

**SISTEM PERANGKAT LUNAK UNTUK SIMULASI KELUAR MASUK
DAN PEMBAYARAN PARKIR KENDARAAN BERBASIS RFID**

**SKRIPSI
KONSENTRASI REKAYASA PERANGKAT LUNAK**

Untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Komputer



Disusun Oleh :

FRANSISCHA TIKA SEPTIANA

NIM. 0910683046

**KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
PROGRAM STUDI TEKNIK INFORMATIKA
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2013**

LEMBAR PERSETUJUAN

SISTEM PERANGKAT LUNAK UNTUK SIMULASI KELUAR MASUK DAN PEMBAYARAN PARKIR KENDARAAN BERBASIS RFID

SKRIPSI

KONSENTRASI REKAYASA PERANGKAT LUNAK

Untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer



Disusun oleh :

FRANSISCHA TIKA SEPTIANA

NIM. 0910683046

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Ismiarta Aknuranda, ST., M.Sc., Ph.D

NIK. 740719 06 1 1 0079

Aditya Rachmadi, S.ST., MTI

NIK. 860421 16 1 1 0426

LEMBAR PENGESAHAN
SISTEM PERANGKAT LUNAK UNTUK SIMULASI KELUAR MASUK
DAN PEMBAYARAN PARKIR KENDARAAN BERBASIS RFID

SKRIPSI
KONSENTRASI REKAYASA PERANGKAT LUNAK
Untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer

Disusun oleh :
FRANSISCHA TIKA SEPTIANA
NIM. 0910683046

Skripsi ini telah diuji dan dinyatakan lulus pada
tanggal 3 Mei 2013

Penguji I

Penguji II

Issa Arwani, S.Kom, M.Sc

Denny Sagita Rusdianto, S.Kom. M. Kom.

NIK. 830922 06 1 1 0074

NIK. 851124 06 1 1 0250

Penguji III

Gembong Edhi Setyawan, ST. MT.

NIK. 850920 16 1 1 0373

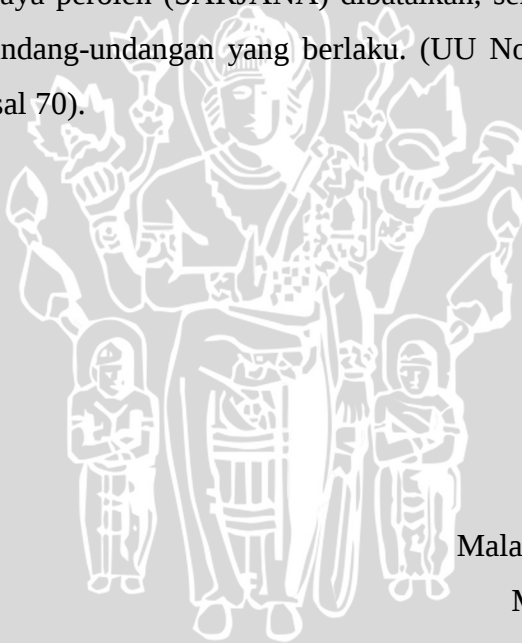
Mengetahui
Ketua Program Studi Teknik Informatika

Drs. Marji, M.Si.
NIP. 19670801 199203 1 001

**PERNYATAAN
ORISINALITAS SKRIPSI**

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah SKRIPSI ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata didalam naskah SKRIPSI ini dapat dibuktikan terdapat unsur-unsur PLAGIASI, saya bersedia SKRIPSI ini digugurkan dan gelar akademik yang telah saya peroleh (SARJANA) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku. (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).



Malang, 8 Mei 2013
Mahasiswa,

Fransischa Tika Septiana
NIM 0910683046

KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadirat Tuhan Yang Maha Esa, karena berkat karunia-Nya penulis dapat menyelesaikan proposal skripsi ini dengan lancar dan tepat waktu. Penyusunan skripsi ini adalah dengan maksud untuk memenuhi salah satu persyaratan dalam mengikuti ujian akhir Sarjana Program Studi Teknik Informatika pada Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya, dengan judul "Sistem Perangkat Lunak Untuk Simulasi Keluar Masuk dan Pembayaran Parkir Kendaraan Berbasis RFID".

Banyak hambatan yang telah dihadapi penulis dalam penyelesaian skripsi ini terutama dalam hal penetapan sistem, tetapi berkat bantuan dan dukungan dari berbagai pihak hambatan – hambatan tersebut bisa dilewati. Oleh karena itu dengan adanya kesempatan ini, penulis ingin mengucapkan terima kasih kepada :

1. Bapak Drs. Marji, M.T, selaku Ketua Program Studi Teknik Informatika Universitas Brawijaya Malang.
2. Bapak Ismiarta Aknuranda, ST., M.Sc.,Ph.D, selaku dosen pembimbing I yang telah memberikan pengarahan dalam pembuatan proposal skripsi ini.
3. Bapak Aditya Rachmadi, S.ST., MTI, selaku dosen pembimbing II yang telah memberikan saran – saran yang sangat berguna dalam penyelesaian proposal skripsi ini.
4. Orang Tua, yang telah memberikan dukungan moral dan material.
5. Shancha Ricardo A., ST, Joseph Kristian T., Pricillia Carolina K., Anisa Aini A., Adam Hendra B. dan Noviana Putri P. yang telah memberikan berbagai macam bantuan dan semangat.
6. Seluruh mahasiswa TIF angkatan 2009, serta seluruh pihak yang telah membantu yang tidak dapat saya sebutkan satu per satu.

Penulis menyadari bahwa proposal skripsi ini jauh dari sempurna, oleh karena itu untuk segala kritik dan saran yang membangun penulis ucapkan terima kasih. Penulis mengharapkan semoga proposal skripsi ini dapat berguna bagi yang membutuhkannya.

Malang, 8 Mei 2013

Penulis

ABSTRAK

Fransisca Tika Septiana. 2013. : Sistem Perangkat Lunak Untuk Simulasi Keluar Masuk dan Pembayaran Parkir Kendaraan Berbasis RFID. Skripsi Program Studi Teknik Informatika, Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya.

Dosen Pembimbing : Bapak Ismiarta Aknuranda, ST., M.Sc., Ph.D dan Bapak Aditya Rachmadi, S.ST, MTI

Lamanya proses keluar masuk area parkir menjadi kelemahan sistem perpajakan manual yang banyak terdapat di Indonesia. Hal ini mendorong penulis membuat sistem baru untuk menyelesaikan permasalahan tersebut. Sistem ini ditunjang dengan aplikasi identifikasi otomatis yang menggunakan RFID untuk pengenalan identitas pengemudi dan pustaka DTK ANPR SDK untuk pengenalan plat nomor. Sistem ini juga menyediakan fitur *sms gateway* untuk informasi pengguna dan skenario sistem pembayaran yang lebih efisien. Sistem ini terbagi menjadi dua subsistem, yaitu subsistem gerbang parkir yang dijalankan secara simulasi dan subsistem administrasi. Pengembangan perangkat lunak ini menggunakan model *waterfall* dengan metode analisis OOAD yang dimodelkan dengan UML dan diimplementasikan dengan bahasa pemrograman Java. Pengujian sistem ini menggunakan metode pengujian validasi dari fungsional sistem dan pengujian performa dari analisis waktu kerja subsistem gerbang parkir. Pada hasil pengujian validasi, fungsionalitas sistem telah memenuhi kebutuhan dan memiliki tingkat akurasi kesuksesan minimal 93,3%. Sementara itu, pada pengujian performa didapatkan hasil jalannya subsistem gerbang parkir lebih cepat 1,15 detik dari sistem perparkiran secara manual.

Kata Kunci : DTK ANPR SDK, keluar-masuk otomatis, OOAD, RFID, sistem parkir, *sms gateway*, UML.

ABSTRACT

Fransisca Tika Septiana. 2013. : *Simulation Software System for Entrance and Exit Parking Vehicle and Payments Based RFID.*

Supervisors : Mr. Ismiarta Aknuranda, ST., M.Sc., Ph.D and Mr. Aditya Rachmadi, S.ST, MTI

The lengthy process of entry into and exit from a parking area has become one of the weaknesses of manually operated parking systems in Indonesia. This problem encouraged the author to make a new system which can resolve it. This system is supported by automatic identification using RFID to identify the driver and DTK ANPR SDK library for plate number recognition. This system also provides an sms gateway feature for customer services and more efficient payment scenarios. The system is simulated into two subsystems: a gate system and an administration system. This software was developed using waterfall model, analysed and designed using OOAD, modeled with UML, and implemented with the Java programming language. This software was tested using validation testing of system's functionality and performance testing of the system's working time. From the result of validation testing, it can be concluded that the functionality of this system has met the requirements and has a minimum success rate of 93.3%. The result of the performance testing demonstrates that the automatic gate system is 1.15 second faster than the manually operated gate system.

Keywords: automated entry-exit, DTK ANPR SDK, OOAD, parking system, RFID, sms gateway, UML.

DAFTAR ISI

	Halaman
JUDUL	i
LEMBAR PERSETUJUAN	ii
LEMBAR PENGESAHAN	iii
PERNYATAAN ORISINALITAS SKRIPSI	iv
KATA PENGANTAR	v
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR GAMBAR	xi
DAFTAR TABEL	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	3
1.4 Tujuan	3
1.5 Manfaat	4
1.6 Sistematika Penulisan	4
1.7 Jadwal Penelitian	5
BAB II DASAR TEORI	6
2.1 Model <i>Waterfall</i>	6
2.2 <i>Object-Oriented Analysis and Design (OOAD)</i>	9
2.2.1 <i>Unified Modeling Language (UML)</i>	10
2.2.2 Pemrograman Berorientasi Objek dan Java	12
2.3 Teknologi RFID	14
2.4 Pengenalan Plat Nomor	17
2.5 Standardisasi Parkir	18
2.6 Keamanan Sistem	20
2.7 Pengujian Perangkat Lunak	21

2.7.1	Pengujian Validasi.....	21
2.7.2	Pengujian Performa	22
BAB III METODE PENELITIAN		23
3.1	Studi Literatur.....	24
3.2	Survei Sistem yang Sudah Ada	24
3.3	Analisis Kebutuhan	24
3.4	Survei Alat.....	25
3.5	Pengumpulan Data.....	25
3.6	Perancangan Sistem.....	25
3.7	Implementasi	26
3.8	Pengujian	26
3.9	Pengambilan Kesimpulan dan Saran	27
BAB IV PERANCANGAN		28
4.1	Analisis Kebutuhan	28
4.1.1	Gambaran Umum Sistem Perangkat Lunak Untuk Simulasi Keluar Masuk dan Pembayaran Parkir Kendaraan Berbasis RFID	28
4.1.2	Identifikasi Aktor.....	33
4.1.3	Analisis Data	34
4.1.4	Daftar Kebutuhan	35
4.1.5	Analisis <i>Use Case</i>	37
4.2	Perancangan Perangkat Lunak	46
4.2.1	Pemodelan Diagram <i>Statechart</i>	47
4.2.2	Perancangan Basis Data	49
4.2.3	Diagram <i>Class</i>	50
4.2.4	Diagram Sequence	56
4.2.5	Diagram Aktivitas	64
4.2.6	Perancangan Antarmuka.....	73
BAB V IMPLEMENTASI.....		83
5.2	Spesifikasi Lingkungan Sistem	83
5.2.1	Spesifikasi Lingkungan Perangkat Keras	83
5.2.2	Spesifikasi Lingkungan Perangkat Lunak	84

5.3	Batasan – Batasan Implementasi	84
5.4	Implementasi Basis Data	84
5.5	Implementasi <i>Class</i> dan <i>Interface</i> Pada <i>File</i> Program	88
5.6	Implementasi Algoritma	89
5.6.1	Implementasi Algoritma untuk Proses Keluar Masuk Gerbang Parkir	89
5.6.2	Implementasi Algoritma untuk Melakukan <i>Login</i>	90
5.6.3	Implementasi Algoritma untuk Melakukan Pengolahan Admin	91
5.6.4	Implementasi Algoritma untuk Melakukan Pengolahan Pengendara	92
5.6.5	Implementasi Algoritma untuk Melakukan Pengolahan Kendaraan	93
5.6.6	Implementasi Algoritma untuk Melakukan Perbaruhi Pengaturan	94
5.6.7	Implementasi Algoritma untuk Melakukan Pengolahan Saldo	95
5.6.8	Implementasi Algoritma untuk Melakukan Pengolahan Log Parkir	97
5.7	Implementasi Antarmuka	99
5.7.1	Implementasi Antarmuka Subsistem Gerbang Parkir	100
5.7.2	Implementasi Antarmuka Subsistem Administrasi	100
BAB VI PENGUJIAN DAN PEMBAHASAN		107
6.1	Pengujian Validasi	107
6.1.1	Pengujian Fungsional Berdasarkan Use Case Diagram	108
6.1.2	Pengujian Kondisi Gerbang Parkir	121
6.2	Pengujian Performa	122
6.2.1	Hasil Pengujian Performa	123
6.2.2	Pembahasan Pengujian Performa	125
BAB VII PENUTUP		126
7.1	Kesimpulan	126
7.2	Saran	126
DAFTAR PUSTAKA		128
LAMPIRAN		130

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Model <i>Waterfall</i>	6
Gambar 2.2 Klasifikasi Sistem Identifikasi Otomatis.....	15
Gambar 2.3 Penampangan <i>tag</i> RFID dan RFID <i>Reader</i> merk mifare	16
Gambar 2.4 Pintu Masuk dan Keluar Terpisah.....	18
Gambar 2.5 Pintu Masuk dan Keluar Menjadi Satu	19
Gambar 3.1 Diagram Blok Metode Penelitian.....	23
Gambar 4.1 Rencana Penerapan Alur Subsistem Gerbang Parkir Secara Nyata	32
Gambar 4.2 Alur Subsistem Gerbang Parkir Secara Simulasi.....	33
Gambar 4.3 Diagram <i>use case</i> subsistem gerbang parkir	38
Gambar 4.4 Diagram <i>use case</i> subsistem administrasi	40
Gambar 4.5 Diagram <i>statechart</i> subsistem gerbang parkir.....	47
Gambar 4.6 Diagram <i>entity relationship</i> sistem	49
Gambar 4.7 Diagram <i>class</i> subsistem gerbang parkir.....	50
Gambar 4.8 Diagram <i>class</i> subsistem administrasi	52
Gambar 4.9 Diagram <i>sequence</i> Subsistem Gerbang Parkir	57
Gambar 4.10 Diagram <i>sequence</i> Login.....	57
Gambar 4.11 Diagram <i>sequence</i> Sms Peringatan	58
Gambar 4.12 Diagram <i>sequence</i> Pengaturan Admin.....	59
Gambar 4.13 Diagram <i>sequence</i> Pengaturan Kendaraan.....	60
Gambar 4.14 Diagram <i>sequence</i> Pengaturan Pengendara	61
Gambar 4.15 Diagram <i>sequence</i> Pengaturan Saldo	62
Gambar 4.16 Diagram <i>sequence</i> Pengaturan Log Parkir.....	63
Gambar 4.17 Diagram <i>sequence</i> Pengaturan	64
Gambar 4.18 Diagram aktivitas proses masuk.....	65
Gambar 4.19 Diagram aktivitas proses keluar	66
Gambar 4.20 Diagram aktivitas Login.....	67
Gambar 4.21 Diagram aktivitas pengaturan admin	68
Gambar 4.22 Diagram aktivitas pengaturan pengendara	69
Gambar 4.23 Diagram aktivitas pengaturan kendaraan	70

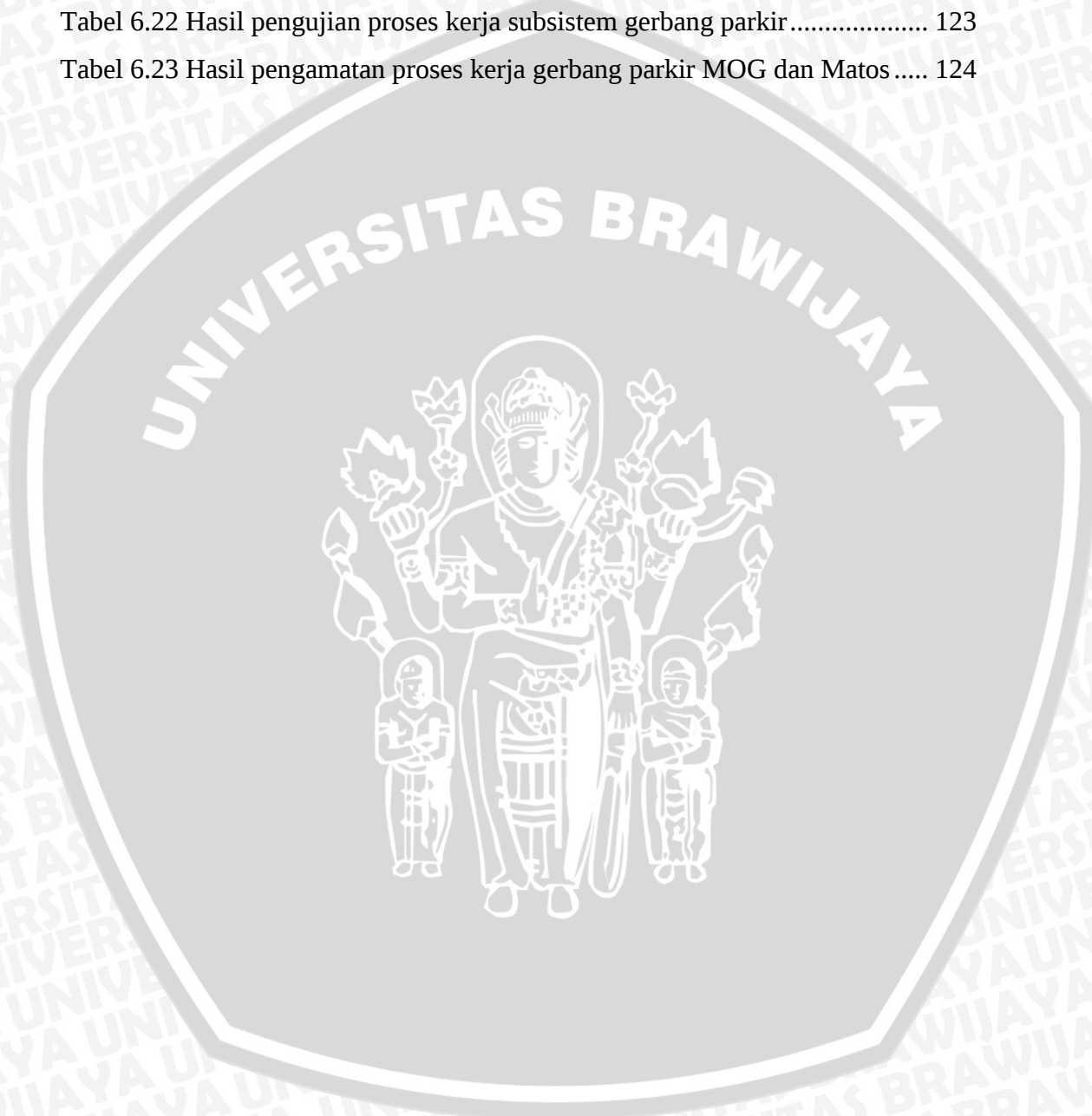
Gambar 4.24 Diagram aktivitas pengaturan saldo	71
Gambar 4.25 Diagram aktivitas pengaturan log parkir.....	72
Gambar 4.26 Diagram aktivitas pengaturan	73
Gambar 4.27 Tampilan antarmuka subsistem gerbang parkir	74
Gambar 4.28 <i>Sitemap</i> antarmuka pengguna subsistem administrasi	75
Gambar 4.29 Tampilan antarmuka halaman login.....	76
Gambar 4.30 Tampilan antarmuka halaman utama	76
Gambar 4.31 Tampilan antarmuka halaman pengaturan admin	77
Gambar 4.32 Tampilan antarmuka halaman pengaturan pengendara.....	78
Gambar 4.33 Tampilan antarmuka halaman pengaturan kendaraan.....	79
Gambar 4.34 Tampilan antarmuka halaman pengaturan saldo.....	80
Gambar 4.35 Tampilan antarmuka halaman pengaturan log parkir.....	81
Gambar 4.36 Tampilan antarmuka halaman pengaturan	82
Gambar 5.1 Diagram ER konseptual dari sistem.....	85
Gambar 5.2 Implementasi antarmuka subsistem gerbang parkir	100
Gambar 5.3 Implementasi antarmuka halaman login	101
Gambar 5.4 Implementasi antarmuka halaman utama admin.....	102
Gambar 5.5 Implementasi antarmuka halaman pengaturan admin.....	102
Gambar 5.6 Implementasi antarmuka halaman pengaturan pengendara	103
Gambar 5.7 Implementasi antarmuka halaman pengaturan kendaraan	104
Gambar 5.8 Implementasi antarmuka halaman pengaturan saldo	105
Gambar 5.9 Implementasi antarmuka halaman pengaturan log parkir	105
Gambar 5.10 Implementasi antarmuka pengaturan setting.....	106

DAFTAR TABEL

	Halaman
Tabel 1.1 Jadwal penelitian.....	5
Tabel 2.1 Perbandingan <i>tag</i> aktif dan <i>tag</i> pasif	15
Tabel 4.1 Identifikasi aktor	34
Tabel 4.2 Spesifikasi kebutuhan fungsional subsistem gerbang parkir	35
Tabel 4.3 Spesifikasi kebutuhan fungsional subsistem administrasi	35
Tabel 4.4 Spesifikasi kebutuhan non-fungsional	36
Tabel 4.5 Skenario <i>Use case</i> Menunjukkan RFID.....	38
Tabel 4.6 Skenario <i>Use case</i> Membawa Kendaraan untuk Dapat Diambil Gambarnya.....	39
Tabel 4.7 Skenario <i>Use case</i> Login	40
Tabel 4.8 Skenario <i>Use case</i> Mengelola Data Admin	41
Tabel 4.9 Skenario <i>Use case</i> Mengelola Data Kendaraan	42
Tabel 4.10 Skenario <i>Use case</i> Mengelola Data Pengendara.....	43
Tabel 4.11 Skenario <i>Use case</i> Mengelola Data Saldo	44
Tabel 4.12 Skenario <i>Use case</i> Mengelola Data Pengaturan	45
Tabel 4.13 Skenario <i>Use case</i> Mengelola Data Log Parkir	45
Tabel 4.14 Deskripsi <i>class</i> ManageGbg	51
Tabel 4.15 Deskripsi <i>class</i> Processing	51
Tabel 4.16 Deskripsi <i>class</i> ViewCameraFrame	51
Tabel 4.17 Deskripsi <i>class</i> Detection.....	51
Tabel 4.18 Deskripsi <i>class</i> ManageAdm	52
Tabel 4.19 Deskripsi <i>class</i> Login.....	53
Tabel 4.20 Deskripsi <i>class</i> Form	53
Tabel 4.21 Deskripsi <i>class</i> ManageAdmin	53
Tabel 4.22 Deskripsi <i>class</i> ManageKendaraan	53
Tabel 4.23 Deskripsi <i>class</i> ManagePengendara.....	54
Tabel 4.24 Deskripsi <i>class</i> ManageLogParkir	54
Tabel 4.25 Deskripsi <i>class</i> ManageSaldo	54
Tabel 4.26 Deskripsi <i>class</i> Setting.....	54

Tabel 4.27 Deskripsi <i>class</i> SmsPeringatan	55
Tabel 4.28 Deskripsi <i>class</i> Sender	55
Tabel 4.29 Deskripsi <i>class</i> SMSClient	55
Tabel 4.30 Deskripsi <i>class</i> SerialConnection	55
Tabel 4.31 Deskripsi <i>class</i> SerialParameters	56
Tabel 5.1 Spesifikasi lingkungan perangkat keras komputer	83
Tabel 5.2 Spesifikasi lingkungan perangkat lunak komputer	84
Tabel 5.3 Struktur tabel admin.....	85
Tabel 5.4 Struktur tabel pengaturan.....	86
Tabel 5.5 Struktur tabel kendaraan	86
Tabel 5.6 Struktur tabel pengendara	86
Tabel 5.7 Struktur tabel saldo	87
Tabel 5.8 Struktur tabel logsaldo	87
Tabel 5.9 Struktur tabel logparkir	88
Tabel 5.10 Implementasi <i>class</i> pada kode program *.java	88
Tabel 6.1 Kasus uji untuk menunjukkan tag RFID.....	108
Tabel 6.2 Kasus uji membawa kendaraan untuk dapat diambil gambarnya.....	108
Tabel 6.3 Kasus uji untuk <i>login</i>	109
Tabel 6.4 Kasus uji untuk memasukkan data admin.....	110
Tabel 6.5 Kasus uji untuk perbarui data admin	110
Tabel 6.6 Kasus uji untuk hapus data admin	111
Tabel 6.7 Kasus uji untuk memasukkan data kendaraan	112
Tabel 6.8 Kasus uji untuk perbarui data kendaraan	112
Tabel 6.9 Kasus uji untuk hapus data kendaraan	113
Tabel 6.10 Kasus uji untuk memasukkan data pengendara	114
Tabel 6.11 Kasus uji untuk perbarui data pengendara	114
Tabel 6.12 Kasus uji untuk hapus data pengendara	115
Tabel 6.13 Kasus uji untuk menambah saldo	116
Tabel 6.14 Kasus uji untuk mencetak log saldo.....	116
Tabel 6.15 Kasus uji untuk mengirim SMS ke pemilik kendaraan	117
Tabel 6.16 Kasus uji untuk mengirim SMS ke semua pemilik kendaraan	118
Tabel 6.17 Kasus uji untuk mengelolah data pengaturan	118

Tabel 6.18 Kasus uji untuk mencari data log parkir	119
Tabel 6.19 Kasus uji untuk mencetak data log parkir.....	119
Tabel 6.20 Kasus uji untuk melaporkan berita kehilangan kendaraan	120
Tabel 6.21 Hasil pengujian kondisi gerbang parkir	121
Tabel 6.22 Hasil pengujian proses kerja subsistem gerbang parkir.....	123
Tabel 6.23 Hasil pengamatan proses kerja gerbang parkir MOG dan Matos	124



BAB I

PENDAHULUAN

1.1 Latar Belakang

Dewasa ini banyak sistem perparkiran di Indonesia masih menggunakan sistem perparkiran yang manual. Salah satu skenario keluar masuk parkir yang telah diterapkan di berbagai tempat di Indonesia adalah dengan adanya pencatatan plat nomor beserta STNK lalu pemberian karcis parkir pada saat pengendara memasuki area parkir. Pada saat keluar, STNK dicocokkan dengan karcis parkir yang telah diberikan pada saat masuk dan dengan plat nomor kendaraan. Sistem tersebut membutuhkan petugas yang beroperasi dalam waktu lama (misalnya secara bergantian selalu siap siaga 24 jam) tanpa melakukan kesalahan signifikan (misal: meninggalkan tempat, salah mencatat plat nomor dan mengabaikan STNK) dan lamanya proses keluar masuk parkir tergantung oleh petugas yang melakukannya. Skenario tersebut terkadang membuat pengendaraan menjadi tidak nyaman, karena waktu transaksi dalam sistem parkir ini cukup membuat pelanggan menghabiskan waktu cukup lama dan pengendara juga harus menyiapkan biaya tarif parkir dengan uang pas. Jika pengendara menggunakan uang yang lebih besar, maka mereka harus menghabiskan waktu yang lebih lama lagi untuk melakukan transaksi parkir tersebut. Waktu yang lama dalam proses keluar masuk parkir ini tidak jarang berdampak pada kemacetan lalu lintas. Dari hasil survey awal penulis (dapat dilihat pada lampiran 1), 8 dari 10 pengendara mengeluhkan ketidaknyamanan sistem parkir tersebut.

Sistem yang dibuat dalam skripsi ini mengusahakan agar permasalahan yang dilakukan secara manual dapat terselesaikan, tetapi keamanan sistem perparkiran tetap terjaga seperti yang telah dilakukan secara manual. Sistem ini ditunjang dengan aplikasi identifikasi yang otomatis yaitu dengan menggunakan RFID, pengenalan plat nomor dan pembayaran yang lebih efisien. RFID (*Radio Frequency Identification*) merupakan teknologi identifikasi otomatis yang murah yang dapat diimplementasikan pada sistem atau alat yang relatif murah.[NAT-06]

Dalam penelitian sebelumnya, Hamid [HAM-10] telah membuat sistem otomatisasi parkir dengan menggunakan RFID. Skenario yang dibuat oleh Hamid

adalah saat kendaraan masuk pengemudi mengeluarkan dan mendekatkan *tag* RFID ke *reader* yang terletak pada loket masuk. Pada saat *tag* ada dalam jangkauan *reader* kemudian *reader* membaca identitas kendaraan yang kemudian menjadi *input* dalam proses awal sistem parkir ini.

Selain memperbaiki skenario identifikasi manual kendaraan, skripsi yang penulis buat ini juga berusaha meningkatkan level otomatisasi sistem yang telah dibuat oleh Hamid. Skenario sistem keluar masuk parkir dalam skripsi ini adalah dengan membaca identitas pengemudi menggunakan teknologi RFID, lalu menyimpannya bersama dengan data plat nomor yang gambarnya diambil secara otomatis oleh sistem. Berbeda dengan skenario yang dibuat oleh Hamid, karena data plat nomor kendaraan disimpan dalam RFID *tag*, jadi sistem hanya dapat membaca identitas kendaraan saja. Tambahan fitur jika terjadi kehilangan kendaraan dalam sistem ini, berdasarkan data plat nomor dan pengemudi yang tersimpan dalam sistem, admin dapat melaporkannya lewat sms kepada bagian keamanan. Dengan demikian proses keluar masuk parkir yang dirancang dalam skripsi ini menjadi lebih cepat, tetapi keamanan kendaraan tetap terjaga.

Selain menerapkan teknologi RFID untuk keluar masuk kendaraan, sistem ini juga mengolah sistem pembayaran tarif parkir menjadi dapat dilakukan secara pra bayar atau pasca bayar. Pengurangan saldo terjadi saat kendaraan keluar dari area parkir. Sistem juga akan menjadi lebih efisien karena adanya sistem pembayaran dengan cara pengurangan saldo secara otomatis, sehingga tidak membutuhkan waktu yang lama.

Tujuan membuat sistem yang seperti ini adalah agar pelanggan yang menggunakan sistem ini merasa lebih nyaman dan proses yang terjadi lebih efisien. Sistem ini bisa digunakan di segala tempat yang memiliki skenario yang sama seperti apartemen, kampus dan sebagainya.

1.2 Rumusan Masalah

Berdasarkan permasalahan yang telah dijabarkan pada bagian latar belakang, maka rumusan masalah dikhususkan pada:

1. Bagaimana perancangan Sistem Perangkat Lunak Untuk Simulasi Keluar Masuk dan Pembayaran Parkir Kendaraan Berbasis RFID?

2. Bagaimana implementasi Sistem Perangkat Lunak Untuk Simulasi Keluar Masuk dan Pembayaran Parkir Kendaraan Berbasis RFID?
3. Bagaimana pengujian dan pembahasan Sistem Perangkat Lunak Untuk Simulasi Keluar Masuk dan Pembayaran Parkir Kendaraan Berbasis RFID?

1.3 Batasan Masalah

Berdasarkan rumusan masalah yang telah diuraikan, maka berikut ini dibuat beberapa batasan masalah:

1. Fokus dari sistem ini adalah sistem perparkiran di apartemen, tetapi sistem ini tidak menutup kemungkinan untuk diterapkan di tempat lain yang memiliki skenario operasi yang sama.
2. Obyek dari perancangan sistem ini adalah orang-orang yang telah memiliki RFID yang telah ditentukan sistem.
3. Pemrograman RFID reader berupa komponen dengan inputan berupa RFID tag dan output berupa data id.
4. Transaksi pengisian saldo diperbolehkan untuk dilakukan secara pra bayar atau pasca bayar.
5. Pengenalan plat nomor berupa komponen dengan input berupa gambar dan output berupa data teks.

1.4 Tujuan

Berdasarkan rumusan masalah yang telah dijabarkan, maka tujuan dari skripsi ini adalah:

1. Merancang sistem perangkat lunak untuk simulasi keluar masuk dan pembayaran parkir kendaraan berbasis RFID menggunakan metode pengembangan perangkat lunak SDLC menggunakan model *waterfall*.
2. Mengimplementasikan sistem perangkat lunak untuk simulasi keluar masuk dan pembayaran parkir kendaraan berbasis RFID berupa simulasi kerja.
3. Menguji dan membahas hasil pengujian sistem perangkat lunak untuk simulasi keluar masuk dan pembayaran parkir kendaraan berbasis RFID dengan menggunakan pengujian validasi dan pengujian performa.

1.5 Manfaat

Manfaat yang diharapkan dengan adanya Sistem Perangkat Lunak Untuk Simulasi Keluar Masuk dan Pembayaran Parkir Kendaraan Berbasis RFID:

- a. Bagi pengelola lahan parkir
Memudahkan pihak pengelola lahan parkir untuk menjaga keamanan dan memperlancar keluar masuk kendaraan.
- b. Bagi pengendara
Membuat pengendara kendaraan menjadi merasa aman dan lebih nyaman.
- c. Bagi penulis
Menerapkan ilmu yang telah diperoleh dari Teknik Informatika Universitas Brawijaya. Mendapatkan pemahaman tentang perancangan dan pengembangan aplikasi sistem perangkat lunak untuk simulasi keluar masuk dan pembayaran parkir kendaraan berbasis RFID.

1.6 Sistematika Penulisan

Sistematika penulisan dalam skripsi ini sebagai berikut:

BAB I Pendahuluan

Memuat latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, metodologi pembahasan, dan sistematika penulisan.

BAB II Dasar teori

Memaparkan teori dasar dan teori pendukung yang berhubungan dengan sistem perangkat lunak untuk simulasi keluar masuk dan pembayaran parkir kendaraan berbasis RFID.

BAB III Metode Penelitian

Membahas metode yang digunakan dalam penelitian untuk merekayasa sistem dengan tahapan kegiatan yaitu studi literatur, pengumpulan data, perancangan perangkat lunak, implementasi perangkat lunak, pengujian dan analisis, serta pengambilan kesimpulan dan saran.

BAB IV Perancangan

Membahas perancangan sistem perangkat lunak untuk simulasi keluar masuk dan pembayaran parkir kendaraan berbasis RFID.

BAB V Implementasi

Membahas tentang implementasi dari sistem yaitu membuat model dari sistem yang dibahas menggunakan simulasi.

BAB VI Pengujian

Memuat proses dan hasil pengujian terhadap sistem yang telah direalisasikan.

BAB VII Penutup

Memuat kesimpulan serta saran yang diperoleh dari pembuatan dan pengujian sistem untuk pengembangan lebih lanjut.

1.7 Jadwal Penelitian

Berikut ini disajikan tabel jadwal pengerjaan penelitian yang akan dilaksanakan:

Tabel 1.1 Jadwal penelitian

No	Kegiatan	Bulan dan Minggu ke :																							
		Bulan 1				Bulan 2				Bulan 3				Bulan 4				Bulan 5				Bulan 6			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1.	Studi literatur																								
2.	Survei sistem yang sudah ada																								
3.	Pengumpulan data																								
4.	Analisa kebutuhan																								
5.	Survei alat																								
6.	Perancangan Sistem																								
7.	Implementasi Sistem																								
8.	Pengujian perangkat lunak																								
9.	Penulisan laporan																								

Sumber: Pendahuluan

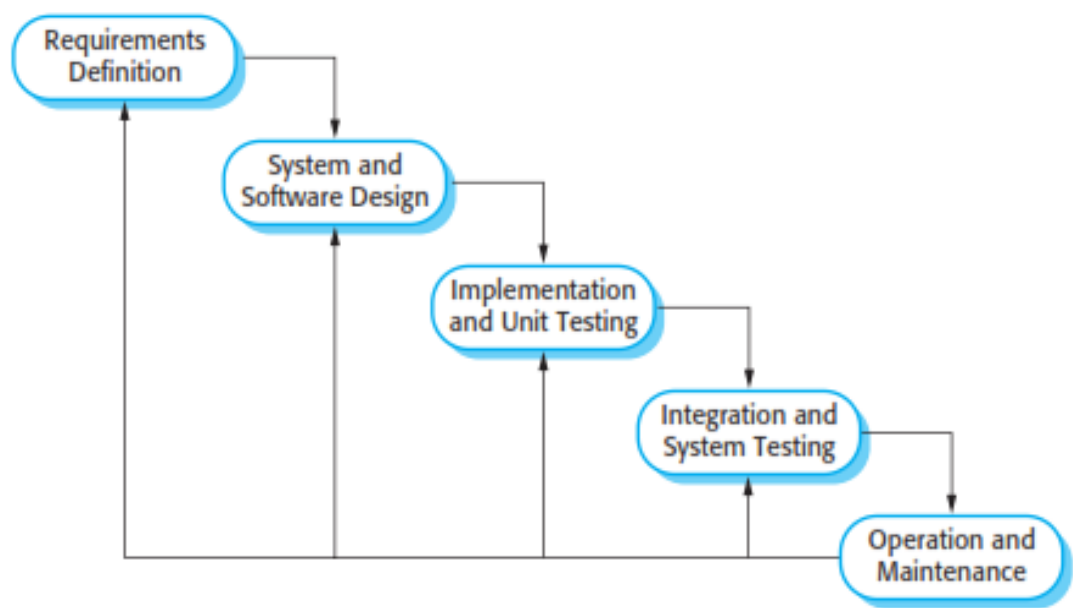
BAB II

DASAR TEORI

Bab ini membahas dasar-dasar teori yang digunakan untuk menunjang penulisan skripsi “Sistem Perangkat Lunak Untuk Simulasi Keluar Masuk dan Pembayaran Parkir Kendaraan Berbasis RFID”. Dasar teori yang dibutuhkan berdasarkan latar belakang dan rumusan masalah adalah Model *waterfall*, *Object-Oriented Analysis and Design* (OOAD), teknologi RFID, pengenalan plat nomor, standarisasi parkir, keamanan sistem dan pengujian perangkat lunak.

2.1 Model Waterfall

SDLC (Systems Development Life Cycle) dalam rekayasa sistem dan rekayasa perangkat lunak adalah proses pembuatan dan pengubahan sistem serta model dan metodologi yang digunakan untuk mengembangkan suatu sistem. Salah satu model yang cukup dikenal dalam dunia rekayasa perangkat lunak adalah model *waterfall*. Model *waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam membangun software. Tahapan utama dari model *waterfall* yang mencerminkan kegiatan pengembangan yang mendasar[SOM-11]:



Gambar 2.1 Model Waterfall

Sumber: [SOM-11]

1. Requirements Analysis and Definition

Mengumpulkan kebutuhan secara lengkap kemudian dianalisis dan didefinisikan kebutuhan yang harus dipenuhi oleh software yang akan dibangun. Hal ini sangat penting, mengingat software harus dapat berinteraksi dengan elemen-elemen yang lain seperti hardware, database, dan sebagainya.

2. System and Software Design

Proses pencarian kebutuhan diintensifkan dan difokuskan pada software. Untuk mengetahui sifat dari program yang akan dibuat, maka para software engineer harus mengerti tentang domain informasi dari software, misalnya fungsi yang dibutuhkan, user interface dan sebagainya. Dari dua aktivitas tersebut (pencarian kebutuhan sistem dan software) harus didokumentasikan dan ditunjukkan kepada user. Proses software design untuk mengubah kebutuhan-kebutuhan di atas menjadi representasi ke dalam bentuk “blueprint” software sebelum coding dimulai. Desain harus dapat mengimplementasikan kebutuhan yang telah disebutkan pada tahap sebelumnya. Seperti dua aktivitas sebelumnya, maka proses ini juga harus didokumentasikan sebagai konfigurasi dari software.

3. Implementation and Unit Testing

Desain program diterjemahkan ke dalam kode-kode dengan menggunakan bahasa pemrograman yang sudah ditentukan. Program yang dibangun langsung diuji baik secara unit.

4. Integration and System Testing

Untuk dapat dimengerti oleh mesin, dalam hal ini adalah komputer, maka desain tadi harus diubah bentuknya menjadi bentuk yang dapat dimengerti oleh mesin, yaitu ke dalam bahasa pemrograman melalui proses coding. Tahap ini merupakan implementasi dari tahap design yang secara teknis nantinya dikerjakan oleh programmer. Penyatuan unit-unit program kemudian diuji secara keseluruhan (system testing).

