false);
    }
}

private void pdb_tombol_editMouseClicked(java.awt.event.MouseEvent evt)
{
    if (pdb_edit_delete_status && cek_field()) {
        try {
```

```
String query = "UPDATE user SET user_nomer = " +
pdb_field_nim.getText()
+ ", user_nama = " + pdb_field_nama.getText() + ",
user_jabatan = " + pdb_field_jabatan.getSelectedItem()
+ ", user_angkatan = " + pdb_field_angkatan.getText() + ",
user_merk = " + pdb_field_merk.getText()
+ ", user_warna = " + pdb_field_warna.getText() + ",
user_nopol = " + pdb_field_no_polisi.getText()
+ " WHERE user_rfid = " + pdb_field_rfid.getText();

Statement st = kon.get_koneksi().createStatement();
st.executeUpdate(query);
reset_field();
refresh_table1();
} catch (Exception e) {
System.out.println("error : " + e.getMessage());
}
table1.clearSelection();
refresh_table1();
} else {
JOptionPane.showMessageDialog(null, "Lengkapi data yang ada!",
"Warning", JOptionPane.WARNING_MESSAGE);
}
}

public boolean cek_field() {
boolean r = false;
if (!pdb_field_nim.getText().equals("") &&
!pdb_field_nama.getText().equals("") &&
!pdb_field_angkatan.getText().equals("") &&
!pdb_field_merk.getText().equals("") &&
!pdb_field_warna.getText().equals("") &&
!pdb_field_no_polisi.getText().equals("") &&
```

```
        !pdb_field_rfid.getText().equals("")) {
            r = true;
        } else {
            r = false;
        }
        return r;
    }

    private void pdb_tombol_deleteMouseClicked(java.awt.event.MouseEvent
    evt) {
        if (pdb_edit_delete_status) {
            try {
                String query = "UPDATE user SET user_status = 0 WHERE
    user_rfid = " + table1.getValueAt(table1.getSelectedRow(), 1);
                //      System.out.println(query);
                Statement st = kon.get_koneksi().createStatement();
                st.executeUpdate(query);

                reset_field();
                refresh_table1();
            } catch (Exception e) {
                System.out.println("error : " + e.getMessage());
            }
            table1.clearSelection();
            refresh_table1();
        }
    }

    private void pdb_resetMouseClicked(java.awt.event.MouseEvent evt) {
        reset_field();

        pdb_edit_delete_status = false;
    }
}
```

```
        pdb_tombol_add.setEnabled(true);
        pdb_tombol_edit.setEnabled(false);
        pdb_tombol_delete.setEnabled(false);
    }

    private void table1MouseClicked(java.awt.event.MouseEvent evt) {
        pdb_edit_delete_status = true;

        pdb_tombol_add.setEnabled(false);
        pdb_tombol_edit.setEnabled(true);
        pdb_tombol_delete.setEnabled(true);

        try {
            String query = "SELECT * FROM user WHERE user_rfid = " +
table1.getValueAt(table1.getSelectedRow(), 1);
            Statement st = kon.get_koneksi().createStatement();
            ResultSet rs = st.executeQuery(query);

            if (rs.next()) {
                pdb_field_rfid.setText(rs.getString("user_rfid"));
                pdb_field_rfid.setEditable(false);
                pdb_field_nim.setText(rs.getString("user_nomer"));
                pdb_field_nama.setText(rs.getString("user_nama"));
                pdb_field_jabatan.setSelectedItem(rs.getString("user_jabatan"));
                pdb_field_angkatan.setText(rs.getString("user_angkatan"));
                pdb_field_merk.setText(rs.getString("user_merk"));
                pdb_field_warna.setText(rs.getString("user_warna"));
                pdb_field_no_polisi.setText(rs.getString("user_nopol"));
            }
        } catch (Exception e) {
```

```
        System.out.println("error : " + e.getMessage());
    }
}

private void pdb_field_cariKeyReleased(java.awt.event.KeyEvent evt) {
    if (evt.getKeyCode() == 10) {
        String query = "SELECT user_rfid, user_nomer, user_nama,
user_jabatan, user_angkatan, user_nopol, user_status FROM user"
            + " WHERE (user_rfid = '" + pdb_field_cari.getText() + "' OR
user_nama LIKE '%" + pdb_field_cari.getText() + "%') AND user_status = 1";
        try {
            String[] judul = {"No", "RFID", "NIM / NIK / NIP", "Nama",
"Jabatan", "Angkatan", "Plat Nomer", "Status"};
            Statement st = kon.get_koneksi().createStatement();
            ResultSet rs = st.executeQuery(query);
            DefaultTableModel model = new DefaultTableModel(null, judul);
            int urut = 0;
            if (rs.next()) {
                String user_rfid = rs.getString("user_rfid");
                String user_nomer = rs.getString("user_nomer");
                String user_nama = rs.getString("user_nama");
                String user_jabatan = rs.getString("user_jabatan");
                String user_angkatan = rs.getString("user_angkatan");
                String user_nopol = rs.getString("user_nopol");
                String user_status = rs.getString("user_status");
                Object[] d = {++urut, user_rfid, user_nomer, user_nama,
user_jabatan, user_angkatan, user_nopol, ((user_status.equals("1")) ? "Aktif" :
"Non-Aktif")};
                model.addRow(d);
            }
            table1.setModel(model);
        }
    }
}
```

```
        pdb_field_cari.setText("");
        if (table1.getColumnModel().getColumnCount() > 0) {
            table1.getColumnModel().getColumn(0).setResizable(false);
            table1.getColumnModel().getColumn(0).setPreferredWidth(25);
            table1.getColumnModel().getColumn(1).setResizable(false);
            table1.getColumnModel().getColumn(1).setPreferredWidth(95);
            table1.getColumnModel().getColumn(2).setResizable(false);
            table1.getColumnModel().getColumn(2).setPreferredWidth(135);
            table1.getColumnModel().getColumn(3).setResizable(false);
            table1.getColumnModel().getColumn(3).setPreferredWidth(160);
            table1.getColumnModel().getColumn(4).setResizable(false);
            table1.getColumnModel().getColumn(4).setPreferredWidth(75);
            table1.getColumnModel().getColumn(5).setResizable(false);
            table1.getColumnModel().getColumn(5).setPreferredWidth(85);
            table1.getColumnModel().getColumn(6).setResizable(false);
            table1.getColumnModel().getColumn(6).setPreferredWidth(85);
            table1.getColumnModel().getColumn(7).setResizable(false);
            table1.getColumnModel().getColumn(7).setPreferredWidth(68);
        }
    } catch (Exception e) {
        System.out.println("error : " + e.getMessage());
    }
}
}

private void reset_field() {
    pdb_field_cari.setText("");
    pdb_field_rfid.setText("");
    pdb_field_nim.setText("");
    pdb_field_nama.setText("");
    pdb_field_jabatan.setSelectedIndex(0);
    pdb_field_angkatan.setText("-");
}
```

```
    pdb_field_merk.setText("");
    pdb_field_warna.setText("");
    pdb_field_no_polisi.setText("");

    pdb_field_rfid.setEditable(true);
    pdb_field_rfid.requestFocus();
    table1.clearSelection();

