# RANCANG BANGUN SISTEM PENGELOLAAN PARKIR MENGGUNAKAN PERANGKAT NFC BERBASIS ANDROID

## **SKRIPSI**

Untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Komputer

Disusun oleh: Hidayat Akbar

NIM: 115060800111113



PROGRAM STUDI INFORMATIKA/ILMU KOMPUTER
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2016

## **PENGESAHAN**

# RANCANG BANGUNG SISTEM PENGELOLAAN PARKIR MENGGUNAKAN PERANGKAT NFC BERBASIS ANDROID

**SKRIPSI** 

Diajukan untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Komputer

Disusun Oleh : Hidayat Akbar NIM: 115060800111113

Skripsi ini telah diuji dan dinyatakan lulus pada 15 Januari 2016 Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Eriq M. Adams Jonemaro, S.T., M.Kom. NIP: 19850410 201212 1 001 Denny Sagita R, S.Kom., M.Kom. NIP: 19851124 201504 1 001

Mengetahui Ketua Program Studi Informatika/Ilmu Komputer

> <u>Drs. Marji, M.T.</u> NIP: 19670801 199203 1 001

## PERNYATAAN ORISINALITAS

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 disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsurunsur 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, 20 Januari 2016

**Hidayat Akbar** 

NIM: 115060800111113



## KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT, karena hanya dengan rahmat dan bimbingannya penulis dapat menyelesaikan tugas akhir dengan judul "Rancang Bangun Sistem Pengelolaan Parkir Menggunakan Perangkat NFC Berbasis Android". Tanpa rahmat dan bimbingan dari Allah SWT, maka niscaya penulis tidak akan dapat menyelesaikan tugas akhir ini dengan baik.

Penulis menyadari bahwa skripsi ini tidak akan berhasil tanpa bantuan dari beberapa pihak. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan terima kasih kepada:

- 1. Bapak Eriq Muh. Adams Jonemaro, S.T., M.Kom. selaku dosen pembimbing I yang telah memberikan ilmu dan saran selama penyusunan skripsi ini.
- 2. Bapak Denny Sagita Rusdianto, S.Kom., M.Kom. selaku dosen pembimbing II dan dosen pembimbing akademik yang juga memberikan ilmu dan saran selama perkuliahan dan penyusunan skripsi ini.
- 3. Kedua orang tua penulis yakni Alm. Abdullah Layem dan Lies Maria, kakak Rodiyatul Amaliyah, Abdul Hamid, Ahmad Kholil, Muhammad, Hamidah Andiyani, dan seluruh keluarga besar penulis atas segala nasihat, kasih sayang, perhatian dan kesabarannya di dalam membesarkan dan mendidik penulis, serta yang senantiasa tiada henti-hentinya memberikan doa dan semangat demi terselesaikannya skripsi ini.
  - 4. Bapak Jauhari selaku Kepala Divisi Parkir Universitas Brawijaya.
  - 5. Bapak Drs. Marji, M.T. selaku ketua Program Studi Informatika.
- 6. Segenap bapak dan ibu dosen yang telah mendidik dan mengajarkan ilmunya kepada penulis selama menempuh pendidikan di FILKOM UB.
- 7. Teman-teman Cuntel Reborn, IF-H 2011, dan IF 2011 yang mendukung dan membantu penulis dalam perkuliahan serta pengerjaan tugas akhir ini.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih banyak kekurangan, sehingga saran dan kritik yang membangun sangat penulis harapkan. Akhir kata penulis berharap skripsi ini dapat membawa manfaat bagi semua pihak yang menggunakannya.

Malang, 20 Januari 2016

**Penulis** 

hidaakbar@gmail.com

## **ABSTRAK**

Area parkir pada Universitas Brawijaya terdapat 48 area parkir. Masingmasing area parkir dijaga oleh minimal 2 petugas parkir dengan 2 kali shift dalam sehari. Dari 48 area parkir tersebut dapat menampung sekitar 18110 kendaraan baik itu roda dua maupun roda empat. Dalam tiap area parkir tersebut dilengkapi dengan kartu parkir bagi pengguna area parkir yang disesuaikan dengan kapasitas tiap area parkir. Untuk keluar-masuk area parkir diperlukan kartu parkir dan pengecekan STNK. Permasalahan yang muncul adalah ketika jumlah kartu yang tersedia tidak seimbang dengan jumlah kendaraan yang masuk pada area parkir. Hal ini membuat petugas parkir membolehkan kendaraan masuk tanpa kartu parkir. Selain itu juga kartu parkir seringkali tidak diserahkan oleh pengendara pada petugas parkir sehingga kartu yang tersedia semakin berkurang.

Dari paparan permasalahan tersebut, dibutuhkan sebuah sistem pengeloaan parkir yang membantu pengelolaan parkir menjadi lebih optimal. Sistem yang dibuat dalam skripsi ini adalah sistem pengelolaan parkir menggunakan perangkat NFC berbasis android. Sistem ini menawarkan kemudahan dalam aktivitas parkir karena memakai kartu NFC yang difungsikan sebagai identitas pemakai sistem. Sistem ini mengusahakan permasalahan-permasalahan yang sering terjadi dalam sistem perparkiran manual dapat teratasi tanpa mengabaikan aspek-aspek lain seperti keamanan, integritas tetap terjaga bahkan lebih optimal bila dibandingkan dengan sistem perparkiran manual.

Kata kunci: pengelolaan parkir, NFC, android, kartu cerdas

## **ABSTRACT**

In Brawijaya University there are 48 parking lot areas. Each parking lot guarded by at least two personnel with two shift a day. All parking lot in UB can accommodate 18110 vehicles both car and motorcycle. Each parking lot have parking card adjusted by capacity of vehicles can accommodate. For activities in parking lot such as get in or get out must required parking card and checking STNK. The problem is when parking card avaibility does not equal vehicles who entering parking lot. This make parking guard allow vehicles to get in parking lot without parking card. In addition, parking cards are often not handed over by motorists on parking guard so that the parking card avaibility is on the wane.

Based from these problems, we need a parking management system which enables parking management become more optimal. The system made in this research is a parking management system using NFC devices based on Android.. This system offers the ease of parking activity for wearing the NFC card that functioned as the identity of users of the system. These systems exploit the problems that often occur in manual parking system can be resolved without neglecting other aspects such as security, integrity is maintained even more optimal when compared with manual parking system.

Keywords: parking management, NFC, android, smartcard

# **DAFTAR ISI**

	SAHAN	
	ATAAN ORISINALITAS	
	PENGANTAR	
	AK	
	ACT	
	R ISI	
DAFTA	R TABELR GAMBAR	х
DAFTA	R GAMBAR	xiii
	R LAMPIRAN	
BAB 1	PENDAHULUAN	
	1.1 Latar Belakang	1
	1.2 Rumusan Masalah	2
	1.3 Tujuan	
	1.4 Manfaat	
	1.5 Batasan Masalah	3
	1.6 Sistematika Pembahasan	
BAB 2	LANDASAN KEPUSTAKAAN	5
	2.1 Kajian Pustaka	5
	2.2 Parkir	
	2.3 NFC	6
	2.3.1 Perbandingan NFC dengan RFID	7
	2.3.2 Perbandingan NFC dengan Bluetooth	8
	2.3.3 Perbandingan NFC dengan QR Code	8
	2.4 Android	9
	2.4.1 Siklus Hidup Android Activity	9
	2.4.2 Arsitektur Android	10
	2.5 Rekayasa Perangkat Lunak	
	2.6 Model Waterfall	12
	2.7 Pola Arsitektur MVC (Model, View, Controller)	13
	2.8 Codelgniter	13

2.9 Pengujian Perangkat Lunak	
2.9.1 Pengujian <i>Reliability</i>	15
2.9.2 Pengujian <i>Compatibility</i>	
BAB 3 METODOLOGI	
3.1 Studi Literatur	
3.2 Observasi	
3.3 Analisis Kebutuhan Sistem	
3.4 Perancangan Sistem	
3.5 Implementasi Sistem	19
3.6 Pengujian dan Analisis Sistem	20
3.6.1 Pengujian White-Box	
3.6.2 Pengujian Black-Box	20
3.6.3 Pengujian Performa	
3.7 Pengambilan Kesimpulan dan Saran	
BAB 4 HASIL	
4.1 Gambaran Umum Sistem	23
4.2 Analisis Kebutuhan4.2.1 Elisitasi Kebutuhan	<b>2</b> 3
4.2.1 Elisitasi Kebutuhan	24
4.2.2 Identifikasi Aktor	
4.2.3 Spesifikasi Kebutuhan	
4.2.4 Diagram Use Case	
BAB 5 PEMBAHASAN	
5.1 Perancangan Sistem	49
5.1.1 Perancangan Arsitektural	50
5.1.2 Perancangan Diagram Aktivitas	53
5.1.3 Perancangan Diagram Sekuen	
5.1.4 Perancangan Diagram Kelas	
5.1.5 Perancangan Basis Data	
5.1.6 Perancangan Antarmuka	
5.2 Implementasi Sistem	
5.2.1 Spesifikasi Lingkungan Implementasi	66
5.2.2 Batasan Implementasi	67

5.2.3 Implementasi Basis Data	
5.2.4 Implementasi Kelas	68
5.2.5 Implementasi Algoritma	
5.2.6 Implementasi Antarmuka	
BAB 6 PENGUJIAN DAN EVALUASI	75
6.1 Pengujian White-Box	75
6.1.1 Tujuan	
6.1.2 Mekanisme	
6.1.3 Hasil Pengujian	75
6.1.4 Analisis	83
6.2 Pengujian Black-Box	84
6.2.1 Validasi	
6.2.2 Reliability	
6.2.3 Compatibility	98
6.3 Pengujian Performa	99
BAB 7 PENUTUP	109
7.1 Kesimpulan	109
7.2 Saran	109
DAFTAR PUSTAKA	110
LAMPIRAN A HASIL WAWANCARA KOORDINATOR DIVISI PARKIR UB	

# **DAFTAR TABEL**

Tabel 2.1 Kajian Pustaka	5
Tabel 4.1 Identifikasi Aktor	28
Tabel 4.2 Spesifikasi Kebutuhan Fungsional	28
Tabel 4.3 Spesifikasi Kebutuhan Non Fungsional	
Tabel 4.4 Skenario Masuk Area Parkir	32
Tabel 4.5 Skenario Keluar Area Parkir	32
Tabel 4.6 Skenario Keluar Area Kampus  Tabel 4.7 Skenario Login	33
Tabel 4.7 Skenario Login	33
Tabel 4.8 Skenario Menginput Kendaraan Masuk Area Parkir	34
Tabel 4.9 Skenario Menginput Kendaraan Keluar Area Parkir	35
Tabel 4.10 Skenario Melihat Aktivitas Parkir	35
Tabel 4.11 Skenario Membuat Laporan Kehilangan	
Tabel 4.12 Skenario Membuat Laporan Ditemukan	
Tabel 4.13 Skenario Klaim Laporan Ditemukan	37
Tabel 4.14 Skenario Mengedit Profil	38
Tabel 4.15 Skenario Lupa Password	
Tabel 4.16 Skenario Memblokir Kendaraan	
Tabel 4.17 Skenario Mencabut Blokir Kendaraan	40
Tabel 4.18 Skenario Memproses Laporan Kehilangan	41
Tabel 4.19 Skenario Menginput Kendaraan Keluar Area Kampus	42
Tabel 4.20 Skenario Mengelola Data Pengendara	42
Tabel 4.21 Skenario Mengelola Data Kendaraan	43
Tabel 4.22 Skenario Mengelola Data Petugas Parkir	45
Tabel 4.23 Skenario Memvalidasi Laporan Kehilangan	
Tabel 4.24 Skenario Mengelola Data Area Parkir	
Tabel 4.25 Skenario Memvalidasi Laporan Ditemukan	48
Tabel 5.1 Spesifikasi Perangkat Keras	
Tabel 5.2 Spesifikasi Perangkat Lunak	
Tabel 5.3 Implementasi Kelas	
Tabel 6.1 Kasus Uji Algoritma Masuk Area Parkir	78

Tabel 6.2 Kasus Uji Algoritma Keluar Area Parkir	80
Tabel 6.3 Kasus Uji Algoritma Menginput Kendaraan Keluar Area Parkir	83
Tabel 6.4 Kasus Uji Validasi Masuk Area Parkir	84
Tabel 6.5 Kasus Uji Validasi Keluar Area Parkir	
Tabel 6.6 Kasus Uji Validasi Keluar Area Kampus	
Tabel 6.7 Kasus Uji Validasi Login	85
Tabel 6.8 Kasus Uji Validasi Menginput Kendaraan Masuk Parkir	86
Tabel 6.9 Kasus Uji Validasi Menginput Kendaraan Keluar Parkir	86
Tabel 6.10 Kasus Uji Validasi Melihat Aktivitas Parkir	87
Tabel 6.11 Kasus Uji Validasi Membuat laporan kehilangan	87
Tabel 6.12 Kasus Uji Validasi Membuat laporan ditemukan	87
Tabel 6.13 Kasus Uji Validasi Klaim Laporan Ditemukan	88
Tabel 6.14 Kasus Uji Validasi Mengedit Profil	88
Tabel 6.15 Kasus Uji Validasi Lupa Password	
Tabel 6.16 Kasus Uji Validasi Blokir Kendaraan	
Tabel 6.17 Kasus Uji Validasi Cabut Blokir Kendaraan	89
Tabel 6.18 Kasus Uji Validasi Memproses Laporan Kehilangan	90
Tabel 6.19 Kasus Uji Menginput Kendaraan Keluar Kampus	90
Tabel 6.20 Kasus Uji Validasi Mengelola Data Pengendara	91
Tabel 6.21 Kasus Uji Validasi Mengelola Data Kendaraan	91
Tabel 6.22 Kasus Uji Validasi Mengelola Data Petugas Parkir	92
Tabel 6.23 Kasus Uji Validasi Memvalidasi Laporan Kehilangan	92
Tabel 6.24 Kasus Uji Validasi Mengelola Data Area Parkir	93
Tabel 6.25 Kasus Uji Validasi Memvalidasi Laporan Ditemukan	93
Tabel 6.26 Hasil Pengujian Validasi	94
Tabel 6.27 Ringkasan Hasil Pengujian Reliability	98
Tabel 6.28 Ringkasan Hasil Pengujian Compatibility	
Tabel 6.29 Alur Masuk Area Parkir	
Tabel 6.30 Alur Keluar Area Parkir	
Tabel 6.31 Alur Keluar Area Kampus	102
Tabel 6.32 Ringkasan Hasil Pengujian Waktu (dalam detik)	103
Tabel 6.33 Ringkasan Hasil Pengujian Waktu Masuk Area Parkir (dalam deti	ik) 104

Tabel 6.34 Ringkasan Hasil Pengujian Wakt	u Keluar Area Parkir (dalam detik). 105
Tabel 6.35 Ringkasan Hasil Pengujian Wak	ktu Keluar Area Kampus (dalam detik) 106
Tabel 6.36 Ringkasan Hasil Pengamatan Kampus (dalam detik)	Waktu Keluar Area Parkir dan Area





## **DAFTAR GAMBAR**

Gambar 2.1 Siklus Hidup Android	
Gambar 2.2 Arsitektur Android	
Gambar 2.3 Tahapan Umum Rekayasa Perangkat Lunak	12
Gambar 2.4 Ilustrasi Model Waterfall	12
Gambar 2.5 Pola Arsitektur MVC (Model, View, Controller)	13
Gambar 2.6 Struktur Folder Codelgniter	
Gambar 3.1 Diagram Alir Tahapan Penelitian	16
Gambar 3.2 Diagram Alir Pengembangan Sistem dengan Model Waterfa	
Gambar 3.3 Tahapan Pengujian Unit	20
Gambar 3.4 Tahapan Pengujian Validasi	21
Gambar 3.5 Tahapan Pengujian Reliability	
Gambar 3.6 Tahapan Pengujian Compatibility	
Gambar 4.1 Diagram Alir Analisis Kebutuhan	24
Gambar 4.2 Prosedur Masuk Area Parkir pada UB	25
Gambar 4.3 Prosedur Keluar Area Parkir pada UB	
Gambar 4.4 Prosedur Keluar Area Kampus pada UB	
Gambar 4.5 Diagram <i>Use Case</i> Sistem	
Gambar 5.1 Diagram Alir Perancangan Sistem	49
Gambar 5.2 Arsitektur Sistem	
Gambar 5.3 Alur Masuk Area Parkir	
Gambar 5.4 Alur Keluar Area Parkir	51
Gambar 5.5 Alur Keluar Area Kampus	52
Gambar 5.6 Alur Pendaftaran Pengendara dan Kendaraan	53
Gambar 5.7 Diagram Aktivitas Menginput Kendaraan Masuk Area Parkir	54
Gambar 5.8 Diagram Aktivitas Menginput Kendaraan Keluar Area Parkir.	54
Gambar 5.9 Diagram Aktivitas Melihat Data Pengendara	55
Gambar 5.10 Diagram Aktivitas Memvalidasi Laporan Kehilangan	55
Gambar 5.11 Diagram Aktivitas Mencabut Blokir Kendaraan	56
Gambar 5.12 Diagram Aktivitas Memproses Laporan Kehilangan	56
Gambar 5.13 Diagram Aktivitas Login	57

Gambar 5.14 Diagram Sekuen Login	58
Gambar 5.15 Diagram Sekuen Menginput Kendaraan Masuk	
Gambar 5.16 Diagram Sekuen Melihat Data Pengendara	
Gambar 5.17 Diagram Sekuen Mencabut Blokir Kendaraan	60
Gambar 5.18 Diagram Kelas Sistem Level Analisis	61
Gambar 5.19 Diagram Kelas Controller Sistem Level Implementasi	61
Gambar 5.20 Diagram Kelas <i>Model</i> Sistem Level Implementasi	62
Gambar 5.21 Diagram Kelas View Sistem Level Implementasi Aktor Admin	62
Gambar 5.22 Diagram Kelas View Sistem Level Implementasi Aktor Petugas	63
Gambar 5.23 Diagram Kelas <i>View</i> Sistem Level Implementasi Aktor Petug Kemananan	
Gambar 5.24 Entity Relationship Diagram (ERD) Sistem	64
Gambar 5.25 Perancangan Antarmuka Input Kendaraan Keluar	64
Gambar 5.26 Perancangan Antarmuka List Petugas	65
Gambar 5.27 Perancangan Antarmuka Form Tambah Petugas	65
Gambar 5.28 Diagram Alir Implementasi	
Gambar 5.29 Skema Basis Data Sistem	
Gambar 5.30 Algoritma Menambahkan Data Petugas	
Gambar 5.31 Algoritma Mengubah Data Petugas	
Gambar 5.32 Algoritma Menghapus Data Petugas	72
Gambar 5.33 Algoritma Melihat Data Petugas	72
Gambar 5.34 Implementasi Antarmuka <i>Login</i>	73
Gambar 5.35 Implementasi Antarmuka NFC Reader	73
Gambar 5.36 Implementasi Antarmuka Proses Keluar Area Parkir	74
Gambar 5.37 Implementasi Antarmuka Scanning Smartcard NFC Reader	74
Gambar 6.1 Pembentukan <i>Node</i> Algoritma Masuk Area Parkir	76
Gambar 6.2 Flowgraph Algoritma Masuk Area Parkir	77
Gambar 6.3 Pembentukan <i>Node</i> Algoritma Keluar Area Parkir	79
Gambar 6.4 Flowgraph Algoritma Keluar Area Parkir	79
Gambar 6.5 Pembentukan <i>Node</i> Menginput Kendaraan Keluar Area Parkir	81
Gambar 6.6 Flowgraph Algoritma Menginput Kendaraan Keluar Area Parkir	82
Gambar 6.7 Alur Masuk Area Parkir10	00

Gambar 6.8 Alur Keluar Area Parkir	101
Gambar 6.9 Alur Keluar Area Kampus	101





## **DAFTAR LAMPIRAN**

LAMPIRAN A HASIL WAWANCARA KOORDINATOR DIVISI PARKIR UB	114
LAMPIRAN B HASIL PENGUJIAN RELIABILITY	116
I AMPIRAN C HASIL PENGULIAN COMPATIBILITY	118





## **BAB 1 PENDAHULUAN**

## 1.1 Latar Belakang

Pengelolaan parkir pada saat ini menjadi bahan pembicaraan yang hangat di masyarakat karena banyak permasalahan yang terjadi didalamnya. Seringkali pengguna kendaraan bermotor kesulitan dalam mencari lahan parkir dikarenakan sudah penuh, dan harus mencari lahan parkir lain agar dapat memarkir kendaraan mereka (Setiawan, 2007). Beberapa masyarakat memarkir kendaraan mereka pada tempat yang bukan diperuntukkan untuk parkir, seperti pada bahu jalan, trotoar, lapangan, lahan terbuka hijau, dan lain-lain. Selain itu kendaraan yang mereka parkir tidak selalu terjamin keamanannya meskipun ada petugas yang berjaga, karena curanmor selalu mengintai dan beraksi disaat ada kesempatan (Kurniawan, 2010).

Sistem perparkiran secara manual masih diterapkan dalam lingkungan Universitas Brawijaya (UB). Salah satu skenario sistem perparkiran yang ada adalah ketika kendaraan akan masuk ke dalam area parkir, maka petugas akan memberikan kartu parkir dan menyerahkan kepada pengendara sebagai tanda bukti parkir. Kemudian ketika kendaraan tersebut akan keluar dari area parkir, maka petugas akan melakukan pengecekan dengan meminta kartu parkir kepada pengendara serta surat tanda nomer kendaraan (STNK). Petugas melakukan pencocokan antara plat nomor kendaraan dengan STNK kendaraan, setelah itu kendaraan dapat keluar dari area parkir.

Dari salah satu skenario sistem perparkiran tersebut menuai banyak permasalahan yaitu waktu yang dibutuhkan cukup lama dalam proses keluarmasuk dikarenakan pengendara harus menunggu petugas parkir untuk melakukan pengecekan plat nomor kendaraan dengan STNK yang tentunya memakan waktu yang tidak sebentar, selain itu kartu parkir yang ada tidak sepadan bila dibandingkan dengan kendaraan yang parkir dalam area parkir, sehingga terkadang ada kendaraan yang parkir tanpa diberikan kartu parkir sebagai tanda parkir. Hal ini mengakibatkan kendaraan tidak memiliki tanda bukti bahwa telah masuk area parkir sehingga bila terjadi kehilangan bukan menjadi tanggung jawab pengelola area parkir dan hal ini tentunya merugikan pengendara.

Dari paparan permasalahan tersebut, dibutuhkan sebuah sistem pengeloaan parkir yang dapat membantu pengelolaan parkir menjadi lebih optimal. Tolak ukur yang digunakan dalam penelitian ini adalah keamanan dan manajemen data. Sistem yang akan dibuat dalam skripsi ini adalah sistem pengelolaan parkir menggunakan perangkat NFC berbasis android. Near Field Communication (NFC) adalah teknologi komunikasi nirkabel jarak dekat yang ada pada ponsel (Mazzocchi, 2014). Kelebihan NFC bila dibandingkan dengan bluetooth adalah kemampuannya berkomunikasi tanpa proses pairing. Selain itu NFC menguras energi baterai lebih sedikit daripada bluetooth (Kristiani, 2013).