5. Operation and Maintenance

Sesuatu yang dibuat haruslah diujicobakan. Demikian juga dengan software. Semua fungsi-fungsi software harus diujicobakan, agar software bebas dari error, dan hasilnya harus benar-benar sesuai dengan kebutuhan yang sudah

didefinisikan sebelumnya. Pemeliharaan suatu software diperlukan, termasuk di dalamnya adalah pengembangan, karena software yang dibuat tidak selamanya hanya seperti itu. Ketika dijalankan mungkin saja masih ada error kecil yang tidak ditemukan sebelumnya, atau ada penambahan fitur-fitur yang belum ada pada software tersebut. Pengembangan diperlukan ketika adanya perubahan dari eksternal perusahaan seperti ketika ada pergantian sistem operasi, atau perangkat lainnya.

Penerapan model *waterfall* dalam membangun suatu perangkat lunak memiliki beberapa kelebihan dan kelemahan. Kelebihan menggunakan model *waterfall* adalah[KHO-11]:

- Kualitas dari sistem yang dihasilkan akan baik. Ini dikarenakan oleh pelaksanaannya secara bertahap. Sehingga tidak terfokus pada tahapan tertentu.
- Document pengembangan sistem sangat terorganisir, karena setiap fase harus terselesaikan dengan lengkap sebelum melangkah ke fase berikutnya. Jadi setiap fase atau tahapan akan mempunyai dokumen tertentu.
- Metode ini masih lebih baik digunakan walaupun sudah tergolong kuno, daripada menggunakan pendekatan asal-asalan. Selain itu, metode ini juga masih masuk akal jika kebutuhan sudah diketahui dengan baik.

Kelemahan menggunakan model *waterfall* adalah[KHO-11]:

- Diperlukan majemen yang baik, karena proses pengembangan tidak dapat dilakukan secara berulang sebelum terjadinya suatu produk.
- Kesalahan kecil akan menjadi masalah besar jika tidak diketahui sejak awal pengembangan yang berakibat pada tahapan selanjutnya.
- Pelanggan sulit menyatakan kebutuhan secara eksplisit sehingga tidak dapat mengakomodasi ketidakpastian pada saat awal pengembangan.
- Pelanggan harus sabar, karena pembuatan perangkat lunak akan dimulai ketika tahap desain sudah selesai. Sedangkan pada tahap sebelum desain bisa memakan waktu yang lama.
- Pada kenyataannya, jarang mengikuti urutan sekuensial seperti pada teori. Iterasi sering terjadi menyebabkan masalah baru.

2.2 *Object-Oriented Analysis and Design (OOAD)*

Object-Oriented Analysis and Design (OOAD) mengacu pada penerapan pemodelan berorientasi objek untuk analisis sistem atau persyaratan sistem dan untuk merancang sistem berorientasi objek yang memenuhi persyaratan. Sistem ini biasanya termasuk dalam: organisasi (misalnya perusahaan), unit bisnis, sistem perangkat lunak yang ada dalam spesifikasi persyaratan ditulis, sebuah data gudang atau bahkan sistem hardware.[SOL-12]

Skripsi ini menggunakan pendekatan OOAD karena menjadi acuan yang konsisten dari analisis, perancangan sampai pemrograman, memudahkan abstraksi data dan informasi, software reuse, mudah perawatan dan implementasi yang fleksibel.[SOL-12] Pemodelan untuk OOAD dalam skripsi ini dikerjakan dengan UML dan diterapkan menggunakan bahasa pemrograman Java.

Pendekatan OOAD menggunakan UML bertujuan untuk tetap dekat dengan domain masalah dan memungkinkan untuk pemetaan langsung ke berbagai bahasa pemrograman, khususnya ke bahasa pemrograman berorientasi objek.[SOL-12] UML menyediakan media komunikasi yang ideal antara aktor berbagai analisis sistem dan proses pembangunan, dari pakar domain untuk analisis, desainer, dan implementor maintainer.[SOL-12] Bahasa grafis seperti Unified Modeling Language (UML) memungkinkan untuk desain diagram yang memberikan pandangan selektif, hanya berfokus pada unsur-unsur yang relevan dengan titik-titik yang saat ini ingin di gambarkan.[SOL-12]

Bahasa pemrograman Java cocok untuk mengimplementasikan pemodelan menggunakan OOAD karena Java menggunakan pemrograman berorientasi objek yang membuat program dapat dibuat secara modular dan dapat dipergunakan kembali.[MUL-13] *Libraries networking* yang terintegrasi pada Java membuat aplikasi terdistribusi secara mudah.[MUL-13] Kemampuan mendeteksi error Java lebih teliti dibandingkan bahasa pemrograman lain.[MUL-13] Source code maupun program Java dapat dengan mudah dibawa ke platform yang berbeda-beda tanpa harus dikompilasi ulang.[MUL-13] Selain kelebihan yang telah disebutkan diatas Java juga mempunyai kemampuan untuk membuat suatu program yang dapat melakukan beberapa pekerjaan secara sekaligus dan simultan, didesain untuk dapat dijalankan pada lingkungan yang dinamis dengan maksud

perubahan pada suatu class dengan menambahkan properties ataupun method dapat dilakukan tanpa mengganggu program yang menggunakan class tersebut.[MUL-13]

2.2.1 *Unified Modeling Language (UML)*

UML adalah Unified Modeling Language yang merupakan bahasa berorientasi objek pemodelan diagram. Karenanya mereka menggunakan diagram untuk mendokumentasikan dekomposisi berbasis obyek dari sistem dan untuk menunjukkan interaksi antara obyek dan dinamika benda tersebut. Alasan untuk menggunakan UML adalah[SOL-12]:

- Sebuah tampilan grafis menyederhanakan pemahaman tentang sistem.
- UML diagram menyediakan mekanisme komunikasi yang nyaman antara semua pemain peran dalam bidang pengembangan perangkat lunak dari klien ke analis bisnis untuk desainer arsitek dan pengembang. Komunikasi yang meningkat biasanya mengurangi risiko kegagalan, meningkatkan produktivitas, meningkatkan kualitas, memfasilitasi pemeliharaan dan membuat sistem dinamis dan biasanya hasil dalam kompleksitas berkurang.
- UML adalah pemodelan berorientasi objek yang mendukung semua bahasa standar konsep berorientasi obyek.

UML mencakup sembilan diagram yaitu[BOO-98]:

1. *Class diagram*

Class diagram menunjukkan satu set kelas, interface, dan kolaborasi dan hubungan mereka. Diagram ini adalah diagram yang paling umum ditemukan dalam pemodelan sistem berorientasi objek. *Class diagram* mengatasi tampilan desain statis dari sebuah sistem. *Class diagram* yang mencakup kelas yang aktif mengatasi proses tampilan statis dari sebuah sistem.

2. *Object diagram*

Object diagram menunjukkan satu set objek dan hubungan mereka. *Object diagram* merupakan *snapshot* statis dari contoh hal-hal yang ditemukan dalam *class diagram*. Diagram ini mengatasi tampilan desain statis atau tampilan proses statis dari sebuah sistem seperti halnya diagram kelas, tetapi dari perspektif nyata atau kasus prototipikal.

3. Use case diagram

Use case diagram menunjukkan satu set kasus penggunaan dan aktor (jenis khusus dari kelas) dan hubungannya. Gunakan *use case diagram* mengatasi kasus penggunaan tampilan statis dari sebuah sistem. Diagram ini sangat penting dalam mengatur dan pemodelan perilaku sistem.

4. Sequence diagram & Collaboration diagram

Sequence diagram dan *collaboration diagram* adalah jenis diagram interaksi. Menampilkan sebuah interaksi, yang terdiri dari satu set objek dan hubungan mereka, termasuk pesan yang dapat dikirim antara mereka. Diagram interaksi mengatasi tampilan dinamis dari sebuah sistem. *Sequence diagram* adalah diagram interaksi yang menekankan waktu permintaan pesan, *collaboration diagram* adalah diagram interaksi yang menekankan organisasi struktural dari objek yang mengirim dan menerima pesan. *Sequence diagram* dan *collaboration diagram* adalah isomorfik, yang berarti bahwa anda dapat mengambil satu dan mentransformasikannya menjadi yang lain.

5. Statechart diagram

Statechart diagram menampilkan keadaan mesin, yang terdiri dari keadaan-keadaan, transisi, peristiwa, dan kegiatan. *Statechart diagram* mengatasi tampilan dinamis dari sebuah sistem. *Statechart diagram* sangat penting dalam pemodelan perilaku antarmuka, kelas, atau kolaborasi dan menekankan pada kejadian yang diperintahkan dari perilaku suatu objek, yang terutama berguna dalam pemodelan sistem reaktif.

6. Activity diagram

Activity diagram adalah diagram jenis khusus dari statechart diagram yang menunjukkan aliran dari aktivitas ke aktivitas dalam suatu sistem. *Activity diagram* mengatasi tampilan dinamis dari sebuah sistem. *Activity diagram* sangat penting dalam pemodelan fungsi sistem dan menekankan aliran kontrol antara objek-objek.

7. Component diagram

Component diagram menunjukkan organisasi dan ketergantungan antar satu set komponen. *Component diagram* mengatasi pandangan implementasi statis dari sebuah sistem. *Component diagram* terkait dengan class diagram dalam

komponen yang biasanya memetakan satu atau lebih kelas, interface, atau kolaborasi.

8. *Deployment diagram*

Deployment diagram menunjukkan konfigurasi dari proses *run-time node* dan komponen yang ada pada mereka. *Deployment diagram* mengatasi arsitektur tampilan pengembangan statis. *Deployment diagram* terkait dengan component diagram dalam sebuah node biasanya membungkus satu atau lebih komponen.

Ini bukan keseluruhan dari daftar diagram. Aplikasi lain dapat menggunakan UML untuk memberikan bentuk diagram lainnya, meskipun sembilan diagram tadi yang paling umum yang akan anda hadapi dalam praktek.[BOO-98] Skripsi ini hanya menggunakan 5 diagram dari 9 diagram yang telah dijabarkan tadi, yaitu: *class diagram*, *use case diagram*, *sequence diagram*, *activity diagram* dan *statechart diagram*.

2.2.2 Pemrograman Berorientasi Objek dan Java

Pemrograman berorientasi objek merupakan paradigma pemrograman yang berorientasikan kepada objek. Semua data dan fungsi di dalam paradigma ini dibungkus dalam *kelas-kelas* atau *objek-objek*. Bandingkan dengan logika pemrograman terstruktur. Setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya. Ada beberapa konsep dasar pemrograman berorientasi objek sebagai berikut[LIN-99]:

- Object

Objek adalah entitas runtime penting dalam metode berorientasi objek. Objek dapat menandai lokasi, rekening bank, dan tabel data atau catatan bahwa program harus ditangani.

- Class

Kelas adalah seperangkat benda dengan sifat yang mirip (atribut), perilaku umum (operasi), dan memiliki hubungan umum dengan objek lain. Kelengkapan data dan kode objek dapat dibuat pengguna didefinisikan tipe data dengan bantuan kelas.

- Data abstraction

Data abstraksi mengacu pada tindakan mewakili gambaran penting tanpa termasuk rincian latar belakang atau penjelasan. Kelas menggunakan konsep abstraksi dan didefinisikan sebagai daftar atribut abstrak seperti ukuran, biaya dan fungsi beroperasi pada atribut tersebut. Mereka meringkas semua sifat penting dari objek yang akan dibuat. Kelas menggunakan konsep abstraksi data dan ini disebut sebagai Tipe Data Abstrak (ADT).

- Data encapsulation

Data enkapsulasi berarti membungkus data dan fungsi ke dalam satu unit (yaitu kelas). Ini adalah fitur yang paling berguna dari kelas. Data ini tidak mudah untuk sampai ke dunia luar dan hanya fungsi-fungsi yang diapit oleh kelas dapat mengaksesnya. Fungsi ini memberikan batas antara data obyek dan program. Ini insulasi data dari akses langsung dengan program ini disebut sebagai *Data hiding*.

- Inheritance

Warisan adalah proses dimana objek dari satu kelas bisa mendapatkan properti dari objek dari kelas lain. Warisan mempertahankan klasifikasi hirarkis di mana kelas mewarisi dari orang tuanya. Warisan menyediakan fitur penting dari OOP yang dapat digunakan kembali. Warisan merupakan hubungan antara kelas dimana satu kelas adalah kelas induk dari kelas lain (turunan). Kelas turunan memegang sifat dan perilaku dari kelas dasar selain sifat dan perilaku dari kelas turunan.

- Polymorphism

Polimorfisme berarti kemampuan untuk mengambil lebih dari satu bentuk. Polimorfisme memainkan peran utama dalam mengalokasikan objek memiliki struktur internal yang berbeda untuk berbagi antarmuka eksternal yang sama. Ini berarti bahwa kelas umum operasi dapat diakses dengan cara yang sama meskipun kegiatan spesifik yang terkait dengan setiap operasi mungkin berbeda. Polimorfisme secara luas digunakan dalam menerapkan warisan.

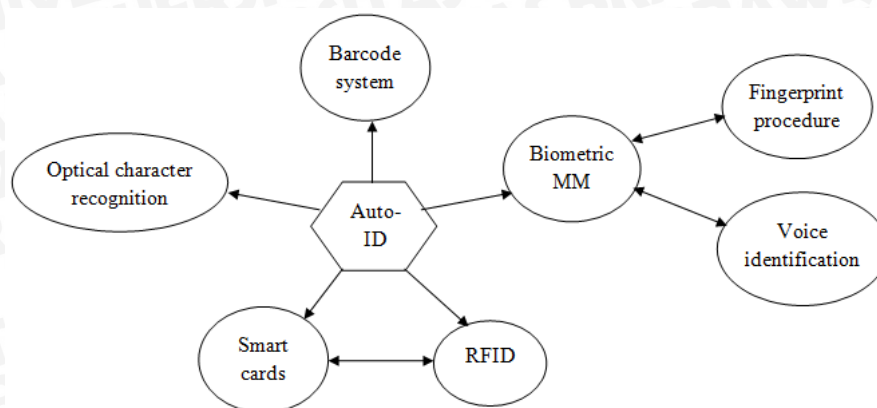
Bahasa pemrograman Java adalah tujuan umum bahasa berorientasi objek. Sintaks Java mirip dengan C dan C + +, tetapi dihilangkan banyak fitur yang membuat C dan C++ yang kompleks, membingungkan, dan tidak aman. Platform

Java awalnya dikembangkan untuk mengatasi masalah pengembangan perangkat lunak untuk perangkat konsumen jaringan. Hal ini dirancang untuk mendukung arsitektur beberapa host dan untuk memungkinkan pengiriman dari komponen software menjadi aman. Untuk memenuhi persyaratan ini, perjalanan kode harus bertahan melewati antar jaringan, beroperasi pada setiap klien, dan meyakinkan klien bahwa itu aman untuk dijalankan.[LIN-99]

2.3 Teknologi RFID

Sejarah RFID sebenarnya sudah dimulai sejak ditemukannya energi elektromagnetik oleh Michael Faraday tahun 1840. Tahun 1920 teknologi radar yang ditemukan digunakan untuk mendeteksi dan menentukan lokasi sebuah obyek melalui pantulan dari gelombang radio. RFID adalah sebuah kombinasi dari teknologi gelombang radio dan radar. [SRI-05] Aplikasi awal dari RFID yaitu penggunaan *tag* identifikasi generasi pertama digunakan ketika perang dunia ke-2 sebagai program *identification of friend or foe* (IFF) yaitu untuk mengidentifikasi pesawat kawan atau musuh. Pada tahun 1999 jutaan *tag* RFID telah digunakan dalam berbagai macam aplikasi termasuk untuk akses jalan tol, kartu akses pintu, dan *tracking* kontainer.[SRI-05].

Beberapa tahun terakhir ini, sistem identifikasi otomatis (Auto-ID) menjadi sangat populer dalam berbagai macam industri seperti industri jasa, pembelian dan distribusi barang, perusahaan manufaktur, dan lain-lain. Label *barcode* yang telah hadir dimana-mana merupakan pencetus terjadinya revolusi sistem identifikasi otomatis. Meskipun *barcode* sangat murah namun perkembangan dari *barcode* ini tersandung dengan kapasitas penyimpanannya yang rendah dan tidak adanya kemampuan untuk diprogram ulang. Solusi optimal secara teknis adalah dengan memanfaatkan sebuah *silicon chip* sebagai media penyimpanan yang kemudian diadopsi dalam sistem RFID.[FIN-03]



Gambar 2.2 Klasifikasi Sistem Identifikasi Otomatis

Sumber: [FIN-03]

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 yang disimpan saat sebuah *tag* diproduksi. Selain itu, ada beberapa sel lain yang dapat ditulis dan dibaca secara berulang.[WEI-05]

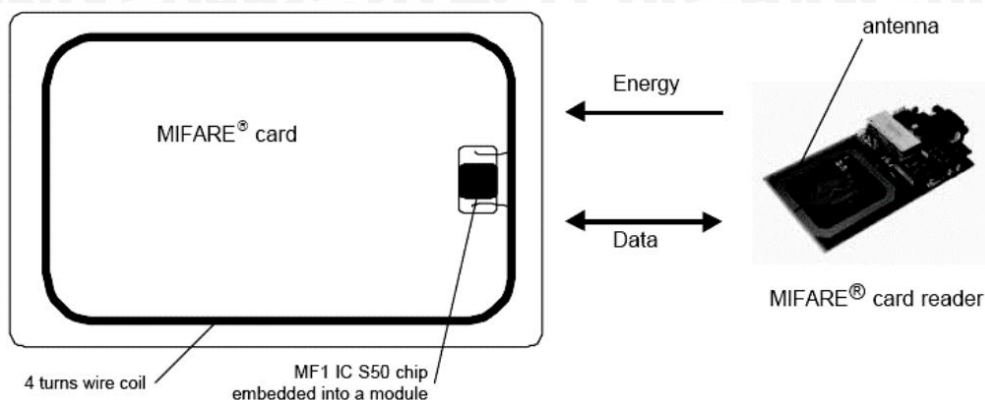
Tag RFID ada berbagai macam namun secara garis besar dibagi menjadi dua yaitu *tag* aktif dan *tag* pasif.

Tabel 2.1 Perbandingan *tag* aktif dan *tag* pasif

No	<i>tag</i> aktif	<i>tag</i> pasif
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: [HAM-10]

Mifare, yang nantinya akan digunakan penulis dalam penelitiannya, adalah salah satu *tag* RFID pasif, karena tidak ada perubahan data yang akan dilakukan terhadap *tag* RFID.



Gambar 2.3 Penampangan *tag* RFID dan RFID Reader merk mifare

Sumber: [HAM-10]

Gambar di atas menunjukkan *tag* pasif merk Mifare dengan tipe MF1 IC S50 dan alat *reader/writer* nya. Pada gambar ditunjukkan arah tanda panah *energy* dari *reader/writer* ke kearah *tag* karena catu daya *tag* didapatkan dari *reader/writer* nya. Data dengan arah tanda panah bolak-balik karena alat bias membaca dan menulis ke dalam *tag*. Adapun karakter penggunaan *tag* mifare ini adalah [HAM-10]:

1. transmisi datanya secara *contactless*
2. secara ekonomis harganya yang lebih murah dibandingkan dengan *tag* aktif ataupun *tag* pasif tipe lain
3. *transfer* data hingga 106 Kbit/s
4. penyimpanan data yang cukup (1Kbyte)
5. ketahanan data hingga 10 tahun
6. penulisan data hingga 100.000 kali.

RFID bukan hanya sekedar pengganti barcode. RFID juga memberikan berbagai macam fitur yang tidak diberikan barcode seperti, keamanan, cara pembacaan yang tidak harus secara langsung. Want [WAN-04] mengatakan bahwa nantinya RFID akan di gunakan secara global sebagai metode identifikasi otomatis.

RFID sudah merupakan bagian yang terintegrasi dalam kehidupan sehari-hari yang dapat meningkatkan produktivitas dan kenyamanan. Dikatakan juga

bahwa RFID sangat berguna bagi peningkatan produktivitas industri serta sangat berguna bagi pengembangan perangkat lunak.[LAN-05]

2.4 Pengenalan Plat Nomor

Sistem parkir dalam skripsi ini menggunakan DTK ANPR SDK. DTK ANPR ini digunakan karena sesuai bila diimplementasikan dalam skenario yang dirancang dalam skripsi ini dan memiliki keuntungan teknis yang sesuai dengan tujuan dalam skripsi ini.

DTK ANPR SDK (Automatic Number Plate Recognition) adalah pengembangan perangkat lunak kit untuk license plate recognition (LPR).[SIA-13] *Library* ini digunakan agar dapat mengintegrasikan fungsi pengenalan plat nomor ke perangkat lunak yang akan dibuat sendiri.[SIA-13] Perangkat lunak ini dapat mendeteksi plat nomor dari gambar diam maupun sumber video.[SIA-13]

DTK ANPR SDK adalah inti dasar komponen untuk solusi sistem otomatis berurusan dengan pengenalan dan pengolahan plat nomor kendaraan untuk tahap selanjutnya.[SIA-13] Di bawah ini adalah sejumlah sistem ANPR yang banyak digunakan[SIA-13]:

1. Kendaraan Area parkir sistem manajemen (masuk / keluar, pelacakan, pengisian)
2. Jasa video keamanan (Pemantauan stasiun gas, dll)
3. Akses sistem kontrol (gerbang elektronik, hambatan atau pagar).
4. Layanan deteksi pencurian mobil (baik berdasarkan stasiun pemantauan statis atau peralatan kendaraan bergerak).
5. Solusi parkir otomatis
6. Pengolahan gambar otomatis yang ditangkap oleh photoradars

DTK ANPR bekerja dalam dua modus yang berbeda: modus foto tunggal dan mode Video.[SIA-13] Sementara modus gambar tunggal dioptimalkan untuk menangani gambar tunggal dengan resolusi tinggi, modus video dibuat untuk frame pemrosesan video secara real time.[SIA-13] Keuntungan teknis menggunakan DTK ANPR adalah[SIA-13]:

1. Pengenalan on stream video dan masih-gambar pada tingkat tinggi.
2. Mudah untuk mengkonfigurasi sistem pengenalan tanpa kompleksitas.

3. Video streaming pengenalan secara real time.
4. Konsumsi sumber daya substansial rendah.

2.5 Standardisasi Parkir

Parkir adalah keadaan tidak bergerak suatu kendaraan yang tidak bersifat sementara. Fasilitas parkir adalah lokasi yang ditentukan sebagai tempat pemberhentian kendaraan yang tidak bersifat sementara untuk melakukan kegiatan pada suatu kurun waktu. Kawasan parkir adalah kawasan atau areal yang memanfaatkan badan jalan sebagai fasilitas parkir dan terdapat pengendalian parkir melalui pintu masuk.

Ukuran lebar pintu keluar-masuk dapat ditentukan, yaitu lebar 3 meter dan panjangnya harus dapat menampung tiga mobil berurutan dengan jarak antarmobil (spacing) sekitar 1,5 meter. Oleh karena itu, panjang-lebar pintu keluar masuk minimum 15 meter[DEP-96].

1. Pintu Masuk dan Keluar Terpisah

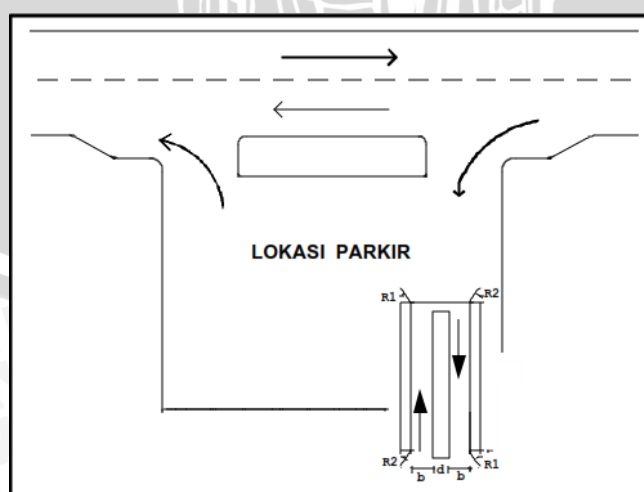
Satu jalur : Dua jalur:

$$b = 3,00 - 3,50 \text{ m} \quad b = 6,00 \text{ m}$$

$$d = 0,80 - 1,00 \text{ m} \quad d = 0,80 - 1,00 \text{ m}$$

$$R_1 = 6,00 - 6,50 \text{ m} \quad R_1 = 3,50 - 5,00 \text{ m}$$

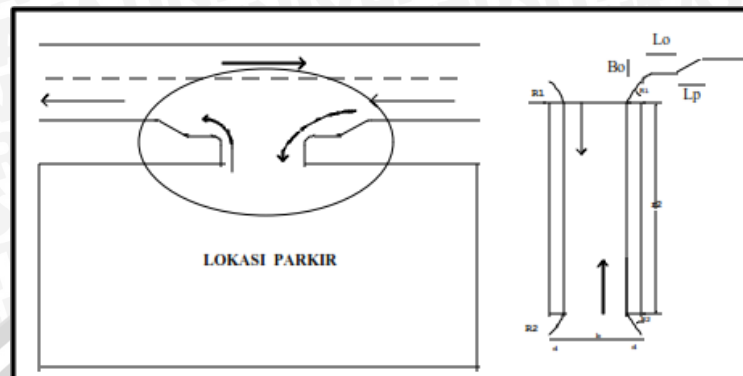
$$R_2 = 3,50 - 4,00 \text{ m} \quad R_2 = 1,00 - 2,50 \text{ m}$$



Gambar 2.4 Pintu Masuk dan Keluar Terpisah

Sumber: [DEP-96]

2. Pintu Masuk dan Keluar Menjadi Satu



Gambar 2.5 Pintu Masuk dan Keluar Menjadi Satu

Sumber: [DEP-96]

Dalam skripsi ini akan digunakan pintu masuk dan keluar terpisah, dikarenakan untuk memperlancar jalannya proses keluar masuk parkir. Selain untuk memperlancar jalannya proses keluar masuk parkir, adanya proses yang sedikit berbeda antara proses keluar dengan proses masuk area parkir.

Sesuai dengan jenis fasilitasnya, tata cara parkir adalah sebagai berikut[DEP-96]:

1. Fasilitas parkir tanpa pengendalian parkir :
 - a. dalam melakukan parkir, juru parkir dapat memandu pengemudi kendaraan;
 - b. juru parkir memberi karcis bukti pembayaran sebelum kendaraan meninggalkan ruang parkir;
 - c. juru parkir harus mengenakan seragam dan identitas.
2. Fasilitas parkir dengan pengendalian parkir (menggunakan pintu masuk/keluar):
 - a. pada pintu masuk, baik dengan petugas maupun dengan pintu otomatis, pengemudi harus mendapatkan karcis tanda parkir, yang mencantumkan jam masuk (bila diperlukan, petugas mencatat nomor kendaraan);
 - b. dengan dan tanpa juru parkir, pengemudi memarkirkan kendaraan sesuai dengan tata -cara parkir;

- c. pada pintu keluar, petugas harus memeriksa kebenaran karcis tanda parkir, mencatat lama parkir, menghitung tarif parkir sesuai dengan ketentuan, menerima pembayaran parkir dengan menyerahkan karcis bukti pembayaran pada pengemudi.

Fasilitas parkir dengan pengendalian parkir digunakan dalam skripsi ini. Pengendalian parkir yang telah dijelaskan diatas dilakukan secara otomatis dalam skripsi ini, tanpa mengurangi hal-hal penting yang telah dijabarkan.

2.6 Keamanan Sistem

Keamanan komputer adalah menjamin data atau informasi tidak dibaca, tidak dimodifikasi oleh orang lain yang tidak diberi otorisasi. Sementara itu kebutuhan keamanan sistem komputer dapat dikategorikan menjadi aspek-aspek sebagai berikut[APR-12]:

1. Privacy / Confidentiality

Inti utama aspek privacy atau confidentiality adalah usaha untuk menjaga informasi dari orang yang tidak berhak mengakses. Privacy lebih kearah data-data yang sifatnya privat sedangkan confidentiality biasanya berhubungan dengan data yang diberikan ke pihak lain untuk keperluan tertentu (misalnya sebagai bagian dari pendaftaran sebuah servis) dan hanya diperbolehkan untuk keperluan tertentu tersebut.

2. Integrity

Aspek ini menekankan bahwa informasi tidak boleh diubah tanpa seijin pemilik informasi. Adanya virus, trojan horse, atau pemakai lain yang mengubah informasi tanpa ijin merupakan contoh masalah yang harus dihadapi. Sebuah e-mail dapat saja “ditangkap” (intercept) di tengah jalan, diubah isinya (altered, tampered, modified), kemudian diteruskan ke alamat yang dituju. Dengan kata lain, integritas dari informasi sudah tidak terjaga. Penggunaan encryption dan digital signature, misalnya, dapat mengatasi masalah ini.

3. Authentication

Aspek ini berhubungan dengan metoda untuk menyatakan bahwa informasi betul-betul asli, orang yang mengakses atau memberikan

informasi adalah betul-betul orang yang dimaksud, atau server yang kita hubungi adalah betul-betul server yang asli.

4. Availability

Aspek availability atau ketersediaan berhubungan dengan ketersediaan informasi ketika dibutuhkan. Sistem informasi yang diserang atau dijebol dapat menghambat atau meniadakan akses ke informasi.

Availability dijadikan acuan kebutuhan non-fungsional sistem dalam skripsi ini. Privacy dan Authentication akan dimasukkan dalam kebutuhan fungsional sistem. Privacy dalam skripsi ini adalah keterbatasan hak akses untuk mengelola pengaturan administrasi yang hanya dapat dilakukan oleh petugas. Authentication dalam skripsi ini diimplementasikan pada teknologi RFID yang dirancang dalam skripsi ini.

2.7 Pengujian Perangkat Lunak

Pengujian perangkat lunak merupakan elemen penting dari penjaminan kualitas perangkat lunak dan merepresentasikan tinjauan utama spesifikasi, desain, dan pembuatan kode [PRE-10]. Dalam skripsi ini akan dilakukan 2 macam pengujian, yaitu pengujian validasi dan pengujian performa.

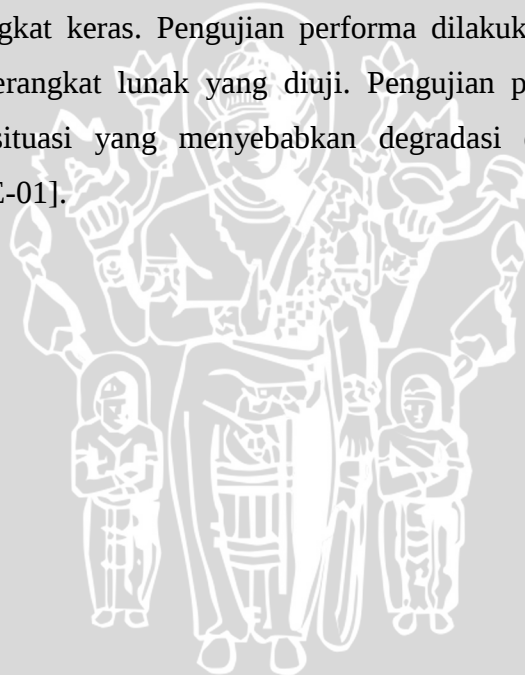
2.7.1 Pengujian Validasi

Pada kulminasi pengujian terintegrasi, perangkat lunak secara lengkap dirakit sebagai suatu paket; kesalahan *interfacing* telah diungkap dan dikoreksi, dan seri akhir dari pengujian perangkat lunak, yaitu pengujian validasi dapat dimulai. Validasi dapat ditentukan dengan berbagai cara, tetapi definisi yang sederhana adalah bahwa validasi berhasil bila perangkat lunak berfungsi dengan cara yang dapat diharapkan secara bertanggung jawab oleh pelanggan. Validasi perangkat lunak dicapai melalui sederetan pengujian *black-box* yang memperlihatkan konformitas dengan persyaratan. Rencana pengujian menguraikan kelas-kelas pengujian yang akan dilakukan, dan prosedur pengujian menentukan *test case* spesifik yang akan digunakan untuk mengungkap kesalahan dalam konformitas dengan persyaratan. Baik rencana dan prosedur didesain untuk memastikan apakah semua persyaratan fungsional dipenuhi; semua persyaratan

kinerja dicapai; dokumentasi benar dan direkayasa oleh manusia: dan persyaratan lainnya dipenuhi (transportabilitas, kompatibilitas, pembetulan kesalahan, maintainabilitas) [PRE-01].

2.7.2 Pengujian Performa

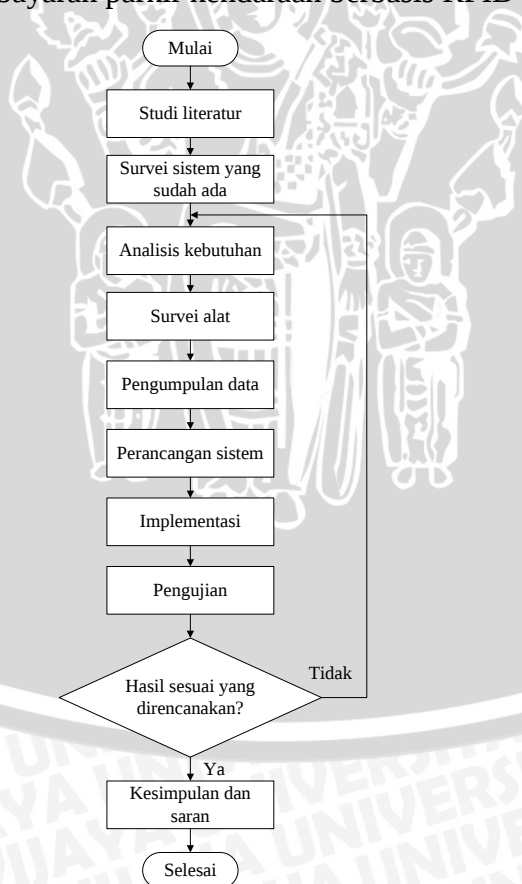
Setelah semua langkah pengujian perangkat lunak secara terstruktur dilakukan, maka perlu dilakukan pengujian sistem di lingkungan dimana dia bekerja untuk mengetahui performa dari perangkat lunak tersebut. Pengujian sistem dirancang untuk menguji kinerja *run-time* dari perangkat lunak dalam konteks sistem terintegrasi. Pengujian performa melibatkan *monitoring* pemanfaatan sumber daya dari perangkat lunak yang diuji seperti perangkat lunak pendukung dan perangkat keras. Pengujian performa dilakukan secara spesifik sesuai dengan tipe perangkat lunak yang diuji. Pengujian performa bertujuan untuk mengungkap situasi yang menyebabkan degradasi dan kemungkinan kegagalan sistem [PRE-01].



BAB III

METODE PENELITIAN

Bab ini menjelaskan langkah-langkah pengembangan proses perangkat lunak yang akan ditempuh dalam penyusunan skripsi. Pengembangan proses perangkat lunak sistem dikembangkan dengan metode pengembangan perangkat lunak SDLC menggunakan model *waterfall*. Langkah-langkah pengembangan yang akan ditempuh yaitu studi literatur, survei sistem yang sudah ada, pengumpulan data, analisis kebutuhan, survei alat, perancangan sistem, implementasi, pengujian dan pengambilan kesimpulan dan saran dari aplikasi perangkat lunak yang akan dibuat. Kesimpulan dan saran disertakan sebagai catatan atas aplikasi dan kemungkinan arah pengembangan aplikasi selanjutnya. Gambar 3.1 menunjukkan desain penelitian sistem perangkat lunak untuk simulasi keluar masuk dan pembayaran parkir kendaraan berbasis RFID secara umum.



Gambar 3.1 Diagram Blok Metode Penelitian

Sumber: Metode Penelitian

3.1 Studi Literatur

Studi literatur digunakan untuk mempelajari berbagai referensi yang diambil dari buku, jurnal, laporan penelitian, maupun artikel. Referensi utama yang diperlukan untuk menunjang penulisan skripsi ini berhubungan dengan:

- Model *waterfall*
- *Object-Oriented Analysis and Design (OOAD)*
- Teknologi RFID
- Pengenalan Plat Nomor
- Standardisasi Parkir
- Keamanan Sistem
- Pengujian Perangkat Lunak

Referensi-referensi utama tersebut telah dijabarkan di bab II.

3.2 Survei Sistem yang Sudah Ada

Survei sistem yang sudah ada ini digunakan untuk mempelajari cara kerja sistem perparkiran di Indonesia. Kegunaan mempelajari sistem yang sudah ada adalah sebagai dasar untuk pembuatan modif sistem dalam pengenalan sehingga dapat menghasilkan sistem baru yang lebih efisien dan aman. Survei ini juga bertujuan sebagai dasaran untuk mengetahui data-data apa saja yang diperlukan dalam sistem perparkiran yang akan dirancang ini. Survei akan dilakukan dengan cara mendatangi dan mempelajari cara kerja beberapa parkiran apartemen, mencari referensi sistem parkir yang sudah ada dari internet dan mendata seberapa aman, efektif dan nyaman sistem yang sudah ada tersebut bagi pengguna.

3.3 Analisis Kebutuhan

Analisis kebutuhan bertujuan untuk menganalisis dan mendapatkan semua kebutuhan yang diperlukan Sistem Perangkat Lunak Untuk Simulasi Keluar Masuk dan Pembayaran Parkir Kendaraan Berbasis RFID. Analisis kebutuhan disesuaikan dengan lokasi dan variabel penelitian, menentukan kebutuhan data yang akan digunakan, dan mempersiapkan alat dan bahan penelitian.

Metode analisis yang digunakan adalah *Object Oriented Analysis and Design* (OOAD) dengan menggunakan bahasa pemodelan UML (*Unified Modeling Language*). Pemrograman berorientasi objek (*object-oriented programming*) merupakan paradigma pemrograman yang berorientasikan kepada objek. Semua data dan fungsi di dalam paradigma ini dibungkus dalam *kelas-kelas* atau *objek-objek*. Bandingkan dengan logika pemrograman terstruktur. Setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya dalam artian setiap objek dapat berdiri sendiri.

3.4 Survei Alat

Survei alat diperlukan untuk pencarian alat yang tepat untuk sistem yang akan dirancang ini. Dengan adanya survei alat ini, perancang dapat mengetahui secara pasti cara kerja alat secara spesifik dan menggabungkan cara kerja alat tersebut dengan rancangan cara kerja sistem sehingga menghasilkan sistem yang sesuai dengan kebutuhan. Alat yang akan di survei adalah RFID reader dan tagnya.

3.5 Pengumpulan Data

Pada tahap ini dilakukan pengumpulan data yang dibutuhkan untuk mencapai tujuan dari skripsi ini, yaitu otomatisasi keluar masuk parkir dan pembayaran parkir. Data yang diperlukan untuk menjaga keamanan parkir adalah data pengemudi yang dihubungkan dengan data plat nomor kendaraan. Kedua data tersebut diperlukan karena saat pelacakan dilakukan, data yang diperlukan adalah siapa yang membawa kendaraan dan kendaraan apa yang dibawa. Data yang diperlukan untuk pembayaran parkir adalah plat nomor kendaraan dan waktu keluar-masuk kendaraan. Cukup dengan kedua data tersebut biaya parkir dapat dihitung dan pembayaran parkir dapat dilakukan.

3.6 Perancangan Sistem

Perancangan sistem dibangun dengan pendekatan berorientasi objek menggunakan bahasa pemodelan UML berdasarkan hasil pengambilan data dan analisis kebutuhan yang telah dilakukan. Diagram *Use Case* digunakan untuk

mendeskripsikan kebutuhan-kebutuhan dan fungsionalitas sistem dari perspektif *end-user* dan menunjukkan aksi-aksi yang dapat dilakukan oleh *user*. Siklus kerja sistem ditunjukkan dengan diagram *statechart*. Diagram *statechart* adalah diagram yang menggambarkan kelakuan sistem secara keseluruhan. Untuk menunjukkan keseluruhan dari aliran kontrol digunakan *Activity Diagram*. *Activity diagram* dapat digunakan untuk menjelaskan alur kerja dari komponen suatu sistem. Sedangkan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah kejadian untuk menghasilkan keluaran tertentu digunakan *Sequence Diagram*. *Sequence Diagram* diawali dari apa yang memicu aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan keluaran apa yang dihasilkan. *Class diagram* menggambarkan struktur statis dari sebuah sistem. *Class Diagram* adalah diagram yang menunjukkan class-class yang ada dari sebuah sistem dan hubungannya secara logika.