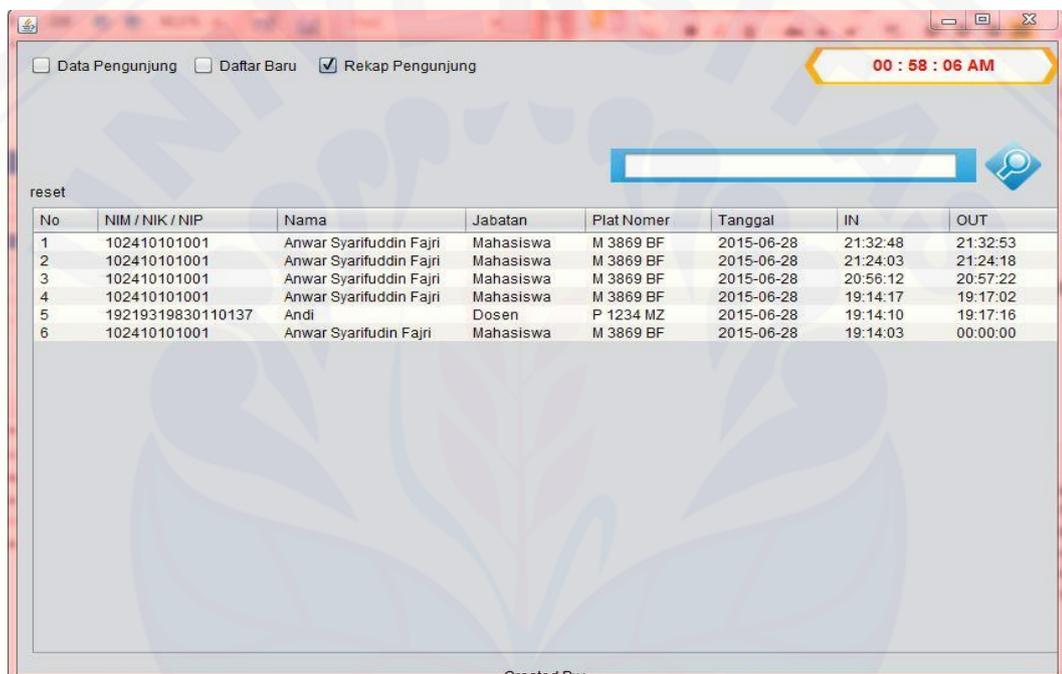
    refresh_table1();
}

private void refresh_table1() {
    try {
        String query = "SELECT user_rfid, user_nomer, user_nama,
user_jabatan, user_angkatan, user_nopol, user_status FROM user WHERE
user_status = 1 ORDER BY user_jabatan ASC, user_nomer DESC";
        String[] judul = {"No", "RFID", "NIM / NIK / NIP", "Nama", "Jabatan",
"Angkatan", "Plat Nomer", "Status"};
        Statement st = kon.get_koneksi().createStatement();
        ResultSet rs = st.executeQuery(query);
        DefaultTableModel model = new DefaultTableModel(null, judul);
        int urut = 0;
        while (rs.next()) {
            String user_rfid = rs.getString("user_rfid");
            String user_nomer = rs.getString("user_nomer");
            String user_nama = rs.getString("user_nama");
            String user_jabatan = rs.getString("user_jabatan");
            String user_angkatan = rs.getString("user_angkatan");
            String user_nopol = rs.getString("user_nopol");
            String user_status = rs.getString("user_status");
```

```
        Object[] d = {++urut, user_rfid, user_nomer, user_nama,
user_jabatan, user_angkatan, user_nopol, ((user_status.equals("1")) ? "Aktif" :
"Non-Aktif")};
        model.addRow(d);
    }
    table1.setModel(model);
    if (table1.getColumnModel().getColumnCount() > 0) {
        table1.getColumnModel().getColumn(0).setResizable(false);
        table1.getColumnModel().getColumn(0).setPreferredWidth(25);
        table1.getColumnModel().getColumn(1).setResizable(false);
        table1.getColumnModel().getColumn(1).setPreferredWidth(95);
        table1.getColumnModel().getColumn(2).setResizable(false);
        table1.getColumnModel().getColumn(2).setPreferredWidth(135);
        table1.getColumnModel().getColumn(3).setResizable(false);
        table1.getColumnModel().getColumn(3).setPreferredWidth(160);
        table1.getColumnModel().getColumn(4).setResizable(false);
        table1.getColumnModel().getColumn(4).setPreferredWidth(75);
        table1.getColumnModel().getColumn(5).setResizable(false);
        table1.getColumnModel().getColumn(5).setPreferredWidth(85);
        table1.getColumnModel().getColumn(6).setResizable(false);
        table1.getColumnModel().getColumn(6).setPreferredWidth(85);
        table1.getColumnModel().getColumn(7).setResizable(false);
        table1.getColumnModel().getColumn(7).setPreferredWidth(68);
    }
} catch (Exception e) {
    System.out.println("error : " + e.getMessage());
}
}
```

5.2.3 Rekap Pengunjung

Fitur rekap pengunjung hanya bisa diakses oleh petugas parkir. Pada fitur ini hanya menampilkan data *record* pengguna dari sistem. Pada fitur ini terdapat *field* cari yang berguna untuk mencari data yang ada pada sistem dengan cara menulis nama pada *shield* tersebut atau dengan cara menscan *smartcard* smartcard untuk mengetahui id RFID yang nantinya akan secara otomatis akan muncul data secara keseluruhan. Halaman dari fitur Rekap Pengunjung terdapat pada Gambar 5.6.



No	NIM / NIK / NIP	Nama	Jabatan	Plat Nomer	Tanggal	IN	OUT
1	102410101001	Anwar Syarifuddin Fajri	Mahasiswa	M 3869 BF	2015-06-28	21:32:48	21:32:53
2	102410101001	Anwar Syarifuddin Fajri	Mahasiswa	M 3869 BF	2015-06-28	21:24:03	21:24:18
3	102410101001	Anwar Syarifuddin Fajri	Mahasiswa	M 3869 BF	2015-06-28	20:56:12	20:57:22
4	102410101001	Anwar Syarifuddin Fajri	Mahasiswa	M 3869 BF	2015-06-28	19:14:17	19:17:02
5	19219319830110137	Andi	Dosen	P 1234 MZ	2015-06-28	19:14:10	19:17:16
6	102410101001	Anwar Syarifudin Fajri	Mahasiswa	M 3869 BF	2015-06-28	19:14:03	00:00:00

Gambar 5.6 Screenshot halaman rekap pengunjung

Listing program dari fitur Rekap Pengunjung terdapat pada tabel 5.3.

Tabel 5.3 Source code Rekap Pengunjung

```
Private void prp_field_cariKeyReleased(java.awt.event.KeyEvent evt) {
    if (evt.getKeyCode() == 10) {
        try {
            String query = "SELECT user_nomer, user_nama, user_jabatan,
user_nopol, DATE(masuk) AS tanggal, TIME(masuk) AS masuk,
TIME(keluar) AS keluar FROM user u, record r WHERE u.user_id = r.user_id
AND (user_rfid = " + prp_field_cari.getText() + " OR user_nama LIKE '%" +
```

```
prp_field_cari.getText() + "%') ORDER BY record_id DESC";

    String[] judul = {"No", "NIM / NIK / NIP", "Nama", "Jabatan", "Plat
Nomer", "Tanggal", "IN", "OUT"};

    Statement st = kon.get_koneksi().createStatement();
    ResultSet rs = st.executeQuery(query);

    DefaultTableModel model = new DefaultTableModel(null, judul);

    int urut = 0;
    while (rs.next()) {
        String user_nomer = rs.getString("user_nomer");
        String user_nama = rs.getString("user_nama");
        String user_jabatan = rs.getString("user_jabatan");
        String user_nopol = rs.getString("user_nopol");
        String tanggal = rs.getString("tanggal");
        String masuk = rs.getString("masuk");
        String keluar = rs.getString("keluar");

        Object[] d = {++urut, user_nomer, user_nama, user_jabatan,
user_nopol, tanggal, masuk, keluar};

        model.addRow(d);
    }

    table2.setModel(model);
    prp_field_cari.setText("");

    if (table2.getColumnModel().getColumnCount() > 0) {
        table2.getColumnModel().getColumn(0).setResizable(false);
        table2.getColumnModel().getColumn(0).setPreferredWidth(36);
        table2.getColumnModel().getColumn(1).setResizable(false);
```

```
        table2.getColumnModel().getColumn(1).setPreferredWidth(128);
        table2.getColumnModel().getColumn(2).setResizable(false);
        table2.getColumnModel().getColumn(2).setPreferredWidth(135);
        table2.getColumnModel().getColumn(3).setResizable(false);
        table2.getColumnModel().getColumn(3).setPreferredWidth(80);
        table2.getColumnModel().getColumn(4).setResizable(false);
        table2.getColumnModel().getColumn(4).setPreferredWidth(85);
        table2.getColumnModel().getColumn(5).setResizable(false);
        table2.getColumnModel().getColumn(5).setPreferredWidth(85);
        table2.getColumnModel().getColumn(6).setResizable(false);
        table2.getColumnModel().getColumn(6).setPreferredWidth(73);
        table2.getColumnModel().getColumn(7).setResizable(false);
        table2.getColumnModel().getColumn(7).setPreferredWidth(73);
    }
} catch (Exception e) {
    System.out.println("error : " + e.getMessage());
}
}
}
// dari ini
private void prp_resetMouseClicked(java.awt.event.MouseEvent evt) {
    refresh_table2();
}

private void refresh_table2() {
    try {
        String query = "SELECT user_nomer, user_nama, user_jabatan,
user_nopol, DATE(masuk) AS tanggal, TIME(masuk) AS masuk,
TIME(keluar) AS keluar FROM user u, record r WHERE u.user_id = r.user_id
ORDER BY record_id DESC";

        String[] judul = {"No", "NIM / NIK / NIP", "Nama", "Jabatan", "Plat
```

```
Nomer", "Tanggal", "IN", "OUT"};

Statement st = kon.get_koneksi().createStatement();
ResultSet rs = st.executeQuery(query);

DefaultTableModel model = new DefaultTableModel(null, judul);

int urut = 0;
while (rs.next()) {
    String user_nomer = rs.getString("user_nomer");
    String user_nama = rs.getString("user_nama");
    String user_jabatan = rs.getString("user_jabatan");
    String user_nopol = rs.getString("user_nopol");
    String tanggal = rs.getString("tanggal");
    String masuk = rs.getString("masuk");
    String keluar = rs.getString("keluar");