Sistem ini menawarkan kemudahan dalam transaksi parkir karena memakai perangkat NFC yang difungsikan sebagai identitas pemakai sistem parkiran. Penggunaan perangkat NFC dalam kartu pintar membuat semua orang dapat menggunakan sistem ini. Sistem ini mengusahakan permasalahan-permasalahan yang sering terjadi dalam sistem perparkiran manual dapat teratasi tanpa mengabaikan aspek-aspek lain seperti keamanan, integritas tetap terjaga bahkan lebih optimal bila dibandingkan dengan sistem perparkiran manual.

Dalam penelitian sebelumnya, Septiana telah membuat sistem perangkat lunak untuk simulasi keluar masuk dan pembayaran kendaraan berbasis RFID. Skenario yang dibuat oleh Septiana adalah ketika kendaraan keluar masuk area parkir maka identitas pengendara akan dibaca oleh RFID, lalu disimpan dengan data plat nomer yang telah diambil gambarnya secara otomatis pada sistem. Selain itu sistem tersebut memiliki fitur kehilangan kendaraan, jika terjadi kehilangan kendaraan admin sistem dapat melaporkan kepada bagian keamanan (Septiana, 2013).

Dalam skripsi ini berusaha memperbaiki sistem parkir yang telah dibuat oleh Fransischa dengan penggantian RFID dengan NFC. hal ini karena NFC dapat dijumpai dalam ponsel-ponsel android yang dijadikan sebagai platform dalam sistem ini. dengan penggunaan ponsel android sebagai *reader* dan kartu NFC pengganti karcis langganan bagi pengguna sistem perparkiran. Dengan bantuan perangkat NFC yang terintegrasi pada sistem diharapkan dapat membantu pengguna kendaraan bermotor dalam memberi rasa aman bagi pengguna kendaraan yang sedang parkir karena perangkat NFC tersebut hanya bisa dioperasikan oleh pemilik kendaraan tersebut, serta dapat memonitor aktivitas parkir yang dicatat pada sistem.

Smartphone dalam sistem pengelolaan parkir ini akan dipergunakan sebagai media komunikasi dengan kartu pintar menggunakan perangkat NFC sebagai perantara dengan hanya mendekatkan kedua perangkat tersebut (Kristiani, 2013). Diharapkan dengan penggunaan perangkat NFC tersebut pelayanan parkir dapat menjadi lebih cepat sehingga meningkatkan produktivitas dan efisiensi waktu bila dibandingkan dengan proses yang ada pada saat ini.

#### 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan sebelumnya maka didapatkan rumusan masalah sebagai berikut:

- 1. Bagaimana tingkat efisiensi proses parkir dengan menggunakan aplikasi NFCPS (NFC Parking System)?
- 2. Bagaimana tingkat reliabilitas aplikasi NFCPS (NFC Parking System) bila diakses oleh seluruh area parkir secara bersamaan?
- 3. Bagaimana tingkat kompabilitas aplikasi NFCPS (NFC Parking System) bila dijalankan dalam browser yang berbeda?

## 1.3 Tujuan

Tujuan yang hendak diperoleh dalam penelitian ini adalah:

- 1. Memodelkan proses bisnis dalam pengelolaan parkir yang ada dalam lingkungan Universitas Brawijaya agar sesuai dengan kebutuhan organisasi.
- 2. Menerapkan perangkat NFC yang terdapat dalam *smartcard* sebagai identitas pengguna sistem.
- 3. Membuat sistem pengelolaan parkir yang sesuai dengan kebutuhan organisasi.

### 1.4 Manfaat

Berdasarkan penjelasan yang telah dipaparkan, maka manfaat yang diharapkan dari penelitian ini antara lain:

- 1. Memberikan kemudahan kepada pengguna layanan parkir pada Universitas Brawijaya.
- 2. Dapat meningkatkan aspek keamanan dalam area parkir karena setiap aktivitas keluar-masuk area parkir selalu dilakukan pencatatan dan menggunakan *smartcard* sebagai identitas pengguna.

#### 1.5 Batasan Masalah

Berdasarkan latar belakang dan rumusan masalah yang telah diuraikan, agar permasalahan yang dirumuskan dapat lebih terfokus, maka penelitian ini dibatasi dalam hal-hal sebagai berikut:

- 1. Perangkat bergerak yang digunakan dalam pembuatan aplikasi ini adalah *smartphone android* yang memiliki sensor NFC.
- 2. Sistem pengelolaan parkir terbatas pada pengguna kendaraan roda dua yang berlangganan parkir.
- Fokus dari sistem ini adalah sistem perparkiran di kampus, dengan studi kasus proses parkir pada kampus Universitas Brawijaya dengan sampel penelitian pada area parkir PTIIK depan.
- 4. Untuk pengguna diberikan *smartcard* menggunakan perangkat NFC sebagai identitas pengguna area parkir.

#### 1.6 Sistematika Pembahasan

Sistematika penulisan ini ditujukan untuk memberikan gambaran dan uraian singkat terkait penulisan skripsi ini yang meliputi beberapa bab, yaitu:

#### 1. BAB 1 Pendahuluan

Bab ini memuat tentang latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, dan sistematika penulisan.

## 2. BAB 2 Landasan Kepustakaan

Bab ini membahas kajian dari penelitian yang telah dilakukan sebelumnya serta dasar teori yang mendukung penyelesaian masalah dalam penelitian ini.

## 3. BAB 3 Metodologi

Bab ini memaparkan metode atau cara yang digunakan dalam penelitian ini secara umum.

## 4. BAB 4 Rekayasa persyaratan

Bab ini membahas tentang analisis kebutuhan, dan perancangan, serta implementasi dari Sistem.

## 5. BAB 5 Perancangan dan implementasi

Bab ini membahas tentang analisis kebutuhan, dan perancangan, serta implementasi dari Sistem.

## 6. BAB 6 Pengujian

Pada bab ini akan dimuat hasil pengujian dan analisis terhadap sistem yang telah dikembangkan.

#### 7. BAB 7 Penutup

Memuat tentang kesimpulan yang didapatkan dari pengembangan dan pengujian sistem dalam penelitian ini serta saran-saran untuk pengembangan lebih lanjut.

## **BAB 2 LANDASAN KEPUSTAKAAN**

Pada bab ini akan dipaparkan tentang kajian pustaka dan dasar teori yang mendukung pengembangan perangkat lunak yang dilakukan.

## 2.1 Kajian Pustaka

Analisis kebutuhan pada penelitian ini dilakukan dengan mengkaji penelitian terdahulu. Penulis menemukan beberapa penelitian yang relevan terkait pengembangan sistem pengelolaan parkir. Terdapat empat penelitian yang penulis gunakan sebagai bahan untuk menganalisis kebutuhan sistem. Tabel 2.1 memodelkan modul-modul yang diimplementasikan pada masing-masing penelitian.

**Tabel 2.1 Kajian Pustaka** 

No.	Penelitian	Fokus Penelitian
1.	Pengembangan prototipe sistem informasi perparkiran sepeda motor di Universitas Brawijaya (Zufti, 2014) .	Menyajikan informasi mengenai lahan parkir dengan sms gateway dan pencatatan aktivitas pada area parkir.
2.	Sistem perangkat lunak untuk simulasi keluar masuk dan pembayaran parkir kendaraan berbasis RFID (Septiana, 2013).	Menerapkan RFID dalam proses keluar-masuk kendaraan dan pembayaraan parkir.
3.	Analisis Biaya dan Manfaat pada Perencanaan Sistem Keluar-Masuk Kendaraan dengan menggunakan barcode di Universitas Brawijaya (Fuati, 2014).	Menganalisis biaya dan manfaat penggunaan barcode pada sistem keluar-masuk kendaraan di Universitas Brawijaya

Penelitian yang dilakukan oleh Fikri Zufti dengan judul "Pengembangan prototipe sistem informasi perparkiran sepeda motor di universitas brawijaya" membahas tentang prototipe sistem informasi perparkiran sepeda motor di Universitas Brawijaya. Penekanan penelitiannya terletak pada pencatatatan aktivitas pada area parkir dan penggunaan sms gateway untuk menginformasikan status kendaraan yang diparkir. Pada akhir penelitiannya, ia menyimpulkan bahwa implementasi sistem tersebut dapat membuat proses

perparkiran lebih efisien dan membantu stakeholder yang terlibat didalamnya (Zufti, 2014).

Penelitian selanjutnya dilakukan oleh Fransischa Tika Septiana dengan judul "Sistem perangkat lunak untuk simulasi keluar masuk dan pembayaran parkir kendaraan berbasis RFID". Ia mengembangkan sistem yang dapat mengotomatisasi proses keluar masuk kendaraan menggunakan perangkat RFID. Sistem tersebut juga dilengkapi dengan pencatatan aktivitas pada area parkir (Septiana, 2013).

Penelitian terakhir yang penulis kaji berjudul "Analisis Biaya dan Manfaat pada Perencanaan Sistem Keluar-Masuk Kendaraan dengan menggunakan barcode di Universitas Brawijaya" oleh Laily Nur Fuati. Penelitian ini bertujuan mengkaji penerapan barcode pada sistem keluar-masuk kendaraan di Universitas Brawijaya dari segi manfaat dan biaya dengan metode Enterprise Architecture Planning (EAP), dan perhitungan biaya dengan LOC dan COCOMO (Fuati, 2014).

Berdasarkan kajian terhadap empat penelitian yang telah diuraikan sebelumnya, maka didapatkan beberapa proses dalam aktivitas parkir yang dapat dijadikan kebutuhan sistem untuk kemudian divalidasi dan verifikasi guna mendapatkan kebutuhan sistem yang sesuai.

#### 2.2 Parkir

Menurut Direktur Jenderal Perhubungan Darat (1996), parkir merupakan keadaan tidak bergerak suatu kendaraan yang bersifat sementara. Parkir merupakan suatu kebutuhan bagi pemilik kendaraan dan menginginkan kendaraannya parkir di tempat, dimana tempat tersebut mudah untuk dicapai (Darat, 1996). Dalam kehidupan sehari-hari parkir menjadi suatu aktivitas yang vital bagi pengguna kendaraan bermotor untuk menempatkan kendaraannya pada tempat yang telah disediakan.

Menurut Hobbs (1995), penyediaan tempat-tempat parkir menjadi bagian yang tidak bisa dipisahkan dalam perencanaan transportasi. Karena lalu lintas menuju suatu tempat tujuan dan setelah mencapai tempat tersebut kendaraan harus diparkir, sementara pengendaranya melakukan berbagai urusan, misalnya keperluan pribadi, keperluan umum, rekreasi, dan sebagainya (Hobbs, 1995).

#### 2.3 NFC

Near Field Communication (NFC) adalah satu set standar untuk smartphone dan perangkat serupa untuk menjalin komunikasi radio satu sama lain dengan menyentuh mereka bersama-sama atau membawa mereka ke dalam jarak dekat, biasanya tidak lebih dari beberapa inci (Lee, Hong, Kao, & Cheng, 2014). NFC adalah sebuah teknologi baru yang beroperasi di pita 13,56 MHz. NFC memungkinkan pertukaran informasi jarak pendek antara dua perangkat (Essbach, 2012).

NFC adalah teknologi komunikasi berbasis nirkabel yang berdasarkan pada Radio Frequency Identification (RFID) dan kompatibel dengan ISO/IEC 14443 yang memenuhi *smartcard contectless* dan *reader* dari kartu tersebut. Kartu ini termasuk yang digunakan dalam transportasi publik di seluruh dunia seperti *creditcard contactless* (Juntunen, Luukkainen, & Tuunainen, 2010). Dalam beberapa tahun terakhir NFC telah terutama digunakan dalam sistem kontrol akses, identifikasi pengguna dan sebagai elektronik atau mobile payment dengan berbagai macam bentuk seperti mobile money, mobile wallet dan transfer mobile (Essbach, 2012).

Pada dasarnya, fungsi pertukaran data pada NFC diaktifkan dengan mengambil dua perangkat NFC yang aktif saling berdekatan. Untuk perangkat dengan NFC tag yang sudah tertanam, perangkat memiliki fungsi NFC. NFC memungkinkan perangkat untuk menggunakan tag dan menghubungkan mereka dengan antena. Antena membuat NFC memungkinkan perangkat bertukar data. NFC berkomunikasi melalui induksi medan magnet (Wu & Yang, 2013).

## 2.3.1 Perbandingan NFC dengan RFID

RFID memiliki tiga bagian dari sistemnya yaitu tag, antenna, dan reader. Dari segi frekuensi, RFID memiliki tiga area frekuensi yaitu low frequency (LF) beroperasi pada 125-134 kHz yang memiliki jarak sambungan koneksi hingga 10 cm, high frequency (HF) beroperasi pada 13,56 MHz yang memiliki jarak sambungan koneksi hingga 30 cm, dan ultra high frequency (UHF) beroperasi pada 856-960 MHz yang memiliki jarak sambungan koneksi hingga 100 meter.

RFID memiliki dua mode operasi yaitu aktif dan pasif. Pada mode aktif RFID dapat melakukan *broadcast* hingga jarak 100 meter dengan menggunakan *power source* sendiri. Pada mode pasif RFID tidak membutuhkan *power source* melainkan reader RFID yang akan memberikan *power*. Jarak pembacaan RFID pada mode pasif ini hingga 25 meter. RFID ini lebih populer digunakan sebagai *assets tracking, race timing,* dan *tool tracking*.

NFC sendiri beroperasi pada frekuensi high frequency RFID yaitu 13,56 MHz. NFC dapat berperan sebagai reader dan tag, dengan jarak yang sangat dekat dikarenakan pembancaan NFC dibatasi pada jarak 10 cm. NFC populer digunakan sebagai *smart posters, contactless payment,* dan *information sharing* (Famoco, 2015).

Fungsi NFC dibagi menjadi tiga mode yang berbeda yaitu reader/writer, peer-to-peer dan card emulation, hanya satu mode yang dapat dipilih pada satu waktu yang berarti bahwa misalnya ketika modus reader/writer aktif, modus emulasi kartu tidak dapat digunakan. Berikut ini penjelasan lebih lengkap dari ketiga mode yang ada pada NFC:

#### Reader / Writer

Mode ini memungkinkan perangkat NFC untuk membaca data dari tipe tag NFC standar yang berbeda.

#### 2. Peer-To-Peer

Mode ini memungkinkan dua perangkat NFC untuk bertukar data satu sama lain. Modus peer-to-peer adalah standar dalam ISO/IEC 18092. Dalam modus peer-to-peer, perangkat NFC dapat berada dalam mode aktif atau pasif dan komunikasi dilakukan melalui saluran setengah duplex dua arah.

#### 3. Card Emulation

Dalam modus emulasi kartu, sebuah tindakan perangkat NFC sebagai smart card contactless. Dalam modus ini, perangkat NFC tidak akan menghasilkan kode Radio Frequency sendiri. Jenis kartu yang didukung adalah Mifare ISO/IEC 14443 Tipe A dan Tipe B, FeliCa dan ISO 15693 (Mazzocchi, 2014).

## 2.3.2 Perbandingan NFC dengan Bluetooth

NFC berbeda dari yang sudah ada teknologi jarak dekat seperti Bluetooth. Dalam NFC tidak memerlukan *pairing* atau proses mencari pasangan perangkat. Dua perangkat yang memiliki NFC dapat didekatkan satu sama lain dan dapat mengidentifikasi satu sama lain dengan mode komunikasi dan kecepatan data secara otomatis. Ini berarti bahwa pengguna tidak diperlukan untuk melakukan *setup* atau memberikan izin untuk komunikasi antar perangkat, melainkan tindakan mendekatkan dua perangkat dalam jarak dekat untuk membuat sambungan. Rentang jarak dari kedua perangkat yang didekatkan ini hanya sampai sekitar 10 cm (Opperman & Haneke, 2011).

Selain itu, NFC memiliki kecepatan lambat dan jarak yang pendek dalam komunikasi, hal ini memungkinkan NFC untuk mengkonsumsi daya sesedikit mungkin sehingga dapat dinyalakan sepanjang waktu dan tidak mempengaruhi baterai telepon terlalu banyak (Saparkhojayev, Dauitbayeva, Nurtayev, & Baimenshina., 2014) jika dibandingkan dengan teknologi standar Bluetooth. Perangkat yang terhubung menggunakan NFC harus dekat satu sama lain. Itu juga berguna dalam lokasi ramai karena mencegah gangguan komunikasi yang disebabkan perangkat lain yang ada dan berusaha untuk berkomunikasi (Lee, Hong, Kao, & Cheng, 2014).

#### 2.3.3 Perbandingan NFC dengan QR Code

Kode QR atau biasa dikenal dengan istilah QR Code adalah bentuk evolusi kode batang dari satu dimensi menjadi dua dimensi. Kode QR adalah suatu jenis kode matriks atau kode batang dua dimensi dengan fungsionalitas utama yaitu dapat dengan mudah dibaca oleh pemindai. QR merupakan singkatan dari quick response atau respons cepat, yang sesuai dengan tujuannya adalah untuk menyampaikan informasi dengan cepat dan mendapatkan respons yang cepat pula. Kode QR mampu menyimpan informasi secara horizontal dan vertikal, oleh karena itu secara otomatis Kode QR dapat menampung informasi yang lebih banyak.

Kode QR berfungsi bagaikan hipertaut fisik yang dapat menyimpan alamat dan URL, nomer telepon, teks dan sms yang dapat digunakan pada majalah, surat harian, iklan, pada tanda-tanda bus, kartu nama ataupun media lainnya. Atau dengan kata lain sebagai penghubung secara cepat konten daring dan konten luring. Kapasitas data yang dapat disimpan oleh QR Code bisa menyimpan sampai ratusan data. Ukuran cetak untuk QR Code jauh lebih kecil karena QR Code dapat menyimpan data baik secara horisontal maupun vertikal. Hasil cetakan di QR Code lebih tahan terhadap kerusakan seperti debu sampai robek bahkan data di QR Code masih bisa dibaca walaupun sebagian kode sudah rusak atau robek -maksimum tingkat kerusakan 30%. QR Code dapat dibaca dari segala arah atau sudut (360 derajat) sehingga kemungkinan gagal dalam membaca QR Code sangat kecil (Travo, 2011).

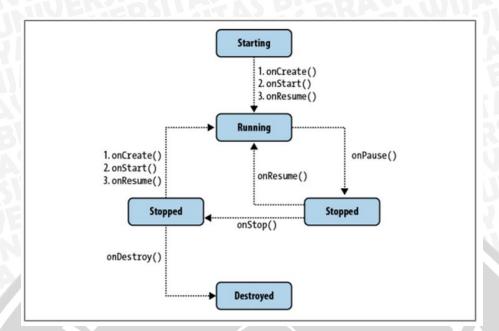
NFC dan kode QR memiliki fungsionalitas yang hampir sama yakni dapat menyimpan informasi. Namun, kode QR dapat dibuat secara bebas dan informasi dalam kode QR bisa dimanipulasi. Kode QR sangat sulit dibaca apabila mengalami penyusutan ukuran, dan kualitas pemindaian kode QR bergantung pada kualitas alat pemindai (Taufik, 2014). Dari beberapa kekurangan tersebut, dapat diminimalisir dengan menggunakan kartu NFC.

#### 2.4 Android

Android adalah platform *open source* komprehensif yang dirancang untuk perangkat *mobile*. Android didesain untuk berjalan pada segala bentuk perangkat fisik. Android tidak membuat asumsi apapun tentang ukuran perangkat layar, resolusi, *chipset* dan sebagainya. Inti dari Android di desain agar dapat *portable* (Gargenta, 2011).

## 2.4.1 Siklus Hidup Android Activity

Aplikasi Android biasanya terdiri dari beberapa activity yang saling terkait. Activity sendiri merupakan suatu komponen aplikasi yang menyediakan antarmuka pada layar pengguna, sehingga pengguna dapat melakukan interaksi dengan mudah pada aplikasi tersebut. Penjelasan dari siklus hidup Android activity dapat dilihat pada gambar 2.1.



**Gambar 2.1 Siklus Hidup Android** 

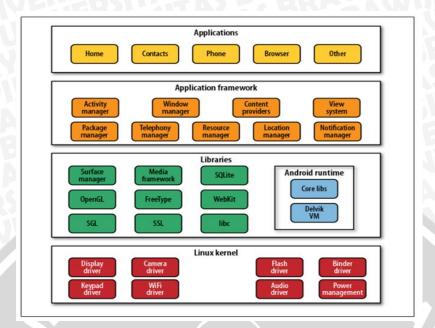
Sumber: (Gargenta, 2011)

Terdapat beberapa state dalam siklus hidup activity Android, antara lain:

- 1. Starting State yaitu keadaan ketika activity belum ada di memori.
- 2. Running State yaitu kondisi ketika berada di layar dan sedang berinteraksi dengan pengguna.
- 3. Paused State yaitu keadaan ketika activity tidak terlihat, namun masih ada di memori.
- 4. Destroyed State yaitu keadaan ketika activity sudah tidak berada lagi di memori.

#### 2.4.2 Arsitektur Android

Arsitektur Android sendiri terdiri 5 lapisan yaitu antara lain *linux kernel, android runtime, libraries, application framework,* dan *applications*. Dimana tiap lapisan mempunyai karakteristik dan tujuan masing-masing (Gargenta, 2011). Penjelasan dari arsitektur android dapat dilihat pada gambar 2.2.



Gambar 2.2 Arsitektur Android

Sumber: (Gargenta, 2011)

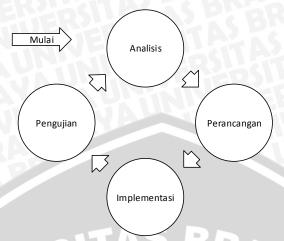
Dari gambar di atas diketahui beberapa lapisannya yaitu :

- 1. Kernel Linux, dimana Android sendiri dibangun di atas kernel linux. Terdapat beberapa alasan mengapa linux dipakai sebagai dasar arsitekturnya, yaitu karena fiturnya, keamanannya, dan juga portability.
- 2. Di lapisan Android *runtime*, terdapat *dalvik* yang merupakan *virtual-machine* yang dirancang khusus untuk Android.
- 3. Di lapisan *libraries*, biasanya terdapat pustaka umum yang diambil dari komunitas *open source* yang menyediakan layanan tertentu untuk lapisan aplikasi.
- 4. Application framework menyediakan lingkungan kaya yang menyediakan services yang dapat membantu app developer untuk menyelesaikan pekerjaan.

Di lapisan *application*, merupakan lapisan yang kita lihat secara langsung dan setiap aplikasi Android harus di *signed* sebelum di *install* di sebuah *device*.