3.7 Implementasi

Implementasi aplikasi akan dilakukan dengan mengacu kepada perancangan sistem keamanan dan pengatur keluar masuk kendaraan dalam suatu area apartemen. Implementasi perangkat lunak dilakukan dengan menggunakan bahasa Java. Hasil yang akan diberikan dari sistem ini adalah kumpulan data keluar masuknya kendaraan dalam suatu area dan akses keluar masuknya kendaraan dengan berbasis RFID beserta transaksi pengisian saldonya. Implementasi yang ditampilkan berupa simulasi cara kerja sistem.

3.8 Pengujian

Pengujian akan dilakukan berdasarkan implementasi yang telah dibuat secara simulasi menggunakan metode pengujian validasi dan pengujian performa. Pengujian validasi digunakan untuk menguji kesesuaian fungsional sistem. Sedangkan pengujian performa digunakan untuk menguji tentang berapa lama waktu proses keluar masuk kendaraan yang akan dibutuhkan oleh sistem yang dirancang dalam skripsi ini, kemudian dibandingkan dengan waktu proses keluar masuk kendaraan yang dibutuhkan pada sistem perparkiran yang ada saat ini.

3.9 Pengambilan Kesimpulan dan Saran

Pengambilan kesimpulan dilakukan setelah semua tahapan perancangan, implementasi dan pengujian sistem aplikasi telah selesai dilakukan. Kesimpulan diambil dari hasil pengujian dan analisis terhadap sistem yang dibangun. Tahap terakhir dari penulisan adalah saran yang dimaksudkan untuk memperbaiki kesalahan-kesalahan yang terjadi dan menyempurnakan penulisan serta untuk memberikan pertimbangan atas pengembangan aplikasi selanjutnya.



BAB IV

PERANCANGAN

Bab ini membahas mengenai analisis kebutuhan dan perancangan sistem perangkat lunak untuk simulasi keluar masuk dan pembayaran parkir kendaraan berbasis RFID. Analisis kebutuhan terdiri atas lima langkah yaitu melakukan penjabaran tentang gambaran umum Sistem Perangkat Lunak Untuk Simulasi Keluar Masuk dan Pembayaran Parkir Kendaraan Berbasis RFID, melakukan proses identifikasi aktor yang terlibat dalam sistem perangkat lunak, melakukan proses analisis data yang diperlukan, membuat daftar kebutuhan pengguna berdasarkan penjabaran gambaran umum perangkat lunak dan menggunakan pemodelan diagram *use case* untuk menggambarkan kebutuhan tersebut. Proses perancangan perangkat lunak memiliki lima tahap, yaitu perancangan arsitektural, perancangan basis data, pemodelan diagram *class* untuk menggambarkan perancangan struktur *class – class* yang menyusun sistem, pemodelan diagram *sequence* untuk menggambarkan interaksi antar objek atau *class* di dalam sistem, dan perancangan antarmuka pengguna dari sistem.

4.1 Analisis Kebutuhan

Proses analisis kebutuhan mengacu pada gambaran umum sistem perangkat lunak untuk simulasi keluar masuk dan pembayaran parkir kendaraan berbasis RFID. Proses analisis kebutuhan ini diawali dengan penjabaran gambaran umum sistem, identifikasi aktor – aktor yang terlibat dalam sistem, analisis data yang akan disimpan, penjabaran tentang daftar kebutuhan dan kemudian memodelkannya ke dalam diagram *use case*. Analisis kebutuhan ini bertujuan untuk menggambarkan kebutuhan – kebutuhan yang harus disediakan oleh sistem agar dapat memenuhi kebutuhan pengguna.

4.1.1 Gambaran Umum Sistem Perangkat Lunak Untuk Simulasi Keluar Masuk dan Pembayaran Parkir Kendaraan Berbasis RFID

Pembahasan gambaran umum Sistem Perangkat Lunak Untuk Simulasi Keluar Masuk dan Pembayaran Parkir Kendaraan Berbasis RFID terdiri atas dua

bagian, yaitu deskripsi umum sistem perangkat lunak untuk simulasi keluar masuk dan pembayaran parkir kendaraan berbasis RFID dan cara penggunaan Sistem Perangkat Lunak Untuk Simulasi Keluar Masuk dan Pembayaran Parkir Kendaraan Berbasis RFID.

4.1.1.1 Deskripsi Sistem Perangkat Lunak Untuk Simulasi Keluar Masuk dan Pembayaran Parkir Kendaraan Berbasis RFID

Sistem perangkat lunak yang akan dikembangkan sebagai simulasi dalam proyek skripsi ini adalah sistem perangkat lunak untuk simulasi keluar masuk dan pembayaran parkir kendaraan berbasis RFID. Sistem perangkat lunak untuk simulasi keluar masuk dan pembayaran parkir kendaraan berbasis RFID adalah sistem perangkat lunak yang digunakan untuk mengatur proses keluar masuk kendaraan di area parkir dan mengatur administrasi dari parkir. Perangkat lunak ini nantinya bisa diimplementasikan di semua wilayah yang menerapkan sistem parkir yang berbayar yang efisien dan otomatis.

Sistem perangkat lunak untuk keamanan dan pembayaran parkir di area apartemen berbasis RFID memiliki 2 bagian utama yaitu, subsistem gerbang parkir dan subsistem administrasi.

a. Subsistem gerbang parkir

Subsistem gerbang parkir dari sistem perangkat lunak untuk simulasi keluar masuk dan pembayaran parkir kendaraan berbasis RFID adalah aplikasi yang dapat mengatur proses keluar masuknya kendaraan dalam area parkir, seperti mengenali pengemudi kendaraan dan kendaraan yang di bawa, mengurangi saldo kendaraan dan mengontrol kerja portal.

b. Subsistem administrasi

Subsistem administrasi dari sistem perangkat lunak untuk simulasi keluar masuk dan pembayaran parkir kendaraan berbasis RFID adalah aplikasi yang digunakan untuk mengelola elemen – elemen dalam sistem perangkat lunak tersebut. Proses – proses utama yang terdapat dalam aplikasi administrasi adalah sebagai berikut :

1. Login

Operasi ini bertujuan untuk memberi seleksi kepada pengguna sehingga hanya admin saja yang dapat mengelola sistem.

2. Mengelola Data Admin

Operasi ini bertujuan untuk mengolah data admin yang disimpan di tabel admin pada basis data. Operasi ini terdiri atas beberapa operasi, yaitu :

- Melakukan masukkan pada tabel admin.
- Melakukan hapus pada tabel admin.
- Melakukan perbaruhi pada tabel admin.

3. Mengelola Data Kendaraan

Operasi ini bertujuan untuk mengolah data kendaraan yang disimpan di tabel kendaraan pada basis data. Operasi ini terdiri atas beberapa operasi, yaitu :

- Melakukan masukkan pada tabel kendaraan.
- Melakukan hapus pada tabel kendaraan.
- Melakukan perbaruhi pada tabel kendaraan.

4. Mengelola Data Pengendara

Operasi ini bertujuan untuk mengolah data pengendara yang disimpan di tabel pengendara pada basis data. Operasi ini terdiri atas beberapa operasi, yaitu :

- Melakukan masukkan pada tabel pengendara.
- Melakukan hapus pada tabel pengendara.
- Melakukan perbaruhi pada tabel pengendara.

5. Mengelola Data Pengaturan

Operasi ini bertujuan untuk mengolah data pengaturan yang disimpan di tabel pengaturan pada basis data. Operasi ini terdiri atas beberapa operasi, yaitu :

- Melakukan perbarui pada tabel pengaturan.

6. Mengelola Data Saldo

Operasi ini bertujuan untuk mengolah data saldo yang disimpan di tabel saldo pada basis data. Operasi ini terdiri atas beberapa operasi, yaitu :

- Melakukan tambah saldo pada tabel saldo.
- Melakukan pengiriman sms pemberitahuan tentang sisa saldo kepada *user*.
- Melakukan pencetakan transaksi saldo baik secara keseluruhan maupun per transaksi.

7. Mengelola Data Log Parkir

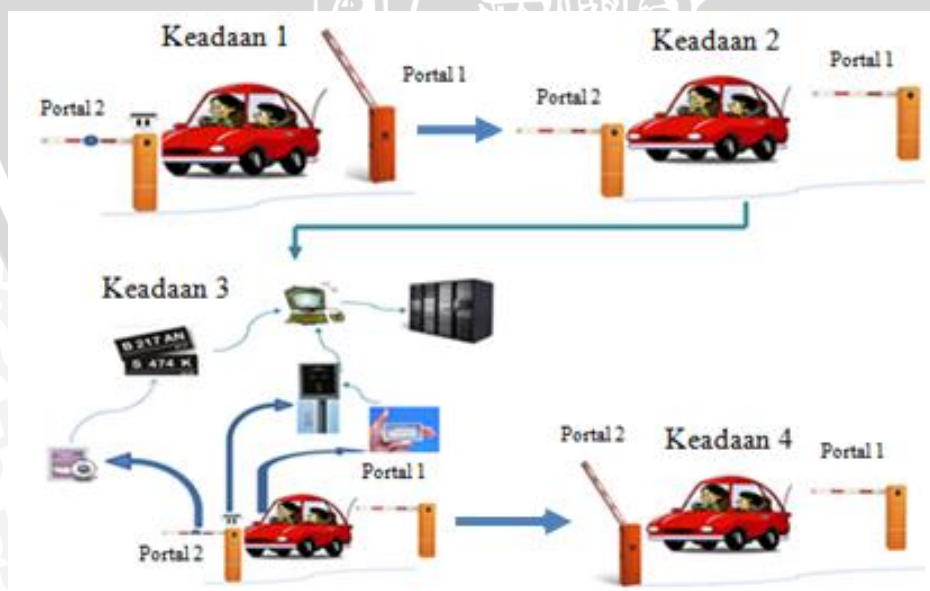
Operasi ini bertujuan untuk melakukan mengolah data log parkir yang disimpan di tabel logparkir pada basis data. Operasi ini terdiri atas beberapa operasi, yaitu :

- Melakukan pencarian transaksi tertentu menurut tanggal, nomor plat maupun rfid pada tabel logparkir.
- Melakukan pencetakan transaksi secara keseluruhan, tanggal tertentu, plat nomor tertentu maupun rfid tertentu pada tabel logparkir.
- Melakukan pelaporan kepada manager dan bidang keamanan bila terjadi kehilangan kendaraan.

4.1.1.2 Cara Penggunaan Perangkat Lunak untuk Keamanan dan Pembayaran Parkir di Area Apartemen Berbasis RFID

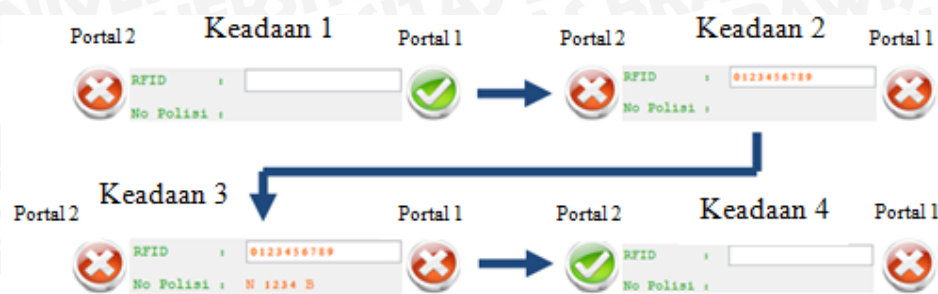
Dalam skripsi ini dirancang perangkat lunak untuk simulasi keluar masuk dan pembayaran parkir kendaraan berbasis RFID. Sebelum merancang dalam bentuk simulasi, penulis merancang perangkat lunak yang dapat diimplementasikan dalam dunia nyata. Pada gambar 4.1 dapat dilihat proses masuk kendaraan pada sistem ini adalah saat ada kendaraan masuk, melewati sensor yang terdapat pada portal 1, deteksi sensor guna menutup portal 1. Dalam

simulasi seperti Gambar 4.2 *icon* 1 yang awalnya berwarna hijau akan menjadi berwarna merah. Pengemudi memasukkan data pengemudi melalui tag RFID dan simulasinya akan sama, tetapi dalam simulasi ID akan ditampilkan dalam aplikasi. Pembacaan RFID menjadi pemicu untuk kamera mengambil gambar plat nomor kemudian diproses menjadi data teks, dalam simulasinya akan dimunculkan foto kendaraan dan akan diolah sehingga menghasilkan tampilan data teks di aplikasi. Data plat nomor, data RFID dan waktunya disimpan dalam basis data server. Proses masuknya data tersebut ke basis data disimulasikan dengan munculnya tulisan “SUKSES” dibawah tampilan data tersebut. Setelah proses selesai maka portal 2 yang berada di depan kendaraan terbuka, disimulasikan dengan *icon* 2 akan berubah dari warna merah menjadi warna hijau. Kemudian kendaraan memasuki area parkir melewati sensor yang berada pada portal utama sehingga sensor mendeteksi dan menyebabkan portal utama tertutup kembali, akan disimulasikan dengan hilangnya data plat nomor dan id pengguna dari tampilan dan *icon* pertama akan berubah menjadi warna merah kembali. Setelah portal utama menutup sempurna kemudian portal sekunder akan terbuka kembali dan disimulasikan dengan warna *icon* kedua menjadi hijau kembali.



Gambar 4.1 Rencana Penerapan Alur Subsistem Gerbang Parkir Secara Nyata

Sumber: Perancang



Gambar 4.2 Alur Subsistem Gerbang Parkir Secara Simulasi

Sumber: Perancang

Proses keluar tidak jauh berbeda dengan saat proses masuk. Perbedaan dengan proses masuk hanya pada saat setelah pembacaan tag RFID. Pada saat proses masuk tidak dilakukan apapun selain hanya membaca tag RFID, tetapi pada saat proses keluar setelah membaca tag RFID, saldo yang tersimpan akan dikurangi dengan waktu parkir dikalikan dengan tarif parkir perjamnya. Setiap orang memiliki RFID yang berbeda-beda dan setiap kendaraan dimiliki oleh satu RFID atau satu pemilik kendaraan. Kendaraan dapat keluar masuk dengan RFID yang berbeda-beda.

Bila terjadi kehilangan kendaraan, pemilik kendaraan dapat melaporkan kepada petugas. Kemudian petugas akan melacak kendaraan melalui database yang menyimpan data kendaraan yang keluar masuk beserta data pengemudinya. Dari data yang telah tercatat itu bisa dicari keterangan tentang data diri pengemudi dan data kendaraan secara lengkap. Bila sudah dipastikan data yang dimaksud adalah benar, maka petugas akan langsung mengirimkan sms laporan kehilangan kepada pihak manajer apartemen dan bagian keamanan yang bertugas.

Pada proses pengisian saldo, pengguna akan datang ke petugas dengan membawa tag RFID yang dimilikinya. Petugas akan memasukkan RFID orang yang membayarkan tagihan tersebut dan nominal yang dibayarkan. Data tersebut akan disimpan dan saldo akan diperbarui ke dalam basis data.

4.1.2 Identifikasi Aktor

Tahap ini adalah tahap untuk melakukan identifikasi terhadap aktor - aktor yang akan berinteraksi dengan Sistem Perangkat Lunak Untuk Simulasi Keluar

Masuk dan Pembayaran Parkir Kendaraan Berbasis RFID. Tabel 4.1 memperlihatkan aktor – aktor yang terlibat beserta penjelasannya masing-masing yang merupakan hasil dari proses identifikasi aktor.

Tabel 4.1 Identifikasi aktor

Aktor	Deskripsi
Pengendara	Pengendara adalah pengguna biasa yang hanya bisa menggunakan subsistem gerbang parkir dari sistem perangkat lunak untuk simulasi keluar masuk dan pembayaran parkir kendaraan berbasis RFID.
Administrator	<i>Administrator</i> adalah pengguna subsistem administrasi sistem perangkat lunak untuk simulasi keluar masuk dan pembayaran parkir kendaraan berbasis RFID yang memiliki hak untuk mengelola elemen-elemen yang terdapat dalam sistem tersebut lebih lanjut.

Sumber : Perancangan

4.1.3 Analisis Data

Analisis data bertujuan untuk mendapatkan struktur penyimpanan data yang dibutuhkan sistem perangkat lunak untuk simulasi keluar masuk dan pembayaran parkir kendaraan berbasis RFID. Struktur penyimpanan data pada sistem perangkat lunak untuk simulasi keluar masuk dan pembayaran parkir kendaraan berbasis RFID disusun berdasarkan analisis data sebagai berikut :

- Data admin yang terdiri atas *username*, *password* dan nama.
- Data kendaraan yang terdiri atas nomor polisi, merek, tipe, tahun dan rfid.
- Data pengendara yang terdiri atas rfid, nomor ktp, nama, jenis kelamin, alamat dan nomor *handphone*.
- Data saldo yang terdiri atas rfid, saldo dan waktu terakhir pencetakan.
- Data pengaturan yang terdiri atas jenis pengaturan dan isi dari jenis pengaturan tersebut.
- Data log saldo yang terdiri atas nomor, rfid, waktu pengisian saldo, jumlah saldo yang ditambahkan dan saldo akhir setelah diproses.
- Data log parkir yang terdiri atas nomor identitas, nomor polisi, rfid yang membawa kendaraan masuk ke dalam area parkir beserta waktu masuknya, rfid yang membawa kendaraan keluar dari area parkir beserta

waktu keluarnya dan sisa saldo yang dimiliki oleh pemilik kendaraan tersebut.

4.1.4 Daftar Kebutuhan

Daftar kebutuhan terdiri dari kebutuhan fungsional dan kebutuhan non-fungsional. Pada daftar kebutuhan akan dispesifikasikan menjadi dua yaitu spesifikasi kebutuhan fungsional subsistem gerbang parkir dan spesifikasi kebutuhan fungsional subsistem administrasi. Spesifikasi kebutuhan fungsional subsistem gerbang parkir ditunjukkan pada Tabel 4.2.

Tabel 4.2 Spesifikasi kebutuhan fungsional subsistem gerbang parkir

Nomor SRS	Kebutuhan	Use Case
SRS_001_01	Perangkat lunak harus mampu untuk membaca RFID dan memasukkannya ke dalam database.	Menunjukkan tag RFID
SRS_001_02	Perangkat lunak harus mampu untuk membaca plat nomor kendaraan lalu mengubahnya menjadi data teks dan menyimpannya ke dalam database	Membawa kendaraan untuk dapat diambil gambarnya

Sumber : Perancangan

Spesifikasi kebutuhan fungsional subsistem administrasi ditunjukkan pada Tabel 4.3.

Tabel 4.3 Spesifikasi kebutuhan fungsional subsistem administrasi

Nomor SRS	Kebutuhan	Use Case
SRS_002_01	Perangkat lunak harus menyediakan fasilitas untuk login sehingga hanya <i>administrator</i> saja yang dapat mengelola sistem.	Login
SRS_002_02	Perangkat lunak harus menyediakan fasilitas untuk melakukan pengolahan data admin.	Mengelola Data Admin
SRS_002_03	Perangkat lunak harus menyediakan fasilitas untuk melakukan	Mengelola Data Kendaraan

	pengolahan data kendaraan.	
SRS_002_04	Perangkat lunak harus menyediakan fasilitas untuk melakukan pengolahan data pengendara.	Mengelola Data Pengendara
SRS_002_05	Perangkat lunak harus menyediakan fasilitas untuk melakukan penambahan data saldo, dapat mengirimkan informasi saldo kepada <i>user</i> dan dapat mencetak transaksi saldo.	Mengelola Data Saldo
SRS_002_06	Perangkat lunak harus menyediakan fasilitas untuk melakukan pembaharuan isi dari data-data pengaturan.	Mengelola Data Pengaturan
SRS_002_07	Perangkat lunak harus menyediakan fasilitas untuk melakukan pencarian data <i>history</i> keluar masuknya kendaraan beserta saldonya sesuai dengan kebutuhan, pencetakan <i>history</i> dan pelaporan kepada manager apartemen dan bagian keamanan bila terjadi kehilangan.	Mengelola Data Log Parkir

Sumber : Perancangan

Daftar kebutuhan non-fungsional sistem perangkat lunak untuk simulasi keluar masuk dan pembayaran parkir kendaraan berbasis RFID ditunjukkan pada Tabel 4.4.

Tabel 4.4 Spesifikasi kebutuhan non-fungsional

Parameter	Deskripsi Kebutuhan
<i>Availability</i>	Perangkat lunak harus dapat beroperasi terus - menerus (24 jam).
<i>Response Time</i>	Perangkat lunak harus dapat berjalan dengan waktu proses 13,25 detik.
<i>Control</i>	Membatasi akses penggunaan perangkat lunak dengan menyediakan 2 sisi aplikasi, yaitu <i>Gerbang Keluar Masuk</i> dan <i>administrator</i> .
<i>Usability</i>	Perangkat lunak harus dapat dengan mudah digunakan pengguna.
<i>Security</i>	Perangkat lunak harus dapat menyimpan data yang terdeteksi dengan benar, adanya hak akses yang hanya memperbolehkan <i>administrator</i> untuk

	mengelola elemen-elemen sistem perparkiran dan dapat melaporkan kehilangan secara cepat kepada bagian keamanan dan pihak yang terkait.
--	--

Sumber : Perancangan

Availability dan *Usability* tidak diuji karena dalam skripsi ini sistem perangkat lunak untuk simulasi keluar masuk dan pembayaran parkir kendaraan berbasis RFID dilakukan secara simulasi. Dalam *security*, perangkat lunak harus dapat menyimpan data yang terdeteksi dengan benar, dapat disebut dengan *authentication* yang akan diuji dalam fungsional sistem yang mengacu pada use case subsistem gerbang parkir. Sedangkan adanya hak akses yang hanya memperbolehkan *administrator* untuk mengelola elemen-elemen sistem perparkiran yang termasuk dalam *privacy* dan laporan kehilangan secara cepat kepada pihak bagian keamanan dan pihak yang terkait atau *quick alert* akan diuji dalam fungsional sistem yang mengacu pada *use case* subsistem administrasi.

4.1.5 Analisis Use Case

Pemodelan diagram *use case* yang menggambarkan fungsionalitas sistem perangkat lunak untuk simulasi keluar masuk dan pembayaran parkir kendaraan berbasis RFID dibagi menjadi dua yaitu diagram *use case* untuk subsistem gerbang parkir dan diagram *use case* untuk subsistem administrasi. Diagram *use case* adalah salah satu diagram untuk memodelkan aspek perilaku sistem. Diagram *use case* menunjukkan sekumpulan *use case*, aktor, dan hubungannya. *Use case* merupakan fungsionalitas dari sistem yang diinisialisasi oleh aktor.

4.1.5.1 Analisis Use Case Subsistem Gerbang Parkir

Diagram *use case* ini melibatkan *user* sebagai aktor dan satu buah *use case*. Satu buah *use case* ini termasuk dalam bagian sistem perangkat lunak untuk simulasi keluar masuk dan pembayaran parkir kendaraan berbasis RFID. Satu buah *use case* ini juga akan disertai dengan skenario *use case* untuk menjelaskan rangkaian aktivitas yang terjadi di *use case* tersebut. Diagram *use case* untuk Aplikasi gerbang parkir ditunjukkan pada Gambar 4.3 berikut ini :



Gambar 4.3 Diagram *use case* subsistem gerbang parkir

Sumber : Perancangan

1. Skenario *Use Case* Menunjukkan RFID

Tabel 4.5 Skenario *Use case* Menunjukkan RFID

Skenario Kasus Pada Sistem	
Nomor <i>Use Case</i>	SRS_001_01
Nama	Menunjukkan RFID
Tujuan	Untuk menunjukkan tag RFID ke <i>reader</i> RFID.
Deskripsi	<i>Use Case</i> ini menjelaskan bagaimana <i>user</i> melakukan proses menunjukkan tag RFID ke <i>reader</i> RFID.
Aktor	<i>User</i>
Skenario Utama	
Kondisi Awal	Perangkat lunak sudah berjalan dan akses ke <i>reader</i> RFID sudah terkoneksi.
Aksi Aktor	Reaksi Sistem
<i>User</i> mengarahkan atau menunjukkan tag RFID ke <i>reader</i> RFID.	1. Tag RFID akan ter- <i>detect</i> oleh <i>reader</i> RFID. 2. Data hasil deteksi menjadi trigger untuk pemrosesan pengambilan gambar plat nomor lebih lanjut.
Kondisi Akhir	Data hasil deteksi ditampilkan pada <i>text field</i> program.

Sumber : Perancangan

2. Skenario *Use Case* Membawa Kendaraan untuk Dapat Diambil Gambarnya

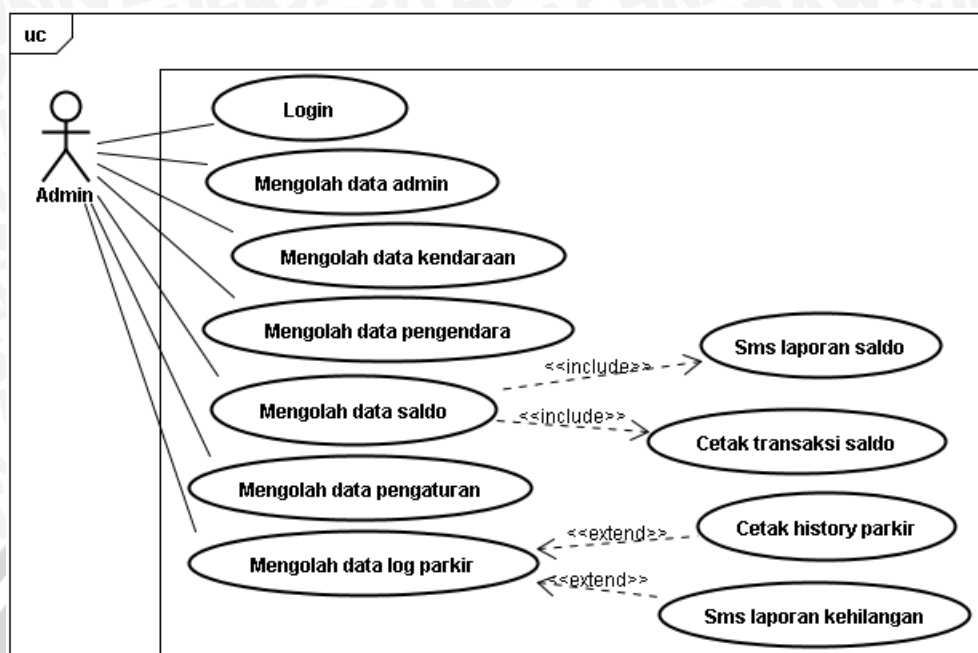
Tabel 4.6 Skenario *Use case* Membawa Kendaraan untuk Dapat Diambil Gambarnya

Skenario Kasus Pada Sistem	
Nomor <i>Use Case</i>	SRS_001_02
Nama	Membawa kendaraan untuk dapat diambil gambarnya
Tujuan	Untuk menunjukkan plat nomor kendaraan ke kamera.
Deskripsi	<i>Use Case</i> ini menjelaskan bagaimana <i>user</i> melakukan proses menunjukkan plat nomor kendaraan ke kamera.
Aktor	<i>User</i>
Skenario Utama	
Kondisi Awal	Perangkat lunak sudah berjalan, akses ke kamera sudah terkoneksi dan perangkat lunak telah berhasil mendeteksi RFID.
Aksi Aktor	Reaksi Sistem
<i>User</i> mengarahkan atau menunjukkan tag RFID ke <i>reader</i> RFID.	1. Plat nomor akan ter-<i>capture</i> oleh kamera. 2. Data hasil <i>capture</i> diolah menjadi data teks plat nomor kendaraan dan kemudian bersama dengan data hasil diteksi tag RFID disimpan ke dalam basis data.
Kondisi Akhir	Data hasil deteksi ditampilkan pada <i>label</i> dalam program.

Sumber : Perancangan

4.1.5.2 Analisis Use Case Subsistem Administrasi

Diagram *use case* ini melibatkan *admin* sebagai aktor dan satu buah *use case*. Satu buah *use case* ini termasuk dalam bagian Sistem Perangkat Lunak Untuk Simulasi Keluar Masuk dan Pembayaran Parkir Kendaraan Berbasis RFID. Satu buah *use case* ini juga akan disertai dengan skenario *use case* untuk menjelaskan rangkaian aktivitas yang terjadi di *use case* tersebut. Diagram *use case* untuk subsistem administrasi ditunjukkan pada Gambar 4.4 berikut ini :



Gambar 4.4 Diagram use case subsistem administrasi

Sumber : Perancangan

1. Skenario Use Case Login

Tabel 4.7 Skenario Use case Login

Skenario Kasus Pada Sistem	
Nomor Use Case	SRS_002_01
Nama	Login
Tujuan	Untuk menyeleksi <i>administrator</i> yang benar.
Deskripsi	<i>Use case</i> ini menjelaskan bagaimana <i>administrator</i> melakukan <i>login</i> untuk dapat menampilkan <i>form</i> utama dari <i>administrator</i> .
Aktor	<i>Administrator</i>
Skenario Utama	
Kondisi Awal	Sistem menampilkan <i>form</i> login.
Aksi Aktor	Reaksi Sistem
1. <i>Administrator</i> memasukkan data login (<i>username</i> dan <i>password</i>), lalu menekan tombol Login.	2. Sistem menerima data login dan kemudian sistem melakukan pengecekan terhadap data yang dimasukkan oleh <i>administrator</i> . Jika data benar, maka akan ditampilkan halaman utama <i>administrator</i> .
Skenario Alternatif 1 : Jika <i>username</i> atau <i>password</i> kosong atau salah	

	3. Sistem akan menampilkan pesan peringatan bahwa <i>login</i> gagal.
Kondisi Akhir	Sistem menampilkan <i>form</i> utama sisi <i>administrator</i> .

Sumber : Perancangan

2. Skenario Use Case Mengelola Data Admin

Tabel 4.8 Skenario Use case Mengelola Data Admin

Skenario Kasus Pada Sistem	
Nomor Use Case	SRS_002_02
Nama	Mengelola Data Admin
Tujuan	Untuk melakukan pengelolaan data admin.
Deskripsi	Use case ini menjelaskan bagaimana <i>administrator</i> dapat melakukan pengelolaan data admin (masukkan, perbarui atau hapus).
Aktor	<i>Administrator</i>
Skenario Utama	
Kondisi Awal	Admin telah berhasil login dan sistem menampilkan <i>form</i> administrasi data admin.
Aksi Aktor	Reaksi Sistem
1. <i>Administrator</i> memasukkan data admin (<i>username</i> admin, <i>password</i> , dan nama admin) lalu menekan tombol masukkan, atau memilih data dari daftar kemudian menekan tombol perbarui atau hapus.	2. Sistem menerima data admin yang akan ditambahkan atau yang akan diubah atau dihapus ke dalam basis data, kemudian sistem melakukan proses masukkan data, perbarui data atau hapus data terhadap data admin yang dimasukkan oleh <i>administrator</i> sesuai dengan tombol yang ditekan. 3. Sistem akan menampilkan pesan bahwa proses berhasil di masukkan ke dalam basis data.
Skenario Alternatif 1 : Jika <i>username</i> sudah ada	
	4. Sistem akan menampilkan pesan peringatan bahwa ada duplikat entry <i>username</i> . 5. Sistem akan menampilkan pesan bahwa proses gagal dimasukan ke database
Kondisi Akhir	Data admin yang berada di basis data berhasil ditambah, dirubah atau dihapus dan data pada daftar akan berubah sesuai dengan

	operasi yang dilakukan.
--	-------------------------

Sumber : Perancangan

3. Skenario *Use Case* Mengelola Data Kendaraan

Tabel 4.9 Skenario *Use case* Mengelola Data Kendaraan

Skenario Kasus Pada Sistem	
Nomor <i>Use Case</i>	SRS_002_03
Nama	Mengelola Data Kendaraan
Tujuan	Untuk melakukan pengelolaan data kendaraan.
Deskripsi	<i>Use case</i> ini menjelaskan bagaimana <i>administrator</i> dapat melakukan pengelolaan data kendaraan (masukkan, perbarui atau hapus).
Aktor	<i>Administrator</i>
Skenario Utama	
Kondisi Awal	Admin telah berhasil login dan sistem menampilkan <i>form</i> administrasi data kendaraan.
Aksi Aktor	Reaksi Sistem
1. <i>Administrator</i> memasukkan data kendaraan (plat nomor, merek, tipe, tahun dan nomor ktp) lalu menekan tombol masukkan, atau memilih data dari daftar kemudian menekan tombol perbarui atau hapus.	2. Sistem menerima data kendaraan yang akan ditambahkan atau yang akan diubah atau dihapus ke dalam basis data, kemudian sistem melakukan proses masukkan data, perbarui data atau hapus data terhadap data kendaraan yang dimasukkan oleh <i>administrator</i> sesuai dengan tombol yang ditekan. 3. Sistem akan menampilkan pesan bahwa proses berhasil di masukkan ke dalam basis data.
Skenario Alternatif 1 : Jika plat nomor sudah ada	
	4. Sistem akan menampilkan pesan peringatan bahwa ada duplikat entry plat nomor. 5. Sistem akan menampilkan pesan bahwa proses gagal dimasukan ke database
Kondisi Akhir	Data kendaraan yang berada di basis data berhasil ditambah, dirubah atau dihapus dan data pada daftar akan berubah sesuai dengan operasi yang dilakukan.

Sumber : Perancangan

4. Skenario *Use Case* Mengelola Data Pengendara

Tabel 4.10 Skenario *Use case* Mengelola Data Pengendara

Skenario Kasus Pada Sistem	
Nomor <i>Use Case</i>	SRS_002_04
Nama	Mengelola Data Pengendara
Tujuan	Untuk melakukan pengelolaan data pengendara.
Deskripsi	<i>Use case</i> ini menjelaskan bagaimana <i>administrator</i> dapat melakukan pengelolaan data pengendara (masukkan, perbarui atau hapus).
Aktor	<i>Administrator</i>
Skenario Utama	
Kondisi Awal	Admin telah berhasil login dan sistem menampilkan <i>form</i> administrasi data pengendara.
Aksi Aktor	Reaksi Sistem
1. <i>Administrator</i> memasukkan data pengendara (rfid, nomor ktp, nama, jenis kelamin, alamat dan nomor hp) lalu menekan tombol masukkan, atau memilih data dari daftar kemudian menekan tombol perbarui atau hapus.	2. Sistem menerima data pengendara yang akan ditambahkan atau yang akan diubah atau dihapus ke dalam basis data, kemudian sistem melakukan proses masukkan data, perbarui data atau hapus data terhadap data pengendara yang dimasukkan oleh <i>administrator</i> sesuai dengan tombol yang ditekan. 3. Sistem akan menampilkan pesan bahwa proses berhasil di masukkan ke dalam basis data.
Skenario Alternatif 1 : Jika rfid sudah ada	
	4. Sistem akan menampilkan pesan peringatan bahwa ada duplikat entry rfid. 5. Sistem akan menampilkan pesan bahwa proses gagal dimasukan ke database
Kondisi Akhir	Data pengendara yang berada di basis data berhasil ditambah, dirubah atau dihapus dan data pada daftar akan berubah sesuai dengan operasi yang dilakukan.

Sumber : Perancangan

5. Skenario Use Case Mengelola Data Saldo

Tabel 4.11 Skenario Use case Mengelola Data Saldo

Skenario Kasus Pada Sistem	
Nomor Use Case	SRS_002_05
Nama	Mengelola Data Saldo
Tujuan	Untuk melakukan pengelolaan data saldo.
Deskripsi	Use case ini menjelaskan bagaimana <i>administrator</i> dapat melakukan pengelolaan data saldo (tambah saldo, cetak saldo dan sms laporan saldo).
Aktor	<i>Administrator</i>
Skenario Utama	
Kondisi Awal	Admin telah berhasil login dan sistem menampilkan <i>form</i> administrasi data saldo.
Aksi Aktor	Reaksi Sistem
1. <i>Administrator</i> menambah data saldo pengendara lalu menekan tombol tambah.	2. Sistem menerima data saldo yang akan ditambahkan ke dalam basis data, kemudian sistem melakukan proses tambah saldo terhadap data saldo yang dimasukkan oleh <i>administrator</i> .
4. <i>Administrator</i> menekan tombol kirim sms ke semua atau memilih data dari daftar lalu menekan tombol kirim sms.	3. Sistem akan menampilkan pesan bahwa proses berhasil di masukkan ke dalam basis data.
6. <i>Administrator</i> menekan tombol ok atau no.	5. Sistem menerima data nomor hp tujuan pengiriman sms, lalu menampilkan format sms yang akan dikirim.
8. <i>Administrator</i> memilih data dari daftar lalu menekan tombol cetak log.	7. Bila <i>administrator</i> menekan tombol ok sistem mengirim sms laporan bulanan kepada nomor hp yang telah ditentukan dengan isi seperti yang sudah disetujui. Bila <i>administrator</i> menekan tombol no maka proses dibatalkan.
10. <i>Administrator</i> memilih ok, no atau cacle.	9. Sistem menerima data saldo yang akan dicetak, kemudian menampilkan pilihan untuk mencetak dari awal atau tidak.
	11. Bila <i>administrator</i> menekan tombol ok maka transaksi akan dicetak dari awal, bila no maka transaksi akan dicetak yang terakhir. Bila <i>administrator</i> menekan tombol cacle maka proses tidak dilanjutkan.
Skenario Alternatif 1 : Jika sms gagal dikirim	

	12. Sistem akan menampilkan pesan peringatan bahwa sms gagal dikirim.
Kondisi Akhir	Data saldo yang berada di basis data berhasil ditambah dan data pada daftar akan berubah.

Sumber : Perancangan

6. Skenario *Use Case* Mengelola Data Pengaturan

Tabel 4.12 Skenario *Use case* Mengelola Data Pengaturan

Skenario Kasus Pada Sistem	
Nomor <i>Use Case</i>	SRS_002_06
Nama	Mengelola Data Pengaturan
Tujuan	Untuk melakukan pengelolaan data pengaturan.
Deskripsi	<i>Use case</i> ini menjelaskan bagaimana <i>administrator</i> dapat melakukan pengolahan data pengaturan (perbarui).
Aktor	<i>Administrator</i>
Skenario Utama	
Kondisi Awal	Admin telah berhasil login dan sistem menampilkan <i>form</i> administrasi data pengaturan.
Aksi Aktor	Reaksi Sistem
1. <i>Administrator</i> memilih data dari daftar kemudian menekan tombol perbarui.	2. Sistem menerima data pengaturan yang akan diubah ke dalam basis data, kemudian sistem melakukan proses perbarui data terhadap data pengaturan yang dimasukkan oleh <i>administrator</i> . 3. Sistem akan menampilkan pesan bahwa proses berhasil di masukkan ke dalam basis data.
Kondisi Akhir	Data pengaturan yang berada di basis data berhasil dirubah dan data pada daftar akan berubah.

Sumber : Perancangan

7. Skenario *Use Case* Mengelola Data Log Parkir

Tabel 4.13 Skenario *Use case* Mengelola Data Log Parkir

Skenario Kasus Pada Sistem	
Nomor <i>Use Case</i>	SRS_002_07

Nama	Mengelola Data Log Parkir
Tujuan	Untuk melakukan pengelolaan data log parkir.
Deskripsi	<i>Use case</i> ini menjelaskan bagaimana <i>administrator</i> dapat melakukan pengelolaan data log parkir (search, cetak atau sms laporan kehilangan).
Aktor	<i>Administrator</i>
Skenario Utama	
Kondisi Awal	Admin telah berhasil login dan sistem menampilkan <i>form</i> administrasi data log parkir.
Aksi Aktor	Reaksi Sistem
1. <i>Administrator</i> menekan pilihan kriteria pencarian sesuai yang di perlukan kemudian menekan tombol filter.	2. Sistem menerima data criteria pencarian yang akan dicarikan ke dalam basis data, kemudian sistem melakukan proses <i>search history</i> parkir terhadap criteria pencarian yang dimasukkan oleh <i>administrator</i> .
4. <i>Administrator</i> menekan tombol <i>icon</i> cetak.	3. Sistem akan menampilkan <i>history</i> hasil pencarian dari basis data. 5. Sistem menerima data <i>histoy</i> yang akan dicetak, kemudian menampilkan <i>print preview</i> dari data yang akan dicetak.
6. <i>Administrator</i> memilih data dari daftar lalu menekan tombol laporkan.	7. Sistem menerima data plat nomor yang akan dilaporkan, kemudian menampilkan <i>format</i> sms yang akan dikirim.
8. <i>Administrator</i> menekan tombol ok atau no.	9. Bila <i>administrator</i> menekan tombol ok sistem mengirim sms laporan kehilangan kepada manager dan bagian keamanan dengan isi seperti yang sudah disetujui. Bila <i>administrator</i> menekan tombol no maka proses dibatalkan.
Kondisi Akhir	Data waktu print yang berada di basis data berhasil dirubah bila <i>administrator</i> mencetak menggunakan kriteria pencarian berdasarkan RFID.