    Object[] d = {++urut, user_nomer, user_nama, user_jabatan,
user_nopol, tanggal, masuk, keluar};

    model.addRow(d);
}

table2.setModel(model);

if (table2.getColumnModel().getColumnCount() > 0) {
    table2.getColumnModel().getColumn(0).setResizable(false);
    table2.getColumnModel().getColumn(0).setPreferredWidth(36);
    table2.getColumnModel().getColumn(1).setResizable(false);
    table2.getColumnModel().getColumn(1).setPreferredWidth(128);
    table2.getColumnModel().getColumn(2).setResizable(false);
```

```
table2.getColumnModel().getColumn(2).setPreferredWidth(135);
table2.getColumnModel().getColumn(3).setResizable(false);
table2.getColumnModel().getColumn(3).setPreferredWidth(80);
table2.getColumnModel().getColumn(4).setResizable(false);
table2.getColumnModel().getColumn(4).setPreferredWidth(85);
table2.getColumnModel().getColumn(5).setResizable(false);
table2.getColumnModel().getColumn(5).setPreferredWidth(85);
table2.getColumnModel().getColumn(6).setResizable(false);
table2.getColumnModel().getColumn(6).setPreferredWidth(73);
table2.getColumnModel().getColumn(7).setResizable(false);
table2.getColumnModel().getColumn(7).setPreferredWidth(73);
}
} catch (Exception e) {
    System.out.println("error : " + e.getMessage());
}
}
```

BAB 6. PENUTUP

6.1 Kesimpulan

- ^ Kesimpulan dalam penelitian ini adalah sebagai berikut :
- Sistem informasi perparkiran kendaraan roda dua menggunakan teknologi RFID ini dibuat untuk menggantikan sistem perparkiran yang sebelumnya menggunakan cara manual sistem informasi ini berbasis desktop dan diintegrasikan dengan *smartcard* untuk menjalankannya
 - Sistem Informasi ini untuk memberikan keamanan dan kenyamanan bagi pengguna dan juga petugas parkir

6.2 Saran

Saran dan masukan berikut yang memberikan perbaikan dalam penelitian selanjutnya adalah :

- Dapat dilakukan perbandingan dengan metode perparkiran yang lain untuk mendapatkan gambaran sistem yang lebih sesuai
- Membuat sistem parkir kendaraan menggunakan teknologi RFID yang berbasis *Web*
- Dapat dilakukan penelitian untuk mengkombinasikan sistem parkir yang menggunakan teknologi RFID dengan pendeteksi lokasi. Sehingga setiap kendaraan roda dua yang akan masuk ke area parkir bisa mengetahui area parkir mana yang belum terisi.

DAFTAR PUSTAKA

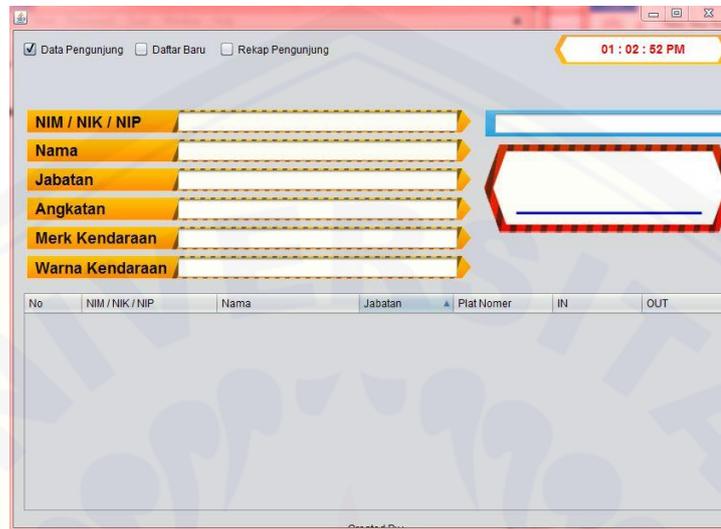
- Ahson, Syed. A, Ilyas. M. 2008. *RFID Handbook: Applications, Technology, Security, and Privacy*. New York : Taylor & Francis Group.
- Brown, Dennis. 2005. *RFID Implementation*. McGraw-Hill Companies
- Capra F., 2004. *Titik Balik Peradaban-Sains, Masyarakat, dan Kebangkitan Kebudayaan*. Terjemahan dari buku *The Turning Point* oleh Thoyibi M., Yogyakarta: Penerbit Bentang Pustaka
- Departemen Pendidikan dan Kebudayaan. 1990. *Kamus Besar Bahasa Indonesia*. Jakarta: Balai Pustaka.
- Direktorat Perhubungan Darat. 1996. *Pedoman Teknis Penyelenggaraan Fasilitas Parkir*. Jakarta : Direktorat Perhubungan Darat
- Finkenzeller, Klaus. 2003. *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification*. London: John Wiley & Sons.
- Furchan, Arief. (2007). *Pengantar Penelitian dalam Pendidikan*. Malang : Pustaka Pelajar.
- Garfinkel, Simson, Beth Rosenberg. 2005. *RFID: Applications, Security, and Privacy*. USA: Prattice Hall
- Hanifah, Amalia; dkk. 2010. *Aplikasi Smartcard sebagai Pengunci Elektronik pada Smart Home*. Semarang: Universitas Diponegoro
- Henlia. 2006. *Mengenal RFID*. Bandung : ITB
- Jogiyanto. 2007. *Metode Penelitian Bisnis : Salah Kaprah dan Pengalaman - pengalaman*. Yogyakarta : BPF