## 2.5 Rekayasa Perangkat Lunak

Rekayasa perangkat lunak merupakan salah satu disiplin ilmu rekayasa yang fokus kepada tahapan produksi perangkat lunak dari tahap analisis spesifikasi kebutuhan hingga tahap pemeliharaan perangkat lunak (Sommerville, 2007). Tahapan umum rekayasa tersebut dapat dilihat pada Gambar 2.3.



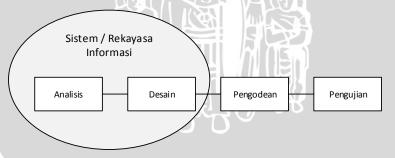
Gambar 2.3 Tahapan Umum Rekayasa Perangkat Lunak

Sumber: (Sukamto & Shalahuddin, 2013)

Proses perangkat lunak merupakan beberapa kumpulan aktivitas yang mempunyai tujuan untuk mengembangkan atau mengubah perangkat lunak. Secara umum proses perangkat lunak terdiri dari pengumpulan spesifikasi (specification), pengembangan (development), validasi (validation), dan evolusi (evolution) (Sukamto & Shalahuddin, 2013).

## 2.6 Model Waterfall

Salah satu model SDLC (*Software Development Lifecycle*) yang paling banyak digunakan adalah model *waterfall*. Model ini sering juga disebut sebagai model *sequential linier* atau alur hidup klasik (Sukamto & Shalahuddin, 2013). Gambar 2.4 mengilustrasikan tahapan pada model *waterfall*.



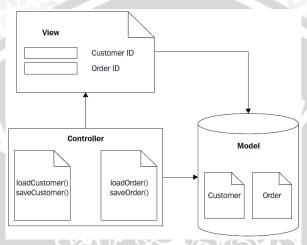
Gambar 2.4 Ilustrasi Model Waterfall

Sumber: (Sukamto & Shalahuddin, 2013)

Kelemahan yang ada pada model waterfall adalah ketika adanya perubahan spesifikasi. Perubahan tersebut sangat sulit dilakukan sebab tahapan pada proses pengembangan ini bersifat statis. Oleh karena itu, model ini sangat cocok digunakan untuk pengembangan sistem dengan spesifikasi yang tidak berubah-ubah (Sukamto & Shalahuddin, 2013). Pada penelitian ini, sistem yang dikembangkan bersifat generik, di mana pada umumnya sistem tersebut memiliki spesifikasi yang sudah tetap di awal, sehingga model SDLC yang digunakan adalah model waterfall.

## 2.7 Pola Arsitektur MVC (Model, View, Controller)

MVC (*Model, View, Controller*) merupakan salah satu pola yang dikenal luas sebagai konsep yang melakukan pemisahan proses antara lapisan presentasi (antarmuka), logika bisnis, dan data pada satu aplikasi. MVC dapat mempermudah pekerjaan jika terdapat pengubahan, perbaikan, dan perkembangan dalam sebuah sistem (Abeysinghe, 2009). Gambar 2.5 merepresentasikan pola arsitektur MVC.



Gambar 2.5 Pola Arsitektur MVC (Model, View, Controller)

Sumber: (Abeysinghe, 2009)

Terdapat tiga komponen yang menunjukkan lapisan komponen dalam MVC yaitu:

Model, merupakan komponen yang melakukan pengelolaan terhadap akses data yang terjadi pada aplikasi. Model bertugas terhadap respon jika view melakukan permintaan data dan respon jika controller melakukan perintah untuk pengubahan data (Abeysinghe, 2009).

View, merupakan komponen yang bertugas untuk menampilkan data dari model ke dalam bentuk interaksi antarmuka. Dengan kata lain, view melakukan pengelolaan terhadap informasi yang ditampilkan (Abeysinghe, 2009).

Controller, merupakan komponen yang bertugas melakukan respon dari perintah-perintah pengguna aplikasi seperti pemanggilan view atau pengubahan/permintaan data yang melibatkan pemanggilan model. Dalam konteks nyata, controller berisi kode yang terkait dengan logika proses bisnis aplikasi (Abeysinghe, 2009).

## 2.8 Codelgniter

Salah satu framework PHP terkenal yang menerapkan pola arsitektur MVC adalah Codelgniter. Framework sendiri merupakan sebuah kerangka kerja yang mempermudah pengembang dalam mengembangkan aplikasi sehingga ia dapat melakukan pengubahan dalam aplikasinya dengan mudah dan dapat digunakan

untuk aplikasi lainnya yang sejenis. *Framework* Codelgniter dikembangkan dan dipelihara oleh Ellis Lab. Versi pertama Codelgniter dirilis ke publik pada 28 Februari 2006. Codelgniter tersedia pada laman http://codeigniter.com/downloads/ (Orr & Zadik, 2013) (Sukamto & Shalahuddin, 2013). Gambar 2.6 menunjukkan komponen-komponen yang ada pada *framework* Codelgniter.

codeigniter/ index.php application/ cache/ config/ controllers/ core/ errors/ helpers/ hooks/ language/ libraries/ logs/ third party/ views/ system/ core/ database/ fonts/ helpers/ language/ libraries/

## Gambar 2.6 Struktur Folder Codelgniter

Sumber: (Orr & Zadik, 2013)

Terdapat beberapa komponen folder yang wajib diperhatikan yaitu:

- 1. application/config
  - Folder ini berisi tentang konfigurasi dari Codelgniter (Orr & Zadik, 2013).
- 2. application/controller
  - Folder ini berisi semua kelas *controller* dari aplikasi. *Controller* pada folder ini memiliki hubungan generalisasi kepada kelas CI\_Controller. Fungsi yang terdapat pada *controller* dapat dipanggil melalui URL (Orr & Zadik, 2013).
- 3. application/view
  - Folder ini berisi semua *file view*. Sebuah komponen *view* berisi konten HTML yang dieksekusi oleh *browser* pengguna aplikasi. Sebuah *view* dapat berupa halaman web atau halaman RSS (Orr & Zadik, 2013).
- 4. application/model
  - Folder ini berisi semua kelas *model* dari aplikasi. *Model* pada folder ini memiliki hubungan generalisasi kepada kelas CI\_Model (Orr & Zadik, 2013).

## 2.9 Pengujian Perangkat Lunak

Pengujian Perangkat Lunak adalah elemen kritis dari jaminan kualitas perangkat lunak dan merepresentasikan kajian pokok dari spesifikasi, desain dan pengkodean (Presmann, 2001).

## 2.9.1 Pengujian Reliability

Karakteristik *reliability* pada sistem akan dinilai dengan melakukan pengujian terhadap request yang dilakukan oleh user dalam selang waktu tertentu yang dapat diterima oleh sistem. Kriteria yang diuji pada karakteristik *reliability* yaitu *sessions*, *pages*, dan *hits* (Setialana, 2014).

Karakteristik *reliability* pada sistem akan dinilai dengan melakukan pengujian terhadap kriteria *sessions, pages,* dan *hits. Sessions* merupakan skenario uji pengguna dari melakukan *request* hingga mendapatkan respons dari sistem, sedangkan *pages* merupakan halaman yang sukses diunduh, sedangkan *hits* merupakan *request* yang sukses diterima sistem.

Jika pengujian telah didapatkan hasilnya maka akan dihitung dengan menggunakan Persamaan 2.1 sesuai dengan ISO (2002). Hasil dari persamaan tersebut dibandingkan dengan standar Telcordia yaitu 95% dari kasus uji harus berhasil (Asthana & Olivieri, 2009) untuk mendapatkan nilai *reliability* dari sistem.

$$X = \frac{A}{B} \tag{2.1}$$

Dengan:

X = Nilai kriteria reliability.

A = Jumlah berhasil dalam pengujian.

B = Jumlah pengujian keseluruhan.

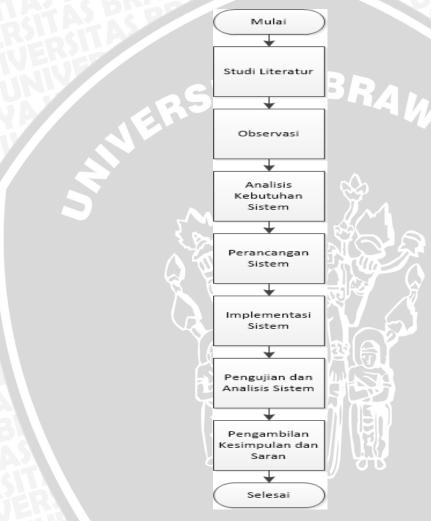
Jika nilai *reliability* dari sistem lebih dari sama dengan 95% maka sistem memenuhi karakteristik *reliability*. Namun jika nilai *reliability* sistem masih kurang dari 95% maka *request* dari pengguna harus dibatasi.

## 2.9.2 Pengujian Compatibility

Karakteristik compatibility pada sistem akan dinilai dengan melakukan pengujian terhadap sistem dengan dijalankan pada browser yang berbeda. Untuk dapat memenuhi standar compatibility, diperlukan sistem yang tidak memiliki error dan warning jika dioperasikan di berbagai browser. Selain error dan warning terdapat beberapa permasalahan yang terjadi dalam compatibility pada cross browser diantaranya inkonsistensi pada page layout, inkonsistensi pada grid, dan page validation yang tidak bekerja (Abodeqa, 2012).

### **BAB 3 METODOLOGI**

Bab ini membahas metode yang digunakan dalam penelitian yang terdiri dari beberapa tahapan. Beberapa tahapan tersebut yaitu observasi, studi literatur, analisis kebutuhan sistem, perancangan sistem, implementasi sistem, pengujian dan analisis sistem, serta pengambilan kesimpulan dan saran. Gambar 3.1 merepresentasikan alur kerja dari tahapan yang telah diuraikan sebelumnya.



**Gambar 3.1 Diagram Alir Tahapan Penelitian** 

### 3.1 Studi Literatur

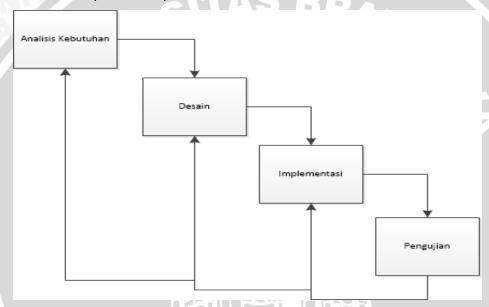
Studi literatur memaparkan kajian pustaka dan dasar teori yang mendukung dilakukannya penelitian ini. Pustaka dan teori yang terkait dengan penelitian ini meliputi:

- 1. Proses Bisnis pada sistem parkir
- 2. NFC
- 3. Android



- 4. Rekayasa Perangkat Lunak
- 5. Model Waterfall
- 6. Pola Arsitektur MVC (Model, View, Controller)
- 7. Codelgniter

Model pengembangan sistem yang digunakan pada penelitian ini adalah model waterfall. Pemilihan model pengembangan tersebut didasarkan pada jenis sistem yang bersifat generik. Spesifikasi kebutuhan sistem generik tersebut umumnya sudah tetap di tahap awal pengembangan sehingga cocok terhadap model waterfall yang harus meminimalisir perubahan spesifikasi kebutuhan. Tahapan pengembangan yang dilakukan berdasarkan model waterfall pada penelitian ini dapat dilihat pada Gambar 3.2.



Gambar 3.2 Diagram Alir Pengembangan Sistem dengan Model Waterfall

Pengembangan sistem dengan model waterfall pada penelitian ini melibatkan empat tahap yaitu:

#### 1. Analisis Kebutuhan

Pada tahap ini, spesifikasi kebutuhan dari sistem pengelolaan parkir dianalisis dengan melakukan observasi langsung kepada stakeholder yang mengetahui proses bisnis dalam pengelolaan parkir dalam hal ini yaitu koordinator parkir serta mengkaji penelitian-penelitian terdahulu. Metode tersebut dipilih sebab proses dan fitur pada sistem harus sesuai dengan kebutuhan organisasi. Sehingga, teknik yang digunakan dalam tahap ini adalah melakukan observasi serta studi literatur. Hasil dari tahap ini berupa spesifikasi kebutuhan perangkat lunak yang disertai dengan identifikasi aktor pada sistem dan diagram *use case* beserta skenarionya.

#### 2. Desain

Pada tahap desain, sistem akan dirancang berdasarkan hasil yang didapatkan pada tahap analisis. Hasil dari tahap ini berupa desain arsitektural, desain

diagram kelas, diagram aktivitas, diagram sekuen, dan desain antarmuka sistem.

## 3. Implementasi

Pada tahap ini, perancangan sistem yang telah dibuat pada tahap desain akan diubah menjadi kode program. Kode program tersebut diimplementasikan dengan bahasa PHP pada sisi *back end* dan sintaks HTML, CSS, dan aplikasi android native pada sisi *front end*. Hasil dari tahap ini berupa sistem yang sesuai dengan rancangan sistem pada tahap desain.

#### 4. Pengujian

Pada tahap pengujian, sistem akan diuji dengan rencana pengujian yang disesuaikan dengan kebutuhan pengujian terhadap sistem.

#### 5. Perawatan

Pada tahap perawatan, sistem akan diinstalasi dan digunakan kemudian mengumpulkan *error* yang belum ditemukan pada fase sebelumnya. Dalam penelitian ini pada tahap perawatan tidak dilakukan karena sistem tidak diinstalasi pada obyek penelitian dalam hal ini divisi parkir UB.

## 3.2 Observasi

Dalam observasi ini, penulis melakukan wawancara kepada *stakeholder* yang mengetahui sistem pengelolaan parkir di Universitas Brawijaya yaitu koordinator divisi parkir dan sudut pandang penulis sebagai mahasiswa yang menjadi salah satu pengguna parkiran untuk mengetahui peran-peran dan proses bisnis yang melibatkan mereka dalam sistem pengelolaan parkir. Penulis juga melakukan observasi langsung kepada area parkir yang dijadikan sampel penelitian untuk mengetahui proses bisnis yang terjadi dalam area parkir yaitu pada area parkir PTIIK depan.

#### 3.3 Analisis Kebutuhan Sistem

Analisis kebutuhan yang dilakukan pada penelitian ini mengacu kepada penelitian-penelitian terdahulu yang terkait dengan pengembangan sistem pengelolaan parkiran serta hasil observasi yang didapatkan oleh penulis. Sehingga teknik yang dilakukan dalam pengumpulan data menggunakan teknik studi literatur dan observasi kepada *stakeholder* yang terkait dengan sistem.

Setelah mendapatkan data yang dibutuhkan, dilakukan verifikasi dan validasi kebutuhan sistem guna mendapatkan kebutuhan-kebutuhan dalam membangun sistem. Setelah melakukan identifikasi kebutuhan fungsional dan non fungsional maka dapat dilakukan identifikasi aktor. Ketiga elemen tersebut digunakan sebagai acuan pemodelan sistem dengan diagram *use case* beserta skenario *use case* sehingga dapat dihasilkan gambaran eksternal sebuah sistem ini.

## 3.4 Perancangan Sistem

Perancangan sistem pada penelitian ini dibuat berdasarkan hasil pemodelan diagram use case beserta skenario use case yang telah dilakukan pada tahap

analisis kebutuhan. Kedua elemen tersebut digunakan sebagai acuan dalam pemodelan diagram kelas, diagram aktivitas, dan diagram sekuen. Pada penelitian ini, perancangan sistem dibagi menjadi beberapa tahap dengan penjelasan sebagai berikut:

- 1. Pada tahap awal, penulis merancang desain arsitektural dari sistem. Di mana arsitektur yang digunakan adalah model *client-server*.
- 2. Pada tahap kedua akan dibuat diagram aktivitas yang mengacu kepada skenario *use case*, operasi diagram kelas, dan proses bisnis pengelolaan parkiran yang diperoleh dari hasil observasi. Diagram aktivitas tersebut akan dibuat pada setiap skenario *use case*.
- 3. Pada tahap ketiga, diagram sekuen akan dibuat dengan mengacu kepada alur skenario yang ada dalam skenario use case dan diagram aktivitas. Operasi yang telah didefinisikan pada diagram kelas digunakan sebagai acuan dalam pembuatan diagram sekuen. Diagram ini juga akan dibuat pada setiap skenario use case.
- 4. Pada tahap keempat, penulis merancang diagram kelas yang akan dibagi menjadi dua package yaitu model dan controller sebab sistem yang dikembangkan menggunakan pola arsitektur MVC (Model, View, Controller). Kelas pada controller merupakan kelas proses yang didapatkan dari diagram use case. Sedangkan kelas pada model merupakan kelas data yang berinteraksi dengan basis data sistem yang juga mengacu kepada diagram use case. Atribut kelas data didapatkan dari studi literatur penelitian terdahulu dan observasi terhadap data mengenai parkir yang diperoleh dari stakeholder yang terlibat dalam sistem.
- 5. Pada tahap kelima akan dilakukan perancangan basis data yang akan terhubung dengan sistem.
- 6. Pada tahap akhir perancangan sistem akan dilakukan pemodelan antarmuka untuk interaksi antara aktor terhadap sistem.

## 3.5 Implementasi Sistem

Implementasi sistem yang dilakukan pada penelitian ini mengacu kepada perancangan sistem. Implementasi sistem ini meliputi:

- 1. Pembuatan aplikasi android native untuk membaca kartu NFC yang akan digunakan sebagai android reader.
- 2. Pembuatan basis data pada MySQL *database server* untuk implementasi skema basis data yang telah dirancang.
- 3. Pembuatan kode *back end* yang berupa bahasa pemrograman PHP dengan *framework* Codelgniter. *Framework* Codelgniter dipilih karena sistem ini menggunakan pola arsitektur MVC (*Model, View, Controller*).

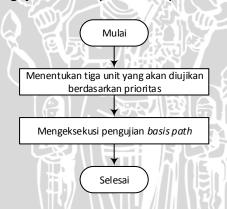
## 3.6 Pengujian dan Analisis Sistem

Pengujian perangkat lunak merupakan tahapan untuk mengetahui terpenuhinya kinerja dan performa sistem sesuai dengan kebutuhan yang telah ditentukan sebelumnya melalui serangkai proses uji. Ada tiga pengujian yang akan dilakukan yaitu pengujian white-box, pengujian black-box, dan pengujian performa.

## 3.6.1 Pengujian White-Box

Pengujian white-box adalah pengujian yang didasarkan pada pengecekan terhadap detail perancangan, menggunakan struktur kontrol dari desain program secara prosedural untuk membagi pengujian ke dalam beberapa kasus pengujian. Dalam pengujian white-box ini memakai pengujian unit untuk menguji algoritma pada fungsi yang ada dalam sistem untuk menemukan kesalahan.

Pengujian unit pada penelitian ini dilakukan untuk memastikan unit-unit yang memiliki prioritas tinggi memiliki hasil implementasi yang sesuai dengan harapan di awal analisis kebutuhan. Pengujian unit yang dilakukan dengan menggunakan teknik white box dengan jenis pengujian basis path. Unit yang akan diujikan pada pengujian ini akan dipilih berdasarkan tiga unit dengan prioritas tinggi. Tahapan yang dilakukan pada pengujian unit dapat dilihat pada Gambar 3.3.



Gambar 3.3 Tahapan Pengujian Unit

### 3.6.2 Pengujian Black-Box

Pengujian black-box adalah pengujian yang dilakukan untuk memeriksa hasil eksekusi aplikasi dan fungsionalitasnya untuk mengetahui sistem telah berjalan sesuai dengan perancangan. Dalam pengujian black-box ini memakai pengujian validasi untuk memeriksa hasil eksekusi dari sistem berfungsi seperti apa yang diharapkan.

### 3.6.2.1 Pengujian Validasi

Pengujian validasi pada penelitian ini dilakukan untuk memastikan seluruh spesifikasi kebutuhan perangkat lunak telah diimplementasikan sesuai dengan yang diharapkan. Tahapan yang dilakukan pada pengujian validasi dapat dilihat pada Gambar 3.4.



Gambar 3.4 Tahapan Pengujian Validasi

#### 3.6.2.2 Reliability

Pengujian karakteristik *reliability* pada penelitian ini fokus pada kriteria *sessions, pages,* dan *hits* dengan menggunakan *tool* WAPT (*Web Application Performance Tools*). Hasil dari pengujian dibandingkan dengan standar Telcordia. Gambar 3.5 merepresentasikan tahapan dalam bentuk diagram alir.



**Gambar 3.5 Tahapan Pengujian Reliability** 

### 3.6.2.3 Compatibility

Pengujian pada karakteristik compatibility pada penelitian ini dilakukan pada tujuh jenis dekstop browser dengan menggunakan tool BrowseEmAll. Untuk mendukung 80% pengguna sistem web browser diperlukan pengujian dengan tujuh versi web browser yang berbeda dari tiga browser terkenal (Salonen, 2012). Sistem memenuhi compatibility bila tidak ditemukan error pada hasil pengujian menggunakan tool BrowseEmAll. Gambar 3.6 memperlihatkan tahapan dari pengujian karakteristik compatibility.



**Gambar 3.6 Tahapan Pengujian Compatibility** 

## 3.6.3 Pengujian Performa

Pengujian performa dilakukan untuk menguji sistem berdasarkan waktu yang diperlukan dalam menjalankan proses bisnis yang telah direkayasa dalam sistem. Hal ini digunakan untuk melihat keoptimalan sistem yang telah dibangun bila dibandingkan dengan proses bisnis pada sistem parkir yang ada pada saat ini.

# 3.7 Pengambilan Kesimpulan dan Saran

Tahap akhir dalam penelitian ini adalah pengambilan kesimpulan. Tahap ini dilakukan setelah tahapan sebelumnya yaitu dari analisis kebutuhan hingga pengujian selesai. Tahap ini terdiri dari dua bagian yaitu kesimpulan dan saran. Kesimpulan digunakan untuk menjawab rumusan masalah yang telah diuraikan sebelumnya. Sedangkan saran digunakan sebagai perbaikan kesalahan-kesalahan yang terjadi dan penyempurnaan penulisan serta pertimbangan atas pengembangan sistem selanjutnya. Pengambilan kesimpulan dan saran tersebut mengacu kepada hasil dari pengujian dan analisis.

#### **BAB 4 HASIL**

Bab ini memaparkan perancangan dari sistem pengelolaan parkir menggunakan perangkat NFC berbasis android. Perancangan yang dilakukan meliputi dua tahap utama. Tahap pertama yang dilakukan adalah analisis kebutuhan. Kemudian tahap kedua yaitu perancangan sistem.

### 4.1 Gambaran Umum Sistem

NFCPS (NFC Parking System) adalah sistem pengelolaan parkir menggunakan perangkat NFC berbasis android merupakan sistem yang dapat membantu kegiatan operasional pengelolaan parkir. Sistem pengelolaan parkir ini merupakan sistem yang dibuat untuk mengelola aktivitas parkir yang sesuai dengan kebutuhan organisasi. Sistem ini dirancang berdasarkan hasil observasi dari penulis terhadap sistem parkir yang dijalankan di Universitas Brawijaya dengan sampel area parkir PTIIK depan serta dukungan dari studi literatur terhadap penelitian mengenai pengelolaan parkir.