Sumber : Perancangan

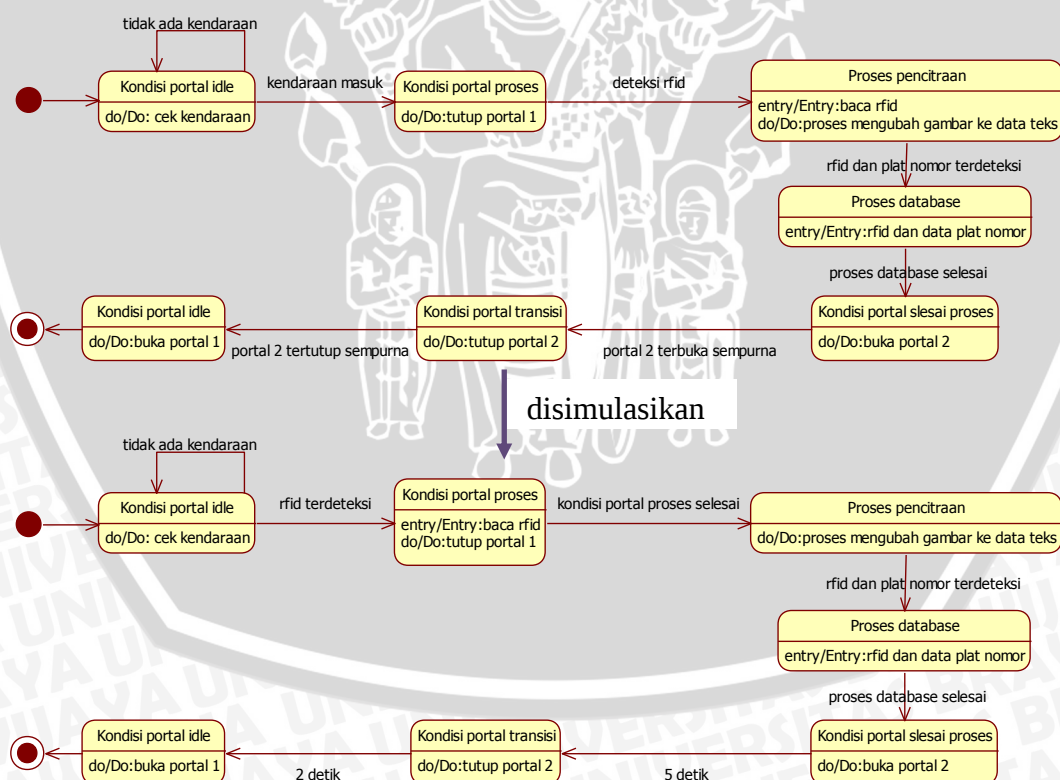
4.2 Perancangan Perangkat Lunak

Perancangan perangkat lunak dilakukan dalam lima tahap, yaitu pemodelan diagram *statechart*, perancangan basis data, pemodelan diagram *class* untuk menggambarkan perancangan struktur *class – class* yang menyusun sistem

perangkat lunak untuk keamanan dan pembayaran parkir di area apartemen berbasis RFID, pemodelan diagram *sequence* untuk menggambarkan interaksi antar objek atau *class* di dalam sistem, dan perancangan antarmuka pengguna dari sistem. Perancangan perangkat lunak pada skripsi ini menggunakan pendekatan desain berorientasi objek yang direpresentasikan dengan menggunakan UML (*Unified Modelling Language*).

4.2.1 Pemodelan Diagram Statechart

Diagram *statechart* adalah diagram yang menggambarkan kelakuan sistem secara keseluruhan. Konsep utama *statechart* adalah *event* dan *state*. *Event* merepresentasikan rangsangan eksternal yang memicu transisi (pesan, timeout, signal). Sedangkan diagram *state* dapat merepresentasikan siklus kehidupan secara kontinyu atau secara terbatas. Pemodelan diagram *statechart* subsistem gerbang parkir ditunjukkan pada Gambar 4.5.



Gambar 4.5 Diagram *statechart* subsistem gerbang parkir

Sumber : Perancangan

Aplikasi gerbang parkir dalam skripsi ini akan dibuat secara simulasi, tetapi penulis juga merancang bagaimana alur sistem dalam kondisi *real-time*-nya. Kondisi real dan simulasi Aplikasi gerbang parkir dapat dilihat seperti yang dapat dilihat pada Gambar 4.5 di atas. Saat kendaraan melewati sensor masuk yang letaknya dibawah portal 1, maka terjadi kondisi portal proses. Kondisi portal proses adalah kondisi di mana portal 1 dan portal 2 menutup. Setelah kondisi portal proses selanjutnya adalah proses pendeteksian tag RFID dimana pengendara mendekatkan tag RFID dengan *reader* RFID yang telah disediakan. RFID yang telah terbaca, menjadi pemicu proses pencitraan. Simulasi aplikasi tidak menggunakan sensor dan portal di ganti dengan *icon* yang berwarna merah yang mengartikan portal menutup dan hijau yang mengartikan portal terbuka. Sensor yang berfungsi untuk mendeteksi kendaraan yang memasuki gerbang dalam simulasi dihilangkan dan langsung dilakukan proses pendeteksian tag RFID yang menyebabkan kondisi portal proses, yaitu dimana *icon* 1 menjadi berwarna merah yang menyimbolkan portal 1 menutup dan *icon* 2 berwarna merah yang menyimbolkan portal 2 tetap menutup. Setelah kondisi portal proses selesai langsung dilanjutkan dengan proses pencitraan.

Proses pencitraan adalah proses dimana kamera mengambil gambar kendaraan tampak depan, kemudian gambar diproses untuk menghasilkan data teks plat nomor. Setelah proses pencitraan selesai dilakukan, dilanjutkan dengan proses database. Pada awalnya, nomor plat kendaraan dicek terlebih dahulu sudah memasuki area parkir atau tidak, jika sudah maka kendaraan dideteksi sebagai kendaraan keluar dan dilakukan pengurangan saldo, jika belum maka akan dicatat sebagai kendaraan masuk.

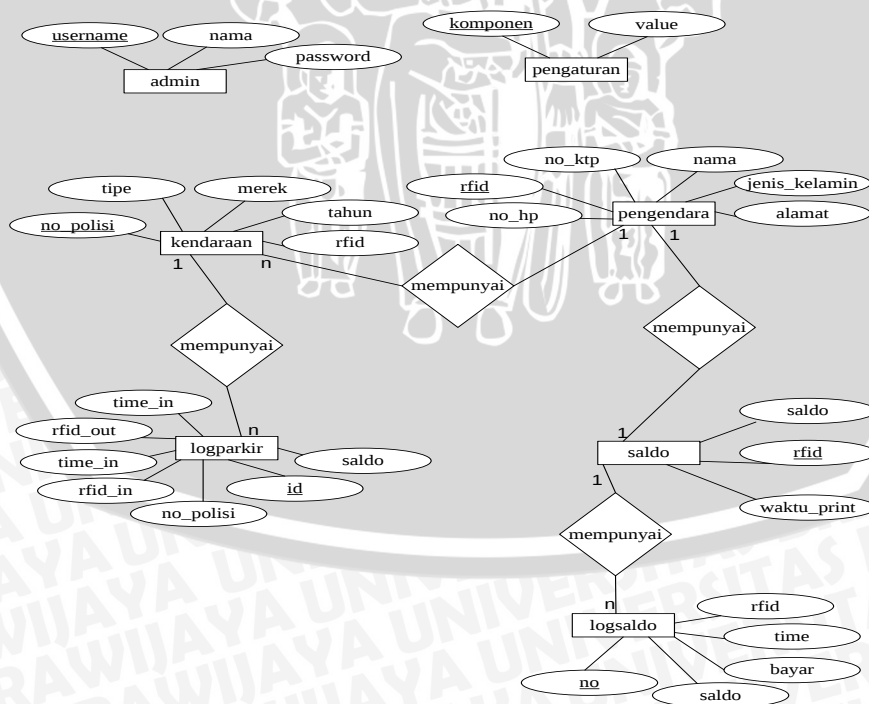
Setelah proses database selesai maka selanjutnya adalah kondisi portal selesai. Kondisi portal selesai adalah kondisi dimana portal 2 membuka atau dalam simulasi *icon* 2 berwarna hijau dan portal 1 tetap menutup atau dalam simulasi *icon* 1 tetap berwarna merah. Setelah itu kendaraan akan keluar dari gerbang melewati sensor 2 yang letaknya dibawah portal 2. Setelah sensor 2 mendeteksi adanya kendaraan yang telah melewatinya, maka dilanjutkan dengan kondisi portal transisi. Dalam simulasi, pendeteksian kendaraan keluar dari

gerbang digantikan dengan waktu tunggu selama 5 detik, kemudian dilanjutkan dengan kondisi portal transisi.

Kondisi portal transisi adalah kondisi dimana portal 2 kembali menutup atau dalam simulasi *icon* 2 kembali berwarna merah dan portal 1 tetap menutup atau dalam simulasi *icon* 1 tetap berwarna merah. Setelah portal 2 telah menutup sempurna yang dalam simulasi digantikan dengan proses tunggu selama 2 detik, dilanjutkan dengan kondisi portal idle. Kondisi portal idle adalah kondisi dimana portal 2 menutup atau dalam simulasi *icon* 2 berwarna merah dan portal 1 membuka atau dalam simulasi *icon* 1 menjadi berwarna hijau.

4.2.2 Perancangan Basis Data

Basis data berfungsi sebagai tempat menyimpan data. Perancangan basis data digunakan untuk merancang basis data yang akan dibuat agar masukan dan keluaran program sesuai dengan apa yang diharapkan. Perancangan basis data mengambil acuan dari proses analisis data yang dilakukan pada tahap analisis kebutuhan. Arsitektur basis data yang akan dirancang dijelaskan pada Gambar 4.6.



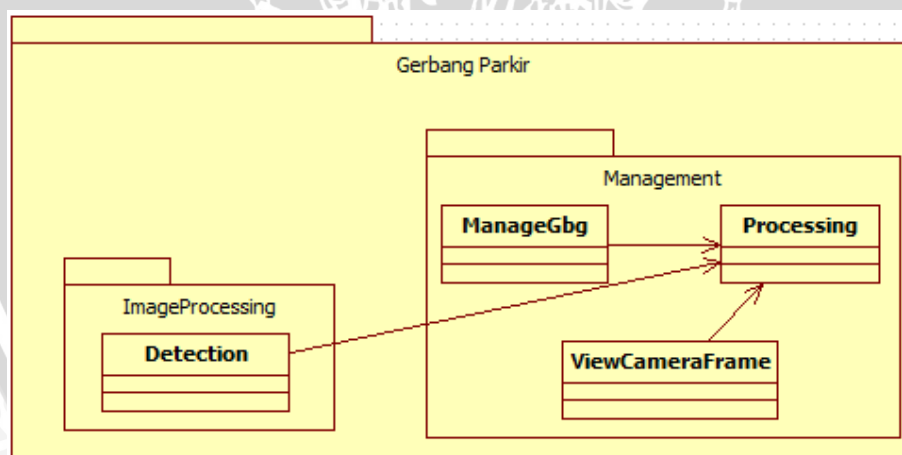
Gambar 4.6 Diagram entity relationship sistem

Sumber : Perancangan

Diagram *entity relationship* pada Gambar 4.6 dapat dibaca dengan admin dengan *primary key username* dan pengaturan dengan *primary key* komponen berdiri sendiri tanpa ada relasi dengan tabel yang lain. Tabel pengendara dengan *primary key* RFID mempunyai tabel kendaraan dengan *primary key* no_polisi dan mempunyai tabel saldo dengan *primary key* RFID. Sedangkan tabel saldo sendiri berelasi mempunyai dengan tabel logsaldo dengan *primary key* no. *History* dari proses parkir yang terjadi disimpan pada tabel logparkir dengan *primary key* id yang berelasi mempunyai dengan tabel kendaraan, pengendara dan saldo.

4.2.3 Diagram Class

Diagram *class* memberikan gambaran pemodelan elemen – elemen *class* yang membentuk sebuah perangkat lunak. *Class* bisa didapatkan dengan menganalisis secara detail terhadap *use case* yang dimodelkan. Gambar 4.7 menunjukkan diagram *class* dari subsistem gerbang parkir dan Gambar 4.8 menunjukkan diagram *class* dari subsistem administrasi yang akan dibuat.



Gambar 4.7 Diagram *class* subsistem gerbang parkir

Sumber : Perancangan

Pada diagram *class* di atas digambarkan beberapa *class* utama yang menyusun aplikasi gerbang parkir. Deskripsi dari beberapa *class* utama akan dijelaskan sebagai berikut:

1. *Class ManageGbg***Tabel 4.14** Deskripsi *class* ManageGbg

Nama Class : ManageGbg
Deskripsi : <i>Class</i> ini berfungsi untuk mengkoneksikan dengan database dan mengatur segala kegiatan yang berhubungan dengan database yang akan digunakan oleh <i>class</i> yang lainnya.

Sumber : Perancangan2. *Class Processing***Tabel 4.15** Deskripsi *class* Processing

Nama Class : Processing
Deskripsi : <i>Class</i> ini merupakan proses utama dalam aplikasi gerbang parkir. <i>Class</i> Preprocessing berfungsi untuk memproses pengaturan kerja gerbang parkir.

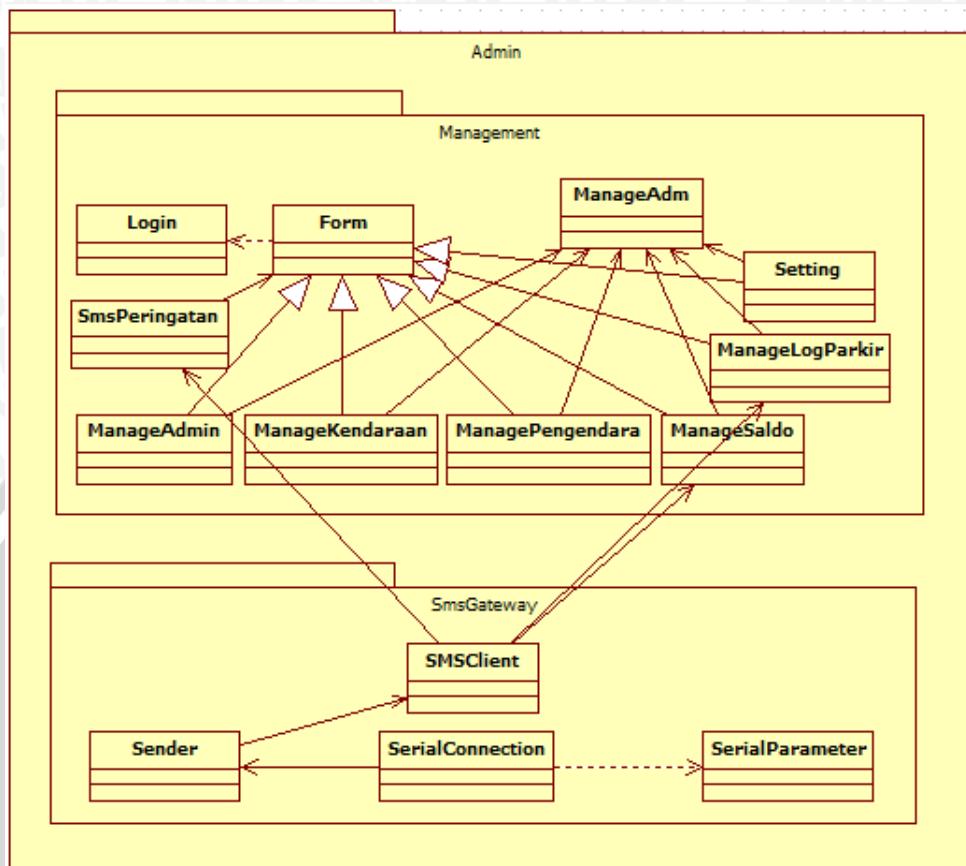
Sumber : Perancangan3. *Class ViewCameraFrame***Tabel 4.16** Deskripsi *class* ViewCameraFrame

Nama Class : ViewCameraFrame
Deskripsi : <i>Class</i> ini merupakan <i>class</i> untuk menampilkan <i>live camera</i> .

Sumber : Perancangan4. *Class Detection***Tabel 4.17** Deskripsi *class* Detection

Nama Class : Detection
Deskripsi : <i>Class</i> ini merupakan <i>class</i> yang berfungsi untuk memproses pengolahan gambar plat nomor menjadi data teks.

Sumber : Perancangan



Gambar 4.8 Diagram *class* subsistem administrasi

Sumber : Perancangan

Pada diagram *class* di atas digambarkan beberapa *class* utama yang menyusun aplikasi gerbang parkir. Deskripsi dari beberapa *class* utama akan dijelaskan sebagai berikut :

5. *Class* ManageAdm

Tabel 4.18 Deskripsi *class* ManageAdm

Nama Class : ManageAdm
Deskripsi : Class ini berfungsi untuk mengkoneksikan dengan database dan mengatur segala kegiatan yang berhubungan dengan database yang akan digunakan oleh <i>class</i> yang lainnya.

Sumber : Perancangan

6. *Class Login***Tabel 4.19** Deskripsi *class Login*

Nama Class : Login
Deskripsi : Class ini berfungsi untuk menyeleksi hanya admin yang terdaftar saja yang bisa menggunakan proses selanjutnya.

Sumber : Perancangan7. *Class Form***Tabel 4.20** Deskripsi *class Form*

Nama Class : Form
Deskripsi : Class ini berfungsi sebagai tempat utama untuk menampung akses ke menu-menu yang disediakan.

Sumber : Perancangan8. *Class ManageAdmin***Tabel 4.21** Deskripsi *class ManageAdmin*

Nama Class : ManageAdmin
Deskripsi : Class ini berfungsi sebagai tempat untuk mengelolah data admin.

Sumber : Perancangan9. *Class ManageKendaraan***Tabel 4.22** Deskripsi *class ManageKendaraan*

Nama Class : ManageKendaraan
Deskripsi : Class ini berfungsi sebagai tempat untuk mengelolah data kendaraan.

Sumber : Perancangan

10. *Class ManagePengendara***Tabel 4.23** Deskripsi *class* ManagePengendara

Nama Class : ManagePengendara
Deskripsi : Class ini berfungsi sebagai tempat untuk mengelolah data pengendara.

Sumber : Perancangan11. *Class ManageLogParkir***Tabel 4.24** Deskripsi *class* ManageLogParkir

Nama Class : ManageLogParkir
Deskripsi : Class ini berfungsi sebagai tempat untuk mengelolah data log parkir.

Sumber : Perancangan12. *Class ManageSaldo***Tabel 4.25** Deskripsi *class* ManageSaldo

Nama Class : ManageSaldo
Deskripsi : Class ini berfungsi sebagai tempat untuk mengelolah data saldo.

Sumber : Perancangan13. *Class Setting***Tabel 4.26** Deskripsi *class* Setting

Nama Class : Setting
Deskripsi : Class ini berfungsi sebagai tempat untuk mengelolah data setting.

Sumber : Perancangan

14. *Class SmsPeringatan***Tabel 4.27** Deskripsi *class SmsPeringatan*

Nama Class : SmsPeringatan
Deskripsi : Class ini berfungsi untuk mengirim sms ke semua pelanggan yang saldonya sama dengan atau dibawah Rp 0.

Sumber : Perancangan15. *Class Sender***Tabel 4.28** Deskripsi *class Sender*

Nama Class : Sender
Deskripsi : Class ini berfungsi untuk mengirim sms.

Sumber : Perancangan16. *Class SMSClient***Tabel 4.29** Deskripsi *class SMSClient*

Nama Class : SMSClient
Deskripsi : Class ini berfungsi sebagai proses pengiriman sms.

Sumber : Perancangan17. *Class SerialConnection***Tabel 4.30** Deskripsi *class SerialConnection*

Nama Class : SerialConnection
Deskripsi : Class ini berfungsi untuk mengatur pengaturan pengiriman sms.

Sumber : Perancangan

18. *Class SerialParameters***Tabel 4.31** Deskripsi *class SerialParameters*

Nama Class : SerialParameters
Deskripsi : <i>Class ini berfungsi untuk menyimpan parameter untuk port serial.</i>

Sumber : Perancangan

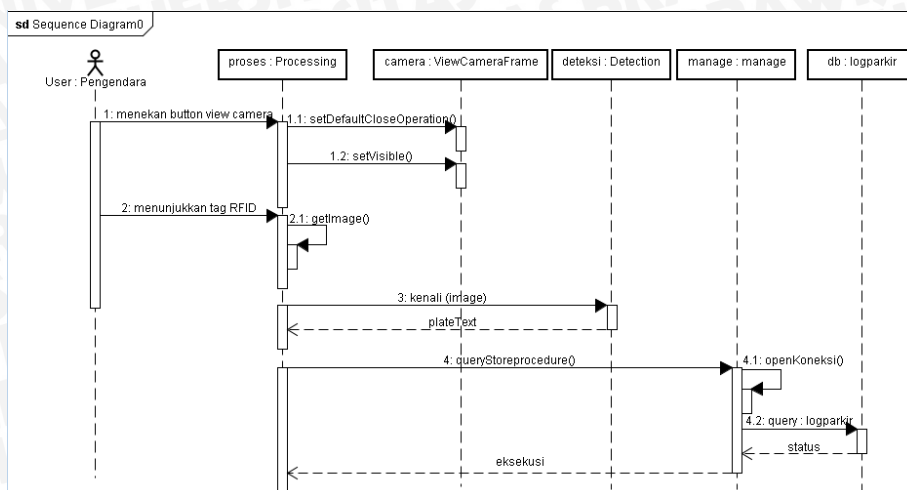
Deskripsi dari beberapa *class* yang terdapat dalam kelas diagram yang telah digambarkan diatas akan dijabarkan secara lengkap di lampiran 2.

4.2.4 Diagram Sequence

Diagram *sequence* menunjukkan pemodelan aliran jalannya proses interaksi antar objek atau *class* yang disusun berdasarkan urutan waktu. *Sequence diagram* digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-pemicu aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan. Diagram sekuensial (*sequence diagram*) disusun dengan mengambil acuan pada *use case* dan *class - class* yang membentuk fungsionalitas yang digambarkan pada *use case* tersebut.

4.2.4.1 Diagram Sequence Subsistem Gerbang Parkir

Gambar 4.9 merupakan diagram sekuensial Subsistem Gerbang Parkir yang dibuat dalam skripsi ini secara simulasi. Diagram sekuensial ini menggambarkan interaksi antar objek ketika pengendara memasuki atau keluar dari area parkir.



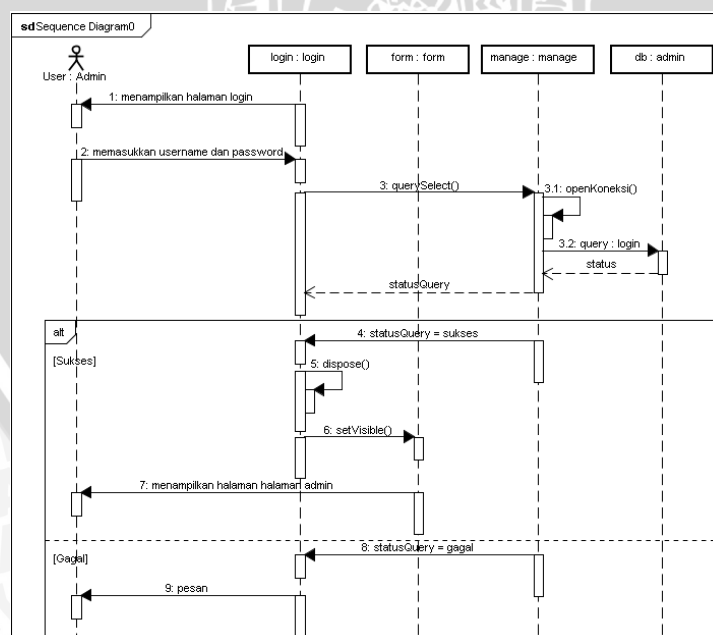
Gambar 4.9 Diagram *sequence* Subsistem Gerbang Parkir

Sumber : Perancangan

4.2.4.2 Diagram Sequence Aplikasi Administrator

1. Diagram *Sequence* Login

Gambar 4.10 merupakan diagram sekuensial Login. Diagram sekuensial ini menggambarkan interaksi ketika *administrator* melakukan proses login.



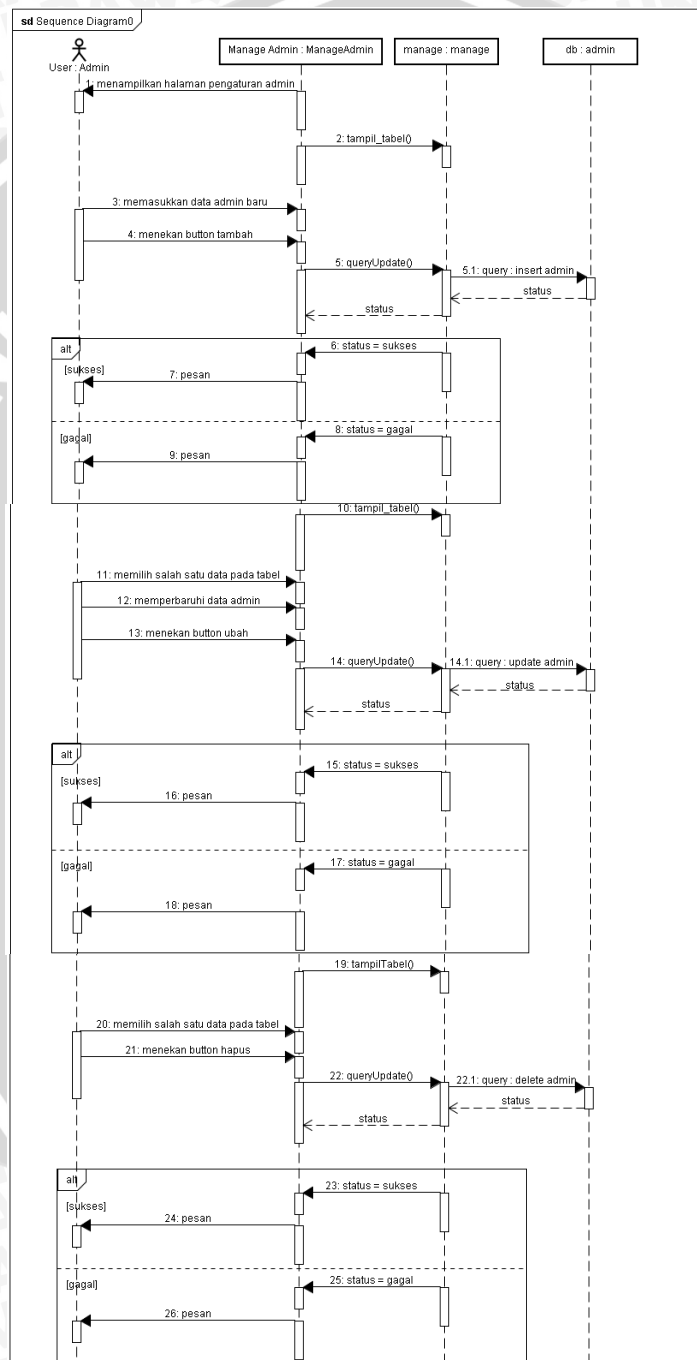
Gambar 4.10 Diagram *sequence* Login

Sumber : Perancangan

Sumber : Perancangan

3. Diagram *Sequence* Pengaturan Admin

Gambar 4.12 merupakan diagram sekuensial Pengaturan Admin. Diagram sekuensial ini menggambarkan interaksi ketika *administrator* melakukan pengolahan data admin.

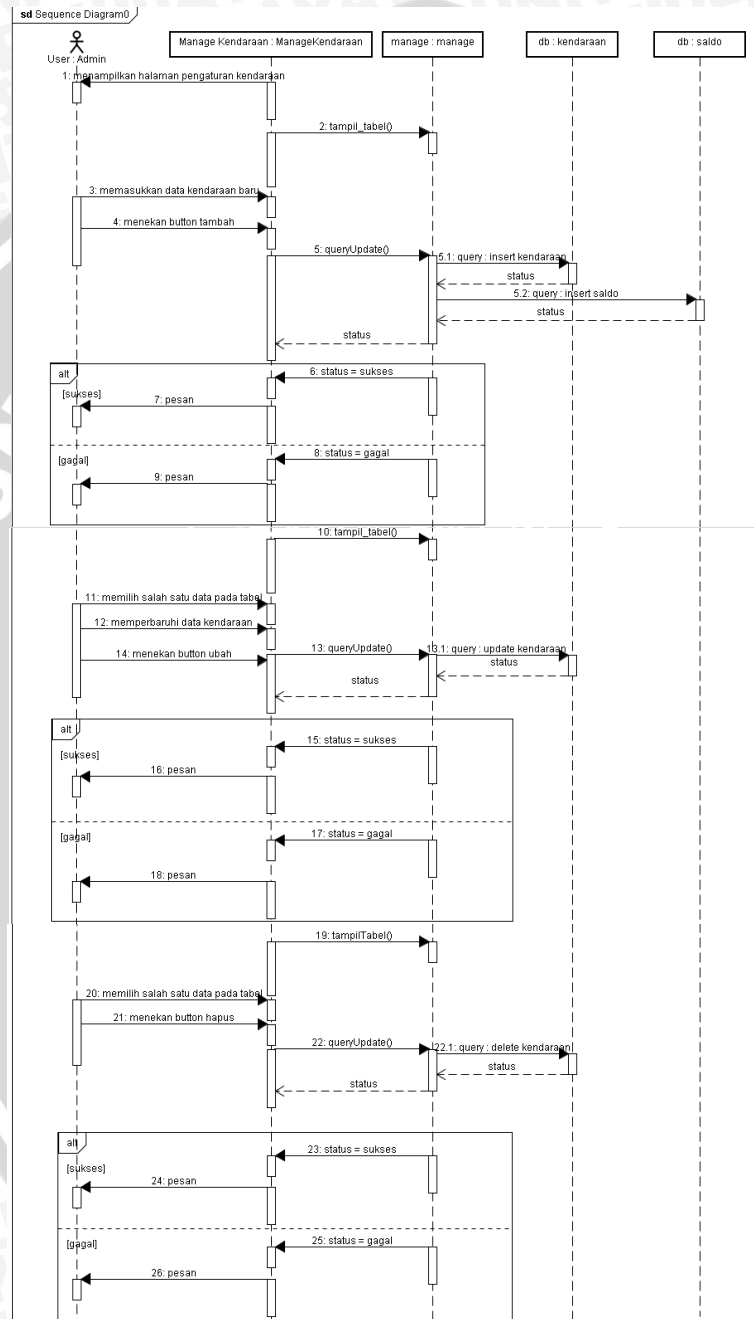


Gambar 4.12 Diagram *sequence* Pengaturan Admin

Sumber : Perancangan

4. Diagram *Sequence* Pengaturan Kendaraan

Gambar 4.13 merupakan diagram sekuensial Pengaturan Kendaraan. Diagram sekuensial ini menggambarkan interaksi ketika *administrator* melakukan pengolahan data kendaraan.

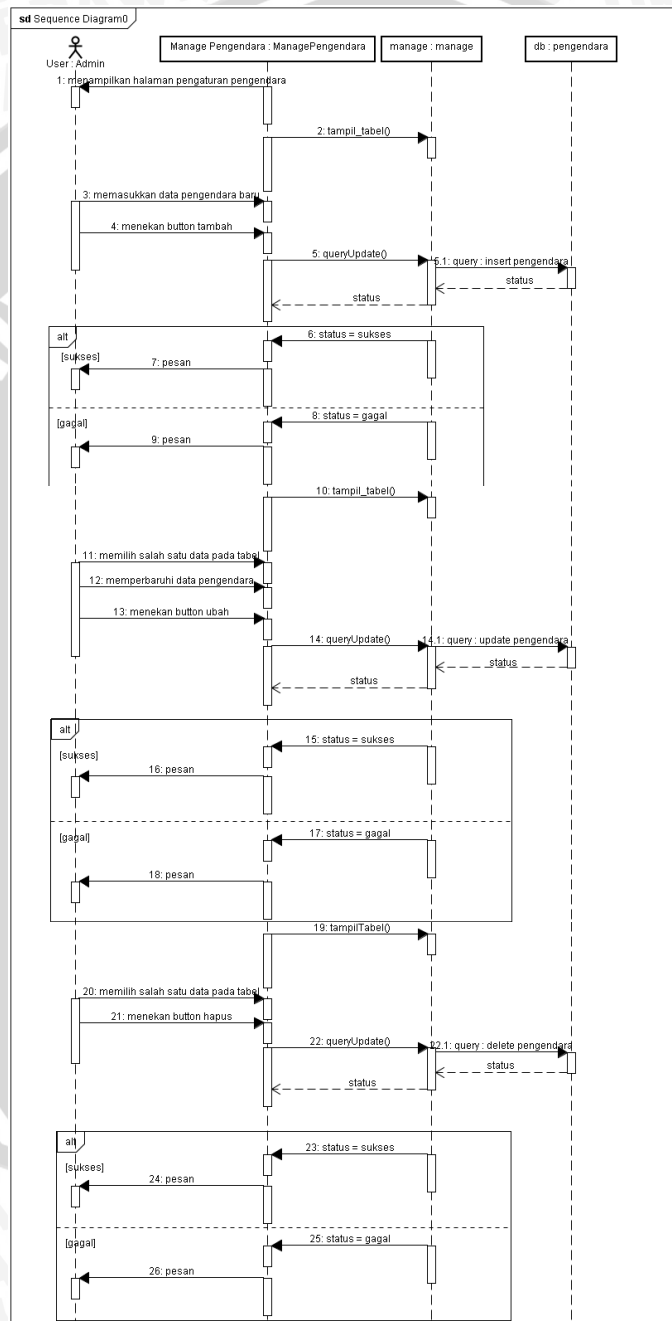


Gambar 4.13 Diagram *sequence* Pengaturan Kendaraan

Sumber : Perancangan

5. Diagram *Sequence* Pengaturan Pengendara

Gambar 4.14 merupakan diagram sekuensial Pengaturan Pengendara. Diagram sekuensial ini menggambarkan interaksi ketika *administrator* melakukan pengolahan data pengendara.

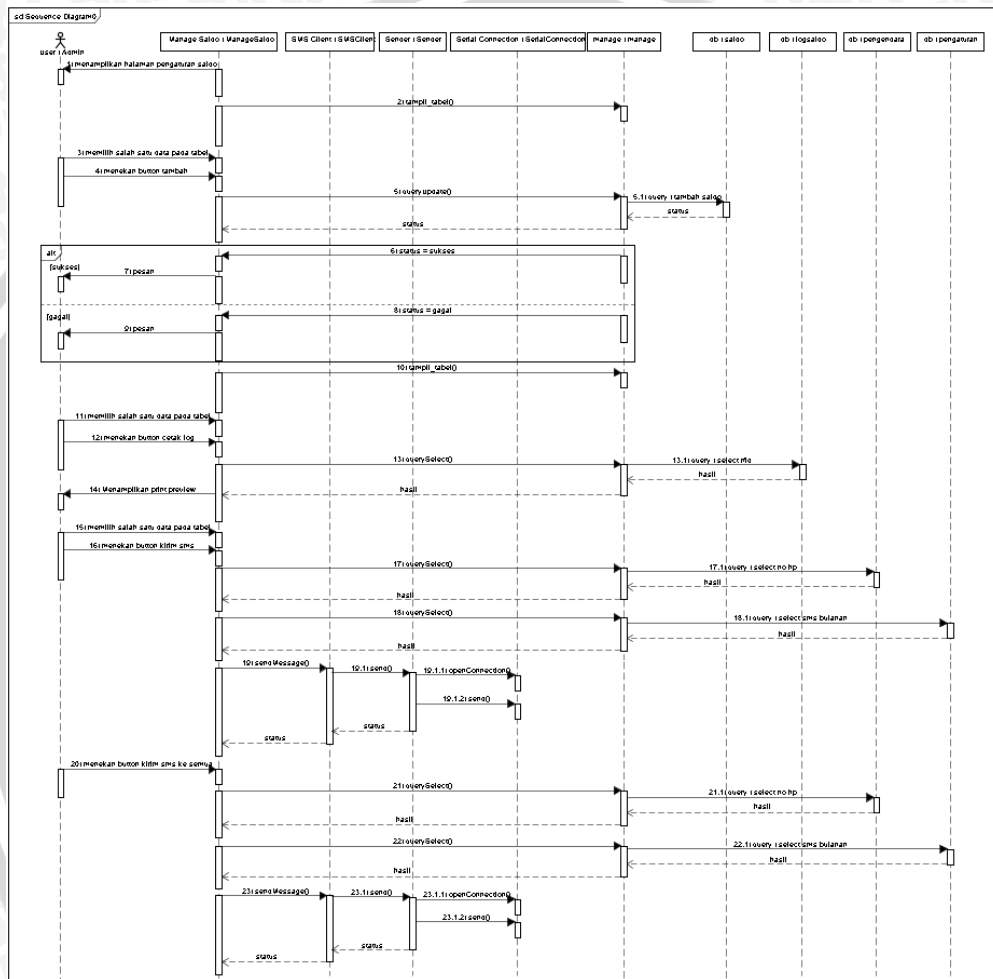


Gambar 4.14 Diagram *sequence* Pengaturan Pengendara

Sumber : Perancangan

6. Diagram Sequence Pengaturan Saldo

Gambar 4.15 merupakan diagram sekuensial Pengaturan Saldo. Diagram sekuensial ini menggambarkan interaksi ketika *administrator* melakukan pengolahan data saldo.

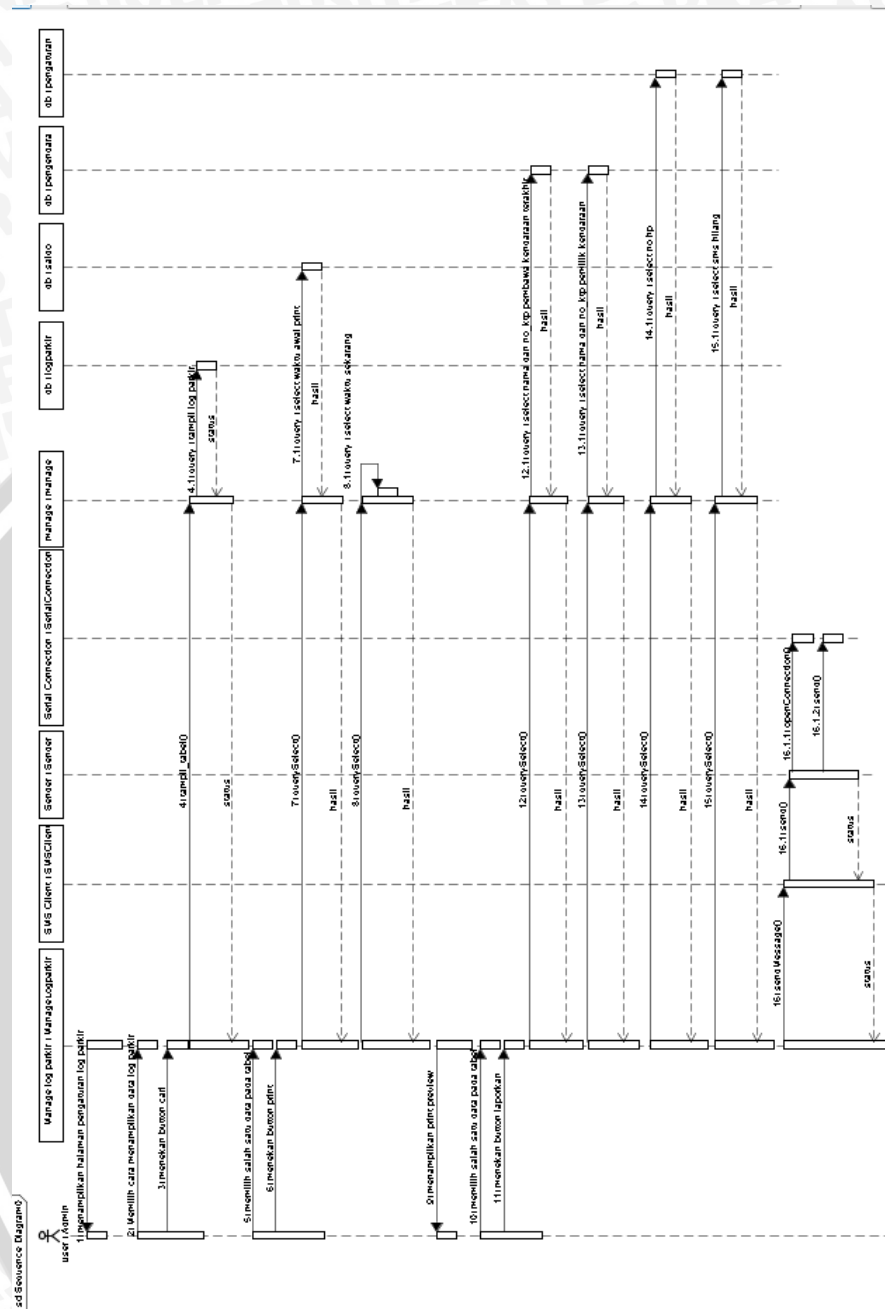


Gambar 4.15 Diagram sequence Pengaturan Saldo

Sumber : Perancangan

7. Diagram Sequence Pengaturan Log Parkir

Gambar 4.16 merupakan diagram sekuensial Pengaturan Log Parkir. Diagram sekuensial ini menggambarkan interaksi ketika *administrator* melakukan pengolahan data history parkir.