- Karygiannis, T. 2007. *Guidance Securing Radio Frequency Identification (RFID) System*. Csrt.Nist.Gov/Publications/Nistpubs/800-98/Sp800-98_Rfid-2007.Pdf. D [16 November 2014]
- Kusuma, 2007. *Pengaman Rumah Menggunakan Kunci Elektronik Berbasis Radio Frequency Identification*.
- Maryono. 2005. *Dasar-dasar Radio Frequency Identification (RFID) Teknologi Yang Berpengaruh di Perpustakaan*. Yogyakarta: Media Informasi 205 Vol. XIV No.20.

- Miller, P. G., Strang, J., & Miller, P. M. 2010. *Addiction Reseach Methods*. Oxford: Blackwell Publishing Ltd
- Miles, dkk. 2008. *RFID Technology and Applications*. New York : Cambridge University Press.
- Nawawi, Hadari dan Mimi Martini. 1994. *Penelitian Terapan*. Yogyakarta: Gajah Mada University Press.
- Pandian, Paul. 2009. *RFID For Libraries: A Practical Guide*. New York: Chandos Publishing.
- Pressman, R.S. 2010. *Software Engineering : a practitioner's approach*. New York : McGraw-Hill
- Riduwan. 2010. *Skala Pengukuran Variabel-Variabel Penelitian*. Bandung: Alfabeta
- Sariasih, Christine. 1999. *Rancangan Keamanan Data Sistem Smartcard Kesehatan sesuai kebutuhan di Indonesia*. Depok: Universitas Indonesia
- Sugiyono. 2011. *Metode Penelitian Kuantitatif kualitatif dan R&D*. Bandung: Alfabeta
- Sukmadinata, Syaodih Nana. 2006. *Metode Penelitian Pendidikan*. Bandung : Remaja Rosdakarya.
- Supandri, Mohamad. 2004. *Konsep Keamanan Pada Radio Frequency Identification*. Bandung: ITB
- Want, Roy. 2004. *Enabling Ubiquitous Sensing with RFID*, Computer, vol. 37, no. 4, 2004, pp. 84–86.
- Weinstein, Ron. 2005. *RFID: A Technical Overview and Its Application to the Enterprise*. IT Professional, vol. 7, no. 3, pp. 27-33.



LAMPIRAN A. SCREENSHOT PROGRAM

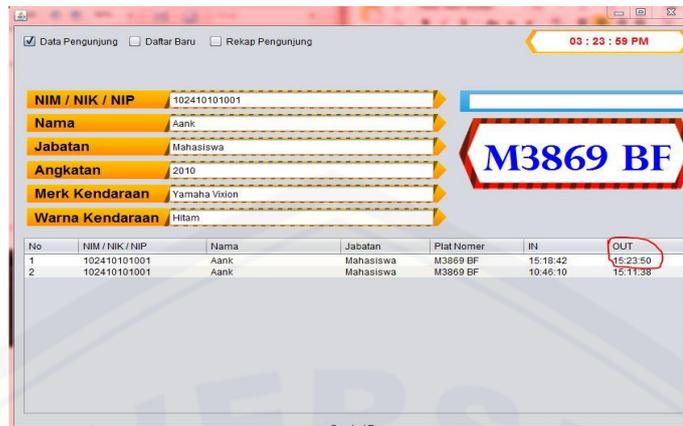
Lampiran A1. User Interface Data Pengunjung



Sumber : Hasil analisis 2015

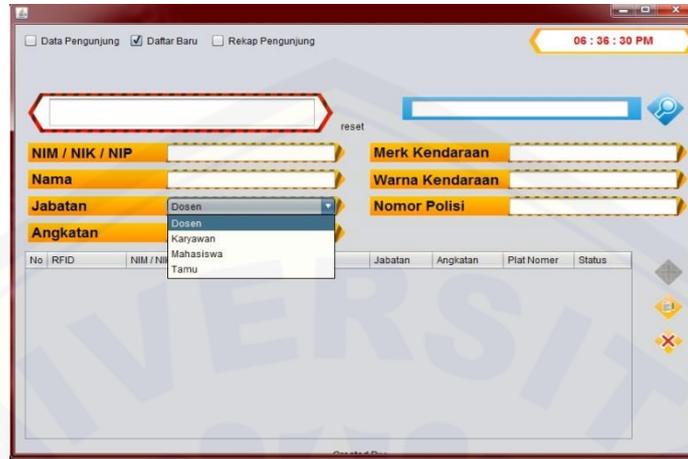


Gambar A.2 User Interface Data Pengunjung masuk

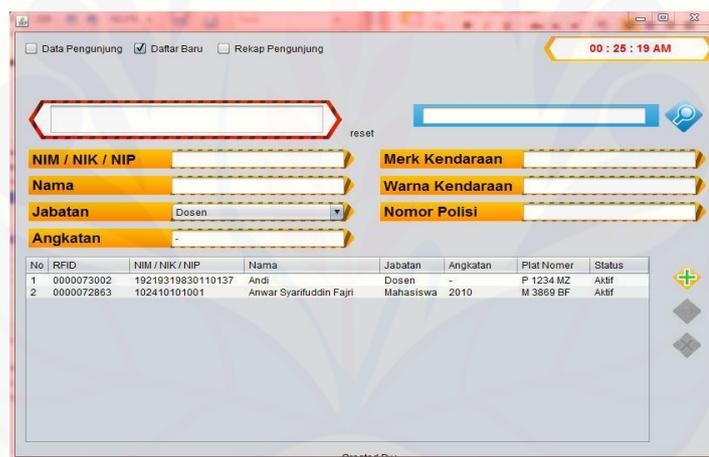


Gambar A3. User interface Data Pengunjung Keluar

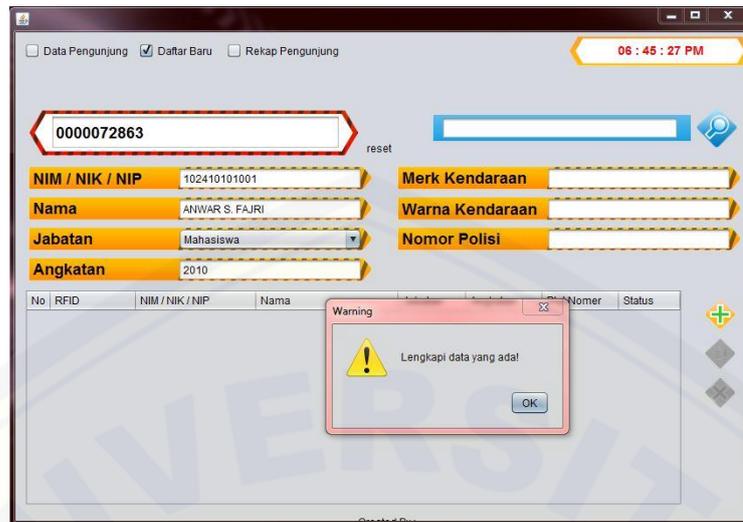
Lampiran A2. *User interface* Daftar Baru



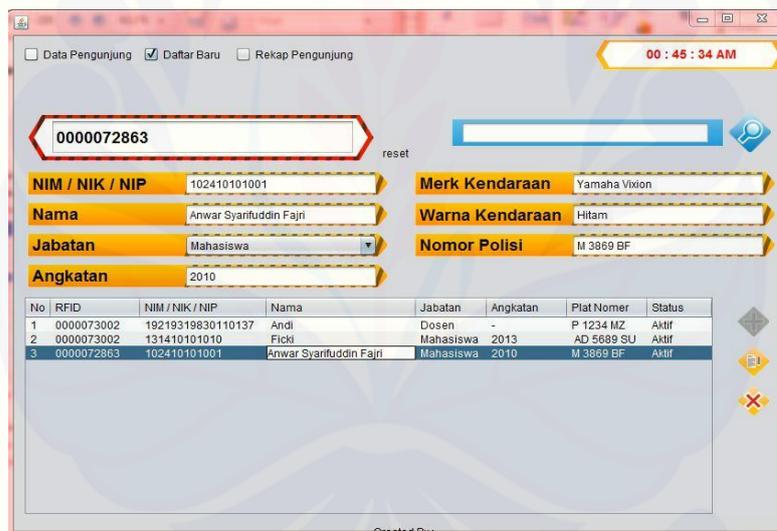
Sumber Analisis : 2015



Gambar A5 *User interface* Tambah Pengguna Sistem



Gambar A6. *User interface* Tambah Pengguna Sistem Gagal (Form Tidak Diisi Lengkap)



Gambar A7. *User interface* Ubah dan hapus Data Pengguna

Data Pengunjung Daftar Baru Rekap Pengunjung 06 : 36 : 30 PM

NIM / NIK / NIP **Merk Kendaraan**

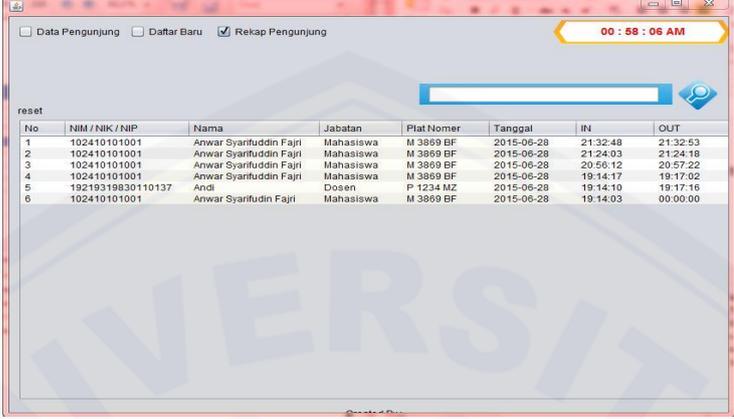
Nama **Warna Kendaraan**

Jabatan **Nomor Polisi**

Angkatan

No	RFID	NIM / NIK	Jabatan	Angkatan	Plat Nomer	Status
			Dosen			
			Dosen			
			Karyawan			
			Mahasiswa			
			Tamu			

Gambar A8. *User interface* Memilih Jabatan (Dosen, Mahasiswa, Tamu dan Karyawan)

Lampiran A3. Rekap Pengunjung

00 : 58 : 06 AM

reset

No	NIM / NIK / NIP	Nama	Jabatan	Plat Nomer	Tanggal	IN	OUT
1	102410101001	Anwar Syarifuddin Fajri	Mahasiswa	M 3869 BF	2015-06-28	21:32:48	21:32:53
2	102410101001	Anwar Syarifuddin Fajri	Mahasiswa	M 3869 BF	2015-06-28	21:24:03	21:24:18
3	102410101001	Anwar Syarifuddin Fajri	Mahasiswa	M 3869 BF	2015-06-28	20:56:12	20:57:22
4	102410101001	Anwar Syarifuddin Fajri	Mahasiswa	M 3869 BF	2015-06-28	19:14:17	19:17:02
5	19219319830110137	Andi	Dosen	P 1234 MZ	2015-06-28	19:14:10	19:17:16
6	102410101001	Anwar Syarifuddin Fajri	Mahasiswa	M 3869 BF	2015-06-28	19:14:03	00:00:00

Sumber Analisis 2015

LAMPIRAN B. PENGUJIAN SISTEM

Lampiran B1. Pengujian Sistem

a. Listing Program

Tahap pertama pada pengujian white box yaitu dengan cara memberikan penomoran pada program berdasarkan statemen dan logika dari setiap fitur yang ada pada sistem yang dapat dilihat pada gambar B.1