Setiap pengguna yang akan keluar-masuk area parkir harus memiliki NFC smartcard sebagai identitas dari kendaraan. Pengguna fasilitas parkir harus mendaftarkan kendaraan yang digunakan kepada koordinator parkir agar dapat teridentifikasi dalam sistem. Dalam melakukan pendaftaran, pengguna diwajibkan membawa fotokopi identitas semisal ktm bagi mahasiswa atau kartu pegawai bagi dosen/pegawai dan membawa juga fotokopi stnk sebagai syarat untuk mendaftar. Setelah didaftarkan maka pengguna akan mendapatkan smartcard NFC sebagai identitas bagi kendaraan.

Sistem ini berbasis web untuk pengelolaan data yang dilakukan, dan untuk aktivitas parkir seperti masuk-keluar kendaraan digunakan *smartphone* berbasis android menggunakan perangkat NFC sebagai pembaca identitas kendaraan yang terdapat pada kartu yang dimiliki oleh pengguna. Di samping itu, sistem ini dioperasikan pada jaringan lokal agar waktu respon lebih cepat dan stabil.

#### 4.2 Analisis Kebutuhan

Tahap analisis kebutuhan pada penelitian ini diawali dengan elisitasi kebutuhan, identifikasi aktor yang terlibat dalam sistem, pendefinisian daftar spesifikasi kebutuhan yang kemudian ditranslasikan ke dalam diagram *use case*. Tahapan ini dilakukan untuk menggambarkan kebutuhan-kebutuhan yang dimiliki oleh sistem agar dapat memenuhi kebutuhan sistem pengelolaan parkir pada umumnya. Gambar 4.1 merepresentasikan tahapan dari analisis kebutuhan.



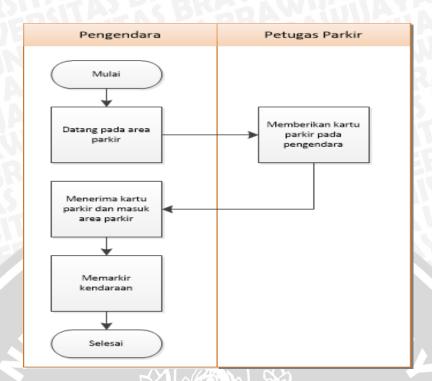
Gambar 4.1 Diagram Alir Analisis Kebutuhan

#### 4.2.1 Elisitasi Kebutuhan

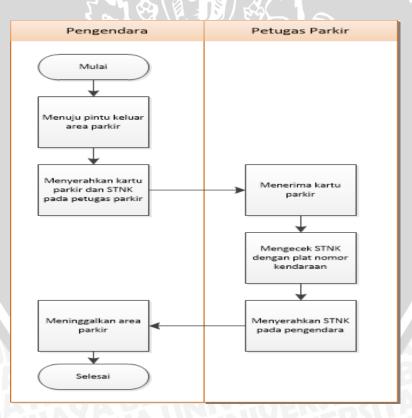
Tahap ini dilakukan untuk mengetahui masalah yang perlu dipecahkan dan kebutuhan dari sistem. Teknik yang digunakan dalam elisitasi kebutuhan adalah studi literatur (literature review) dan observasi stakeholder. Studi literatur yang dilakukan mengacu kepada empat penelitian terdahulu yang telah dipaparkan pada bagian Kajian Pustaka, sedangkan observasi stakeholder dilakukan dengan wawancara dengan koordinator parkir untuk mengetahui proses bisnis yang terjadi dalam aktivitas pengelolaan parkir.

Berdasarkan hasil observasi penulis, dalam sistem parkir yang ada pada Universitas Brawijaya terdapat 48 area parkir dengan masing-masing area parkir dijaga oleh minimal 2 petugas parkir dengan 2 kali shift dalam sehari. Dari 48 area parkir tersebut dapat menampung sekitar 1810 kendaraan baik itu roda dua maupun roda empat. Area parkir yang dapat menampung kendaraan roda dua berjumlah 39 area parkir. Dalam tiap area parkir tersebut dilengkapi dengan kartu parkir bagi pengguna lahan parkir yang disesuaikan dengan kapasitas tiap area parkir.

Prosedur masuk area parkir dan keluar area parkir yang masih memakai sistem yang manual dapat dilihat pada Gambar 4.2 dan 4.3.



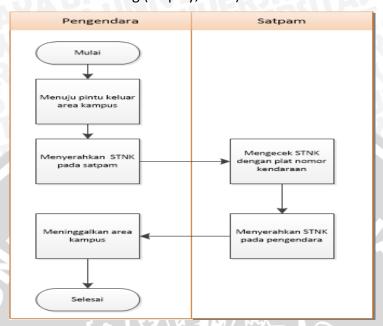
Gambar 4.2 Prosedur Masuk Area Parkir pada UB



Gambar 4.3 Prosedur Keluar Area Parkir pada UB

Berdasarkan pada gambar 4.2 dan 4.3 tersebut, untuk keluar-masuk area parkir diperlukan kartu parkir dan pengecekan STNK. Permasalahan yang muncul adalah ketika jumlah kartu yang tersedia tidak seimbang dengan jumlah

kendaraan yang masuk pada area parkir. Hal ini membuat petugas parkir membolehkan kendaraan masuk tanpa kartu parkir. Selain itu juga kartu parkir seringkali tidak diserahkan oleh pengendara pada petugas parkir sehingga kartu yang tersedia semakin berkurang (Display, 2015).



Gambar 4.4 Prosedur Keluar Area Kampus pada UB

Berdasarkan pada gambar 4.4 tersebut, untuk keluar area kampus diperlukan pengecekan STNK. Permasalahan yang muncul adalah ketika pengecekan STNK yang dilakukan oleh satpam tidak detail sehingga bila curanmor sudah mengganti plat nomor kendaraan dan memalsukan STNK maka kendaraan dapat melenggang keluar gerbang kampus (Jauhari, 2015). Seringkali STNK diletakkan pada dompet atau pada tempat STNK yang bila dikeluarkan untuk pengecekan oleh satpam, maka pengendara membutuhkan waktu lebih lama untuk mengeluarkannya. Bagi satpam untuk mengecek STNK yang berada pada dompet atau tempat STNK terkadang mengalami kesulitan karena detail pada STNK tidak terlihat akibat STNK terlipat atau terbatas oleh tempat STNK itu sendiri.

Berdasarkan hasil dari teknik studi literatur dan observasi maka ditemukan kebutuhan yang harus ada dalam sistem pengelolaan parkir yaitu meliputi:

- 1. Area fungsional pengendara
  - a. Sistem harus menyediakan fitur untuk keluar-masuk area parkir (SKPL 01).
  - b. Sistem harus menyediakan fitur untuk keluar area kampus (SKPL 02).
  - c. Sistem harus menyediakan fitur untuk autentifikasi pengguna dengan tujuan agar sistem dapat dioperasikan oleh pengguna yang berwenang (SKPL 03).
- 2. Area fungsional petugas parkir
  - a. Sistem harus menyediakan fitur untuk menginput kendaraan yang keluarmasuk dari area parkir (SKPL 04).

- b. Sistem harus menyediakan fitur untuk menampilkan aktivitas keluar-masuk kendaraan pada area parkir (SKPL 05).
- c. Sistem harus menyediakan fitur untuk membuat laporan kehilangan kendaraan (SKPL\_06).
- d. Sistem harus menyediakan fitur untuk membuat laporan ditemukan barang (SKPL 07).
- e. Sistem harus menyediakan fitur untuk mengklaim laporan ditemukan barang (SKPL 08).
- f. Sistem harus menyediakan fitur untuk mengedit profil (SKPL 09).
- g. Sistem harus menyediakan fitur untuk mengirim ulang password (SKPL 10).

#### 3. Area fungsional keamanan

- a. Sistem harus menyediakan fitur untuk memblokir kendaraan dan mencabut blokir pada kendaraan (SKPL\_11).
- b. Sistem harus menyediakan fungsi untuk memproses laporan kehilangan (SKPL 12).
- c. Sistem harus menyediakan fitur untuk menginput kendaraan keluar area kampus (SKPL 13).
- d. Sistem harus menyediakan fitur untuk membuat laporan ditemukan barang (SKPL\_07).
- e. Sistem harus menyediakan fitur untuk mengklaim laporan ditemukan barang (SKPL 08).
- f. Sistem harus menyediakan fitur untuk mengedit profil (SKPL\_09).
- g. Sistem harus menyediakan fitur untuk mengirim ulang password (SKPL 10).

#### 4. Area fungsional koordinator parkir

- a. Sistem harus menyediakan fitur untuk mengelola data pengendara (SKPL 14).
- b. Sistem harus menyediakan fitur untuk mengelola data kendaraan (SKPL 15).
- c. Sistem harus menyediakan fitur untuk mengelola data petugas parkir (SKPL 16).
- d. Sistem harus menyediakan fitur untuk memvalidasi laporan kehilangan (SKPL 17).
- e. Sistem harus menyediakan fitur untuk mengelola area parkir (SKPL 18).
- f. Sistem harus menyediakan fitur untuk membuat laporan ditemukan barang (SKPL\_07).
- g. Sistem harus menyediakan fitur untuk memvalidasi laporan ditemukan barang (SKPL 19).
- h. Sistem harus menyediakan fitur untuk mengedit profil (SKPL 09).
- i. Sistem harus menyediakan fitur untuk mengirim ulang password (SKPL\_10).

#### 4.2.2 Identifikasi Aktor

Tahap ini dilakukan untuk mengidentifikasi aktor yang terlibat secara langsung terhadap sistem yang dikembangkan. Pengidentifikasian aktor pada tahap ini mengacu kepada tahap elisitasi kebutuhan. Tabel 4.1 merepresentasikan aktor-aktor tersebut beserta deskripsinya.

Tabel 4.1 Identifikasi Aktor

Aktor	Deskripsi
Pengendara	Pengendara merupakan pengguna yang hanya memiliki hak akses untuk keluar-masuk area parkir serta keluar area kampus.
Petugas Parkir	Petugas Parkir merupakan pengguna yang hanya memiliki hak akses pada bagian petugas parkir.
Petugas Keamanan	Petugas Keamanan merupakan pengguna yang memiliki hak untuk mengakses bagian petugas keamanan.
Koordinator Parkir	Koordinator parkir merupakan pengguna yang memiliki hak untuk mengakses bagian koordinator parkir.

## 4.2.3 Spesifikasi Kebutuhan

Spesifikasi kebutuhan yang disediakan pada sistem ini mengacu kepada elisitasi kebutuhan pada bagian sebelumnya. Penentuan prioritas pada daftar spesifikasi kebutuhan fungsional didasarkan atas tingkat keterlibatannya terhadap proses bisnis yang telah diusulkan pada tahap elisitasi kebutuhan. Tabel 4.2 menjelaskan spesifikasi kebutuhan fungsional, sedangkan Tabel 4.3 menjelaskan spesifikasi kebutuhan non fungsional dari sistem ini.

**Tabel 4.2 Spesifikasi Kebutuhan Fungsional** 

Tabel 4.2 Spesifikasi Kebutulian Fungsional				
Nomor SKPL	Kebutuhan	Use Case	Aktor	Prioritas
SKPL_01_ 1	Sistem dapat menerima kendaraan masuk area parkir dengan kartu NFC sebagai identitas	Masuk Area Parkir	Pengendara	Tinggi
SKPL_01_ 2	Sistem dapat menerima kendaraan keluar area parkir dengan kartu NFC sebagai identitas	Keluar Area Parkir	Pengendara	Tinggi
SKPL_02	Sistem dapat menerima kendaraan keluar area kampus dengan kartu NFC sebagai identitas	Keluar Area Kampus	Pengendara	Tinggi

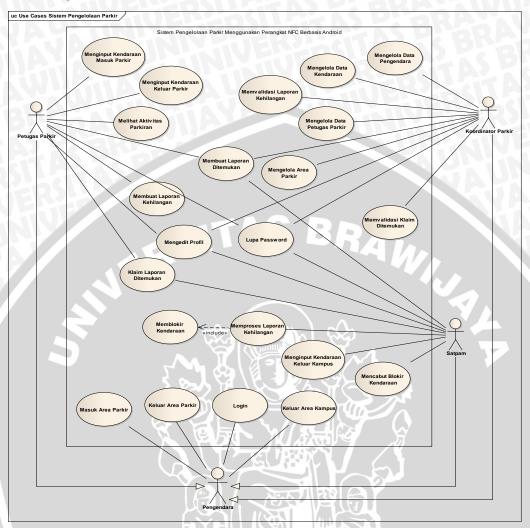
SKPL_03	Sistem harus menyediakan fitur autentifikasi pengguna	Login	Pengendara	Tinggi
SKPL_04_ 1	Sistem dapat menginput kendaraan yang masuk area parkir	Menginput Kendaraan Masuk Parkir	Petugas Parkir	Tinggi
SKPL_04_ 2	Sistem dapat menginput kendaraan yang keluar area parkir	Menginput Kendaraan Keluar Parkir	Petugas Parkir	Tinggi
SKPL_05	Sistem dapat melihat aktivitas keluar-masuk kendaraan pada area parkir	Melihat Aktivitas Parkir	Petugas Parkir	Sedang
SKPL_06	Sistem dapat membuat laporan kehilangan kendaraan	Membuat Laporan Kehilangan	Petugas Parkir	Tinggi
SKPL_07	Sistem dapat membuat laporan ditemukan barang	Membuat Laporan Ditemukan	Petugas Parkir, Petugas Keamanan, Koordinator Parkir	Sedang
SKPL_08	Sistem dapat membuat klaim laporan ditemukan barang	Klaim Laporan Ditemukan	Petugas Parkir, Petugas Keamanan	Sedang
SKPL_09	Sistem dapat mengedit profil	Mengedit Profil	Petugas Parkir, Petugas Keamanan, Koordinator Parkir	Rendah
SKPL_10	Sistem dapat mengirim ulang password	Lupa Password	Petugas Parkir, Petugas Keamanan, Koordinator Parkir	Rendah
SKPL_11_ 1	Sistem dapat memberikan blokir kepada kendaraan agar tidak dapat keluar	Blokir Kendaraan	Petugas Keamanan	Tinggi
AS E	area kampus			

2	blokir kepada kendaraan	Kendaraan	Keamanan	VATTV
SKPL_12	Sistem dapat memproses laporan kehilangan yang telah divalidasi	Memproses Laporan Kehilangan	Petugas Keamanan	Tinggi
SKPL_13	Sistem dapat menginput kendaraan yang akan keluar kampus	Menginput Kendaraan Keluar Kampus	Petugas Keamanan	Tinggi
SKPL_14	Sistem dapat mengelola data pengendara	Mengelola Data Pengendara	Koordinator Parkir	Tinggi
SKPL_15	Sistem dapat mengelola data kendaraan	Mengelola Data Kendaraan	Koordinator Parkir	Tinggi
SKPL_16	Sistem dapat mengelola data petugas parkir	Mengelola Data Petugas Parkir	Koordinator Parkir	Sedang
SKPL_17	Sistem dapat memvalidasi laporan kehilangan kendaraan	Memvalidasi Laporan Kehilangan	Koordinator Parkir	Tinggi
SKPL_18	Sistem dapat mengelola data area parkir	Mengelola Area Parkir	Koordinator Parkir	Tinggi
SKPL_19	Sistem dapat memvalidasi laporan ditemukan barang	Memvalidasi Laporan Ditemukan	Koordinator Parkir	Sedang

**Tabel 4.3 Spesifikasi Kebutuhan Non Fungsional** 

Nomor SKPL	Parameter	Deskripsi Kebutuhan
SKPL_20	Reliability	Sistem dapat melayani perintah pengguna dengan batas toleransi <i>error</i> hingga 5%.
SKPL_21	Compatibility	Sistem dapat dijalankan dalam berbagai browser yang berbeda

### 4.2.4 Diagram Use Case



Gambar 4.5 Diagram Use Case Sistem

Pada gambar 4.5 menggambarkan diagram *use case* dari sistem secara keseluruhan. Diagram *use case* merupakan salah satu diagram UML (*Unified Modeling Language*) yang memodelkan perilaku sistem dari tampak luar. Diagram ini menggambarkan aktor, *use case*, dan hubungan terkait keduanya pada sistem yang dibangun.

Setiap *use case* yang dimodelkan akan dijelaskan lebih detail pada bagian skenario *use case*. Tabel skenario *use case* terdiri dari nomor SKPL, nama *use case* tersebut, aktor yang menggunakannya, tujuan dan deskripsi *use case*, beserta skenario utama dan skenario alternatif yang dilakukan ketika menggunakan *use case* tersebut.

#### 1. Use Case Masuk Area Parkir

*Use case* ini menjelaskan alur untuk masuk area parkir secara detail. Skenario masuk area parkir dapat dilihat pada Tabel 4.4.

**Tabel 4.4 Skenario Masuk Area Parkir** 

Kode SKPL	SKPL_01_1	
Aktor	Pengendara	
Tujuan	Masuk area parkir dengan menggunakan kartu	
	NFC sebagai identitas	
Deskripsi	Use case ini mendeskripsikan alur dari proses	
AS REBRAS	masuk area parkir oleh aktor	
Kondisi Awal	Aktor menyiapkan kartu NFC	
Main Flow	Aktor mendekatkan kartu pada android	
	reader	
y/ cR	2. Sistem menampilkan kolom uid yang berisi	
	identitas pengguna	
	3. Aktor memilih menu masuk area parkir	
	4. Sistem akan menginput masuk area parkir	
	dan aplikasi akan menutup otomatis	
Kondisi Akhir	Data telah terinput pada sistem	

## 2. Use Case Keluar Area Parkir

Use case ini menjelaskan alur untuk keluar area parkir secara detail. Skenario keluar area parkir dapat dilihat pada Tabel 4.5.

Tabel 4.5 Skenario Keluar Area Parkir

Tabel	1 4.5 Skenario Keluar Area Parkir	
Kode SKPL	SKPL_01_2	
Aktor	Pengendara	
Tujuan	Keluar area parkir dengan menggunakan kartu	
	NFC sebagai identitas	
Deskripsi	Use case ini mendeskripsikan alur dari proses	
## \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \	keluar area parkir oleh aktor	
Kondisi Awal	Aktor menyiapkan kartu NFC	
Main Flow	Aktor mendekatkan kartu pada android	
	reader	
RITIARI	2. Sistem menampilkan kolom uid yang berisi	
SHAWKIII	identitas pengguna	
PUBRAGAI	3. Aktor memilih menu keluar area parkir	
ATAS POB	4. Sistem akan menginput keluar area parkir dan	

	aplikasi akan menutup otomatis	HTT
Kondisi Akhir	Data telah terinput pada sistem	

### 3. Use Case Keluar Area Kampus

*Use case* ini menjelaskan alur untuk keluar area kampus secara detail. Skenario keluar area kampus dapat dilihat pada Tabel 4.6.

**Tabel 4.6 Skenario Keluar Area Kampus** 

Tabel 4.0 Skellario keluar Area kampus		
Kode SKPL	SKPL_02	
Aktor	Pengendara	
Tujuan	Keluar area kampus dengan menggunakan kartu	
	NFC sebagai identitas	
Deskripsi	Use case ini mendeskripsikan alur dari proses	
	keluar area kampus oleh aktor	
Kondisi Awal	Aktor menyiapkan kartu NFC	
Main Flow	<ol> <li>Aktor mendekatkan kartu pada android reader</li> <li>Sistem menampilkan kolom uid yang berisi identitas pengguna</li> <li>Aktor memilih menu keluar area kampus</li> </ol>	
Kondisi Akhir	4. Sistem akan menginput keluar area kampus dan aplikasi akan menutup otomatis  Data telah terinput pada sistem	
Kondisi / Kim	Bata telah telihipat pada sistem	

## 4. Use Case Login

Use case ini menjelaskan alur untuk autentifikasi pengguna secara detail. Skenario login dapat dilihat pada Tabel 4.7.

**Tabel 4.7 Skenario Login** 

Kode SKPL	SKPL_03
Aktor	Pengendara
Tujuan	Agar sistem dapat mengenali pengguna yang
	mengoperasikan sistem
Deskripsi	Use case ini mendeskripsikan alur dari proses
	autentifikasi pengguna oleh aktor
Kondisi Awal	Aktor memiliki username dan password

Main Flow	1. Aktor mengakses halaman login pada sistem
Printyfile	2. Sistem menampilkan form login
<b>AUAUPIN</b>	3. Aktor mengisi form login dengan username
litiay Aua	dan password
	4. Aktor menekan tombol login
Brandwill	5. Sistem akan menampilkan notifikasi berhasil
ALKS BROSS	login
Alternative Flow	5.1 Bila data yang diinputkan tidak terdaftar pada
	sistem maka akan muncul notifikasi gagal
	login S B R
Kondisi Akhir	Aktor telah masuk dalam sistem

# 5. Use Case Menginput Kendaraan Masuk Area Parkir

Use case ini menjelaskan alur untuk menginput kendaraan masuk secara detail. Skenario menginput kendaraan masuk dapat dilihat pada Tabel 4.8.

Tabel 4.8 Skenario Menginput Kendaraan Masuk Area Parkir

Kode SKPL	SKPL_04_1
Aktor	Petugas Parkir
Tujuan	Menginput kendaraan yang masuk area parkir
Deskripsi	Use case ini mendeskripsikan alur dari proses menginput kendaraan masuk oleh petugas parkir
Kondisi Awal	Petugas parkir telah login pada sistem
Main Flow	<ol> <li>Aktor memilih menu input kendaraan masuk</li> <li>Aktor melihat pada list data kendaraan yang diperoleh dari android reader</li> <li>Aktor menekan tombol proses</li> <li>Muncul notifikasi dari sistem kendaraan berhasil masuk area parkir</li> </ol>
Alternative Flow	4.1 Bila gagal, pada sistem maka akan muncul notifikasi kendaraan diblokir
Kondisi Akhir	Data telah terinput pada sistem

## 6. Use Case Menginput Kendaraan Keluar Area Parkir

Use case ini menjelaskan alur untuk menginput kendaraan keluar area parkir secara detail. Skenario menginput kendaraan masuk dapat dilihat pada Tabel 4.9.

**Tabel 4.9 Skenario Menginput Kendaraan Keluar Area Parkir** 

Kode SKPL	SKPL_04_2
Aktor	Petugas Parkir
Tujuan	Menginput kendaraan yang keluar area parkir
Deskripsi	Use case ini mendeskripsikan alur dari proses
BRADAWNI	menginput kendaraan keluar oleh petugas parkir
Kondisi Awal	Petugas parkir telah login pada sistem
Main Flow	Aktor memilih menu input kendaraan keluar
E	2. Aktor melihat pada list data kendaraan yang
	diperoleh dari reader
Y/ CR	3. Aktor menekan tombol proses
	4. Sistem memunculkan detail data parkir
	5. Aktor menekan tombol submit
	6. Muncul notifikasi dari sistem kendaraan
7	berhasil keluar area parkir
Alternative Flow	6.1 Bila gagal, pada sistem akan muncul notifikasi
	kendaraan diblokir
Kondisi Akhir	Data telah terinput pada sistem

## 7. Use Case Melihat Aktivitas Parkir

Use case ini menjelaskan alur untuk melihat aktivitas parkir secara detail. Skenario melihat aktivitas parkir dapat dilihat pada Tabel 4.10.