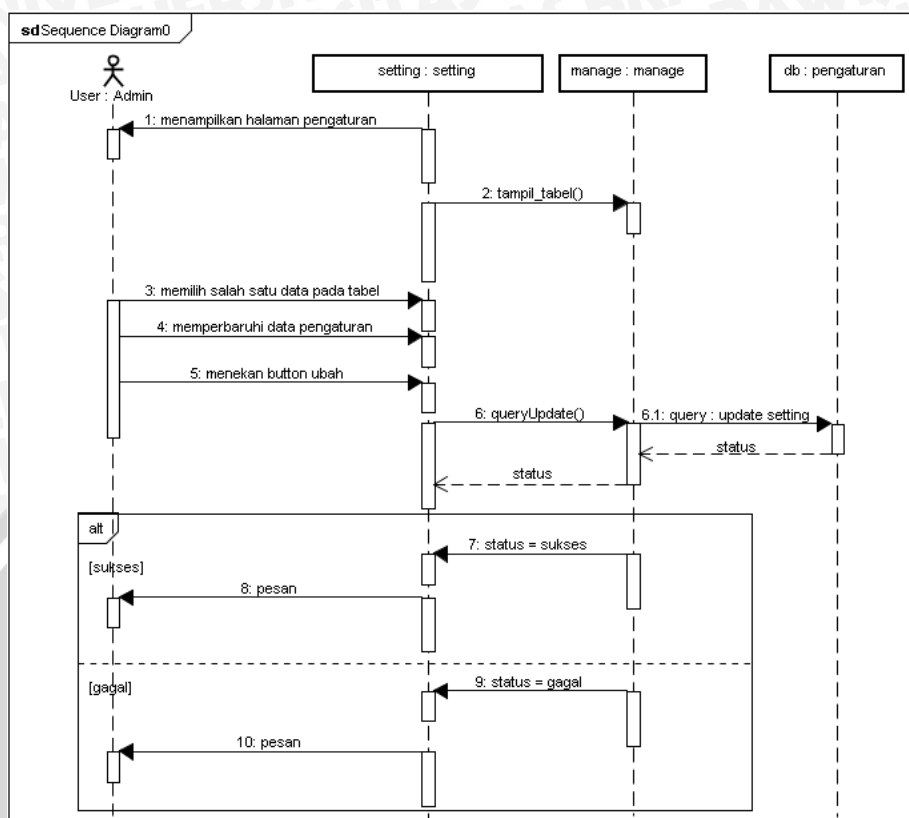


Gambar 4.16 Diagram *sequence* Pengaturan Log Parkir

Sumber : Perancangan

8. Diagram *Sequence* Pengaturan

Gambar 4.17 merupakan diagram sekuensial Pengaturan. Diagram sekuensial ini menggambarkan interaksi ketika *administrator* melakukan pengolahan data pengaturan.



Gambar 4.17 Diagram *sequence* Pengaturan

Sumber : Perancangan

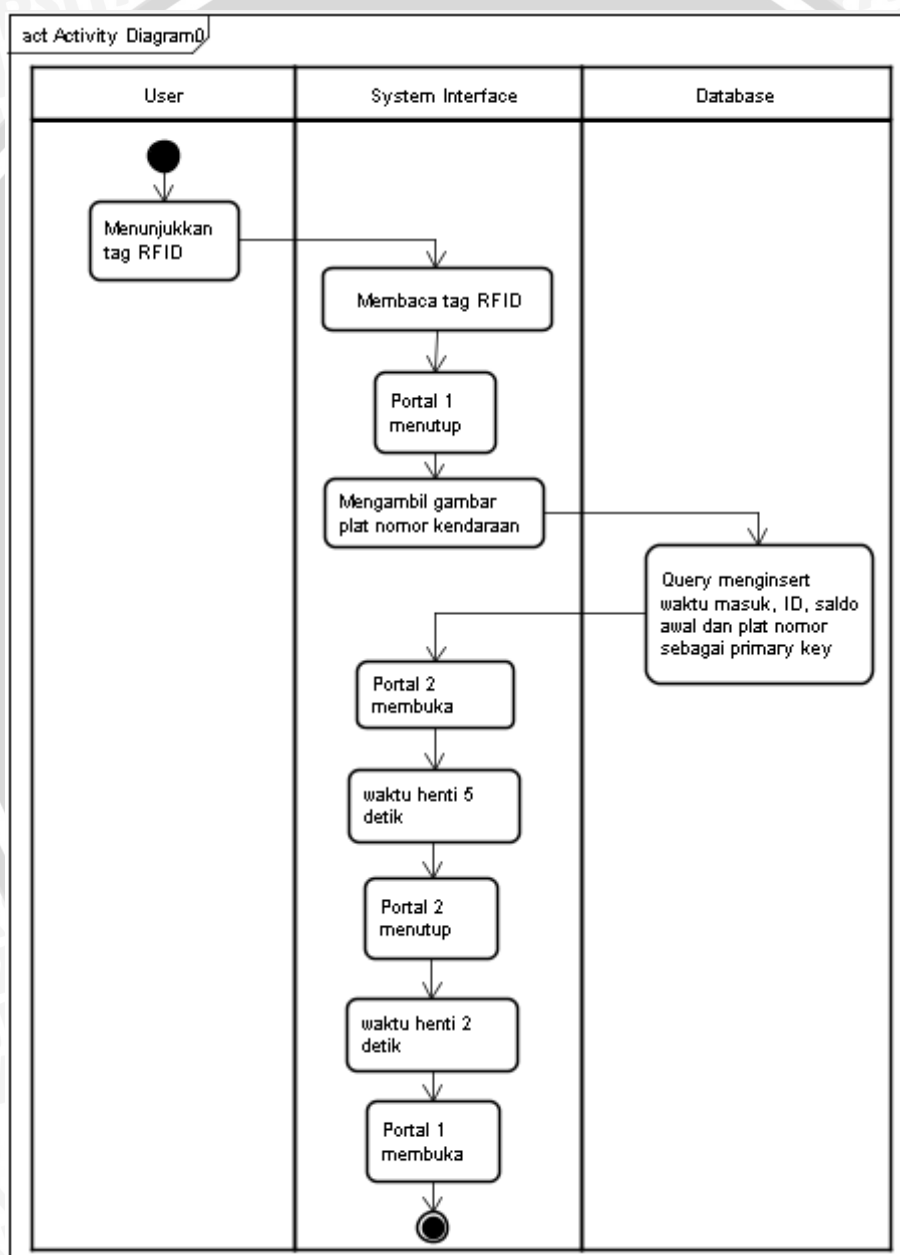
4.2.5 Diagram Aktivitas

Diagram aktivitas menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, keputusan yang mungkin terjadi, dan bagaimana mereka berakhir. Diagram aktivitas juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Diagram aktivitas merupakan state diagram khusus, di mana sebagian besar state adalah action dan sebagian besar transisi di-trigger oleh selesainya state sebelumnya (internal processing). Oleh karena itu diagram aktivitas tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. Menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses. Dipakai pada business modeling untuk memperlihatkan urutan aktivitas proses bisnis. Struktur diagram ini mirip flowchart atau Data Flow Diagram pada perancangan terstruktur.

4.2.5.1 Diagram Aktivitas Subsistem Gerbang Parkir

1. Diagram Aktivitas Proses Masuk

Gambar 4.18 merupakan diagram aktivitas proses masuk dalam simulasi. Diagram aktivitas ini menggambarkan aktivitas ketika user melakukan proses masuk ke area parkir.

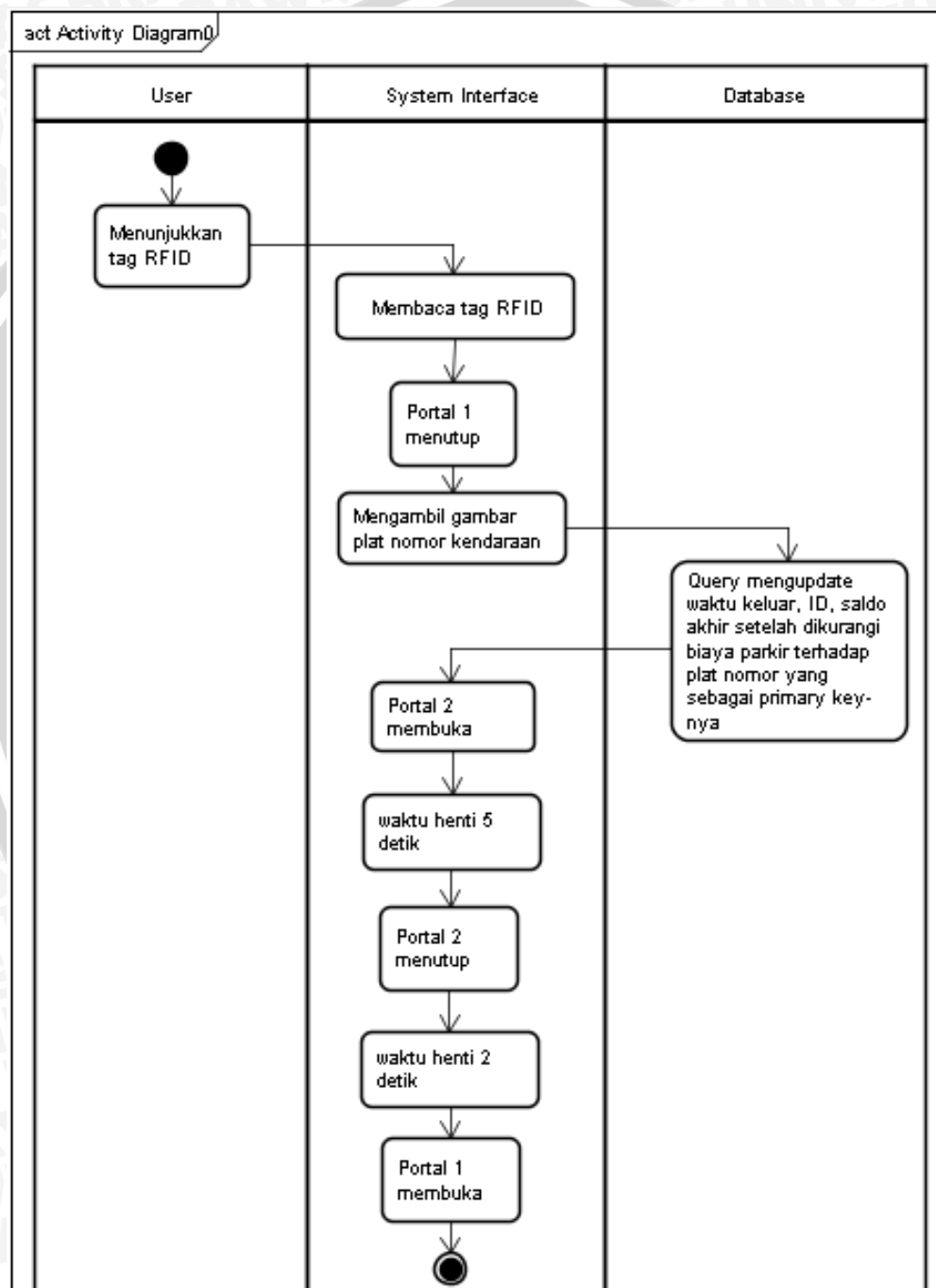


Gambar 4.18 Diagram aktivitas proses masuk

Sumber : Perancangan

2. Diagram Aktivitas Proses Keluar

Gambar 4.19 merupakan diagram aktivitas proses keluar dalam simulasi. Diagram aktivitas ini menggambarkan aktivitas ketika user melakukan proses keluar dari area parkir.



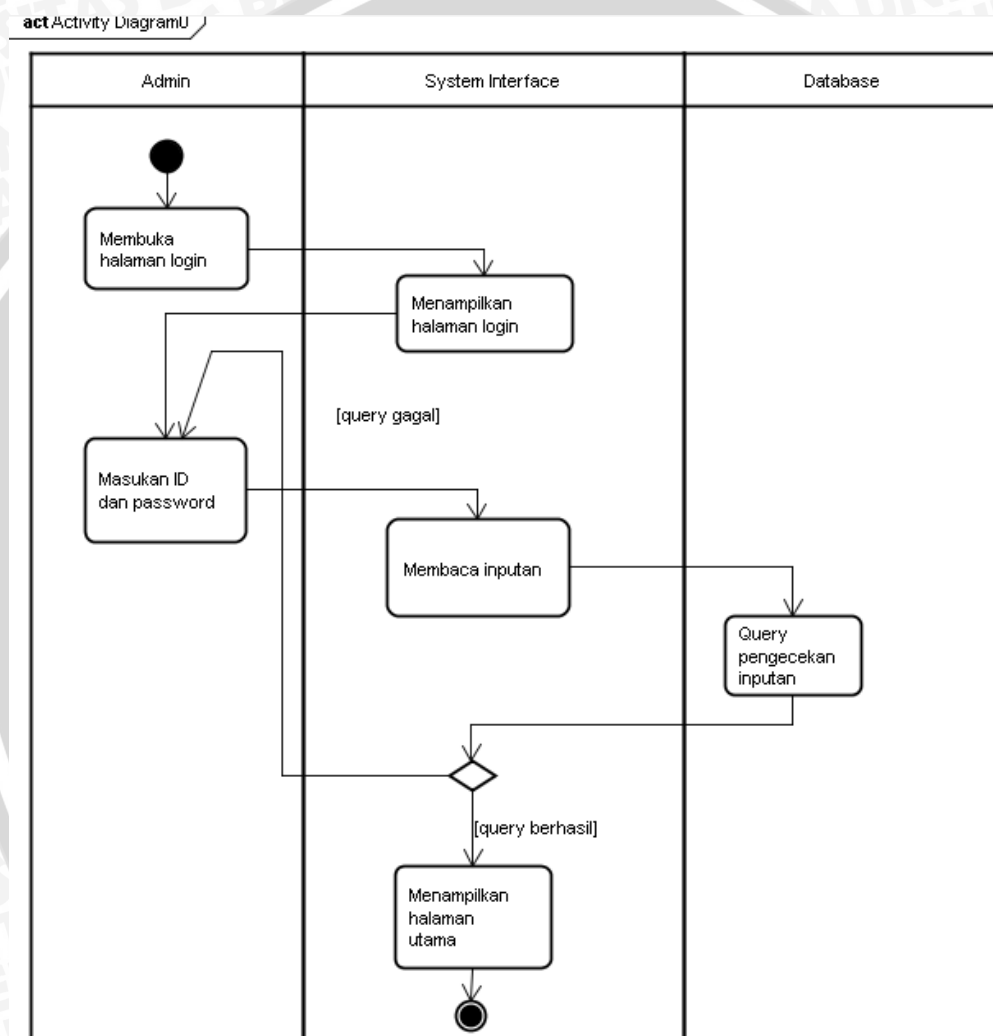
Gambar 4.19 Diagram aktivitas proses keluar

Sumber : Perancangan

4.2.5.2 Diagram Aktivitas Aplikasi Administrasi

1. Diagram Aktivitas Login

Gambar 4.20 merupakan diagram aktivitas Login. Diagram aktivitas ini menggambarkan alur ketika *administrator* melakukan proses login.

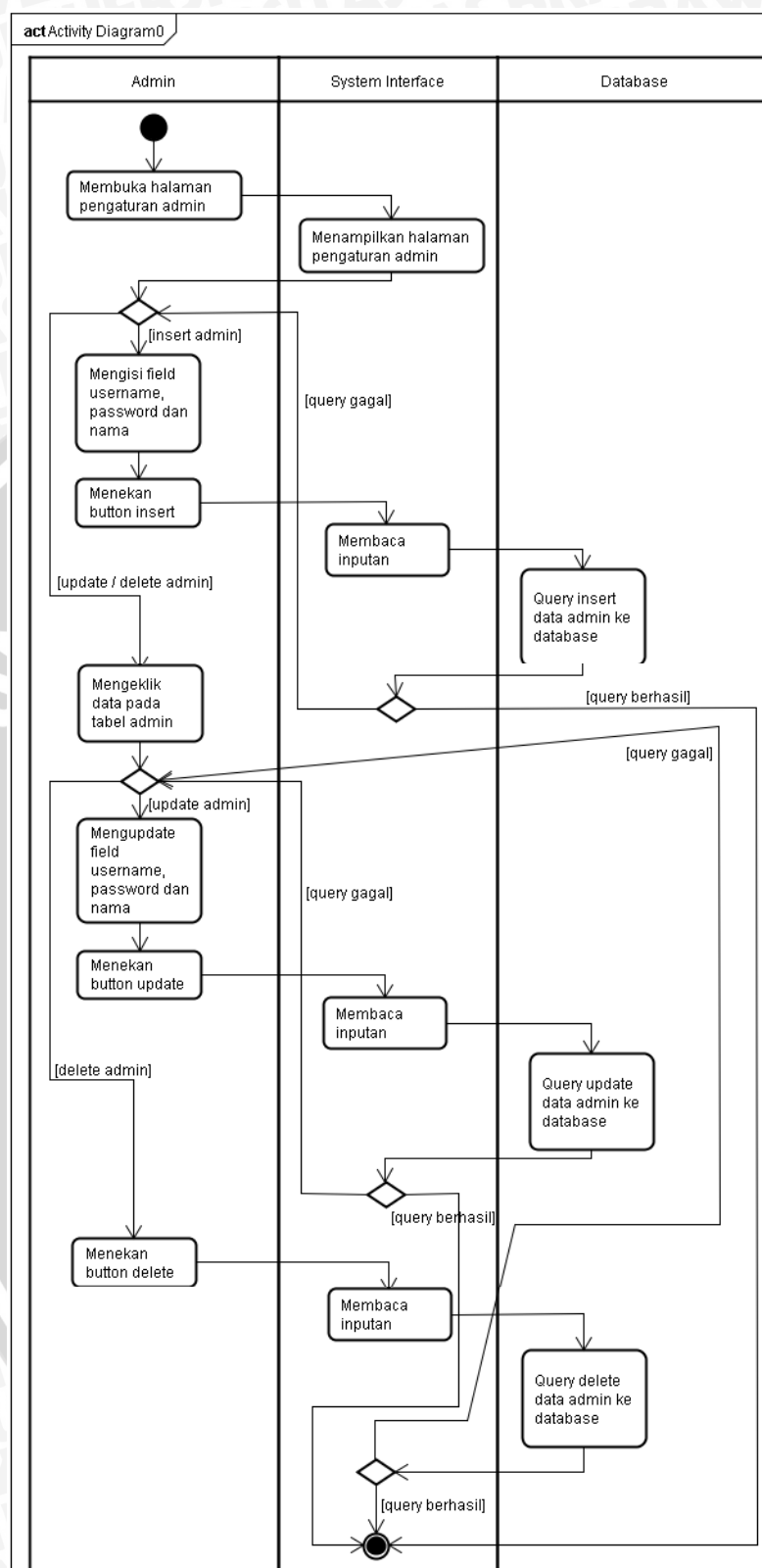


Gambar 4.20 Diagram aktivitas Login

Sumber : Perancangan

2. Diagram Aktivitas Pengaturan Admin

Gambar 4.21 merupakan diagram aktivitas pengaturan admin. Diagram aktivitas ini menggambarkan alur ketika *administrator* melakukan proses pengaturan admin.

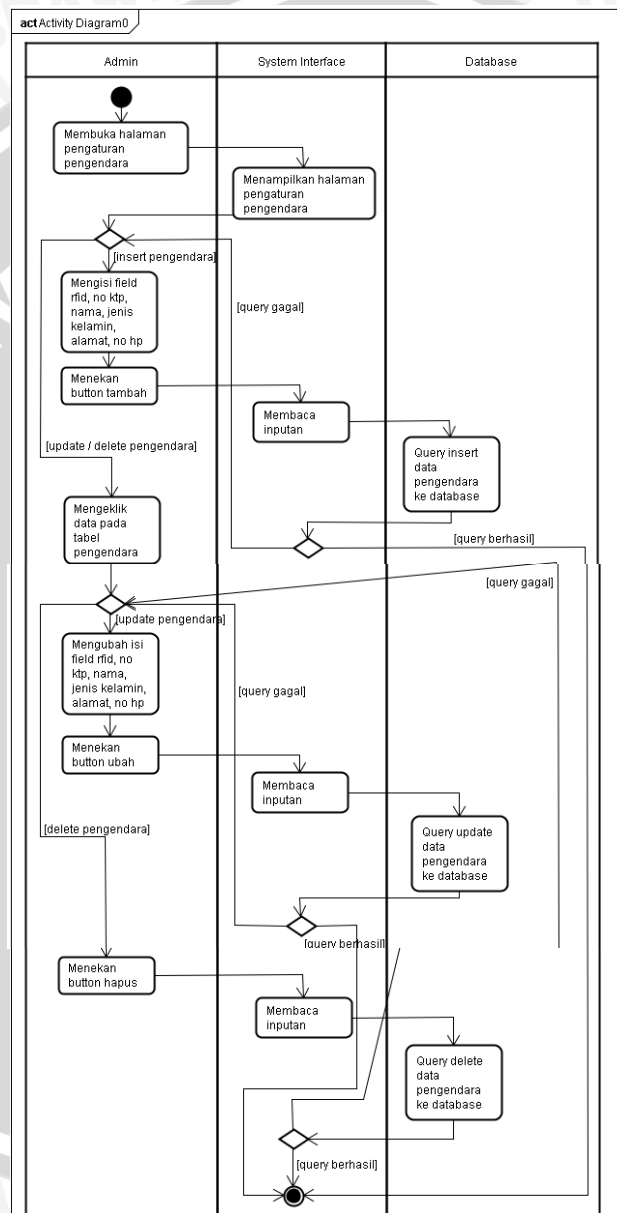


Gambar 4.21 Diagram aktivitas pengaturan admin

Sumber : Perancangan

3. Diagram Aktivitas Pengaturan Pengendara

Gambar 4.22 merupakan diagram aktivitas pengaturan pengendara. Diagram aktivitas ini menggambarkan alur ketika *administrator* melakukan proses pengaturan pengendara.

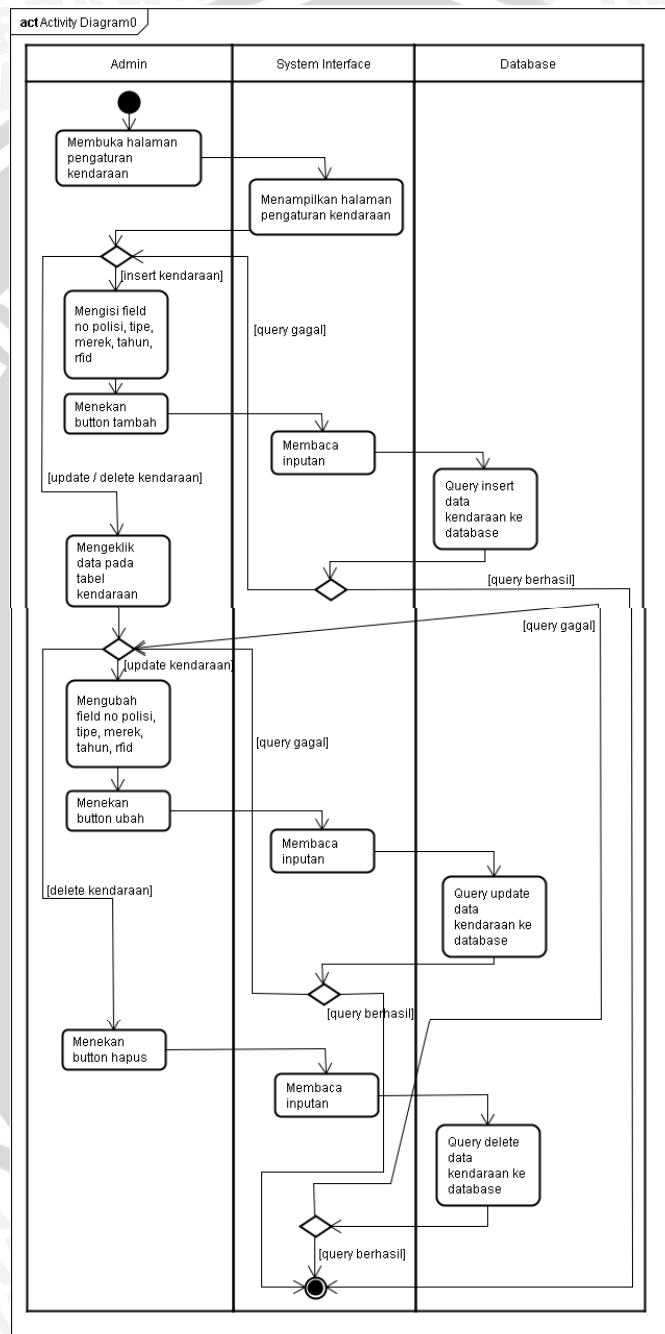


Gambar 4.22 Diagram aktivitas pengaturan pengendara

Sumber : Perancangan

4. Diagram Aktivitas Pengaturan Kendaraan

Gambar 4.23 merupakan diagram aktivitas pengaturan kendaraan. Diagram aktivitas ini menggambarkan alur ketika *administrator* melakukan proses pengaturan kendaraan.

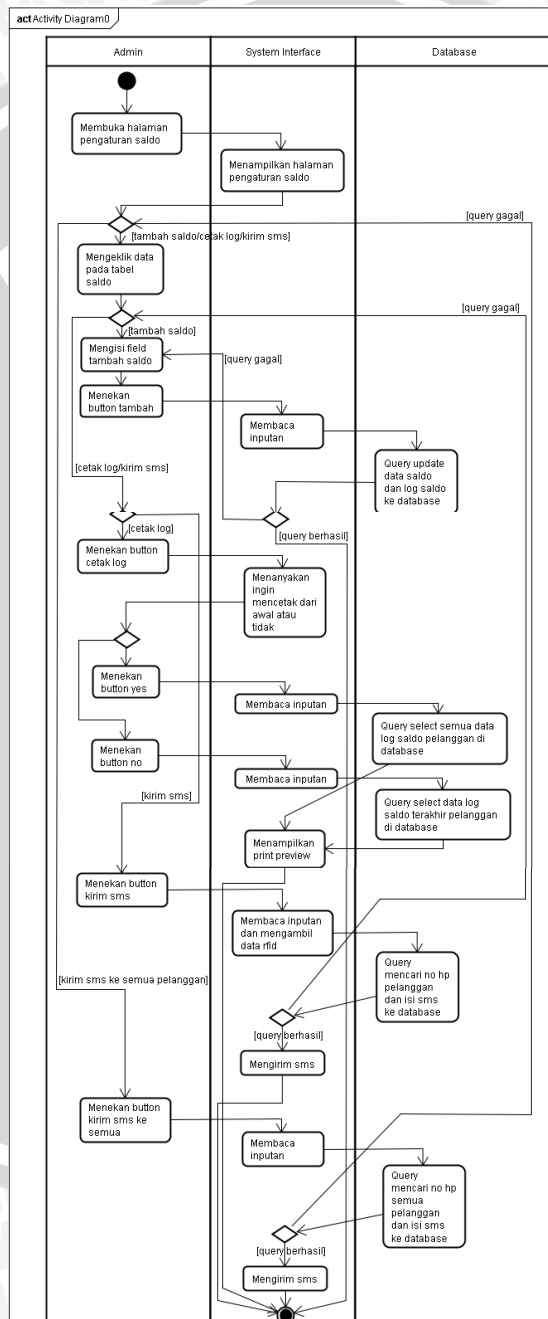


Gambar 4.23 Diagram aktivitas pengaturan kendaraan

Sumber : Perancangan

5. Diagram Aktivitas Pengaturan Saldo

Gambar 4.24 merupakan diagram aktivitas pengaturan saldo. Diagram aktivitas ini menggambarkan alur ketika *administrator* melakukan proses pengaturan saldo.

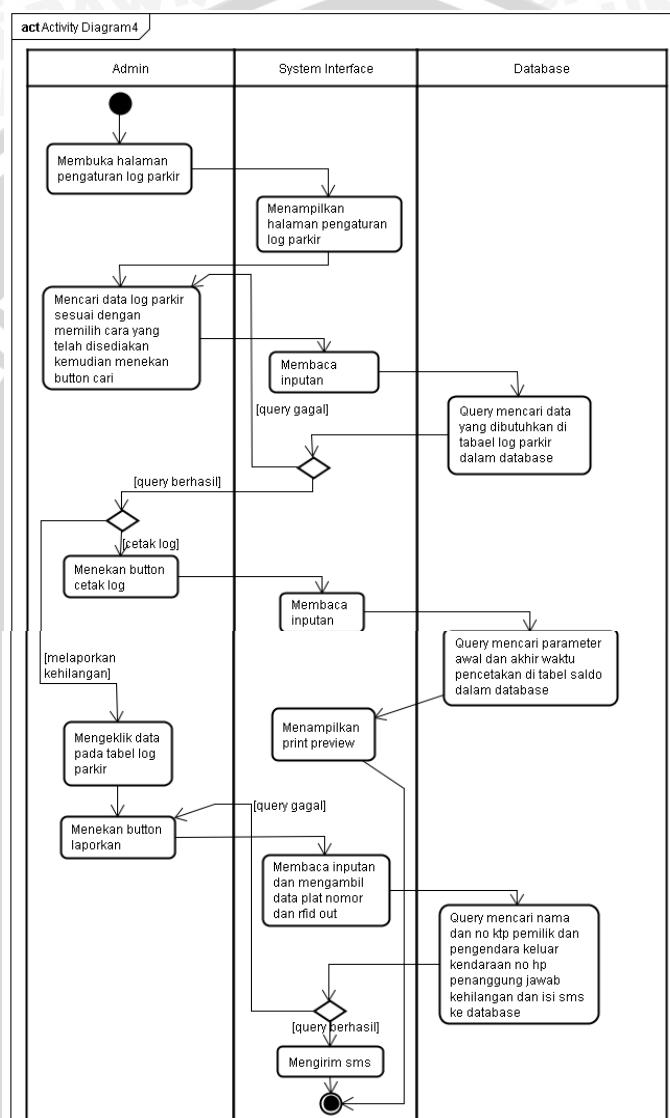


Gambar 4.24 Diagram aktivitas pengaturan saldo

Sumber : Perancangan

6. Diagram Aktivitas Pengaturan Log Parkir

Gambar 4.25 merupakan diagram aktivitas pengaturan log parkir. Diagram aktivitas ini menggambarkan alur ketika *administrator* melakukan proses pengaturan log parkir.

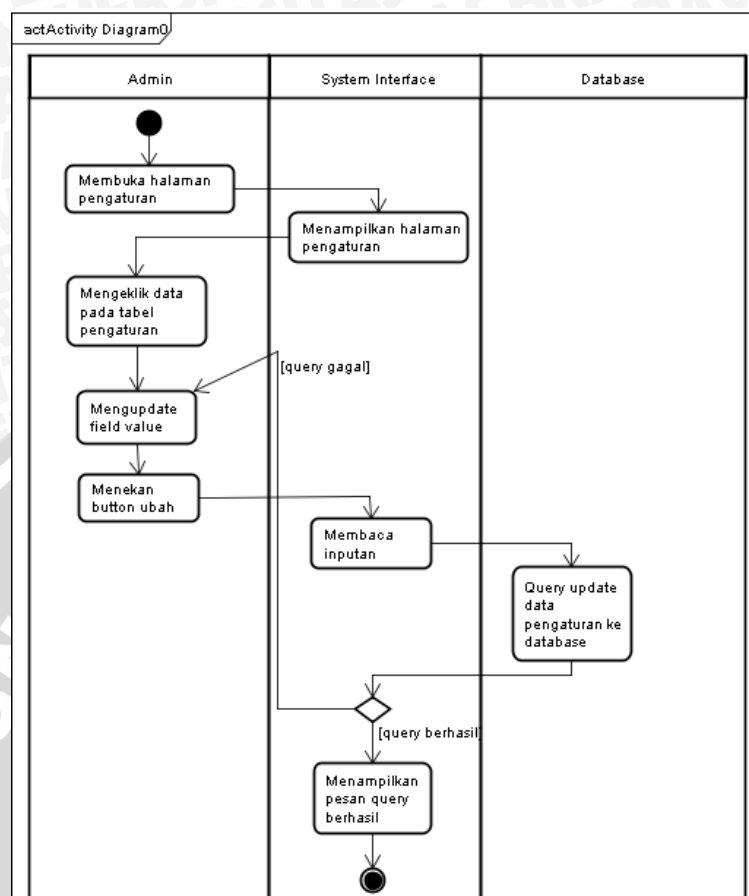


Gambar 4.25 Diagram aktivitas pengaturan log parkir

Sumber : Perancangan

7. Diagram Aktivitas Pengaturan

Gambar 4.26 merupakan diagram aktivitas pengaturan. Diagram aktivitas ini menggambarkan alur ketika *administrator* melakukan proses pengaturan.



Gambar 4.26 Diagram aktivitas pengaturan

Sumber : Perancangan

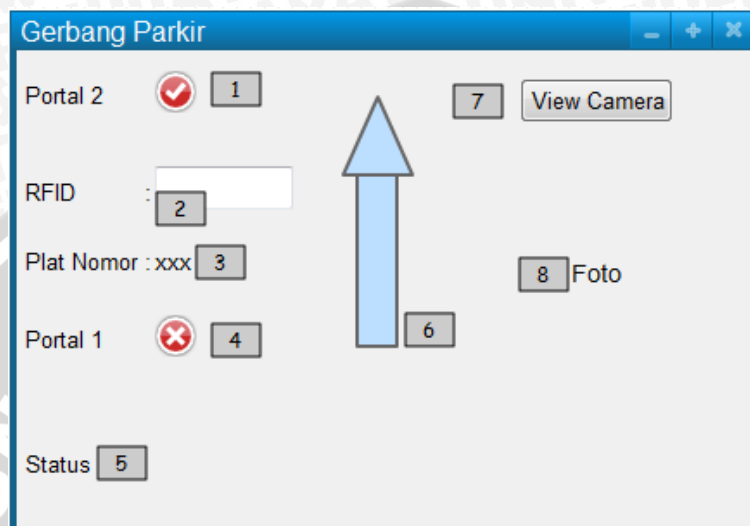
4.2.6 Perancangan Antarmuka

Pada bagian ini akan dijelaskan tentang perancangan antarmuka sistem perangkat lunak untuk keamanan dan pembayaran parkir di area apartemen berbasis RFID. Antarmuka perangkat lunak ini akan digunakan oleh pengguna untuk berinteraksi dengan sistem ini. Antarmuka perangkat lunak ini dibagi menjadi dua, yaitu antarmuka untuk subsistem gerbang parkir dan subsistem administrasi.

4.2.6.1 Perancangan Antarmuka Subsistem Gerbang Parkir

Antarmuka pengguna untuk subsistem gerbang parkir berupa sebuah halaman simulasi dari rancangan dari sistem perangkat lunak secara *real*. Antarmuka pengguna subsistem gerbang parkir berfungsi untuk mencatat keluar

masuknya kendaraan dari area parkir dan mengurangi saldonya, kemudian memasukkannya ke dalam database. Gambar 4.27 menggambarkan perancangan antarmuka untuk subsistem gerbang parkir.



Gambar 4.27 Tampilan antarmuka subsistem gerbang parkir

Sumber : Perancangan

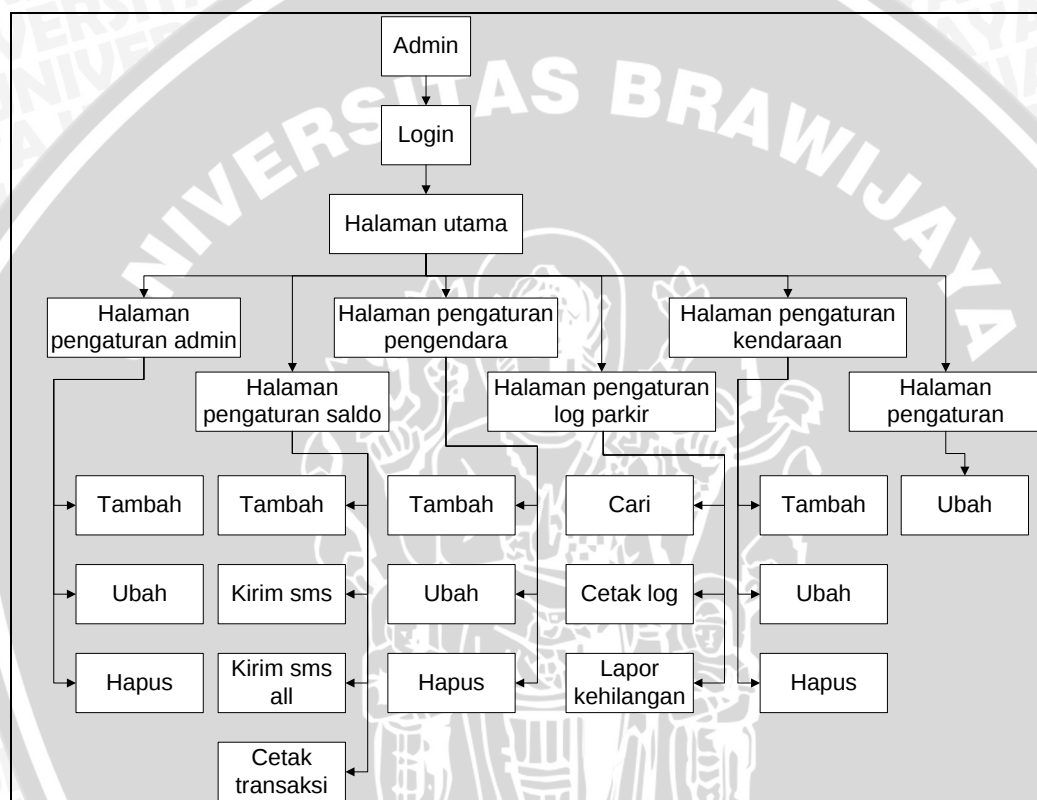
Gambar 4.27 memiliki keterangan sebagai berikut :

1. *Icon* sebagai pengganti portal 1
2. Field untuk mengisi RFID
3. Label untuk menampilkan plat nomor
4. *Icon* sebagai pengganti portal 2
5. *Label Status* untuk menampilkan pesan dari proses keluar masuk kendaraan
6. Gambar untuk menunjukkan arah masuk kendaraan
7. *Button* untuk melihat *live camera*
8. *Label* untuk menampilkan hasil foto yang telah diambil

4.2.6.2 Perancangan Antarmuka Subsistem Administrasi

Antarmuka pengguna untuk subsistem administrasi berupa halaman yang dapat digunakan untuk mengelola data yang diperlukan oleh sistem perangkat lunak untuk simulasi keluar masuk dan pembayaran parkir kendaraan berbasis

RFID. Antarmuka pengguna untuk subsistem administrasi terdiri atas halaman login, halaman utama, halaman pengaturan admin, halaman pengaturan kendaraan, halaman pengaturan pengemudi, halaman pengaturan saldo, halaman pengaturan log parkir, dan halaman pengaturan. *Sitemap* subsistem administrasi dari sistem perangkat lunak untuk simulasi keluar masuk dan pembayaran parkir kendaraan berbasis RFID ditunjukkan oleh Gambar 4.28.