```

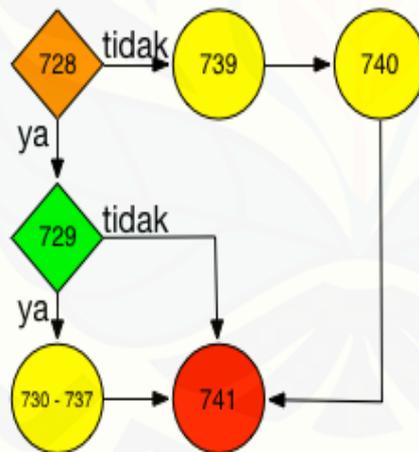
728 private void set_field(ResultSet rs) {
729     try {
730         while (rs.next()) {
731             pdp_field_nim.setText(rs.getString("user_nomer"));
732             pdp_field_nama.setText(rs.getString("user_nama"));
733             pdp_field_jabatan.setText(rs.getString("user_jabatan"));
734             pdp_field_angkatan.setText(rs.getString("user_angkatan"));
735             pdp_field_merk.setText(rs.getString("user_merk"));
736             pdp_field_warna.setText(rs.getString("user_warna"));
737             pdp_label_no_polisi.setText(rs.getString("user_nopol"));
738         }
739     } catch (Exception e) {
740         Logger.getLogger(Parkir.class.getName()).log(Level.SEVERE, null, e);
741     }
742 }

```

Diagram annotations in the image: A bracket labeled '1' spans the try block. A bracket labeled '2' spans the while loop body. A bracket labeled '3' spans the catch block.

Gambar B1 Listing Program untunk set_field

b. Grafik Alir



Gambar B2. Grafik Alir

c. Kompleksitas Siklomatis pada set_field

Perhitunga Kompleksitas Siklomatis diperoleh dari rumus

$$V(G) = E - N + 2 = 7 - 6 + 2 = 3$$

Lampiran B2. Pengujian Pada Menu Data Parkir keluar dan Masuk

a. Listing Program Menu Data Parkir keluar dan Masuk

Tahap pertama pada pengujian white box yaitu dengan cara memberikan penomoran pada program berdasarkan statemen dan logika dari setiap fitur yang ada pada sistem yang dapat dilihat pada gambar B.3

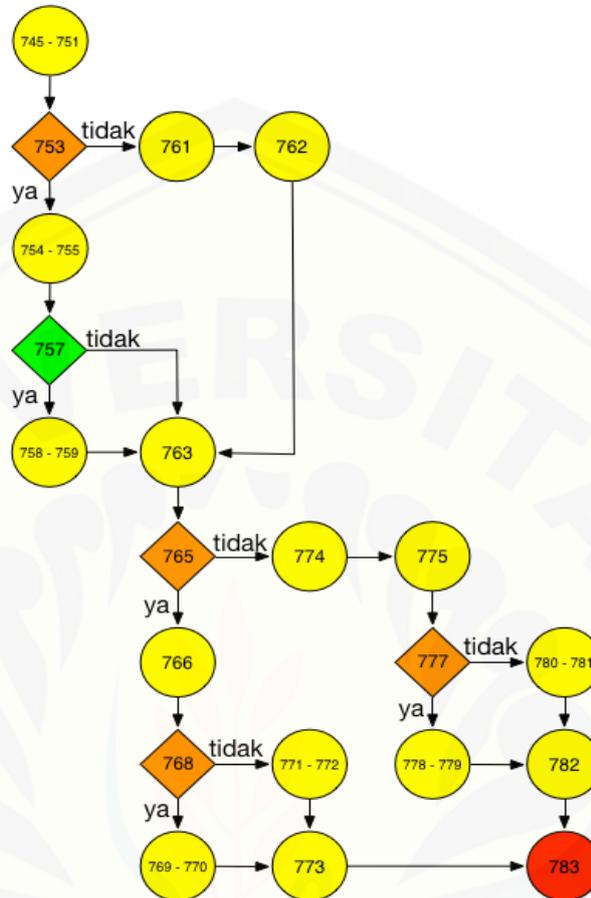
```

744 private void insert_parkir() {
745     String query = "SELECT r.record_id"
746         + " FROM user u, record r"
747         + " WHERE user_rfid = '" + pdp_field_rfid.getText() + "' AND use
748         + "         AND u.user_id = r.user_id AND DATE(masuk) = DATE(NOW())" }1
749
750     int baris = 0;
751     String id = "";
752
753     try { }2
754         Statement st = kon.get_koneksi().createStatement(); }3
755         ResultSet rs = st.executeQuery(query);
756
757         while (rs.next()) { }4
758             id = rs.getString("record_id"); }5
759             baris++;
760         }
761     } catch (Exception e) { }6
762         System.out.println("pesan sistem 1 : " + e.getMessage()); }7
763     }
764
765     if (baris == 0) { }8
766         String query2 = "INSERT INTO record VALUES (NULL, (SELECT user_id FR" }9
767
768         try { }10
769             Statement st = kon.get_koneksi().createStatement(); }11
770             st.executeUpdate(query2);
771         } catch (Exception e) { }12
772             System.out.println("pesan sistem 2 : " + e.getMessage()); }13
773         }
774     } else { }14
775         String query2 = "UPDATE record SET keluar = NOW() WHERE record_id = " }15
776
777         try { }16
778             Statement st = kon.get_koneksi().createStatement(); }17
779             st.executeUpdate(query2);
780         } catch (Exception e) { }18
781             System.out.println("pesan sistem 3 : " + e.getMessage()); }19
782         }
783     }

```

Gambar B3. Listing program Menu Data Parkir keluar dan Masuk

b. Grafik Alir



Gambar B4. Grafik Alir Menu Data Parkir keluar dan Masuk

c. Kompleksitas Siklomatis pada set_field

Perhitungan Kompleksitas Siklomatis diperoleh dari rumus

$$V(G) = E - N + 2 = 25 - 21 + 2 = 6$$

Lampiran B3. Pengujian pada Refresh Table

a. Listing Program pada Refresh Table

Tahap pertama pada pengujian white box yaitu dengan cara memberikan penomoran pada program berdasarkan statemen dan logika dari setiap fitur yang ada pada sistem yang dapat dilihat pada gambar B.5

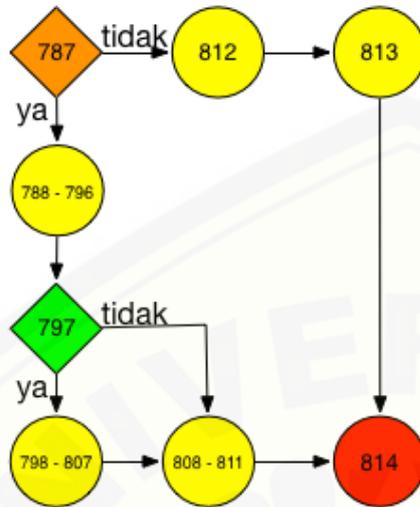
```

787     try {}1
788         String query = "SELECT user_nomer, user_nama, user_jabatan, user_nopol";
789         String[] judul = {"No", "NIM / NIK / NIP", "Nama", "Jabatan", "Plat No"};
790
791         Statement st = kon.get_koneksi().createStatement();
792         ResultSet rs = st.executeQuery(query);
793
794
795         DefaultTableModel model = new DefaultTableModel(null, judul);
796
797         int urut = 0;
798         while (rs.next()) {}3
799             String user_nomer = rs.getString("user_nomer");
800             String user_nama = rs.getString("user_nama");
801             String user_jabatan = rs.getString("user_jabatan");
802             String user_nopol = rs.getString("user_nopol");
803             String masuk = rs.getString("masuk");
804             String keluar = rs.getString("keluar");
805
806             Object[] d = {++urut, user_nomer, user_nama, user_jabatan, user_nopol, masuk, keluar};
807
808             model.addRow(d);
809         }
810         table.setModel(model); }5
811
812         if (table.getColumnModel().getColumnCount() > 0) {
813             table.getColumnModel().getColumn(0).setResizable(false);
814             table.getColumnModel().getColumn(0).setPreferredWidth(36);
815             table.getColumnModel().getColumn(1).setResizable(false);
816             table.getColumnModel().getColumn(1).setPreferredWidth(128);

```

Gambar B5. Listing menu Refresh Table

b. Grafik Alir



Gambar B6. Grafik Alir Refresh Table

c. Kompleksitas Siklomatis pada Refresh Table

Perhitungan Kompleksitas Siklomatis diperoleh dari rumus

$$V(G) = E - N + 2 = 9 - 8 + 2 = 3$$

Lampiran B4. Pengujian Pada Tombol Tambah user

a. Listing program pada Tombol tambah User

Tahap pertama pada pengujian white box yaitu dengan cara memberikan penomoran pada program berdasarkan statemen dan logika dari setiap fitur yang ada pada sistem yang dapat dilihat pada gambar B.7

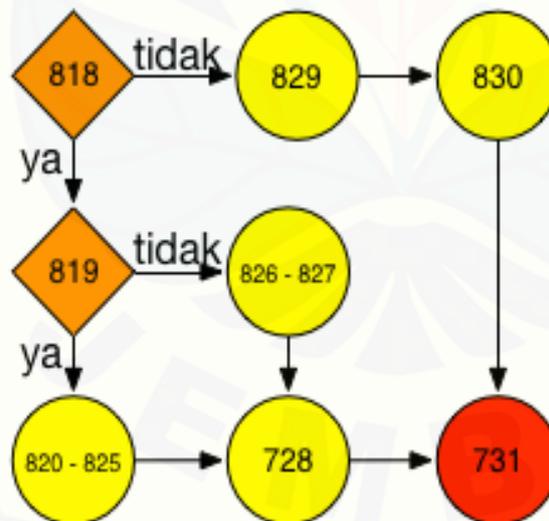
```