**Tabel 4.10 Skenario Melihat Aktivitas Parkir** 

Kode SKPL	SKPL_05
Aktor	Petugas Parkir O O
Tujuan	Memonitor aktivitas dalam area parkir
Deskripsi	Use case ini mendeskripsikan alur dari proses
	melihat aktivitas parkiran oleh petugas parkir
Kondisi Awal	Petugas parkir telah login pada sistem
Main Flow	Aktor memilih menu list aktivitas parkir
KWUAHAY	Sistem menampilkan hasil berupa data
GRAY WILL	aktivitas yang terjadi dalam area parkir
Alternative Flow	2.1 Bila data aktivitas parkir tidak ada maka pada
CITAL PEB	sistem akan muncul notifikasi

Kondisi Akhir	Petugas parkir dapat memonitor aktivitas yang
HINKIU	terjadi pada area parkir

## 8. Use Case Membuat Laporan Kehilangan

Use case ini menjelaskan alur untuk membuat laporan kehilangan secara detail. Skenario membuat laporan kehilangan dapat dilihat pada Tabel 4.11.

**Tabel 4.11 Skenario Membuat Laporan Kehilangan** 

enario iviembuat Laporan Kenilangan
SKPL_06
Petugas Parkir
Membuat laporan kehilangan kendaraan
Use case ini mendeskripsikan alur dari proses
membuat laporan kehilangan oleh petugas parkir
Petugas parkir telah login pada sistem
Aktor memilih menu buat laporan kehilangan
2. Aktor mencari data pelapor dengan
menginputkan no id
3. Sistem menampilkan data pelapor dengan
kendaraan yang hilang
4. Aktor mengisi form kemudian tekan submit
5. Sistem menampilkan notifikasi pembuatan
laporan kehilangan berhasil
5.1 Bila laporan yang dikirimkan tidak diterima
oleh sistem maka akan muncul notifikasi
pembuatan laporan kehilangan gagal
Petugas parkir telah membuat laporan kehilangan

## 9. Use Case Membuat Laporan Ditemukan

Use case ini menjelaskan alur untuk membuat laporan ditemukan barang secara detail. Skenario membuat laporan ditemukan dapat dilihat pada Tabel 4.12.

**Tabel 4.12 Skenario Membuat Laporan Ditemukan** 

Kode SKPL	SKPL_07
Aktor	Petugas Parkir, Petugas Keamanan, Koordinator
BRANKUI	Parkir
Tujuan	Membuat laporan ditemukan barang

Deskripsi	Use case ini mendeskripsikan alur dari proses membuat laporan ditemukan barang oleh aktor
Kondisi Awal	Aktor telah login pada sistem
Main Flow	<ol> <li>Aktor memilih menu buat laporan ditemukan</li> <li>Aktor mengisi form buat laporan ditemukan</li> <li>Aktor kemudian tekan submit</li> <li>Sistem menampilkan notifikasi membuat laporan ditemukan berhasil</li> </ol>
Alternative Flow	4.1 Bila gagal membuat laporan ditemukan maka akan muncul notifiaksi
Kondisi Akhir	Aktor berhasil membuat laporan ditemukan

# 10. Use Case Klaim Laporan Ditemukan

*Use case* ini menjelaskan alur untuk membuat klaim laporan ditemukan secara detail. Skenario membuat klaim laporan ditemukan dapat dilihat pada Tabel 4.13.

**Tabel 4.13 Skenario Klaim Laporan Ditemukan** 

Kode SKPL	SKPL_08
Aktor	Petugas Parkir, Petugas Keamanan
Tujuan	Membuat klaim laporan ditemukan
Deskripsi	Use case ini mendeskripsikan alur dari proses
	klaim laporan ditemukan barang oleh aktor
Kondisi Awal	Aktor telah login pada sistem
Main Flow	Aktor memilih menu list laporan ditemukan
4	2. Aktor memilih data yang akan diklaim,
5	kemudian tekan klaim
###. <b>\</b>	3. Aktor kemudian mengisi form, kemudian
	tekan submit
	4. Sistem menampilkan notifikasi klaim laporan
HYLLYTO	ditemukan berhasil
Alternative Flow	4.1 Bila gagal klaim laporan ditemukan maka
SEAWIGH	akan muncul notifiaksi
Kondisi Akhir	Aktor berhasil mengklaim laporan ditemukan

## 11. Use Case Mengedit Profil

*Use case* ini menjelaskan alur untuk mengedit profil secara detail. Skenario membuat mengedit profil dapat dilihat pada Tabel 4.14.

**Tabel 4.14 Skenario Mengedit Profil** 

Kode SKPL	SKPL_09
Aktor	Petugas Parkir, Petugas Keamanan, Koordinator
	Parkir
Tujuan	Mengedit profil
Deskripsi	Use case ini mendeskripsikan alur dari proses
	mengedit profil oleh aktor
Kondisi Awal	Aktor telah login pada sistem
Main Flow	Aktor memilih menu edit profil
<b>3</b>	2. Aktor mengubah username dan password
5	pada menu edit profil
	3. Aktor kemudian tekan submit
	4. Sistem menampilkan notifikasi edit profil
	berhasil
Alternative Flow	4.1 Bila gagal edit profil maka akan muncul
	notifiaksi
Kondisi Akhir	Aktor berhasil mengedit profil

## 12. *Use Case* Lupa Password

Use case ini menjelaskan alur untuk mengirim ulang password secara detail. Skenario mengirim ulang password dapat dilihat pada Tabel 4.15.

**Tabel 4.15 Skenario Lupa Password** 

er 1125 Skeriario Eapa i assirora
SKPL_10
Petugas Parkir, Petugas Keamanan, Koordinator
Parkir
Mengirim ulang password
Use case ini mendeskripsikan alur dari proses
mengirim ulang password oleh aktor
Aktor membuka halaman login
Aktor memilih menu lupa password

MIVERERS	2. Aktor menginputkan email dan tekan submit
HUNKTUE	3. Sistem menampilkan notifikasi email
GUAUSTIN	ditemukan
Litiay Aua	4. Aktor kemudian tekan submit
	5. Sistem menampilkan notifikasi data berhasil
BRANNI	dikirim
Alternative Flow	5.1 Bila gagal terkirim muncul error
Kondisi Akhir	Password berhasil dikirim ulang pada email yang
EHE	terdaftar pada sistem

# 13. Use Case Memblokir Kendaraan

Use case ini menjelaskan alur untuk memblokir kendaraan secara detail. Skenario memblokir kendaraan dapat dilihat pada Tabel 4.16.

Tabel 4.16 Skenario Memblokir Kendaraan

Kode SKPL	SKPL_11_1
Aktor	Petugas Keamanan
Tujuan	Memblokir kendaraan agar tidak dapat keluar area kampus
Deskripsi	Use case ini mendeskripsikan alur dari proses blokir kendaraan oleh keamanan
Kondisi Awal	Aktor telah login pada sistem, dan sedang memproses laporan kehilangan
Main Flow	<ol> <li>Aktor melakukan proses laporan kehilangan</li> <li>Sistem menampilkan data detail laporan kehilangan</li> <li>Aktor menekan tombol proses.</li> <li>Sistem akan memproses laporan kehilangan dan memblokir kendaraan.</li> <li>Sistem akan menampilkan notifikasi atas perubahan data yang dilakukan oleh aktor</li> </ol>
Alternative Flow	<ul><li>2.1 Bila data kendaraan tidak muncul pada sistem maka akan muncul notifikasi</li><li>5.1 Bila perubahan data tidak berhasil pada</li></ul>

JIVERERS	sistem maka akan muncul notifikasi
Kondisi Akhir	Kendaraan telah diblokir

## 14. Use Case Mencabut Blokir Kendaraan

Use case ini menjelaskan alur untuk mencabut blokir kendaraan secara detail. Skenario mencabut blokir kendaraan dapat dilihat pada Tabel 4.17.

**Tabel 4.17 Skenario Mencabut Blokir Kendaraan** 

Kode SKPL	SKPL_11_2
Aktor	Petugas Keamanan
Tujuan	Mencabut blokir yang diberikan kepada
	kendaraan
Deskripsi	Use case ini mendeskripsikan alur dari proses
	mencabut blokir kendaraan oleh keamanan
Kondisi Awal	Aktor telah login pada sistem, dan laporan
5	kehilangan sudah diproses
Main Flow	Aktor memilih menu list blokir kendaraan
	2. Sistem menampilkan data list kendaraan yang
St.	sedang diblokir
	3. Aktor memilih data kendaraan yang akan
	dicabut pemblokirannya.
	4. Pilih tombol cabut blokir kendaraan.
	5. Sistem akan menampilkan notifikasi atas
	perubahan data yang dilakukan oleh
	keamanan
Alternative Flow	2.1 Bila data kendaraan tidak muncul pada
B	sistem maka akan muncul notifikasi
TVE	5.1 Bila perubahan data tidak berhasil pada
	sistem maka akan muncul notifikasi
Kondisi Akhir	Aktor dapat mencabut blokir terhadap kendaraan

### 15. Use Case Memproses Laporan Kehilangan

*Use case* ini menjelaskan alur untuk memproses laporan kehilangan secara detail. Skenario mememproses laporan kehilangan dapat dilihat pada Tabel 4.18.

Tabel 4.18 Skenario Memproses Laporan Kehilangan

Kode SKPL	SKPL_12
Aktor	Petugas Keamanan
Tujuan	Memproses laporan kehilangan dan memblokir
PHATE.	kendaraan agar tidak dapat keluar area kampus
Deskripsi	Use case ini mendeskripsikan alur dari proses
	memproses laporan kehilangan oleh aktor
Kondisi Awal	Aktor telah login pada sistem
Main Flow	Aktor memilih menu list laporan kehilangan
2	masuk
	2. Sistem menampilkan list data laporan
1	kehilangan masuk beserta pilihan menu
	3. Aktor memilih tombol lakukan tindakan
8	4. Sistem menampilkan isi laporan kehilangan,
	kemudian tekan submit
	5. Sistem akan menampilkan notifikasi atas
	pemrosesan laporan oleh aktor
Alternative Flow	2.1 Bila data laporan kehilangan tidak muncul
	pada sistem maka akan muncul notifikasi
	5.1 Bila pemrosesan laporan kehilangan tidak
	berhasil pada sistem maka akan muncul
	notifikasi
Kondisi Akhir	Aktor dapat memproses laporan kehilangan serta
	memblokir kendaraan yang dilaporkan hilang

## 16. Use Case Menginput Kendaraan Keluar Area Kampus

*Use case* ini menjelaskan alur untuk menginput kendaraan keluar area kampus secara detail. Skenario menginput kendaraan keluar area kampus dapat dilihat pada Tabel 4.19.

**Tabel 4.19 Skenario Menginput Kendaraan Keluar Area Kampus** 

Kode SKPL	SKPL_13
Aktor	Petugas Keamanan
Tujuan	Menginput kendaraan yang keluar area kampus
Deskripsi	Use case ini mendeskripsikan alur dari proses
AS PUBRAY	menginput kendaraan keluar area kampus oleh
SILATAS	aktor
Kondisi Awal	Aktor telah login pada sistem
Main Flow	Aktor memilih menu input kendaraan keluar
Y/ CR	kampus
	2. Aktor melihat pada list data kendaraan yang
	diperoleh dari reader
	3. Aktor menekan tombol proses
7	4. Sistem memunculkan detail data parkir
, <u>,</u> ,	5. Aktor menekan tombol submit
	6. Muncul notifikasi dari sistem kendaraan
7	berhasil keluar area kampus
Alternative Flow	6.1 Bila gagal, pada sistem akan muncul notifikasi
	kendaraan diblokir
Kondisi Akhir	Data telah terinput pada sistem dan kendaraan
	berhasil keluar area kampus

# 17. Use Case Mengelola Data Pengendara

Use case ini menjelaskan alur untuk mengelola data pengendara secara detail. Skenario memengelola data pengendara dapat dilihat pada Tabel 4.20.

**Tabel 4.20 Skenario Mengelola Data Pengendara** 

	2 Sheriano mengerola 2 ata 1 engenaara
Kode SKPL	SKPL_14
Aktor	Koordinator Parkir
Tujuan	Mengelola data pengendara dalam sistem
Deskripsi	Use case ini mendeskripsikan alur dari proses mengelola data pengendara oleh koordinator parkir
Kondisi Awal	koordinator parkir telah login pada sistem

Main Flow	1. Koordinator parkir memilih menu kelola data
HINKTO	pengendara
AUAUSTN	2. Sistem menampilkan list data pengendara
!!TilAY PJA	beserta pilihan menu
	3. Koordinator parkir memilih salah satu menu
SKARAVKI	4. Sistem akan menampilkan menu yang dipilih
ALK BRASS	oleh petugas parkir
LATA	5. Sistem akan menampilkan notifikasi atas
	perubahan data yang dilakukan oleh
	koordinator parkir
Alternative Flow	2.1 Bila data pengendara tidak muncul pada
	sistem maka akan muncul notifikasi
	4.1 Bila ingin menambah pengendara pilih
	menu tambah pengendara, kemudian isi form
7	M X (In the last of the last o
, E	yang tersedia dan tekan submit
	4.2 Bila ingin mengedit pengendara pilih
7	tombol edit pengendara, kemudian isi form
$\checkmark$	yang tersedia dan tekan submit
	4.3 Bila ingin menghapus pengendara pilil
	tombol hapus pengendara kemudian data
	telah terhapus
	5.1 Bila perubahan data tidak berhasil pada
	sistem maka akan muncul notifikasi
Kondisi Akhir	Koordinator parkir dapat mengelola data
KUHUISI AKHII	
IVE	pengendara

# 18. Use Case Mengelola Data Kendaraan

Use case ini menjelaskan alur untuk mengelola data kendaraan secara detail. Skenario mengelola data kendaraan dapat dilihat pada Tabel 4.21.

**Tabel 4.21 Skenario Mengelola Data Kendaraan** 

Kode SKPL	SKPL_15
Aktor	Koordinator Parkir

Tujuan	Mengelola data kendaraan dalam sistem
Deskripsi	Use case ini mendeskripsikan alur dari proses
AVALLATIN.	mengelola data kendaraan oleh koordinator
KITTAYAVA	parkir
Kondisi Awal	koordinator parkir telah login pada sistem
Main Flow	1. Koordinator parkir memilih menu kelola data
HAS PROP	kendaraan
263112	2. Sistem menampilkan list data kendaaraan
Hite	beserta pilihan menu
	3. Koordinator parkir memilih salah satu menu
En	4. Sistem akan menampilkan menu yang dipilih
	oleh petugas parkir
5	5. Sistem akan menampilkan notifikasi atas
	perubahan data yang dilakukan oleh
ξ'.	koordinator parkir
Alternative Flow	2.1 Bila data kendaraan tidak muncul pada
S. C.	sistem maka akan muncul notifikasi
	4.1 Bila ingin menambah kendaraan pilih
	menu tambah kendaraan, kemudian isi form
	yang tersedia dan tekan submit
	4.2 Bila ingin mengedit kendaraan pilih
	tombol edit, kemudian isi form yang tersedia
	dan tekan submit
	4.3 Bila ingin menghapus kendaraan pilih
B	tombol hapus kemudian data telah terhapus
IVA	5.1 Bila perubahan data tidak berhasil pada
UNH	
	sistem maka akan muncul notifikasi
Kondisi Akhir	Koordinator parkir dapat mengelola data
AWUSTIAY	kendaraan

# 19. Use Case Mengelola Data Petugas Parkir

Use case ini menjelaskan alur untuk mengelola data petugas parkir secara detail. Skenario mengelola data petugas parkir dapat dilihat pada Tabel 4.22.

	kenario Mengelola Data Petugas Parkir
Kode SKPL	SKPL_16
Aktor	Koordinator Parkir
Tujuan	Mengelola data petugas parkir dalam sistem
Deskripsi	Use case ini mendeskripsikan alur dari proses
AZK BRAS	mengelola data petugas parkir oleh koordinator
DESTRUCTION OF THE PARTY OF THE	parkir
Kondisi Awal	koordinator parkir telah login pada sistem
Main Flow	<ol> <li>Koordinator parkir memilih menu kelola data petugas parkir</li> <li>Sistem menampilkan list data petugas parkir</li> </ol>
	beserta pilihan menu
	3. Koordinator parkir memilih salah satu menu
ξ,	4. Sistem akan menampilkan menu yang dipilih
	oleh petugas parkir
8	5. Sistem akan menampilkan notifikasi atas
	perubahan data yang dilakukan oleh
Ali :: 51	koordinator parkir
Alternative Flow	2.1 Bila data petugas parkir tidak muncul
	pada sistem maka akan muncul notifikasi
	4.1 Bila ingin menambah petugas parkir pilih
	menu tambah petugas parkir, kemudian isi
	form yang tersedia dan tekan submit
B	4.2 Bila ingin mengedit petugas parkir pilih
	tombol edit petugas parkir, kemudian isi form
UNH	yang tersedia dan tekan submit
YAST OF	4.3 Bila ingin menghapus petugas parkir pilih
MARKYASI	tombol hapus petugas parkir kemudian data
AWWINAK	telah terhapus
BKERAWI	5.1 Bila perubahan data tidak berhasil pada
AZAS BRA	sistem maka akan muncul notifikasi
ACHPLACD	TODAY THURSAVEY I

Kondisi Akhir	Koordinator parkir dapat mengelola data petugas
	parkir

# 20. Use Case Memvalidasi Laporan Kehilangan

Use case ini menjelaskan alur untuk memproses validasi laporan kehilangan secara detail. Skenario memvalidasi laporan kehilangan dapat dilihat pada Tabel

Tabel 4.23 Sk	enario Memvalidasi Laporan Kehilangan
Kode SKPL	SKPL_17
Aktor	Koordinator Parkir
Tujuan	Memvalidasi laporan kehilangan dalam sistem
Deskripsi	Use case ini mendeskripsikan alur dari proses
	memvalidasi laporan kehilangan oleh koordinator
	parkir
Kondisi Awal	koordinator parkir telah login pada sistem
Main Flow	Koordinator parkir memilih menu list laporan
, £	kehilangan
	2. Sistem menampilkan list data laporan
	kehilangan
V	3. Koordinator parkir memilih laporan yang akan
	divalidasi, kemudian tekan detail
	4. Sistem menampilkan isi laporan kehilangan
	secara detail, kemudian tekan submit
	5. Sistem akan menampilkan notifikasi atas
	perubahan data yang dilakukan oleh
	koordinator parkir
Alternative Flow	2.1 Bila data laporan kehilangan tidak muncul
457	pada sistem maka akan muncul notifikasi
	5.1 Bila perubahan data tidak berhasil pada
HAYAYAU	sistem maka akan muncul notifikasi
Kondisi Akhir	Koordinator parkir dapat memproses validasi
BRAW WILL	laporan kehilangan

# 21. Use Case Mengelola Area Parkir

Use case ini menjelaskan alur untuk mengelola data area parkir secara detail. Skenario mengelola area parkir dapat dilihat pada Tabel 4.24.

Tabel 4.24 Skenario Mengelola Data Area Parkir

Kode SKPL	SKPL_18
Aktor	Koordinator Parkir
Tujuan	Mengelola data area parkir dalam sistem
Deskripsi	Use case ini mendeskripsikan alur dari proses
STATE OF THE STATE	mengelola data area parkir oleh koordinator
HO	parkir
Kondisi Awal	koordinator parkir telah login pada sistem
Main Flow	1. Koordinator parkir memilih menu area parkir
	2. Sistem menampilkan list data area parkir
	beserta pilihan menu
5	3. Koordinator parkir memilih salah satu menu
1	4. Sistem akan menampilkan menu yang dipilih
	oleh petugas parkir
R	5. Sistem akan menampilkan notifikasi atas
	perubahan data yang dilakukan oleh
	koordinator parkir
Alternative Flow	2.1 Bila data petugas parkir tidak muncul
	pada sistem maka akan muncul notifikasi
	4.1 Bila ingin menambah area parkir pilih
	menu tambah area parkir, kemudian isi form
	yang tersedia dan tekan submit
	4.2 Bila ingin mengedit area parkir pilih
	tombol edit area parkir, kemudian isi form
	yang tersedia dan tekan submit
YAVADATA	4.3 Bila ingin menghapus area parkir pilih
Whitay A.J.	tombol hapus area parkir kemudian data
AWKWURK	telah terhapus
BraRAWI	UNIALA VA PLANIKA UED
(A) KEBRA	5.1 Bila perubahan data tidak berhasil pada

NAMERS	sistem maka akan muncul notifikasi
Kondisi Akhir	Koordinator parkir dapat mengelola data area
	parkir

# 22. Use Case Memvalidasi Laporan Ditemukan

Use case ini menjelaskan alur untuk memproses validasi laporan ditemukan secara detail. Skenario memvalidasi laporan ditemukan dapat dilihat pada Tabel 4.25.

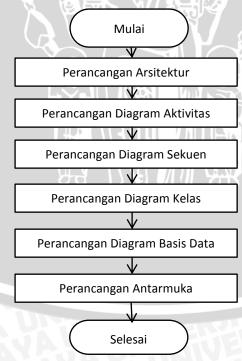
Tabel 4.25 Skenario Memvalidasi Laporan Ditemukan  Kode SKPL SKPL 19	
Kode SKPL	
Aktor	Koordinator Parkir
Tujuan	Memvalidasi laporan ditemukan dalam sistem
Deskripsi	Use case ini mendeskripsikan alur dari proses
	memvalidasi laporan ditemukan oleh koordinator
	parkir
Kondisi Awal	koordinator parkir telah login pada sistem
Main Flow	1. Koordinator parkir memilih menu list laporan
	ditemukan
	2. Sistem menampilkan list data laporan
	ditemukan
	3. Koordinator parkir memilih laporan yang akan
	divalidasi, kemudian tekan detail
	4. Sistem menampilkan isi laporan ditemukan
	secara detail, kemudian tekan submit
	5. Sistem akan menampilkan notifikasi atas
	perubahan data yang dilakukan oleh
	koordinator parkir
Alternative Flow	2.1 Bila data laporan ditemukan tidak muncul
	pada sistem maka akan muncul notifikasi
	5.1 Bila perubahan data tidak berhasil pada
	sistem maka akan muncul notifikasi
Kondisi Akhir	Koordinator parkir dapat memproses validasi
	laporan ditemukan

#### **BAB 5 PEMBAHASAN**

Pembahasan berfungsi untuk menerjemahkan makna dari hasil yang diperoleh untuk menjawab pertanyaan atau masalah penelitian. Fungsi lainnya adalah untuk menjelaskan pemahaman baru yang didapatkan dari hasil penelitian, yang diharapkan berguna dalam pengembangan keilmuan. Dalam penelitian tingkat lanjut, fungsi pembahasan yang kedua ini sangat penting karena dapat menunjukkan kontribusi penulis terhadap pengembangan keilmuan. Akan tetapi, dalam penelitian tingkat skripsi, fungsi yang kedua ini dapat diterapkan secara terbatas karena pendidikan S1 tidak dituntut untuk pengembangan keilmuan secara substansial, tetapi cukup terhadap pemahaman personal dalam implementasi konsep atau teori.