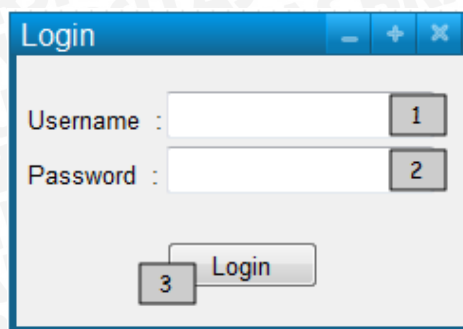


Gambar 4.28 Sitemap antarmuka pengguna subsistem administrasi

Sumber : Perancangan

a. Halaman Login

Halaman login merupakan salah satu antarmuka pengguna untuk subsistem administrasi yang berfungsi sebagai akses masuk admin untuk menggunakan hak-haknya sebagai *administrator*. Gambar 4.29 akan menunjukkan perancangan tampilan antarmuka dari halaman login.



Gambar 4.29 Tampilan antarmuka halaman login

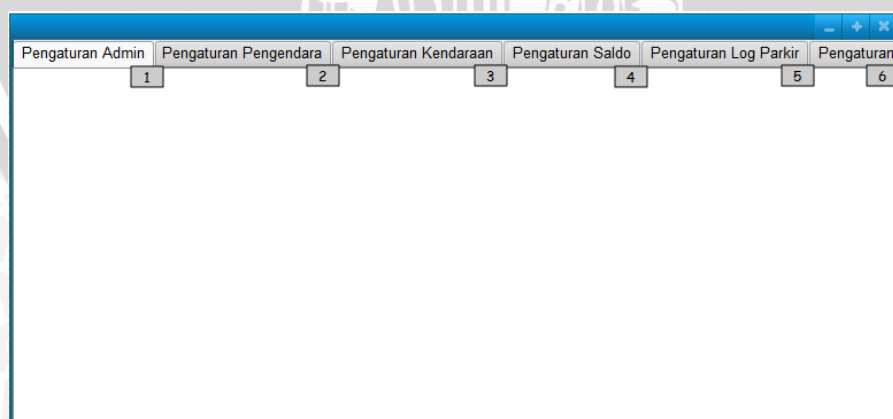
Sumber : Perancangan

Gambar 4.29 memiliki keterangan sebagai berikut :

1. *Field* untuk mengisi *username*.
2. *Field* untuk mengisi *password*.
3. Tombol Login untuk menjalankan proses login *administrator*.

b. Halaman Utama

Halaman utama merupakan salah satu antarmuka pengguna untuk subsistem administrasi yang berfungsi sebagai halaman yang menyediakan menu-menu yang diperlukan admin untuk menjalankan tugasnya. Gambar 4.30 akan menunjukkan perancangan tampilan antarmuka dari halaman utama.



Gambar 4.30 Tampilan antarmuka halaman utama

Sumber : Perancangan

Gambar 4.30 memiliki keterangan sebagai berikut :

1. Menu untuk melanjutkan ke halaman pengaturan admin.
2. Menu untuk melanjutkan ke halaman pengaturan pengendara.
3. Menu untuk melanjutkan ke halaman pengaturan kendaraan.
4. Menu untuk melanjutkan ke halaman pengaturan saldo.
5. Menu untuk melanjutkan ke halaman pengaturan log parkir.
6. Menu untuk melanjutkan ke halaman pengaturan.

c. Halaman Pengaturan Admin

Halaman pengaturan admin merupakan salah satu antarmuka pengguna untuk subsistem administrasi yang berfungsi untuk mengelolah data admin. Gambar 4.31 akan menunjukkan perancangan tampilan antarmuka dari halaman pengaturan admin.

Gambar 4.31 Tampilan antarmuka halaman pengaturan admin

Sumber : Perancangan

Gambar 4.31 memiliki keterangan sebagai berikut :

1. Tabel untuk menampilkan data admin yang ada dalam database.
2. *Field* untuk memasukkan username.
3. *Field* untuk memasukkan password.
4. *Field* untuk memasukkan nama.
5. *Button* untuk memasukkan data admin baru.
6. *Button* untuk meng-perbaruhi data admin.

7. *Button* untuk menghapus data admin.
8. *Label* status untuk menampilkan pesan dari pengolahan data admin.

d. Halaman Pengaturan Pengendara

Halaman pengaturan pengendara merupakan salah satu antarmuka pengguna untuk subsistem administrasi yang berfungsi untuk mengelolah data pengendara. Gambar 4.32 akan menunjukkan perancangan tampilan antarmuka dari halaman pengaturan pengendara.

Gambar 4.32 Tampilan antarmuka halaman pengaturan pengendara

Sumber : Perancangan

Gambar 4.32 memiliki keterangan sebagai berikut :

1. Tabel untuk menampilkan data pengendara yang ada dalam database.
2. *Field* untuk memasukkan RFID.
3. *Field* untuk memasukkan No KTP.
4. *Field* untuk memasukkan nama.
5. *Radio Button* untuk memilih jenis kelamin.
6. *Field* untuk memasukkan alamat.
7. *Field* untuk memasukkan no ponsel.
8. *Button* untuk memasukkan data pengendara baru.
9. *Button* untuk meng-perbaruhi data pengendara.

10. *Button* untuk menghapus data pengendara.

11. *Label* status untuk menampilkan pesan dari proses pengolahan data pengendara.

e. Halaman Pengaturan Kendaraan

Halaman pengaturan kendaraan merupakan salah satu antarmuka pengguna untuk subsistem administrasi yang berfungsi untuk mengelola data kendaraan. Gambar 4.33 akan menunjukkan perancangan tampilan antarmuka dari halaman pengaturan kendaraan.

The screenshot shows a software window titled 'Pengaturan Kendaraan'. It features a tabbed interface with the following tabs: 'Pengaturan Admin', 'Pengaturan Pengendara', 'Pengaturan Kendaraan' (which is the active tab), 'Pengaturan Saldo', 'Pengaturan Log Parkir', and 'Pengaturan'. The active tab displays a table with two columns labeled 'Column 1' and 'Column 2', containing three rows of placeholder text: 'Content 1', 'Content 2', 'Content 3' in the first column and 'Content 4' in the second. Below the table, there are five input fields with labels: 'No Polisi', 'Tipe', 'Merek', 'Tahun', and 'No KTP'. The 'No KTP' field is a dropdown menu showing 'Selection...'. Each input field has a small numbered box next to it, ranging from 2 to 6. At the bottom of the window, there are three buttons labeled 'Tambah', 'Ubah', and 'Hapus', each with a numbered box (7, 8, and 9 respectively). A 'Status' label with a numbered box (10) is located at the bottom left of the main content area.

Gambar 4.33 Tampilan antarmuka halaman pengaturan kendaraan

Sumber : Perancangan

Gambar 4.33 memiliki keterangan sebagai berikut :

1. Tabel untuk menampilkan data kendaraan yang ada dalam database.
2. *Field* untuk memasukkan no polisi kendaraan.
3. *Field* untuk memasukkan tipe kendaraan.
4. *Field* untuk memasukkan merek kendaraan.
5. *Field* untuk memasukkan tahun kendaraan.
6. *Combo box* untuk memasukkan No KTP.
7. *Button* untuk memasukkan data kendaraan baru.
8. *Button* untuk meng-perbaruhi data kendaraan.

9. *Button* untuk menghapus data kendaraan.
10. *Label* status untuk menampilkan pesan dari proses pengolahan data kendaraan.

f. Halaman Pengaturan Saldo

Halaman pengaturan saldo merupakan salah satu antarmuka pengguna untuk subsistem administrasi yang berfungsi untuk mengelolah data saldo. Gambar 4.34 akan menunjukkan perancangan tampilan antarmuka dari halaman pengaturan saldo.

Column 1	Column 2
Content 1	Content 2
Content 3	Content 4

RFID : xxxx

Saldo : yyyy

Tambah saldo :

Tambahkan Cetak Log Kirim SMS Kirim SMS ke semua Status

Gambar 4.34 Tampilan antarmuka halaman pengaturan saldo

Sumber : Perancangan

Gambar 4.34 memiliki keterangan sebagai berikut:

1. Tabel untuk menampilkan data saldo yang ada dalam database.
2. *Label* untuk menampilkan RFID pelanggan.
3. *Label* untuk memasukkan saldo pelanggan.
4. *Field* untuk memasukkan tambahan saldo.
5. *Button* untuk menambahkan data saldo pelanggan.
6. *Button* untuk mencetak transaksi yang telah dilakukan pelanggan.
7. *Button* untuk mengirim sms informasi saldo ke no telepon pelanggan yang telah dipilih.

8. *Button* untuk mengirim sms informasi saldo ke semua no telepon pelanggan.
9. *Label* status untuk menampilkan pesan dari proses pengolahan data saldo.

g. Halaman Pengaturan Log Parkir

Halaman pengaturan log parkir merupakan salah satu antarmuka pengguna untuk subsistem administrasi yang berfungsi untuk mengelolah data log parkir. Gambar 4.35 akan menunjukkan perancangan tampilan antarmuka dari halaman pengaturan log parkir.

Gambar 4.35 Tampilan antarmuka halaman pengaturan log parkir

Sumber : Perancangan

Gambar 4.35 memiliki keterangan sebagai berikut :

1. Tabel untuk menampilkan data log parkir yang ada dalam database.
2. *Radio Button* untuk memilih ingin melihat data log parkir berdasarkan keseluruhan atau waktu dan atau no polisi atau RFID.
3. *Check box* untuk memilih ingin melihat data log parkir berdasarkan waktu dan atau no polisi.
4. *Field* untuk memasukkan batas atas waktu log parkir yang dicari.
5. *Field* untuk memasukkan batas bawah waktu log parkir yang dicari.
6. *Field* untuk memasukkan no polisi kendaraan log parkir yang dicari.
7. *Field* untuk memasukkan RFID pengendara log parkir yang dicari.

8. *Label* untuk menampilkan no polisi kendaraan yang akan dilaporkan hilang.
9. *Button* untuk mengirim sms laporan kehilangan.
10. *Button* untuk menampilkan isi tabel sesuai dengan *range* yang diperlukan.
11. *Button* untuk mencetak log parkir yang di tampilkan dalam tabel.

h. Halaman Pengaturan

Halaman pengaturan merupakan salah satu antarmuka pengguna untuk subsistem administrasi yang berfungsi untuk mengelolah data pengaturan. Gambar 4.36 akan menunjukkan perancangan tampilan antarmuka dari halaman pengaturan.

Gambar 4.36 Tampilan antarmuka halaman pengaturan

Sumber : Perancangan

Gambar 4.36 memiliki keterangan sebagai berikut :

1. Tabel untuk menampilkan data pengaturan yang ada dalam database.
2. *Label* untuk menampilkan isi dari komponen yang telah dipilih.
3. *Field* untuk meng-perbaruhi isi value dari komponen yang telah dipilih.
4. *Button* untuk meng-perbaruhi data pengaturan.
5. *Label* status untuk menampilkan pesan dari proses pengolahan data saldo.

BAB V

IMPLEMENTASI

Bab ini membahas mengenai tahapan implementasi sistem perangkat lunak untuk simulasi keluar masuk dan pembayaran parkir kendaraan berbasis RFID berdasarkan hasil yang telah didapatkan dari analisis kebutuhan dan proses perancangan perangkat lunak. Implementasi terdiri atas penjelasan tentang spesifikasi sistem, batasan-batasan dalam implementasi, implementasi basis data, implementasi tiap *class* pada *file* program, implementasi algoritma, dan implementasi antarmuka.

5.2 Spesifikasi Lingkungan Sistem

Sistem perangkat lunak untuk simulasi keluar masuk dan pembayaran parkir kendaraan berbasis RFID dikembangkan dalam lingkungan implementasi yang terdiri dari perangkat keras dan perangkat lunak.

5.2.1 Spesifikasi Lingkungan Perangkat Keras

Spesifikasi lingkungan perangkat keras yang dipakai dalam proses pengembangan sistem perangkat lunak untuk simulasi keluar masuk dan pembayaran parkir kendaraan berbasis RFID dijelaskan pada Tabel 5.1.

Tabel 5.1 Spesifikasi lingkungan perangkat keras komputer

Notebook Sony Vaio	
Processor	Intel (R) Core (TM)2 Duo CPU P8400 @2.26GHz 2.27GHz
Memory (RAM)	4 GB
Harddisk	Seagate Momentus 5400.6 SATA 3Gb/s 500 GB
Motherboard	Vaio Notebook Intel Motherboard
Graphic Card	NVIDIA GeForce 9300M GS
Monitor	15.6" Widescreen LED Backlit Display
Webcam	Sony Easy Camera 1,3 MP

Sumber : Implementasi

5.2.2 Spesifikasi Lingkungan Perangkat Lunak

Spesifikasi lingkungan perangkat lunak yang dipakai dalam proses pengembangan sistem perangkat lunak untuk simulasi keluar masuk dan pembayaran parkir kendaraan berbasis RFID dijelaskan pada Tabel 5.2.

Tabel 5.2 Spesifikasi lingkungan perangkat lunak komputer

Notebook Sony Vaio	
<i>Operating System</i>	Microsoft Windows 7 Ultimate 64-bit
<i>Programming Language</i>	Java
<i>Software Development Kit</i>	Java Development Kit 7
<i>Programming Environment</i>	Java Runtime Environment 7
<i>Basis data Management System</i>	MySQL 5.1
<i>Integrated Development Environment</i>	Oracle NetBeans IDE 7.2

Sumber : Implementasi

5.3 Batasan – Batasan Implementasi

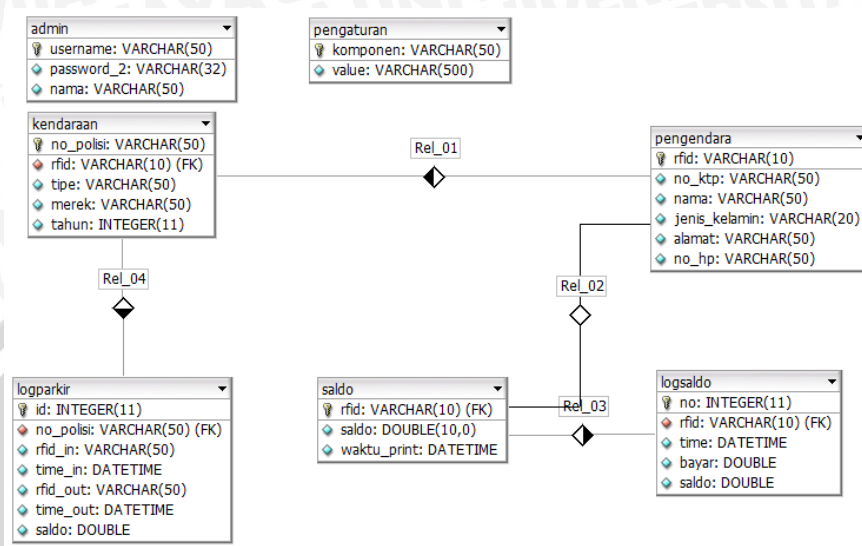
Beberapa batasan dalam mengimplementasikan sistem perangkat lunak untuk simulasi keluar masuk dan pembayaran parkir kendaraan berbasis RFID adalah sebagai berikut :

1. Sistem perangkat lunak untuk simulasi keluar masuk dan pembayaran parkir kendaraan berbasis RFID dirancang dan dijalankan dengan menggunakan *Java Desktop Application*.
2. Perangkat lunak ini dibangun dengan menggunakan teknologi RFID *tag* dan *reader*.
3. Aplikasi gerbang parkir dirancang untuk simulasi.
4. Basis data *Management System* yang digunakan adalah MySQL.

5.4 Implementasi Basis Data

Implementasi penyimpanan data dilakukan dengan basis data *management system* MySQL. Hasil implementasi penyimpanan data ini berupa *script – script* SQL. Hasil implementasi SQL pada basis data ini dimodelkan dalam diagram konseptual *entity relationship*. Gambar 5.1 menggambarkan diagram konseptual

entitiy relationship dari sistem perangkat lunak untuk simulasi keluar masuk dan pembayaran parkir kendaraan berbasis RFID.



Gambar 5.1 Diagram ER konseptual dari sistem

Sumber : Implementasi

Rancangan masing – masing tabel adalah sebagai berikut :

1. Tabel Admin

Nama tabel : admin

Jumlah *field* : 3

Fungsi : Untuk menyimpan data *administrator*

Tabel 5.3 Struktur tabel admin

No.	Nama Field	Tipe	Lebar	Keterangan
1	Username	Varchar	50	Username administrator
2	Nama	Varchar	50	Nama administrator
3	Password	Varchar	30	Password administrator

Sumber : Implementasi

2. Tabel Pengaturan

Nama tabel : pengaturan

Jumlah *field* : 2

Fungsi : Untuk menyimpan data pengaturan

Tabel 5.4 Struktur tabel pengaturan

No.	Nama Field	Tipe	Lebar	Keterangan
1	Komponen	Varchar	50	Komponen pengaturan
2	Value	Varchar	500	Isi dari komponen pengaturan

Sumber : Implementasi

3. Tabel Kendaraan

Nama tabel : kendaraan

Jumlah *field* : 5

Fungsi : Untuk menyimpan data kendaraan

Tabel 5.5 Struktur tabel kendaraan

No.	Nama Field	Tipe	Lebar	Keterangan
1	no_polisi	Varchar	50	No polisi kendaraan
2	Tipe	Varchar	50	Tipe kendaraan
3	Merek	Varchar	50	Merek kendaraan
4	Tahun	Integer	11	Tahun kendaraan
5	Rfid	Varchar	10	RFID pemilik kendaraan

Sumber : Implementasi

4. Tabel Pengendara

Nama tabel : pengendara

Jumlah *field* : 6

Fungsi : Untuk menyimpan data pengendara

Tabel 5.6 Struktur tabel pengendara

No.	Nama Field	Tipe	Lebar	Keterangan
1	Rfid	Varchar	10	RFID pengendara
2	no_ktp	Varchar	50	No KTP pengendara

3	Nama	Varchar	50	Nama pengendara
4	jenis_kelamin	Varchar	20	Jenis kelamin pengendara
5	Alamat	Varchar	50	Alamat pengendara
6	no_hp	Varchar	50	No HP pengendara

Sumber : Implementasi

5. Tabel Saldo

Nama tabel : saldo

Jumlah *field* : 3

Fungsi : Untuk menyimpan data saldo

Tabel 5.7 Struktur tabel saldo

No.	Nama Field	Tipe	Lebar	Keterangan
1	Rfid	Varchar	50	RFID pemilik kendaraan
2	Saldo	Double	10,0	Jumlah saldo pemilik kendaraan
3	waktu_print	Datetime	-	Waktu terakhir pemilik kendaraan mencetak history saldonya

Sumber : Implementasi

6. Tabel Log Saldo

Nama tabel : logsaldo

Jumlah *field* : 5

Fungsi : Untuk menyimpan data log saldo

Tabel 5.8 Struktur tabel logsaldo

No.	Nama Field	Tipe	Lebar	Keterangan
1	No	Integer	11	Nomor log saldo
2	Rfid	Varchar	50	RFID pemilik kendaraan
3	Time	Datetime	-	Waktu pengisian saldo
4	Bayar	Double	-	Jumlah pengisian saldo
5	Saldo	Double	-	Total saldo setelah ditambah pengisian saldo

Sumber : Implementasi

7. Tabel Log Parkir

Nama tabel : logparkir

Jumlah *field* : 7

Fungsi : Untuk menyimpan data log parkir

Tabel 5.9 Struktur tabel logparkir

No.	Nama <i>Field</i>	Tipe	Lebar	Keterangan
1	Id	Integer	11	Id log parkir
2	no_polisi	Varchar	50	Nomor polisi kendaraan yang parkir
3	rfid_in	Varchar	50	RFID pengendara yang membawa kendaraan masuk
4	time_in	Datetime	-	Waktu kendaraan masuk
5	rfid_out	Varchar	50	RFID pengendara yang membawa kendaraan keluar
6	time_out	Datetime	-	Waktu kendaraan keluar
7	Saldo	Double	-	Sisa saldo pemilik kendaraan

Sumber : Implementasi**5.5 Implementasi Class dan Interface Pada File Program**

Setiap *class* yang telah dirancang pada proses perancangan direalisasikan pada sebuah *file* program dengan ekstensi *.java. Tabel 5.3 menjelaskan mengenai pasangan antara *class* dengan *file* program yang digunakan untuk mengimplementasikannya.

Tabel 5.10 Implementasi *class* pada kode program *.java

No.	<i>Package</i>	Nama <i>Class</i>	Nama <i>File Program</i>
1	Admin.Management	ManageAdmin	ManageAdmin.java
2	Admin.Management	ManagePengendara	ManagePengendara.java
3	Admin.Management	ManageKendaraan	ManageKendaraan.java
4	Admin.Management	ManageSaldo	ManageSaldo.java
5	Admin.Management	ManageLogParkir	ManageLogParkir.java
6	Admin.Management	ManageLogSaldo	ManageLogSaldo.java
7	Admin.Management	Setting	Setting.java
8	Admin.Management	SmsPeringatan	SmsPeringatan.java

9	Admin.Management	Form	form.java
10	Admin.Management	Login	Login.java
11	Admin.Management	Manage	manage.java
12	Admin.SmsGateway	SMSCClient	SMSCClient.java
13	Admin.SmsGateway	Sender	Sender.java
14	Admin.SmsGateway	SerialConnection	SerialConnection.java
15	Admin.SmsGateway	SerialConnectionException	SerialConnectionException.java
16	Admin.SmsGateway	SerialParameters	SerialParameters.java
17	GerbangParkir.ImageProcessing	Detection	Detection.java
18	GerbangParkir.Management	Processing	Processing.java
19	GerbangParkir.Management	ViewCameraFrame	ViewCameraFrame.java
20	GerbangParkir.Management	Manage	manage.java

Sumber : Implementasi

5.6 Implemementasi Algoritma

Sistem perangkat lunak untuk simulasi keluar masuk dan pembayaran parkir kendaraan berbasis RFID memiliki beberapa proses atau *method* yang terdapat pada beberapa *class*. Beberapa *method* yang akan dicantumkan dalam penulisan makalah skripsi ini hanya untuk algoritma dari beberapa proses (operasi) saja sehingga tidak semua *method* akan dicantumkan. Implementasi algoritma ini akan direpresentasikan dalam bentuk *pseudocode* dari algoritma – algoritma tersebut.

5.6.1 Implemementasi Algoritma untuk Proses Keluar Masuk Gerbang Parkir

Proses keluar masuk gerbang parkir diimplementasikan secara simulasi. Proses ini dilakukan oleh *user* yang menjalankan proses keluar masuk gerbang parkir. Implementasi algoritma untuk proses keluar masuk gerbang parkir dijelaskan sebagai berikut:

Tabel 5.11 Algoritma proses keluar masuk gerbang parkir

Nama Algoritma : proses keluar masuk gerbang parkir
Deklarasi String → rfid_in, rfid_out, plat_nomor

- Datetime → time_in, time_out
- Double → saldo

Deskripsi

- Masukan : rfid_in, rfid_out, plat_nomor, time_in, time_out
- Proses
 1. *Icon* portal 1 berwarna hijau dan *icon* portal 2 berwarna merah.
 2. *User* masuk ke area pemrosesan keluar masuk gerbang parkir.
 3. *User* mendekatkan RFID *tag* ke RFID *reader*.
 4. RFID *tag* terbaca oleh reader dan ditampilkan dalam aplikasi.
 5. *Icon* portal 1 berwarna merah dan *icon* portal 2 berwarna merah.
 6. Kamera mengambil foto tampak depan kendaraan.
 7. Foto tampak depan kendaraan diproses dalam *class* Detection.
 8. Hasil hasil pemrosesan foto tampak depan kendaraan menjadi data teks plat nomor ditampilkan dalam aplikasi.
 9. *Icon* portal 1 berwarna merah dan *icon* portal 2 berwarna hijau.
 10. Hasil pembacaan RFID *tag* dan data teks plat nomor ditambahkan dengan waktu pendeteksian dikirimkan ke *Store Procedure*.
 11. Pengecekan apakah plat nomor kendaraan telah ada dalam database yang rfid_out-nya masih kosong, bila ada lanjut ke no-9, bila tidak ke no-8.
 12. Memasukkan data teks plat nomor ke plat_nomor, ID dari RFID ke rfid_in dan waktu pendeteksian ke time_in ke basis data, kemudian lanjut ke no-15.
 13. Mencari sisa saldo dari pemilik kendaraan.
 14. Menghitung lama parkir.
 15. Menghitung biaya parkir dan mengurangi dengan sisa saldo.
 16. Memasukkan ID dari RFID ke rfid_out, waktu pendeteksian ke time_out dan sisa saldo ke saldo ke basis data yang memiliki plat_nomor yang sama.
 17. Waktu tunggu 5 detik.
 18. *Icon* portal 1 berwarna merah dan *icon* portal 2 berwarna merah.
 19. Waktu tunggu 2 detik.
 20. *Icon* portal 1 berwarna hijau dan *icon* portal 2 berwarna merah.
- Keluaran : -

Sumber : Implementasi

5.6.2 Implemementasi Algoritma untuk Melakukan *Login*

Login berfungsi untuk melakukan seleksi pada waktu admin akan memasuki halaman utama admin. Implementasi algoritma untuk melakukan seleksi *login* dijelaskan sebagai berikut:

Tabel 5.12 Algoritma login

Nama Algoritma : login
Deklarasi - String → username, password - manage → koneksi Deskripsi - Masukan : username, password - Proses - Admin memasukkan username dan password ke <i>field</i> yang telah tersedia dalam aplikasi. - Menginstansiasi <i>class</i> manage menjadi object koneksi. - Mengirimkan query pada <i>class</i> login ke <i>clas</i> manage untuk dieksekusi apakah data username dan password yang dimasukkan ada di dalam tabel admin. - Jika data benar maka akan dilanjutkan dengan membuka halaman utama admin. - Jika data salah maka tetap akan ada pada halaman login. - Keluaran : -

Sumber : Implementasi

5.6.3 Implemementasi Algoritma untuk Melakukan Pengolahan Admin

Pengolahan admin berfungsi untuk melakukan proses masukkan, perbaruhi dan hapus admin oleh admin yang telah berhasil *login*. Implementasi algoritma untuk melakukan pengolahan admin dijelaskan sebagai berikut:

Tabel 5.13 Algoritma pengolahan admin

Nama Algoritma : pengolahan admin
Deklarasi - String → username, password, nama - manage → koneksi Deskripsi - Masukan : username, password, nama - Proses - Menginstansiasi <i>class</i> manage menjadi object koneksi. <i>Query</i> untuk menampilkan semua data dari tabel admin ke dalam tabel aplikasi dikirimkan ke <i>class</i> manage. - Jika admin akan melakukan masukkan lanjut ke no 3, perbaruhi dan hapus ke no

- 7.
 4. Admin memasukkan username, password dan nama ke *field* yang telah tersedia dalam aplikasi.
 5. Admin menekan *button* masukkan.
 6. Data yang telah diisi oleh admin dimasukkan dalam *query* lalu dikirimkan ke *class* manage, kemudian lanjut ke no 13.
 7. Admin memilih data yang ada dalam tabel dengan cara klik kanan.
 8. Jika admin akan melakukan perbaruhi lanjut ke no 9, jika admin akan melakukan hapus ke no 11.
 9. Admin mengubah data yang ada di *field-field* yang telah tersedia dalam aplikasi.
 10. Admin menekan *button* perbaruhi.
 11. Admin menekan *button* hapus.
 12. Query dikirimkan ke *class* manage.
 13. Status dari hasil pengolahan data di tampilkan di aplikasi bagian bawah.
 14. Perbaruhi tabel dari hasil pengolahan data yang dilakukan admin.
- Keluaran : status proses pengolahan data dan perbaruhi tabel

Sumber : Implementasi

5.6.4 Implemementasi Algoritma untuk Melakukan Pengolahan Pengendara

Pengolahan pengendara berfungsi untuk melakukan proses masukkan, perbaruhi dan hapus pengendara oleh admin yang telah berhasil *login*. Implementasi algoritma untuk melakukan pengolahan pengendara dijelaskan sebagai berikut:

Tabel 5.14 Algoritma pengolahan pengendara

Nama Algoritma : pengolahan pengendara
Deklarasi • String → rfid, no_ktp, nama, jenis_kelamin, alamat, no_hp • manage → koneksi
Deskripsi • Masukan : rfid, no_ktp, nama, jenis_kelamin, alamat, no_hp • Proses 1. Menginstansiasi <i>class</i> manage menjadi object koneksi. 2. <i>Query</i> untuk menampilkan semua data dari tabel pengendara ke dalam tabel aplikasi dikirimkan ke <i>class</i> manage.

3. Jika admin akan melakukan masukkan lanjut ke no 3, perbaruhi dan hapus ke no 7.
 4. Admin memasukkan rfid, no_ktp, nama, jenis_kelamin, alamat dan no_hp ke *field* yang telah tersedia dalam aplikasi.
 5. Admin menekan *button* masukkan.
 6. Data yang telah diisikan oleh admin dimasukkan dalam *query* lalu dikirimkan ke *class* manage, kemudian lanjut ke no 13.
 7. Admin memilih data yang ada dalam tabel dengan cara klik kanan.
 8. Jika admin akan melakukan perbaruhi lanjut ke no 9, jika admin akan melakukan hapus ke no 11.
 9. Admin mengubah data yang ada di *field-field* yang telah tersedia dalam aplikasi.
 10. Admin menekan *button* perbaruhi.
 11. Admin menekan *button* hapus.
 12. Query dikirimkan ke *class* manage.
 13. Status dari hasil pengolahan data di tampilkan di aplikasi bagian bawah.
 14. Perbaruhi tabel dari hasil pengolahan data yang dilakukan admin.
- Keluaran : status proses pengolahan data dan perbaruhi tabel

Sumber : Implementasi

5.6.5 Implementasi Algoritma untuk Melakukan Pengolahan Kendaraan

Pengolahan kendaraan berfungsi untuk melakukan proses masukkan, perbaruhi dan hapus kendaraan oleh admin yang telah berhasil *login*. Implementasi algoritma untuk melakukan pengolahan kendaraan dijelaskan sebagai berikut:

Tabel 5.15 Algoritma pengolahan kendaraan

Nama Algoritma : pengolahan kendaraan
Deklarasi • String → plat_nomor, rfid, tipe, merek, tahun • manage → koneksi Deskripsi • Masukan : plat_nomor, rfid, tipe, merek, tahun • Proses 1. Menginstansiasi <i>class</i> manage menjadi object koneksi. 2. <i>Query</i> untuk menampilkan semua data dari tabel kendaraan ke dalam tabel aplikasi dikirimkan ke <i>class</i> manage.

3. Jika admin akan melakukan masukkan lanjut ke no 3, perbarui dan hapus ke no 7.
 4. Admin memasukkan plat_nomor, rfid, tipe, merek, tahun ke *field* yang telah tersedia dalam aplikasi.
 5. Admin menekan *button* masukkan.
 6. Data yang telah diisikan oleh admin dimasukkan dalam *query* lalu dikirimkan ke *class* manage, kemudian lanjut ke no 13.
 7. Admin memilih data yang ada dalam tabel dengan cara klik kanan.
 8. Jika admin akan melakukan perbarui lanjut ke no 9, jika admin akan melakukan hapus ke no 11.
 9. Admin mengubah data yang ada di *field-field* yang telah tersedia dalam aplikasi.
 10. Admin menekan *button* perbarui.
 11. Admin menekan *button* hapus.
 12. *Query* dikirimkan ke *class* manage.
 13. Status dari hasil pengolahan data di tampilkan di aplikasi bagian bawah.
 14. Jika proses masukkan data maka lanjut ke no 15, jika tidak ke no 20.
 15. *Query* mengecek apakah rfid pemilik kendaraan telah ada dalam tabel saldo.
 16. *Query* dikirimkan ke *class* manage.
 17. Jika rfid pemilik saldo telah ditemukan maka lanjut ke no 20, jika tidak ditemukan ke no 18.
 18. *Query* membuat data baru pada tabel saldo dengan primary key adalah rfid pemilik kendaraan.
 19. *Query* dikirimkan ke *class* manage.
 20. Perbarui tabel dari hasil pengolahan data yang dilakukan admin.
- Keluaran : status proses pengolahan data dan perbarui tabel

Sumber : Implementasi

5.6.6 Implemementasi Algoritma untuk Melakukan Perbarui Pengaturan

Perbarui pengaturan berfungsi untuk melakukan proses perbarui pengaturan oleh admin yang telah berhasil *login*. Implementasi algoritma untuk melakukan perbarui pengaturan dijelaskan sebagai berikut:

Tabel 5.16 Algoritma perbarui pengaturan

Nama Algoritma : perbarui pengaturan
Deklarasi • String → komponen, value • manage → koneksi

Deskripsi

- Masukan : value
- Proses
 1. Menginstansiasi *class* manage menjadi object koneksi.
 2. *Query* untuk menampilkan semua data dari tabel pengaturan ke dalam tabel aplikasi dikirimkan ke *class* manage.
 3. Admin memilih data yang ada dalam tabel dengan cara klik kanan.
 4. Admin mengubah data yang ada di *field-field* yang telah tersedia dalam aplikasi.
 5. Admin menekan *button* perbarui.
 6. *Query* dikirimkan ke *class* manage.
 7. Status dari hasil pengolahan data di tampilkan di aplikasi bagian bawah.
 8. Perbarui tabel dari hasil pengolahan data yang dilakukan admin.
- Keluaran : status proses perbarui data dan perbarui tabel

Sumber : Implementasi

5.6.7 Implemementasi Algoritma untuk Melakukan Pengolahan Saldo

Pengolahan saldo berfungsi untuk melakukan proses perbarui, cetak log dan kirim sms informasi saldo oleh admin yang telah berhasil *login*. Implementasi algoritma untuk melakukan pengolahan saldo dijelaskan sebagai berikut:

Tabel 5.17 Algoritma pengolahan saldo

Nama Algoritma : pengolahan saldo
Deklarasi • String → jumlah, rfid, saldo, tambah_saldo, urutan, awal, akhir, isi • String [][] → ponsel • Datetime → waktu_print • manage → koneksi • HashMap → parameter • JasperPrint → jPrint
Deskripsi • Masukan : saldo • Proses 1. Menginstansiasi <i>class</i> manage menjadi object koneksi. 2. <i>Query</i> untuk menampilkan semua data dari tabel saldo ke dalam tabel aplikasi dikirimkan ke <i>class</i> manage.

3. Admin memilih data yang ada dalam tabel dengan cara klik kanan.
4. Jika admin akan melakukan perbarui ke no 5, cetak log ke no 17 dan kirim sms ke no 29 atau kirim sms ke semua pelanggan ke no 38.
5. Admin mengisi tambahan_saldo yang dibayarkan oleh pelanggan
6. Admin menekan *button* tambah.
7. Data RFID dan tambahan_saldo dikirimkan ke Store Procedure.
8. Tambahkan tambah_saldo dengan saldo menjadi jumlah.
9. Memasukkan jumlah sebagai saldo dengan rfid yang sesuai dengan rfid pelanggan yang menambahkan saldo ke tabel saldo.
10. Mengecek apakah dalam log saldo masih kosong.
11. Jika kosong urutan diisi dengan 0.
12. Menambahkan urutan dengan 1.
13. Memasukkan urutan, rfid, waktu_print, tambah_saldo dan saldo ke tabel logsaldo.
14. Status dari hasil pengolahan data di tampilkan di aplikasi bagian bawah.
15. Perbarui tabel dari hasil pengolahan data yang dilakukan admin.
16. Lanjut ke proses no 41.
17. Menginstansiasi *HashMap* menjadi object parameter.
18. Membuka file LogTambahSaldo.jasper.
19. Mengecek no maks dari log saldo yang rfidnya sama dengan rfid pelanggan dan hasilnya dimasukkan dalam variable akhir.
20. Jika akhir sama dengan 0 maka akhir diisi dengan 1.
21. Memunculkan *dialogbox* yang menanyakan apakah log akan diprint dari awal atau tidak, jika iya lanjut ke no 22, jika tidak ke no 23.
22. Mengisi awal dengan 1, lanjut ke no 24.
23. Mengisi awal sama dengan akhir.
24. Mengisi parameter dengan awal, akhir dan rfid.
25. Menggabungkan jReport, parameter dan koneksi_mysql dengan metode fillReport dalam JasperFillManager ke jPrint.
26. Mengisi metode viewReport dalam JasperViewer dengan jPrint dan false.
27. Menampilkan JasperViewer dengan metode setDefaultLookAndFeelDecorated diisi *true*.
28. Lanjut ke proses no 41.
29. Query mencari no_hp pengendara di tabel pengendara dengan rfid yang sama dengan rfid yang telah dipilih.
30. Query dikirim ke *class* manage dan diproses dengan metode *querySelect*.
31. Hasil query dimasukkan ke variabel ponsel.
32. Mengisi variabel isi dengan query yang mencari isi dari komponen sms_bulanan

- pada tabel pengaturan yang dikirimkan ke *class* manage dan diproses dengan metode *querySelect*.
33. Mengganti isi dari variabel isi yang bertuliskan xxx dengan rfid pelanggan dan yyy dengan jumlah saldo.
 34. Menampilkan *dialogbox* dengan pertanyaan apakah sudah yakin untuk mengirimkan sms ke varibel ponsel dengan isi yang telah diganti di beberapa bagian tadi.
 35. Jika admin memilih yes, lanjut ke no 36, jika memilih no lanjut ke no 41.
 36. Sms akan dikirim dengan menggunakan metode *sendMessage* pada *class* *SMSCClient* dengan parameter isi dan ponsel.
 37. Lanjut ke proses no 41.
 38. Menjalankan seperti langkah no 32, kemudian menampilkan *dialogbox* dengan pertanyaan apakah sudah yakin untuk mengirimkan sms ke semua pelanggan dengan isi seperti yang telah diambil dari tabel pengaturan.
 39. Jika admin memilih yes, maka dikirimkan *query* yang mencari rfid, no_hp dan saldo dari tabel pengemudi dan saldo ke *class* manage dengan metode *querySelectMatrix*.
 40. Melakukan proses no 33 dan 36 secara terus menerus hingga semua pelanggan yang terdaftar telah dikirimkan sms informasi saldo.
 41. Kembali ke halaman manage saldo.
- Keluaran : status proses pengolahan data dan perbarui tabel

Sumber : Implementasi

5.6.8 Implementasi Algoritma untuk Melakukan Pengolahan Log Parkir

Pengolahan log parkir berfungsi untuk melakukan proses search, cetak log dan kirim sms informasi kehilangan oleh admin yang telah berhasil *login*. Implementasi algoritma untuk melakukan pengolahan saldo dijelaskan sebagai berikut:

Tabel 5.18 Algoritma pengolahan log parkir

Nama Algoritma : pengolahan log parkir
Deklarasi
• String → awal, akhir, rfid, nopolisi, isi • String [][] → terakhirbawa, pemilik, ponsel • Datetime → waktu_print

- manage → koneksi
- HashMap → parameter
- JasperPrint → jPrint

Deskripsi

- Masukan : -
- Proses
 1. Menginstansiasi *class* manage menjadi object koneksi.
 2. Admin memilih cara menampilkan data log parkir dalam tabel yang ada pada aplikasi dengan menampilkan keseluruhan atau dengan rentangan waktu yang ditentukan atau dengan no plat nomor tertentu atau dengan waktu yang ditentukan ditambah dengan no plat nomor yang ditentukan atau dengan rfid yang ditentukan.
 3. Mengirimkan query sesuai dengan pilihan admin dengan metode *tampil_tabel* pada *class* manage dengan cara menekan *button* cari.
 4. Jika admin akan melakukan cetak log ke no 5 atau ke no 15 untuk mengirim laporan kehilangan.
 5. Membuka file .jasper sesuai dengan pilihan yang awal telah dipilih oleh admin untuk menampilkan data log parkir.
 6. Jika admin menampilkan data pada tabel dengan cara rentangan waktu tertentu maka awal dan akhir diisi dengan waktu yang telah dimasukkan admin, kemudian lanjut ke no 9, jika dengan rfid maka lanjut ke no 7.
 7. Mengisikan variabel awal dengan data hasil dari query mengambil waktu_print yang rfidnya sama dengan rfid yang telah dituliskan oleh admin dari tabel saldo dengan menggunakan metode *querySelect* pada *class* manage dan variabel akhir dengan waktu sekarang.
 8. Mengperbarui waktu_print yang rfidnya sama dengan rfid yang telah dituliskan oleh admin dari tabel saldo dengan menggunakan *querySelect* pada *class* manage.
 9. Menginstansiasi *HashMap* menjadi object parameter.
 10. Mengisi parameter dengan awal, akhir, rfid dan nopolisi.
 11. Menggabungkan jReport, parameter dan koneksi_mysql dengan metode *fillReport* dalam *JasperFillManager* ke jPrint.
 12. Mengisi metode *viewReport* dalam *JasperViewer* dengan jPrint dan false.
 13. Menampilkan *JasperViewer* dengan metode *setDefaultLookAndFeelDecorated* diisi *true*.
 14. Lanjut ke proses no 41.
 15. Admin memilih data yang ada dalam tabel dengan cara klik kanan.
 16. Mengambil nopolisi yang ada di jLabel5.