833 private void pdb_tombol_addMouseClicked(java.awt.event.MouseEvent evt) {
834     if (!pdb_edit_delete_status && cek_field()) { } 1
835     try { } 2
836         String query = "INSERT INTO user VALUES (NULL, " + pdb_field_rf";
837         System.out.println(query);
838         Statement st = kon.get_koneksi().createStatement();
839         st.executeUpdate(query);
840
841         reset_field();
842         refresh_table1();
843     } catch (Exception e) { } 4
844         System.out.println("error : " + e.getMessage()); } 5
845     }
846 } else { } 6
847 JOptionPane.showMessageDialog(null, "Lengkapi data yang ada!", "Warna"); } 7
848 }

```

Gambar B7. Listing Program Tombol Tambah User

b. Grafik Alir Pada Tombol Tambah User



Gambar B8. Grafik Alir tombol tambah user

c. Kompleksitas Siklomatis pada Tombol tambah user

Perhitunga Kompleksitas Siklomatis diperoleh dari rumus

$$V(G) = E - N + 2 = 9 - 8 + 2 = 3$$

Lampiran B5. Pengujian Pada tombol Edit User

a. Listing Program pada Tombol Edit

Tahap pertama pada pengujian white box yaitu dengan cara memberikan penomoran pada program berdasarkan statemen dan logika dari setiap fitur yang ada pada sistem yang dapat dilihat pada gambar B.9

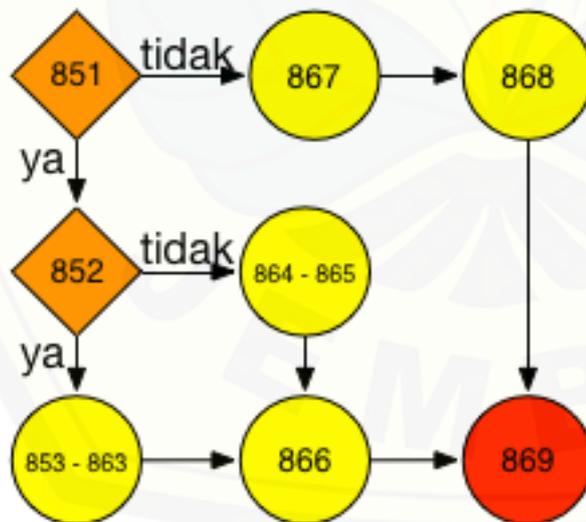
```

867 private void pdb_tombol_editMouseClicked(java.awt.event.MouseEvent evt) {
868     if (pdb_edit_delete_status && cek_field()) { }1
869     try {}2
870         String query = "UPDATE user SET user_nomer = '" + pdb_field_nim.
871             + "', user_nama = '" + pdb_field_nama.getText() + "', us
872             + "', user_angkatan = '" + pdb_field_angkatan.getText()
873             + "', user_warna = '" + pdb_field_warna.getText() + "',
874             + "' WHERE user_rfid = '" + pdb_field_rfid.getText();
875         Statement st = kon.get_koneksi().createStatement();
876         st.executeUpdate(query);
877
878         reset_field();
879         refresh_table1();
880     } catch (Exception e) { }4
881         System.out.println("error : " + e.getMessage()); }5
882     }
883     table1.clearSelection(); }6
884     refresh_table1();
885 } else { }7
886     JOptionPane.showMessageDialog(null, "Lengkapi data yang ada!", "Warna }8
887 }

```

Gambar B.9 Listing program tombol edit user

b. Grafik Alir tombol Edit User



Gambar B10. Grafik Alir Tombol Edit User

c. Kompleksitas Siklomatis pada Tombol Edit user

Perhitungan Kompleksitas Siklomatis diperoleh dari rumus

$$V(G) = E - N + 2 = 9 - 8 + 2 = 3$$



Lampiran B6. Pengujian Pada tombol Delete User

a. Listing Program pada Tombol Delete

Tahap pertama pada pengujian white box yaitu dengan cara memberikan penomoran pada program berdasarkan statemen dan logika dari setiap fitur yang ada pada sistem yang dapat dilihat pada gambar B.11

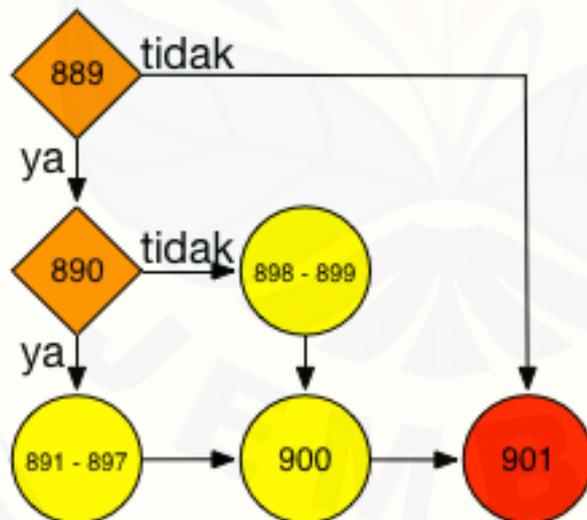
```

906 private void pdb_tombol_deleteMouseClicked(java.awt.event.MouseEvent evt) {
907     if (pdb_edit_delete_status) {1
908         try {2
909             String query = "UPDATE user SET user_status = 0 WHERE user_rfid
910             // System.out.println(query);
911             Statement st = kon.get_koneksi().createStatement();
912             st.executeUpdate(query);
913
914             reset_field();
915             refresh_table1();
916         } catch (Exception e) {4
917             System.out.println("error : " + e.getMessage());5
918         }
919         table1.clearSelection();6
920         refresh_table1();
921     }

```

Gambar B.11 Listing program tombol Delete

b. Grafik Alir tombol Delete



Gambar B12. Grafik Alir Tombol Delete User

c. Kompleksitas Siklomatis pada Tombol Edit user

Perhitunga Kompleksitas Siklomatis diperoleh dari rumus

$$V(G) = E - N + 2 = 7 - 6 + 2 = 3$$

Lampiran B7. Pengujian Pada tombol Reset pada fitur Daftar Baru User

a. Listing Program pada Tombol Reset pada fitur Daftar Baru User

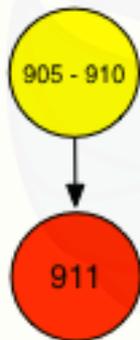
Tahap pertama pada pengujian white box yaitu dengan cara memberikan penomoran pada program berdasarkan statemen dan logika dari setiap fitur yang ada pada sistem yang dapat dilihat pada gambar B.13

```

924 private void pdb_resetMouseClicked(java.awt.event.MouseEvent evt) {
925     reset_field();
926
927     pdb_edit_delete_status = false;
928
929     pdb_tombol_add.setEnabled(true);
930     pdb_tombol_edit.setEnabled(false);
931     pdb_tombol_delete.setEnabled(false);
932 }
  
```

Gambar B.13 Listing Program Tombol reset pada fiur Daftar baru user

b. Grafik alir tombol Reset pada fitur Daftar Baru



Gambar B14. Grafik Alir Tombol Reset

c. Kompleksitas Siklomatis pada Tombol Reset

Perhitunga Kompleksitas Siklomatis diperoleh dari rumus

$$V(G) = E - N + 2 = 1 - 2 + 2 = 1$$

Lampiran B8. Pengujian Pada Daftar Baru mengisi form Table

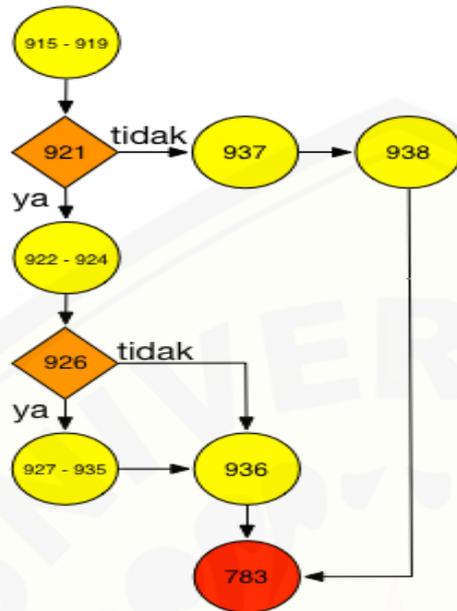
a. Listing Program pada Daftar Baru mengisi Form

Tahap pertama pada pengujian white box yaitu dengan cara memberikan penomoran pada program berdasarkan statemen dan logika dari setiap fitur yang ada pada sistem yang dapat dilihat pada gambar B.15