# **5.1 Perancangan Sistem**

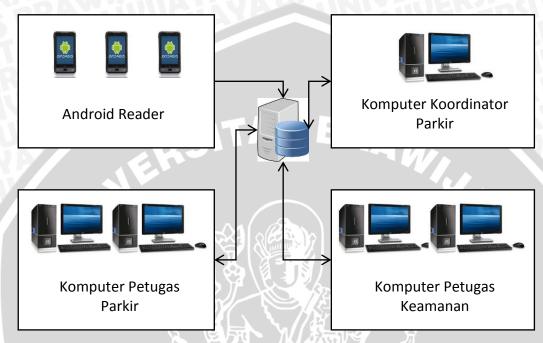
Tahap perancangan sistem pada penelitian ini dilakukan dengan enam tahap pengerjaan yaitu merancang arsitektur dari sistem yang dibangun, memodelkan pada diagram aktivitas, memodelkan pada diagram sekuen, memodelkan pada diagram kelas, kemudian melakukan pemodelan diagram ER (Entity Relationship), dan pada tahap akhir perancangan sistem akan dilakukan pemodelan antarmuka sistem. Perancangan sistem yang dilakukan menggunakan pendekatan beriorientasi objek dengan diagram UML (Unified Modeling Language). Gambar 5.1 merepresentasikan tahapan dari perancangan sistem.



Gambar 5.1 Diagram Alir Perancangan Sistem

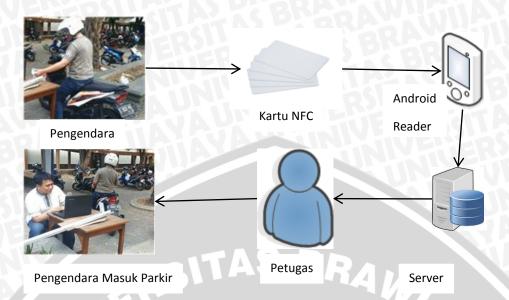
### 5.1.1 Perancangan Arsitektural

Sistem pengelolaan parkir menggunakan perangkat NFC berbasis android ini menggunakan arsitektur *client-server* dengan web service untuk komunikasi dengan aplikasi *native* android pada ponsel android yang menjadi reader. Desain arsitektural sistem yang dikembangkan dapat dilihat pada Gambar 5.2.



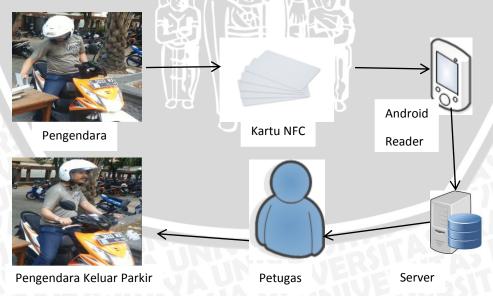
**Gambar 5.2 Arsitektur Sistem** 

Pada alur masuk area parkir, pengendara membawa kartu nfc dan mendekatkan pada android reader. Setelah uid muncul maka pengendara memilih menu masuk area parkir. Petugas parkir kemudian mengecek inputan dari pengendara dengan memilih menu input kendaraan masuk parkir. Petugas parkir kemudian memilih data kendaraan dari inputan pengendara kemudian tekan proses. Setelah muncul notifikasi kendaraan berhasil masuk area parkir maka petugas parkir mempersilahkan pengendara untuk masuk area parkir. Gambar 5.3 menjelaskan alur masuk area parkir.



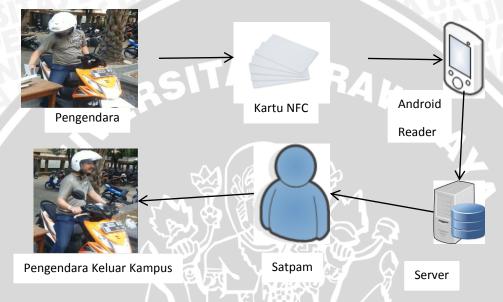
#### Gambar 5.3 Alur Masuk Area Parkir

Pada alur keluar area parkir, pengendara membawa kartu nfc dan mendekatkan pada android reader. Setelah uid muncul maka pengendara memilih menu keluar area parkir. Petugas parkir kemudian mengecek inputan dari pengendara dengan memilih menu input kendaraan keluar parkir. Petugas parkir kemudian memilih data kendaraan dari inputan pengendara kemudian tekan proses. Muncul halaman detail parkir kendaraan lalu petugas membandingkan data pada sistem dengan kendaraan yang digunakan oleh pengendara. Bila cocok maka petugas parkir memilih tombol submit. Setelah muncul notifikasi kendaraan berhasil keluar area parkir maka petugas parkir mempersilahkan pengendara untuk keluar area parkir. Gambar 5.4 menjelaskan alur keluar area parkir.



#### **Gambar 5.4 Alur Keluar Area Parkir**

Pada alur keluar area kampus, pengendara membawa kartu nfc dan mendekatkan pada android reader. Setelah uid muncul maka pengendara memilih menu keluar area kampus. Petugas keamanan kemudian mengecek inputan dari pengendara dengan memilih menu input kendaraan keluar kampus. Petugas keamanan kemudian memilih data kendaraan dari inputan pengendara kemudian tekan proses. Muncul halaman detail parkir kendaraan lalu petugas membandingkan data pada sistem dengan kendaraan yang digunakan oleh pengendara. Bila cocok maka petugas keamanan memilih tombol submit. Setelah muncul notifikasi kendaraan berhasil keluar area kampus maka petugas keamanan mempersilahkan pengendara untuk keluar area kampus. Gambar 5.5 menjelaskan alur keluar area kampus.



**Gambar 5.5 Alur Keluar Area Kampus** 

Pada alur pendaftaran, pengendara membawa fotokopi kartu identitas berupa ktm bagi mahasiswa atau id card bagi pegawai/dosen dan membawa fotokopi stnk kendaraan yang akan didaftarkan. Setelah itu koordinator parkir meregistrasikan pengendara dan kendaraan pada sistem. Koordinator parkir lalu memberikan kartu nfc sebagai kartu identitas untuk keluar-masuk area parkir dan area kampus setelah data pengendara dan kendaraan valid sesuai dengan kartu identitas dan stnk yang didaftarkan. Gambar 5.6 menjelaskan alur pendaftaran pengendara dan kendaraan.



Gambar 5.6 Alur Pendaftaran Pengendara dan Kendaraan

### 5.1.2 Perancangan Diagram Aktivitas

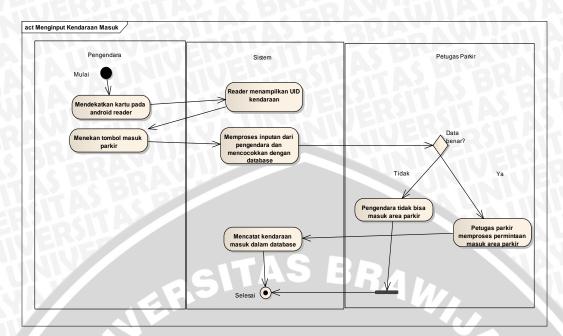
Diagram aktivitas menggambarkan aktivitas yang dapat dilakukan oleh sistem. Diagram ini mengacu kepada skenario *use case* yang telah dibuat sebelumnya.

1. Diagram Aktivitas Area Petugas Parkir

Diagram aktivitas area petugas parkir menggambarkan aktivitas yang dilakukan oleh sistem beserta interaksinya dengan aktor. Aktor yang mengoperasikan bagian ini adalah petugas parkir.

a. Diagram Aktivitas Menginput Kendaraan Masuk Area Parkir

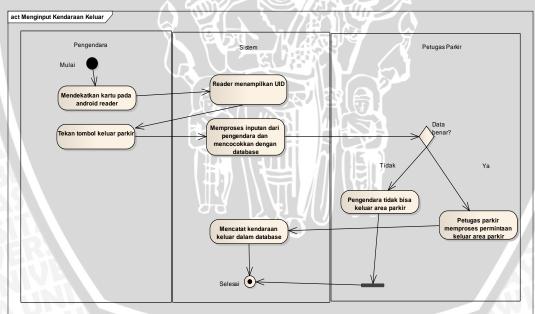
Diagram aktivitas menginput kendaraan masuk merupakan aktivitas ketika petugas parkir memberikan akses untuk masuk area parkir kepada pengendara. Gambar 5.7 menjelaskan aktivitas sistem untuk menginput kendaraan masuk.



Gambar 5.7 Diagram Aktivitas Menginput Kendaraan Masuk Area Parkir

b. Diagram Aktivitas Menginput Kendaraan Keluar Area Parkir

Diagram aktivitas menginput kendaraan keluar merupakan aktivitas ketika petugas parkir memberikan akses kepada pengendara untuk keluar area parkir. Gambar 5.8 menjelaskan aktivitas sistem untuk menginput kendaraan keluar.



Gambar 5.8 Diagram Aktivitas Menginput Kendaraan Keluar Area Parkir

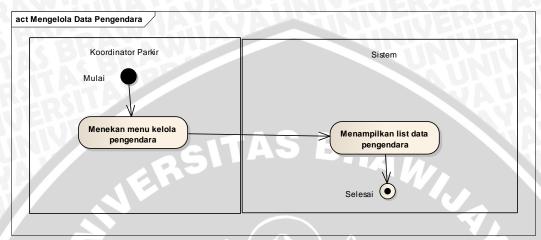
2. Diagram Aktivitas Area Koordinator Parkir

Diagram aktivitas area koordinator parkir menggambarkan aktivitas yang dilakukan oleh sistem beserta interaksinya dengan aktor. Aktor yang mengoperasikan area ini adalah koordinator parkir.

54

### a. Diagram Aktivitas Mengelola Data Pengendara

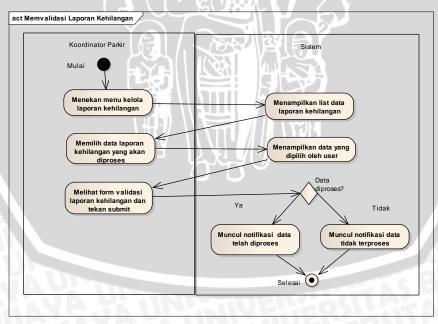
Diagram aktivitas mengelola data pengendara merupakan aktivitas ketika petugas parkir melakukan lihat, tambah, koreksi, dan hapus data pengendara secara detail. Gambar 5.9 menjelaskan aktivitas sistem untuk melihat data pengendara.



**Gambar 5.9 Diagram Aktivitas Melihat Data Pengendara** 

### b. Diagram Aktivitas Memvalidasi Laporan Kehilangan

Diagram aktivitas memvalidasi laporan kehilangan adalah aktivitas ketika koordinator parkir memvalidasi laporan kehilangan yang diajukan oleh petugas parkir. Gambar 5.10 menjelaskan aktivitas sistem untuk memvalidasi laporan kehilangan.



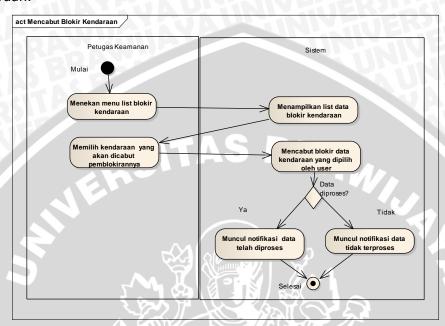
Gambar 5.10 Diagram Aktivitas Memvalidasi Laporan Kehilangan

#### 3. Diagram Aktivitas Area Keamanan

Diagram aktivitas area keamanan menggambarkan aktivitas yang dilakukan oleh sistem beserta interaksinya dengan aktor. Aktor yang mengoperasikan area keamanan adalah petugas keamanan.

### a. Diagram Aktivitas Mencabut blokir Kendaraan

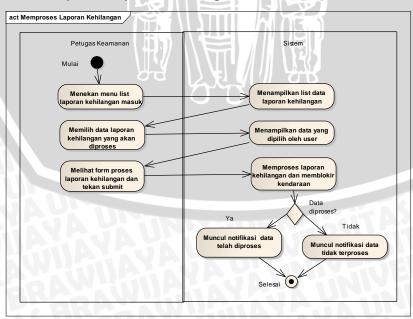
Diagram aktivitas mencabut blokir kendaraan merupakan aktivitas ketika satpam ingin mencabut blokir terhadap kendaraan yang sebelumnya telah diblokir. Gambar 5.11 menjelaskan aktivitas sistem untuk mencabut blokir kendaraan.



Gambar 5.11 Diagram Aktivitas Mencabut Blokir Kendaraan

b. Diagram Aktivitas Memproses Laporan Kehilangan

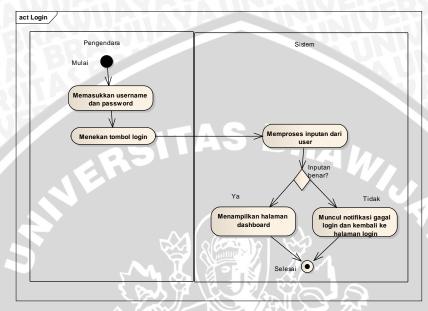
Diagram aktivitas memproses laporan kehilangan merupakan aktivitas ketika satpam memproses laporan kehilangan yang dilaporkan oleh petugas parkir dengan validasi dari koordinator parkir. Gambar 5.12 menjelaskan aktivitas sistem untuk memproses laporan kehilangan.



Gambar 5.12 Diagram Aktivitas Memproses Laporan Kehilangan

### 4. Diagram Aktivitas Login

Diagram aktivitas login merupakan aktivitas ketika pengendara yang telah teregistrasi pada sistem ingin masuk ke dalam sistem. Gambar 5.13 menjelaskan aktivitas sistem untuk aktivitas login.



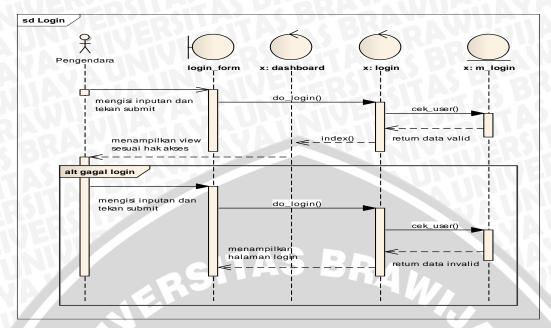
**Gambar 5.13 Diagram Aktivitas Login** 

### 5.1.3 Perancangan Diagram Sekuen

Diagram sekuen merepresentasikan model alir jalannya proses interaksi antar objek. Pemodelan diagram sekuen ini mengacu kepada beberapa diagram yang telah dibuat sebelumnya seperti operasi yang ada pada kelas diagram, skenario use case, dan diagram aktivitas.

### 1. Diagram Sekuen Login

Gambar 5.14 menjelaskan urutan interaksi objek sistem untuk login kedalam sistem.



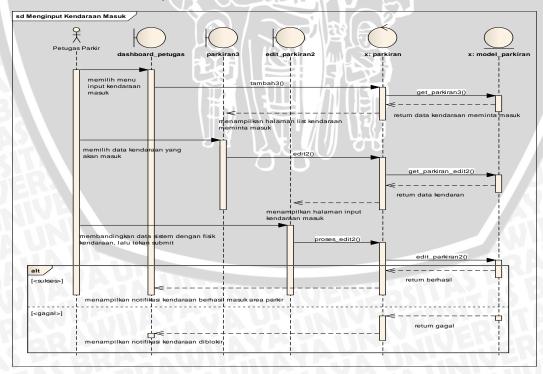
Gambar 5.14 Diagram Sekuen Login

2. Diagram Sekuen Area Petugas Parkir

Diagram sekuen area petugas parkir menggambarkan urutan interaksi antar objek pada area petugas parkir. Aktor yang berinteraksi di sini adalah petugas parkir.

a. Diagram Sekuen Menginput Kendaraan Masuk Area Parkir

Gambar 5.15 menjelaskan urutan interaksi objek sistem untuk menginput kendaraan masuk area parkir.



Gambar 5.15 Diagram Sekuen Menginput Kendaraan Masuk

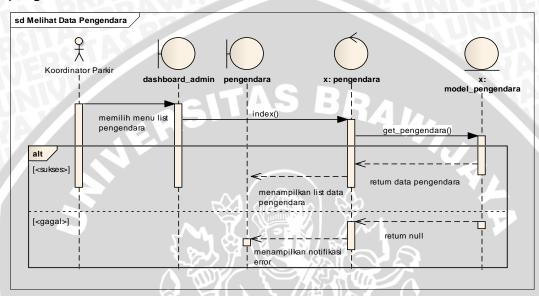
58

3. Diagram Sekuen Area Koordinator Parkir

Diagram sekuen area koordinator parkir menggambarkan urutan interaksi antar objek pada area koordinator parkir. Aktor yang berinteraksi di sini adalah koordinator parkir.

a. Diagram Sekuen Mengelola Data Pengendara

Gambar 5.16 menjelaskan urutan interaksi objek sistem untuk melihat data pengendara.



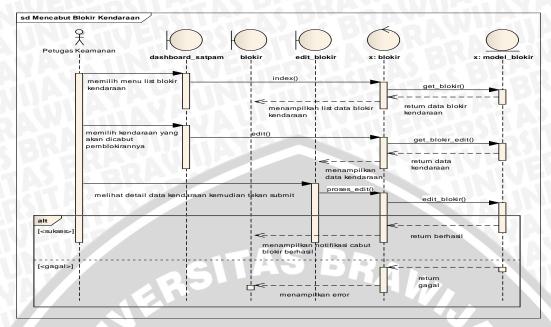
**Gambar 5.16 Diagram Sekuen Melihat Data Pengendara** 

4. Diagram Sekuen Area Keamanan

Diagram sekuen area keamanan menggambarkan urutan interaksi antar objek pada area keamanan. Aktor yang berinteraksi di sini adalah satuan pengamanan.

a. Diagram Sekuen Mencabut Blokir Kendaraan

Gambar 5.17 menjelaskan urutan interaksi objek sistem untuk mencabut blokir kendaraan.

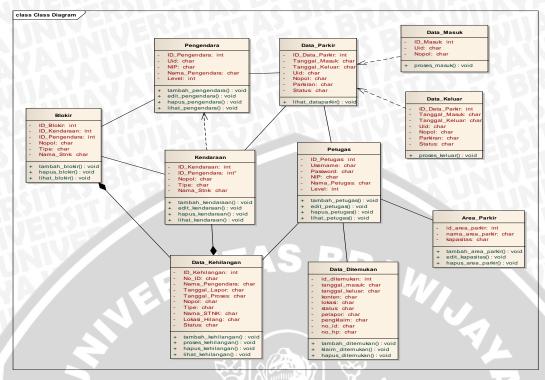


Gambar 5.17 Diagram Sekuen Mencabut Blokir Kendaraan

### 5.1.4 Perancangan Diagram Kelas

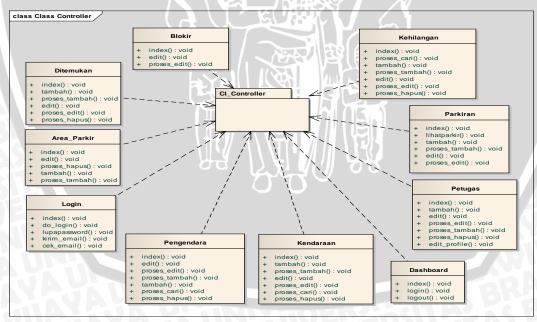
Perancangan diagram kelas pada penelitian ini dibagi menjadi dua tahap yaitu perancangan diagram kelas level analisis dan diagram kelas level implementasi. Diagram kelas level analisis digunakan untuk merepresentasikan atribut data dan interaksi yang digunakan dalam basis data sedangkan diagram kelas level implementasi digunakan untuk merepresentasikan struktur dari kelas yang diimplementasikan dalam sistem.

Diagram kelas level implementasi sendiri dibagi menjadi tiga jenis yaitu kelas proses (controller), kelas data (model) dan kelas halaman (view). Hal tersebut dilakukan sebab sistem ini menggunakan pola arsitektur MVC (Model, View Controller). Gambar 5.18 merepresentasikan diagram kelas level analisis.

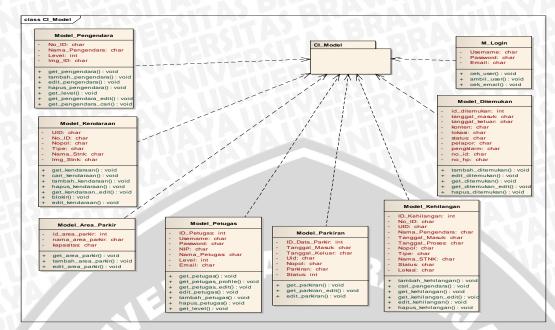


**Gambar 5.18 Diagram Kelas Sistem Level Analisis** 

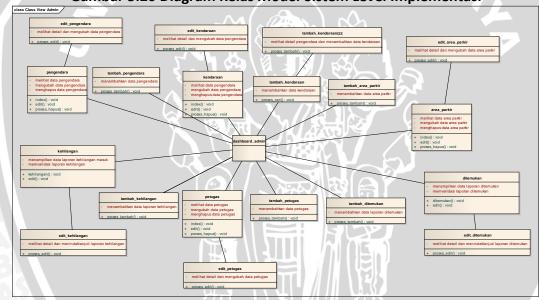
Sedangkan Gambar 5.19 dan Gambar 5.20 merepresentasikan diagram kelas level implementasi. Gambar 5.19 fokus pada bagian *Controller* dari sistem. Sedangkan Gambar 5.20 fokus pada bagian *Model* dari sistem. Gambar 5.21 hingga gambar 5.23 fokus pada bagian *View* dari sistem.