17. Mengisi variabel array terakhirbawa yang berisi nama dan no_ktp pengendara yang terakhir membawa kendaraan yang hilang dengan query mencari nama dan no_ktp pada tabel logparkir data yang terakhir yang nopolisinya sama dengan nopolisi yang dilaporkan dan rfid_out sama dengan rfid pada tabel pengendara dengan metode *querySelectMatrix* pada *class* manage.
 18. Mengisi variabel array pemilik yang berisi nama dan no_ktp pemilik kendaraan yang hilang dengan query mencari nama dan no_ktp pada tabel kendaraan yang no_polisinya sama dengan no_polisi yang dilaporkan dan rfid pada tabel kendaraan sama dengan rfid pada tabel pengendara dengan metode *querySelectMatrix* pada *class* manage.
 19. Mengisi variabel isi dengan query yang mencari isi dari komponen sms_hilang pada tabel pengaturan yang dikirimkan ke *class* manage dan diproses dengan metode *querySelect*.
 20. Mengganti isi dari variabel isi yang bertuliskan xxx dengan nopolisi pelanggan, yyy dengan nama pemilik kendaraan, zzz dengan no_ktp pemilik kendaraan, zzx dengan nama pengendara yang terakhir membawa kendaraan dan zzy dengan no_ktp pengendara yang terakhir membawa kendaraan.
 21. Menampilkan *dialogbox* dengan pertanyaan apakah sudah yakin untuk mengirimkan sms dengan isi yang telah diganti di beberapa bagian tadi.
 22. Jika admin memilih yes, lanjut ke no 23, jika memilih no lanjut ke no 26.
 23. Mengisikan variabel array ponsel dengan *query* yang mencari value dari komponen dengan berisi manager, keamanan1 dan keamanan2 dari tabel pengaturan ke *class* manage dengan metode *querySelectMatrix*.
 24. Sms akan dikirim dengan menggunakan metode *sendMessage* pada *class* *SMSCClient* dengan parameter isi dan ponsel.
 25. Proses no 24 akan dilakukan terus menerus hingga sms berhasil dikirimkan kepada 3 nomor ponsel yang telah dimasukkan pada variabel array ponsel tadi.
 26. Kembali ke halaman manage log parkir.
- Keluaran : -

Sumber : Implementasi

5.7 Implementasi Antarmuka

Implementasi antarmuka sistem perangkat lunak untuk simulasi keluar masuk dan pembayaran parkir kendaraan berbasis RFID terdiri dari 2 bagian, yaitu implementasi antarmuka aplikasi gerbang parkir dan implementasi antarmuka aplikasi administrasi. Implementasi antarmuka subsistem gerbang

parkir dilakukan secara simulasi. Implementasi antarmuka subsistem administrasi dibuat dalam *desktop application*.

5.7.1 Implementasi Antarmuka Subsistem Gerbang Parkir

Antarmuka pengguna subsistem gerbang parkir merupakan sebuah halaman yang mengatur proses yang ada pada gerbang parkir secara simulasi dari sistem perangkat lunak untuk simulasi keluar masuk dan pembayaran parkir kendaraan berbasis RFID. Gambar 5.2 menunjukkan tampilan implementasi antarmuka subsistem gerbang parkir.



Gambar 5.2 Implementasi antarmuka subsistem gerbang parkir

Sumber : Implementasi

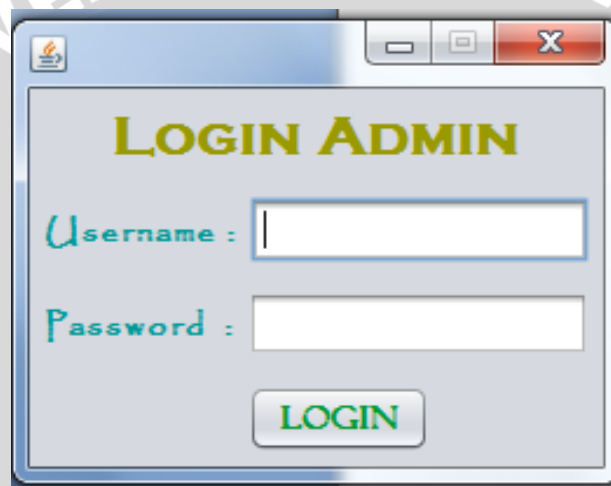
5.7.2 Implementasi Antarmuka Subsistem Administrasi

Antarmuka pengguna subsistem administrasi merupakan halaman administrasi yang dapat digunakan untuk mengelola data dan unsur – unsur lain yang diperlukan oleh Sistem Perangkat Lunak Untuk Simulasi Keluar Masuk dan Pembayaran Parkir Kendaraan Berbasis RFID.

Antarmuka pengguna untuk subsistem administrasi terdiri atas halaman login, halaman utama, halaman admin, halaman pengaturan pengendara, halaman pengaturan kendaraan, halaman pengaturan saldo, halaman pengaturan log parkir dan halaman pengaturan.

a. Halaman Login

Halaman login merupakan halaman yang berfungsi untuk mempermudah admin dalam melakukan login. Gambar 5.3 akan menunjukkan implementasi antarmuka dari halaman login.

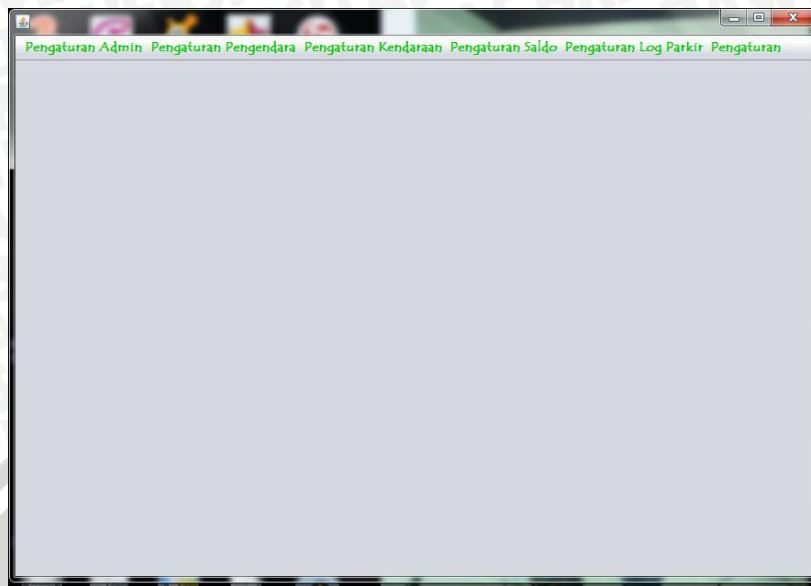


Gambar 5.3 Implementasi antarmuka halaman login

Sumber : Implementasi

b. Halaman Utama

Halaman utama merupakan halaman yang berfungsi untuk mempermudah admin dalam memilih menu admin. Gambar 5.4 akan menunjukkan implementasi antarmuka dari halaman utama.

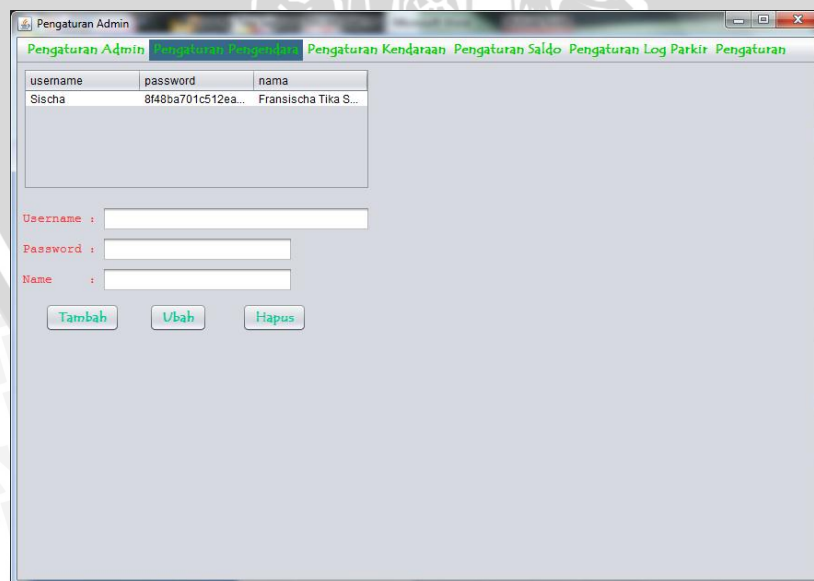


Gambar 5.4 Implementasi antarmuka halaman utama admin

Sumber : Implementasi

c. Halaman Pengaturan Admin

Halaman pengaturan admin merupakan halaman yang berfungsi untuk mengolah data admin. Gambar 5.5 akan menunjukkan implementasi antarmuka dari halaman pengaturan admin.



Gambar 5.5 Implementasi antarmuka halaman pengaturan admin

Sumber : Implementasi

d. Halaman Pengaturan Pengendara

Halaman pengaturan pengendara merupakan halaman yang berfungsi untuk mengolah data pengendara. Gambar 5.6 akan menunjukkan implementasi antarmuka dari halaman pengaturan pengendara.

RFID	No KTP	Nama	Jenis K...	Alamat	No HP
085276...	000000...	DuaY	Laki-laki	street	081913...
085417...	000000...	SatuX	Laki-laki	jalan	085731...

RFID :

No KTP :

Nama :

Jenis Kelamin : ☐ Laki-laki ☐ Perempuan

Alamat :

Ponsel :

Gambar 5.6 Implementasi antarmuka halaman pengaturan pengendara

Sumber : Implementasi

e. Halaman Pengaturan Kendaraan

Halaman pengaturan kendaraan merupakan halaman yang berfungsi untuk mengolah data kendaraan. Gambar 5.7 akan menunjukkan implementasi antarmuka dari halaman pengaturan kendaraan.

No Polisi	Tipe	Merek	Tahun	No KTP
AG 0902 N	Ranger	Ranger	2013	0000000...
N 1925 AG	Juke	Nissan	2012	0000000...
S 15 CHA	Jazz	Honda	2010	0000000...

No Polisi :

Tipe :

Merek :

Tahun :

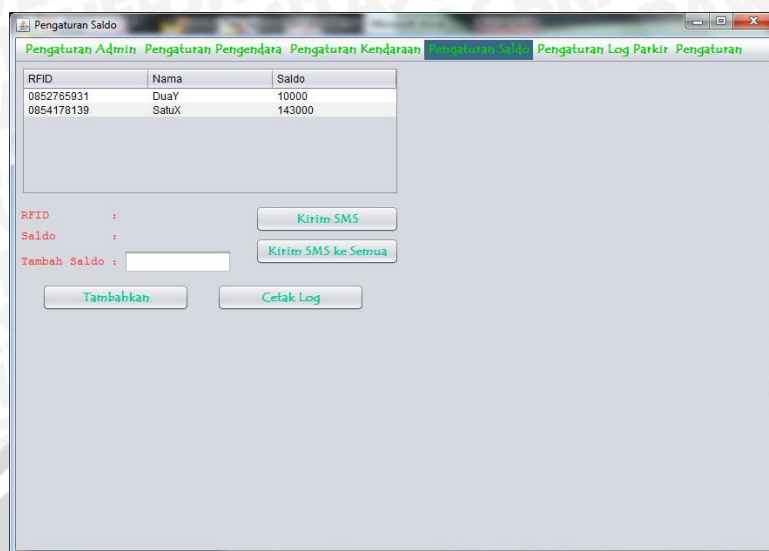
No KTP :

Gambar 5.7 Implementasi antarmuka halaman pengaturan kendaraan

Sumber : Implementasi

f. Halaman Pengaturan Saldo

Halaman pengaturan saldo merupakan halaman yang berfungsi untuk mengolah data saldo. Dalam pengaturan saldo ini juga tersedia fungsi untuk mengirim informasi saldo melalui sms. Secara otomatis sms informasi saldo di kirimkan bila saldo pelanggan kurang atau sama dengan Rp 0. Selain itu, sms informasi saldo juga dikirimkan setiap bulannya oleh petugas admin. Gambar 5.8 akan menunjukkan implementasi antarmuka dari halaman pengaturan saldo.

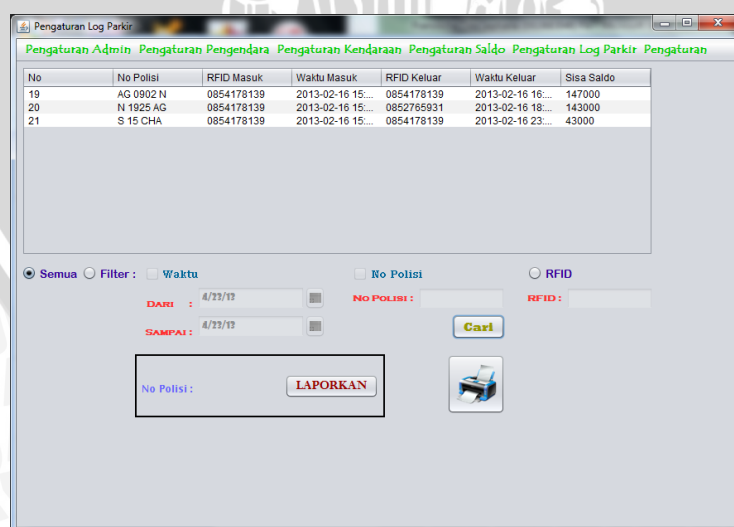


Gambar 5.8 Implementasi antarmuka halaman pengaturan saldo

Sumber : Implementasi

g. Halaman Pengaturan Log Parkir

Halaman pengaturan log parkir merupakan halaman yang berfungsi untuk mengolah data log parkir. Halaman pengaturan log parkir ini memiliki fitur mengirimkan sms laporan kehilangan kendaraan kepada pihak yang berwajib dan bertanggung jawab. Gambar 5.9 akan menunjukkan implementasi antarmuka dari halaman pengaturan log parkir.

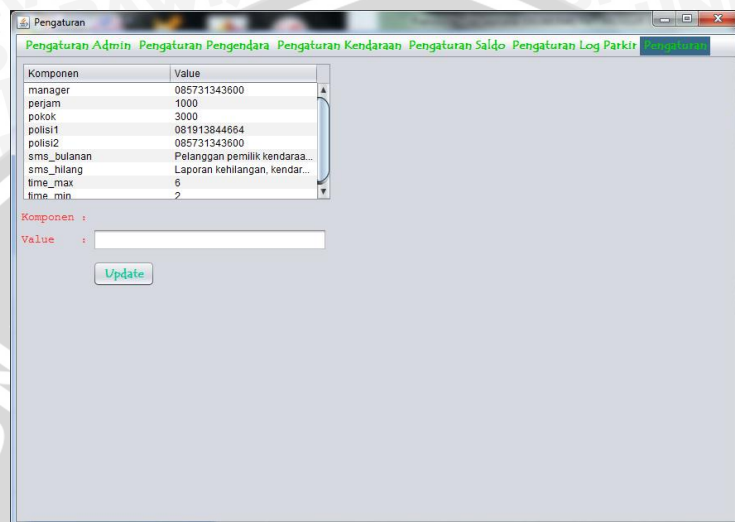


Gambar 5.9 Implementasi antarmuka halaman pengaturan log parkir

Sumber : Implementasi

h. Halaman Pengaturan

Halaman pengaturan merupakan halaman yang berfungsi untuk mengolah data setting. Gambar 5.10 akan menunjukkan implementasi antarmuka dari halaman pengaturan.



Gambar 5.10 Implementasi antarmuka pengaturan setting

Sumber : Implementasi

BAB VI

PENGUJIAN DAN PEMBAHASAN

Bab ini membahas mengenai tahapan pengujian dan pembahasan sistem perangkat lunak untuk simulasi keluar masuk dan pembayaran parkir kendaraan berbasis RFID yang telah dikembangkan. Proses pengujian dilakukan melalui dua macam pengujian yaitu pengujian validasi dan pengujian performa. Pengujian validasi digunakan untuk menguji kebenaran jalannya kebutuhan fungsional sistem perangkat lunak untuk simulasi keluar masuk dan pembayaran parkir kendaraan berbasis RFID. Pengujian performa digunakan untuk menguji kecepatan proses sistem perangkat lunak. Proses pengujian yang dilakukan melalui dua macam pengujian yaitu pengujian validasi dan pengujian performa.

Proses pembahasan bertujuan untuk mendapatkan kesimpulan dari hasil pengujian sistem perangkat lunak untuk simulasi keluar masuk dan pembayaran parkir kendaraan berbasis RFID yang telah dilakukan. Proses pembahasan mengacu pada dasar teori sesuai dengan hasil pengujian yang didapatkan. Pembahasan dilakukan terhadap hasil pengujian di setiap tahap pengujian. Proses pembahasan yang dilakukan meliputi pembahasan hasil pengujian validasi dan pengujian performa.

6.1 Pengujian Validasi

Pengujian validasi digunakan untuk mengetahui apakah sistem yang dibangun sudah benar sesuai dengan yang dibutuhkan. *Item - item* yang telah dirumuskan dalam daftar kebutuhan dan merupakan hasil analisis kebutuhan akan menjadi acuan untuk melakukan pengujian validasi. Pengujian validasi berfungsi untuk menemukan kesesuaian antara kerja sistem dengan daftar kebutuhan yang telah dirancang sebelumnya. Pada skripsi ini dilakukan 2 macam pengujian validasi terhadap sistem perangkat lunak untuk simulasi keluar masuk dan pembayaran parkir kendaraan berbasis RFID, yaitu pengujian fungsional berdasarkan usecase digram dan pengujian kondisi (state) gerbang parkir.

6.1.1 Pengujian Fungsional Berdasarkan Use Case Diagram

Setelah menentukan item-item apa saja yang akan diuji, bagaimana prosedur pengujian dan menentukan hasil yang diharapkan, item-item yang telah didata tadi diuji satu persatu sesuai dengan prosedur pengujian yang telah dituliskan.

6.1.1.1 Kasus dan Hasil Pengujian Fungsional Berdasarkan Use Case Diagram

a. Kasus Uji Menunjukkan tag RFID

Tabel 6.1 Kasus uji untuk menunjukkan tag RFID

Nama Kasus Uji	Kasus Uji Menunjukkan Tag RFID
Objek Uji	Kebutuhan Fungsional (SRS_001_01)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional untuk melakukan pembacaan tag RFID.
Prosedur Uji	1. Sistem dijalankan saat program dieksekusi. 2. <i>User</i> mendekatkan tag RFID pada <i>reader</i> RFID.
Hasil yang Diharapkan	Aplikasi dapat membaca tag RFID, kemudian menampilkannya dalam aplikasi.
Hasil yang Didapatkan	Aplikasi dapat membaca tag RFID, kemudian menampilkannya dalam aplikasi.
Status Validitas	Valid

Sumber : Pengujian dan Analisis

b. Kasus Uji Membawa Kendaraan Untuk Dapat Diambil Gambarnya

Tabel 6.2 Kasus uji membawa kendaraan untuk dapat diambil gambarnya

Nama Kasus Uji	Kasus Uji Membawa Kendaraan Untuk Dapat Diambil Gambarnya
Objek Uji	Kebutuhan Fungsional (SRS_001_02)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional untuk melakukan pengambilan gambar tampak depan kendaraan.

Prosedur Uji	1. Sistem dijalankan saat program dieksekusi. 2. <i>User</i> membawa kendaraan ke area gerbang parkir pada posisi yang telah ditentukan.
Hasil yang Diharapkan	Aplikasi dapat mengambil gambar tampak dengan kendaraan dan dapat membaca plat nomor kendaraan lalu mengubahnya menjadi data teks, kemudian menampilkannya dalam aplikasi.
Hasil yang Didapatkan	Aplikasi dapat mengambil gambar tampak dengan kendaraan dan dapat membaca plat nomor kendaraan lalu mengubahnya menjadi data teks, kemudian menampilkannya dalam aplikasi.
Status Validitas	Valid

Sumber : Pengujian dan Analisis

c. Kasus Uji *Login*

Tabel 6.3 Kasus uji untuk *login*

Nama Kasus Uji	Kasus Uji <i>Login</i>
Objek Uji	Kebutuhan Fungsional (SRS_002_01)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional untuk melakukan proses login.
Prosedur Uji	1. <i>Administrator</i> memasukkan data yang diperlukan di dalam halaman login. 2. <i>Administrator</i> menekan tombol Login.
Hasil yang Diharapkan	Aplikasi dapat melakukan penyeleksian kondisi login pada basis data berdasar data yang dimasukkan. Jika penyeleksian kondisi ini tidak benar, maka akan ditampilkan pernyataan bahwa login salah, tetapi jika login benar maka akan masuk pada halaman utama administrasi.
Hasil yang Didapatkan	Aplikasi dapat melakukan penyeleksian kondisi login pada basis data berdasar data yang dimasukkan. Jika penyeleksian kondisi ini tidak benar, maka akan ditampilkan pernyataan bahwa login salah, tetapi jika login benar maka akan masuk pada halaman utama administrasi.
Status Validitas	Valid

Sumber : Pengujian dan Analisis

d. Kasus Uji Masukkan Data Admin

Tabel 6.4 Kasus uji untuk masukkan data admin

Nama Kasus Uji	Kasus Uji Masukkan Data Admin
Objek Uji	Kebutuhan Fungsional (SRS_002_02)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional untuk melakukan proses masukkan data admin.
Prosedur Uji	1. <i>Administrator</i> memasukkan data yang diperlukan di dalam halaman manage admin. 2. <i>Administrator</i> menekan tombol masukkan.
Hasil yang Diharapkan	Aplikasi dapat melakukan masukkan data admin ke database. Jika pemasukan data admin berhasil, maka akan ditampilkan pernyataan bahwa masukkan data berhasil, tetapi jika masukkan data gagal maka akan ditampilkan pernyataan bahwa masukkan data gagal.
Hasil yang Didapatkan	Aplikasi dapat melakukan masukkan data admin ke database. Jika pemasukan data admin berhasil, maka akan ditampilkan pernyataan bahwa masukkan data berhasil, tetapi jika masukkan data gagal maka akan ditampilkan pernyataan bahwa masukkan data gagal.
Status Validitas	Valid

Sumber : Pengujian dan Analisis

e. Kasus Uji Perbaruhi Data Admin

Tabel 6.5 Kasus uji untuk perbaruhi data admin

Nama Kasus Uji	Kasus Uji Perbaruhi Data Admin
Objek Uji	Kebutuhan Fungsional (SRS_002_02)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional untuk melakukan proses perbaruhi data admin.
Prosedur Uji	1. <i>Administrator</i> mengeklik data tabel yang diperlukan di halaman manage admin. 2. <i>Administrator</i> meng-perbaruhi data yang diperlukan di dalam halaman manage admin.

	3. <i>Administrator</i> menekan tombol perbarui.
Hasil yang Diharapkan	Aplikasi dapat melakukan perbarui data admin ke database. Jika peng-perbarui-an data admin berhasil, maka akan ditampilkan pernyataan bahwa perbarui data berhasil, tetapi jika perbarui data gagal maka akan ditampilkan pernyataan bahwa perbarui data gagal.
Hasil yang Didapatkan	Aplikasi dapat melakukan perbarui data admin ke database. Jika peng-perbarui-an data admin berhasil, maka akan ditampilkan pernyataan bahwa perbarui data berhasil, tetapi jika perbarui data gagal maka akan ditampilkan pernyataan bahwa perbarui data gagal.
Status Validitas	Valid

Sumber : Pengujian dan Analisis

f. Kasus Uji Hapus Data Admin

Tabel 6.6 Kasus uji untuk hapus data admin

Nama Kasus Uji	Kasus Uji Hapus Data Admin
Objek Uji	Kebutuhan Fungsional (SRS_002_02)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional untuk melakukan proses hapus data admin.
Prosedur Uji	1. <i>Administrator</i> mengeklik data tabel yang diperlukan di halaman manage admin. 2. <i>Administrator</i> meng-hapus data yang diperlukan di dalam halaman manage admin. 3. <i>Administrator</i> menekan tombol hapus.
Hasil yang Diharapkan	Aplikasi dapat melakukan hapus data admin ke database. Jika peng-hapus-an data admin berhasil, maka akan ditampilkan pernyataan bahwa hapus data berhasil, tetapi jika hapus data gagal maka akan ditampilkan pernyataan bahwa hapus data gagal.
Hasil yang Didapatkan	Aplikasi dapat melakukan hapus data admin ke database. Jika peng-hapus-an data admin berhasil, maka akan ditampilkan pernyataan bahwa hapus data berhasil, tetapi jika hapus data gagal maka akan ditampilkan pernyataan bahwa hapus data gagal.
Status Validitas	Valid

Sumber : Pengujian dan Analisis

g. Kasus Uji Masukkan Data Kendaraan

Tabel 6.7 Kasus uji untuk masukkan data kendaraan

Nama Kasus Uji	Kasus Uji Masukkan Data Kendara
Objek Uji	Kebutuhan Fungsional (SRS_002_03)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional untuk melakukan proses masukkan data kendaraan.
Prosedur Uji	1. <i>Administrator</i> memasukkan data yang diperlukan di dalam halaman manage kendaraan. 2. <i>Administrator</i> menekan tombol masukkan.
Hasil yang Diharapkan	Aplikasi dapat melakukan masukkan data kendaraan ke database. Jika pemasukan data kendaraan berhasil, maka akan ditampilkan pernyataan bahwa masukkan data berhasil, tetapi jika masukkan data gagal maka akan ditampilkan pernyataan bahwa masukkan data gagal.
Hasil yang Didapatkan	Aplikasi dapat melakukan masukkan data kendaraan ke database. Jika pemasukan data kendaraan berhasil, maka akan ditampilkan pernyataan bahwa masukkan data berhasil, tetapi jika masukkan data gagal maka akan ditampilkan pernyataan bahwa masukkan data gagal.
Status Validitas	Valid

Sumber : Pengujian dan Analisis

h. Kasus Uji Perbaruhi Data Kendaraan

Tabel 6.8 Kasus uji untuk perbaruhi data kendaraan

Nama Kasus Uji	Kasus Uji Perbaruhi Data Kendaraan
Objek Uji	Kebutuhan Fungsional (SRS_002_03)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional untuk melakukan proses perbaruhi data kendaraan.
Prosedur Uji	1. <i>Administrator</i> mengeklik data tabel yang diperlukan di halaman manage kendaraan. 2. <i>Administrator</i> meng-perbaruhi data yang diperlukan di dalam halaman manage kendaraan.

	3. <i>Administrator</i> menekan tombol perbarui.
Hasil yang Diharapkan	Aplikasi dapat melakukan perbarui data kendaraan ke database. Jika peng-perbarui-an data kendaraan berhasil, maka akan ditampilkan pernyataan bahwa perbarui data berhasil, tetapi jika perbarui data gagal maka akan ditampilkan pernyataan bahwa perbarui data gagal.
Hasil yang Didapatkan	Aplikasi dapat melakukan perbarui data kendaraan ke database. Jika peng-perbarui-an data kendaraan berhasil, maka akan ditampilkan pernyataan bahwa perbarui data berhasil, tetapi jika perbarui data gagal maka akan ditampilkan pernyataan bahwa perbarui data gagal.
Status Validitas	Valid

Sumber : Pengujian dan Analisis

i. Kasus Uji Hapus Data Kendaraan

Tabel 6.9 Kasus uji untuk hapus data kendaraan

Nama Kasus Uji	Kasus Uji Hapus Data Kendaraan
Objek Uji	Kebutuhan Fungsional (SRS_002_03)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional untuk melakukan proses hapus data kendaraan.
Prosedur Uji	1. <i>Administrator</i> mengeklik data tabel yang diperlukan di halaman manage kendaraan. 2. <i>Administrator</i> meng-hapus data yang diperlukan di dalam halaman manage kendaraan. 3. <i>Administrator</i> menekan tombol hapus.
Hasil yang Diharapkan	Aplikasi dapat melakukan hapus data kendaraan ke database. Jika peng-hapus-an data kendaraan berhasil, maka akan ditampilkan pernyataan bahwa hapus data berhasil, tetapi jika hapus data gagal maka akan ditampilkan pernyataan bahwa hapus data gagal.
Hasil yang Didapatkan	Aplikasi dapat melakukan hapus data kendaraan ke database. Jika peng-hapus-an data kendaraan berhasil, maka akan ditampilkan pernyataan bahwa hapus data berhasil, tetapi jika hapus data gagal maka akan ditampilkan pernyataan bahwa hapus data gagal.
Status Validitas	Valid

Sumber : Pengujian dan Analisis

j. Kasus Uji Masukkan Data Pengendara

Tabel 6.10 Kasus uji untuk masukkan data pengendara

Nama Kasus Uji	Kasus Uji Masukkan Data Pengendara
Objek Uji	Kebutuhan Fungsional (SRS_002_04)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional untuk melakukan proses masukkan data pengendara.
Prosedur Uji	1. <i>Administrator</i> memasukkan data yang diperlukan di dalam halaman manage pengendara. 2. <i>Administrator</i> menekan tombol masukkan.
Hasil yang Diharapkan	Aplikasi dapat melakukan masukkan data pengendara ke database. Jika pemasukan data pengendara berhasil, maka akan ditampilkan pernyataan bahwa masukkan data berhasil, tetapi jika masukkan data gagal maka akan ditampilkan pernyataan bahwa masukkan data gagal.
Hasil yang Didapatkan	Aplikasi dapat melakukan masukkan data pengendara ke database. Jika pemasukan data pengendara berhasil, maka akan ditampilkan pernyataan bahwa masukkan data berhasil, tetapi jika masukkan data gagal maka akan ditampilkan pernyataan bahwa masukkan data gagal.
Status Validitas	Valid

Sumber : Pengujian dan Analisis

k. Kasus Uji Perbaruhi Data Pengendara

Tabel 6.11 Kasus uji untuk perbaruhi data pengendara

Nama Kasus Uji	Kasus Uji Perbaruhi Data Pengendara
Objek Uji	Kebutuhan Fungsional (SRS_002_04)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional untuk melakukan proses perbaruhi data pengendara.
Prosedur Uji	1. <i>Administrator</i> mengeklik data tabel yang diperlukan di halaman manage pengendara. 2. <i>Administrator</i> meng-perbaruhi data yang diperlukan di dalam halaman manage pengendara.

	3. <i>Administrator</i> menekan tombol perbarui.
Hasil yang Diharapkan	Aplikasi dapat melakukan perbarui data pengendara ke database. Jika peng-perbarui-an data pengendara berhasil, maka akan ditampilkan pernyataan bahwa perbarui data berhasil, tetapi jika perbarui data gagal maka akan ditampilkan pernyataan bahwa perbarui data gagal.
Hasil yang Didapatkan	Aplikasi dapat melakukan perbarui data pengendara ke database. Jika peng-perbarui-an data pengendara berhasil, maka akan ditampilkan pernyataan bahwa perbarui data berhasil, tetapi jika perbarui data gagal maka akan ditampilkan pernyataan bahwa perbarui data gagal.
Status Validitas	Valid

Sumber : Pengujian dan Analisis

1. Kasus Uji Hapus Data Pengendara

Tabel 6.12 Kasus uji untuk hapus data pengendara

Nama Kasus Uji	Kasus Uji Hapus Data Penngendara
Objek Uji	Kebutuhan Fungsional (SRS_002_04)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional untuk melakukan proses hapus data pengendara.
Prosedur Uji	1. <i>Administrator</i> mengeklik data tabel yang diperlukan di halaman manage pengendara. 2. <i>Administrator</i> meng-hapus data yang diperlukan di dalam halaman manage pengendara. 3. <i>Administrator</i> menekan tombol hapus.
Hasil yang Diharapkan	Aplikasi dapat melakukan hapus data pengendara ke database. Jika peng-hapus-an data pengendara berhasil, maka akan ditampilkan pernyataan bahwa hapus data berhasil, tetapi jika hapus data gagal maka akan ditampilkan pernyataan bahwa hapus data gagal.
Hasil yang Didapatkan	Aplikasi dapat melakukan hapus data pengendara ke database. Jika peng-hapus-an data pengendara berhasil, maka akan ditampilkan pernyataan bahwa hapus data berhasil, tetapi jika hapus data gagal maka akan ditampilkan pernyataan bahwa hapus data gagal.
Status Validitas	Valid

Sumber : Pengujian dan Analisis

m. Kasus Uji Tambah Saldo

Tabel 6.13 Kasus uji untuk menambah saldo

Nama Kasus Uji	Kasus Uji Tambah Saldo
Objek Uji	Kebutuhan Fungsional (SRS_002_05)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional untuk melakukan proses penambahan saldo.
Prosedur Uji	1. <i>Administrator</i> mengeklik data tabel yang diperlukan di halaman manage saldo. 2. <i>Administrator</i> menuliskan nilai saldo yang ingin ditambahkan di dalam halaman manage saldo. 3. <i>Administrator</i> menekan tombol tambahkan.
Hasil yang Diharapkan	Aplikasi dapat melakukan penambahan data saldo pemilik kendaraan ke database. Jika penambahan saldo berhasil, maka akan ditampilkan pernyataan bahwa penambahan saldo berhasil, tetapi jika penambahan saldo gagal maka akan ditampilkan pernyataan bahwa penambahan saldo gagal.
Hasil yang Didapatkan	Aplikasi dapat melakukan penambahan data saldo pemilik kendaraan ke database. Jika penambahan saldo berhasil, maka akan ditampilkan pernyataan bahwa penambahan saldo berhasil, tetapi jika penambahan saldo gagal maka akan ditampilkan pernyataan bahwa penambahan saldo gagal.
Status Validitas	Valid

Sumber : Pengujian dan Analisis

n. Kasus Uji Cetak Log Saldo

Tabel 6.14 Kasus uji untuk mencetak log saldo

Nama Kasus Uji	Kasus Uji Cetak Log Saldo
Objek Uji	Kebutuhan Fungsional (SRS_002_05)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional untuk melakukan proses pencetakan log saldo.
Prosedur Uji	1. <i>Administrator</i> mengeklik data tabel yang diperlukan di halaman manage saldo.

	2. <i>Administrator</i> menekan tombol cetak log. 3. <i>Administrator</i> memilih yes atau no saat diberi pertanyaan ingin mencetak log dari awal atau tidak.
Hasil yang Diharapkan	Aplikasi dapat mencetak log saldo dengan rentangan waktu sesuai dengan yang diinginkan.
Hasil yang Didapatkan	Aplikasi dapat mencetak log saldo dengan rentangan waktu sesuai dengan yang diinginkan.
Status Validitas	Valid

Sumber : Pengujian dan Analisis

o. Kasus Uji Kirim SMS ke Pemilik Kendaraan

Tabel 6.15 Kasus uji untuk mengirim SMS ke pemilik kendaraan

Nama Kasus Uji	Kasus Uji Kirim SMS ke Pemilik Kendaraan
Objek Uji	Kebutuhan Fungsional (SRS_002_05)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional untuk melakukan proses pengiriman sms informasi saldo ke pemilik kendaraan.
Prosedur Uji	1. <i>Administrator</i> mengeklik data tabel yang diperlukan di halaman manage saldo. 2. <i>Administrator</i> menekan tombol kirim SMS. 3. <i>Administrator</i> memilih apakah format penulisan telah sesuai atau belum.
Hasil yang Diharapkan	Jika <i>Administrator</i> memilih yes, maka aplikasi dapat mengirim sms ke pemilik kendaraan yang telah dipilih oleh <i>administrator</i> . Jika <i>Administrator</i> memilih no, maka aplikasi kembali menampilkan halaman manage saldo.
Hasil yang Didapatkan	Jika <i>Administrator</i> memilih yes, maka aplikasi dapat mengirim sms ke pemilik kendaraan yang telah dipilih oleh <i>administrator</i> . Jika <i>Administrator</i> memilih no, maka aplikasi kembali menampilkan halaman manage saldo.
Status Validitas	Valid

Sumber : Pengujian dan Analisis

p. Kasus Uji Kirim SMS ke Semua Pemilik Kendaraan

Tabel 6.16 Kasus uji untuk mengirim SMS ke semua pemilik kendaraan

Nama Kasus Uji	Kasus Uji Kirim SMS ke Semua Pemilik Kendaraan
Objek Uji	Kebutuhan Fungsional (SRS_002_05)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional untuk melakukan proses pengiriman sms informasi saldo ke semua pemilik kendaraan.
Prosedur Uji	1. <i>Administrator</i> menekan tombol kirim SMS ke semua. 2. <i>Administrator</i> memilih apakah format penulisan telah sesuai atau belum.
Hasil yang Diharapkan	Jika <i>Administrator</i> memilih yes, maka aplikasi dapat mengirim sms ke semua pemilik kendaraan. Jika <i>Administrator</i> memilih no, maka aplikasi kembali menampilkan halaman manage saldo.
Hasil yang Didapatkan	Jika <i>Administrator</i> memilih yes, maka aplikasi dapat mengirim sms ke semua pemilik kendaraan. Jika <i>Administrator</i> memilih no, maka aplikasi kembali menampilkan halaman manage saldo.
Status Validitas	Valid

Sumber : Pengujian dan Analisis

q. Kasus Uji Mengolah Data Pengaturan

Tabel 6.17 Kasus uji untuk mengelolah data pengaturan

Nama Kasus Uji	Kasus Uji Mengolah Data Pengaturan
Objek Uji	Kebutuhan Fungsional (SRS_002_06)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional untuk melakukan proses pengolahan data pengaturan.
Prosedur Uji	1. <i>Administrator</i> mengeklik data tabel yang diperlukan di halaman setting. 2. <i>Administrator</i> menuliskan isi dari komponen data pengaturan yang ingin diperbaruhi di dalam halaman setting. 3. <i>Administrator</i> menekan tombol perbaruhi.
Hasil yang Diharapkan	Aplikasi dapat melakukan perbaruhi data pengaturan ke database. Jika peng-perbaruhi-an data pengaturan berhasil, maka akan ditampilkan pernyataan bahwa perbaruhi data berhasil, tetapi jika

	perbarui data gagal maka akan ditampilkan pernyataan bahwa perbarui data gagal.
Hasil yang Didapatkan	Aplikasi dapat melakukan perbarui data pengaturan ke database. Jika peng-perbarui-an data pengaturan berhasil, maka akan ditampilkan pernyataan bahwa perbarui data berhasil, tetapi jika perbarui data gagal maka akan ditampilkan pernyataan bahwa perbarui data gagal.
Status Validitas	Valid

Sumber : Pengujian dan Analisis

r. Kasus Uji Search Log Parkir

Tabel 6.18 Kasus uji untuk mencari data log parkir

Nama Kasus Uji	Kasus Uji Search Log Parkir
Objek Uji	Kebutuhan Fungsional (SRS_002_07)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional untuk melakukan proses pencarian data log parkir.
Prosedur Uji	1. <i>Administrator</i> memilih batasan-batasan yang diperlukan untuk mencari data log parkir sesuai dengan kebutuhan. 2. <i>Administrator</i> menekan tombol <i>search</i> .
Hasil yang Diharapkan	Aplikasi dapat menampilkan data-data yang diperlukan yang sesuai dengan batasan-batasan yang telah dipilih.
Hasil yang Didapatkan	Aplikasi dapat menampilkan data-data yang diperlukan yang sesuai dengan batasan-batasan yang telah dipilih.
Status Validitas	Valid

Sumber : Pengujian dan Analisis

s. Kasus Uji Print Log Parkir

Tabel 6.19 Kasus uji untuk mencetak data log parkir

Nama Kasus Uji	Kasus Uji Print Log Parkir
Objek Uji	Kebutuhan Fungsional (SRS_002_07)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional untuk melakukan proses

	pencetakan data log parkir.
Prosedur Uji	1. <i>Administrator</i> menekan tombol print.
Hasil yang Diharapkan	Aplikasi dapat mencetak data-data yang diperlukan yang sesuai dengan data yang telah ditampilkan pada tabel.
Hasil yang Didapatkan	Aplikasi dapat mencetak data-data yang diperlukan yang sesuai dengan data yang telah ditampilkan pada tabel.
Status Validitas	Valid

Sumber : Pengujian dan Analisis

t. Kasus Uji Melaporkan Berita Kehilangan Kendaraan

Tabel 6.20 Kasus uji untuk melaporkan berita kehilangan kendaraan

Nama Kasus Uji	Kasus Uji Melaporkan Berita Kehilangan Kendaraan
Objek Uji	Kebutuhan Fungsional (SRS_002_07)
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional untuk melakukan proses pelaporan berita kehilangan kepada pihak-pihak yang bertanggung jawab dan berwajib.
Prosedur Uji	1. <i>Administrator</i> mengeklik data tabel yang diperlukan di halaman manage log parkir. 2. <i>Administrator</i> menekan tombol laporkan. 3. <i>Administrator</i> memilih apakah format penulisan telah sesuai atau belum.
Hasil yang Diharapkan	Jika <i>Administrator</i> memilih yes, maka aplikasi dapat mengirimkan sms laporan berita kehilangan kendaraan ke nomor telepon yang telah ditentukan. Jika <i>Administrator</i> memilih no, maka aplikasi kembali menampilkan halaman manage log parkir.
Hasil yang Didapatkan	Jika <i>Administrator</i> memilih yes, maka aplikasi dapat mengirimkan sms laporan berita kehilangan kendaraan ke nomor telepon yang telah ditentukan. Jika <i>Administrator</i> memilih no, maka aplikasi kembali menampilkan halaman manage log parkir.
Status Validitas	Valid

Sumber : Pengujian dan Analisis

6.1.1.2 Pembahasan Pengujian Fungsional Berdasarkan Use Case Diagram

Proses pembahasan terhadap hasil pengujian fungsional berdasarkan usecase diagram dilakukan dengan melihat konformitas antara hasil kinerja sistem dengan daftar kebutuhan. Berdasarkan hasil pengujian fungsional berdasarkan use case diagram dapat disimpulkan bahwa implementasi dan fungsionalitas sistem perangkat lunak untuk simulasi keluar masuk dan pembayaran parkir kendaraan berbasis RFID telah memenuhi kebutuhan yang telah dijabarkan pada tahap analisa kebutuhan.