```
934 private void tableMouseClicked(java.awt.event.MouseEvent evt) {  
935     pdb_edit_delete_status = true;  
936  
937     pdb_tombol_add.setEnabled(false);  
938     pdb_tombol_edit.setEnabled(true);  
939     pdb_tombol_delete.setEnabled(true);  
940  
941     try { } 2  
942         String query = "SELECT * FROM user WHERE user_rfid = " + table1.getVa  
943         Statement st = kon.get_koneksi().createStatement();  
944         ResultSet rs = st.executeQuery(query);  
945  
946         if (rs.next()) { } 4  
947             pdb_field_rfid.setText(rs.getString("user_rfid"));  
948             pdb_field_rfid.setEditable(false);  
949             pdb_field_nim.setText(rs.getString("user_nomer"));  
950             pdb_field_nama.setText(rs.getString("user_nama"));  
951             pdb_field_jabatan.setSelectedItem(rs.getString("user_jabatan"));  
952             pdb_field_angkatan.setText(rs.getString("user_angkatan"));  
953             pdb_field_merk.setText(rs.getString("user_merk"));  
954             pdb_field_warna.setText(rs.getString("user_warna"));  
955             pdb_field_no_polisi.setText(rs.getString("user_nopol"));  
956         }  
957     } catch (Exception e) { } 6  
958         System.out.println("error : " + e.getMessage()); } 7  
959     }
```

Gambar B15. Listing Program Daftar baru mengisi Form

b. Grafik Alir Daftar baru mengisi Form



Gambar B 16. Grafik Alir Daftar Baru mengisi Form

c. Kompleksitas Siklomatis pada Daftar Baru mengisi Form

Perhitungan Kompleksitas Siklomatis diperoleh dari rumus

$$V(G) = E - N + 2 = 10 - 8 + 2 = 4$$

Lampiran B9. Pengujian Pada Fitur Daftar Baru pada tombol Cari

a. Listing Program pada Daftar Baru pada tombol Cari

Tahap pertama pada pengujian white box yaitu dengan cara memberikan penomoran pada program berdasarkan statemen dan logika dari setiap fitur yang ada pada sistem yang dapat dilihat pada gambar B.16

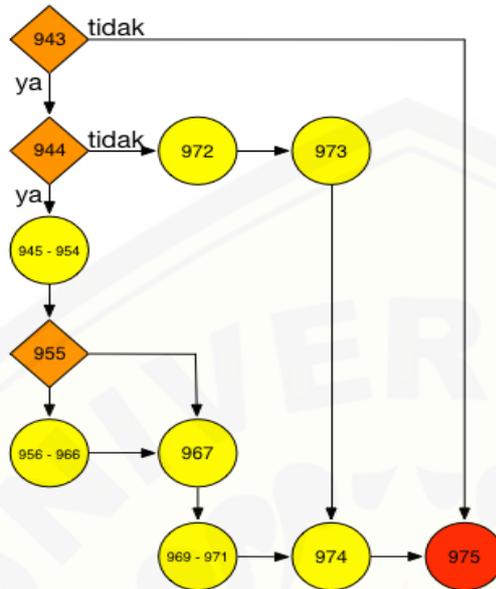
```

962 private void pdb_field_cariKeyReleased(java.awt.event.KeyEvent evt) {
963     if (evt.getKeyCode() == 10) { } 1
964     String query = "SELECT user_rfid, user_nomer, user_nama, user_jabata
965                   + " WHERE (user_rfid = '" + pdb_field_cari.getText() + "
966     try { } 3
967         String[] judul = {"No", "RFID", "NIM / NIK / NIP", "Nama", "Jaba
968
969         Statement st = kon.get_koneksi().createStatement();
970         ResultSet rs = st.executeQuery(query);
971
972         DefaultTableModel model = new DefaultTableModel(null, judul);
973
974         int urut = 0; } 5
975         if (rs.next()) { } 6
976             String user_rfid = rs.getString("user_rfid");
977             String user_nomer = rs.getString("user_nomer");
978             String user_nama = rs.getString("user_nama");
979             String user_jabatan = rs.getString("user_jabatan");
980             String user_angkatan = rs.getString("user_angkatan");
981             String user_nopol = rs.getString("user_nopol");
982             String user_status = rs.getString("user_status");
983
984             Object[] d = {++urut, user_rfid, user_nomer, user_nama, user
985
986             model.addRow(d);
987
988
989         table1.setModel(model);
990         pdb_field_cari.setText("");
991
992         table1.setModel(model);
993         pdb_field_cari.setText("");
994
995         if (table1.getColumnModel().getColumnCount() > 0) {
996             table1.getColumnModel().getColumn(0).setResizable(false);
997             table1.getColumnModel().getColumn(0).setPreferredWidth(25);
998             table1.getColumnModel().getColumn(1).setResizable(false);
999             table1.getColumnModel().getColumn(1).setPreferredWidth(95);
1000             table1.getColumnModel().getColumn(2).setResizable(false);
1001             table1.getColumnModel().getColumn(2).setPreferredWidth(135);
1002             table1.getColumnModel().getColumn(3).setResizable(false);
1003             table1.getColumnModel().getColumn(3).setPreferredWidth(160);
1004             table1.getColumnModel().getColumn(4).setResizable(false);
1005             table1.getColumnModel().getColumn(4).setPreferredWidth(75);
1006             table1.getColumnModel().getColumn(5).setResizable(false);
1007             table1.getColumnModel().getColumn(5).setPreferredWidth(85);
1008             table1.getColumnModel().getColumn(6).setResizable(false);
1009             table1.getColumnModel().getColumn(6).setPreferredWidth(85);
1010             table1.getColumnModel().getColumn(7).setResizable(false);
1011             table1.getColumnModel().getColumn(7).setPreferredWidth(68);
1012         }
1013     } catch (Exception e) { } 9
1014         System.out.println("error : " + e.getMessage()); } 10
1015     }

```

Gambar B16. Listing Program Fitur Daftar Baru Tombol Cari

b. Grafik Alir Fitur Daftar baru tombol cari



Gambar B17. Grafik Alir Fitur Daftar Baru tombol Cari

- c. Kompleksitas Siklomatis pada Daftar Baru tombol Cari
Perhitunga Kompleksitas Siklomatis diperoleh dari rumus
 $V(G) = E - N + 2 = 13 - 11 + 2 = 4$

Lampiran B10. Pengujian Pada Fitur Rekap Data pada tombol Cari

a. Listing Program pada Rekap Data pada tombol Cari

Tahap pertama pada pengujian white box yaitu dengan cara memberikan penomoran pada program berdasarkan statemen dan logika dari setiap fitur yang ada pada sistem yang dapat dilihat pada gambar B.18

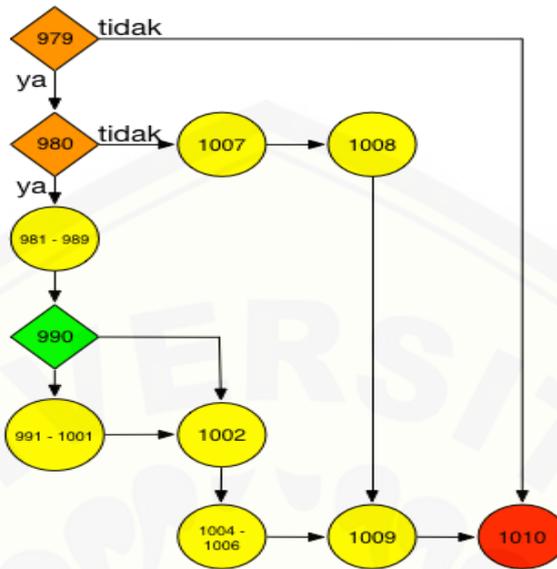
```

1016 private void prp_field_cariKeyReleased(java.awt.event.KeyEvent evt) {
1017     if (evt.getKeyCode() == 10) }-1
1018     try { }-2
1019         String query = "SELECT user_nomer, user_nama, user_jabatan, user_";
1020         String[] judul = {"No", "NIM / NIK / NIP", "Nama", "Jabatan", "E";
1021
1022         Statement st = kon.get_koneksi().createStatement();
1023         ResultSet rs = st.executeQuery(query);
1024
1025         DefaultTableModel model = new DefaultTableModel(null, judul);
1026
1027         int urut = 0;
1028         while (rs.next()) }-4
1029             String user_nomer = rs.getString("user_nomer");
1030             String user_nama = rs.getString("user_nama");
1031             String user_jabatan = rs.getString("user_jabatan");
1032             String user_nopol = rs.getString("user_nopol");
1033             String tanggal = rs.getString("tanggal");
1034             String masuk = rs.getString("masuk");
1035             String keluar = rs.getString("keluar");
1036
1037             Object[] d = {++urut, user_nomer, user_nama, user_jabatan, u;
1038
1039             model.addRow(d);
1040         }
1041
1042         table2.setModel(model);
1043         prp_field_cari.setText("");
1044
1045         if (table2.getColumnModel().getColumnCount() > 0) {
1046             table2.getColumnModel().getColumn(0).setResizable(false);
1047             table2.getColumnModel().getColumn(0).setPreferredWidth(36);
1048             table2.getColumnModel().getColumn(1).setResizable(false);
1049             table2.getColumnModel().getColumn(1).setPreferredWidth(128);
1050             table2.getColumnModel().getColumn(2).setResizable(false);
1051             table2.getColumnModel().getColumn(2).setPreferredWidth(135);
1052             table2.getColumnModel().getColumn(3).setResizable(false);
1053             table2.getColumnModel().getColumn(3).setPreferredWidth(80);
1054             table2.getColumnModel().getColumn(4).setResizable(false);
1055             table2.getColumnModel().getColumn(4).setPreferredWidth(85);
1056             table2.getColumnModel().getColumn(5).setResizable(false);
1057             table2.getColumnModel().getColumn(5).setPreferredWidth(85);
1058             table2.getColumnModel().getColumn(6).setResizable(false);
1059             table2.getColumnModel().getColumn(6).setPreferredWidth(73);
1060             table2.getColumnModel().getColumn(7).setResizable(false);
1061             table2.getColumnModel().getColumn(7).setPreferredWidth(73);
1062         }
1063     } catch (Exception e) }-7
1064         System.out.println("error : " + e.getMessage()); }-8
1065 }

```

Gambar B18. Listing Program Pada fitur Rekap data tombol cari

b. Grafik Alir Fitur Rekap data tombol cari



Gambar B19. Grafik Alir Fitur Rekap Data Tombol cari

c. Kompleksitas Siklomatis pada Rekap Data tombol Cari
 Perhitunga Kompleksitas Siklomatis diperoleh dari rumus
 $V(G) = E - N + 2 = 13 - 11 + 2 = 4$

Lampiran B11. Pengujian Refresh Table1

a. Listing Program pada Refresh Table 1

Tahap pertama pada pengujian white box yaitu dengan cara memberikan penomoran pada program berdasarkan statemen dan logika dari setiap fitur yang ada pada sistem yang dapat dilihat pada gambar B.20

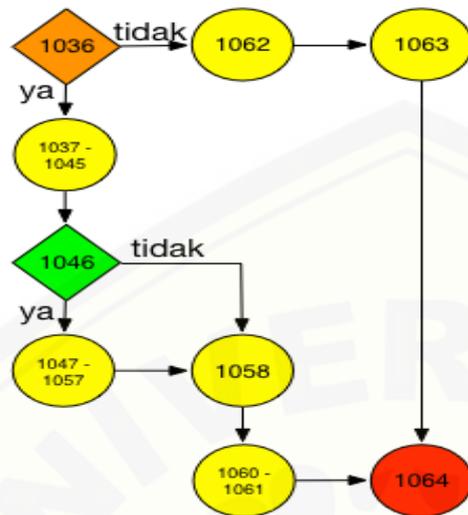
```