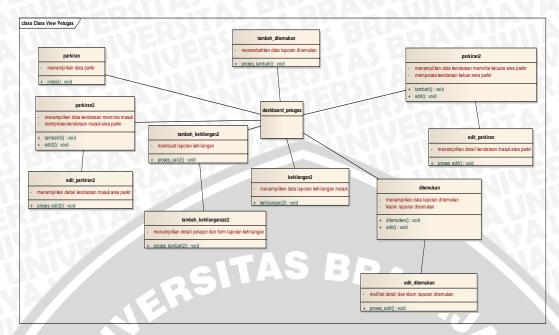
Gambar 5.19 Diagram Kelas Controller Sistem Level Implementasi



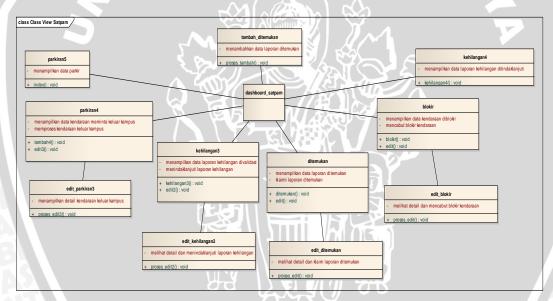
Gambar 5.20 Diagram Kelas Model Sistem Level Implementasi



Gambar 5.21 Diagram Kelas View Sistem Level Implementasi Aktor Admin



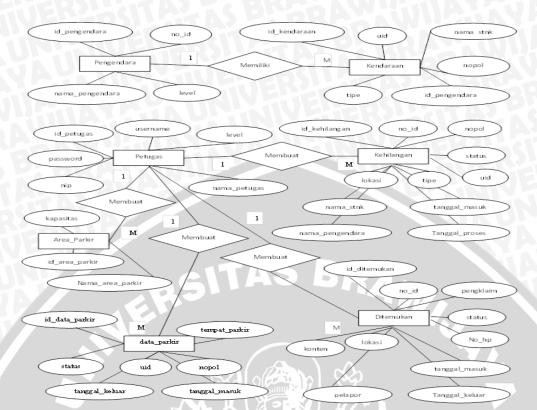
Gambar 5.22 Diagram Kelas View Sistem Level Implementasi Aktor Petugas



Gambar 5.23 Diagram Kelas *View* Sistem Level Implementasi Aktor Petugas Kemananan

### 5.1.5 Perancangan Basis Data

Berdasarkan diagram kelas level analisis yang telah dimodelkan sebelumnya maka dapat dimodelkan bentuk *Entity Relationship Diagram* (ERD) dari sistem ini. Gambar 5.24 memperlihatkan model ERD tersebut.



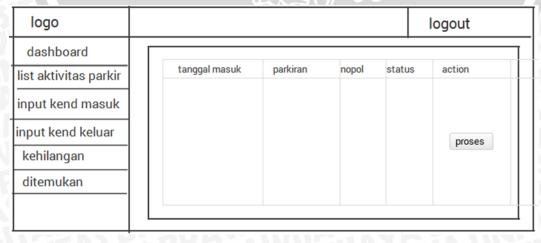
Gambar 5.24 Entity Relationship Diagram (ERD) Sistem

### 5.1.6 Perancangan Antarmuka

Pada bagian ini akan dijelaskan tentang perancangan antarmuka dari sistem pengelolaan parkir. Antarmuka yang dimodelkan yaitu antarmuka untuk halaman login, halaman utama dari petugas parkir, koordinator parkir, dan petugas keamanan.

#### 1. Perancangan Antarmuka Input Kendaraan Keluar

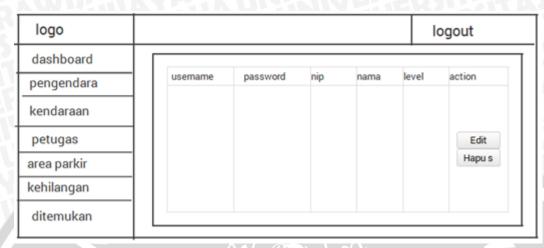
Gambar 5.25 merepresentasikan rancangan antarmuka untuk aktor petugas parkir dalam melakukan proses input kendaraan keluar dari area parkir. Terdapat list kendaraan meminta keluar dari area parkir kemudian petugas parkir memproses agar pengendara dapat keluar area parkir.



Gambar 5.25 Perancangan Antarmuka Input Kendaraan Keluar

#### 2. Perancangan Antarmuka List Petugas

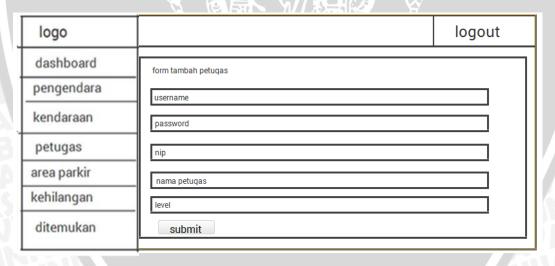
Gambar 5.26 merepresentasikan rancangan antarmuka dari aktor koordinator parkir dalam menu list petugas. Terdapat list data petugas beserta aksi edit data petugas dan hapus data petugas.



**Gambar 5.26 Perancangan Antarmuka List Petugas** 

3. Perancangan Antarmuka Form Tambah Petugas

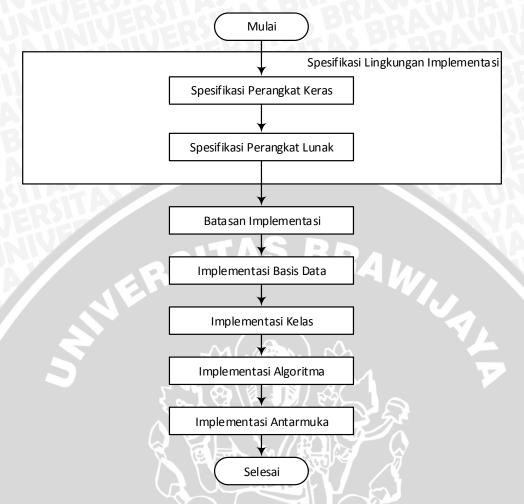
Gambar 5.27 merepresentasikan rancangan antarmuka untuk aktor koordinator parkir dalam menu tambah petugas. Terdapat form untuk menambahka petugas.



Gambar 5.27 Perancangan Antarmuka Form Tambah Petugas

#### 5.2 Implementasi Sistem

Bagian ini memaparkan tahapan implementasi sistem pengelolaan parkir menggunakan perangkat NFC berbasis android yang mengacu kepada tahap perancangan sistem. Bahasan pada sub bab ini terdiri dari penjelasan terkait spesifikasi lingkungan implementasi, batasan dalam implementasi, implementasi basis data, implementasi kelas, implementasi algoritma dan implementasi antarmuka. Gambar 5.28 menjelaskan alur dari implementasi sistem ini.



Gambar 5.28 Diagram Alir Implementasi

# 5.2.1 Spesifikasi Lingkungan Implementasi

Sistem pengelolaan parkir menggunakan perangkat NFC berbasis android ini dikembangkan dalam lingkungan implementasi yang terdiri dari perangkat keras dan perangkat lunak dengan bahasa pemrograman tertentu.

### 5.2.1.1 Spesifikasi Perangkat Keras

Spesifikasi perangkat keras yang digunakan dalam proses implementasi sistem dapat dilihat pada Tabel 5.1.

**Tabel 5.1 Spesifikasi Perangkat Keras** 

Laptop Lenovo G40-45			
Processor AMD Quad Core A6-6310			
Memory (RAM)	4 GB		
Harddisk	500 GB		
VGA	AMD Radeon R4		
Resolusi Layar	1366x768 Pixel		

### 5.2.1.2 Spesifikasi Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan dalam proses implementasi sistem dapat dilihat pada Tabel 5.2.

Tabel 5.2 Spesifikasi Perangkat Lunak

	Laptop Lenovo G40-45			
Sistem Operasi	Microsoft Windows 8.1 Pro with Media Center x64			
Bahasa Pemrograman	PHP, HTML, CSS, Javascript, java			
Perkakas Bantu XAMPP 1.8.3				
ATT -	Sublime Text 3			
0	Enterprise Architect 8			
Y IE	Microsoft Visio 2010			
	Microsoft Word 2010			
3	Android Studio			
2	Library lib-nfc			

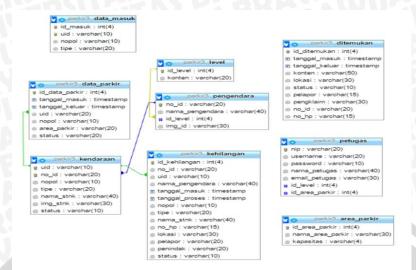
### 5.2.2 Batasan Implementasi

Beberapa batasan dalam implementasi sistem diantaranya:

- 1. Sistem yang dibangun mencakup tiga bagian yang ada dalam pelayanan parkir yaitu petugas parkir, koordinator parkir, dan petugas keamanan.
- 2. Database Management System yang digunakan adalah MySQL versi 5.6.20.
- 3. Framework Codelgniter yang digunakan adalah versi 2.1.0.
- 4. Simulasi pengoperasian sistem menggunakan koneksi Ad-Hoc.
- 5. Kartu NFC yang digunakan adalah Mifare Classic 1k 13.56MHz.

#### 5.2.3 Implementasi Basis Data

Implementasi basis data dilakukan berdasarkan perancangan sistem dengan menggunakan perkakas bantu PHPMyAdmin 4.2.7.1. Skema basis data yang telah diimplementasikan dapat dilihat pada Gambar 5.29.



**Gambar 5.29 Skema Basis Data Sistem** 

### 5.2.4 Implementasi Kelas

Setiap kelas yang telah dirancang pada tahap perancangan sistem direalisasikan pada sebuah *file* sistem dengan ekstensi \*.php. Penjelasan terkait kelas dan *file* sistem yang digunakan pada sistem akan dijelaskan pada Tabel 5.3.

Tabel 5.3 Implementasi Kelas

No	Package	Nama Kelas	Nama File Sistem
1	controllers	Area Parkir	area_parkir.php
2	controllers	Dashboard	dashboard.php
3	controllers	Login	login.php
4	controllers	Pengendara	pengendara.php
5	controllers	Kendaraan	kendaraan.php
6	controllers	Petugas	petugas.php
7	controllers	parkiran	parkiran.php
8	controllers	Kehilangan	kehilangan.php
9	controllers	Ditemukan	ditemukan.php
10	models	M_Login	m_login.php
11	models	Model_Blokir	model_blokir.php
12	models	Model_Pengendara	model_pengendara.php
13	models	Model_Kendaraan	model_kendaraan.php
14	models	Model_Petugas	model_petugas.php
15	models	Model_Parkiran	model_dataparkir.php
16	models	Model_Kehilangan	model_kehilangan.php
17	models	Model_Area_Parkir	model_area_parkir.php
18	models	Model_Ditemukan	model_ditemukan.php
19	views	File HTML	dashboard_admin.php
20	views	File HTML	dashboard_petugas.php
21	views	File HTML	dashboard_satpam.php

		AND BY DIK	
19	views	File HTML	login_form.php
19	views	File HTML	edit_admin_profile.php
19	views	File HTML	edit_petugas_profile.php
19	views	File HTML	edit_satpam_profile.php
19	views	File HTML	lupapassword.php
19	views	File HTML	kirim_email.php
46	views	File HTML	tambah_pengendara.php
47	views	File HTML	edit_pengendara.php
48	views	File HTML	pengendara.php
22	views/area_parkir	File HTML	area_parkir.php
22	views/area_parkir	File HTML	tambah_area_parkir.php
22	views/area_parkir	File HTML	edit_area_parkir.php
22	views/ditemukan	File HTML	ditemukan.php
22	views/ditemukan	File HTML	tambah_ditemukan.php
22	views/ditemukan	File HTML	edit_ditemukan.php
22	<i>views/</i> ditemukan	File HTML	ditemukan2.php
22	views/ditemukan	File HTML	tambah_ditemukan2.php
22	<i>views/</i> ditemukan	File HTML	edit_ditemukan2.php
22	views/petugas	File HTML	petugas.php
23	views/petugas	File HTML	tambah_petugas.php
24	views/petugas File HTML edit_petugas.php		edit_petugas.php
26	<i>views/</i> kehilangan <i>File</i> HTML tambah_kehilangan		tambah_kehilangan.php
27	views/kehilangan	File HTML	tambah_kehilangan2.php
28	views/kehilangan	File HTML	edit_kehilangan.php
29	views/kehilangan	File HTML	edit_kehilangan2.php
30	views/kehilangan	File HTML	kehilangan.php
31	views/kehilangan	File HTML	kehilangan2.php
32	views/kehilangan	File HTML	kehilangan3.php
33	views/kendaraan	File HTML	tambah_kendaraan.php
34	views/kendaraan	File HTML	edit_kendaraan.php
35	views/kendaraan	File HTML	kendaraan.php
36	views/parkiran	File HTML	tambah_parkiran.php
37	views/parkiran	File HTML	tambah_parkiran2.php
38	views/parkiran	File HTML	edit_parkiran.php
39	views/parkiran	File HTML	edit_parkiran2.php
40	views/parkiran	File HTML	edit_parkiran3.php
41	views/parkiran File HTML parkiran.php		parkiran.php
42	views/parkiran	File HTML	parkiran2.php
43	views/parkiran	File HTML	parkiran3.php
44	views/parkiran	File HTML	parkiran4.php
45	views/parkiran	File HTML	parkiran5.php
46	views/pengendara	File HTML	tambah_pengendara.php

47	views/pengendara	File HTML	edit_pengendara.php
48	views/pengendara	File HTML	pengendara.php

#### 5.2.5 Implementasi Algoritma

Berikut ini implementasi algoritma yang diterapkan pada sistem.

#### 5.2.5.1 Algoritma Menambahkan Data Petugas

Algoritma menambahkan data petugas ini merupakan algoritma untuk operasi create\_petugas dari kelas petugas pada *package models*. Algoritma ini berfungsi untuk melakukan penambahan data petugas. Gambar 5.30 menjelaskan algoritma menambahkan petugas.

1	procedure menambahkan petugas;
2	CONTROLLER petugas function tambah();
3	LOAD view(petugas/tambah_petugas);
4	SET username, password, nip, nama_petugas, level;
5	THEN submit;
6	CONTROLLER petugas function proses_tambah();
7	LOAD model(model_petugas);
8	MODEL model_petugas function tambah_petugas();
9	SET username, password, nip, nama_petugas, level;
10	DB INSERT petugas;
11	REDIRECT petugas;
12	LOAD view(petugas);
13	END menambahkan petugas;

#### **Gambar 5.30 Algoritma Menambahkan Data Petugas**

Penjelasan algoritma diatas adalah sebagai berikut :

Baris 2: mengakses fungsi tambah pada kontroller petugas

Baris 3: menampilkan view tambah petugas

Baris 4-5 : mengisi username, password, nip, nama\_petugas, level, kemudian submit

Baris 6: mengakses fungsi proses\_tambah pada kontroller petugas

Baris 7-8 : memanggil model model\_petugas dan mengakses fungsi proses tambah

Baris 9-10 : menginput username, password, nip, nama\_petugas, level pada database

Baris 11-12 : mengakses kembali kontroller petugas dan menampilkan view petugas

#### 5.2.5.2 Algoritma Mengubah Data Petugas

Algoritma menambahkan data petugas ini merupakan algoritma untuk operasi edit\_petugas dari kelas petugas pada *package models*. Algoritma ini berfungsi untuk melakukan pengubahan data petugas. Gambar 5.41 menjelaskan algoritma mengubah data petugas.

<sup>1</sup> procedure mengubah petugas;

```
CONTROLLER petugas function index();
3
           LOAD view (petugas);
           SELECT row(id petugas);
           CONTROLLER petugas function edit();
6
           LOAD model (model petugas);
           MODEL model petugas function get petugas edit(id petugas);
8
           LOAD view(petugas/edit_petugas);
9
               SET username, password, nip, nama petugas, level;
10
               THEN submit;
           CONTROLLER petugas function proses edit();
11
12
           LOAD model (model petugas);
13
               MODEL model petugas function edit petugas();
14
                 SET username, password, nip, nama petugas, level;
15
                 WHERE id_petugas == petugas.id_petugas
16
                 DB UPDATE petugas;
17
           REDIRECT petugas;
           LOAD view (petugas);
18
19
    END mengubah petugas;
```

**Gambar 5.31 Algoritma Mengubah Data Petugas** 

Penjelasan algoritma diatas adalah sebagai berikut :

Baris 2 : mengakses fungsi index pada kontroller petugas

Baris 3: menampilkan view petugas

Baris 4: memilih data yang akan diedit

Baris 5: mengakses fungsi edit pada kontroller petugas

Baris 6-7 : memanggil model model\_petugas dan mengakses fungsi get\_petugas\_edit

Baris 8: menampilkan view edit petugas

Baris 9-10 : menginput username, password, nip, nama\_petugas, level kemudian submit

Baris 11: mengakses fungsi proses\_edit pada kontroller petugas

Baris 12-13 : memanggil model model\_petugas dan mengakses fungsi edit petugas

Baris 14-16 : update username, password, nip, nama\_petugas, level pada database

Baris 17-18 : mengakses kembali kontroller petugas dan menampilkan view petugas

### 5.2.5.3 Algoritma Menghapus Data Petugas

Algoritma menghapus data petugas ini merupakan algoritma untuk operasi delete\_petugas dari kelas petugas pada *package models*. Algoritma ini berfungsi untuk melakukan penghapusan data petugas. Gambar 5.42 menjelaskan algoritma menghapus data petugas.

```
procedure menghapus petugas;
CONTROLLER petugas function index();
LOAD view(petugas);
```

```
SELECT row(id petugas);
5
           CONTROLLER petugas function proses hapus();
           LOAD model (model petugas);
6
           MODEL model_petugas function hapus_petugas(id_petugas);
8
                 SET username, password, nip, nama petugas, level;
                 WHERE id petugas == petugas.id petugas
                 DB DELETE petugas;
10
11
           REDIRECT petugas;
12
           LOAD view (petugas);
13
    END menghapus petugas;
```

**Gambar 5.32 Algoritma Menghapus Data Petugas** 

Penjelasan algoritma diatas adalah sebagai berikut :

Baris 2: mengakses fungsi index pada kontroller petugas

Baris 3 : menampilkan view petugas

Baris 4: memilih data yang akan dihapus

Baris 5: mengakses fungsi proses\_hapus pada kontroller petugas

Baris 6-7 : memanggil model model\_petugas dan mengakses fungsi hapus\_petugas

Baris 8-10: delete username, password, nip, nama\_petugas, level pada database

Baris 11-12 : mengakses kembali kontroller petugas dan menampilkan view petugas

#### 5.2.5.4 Algoritma Melihat Data Petugas

Algoritma melihat data petugas ini merupakan algoritma untuk operasi get\_petugas dari kelas petugas pada *package models*. Algoritma ini berfungsi untuk melihat data petugas. Gambar 5.43 menjelaskan algoritma melihat data petugas.

```
procedure melihat petugas;
2
        CONTROLLER petugas function index();
3
        LOAD view(petugas);
4
        LOAD model (model_petugas);
        MODEL model petugas function get petugas();
5
6
                 DB SELECT petugas;
7
           REDIRECT petugas;
8
          LOAD view (petugas);
9
    END melihat petugas;
```

#### **Gambar 5.33 Algoritma Melihat Data Petugas**

Penjelasan algoritma diatas adalah sebagai berikut :

Baris 2: mengakses fungsi index pada kontroller petugas

Baris 3: menampilkan view petugas

Baris 4-5: memanggil model model\_petugas dan mengakses fungsi get\_petugas

Baris 6 : select data petugas pada database

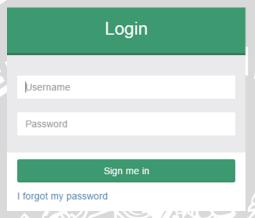
Baris 7-8: mengakses kembali kontroller petugas dan menampilkan view petugas

### 5.2.6 Implementasi Antarmuka

Antarmuka sistem digunakan oleh pengguna agar dapat berinteraksi dengan komponen-komponen sistem yang telah dikembangkan. Antarmuka utama dibagi menjadi tiga bagian yaitu antarmuka untuk petugas parkir, petugas keamanan, dan koordinator parkir.

#### 5.2.6.1 Implementasi Antarmuka Login

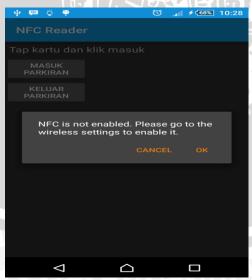
Gambar 5.45 menggambarkan hasil implementasi dari halaman login. Halaman ini digunakan untuk melakukan autentikasi terhadap pengguna yang ingin mengakses sistem.



Gambar 5.34 Implementasi Antarmuka Login

#### 5.2.6.2 Implementasi Antarmuka NFC Reader

Gambar 5.46 menggambarkan hasil implementasi dari antarmuka nfc reader pada ponsel android.

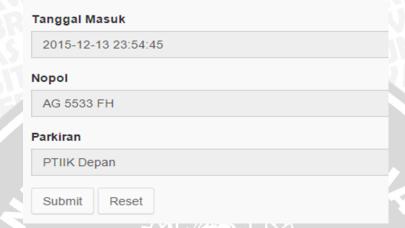


**Gambar 5.35 Implementasi Antarmuka NFC Reader** 

Antarmuka yang digambarkan tersebut merupakan halaman utama pada nfc reader, ketika pertama kali dijalankan maka aplikasi akan mengecek status nfc apakah sudah dinyalakan atau tidak. Bila sudah dinyalakan maka aplikasi dapat digunakan.

#### 5.2.6.3 Implementasi Antarmuka Proses Keluar Area Parkir

Gambar 5.47 menggambarkan hasil implementasi dari antarmuka detail proses keluar area parkir.



### Gambar 5.36 Implementasi Antarmuka Proses Keluar Area Parkir

Antarmuka yang digambarkan tersebut merupakan form untuk melihat detail parkir pengendara yang dilakukan oleh petugas parkir untuk memproses pengendara yang ingin keluar area parkir.

#### 5.2.6.4 Implementasi Antarmuka Scanning Smartcard NFC Reader

Gambar 5.48 menggambarkan hasil implementasi dari antarmuka scanning smartcard NFC reader.



#### Gambar 5.37 Implementasi Antarmuka Scanning Smartcard NFC Reader

Antarmuka yang digambarkan tersebut merupakan aplikasi ketika setelah melakukan scanning pada kartu nfc maka akan menampilkan uid untuk menjalankan aksi yang ada pada aplikasi. Setelah uid muncul maka pengendara dapat melakukan aksi untuk masuk atau keluar area parkir maupun keluar area kampus.

#### **BAB 6 PENGUJIAN DAN EVALUASI**

Pada bab ini akan dilakukan tahap pengujian dan analisis dari sistem pengelolaan parkir menggunakan perangkat nfc berbasis android. Pengujian dilakukan dengan tiga tahap yakni pengujian black-box, pengujian white-box, dan pengujian performa. Pengujian black-box adalah pengujian untuk mengetahui apakah sistem yang dibangun telah bekerja dengan baik dan sesuai kebutuhan, tanpa melihat struktur logika internal pada perangkat lunak. Pengujian white-box adalah pengujian untuk mengetahui logika internal perangkat lunak dengan menganalisa kode program untuk mengetahui terjadi kesalahan atau tidak pada perangkat lunak, dan pengujian performa dilakukan untuk mengukur performa sistem perangkat lunak yang telah dibuat untuk mendapatkan hasil sesuai dengan yang diharapkan. Seluruh pengujian disertakan analisis dari hasil pengujian tersebut dengan mengacu kepada dasar teori yang bertujuan untuk memastikan sistem telah memenuhi kebutuhan yang didefinisikan pada tahap analisis kebutuhan.