6.1.2 Pengujian Kondisi Gerbang Parkir

Pengujian kondisi gerbang parkir dilakukan untuk mengetahui ketepatan cara kerja subsistem gerbang parkir dari sistem perangkat lunak untuk simulasi keluar masuk dan pembayaran parkir kendaraan berbasis RFID.

6.1.2.1 Hasil Pengujian Kondisi Gerbang Parkir

Tabel 6.21 Hasil pengujian kondisi gerbang parkir

State	Kondisi Tampilan	Pengendara				
		1	2	3	4	5
Portal Idle Mulai	Icon 1 merah	v	v	v	v	v
	Icon 2 hijau	v	v	v	v	v
Membaca RFID tag	RFID di tampilkan	v	v	v	v	v
Portal Proses	Icon 1 merah	v	v	v	v	v
	Icon 2 merah	v	v	v	v	v
Mengambil gambar	Gambar di tampilkan	v	v	v	v	v
Mengolah gambar	Teks plat nomor	v	v	v	v	v
Kesesuaian data teks plat	Perbandingan data teks	v	x	x	v	V
Memasukkan data RFID	Database terisi data baru	v	v	v	v	V
Portal Selesai Proses	Icon 1 hijau	v	v	v	v	V
	Icon 2 merah	v	v	v	v	V
Portal Transisi	Icon 1 merah	v	v	v	v	V
	Icon 2 merah	v	v	v	v	V
Portal Idle Selesai	Icon 1 merah	v	v	v	v	V
	Icon 2 hijau	v	v	v	v	V
Presentase Sukses		100%	93,3%	93,3%	100%	100%

Presentase Gagal	0%	6,7%	6,7%	0%	0%
------------------	----	------	------	----	----

Sumber : Pengujian dan Analisis

6.1.2.2 Pembahasan Pengujian Kondisi Gerbang Parkir

Proses pembahasan terhadap hasil pengujian kondisi (state) gerbang parkir dilakukan dengan melihat ketepatan hasil kinerja sistem untuk aplikasi gerbang parkir dengan daftar kebutuhan. Berdasarkan hasil pengujian kondisi (state) gerbang parkir dapat disimpulkan bahwa implementasi dan fungsionalitas aplikasi gerbang parkir sistem perangkat lunak untuk simulasi keluar masuk dan pembayaran parkir kendaraan berbasis RFID telah memenuhi kebutuhan yang telah dijabarkan pada tahap analisa kebutuhan. Aplikasi dinyatakan memenuhi kebutuhan karena hasil minimum kesuksesan akurasi aplikasi adalah 93,3%. Error yang terjadi 6,7% dalam pengujian dikarenakan ketidakcocokan pendeteksian plat nomor kendaraan dengan plat nomor kendaraan aslinya. Error terjadi karena adanya kemiripan bentuk dari huruf dan angka di plat nomor yang diuji, sehingga aplikasi tidak dapat membedakannya. Contoh kasus error yang terjadi adalah ketika pendeteksian huruf D yang penulisannya tidak menggunakan huruf baku pada plat nomor, sistem akan mengartikan sebagai huruf O.

6.2 Pengujian Performa

Pengujian performa dilakukan untuk mengetahui kecepatan kerja aplikasi gerbang parkir dari sistem perangkat lunak untuk simulasi keluar masuk dan pembayaran parkir kendaraan berbasis RFID. Pengujian ini membandingkan hasil pengukuran waktu proses simulasi gerbang parkir dengan hasil pengukuran waktu proses gerbang parkir pada sistem yang ada sekarang. Skenario pengujian kecepatan kerja simulasi subsistem gerbang parkir (proses masuk sama dengan proses keluar) adalah dengan:

1. Memulai penghitungan waktu.
2. RFID *tag* dikeluarkan dari dalam tas, kemudian mendekatkannya pada *reader*.
3. Menunggu proses kerja aplikasi gerbang parkir hingga warna icon (sebagai portal) berubah menjadi warna semula saat aplikasi pertama dijalankan.
4. Menghentikan perhitungan waktu.

Total rincian perhitungan waktu adalah dengan menambahkan waktu 4 kali gerak portal (1 kali gerak portal = 1 detik), waktu meletakkan RFID *tag* pada *reader*-nya dengan waktu pemrosesan data dalam aplikasi. Pengujian tersebut dilakukan sebanyak 10 kali dengan rincian 5 kali proses masuk dan 5 kali proses keluar dengan rata-rata kecepatan 12,1 detik, seperti yang dapat dilihat dalam Tabel 6.23.

Pengamatan proses kerja gerbang parkir yang sudah diberjalan saat ini dilakukan sebagai perbandingan apakah aplikasi yang baru ini dapat lebih efisien. Pengamatan proses kerja gerbang parkir yang sudah berjalan saat ini dilakukan di 2 mall yang terdapat di Malang, yaitu MOG dan Matos. Skenario proses kerja gerbang parkir masuk MOG dan Matos adalah dengan:

1. Petugas memasukan plat nomor kendaraan dalam komputer.
2. Pengendara membayar tarif parkir serta menunjukan STNK kendaraan (untuk pengendara motor).
3. Petugas memberikan karcis parkir (beserta uang kembalian dan STNK kendaraan untuk pengendara motor).
4. Pengendara memasuki area parkir.

Sedangkan skenario proses kerja gerbang parkir keluar MOG dan Matos adalah dengan:

1. Petugas memasukan plat nomor kendaraan dalam komputer.
2. Pengendara membayar tarif parkir (untuk pengendara mobil).
3. Petugas memberikan uang kembalian (untuk pengendara mobil).
4. Petugas membuka portal.
5. Pengendara meninggalkan area parkir.

Hasil pengamatan proses kerja gerbang parkir MOG dan Matos dapat dilihat pada Tabel 6.24.

6.2.1 Hasil Pengujian Performa

Tabel 6.22 Hasil pengujian proses kerja subsistem gerbang parkir

Pengendara	Proses Masuk (detik)	Proses Keluar (detik)
1	14	11
2	12	12

3	12	13	
4	11	11	
5	13	12	
Rata-rata	12.4	11.8	12.1

Sumber : Pengujian dan Analisis

Tabel 6.23 Hasil pengamatan proses kerja gerbang parkir MOG dan Matos

Kendaraan	Proses Masuk (detik)				Proses Keluar (detik)				
	Motor		Mobil		Motor		Mobil		
	MOG	Matos	MOG	Matos	MOG	Matos	MOG	Matos	
1	28	11	3	7	5	19	10	24	
2	11	12	16	8	2	11	12	29	
3	18	39	4	7	5	14	17	20	
4	15	17	11	10	3	10	7	16	
5	12	18	7	9	6	13	15	10	
6	6	18	7	12	7	8	7	13	
7	17	14	7	10	5	8	14	12	
8	22	16	7	8	33	18	14	17	
9	10	11	8	12	7	13	13	36	
10	71	30	7	10	8	10	42	17	
11	13	12	7	7	5	14	9	10	
12	13	19	12	8	8	14	7	21	
13	12	11	11	10	7	26	18	12	
14	16	12	10	8	7	11	8	37	
15	17	18	7	7	16	11	15	15	
16	6	34	17	9	4	7	14	11	
17	13	12	13	13	12	13	23	17	
18	22	12	8	9	21	8	18	17	
19	9	14	8	8	7	8	25	14	
20	7	10	9	8	5	11	15	12	
Rata-rata	16.9	17	8.95	9	8.65	12.35	15.15	18	13.25

Sumber : Pengujian dan Analisis

6.2.2 Pembahasan Pengujian Performa

Proses pembahasan terhadap hasil pengujian performa dilakukan dengan melihat kecepatan hasil kinerja sistem untuk aplikasi gerbang parkir dengan daftar kebutuhan. Berdasarkan hasil pengujian performa dapat disimpulkan bahwa implementasi dan non-fungsionalitas sistem perangkat lunak untuk simulasi keluar masuk dan pembayaran parkir kendaraan berbasis RFID telah memenuhi kebutuhan yang telah dijabarkan pada tahap analisa kebutuhan. Aplikasi dinyatakan memenuhi kebutuhan karena rata-rata waktu kerja aplikasi lebih cepat bila dibandingkan dengan waktu kerja gerbang parkir saat ini, dengan selisih waktu 1,15 detik. Waktu proses yang lama pada waktu kerja parkir yang ada pada saat ini disebabkan oleh beberapa kendala, yaitu:

1. Tidak tersedianya cukup uang untuk mengembalikan kelebihan tarif parkir yang diberikan oleh pengemudi.
2. Keterbatasan kemampuan petugas parkir dalam melakukan proses mekanisme parkir bilamana dalam kondisi kendaraan ramai. Hal ini dikarenakan ketika kondisi keadaran sepi, petugas parkir dapat mencatat plat nomor kendaraan sejak kendaraan masih berjarak cukup jauh sehingga pencatatan menjadi lebih cepat. Ketika kondisi kendaraan ramai, petugas parkir mencatat plat nomor kendaraan ketika kendaraan telah sampai pada pos parkir, sehingga pencatatan menjadi lebih lambat.

BAB VII

PENUTUP

7.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi dan pengujian yang dilakukan, maka diambil kesimpulan sebagai berikut :

1. Subsistem gerbang parkir sistem perangkat lunak untuk simulasi keluar masuk dan pembayaran parkir kendaraan berbasis RFID dapat diimplementasikan dalam bentuk simulasi yang berguna untuk mengatur proses keluar masuk kendaraan dengan membaca RFID dan mengolah data gambar kendaraan menjadi data teks plat nomor kendaraan dengan menggunakan metode identifikasi otomatis.
2. Subsistem administrasi sistem perangkat lunak untuk simulasi keluar masuk dan pembayaran parkir kendaraan berbasis RFID dapat digunakan untuk mengelola elemen-elemen sistem perangkat lunak untuk simulasi keluar masuk dan pembayaran parkir kendaraan berbasis RFID.
3. Pengembangan proses perangkat lunak sistem dapat dikembangkan dengan metode pengembangan perangkat lunak SDLC menggunakan model *waterfall* karena pengaplikasian menggunakan model ini mudah, semua kebutuhan sistem dapat didefinisikan secara utuh dan eksplisit.
4. Pengujian kondisi (state) gerbang parkir sistem menunjukkan bahwa aplikasi gerbang parkir dapat berjalan sesuai rancangan yang telah dibuat dengan presentasi minimum kesuksesan 93,3%.
5. Pengujian performa sistem menunjukkan bahwa aplikasi gerbang parkir dapat berjalan lebih cepat 1,15 detik dari sistem gerbang parkir yang telah berjalan saat ini.

7.2 Saran

Saran yang dapat diberikan untuk pengembangan perangkat lunak ini antara lain :

1. Untuk pengembangan lebih lanjut sebaiknya untuk pengenalan plat nomor menggunakan dengan metode *Connected Component Analysis* yaitu pengenalan plat nomor kendaraan tanpa adanya proses pencarian lokasi plat, namun langsung mencari karakter pada plat, jadi tidak memakan waktu cukup lama lagi untuk menentukan lokasi plat nomornya terlebih dahulu.
2. Untuk pengembangan lebih lanjut sebaiknya ditambahkan fitur yang dapat menahan kendaraan agar tidak dapat meninggalkan area parkir dalam kurun waktu tertentu, sehingga pemilik kendaraan dapat merasa aman bila meninggalkan kendaraannya di area parkir lebih dari 1 hari.



DAFTAR PUSTAKA

- [APR-12] Aprizal, A. (2012). *Fungsi Hash*. Diperoleh dari http://www.academia.edu/1528069/Keamanan_Komputer
- [BOO-98] Booch, G. (1998). *The Unified Modeling Language User Guide* (1st ed.). New York: Addison-Wesley Longman Inc.
- [DEP-96] Departemen Perhubungan. (1996). *Pedoman Teknis Penyelenggaraan Fasilitas Parkir*. Diperoleh dari <http://bstp.hubdat.web.id/data/arsip/parkir.pdf>
- [FIN-03] Finkenzeller, K. (2003). *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification* (2nd ed.). London: John Wiley & Sons.
- [HAM-10] Hamid. (2010). *Bidang Komunikasi Data & Jaringan: Pengembangan Sistem Parkir Terkomputerisasi dengan Otomatisasi Pembiayaan dan Penggunaan RFID Sebagai Pengenal Unik Pengguna*, Seminar Nasional Aplikasi Teknologi Informasi. Universitas Islam Indonesia. Diperoleh dari <http://journal.uui.ac.id/index.php/Snati/article/view/1974/1747>
- [KHO-11] Khoiri. (2011). *Waterfall Model*. Diperoleh dari <http://materikulia-q.blogspot.com/2011/04/waterfall-model.html>
- [LAN-05] Landt, J. (2005). The History of RFID. *IEEE Potentials*, 24(4), 8-11.
- [LIN-99] Lindholm, T. (1999). *The Java™ Virtual Machine Specification* (2nd ed.). Addison-Wesley Professional.
- [MUL-13] Mulyadi, Yadi. (2013). *Java*. Diperoleh dari <http://kliks.heck.in/sistem-java.xhtml>
- [NAT-06] Nath, B. (2006). RFID Technology and Applications. *IEEE Pervasive Computing*, 5(1), 22-24.
- [PRE-01] Pressman, R. (2001). *Software Engineering : A Practitioner's Approach, Fifth Edition*. McGraw Hill.
- [SIA-13] SIA. (2013). DKT Software. Diperoleh dari <http://www.dtksoft.com/dtkanpr.php>

- [SOL-12] Solms, F. (2012). *Object-Oriented Analysis and Design using UML*. Diperoleh dari <http://www.fritsolms.net/sites/default/files/documents/ObjectOrientedAnalysisAndDesignUsingUML.pdf>
- [SOM-11] Sommerville, Ian. (2011). *Software Engineering* (9th ed.). United States of America: Addison-Wesley Pearson Education Inc.
- [SRI-05] Srivastava, L. (2005). *Ubiquitous Network Societies: The Case of Radio Frequency Identification*, ITU Workshop On UNS. Geneva.
- [WAN-04] Want, R. (2004). Enabling Ubiquitous Sensing with RFID. *Computer*, 37(4), 84–86.
- [WEI-05] Weinstein, R. (2005). RFID: A Technical Overview and Its Application to the Enterprise. *IT Professional*, 7(3), 27-33.



LAMPIRAN

Lampiran 1 Hasil Survey

Orang ke-	Waktu yang diperlukan untuk proses masuk dan keluar gerbang parkir		Cara Pembayaran Parkir yang Diinginkan		Sistem Perparkiran yang Dipilih	
	Lama	cepat	card	cash	otomatis	manual
1	V		V		V	
2	V		V		V	
3	V		V		V	
4	V		V		V	
5		V		V		V
6		V		V		V
7	V		V		V	
8	V		V		V	
9	V		V		V	
10	V		V		V	

Sumber : Pendahuluan



Lampiran 2 Deskripsi Dari Kelas Diagram

1. Class ManageGbg

Nama Class : ManageGbg
Deskripsi : <i>Class ini berfungsi untuk mengkoneksikan dengan database dan mengatur segala kegiatan yang berhubungan dengan database yang akan digunakan oleh class yang lainnya.</i>
Nama Method : ManageGbg() Fungsi : <i>Method ini berfungsi sebagai constructor dari class ManageGbg.</i>
Nama Method : cekKoneksi() Fungsi : <i>Method ini berfungsi untuk mengecek apakah koneksi ke database telah berhasil atau tidak.</i>
Nama Method : openKoneksi() Fungsi : <i>Method ini berfungsi untuk membuka koneksi setiap ada eksekusi query.</i>
Nama Method : queryStoreProcedure(java.lang.String sqlQuery) Fungsi : <i>Method ini berfungsi untuk mengirimkan query yang akan diproses oleh store procedure.</i>

Sumber : Perancangan

2. Class Processing

Nama Class : Processing
Deskripsi : <i>Class ini merupakan proses utama dalam aplikasi gerbang parkir. Class Preprocessing berfungsi untuk memproses pengaturan kerja gerbang parkir.</i>
Nama Method : Processing() Fungsi : <i>Method ini berfungsi sebagai constructor dari class Processing.</i>
Nama Method : jTextField2ActionPerformed(java.awt.event.ActionEvent evt) Fungsi : <i>Method ini berfungsi untuk mengambil RFID yang terdeteksi lalu melanjutkan ke proses pengambilan gambar plat nomor dan memasukkan data RFID dan data hasil pemrosesan gambar plat nomor ke database.</i>
Nama Method : jButton3ActionPerformed(java.awt.event.ActionEvent evt) Fungsi : <i>Method ini berfungsi untuk menampilkan live camera.</i>

Nama Method : main(java.lang.String[] args)

Fungsi :

Method ini berfungsi untuk menampilkan tampilan class ini.

Sumber : Perancangan

3. Class ViewCameraFrame

Nama Class : ViewCameraFrame

Deskripsi :

Class ini merupakan class untuk menampilkan live camera.

Nama Method : ViewCameraFrame()

Fungsi :

Method ini berfungsi sebagai constructor dari class ViewCameraFrame.

Nama Method : main(java.lang.String[] args)

Fungsi :

Method ini berfungsi untuk menampilkan tampilan class ini.

Sumber : Perancangan

4. Class Detection

Nama Class : Detection

Deskripsi :

Class ini merupakan class yang berfungsi untuk memproses pengolahan gambar plat nomor menjadi data teks.

Nama Method : Detection()

Fungsi :

Method ini berfungsi sebagai constructor dari class Detection.

Nama Method : kenali()

Fungsi :

Method ini berfungsi untuk mengontrol pendeteksian lalu mengembalikan hasil dari pendeteksian.

Sumber : Perancangan

5. Class ManageAdm

Nama Class : ManageAdm

Deskripsi :

Class ini berfungsi untuk mengkoneksikan dengan database dan mengatur segala kegiatan yang

berhubungan dengan database yang akan digunakan oleh <i>class</i> yang lainnya.
Nama Method : ManageAdm() Fungsi : <i>Method</i> ini berfungsi sebagai <i>constructor</i> dari <i>class</i> ManageAdm.
Nama Method : cekKoneksi() Fungsi : <i>Method</i> ini berfungsi untuk mengecek apakah koneksi ke database telah berhasil atau tidak.
Nama Method : openKoneksi() Fungsi : <i>Method</i> ini berfungsi untuk membuka koneksi setiap ada eksekusi query.
Nama Method : querySelect(java.lang.String sqlQuery) Fungsi : <i>Method</i> ini berfungsi untuk memproses query yang bertujuan untuk mengambil data lengkap dari database dengan informasi yang dipilih.
Nama Method : querySelectMatrix(java.lang.String sqlQuery) Fungsi : <i>Method</i> ini berfungsi untuk memproses query yang membutuhkan nilai kembalian atau hanya berfungsi untuk meng- <i>input</i> -kan data ke database.
Nama Method : queryStoreProcedure(java.lang.String sqlQuery) Fungsi : <i>Method</i> ini berfungsi untuk mengirimkan query yang akan diproses oleh store procedure.
Nama Method : queryUpdate(java.lang.String sqlQuery) Fungsi : <i>Method</i> ini berfungsi untuk memproses query yang tidak membutuhkan nilai kembalian atau hanya berfungsi untuk <i>insert</i> , <i>update</i> dan <i>delete</i> data ke database.
Nama Method : tampil_tabel(java.lang.String sqlQuery) Fungsi : <i>Method</i> ini berfungsi untuk memproses query untuk menampilkan keseluruhan data yang dibutuhkan.

Sumber : Perancangan

6. Class Login

Nama Class : Login
Deskripsi : <i>Class</i> ini berfungsi untuk menyeleksi hanya admin yang terdaftar saja yang bisa menggunakan proses selanjutnya.
Nama Method : Login()

Fungsi : <i>Method</i> ini berfungsi sebagai <i>constructor</i> dari <i>class</i> Login.
Nama Method : jButton2ActionPerformed(java.awt.event.ActionEvent evt) Fungsi : <i>Method</i> ini berfungsi untuk memproses penyeleksian admin yang diperbolehkan melanjutkan proses selanjutnya adalah admin yang terdaftar.
Nama Method : main(java.lang.String[] args) Fungsi : <i>Method</i> ini berfungsi untuk menampilkan tampilan <i>class</i> ini.

Sumber : Perancangan

7. Class Form

Nama Class : Form
Deskripsi : <i>Class</i> ini berfungsi sebagai tempat utama untuk menampung akses ke menu-menu yang disediakan.
Nama Method : Form() Fungsi : <i>Method</i> ini berfungsi sebagai <i>constructor</i> dari <i>class</i> Form.
Nama Method : jMenuItem2MouseClicked(java.awt.event.MouseEvent evt) Fungsi : <i>Method</i> ini berfungsi untuk menampilkan form manage admin.
Nama Method : jMenuItem3MouseClicked(java.awt.event.MouseEvent evt) Fungsi : <i>Method</i> ini berfungsi untuk menampilkan form manage pengendara.
Nama Method : jMenuItem4MouseClicked(java.awt.event.MouseEvent evt) Fungsi : <i>Method</i> ini berfungsi untuk menampilkan form manage kendaraan.
Nama Method : jMenuItem5MouseClicked(java.awt.event.MouseEvent evt) Fungsi : <i>Method</i> ini berfungsi untuk menampilkan form manage logparkir.
Nama Method : jMenuItem6MouseClicked(java.awt.event.MouseEvent evt) Fungsi : <i>Method</i> ini berfungsi untuk menampilkan form manage saldo.
Nama Method : jMenuItem7MouseClicked(java.awt.event.MouseEvent evt) Fungsi : <i>Method</i> ini berfungsi untuk menampilkan form setting.

Nama Method : removeAllPanel()

Fungsi :

Method ini berfungsi untuk menghapus semua tampilan form yang pernah dibuka.

Nama Method : main(java.lang.String[] args)

Fungsi :

Method ini berfungsi untuk menampilkan tampilan class ini.

Sumber : Perancangan

8. Class ManageAdmin

Nama Class : ManageAdmin

Deskripsi :

Class ini berfungsi sebagai tempat untuk mengelolah data admin.

Nama Method : ManageAdmin()

Fungsi :

Method ini berfungsi sebagai constructor dari class ManageAdmin.

Nama Method : jButton1ActionPerformed(java.awt.event.ActionEvent evt)

Fungsi :

Method ini berfungsi untuk memproses memasukan data ke database.

Nama Method : jButton2ActionPerformed(java.awt.event.ActionEvent evt)

Fungsi :

Method ini berfungsi untuk memproses meng-update data ke database.

Nama Method : jButton3ActionPerformed(java.awt.event.ActionEvent evt)

Fungsi :

Method ini berfungsi untuk memproses menghapus data ke database.

Nama Method : jTable1MouseClicked(java.awt.event.MouseEvent evt)

Fungsi :

Method ini berfungsi untuk menampilkan isi tabel admin dari database.

Sumber : Perancangan

9. Class ManageKendaraan

Nama Class : ManageKendaraan

Deskripsi :

Class ini berfungsi sebagai tempat untuk mengelolah data kendaraan.

Nama Method : ManageKendaraan()

Fungsi :

Method ini berfungsi sebagai constructor dari class ManageKendaraan.

Nama Method : jButton1ActionPerformed(java.awt.event.ActionEvent evt) Fungsi : <i>Method ini berfungsi untuk memproses memasukan data ke database.</i>
Nama Method : jButton2ActionPerformed(java.awt.event.ActionEvent evt) Fungsi : <i>Method ini berfungsi untuk memproses meng-update data ke database.</i>
Nama Method : jButton3ActionPerformed(java.awt.event.ActionEvent evt) Fungsi : <i>Method ini berfungsi untuk memproses menghapus data ke database.</i>
Nama Method : jTable1MouseClicked(java.awt.event.MouseEvent evt) Fungsi : <i>Method ini berfungsi untuk menampilkan isi tabel kendaraan dari database.</i>

Sumber : Perancangan

10. Class ManagePengendara

Nama Class : ManagePengendara
Deskripsi : <i>Class ini berfungsi sebagai tempat untuk mengelolah data pengendara.</i>
Nama Method : ManagePengendara() Fungsi : <i>Method ini berfungsi sebagai constructor dari class ManagePengendara.</i>
Nama Method : jButton1ActionPerformed(java.awt.event.ActionEvent evt) Fungsi : <i>Method ini berfungsi untuk memproses memasukan data ke database.</i>
Nama Method : jButton2ActionPerformed(java.awt.event.ActionEvent evt) Fungsi : <i>Method ini berfungsi untuk memproses meng-update data ke database.</i>
Nama Method : jButton3ActionPerformed(java.awt.event.ActionEvent evt) Fungsi : <i>Method ini berfungsi untuk memproses menghapus data ke database.</i>
Nama Method : jTable1MouseClicked(java.awt.event.MouseEvent evt) Fungsi : <i>Method ini berfungsi untuk menampilkan isi tabel pengendara dari database.</i>
Nama Method : jTextField1ActionPerformed(java.awt.event.ActionEvent evt) Fungsi : <i>Method ini berfungsi untuk memindahkan focus kursor setelah field RFID terisi 10 digit.</i>

Sumber : Perancangan

11. Class ManageLogParkir

Nama Class : ManageLogParkir
Deskripsi : <i>Class ini berfungsi sebagai tempat untuk mengelolah data log parkir.</i>
Nama Method : ManageLogParkir() Fungsi : <i>Method ini berfungsi sebagai constructor dari class ManageLogParkir.</i>
Nama Method : jButton3ActionPerformed(java.awt.event.ActionEvent evt) Fungsi : <i>Method ini berfungsi untuk melakukan pencetakan data yang ditampilkan di tabel.</i>
Nama Method : jTable1MouseClicked(java.awt.event.MouseEvent evt) Fungsi : <i>Method ini berfungsi untuk menampilkan isi tabel logparkir dari database.</i>
Nama Method : jButton4ActionPerformed(java.awt.event.ActionEvent evt) Fungsi : <i>Method ini berfungsi untuk memproses pencarian data sesuai dengan keinginan admin.</i>
Nama Method : jButton1ActionPerformed(java.awt.event.ActionEvent evt) Fungsi : <i>Method ini berfungsi untuk melakukan pelaporan kendaraan kehilangan ke nomor yang telah tersimpan dalam database.</i>
Nama Method : jRadioButton1ActionPerformed(java.awt.event.ActionEvent evt) Fungsi : <i>Method ini berfungsi untuk melakukan pencarian secara keseluruhan data yang tersimpan dalam tabel logparkir di database.</i>
Nama Method : jRadioButton2ActionPerformed(java.awt.event.ActionEvent evt) Fungsi : <i>Method ini berfungsi untuk melakukan pencarian berdasarkan waktu atau nomor polisi yang ditentukan admin.</i>
Nama Method : jCheckBox1ActionPerformed(java.awt.event.ActionEvent evt) Fungsi : <i>Method ini berfungsi untuk melakukan pencarian berdasarkan waktu yang ditentukan admin.</i>
Nama Method : jCheckBox2ActionPerformed(java.awt.event.ActionEvent evt) Fungsi : <i>Method ini berfungsi untuk melakukan pencarian berdasarkan plat nomor yang ditentukan admin.</i>
Nama Method : jRadioButton3ActionPerformed(java.awt.event.ActionEvent evt) Fungsi :

Method ini berfungsi untuk melakukan pencarian berdasarkan RFID yang ditentukan admin.

Sumber : Perancangan

12. Class ManageSaldo

Nama Class : ManageSaldo

Deskripsi :

Class ini berfungsi sebagai tempat untuk mengelolah data saldo.

Nama Method : ManageSaldo()

Fungsi :

Method ini berfungsi sebagai constructor dari class ManageSaldo.

Nama Method : jTable1MouseClicked(java.awt.event.MouseEvent evt)

Fungsi :

Method ini berfungsi untuk menampilkan isi tabel saldo dari database.

Nama Method : jButton1ActionPerformed(java.awt.event.ActionEvent evt)

Fungsi :

Method ini berfungsi untuk mengirim informasi sisa saldo ke nomor hp pelanggan yang telah dipilih oleh admin.

Nama Method : jButton2ActionPerformed(java.awt.event.ActionEvent evt)

Fungsi :

Method ini berfungsi untuk mengirim informasi sisa saldo ke semua nomor hp pelanggan yang ada dalam database.

Nama Method : jButton3ActionPerformed(java.awt.event.ActionEvent evt)

Fungsi :

Method ini berfungsi untuk menambahkan saldo pelanggan yang telah dipilih dan di-update ke database.

Nama Method : jButton4ActionPerformed(java.awt.event.ActionEvent evt)

Fungsi :

Method ini berfungsi untuk mencetak history transaksi berdasarkan RFID pelanggan yang dipilih oleh admin.

Sumber : Perancangan

13. Class Setting

Nama Class : Setting

Deskripsi :

Class ini berfungsi sebagai tempat untuk mengelolah data setting.

Nama Method : Setting()

Fungsi : <i>Method</i> ini berfungsi sebagai <i>constructor</i> dari <i>class</i> Setting.
Nama Method : jTable1MouseClicked(java.awt.event.MouseEvent evt) Fungsi : <i>Method</i> ini berfungsi untuk menampilkan isi tabel setting dari database.
Nama Method : jButton1ActionPerformed(java.awt.event.ActionEvent evt) Fungsi : <i>Method</i> ini berfungsi untuk melakukan <i>update</i> saldo pelanggan yang telah dipilih oleh admin.

Sumber : Perancangan

14. Class SmsPeringatan

Nama Class : SmsPeringatan
Deskripsi : <i>Class</i> ini berfungsi untuk mengirim sms ke semua pelanggan yang saldonya sama dengan atau dibawah Rp 0.
Nama Method : SmsPeringatan(String WaktuKirimSMSSaldo) Fungsi : <i>Method</i> ini berfungsi sebagai <i>constructor</i> dari <i>class</i> SmsPeringatan.
Nama Method : kirimSMSAll() Fungsi : <i>Method</i> ini berfungsi untuk mengirimkan sms ke semua pelanggan yang memiliki saldo kurang dari sama dengan Rp 0.
Nama Method : run() Fungsi : <i>Method</i> ini berfungsi sebagai <i>method</i> utama yang akan dijalankan pertama kali oleh <i>class</i> ini untuk mengecek waktu pengiriman dan menjalankan pengiriman sms peringatan.

Sumber : Perancangan

15. Class Sender

Nama Class : Sender
Deskripsi : <i>Class</i> ini berfungsi untuk mengirim sms.
Nama Method : Sender() Fungsi : <i>Method</i> ini berfungsi sebagai <i>constructor</i> dari <i>class</i> Sender.
Nama Method : Sender(String recipient, String message)

Fungsi :

Method ini berfungsi sebagai *constructor* dari *class* Sender.

Nama Method : send ()**Fungsi :**

Method ini berfungsi untuk mengatur settingan pengiriman.

Nama Method : run()**Fungsi :**

Method ini berfungsi sebagai *method* utama yang akan dijalankan pertama kali oleh *class* ini untuk menjalankan proses pengiriman.

Nama Method : log(String s)**Fungsi :**

Method ini berfungsi untuk mencetak status kerja kirim sms pada *compiler*.

Nama Method : getSMS()**Fungsi :**

Method ini berfungsi untuk mengambil data sms.

Sumber : Perancangan

16. Class SMSClient

Nama Class : SMSClient**Deskripsi :**

Class ini berfungsi sebagai proses pengiriman sms.

Nama Method : SMSClient(int mode)**Fungsi :**

Method ini berfungsi sebagai *constructor* dari *class* SMSClient.

Nama Method : getSMS()**Fungsi :**

Method ini berfungsi untuk mengambil data sms.

Nama Method : sendMessage (String recipient, String message)**Fungsi :**

Method ini berfungsi sebagai langkah awal pengiriman sms.

Nama Method : run()**Fungsi :**

Method ini berfungsi sebagai *method* utama yang akan dijalankan pertama kali oleh *class* ini untuk menjalankan proses pengiriman.

Sumber : Perancangan

17. Class SerialConnection

Nama Class : SerialConnection
Deskripsi : <i>Class ini berfungsi untuk mengatur pengaturan pengiriman sms.</i>
Nama Method : SerialConnection(SerialParameters parameters) Fungsi : <i>Method ini berfungsi sebagai constructor dari class SerialConnection.</i>
Nama Method : openConnection() Fungsi : <i>Method ini berfungsi untuk membuka koneksi setiap kali memulai pengiriman sms.</i>
Nama Method : setConnectionParameters() Fungsi : <i>Method ini berfungsi untuk menetapkan setting-an koneksi pengiriman sms.</i>
Nama Method : closeConnection() Fungsi : <i>Method ini berfungsi untuk menutup koneksi setiap kali selesai pengiriman sms.</i>
Nama Method : sendBreak() Fungsi : <i>Method ini berfungsi untuk memberikan 1 detik waktu henti.</i>
Nama Method : isOpen() Fungsi : <i>Method ini berfungsi untuk memberikan laporan status dari port.</i>
Nama Method : serialEvent(SerialPortEvent e) Fungsi : <i>Method ini berfungsi untuk menerima input-an data.</i>
Nama Method : send(String message) Fungsi : <i>Method ini berfungsi untuk mengirim sms.</i>

Sumber : Perancangan

18. Class SerialParameters

Nama Class : SerialParameters
Deskripsi : <i>Class ini berfungsi untuk menyimpan parameter untuk port serial.</i>
Nama Method : SerialParameters() Fungsi :

<i>Method ini berfungsi sebagai constructor dari class SerialParameters.</i>
Nama Method : SerialParameters(String portName, int baudRate, int flowControlIn, int flowControlOut, int databits, int stopbits, int parity) Fungsi : <i>Method ini berfungsi sebagai constructor dari class SerialParameters.</i>
Nama Method : getBaudRate() Fungsi : <i>Method ini berfungsi untuk mengambil baud rate dalam integer.</i>
Nama Method : getBaudRateString() Fungsi : <i>Method ini berfungsi untuk mengambil baud rate dalam String.</i>
Nama Method : getDatabits() Fungsi : <i>Method ini berfungsi untuk mengambil data bits dalam integer.</i>
Nama Method : getDatabitsString() Fungsi : <i>Method ini berfungsi untuk mengambil data bits dalam String.</i>
Nama Method : getFlowControlIn() Fungsi : <i>Method ini berfungsi untuk mengambil alur kontrol untuk membaca dalam integer.</i>
Nama Method : getFlowControlInString() Fungsi : <i>Method ini berfungsi untuk mengambil alur kontrol untuk membaca dalam String.</i>
Nama Method : getFlowControlOut() Fungsi : <i>Method ini berfungsi untuk mengambil alur kontrol untuk menulis dalam integer.</i>
Nama Method : getFlowControlOutString() Fungsi : <i>Method ini berfungsi untuk mengambil alur kontrol untuk menulis dalam String.</i>
Nama Method : getParity() Fungsi : <i>Method ini berfungsi untuk mengambil pengaturan paritas dalam integer.</i>
Nama Method : getParityString() Fungsi : <i>Method ini berfungsi untuk mengambil pengaturan paritas dalam String.</i>
Nama Method : getPortName() Fungsi :

<i>Method ini berfungsi untuk mengambil nama port.</i>
Nama Method : getStopbits() Fungsi : <i>Method ini berfungsi untuk mengambil pengaturan penghentian bits dalam integer.</i>
Nama Method : getStopbitsString() Fungsi : <i>Method ini berfungsi untuk mengambil pengaturan penghentian bits dalam String.</i>
Nama Method : setBaudRate(int baudRate) Fungsi : <i>Method ini berfungsi untuk mengatur baud rate.</i>
Nama Method : setBaudRate(java.lang.String baudRate) Fungsi : <i>Method ini berfungsi untuk mengatur baud rate.</i>
Nama Method : setDatabits(int databits) Fungsi : <i>Method ini berfungsi untuk mengatur data bits.</i>
Nama Method : setDatabits(java.lang.String databits) Fungsi : <i>Method ini berfungsi untuk mengatur data bits.</i>
Nama Method : setFlowControlIn(int flowControlIn) Fungsi : <i>Method ini berfungsi untuk mengatur alur kontrol untuk dibaca.</i>
Nama Method : setFlowControlIn(java.lang.String flowControlIn) Fungsi : <i>Method ini berfungsi untuk mengatur alur kontrol untuk dibaca.</i>
Nama Method : setFlowControlOut(int flowControlOut) Fungsi : <i>Method ini berfungsi untuk mengatur alur kontrol untuk ditulis.</i>
Nama Method : setFlowControlOut(java.lang.String flowControlOut) Fungsi : <i>Method ini berfungsi untuk mengatur alur kontrol untuk ditulis.</i>
Nama Method : setParity(int parity) Fungsi : <i>Method ini berfungsi untuk mengatur pengaturan paritas.</i>
Nama Method : setParity(java.lang.String parity) Fungsi : <i>Method ini berfungsi untuk mengatur pengaturan paritas.</i>

Nama Method : setPortName(java.lang.String portName)

Fungsi :

Method ini berfungsi untuk mengatur nama port.

Nama Method : setStopbits(int stopbits)

Fungsi :

Method ini berfungsi untuk mengatur pemberhentian bits.

Nama Method : setStopbits(java.lang.String stopbits)

Fungsi :

Method ini berfungsi untuk mengatur pemberhentian bits.

Sumber : Perancangan