1091 private void refresh_table1() {
1092     try { } 1
1093         String query = "SELECT user_rfid, user_nomer, user_nama, user_jabata
1094         String[] judul = {"No", "RFID", "NIM / NIK / NIP", "Nama", "Jabatan"
1095
1096         Statement st = kon.get_koneksi().createStatement();
1097         ResultSet rs = st.executeQuery(query);
1098
1099         DefaultTableModel model = new DefaultTableModel(null, judul);
1100
1101         int urut = 0;
1102         while (rs.next()) { } 3
1103             String user_rfid = rs.getString("user_rfid");
1104             String user_nomer = rs.getString("user_nomer");
1105             String user_nama = rs.getString("user_nama");
1106             String user_jabatan = rs.getString("user_jabatan");
1107             String user_angkatan = rs.getString("user_angkatan");
1108             String user_nopol = rs.getString("user_nopol");
1109             String user_status = rs.getString("user_status");
1110
1111             Object[] d = {++urut, user_rfid, user_nomer, user_nama, user_jab
1112
1113             model.addRow(d);
1114         }
1115
1116         table1.setModel(model);
1117
1118         if (table1.getColumnModel().getColumnCount() > 0) {
1119             table1.getColumnModel().getColumn(0).setResizable(false);
1120             table1.getColumnModel().getColumn(0).setPreferredWidth(25);
1121             table1.getColumnModel().getColumn(1).setResizable(false);
1122             table1.getColumnModel().getColumn(1).setPreferredWidth(95);
1123             table1.getColumnModel().getColumn(2).setResizable(false);
1124             table1.getColumnModel().getColumn(2).setPreferredWidth(135);
1125             table1.getColumnModel().getColumn(3).setResizable(false);
1126             table1.getColumnModel().getColumn(3).setPreferredWidth(160);
1127             table1.getColumnModel().getColumn(4).setResizable(false);
1128             table1.getColumnModel().getColumn(4).setPreferredWidth(75);
1129             table1.getColumnModel().getColumn(5).setResizable(false);
1130             table1.getColumnModel().getColumn(5).setPreferredWidth(85);
1131             table1.getColumnModel().getColumn(6).setResizable(false);
1132             table1.getColumnModel().getColumn(6).setPreferredWidth(85);
1133             table1.getColumnModel().getColumn(7).setResizable(false);
1134             table1.getColumnModel().getColumn(7).setPreferredWidth(68);
1135         }
1136     } catch (Exception e) { } 6
1137         System.out.println("error : " + e.getMessage()); } 7
1138     }
1139 }

```

Gambar B.20 Refresh Table1 Fitur Rekap Data

b. Grafik alir refresh table1 rekap data



Gambar B21. Grafik Alir refres table1 Fitur Rekap Data

c. Kompleksitas Siklomatis Refresh Table1 pada fitur Rekap Data

Perhitungan Kompleksitas Siklomatis diperoleh dari rumus

$$V(G) = E - N + 2 = 10 - 9 + 2 = 3$$

Lampiran B12. Pengujian Refresh Table 2

a. Listing Program pada Refresh Table 2

Tahap pertama pada pengujian white box yaitu dengan cara memberikan penomoran pada program berdasarkan statemen dan logika dari setiap fitur yang ada pada sistem yang dapat dilihat pada gambar B.22

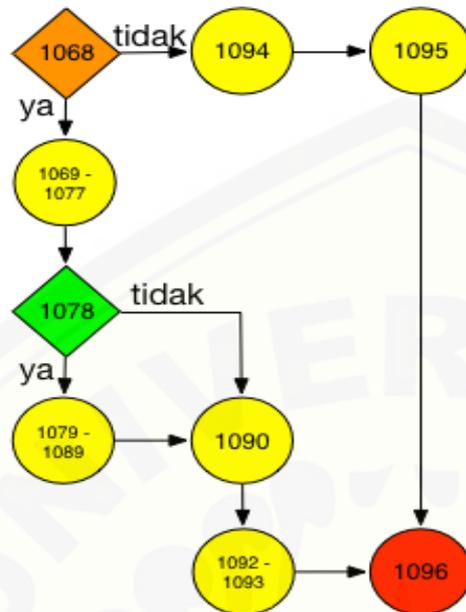
```

1141 private void refresh_table2() {
1142     try { } 1
1143     String query = "SELECT user_nomer, user_nama, user_jabatan, user_nopol";
1144     String[] judul = {"No", "NIM / NIK / NIP", "Nama", "Jabatan", "Plat";
1145
1146     Statement st = kon.get_koneksi().createStatement();
1147     ResultSet rs = st.executeQuery(query);
1148
1149     DefaultTableModel model = new DefaultTableModel(null, judul);
1150
1151     int urut = 0;
1152     while (rs.next()) { } 3
1153         String user_nomer = rs.getString("user_nomer");
1154         String user_nama = rs.getString("user_nama");
1155         String user_jabatan = rs.getString("user_jabatan");
1156         String user_nopol = rs.getString("user_nopol");
1157         String tanggal = rs.getString("tanggal");
1158         String masuk = rs.getString("masuk");
1159         String keluar = rs.getString("keluar");
1160
1161         Object[] d = {++urut, user_nomer, user_nama, user_jabatan, user_nopol, tanggal, masuk, keluar};
1162
1163         model.addRow(d);
1164     }
1165
1166     table2.setModel(model);
1167
1168     if (table2.getColumnModel().getColumnCount() > 0) {
1169         table2.getColumnModel().getColumn(0).setResizable(false);
1170         table2.getColumnModel().getColumn(0).setPreferredWidth(36);
1171         table2.getColumnModel().getColumn(1).setResizable(false);
1172         table2.getColumnModel().getColumn(1).setPreferredWidth(128);
1173         table2.getColumnModel().getColumn(2).setResizable(false);
1174         table2.getColumnModel().getColumn(2).setPreferredWidth(135);
1175         table2.getColumnModel().getColumn(3).setResizable(false);
1176         table2.getColumnModel().getColumn(3).setPreferredWidth(80);
1177         table2.getColumnModel().getColumn(4).setResizable(false);
1178         table2.getColumnModel().getColumn(4).setPreferredWidth(85);
1179         table2.getColumnModel().getColumn(5).setResizable(false);
1180         table2.getColumnModel().getColumn(5).setPreferredWidth(85);
1181         table2.getColumnModel().getColumn(6).setResizable(false);
1182         table2.getColumnModel().getColumn(6).setPreferredWidth(73);
1183         table2.getColumnModel().getColumn(7).setResizable(false);
1184         table2.getColumnModel().getColumn(7).setPreferredWidth(73);
1185     }
1186 } catch (Exception e) { } 6
1187     System.out.println("error : " + e.getMessage()); } 7
1188 }
1189 }

```

Gambar B22. Listing Program Refresh Table2 Pada Fitur Rekap Data

b. Grafik Alir Refresh Table 2



Gambar B23. Grafik Alir Refresh Table2

c. Kompleksitas Siklomatis Refresh Table2 pada fitur Rekap Data

Perhitungan Kompleksitas Siklomatis diperoleh dari rumus

$$V(G) = E - N + 2 = 10 - 9 + 2 = 3$$