### 6.1 Pengujian White-Box

### 6.1.1 Tujuan

Tujuan dilakukannya pengujian white-box adalah untuk memastikan beberapa algoritma yang memiliki prioritas tinggi telah diimplementasikan sesuai dengan yang diharapkan.

#### 6.1.2 Mekanisme

Pengujian unit dilakukan dengan menggunakan teknik white box dengan jenis pengujian basis path. Algoritma yang diujikan pada pengujian ini merupakan algoritma masuk area parkir, keluar area parkir, dan keluar area kampus. Pengujian unit dilakukan pada ketiga algoritma tersebut sebab merupakan domain permasalahan dari sistem yang telah dibangun.

### 6.1.3 Hasil Pengujian

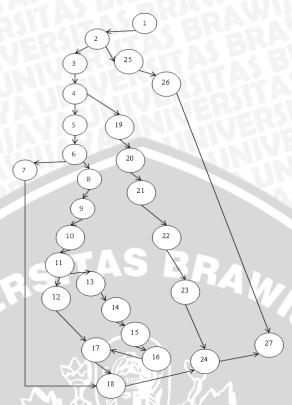
1. Algoritma Masuk Area Parkir.

Gambar 6.1 memaparkan algoritma masuk area parkir beserta node flowgraph.

```
1
     procedure Masuk Area Parkir
2
     Set mid
3
     uid = db select kendaraan_
4
     IF uid.nilai = 1
      Print json response kendaraan tidak diblokir
5
6
       Set status
       Status = db select data parkir
8
       IF status.nilai = 1
9
       Print json_response kendaraan pernah
```

# Gambar 6.1 Pembentukan Node Algoritma Masuk Area Parkir

Berdasarkan Gambar 6.1 maka dapat dibentuk *flowgraph* yang terdiri dari 27 *node*. Gambar 6.2 memperlihatkan *flowgraph* dari algoritma masuk area parkir.



### Gambar 6.2 Flowgraph Algoritma Masuk Area Parkir

Berdasarkan *flowgraph* algoritma masuk area parkir yang dapat dilihat pada Gambar 6.2, maka dapat dihitung nilai *cyclomatic complexity*.

$$V(G)$$
 = E - N + 2  
= 30 - 27 + 2  
= 5  
 $V(G)$  = P + 1

Sehingga berdasarkan hasil persamaan *cyclomatic complexity* maka didapatkan 5 jalur independen yaitu:

1. Jalur 
$$1 = 1 - 2 - 25 - 26 - 27$$

2. Jalur 
$$2 = 1 - 2 - 3 - 4 - 5 - 6 - 7 - 18 - 24 - 27$$

3. Jalur 
$$3 = 1 - 2 - 3 - 4 - 5 - 6 - 8 - 9 - 10 - 11 - 12 - 17 - 18 - 24 - 27$$

4. Jalur 
$$4 = 1 - 2 - 3 - 4 - 5 - 6 - 8 - 9 - 10 - 11 - 13 - 14 - 15 - 16 - 17 - 18 - 24 - 27$$

5. Jalur 
$$5 = 1 - 2 - 3 - 4 - 19 - 20 - 21 - 22 - 23 - 24 - 27$$

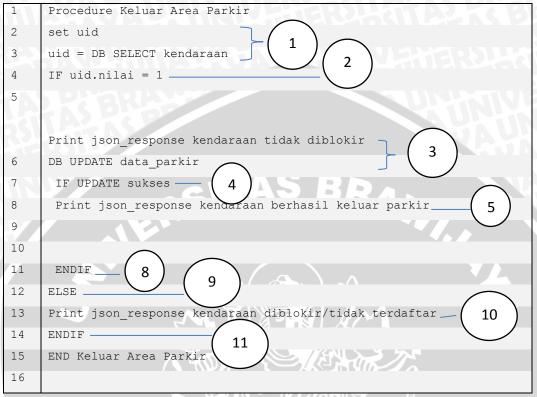
Berdasarkan 5 jalur independen yang telah didefinisikan tersebut maka dapat dibentuk kasus ujinya. Tabel 6.1 memaparkan kasus uji dari algoritma masuk area parkir.

Tabel 6.1 Kasus Uii Algoritma Masuk Area Parkir

111				itma Masuk A		Ct.
No	Jalur	Method	Data Input	Hasil yang Diharapkan	Hasil yang Diperoleh	Status
1 1 1 2 3 1 1	1 - 2 - 25 - 26 - 27	create_par kir()	Uid = 9-10 digit angka	Sistem tidak menginput data kendaraan dikarenakan kendaraan diblokir	Sistem tidak menginput data kendaraan	Valid
2	1-2-3- 4-5-6- 7-18- 24-27	create_par kir()	Uid = 9-10 digit angka	Sistem tidak menginput data kendaraan dikarenakan kendaraan masih berada dalam area parkir	Sistem tidak menginput data kendaraan	Valid
3	1-2-3- 4-5-6- 8-9-10-11-12-17-18-24-27	create_par kir()	Uid = 9-10 digit angka	Sistem tidak menginput data kendaraan dikarenakan area parkir penuh	Sistem tidak menginput data kendaraan	Valid
4	1-2-3- 4-5-6- 8-9-10-11-13-14-15-16-17-18-24-27	create_par kir()	Uid = 9-10 digit angka	Sistem menginput data kendaraan dikarenakan status parkir sebelumnya tidak berada dalam area parkir	Sistem menginput data kendaraan	Valid
5	1-2-3- 4-19- 20-21- 22-23- 24-27	create_par kir()	Uid = 9-10 digit angka	Sistem menginput data kendaraan dikarenakan status parkir sebelumnya tidak ada	Sistem menginput data kendaraan	Valid

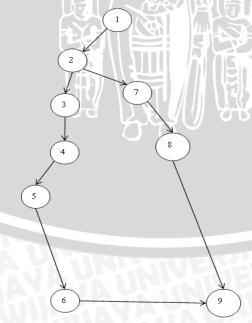
### 2. Algoritma Keluar Area Parkir.

Gambar 6.3 menggambarkan algoritma keluar area parkir beserta *node flowgraph*.



Gambar 6.3 Pembentukan Node Algoritma Keluar Area Parkir

Berdasarkan Gambar 6.3 maka dapat dibentuk *flowgraph* yang terdiri dari 11 *node*. Gambar 6.4 memperlihatkan *flowgraph* dari algoritma keluar area parkir.



Gambar 6.4 Flowgraph Algoritma Keluar Area Parkir

Berdasarkan *flowgraph* algoritma keluar area parkir yang dapat dilihat pada Gambar 6.4, maka dapat dihitung nilai *cyclomatic complexity*.

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

$$= 9 - 9 + 2$$

$$= 2$$

$$V(G) = P + 1$$

$$= 1 + 1$$

$$= 2$$

Sehingga berdasarkan hasil persamaan *cyclomatic complexity* maka didapatkan 2 jalur independen yaitu:

1. Jalur 
$$1 = 1 - 2 - 3 - 4 - 5 - 6 - 9$$

2. Jalur 
$$2 = 1 - 2 - 7 - 8 - 9$$

Berdasarkan 2 jalur independen yang telah didefinisikan tersebut maka dapat dibentuk kasus ujinya. Tabel 6.2 memaparkan kasus uji dari algoritma keluar area parkir.

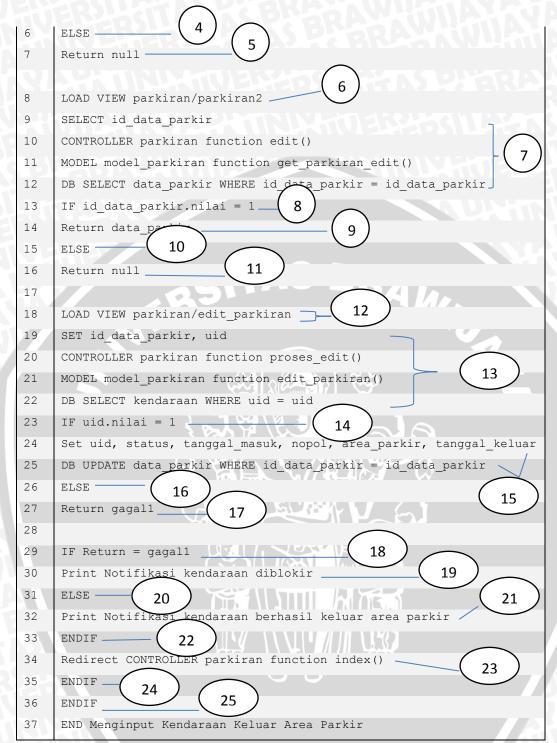
Tabel 6.2 Kasus Uji Algoritma Keluar Area Parkir

No	Jalur	Method	Data Input	Hasil yang Diharapkan	Hasil yang Diperoleh	Status
1	1-2-3- 4-5-6- 9	delete_par kir()	Uid = 9-10 digit angka	Sistem mengubah status parkir menjadi keluar	Status parkir berubah menjadi keluar	Valid
2	1-2-7-8-9	delete_par kir()	Uid = 9-10 digit angka	Sistem tidak mengubah status parkir menjadi keluar dikarenakan kendaraan diblokir	Status parkir tidak berubah	Valid

# 3. Algoritma Menginput Kendaraan Keluar Area Parkir

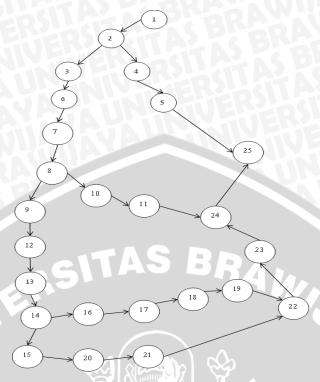
Gambar 6.5 menggambarkan algoritma menginput kendaraan keluar area parkir beserta *node flowgraph*.

	Procedure Menampilkan Kendaraan Meminta Keluar Area Parkir
1	CONTROLLER parkiran function tambah()
2	MODEL model_parkiran function get_parkiran2()
3	DB SELECT data_parkir WHERE status = meminta_keluar
4	IF status.nilai = 1(2)
5	Return data_parkir3



Gambar 6.5 Pembentukan Node Menginput Kendaraan Keluar Area Parkir

Berdasarkan Gambar 6.5 maka dapat dibentuk *flowgraph* yang terdiri dari 25 *node*. Gambar 6.6 memperlihatkan *flowgraph* dari algoritma menginput kendaraan keluar area parkir.



### Gambar 6.6 Flowgraph Algoritma Menginput Kendaraan Keluar Area Parkir

Berdasarkan *flowgraph* algoritma menginput kendaraan keluar area parkir yang dapat dilihat pada Gambar 6.6, maka dapat dihitung nilai *cyclomatic complexity*.

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

$$= 27 - 25 + 2$$

$$= 4$$

$$V(G) = P + 1$$

$$= 3 + 1$$

= 4

Sehingga berdasarkan hasil persamaan *cyclomatic complexity* maka didapatkan 4 jalur independen yaitu:

1. Jalur 
$$1 = 1 - 2 - 4 - 5 - 25$$

2. Jalur 
$$2 = 1 - 2 - 3 - 6 - 7 - 8 - 10 - 11 - 24 - 25$$

3. Jalur 
$$3 = 1 - 2 - 3 - 6 - 7 - 8 - 9 - 12 - 13 - 14 - 16 - 17 - 18 - 19 - 22 - 23 - 24 - 25$$

4. Jalur 
$$4 = 1 - 2 - 3 - 6 - 7 - 8 - 9 - 12 - 13 - 14 - 15 - 20 - 21 - 22 - 23 - 24 - 25$$

Berdasarkan 4 jalur independen yang telah didefinisikan tersebut maka dapat dibentuk kasus ujinya. Tabel 6.3 memaparkan kasus uji dari algoritma menginput kendaraan keluar area parkir.

Tabel 6.3 Kasus Uji Algoritma Menginput Kendaraan Keluar Area Parkir

No	Jalur	Kelas	Method	Data Input	Hasil yang Diharapkan	Hasil yang Diperoleh	Status
1	1 - 2 - 4 - 5 - 25	parkiran	tambah()		Sistem tidak menampilkan data kendaraan meminta keluar dikarenakan tidak ada data	Sistem berhasil tidak menampilkan data kendaraan meminta keluar	Valid
2	1-2-3 -6-7- 8-10- 11-24 -25	parkiran	tambah()	id_parkir = angka	Sistem tidak dapat memproses permintaan keluar area parkir dikarenakan data terhapus	Sistem berhasil tidak dapat memproses permintaan keluar area parkir	Valid
2	1-2-3- 6-7-8- 9-12- 13-14- 15-20- 21-22- 23-24- 25	Parkiran	tambah(), edit(),proses _edit()	Id_parkir = angka, uid = 9-10 digit angka, area_parkir = text ,nopol = text , tipe = text	Sistem menampilkan notifikasi kendaran berhasil keluar area parkir	Sistem berhasil menampilkan notifikasi kendaran berhasil keluar area parkir	Valid
2	1-2-3- 6-7-8- 9-12- 13-14- 16-17- 18-19- 22-23- 24-25	parkiran	tambah(), edit(),proses _edit()	Id_parkir = angka, uid = 9-10 digit angka, area_parkir = text ,nopol = text , tipe = text	Sistem menampilkan notifikasi kendaraan diblokir	Sistem berhasil menampilkan notifikasi kendaraan diblokir	Valid

### 6.1.4 Analisis

Berdasarkan hasil pengujian yang telah dilakukan pada algoritma masuk area parkir, keluar area parkir dan menginput kendaraan keluar area parkir bahwa seluruh kasus uji memiliki hasil yang telah sesuai dengan yang diharapkan atau bernilai valid.

# 6.2 Pengujian Black-Box

#### 6.2.1 Validasi

#### 6.2.1.1 Tujuan

Tujuan dilakukannya pengujian black-box adalah untuk memastikan setiap spesifikasi kebutuhan perangkat lunak yang didefinisikan telah sesuai dengan yang diharapkan.

#### 6.2.1.2 Mekanisme

Setiap spesifikasi kebutuhan perangkat lunak yang didefinisikan pada tahap analisis kebutuhan akan diujikan pada pengujian ini dengan cara mendefinisikan kasus uji terhadap setiap kebutuhan tersebut lalu membandingkannya dengan hasil yang diperoleh.

Tabel 6.4 menjelaskan kasus uji dari pengujian validasi pada fungsi masuk area parkir.

Tabel 6.4 Kasus Uji Validasi Masuk Area Parkir

rabel of Rasas of Validasi Masak Area Farkii		
Nomor Kasus Uji	VAL_001	
Nama Kasus Uji	Kasus uji validasi masuk area parkir	
Nomor SKPL	SKPL_01_1	
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat mengenali pengendara dengan identitas kartu nfc untuk masuk area parkir	
Prosedur Uji	<ol> <li>Penguji mendekatkan kartu nfc pada android reader</li> <li>Sistem akan menampilkan uid dari kartu nfc</li> <li>Penguji menekan tombol masuk area parkir</li> </ol>	
Hasil yang Diharapkan	True: Input dari pengendara untuk masuk area parkir dapat tersimpan dalam sistem  False: Bila kartu tidak terdaftar maka input dari pengendara tidak dapat tersimpan dalam sistem	

Tabel 6.5 menjelaskan kasus uji dari pengujian validasi pada fungsi melihat status pemesanan menu.

Tabel 6.5 Kasus Uji Validasi Keluar Area Parkir

Nomor Kasus Uji	VAL_002			
Nama Kasus Uji	Kasus uji validasi keluar area parkir			
Nomor SKPL	SKPL_01_2			
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat mengenali pengendara dengan identitas kartu nfc untuk keluar area parkir			
Prosedur Uji	1. Penguji mendekatkan kartu nfc pada android			

UKTIVE KER	reader
UKINIXA	2. Sistem akan menampilkan uid dari kartu nfc
MAGAUNA	3. Penguji menekan tombol keluar area parkir
Hasil yang Diharapkan	True: Input dari pengendara untuk keluar area parkir dapat tersimpan dalam sistem
BRANN	False: Bila kartu tidak terdaftar maka input dari pengendara tidak dapat tersimpan dalam sistem

Tabel 6.6 menjelaskan kasus uji dari pengujian validasi pada fungsi keluar area kampus.

Tabel 6.6 Kasus Uji Validasi Keluar Area Kampus

Nomor Kasus Uji	VAL_003
Nama Kasus Uji	Kasus uji validasi keluar area kampus
Nomor SKPL	SKPL_02
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat mengenali pengendara dengan identitas kartu nfc untuk keluar area kampus
Prosedur Uji	<ol> <li>Penguji mendekatkan kartu nfc pada android reader</li> <li>Sistem akan menampilkan uid dari kartu nfc</li> <li>Penguji menekan tombol keluar area kampus</li> </ol>
Hasil yang Diharapkan	True: Input dari pengendara untuk keluar area kampus dapat tersimpan dalam sistem  False: Bila kartu tidak terdaftar maka input dari pengendara tidak dapat tersimpan dalam sistem

Tabel 6.7 menjelaskan kasus uji dari pengujian validasi pada fungsi login.

Tabel 6.7 Kasus Uii Validasi Login

Tabel 6.7 Kasus Oji Validasi Logili	
Nomor Kasus Uji	VAL_004)
Nama Kasus Uji	Kasus uji validasi login
Nomor SKPL	SKPL_03
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat diakses oleh pengguna yang berwenang
Prosedur Uji	<ol> <li>Penguji masuk ke halaman login</li> <li>Penguji mengisi form input username dan password, kemudian tekan submit</li> </ol>
Hasil yang Diharapkan	True: Sistem dapat diakses oleh pengguna yang berwenang False: Bila username/password tidak terdaftar maka
Z.T.C.BR.CO.	sistem tidak dapat diakses

Tabel 6.8 menjelaskan kasus uji dari pengujian validasi pada fungsi menginput kendaraan masuk parkir.

Tabel 6.8 Kasus Uji Validasi Menginput Kendaraan Masuk Parkir

Nomor Kasus Uji	VAL_005
Nama Kasus Uji	Kasus uji validasi menginput kendaraan masuk parkir
Nomor SKPL	SKPL_04_1
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat menginput kendaraan masuk area parkir oleh petugas parkir
Prosedur Uji	<ol> <li>Penguji memilih halaman input kendaraan masuk</li> <li>Sistem menampilkan list data kendaraan meminta masuk</li> <li>Penguji memilih data kendaraan kemudian tekan proses</li> </ol>
Hasil yang Diharapkan	True : Sistem menampilkan notifikasi kendaraan berhasil masuk area parkir False : Sistem menampilkan notifikasi kendaraan diblokir

Tabel 6.9 menjelaskan kasus uji dari pengujian validasi pada fungsi menginput kendaraan keluar parkir.

Tabel 6.9 Kasus Uji Validasi Menginput Kendaraan Keluar Parkir

Nomor Kasus Uji	VAL_006
Nama Kasus Uji	Kasus uji validasi menginput kendaraan keluar area parkir
Nomor SKPL	SKPL_04_2
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat menginput kendaraan keluar area parkir oleh petugas parkir
Prosedur Uji	<ol> <li>Penguji memilih halaman input kendaraan keluar</li> <li>Sistem menampilkan list data kendaraan meminta keluar</li> <li>Penguji memilih data kendaraan kemudian tekan proses</li> <li>Sistem akan menampilkan detail kendaraan yang akan keluar</li> <li>Penguji menekan tombol submit</li> </ol>
Hasil yang Diharapkan	True : Sistem menampilkan notifikasi kendaraan berhasil keluar area parkir False : Sistem menampilkan notifikasi kendaraan diblokir

Tabel 6.10 menjelaskan kasus uji dari pengujian validasi pada fungsi melihat aktivitas parkir.

Tabel 6.10 Kasus Uji Validasi Melihat Aktivitas Parkir

Nomor Kasus Uji	VAL_007
Nama Kasus Uji	Kasus uji validasi melihat aktivitas parkir
Nomor SKPL	SKPL_05
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat melihat aktivitas keluar-masuk kendaraan pada area parkir
Prosedur Uji	Penguji memilih halaman list aktivitas parkiran
Hasil yang Diharapkan	Sistem dapat menampilkan data aktivitas pada area parkir

Tabel 6.11 menjelaskan kasus uji dari pengujian validasi pada fungsi membuat laporan kehilangan.

Tabel 6.11 Kasus Uji Validasi Membuat laporan kehilangan

Tabel 6.11 K	asus Uji Validasi Membuat laporan kenilangan
Nomor Kasus Uji	VAL_008
Nama Kasus Uji	Kasus uji validasi membuat laporan kehilangan
Nomor SKPL	SKPL_06
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat membuat laporan kehilangan kendaraan
Prosedur Uji	<ol> <li>Penguji masuk ke halaman utama petugas parkir</li> <li>Penguji memilih menu buat laporan kehilangan</li> <li>Penguji mengisi form pencari identitas pengendara</li> <li>Penguji melanjutkan mengisi detail laporan kehilangan.</li> </ol>
Hasil yang Diharapkan	True : Laporan kehilangan berhasil dibuat dan tersimpan pada database  False : Data identitas pengendara tidak ditemukan,
作R.	dan laporan kehilangan tidak dapat dibuat

Tabel 6.12 menjelaskan kasus uji dari pengujian validasi pada fungsi membuat laporan ditemukan.

Tabel 6.12 Kasus Uji Validasi Membuat laporan ditemukan

Nomor Kasus Uji	VAL_009
Nama Kasus Uji	Kasus uji validasi membuat laporan ditemukan
Nomor SKPL	SKPL_07
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat membuat laporan ditemukan barang
Prosedur Uji	1. Penguji masuk ke halaman utama petugas

LATIVELY TO	parkir
	2. Penguji memilih menu buat laporan ditemukan
MAJAUNS	3. Penguji mengisi form laporan lalu tekan submit
Hasil yang Diharapkan	True : Laporan ditemukan berhasil dibuat dan tersimpan pada database
SOAWKIII	False : Data laporan ditemukan tidak dapat dibuat

Tabel 6.13 menjelaskan kasus uji dari pengujian validasi pada fungsi klaim laporan ditemukan.

Tabel 6.13 Kasus Uji Validasi Klaim Laporan Ditemukan

Nomor Kasus Uji	VAL_010
Nama Kasus Uji	Kasus uji validasi klaim laporan ditemukan
Nomor SKPL	SKPL_08
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat membuat klaim laporan ditemukan barang
Prosedur Uji	<ol> <li>Penguji masuk ke halaman utama petugas parkir</li> <li>Penguji memilih menu list laporan ditemukan</li> <li>Penguji memilih data yang akan di klaim</li> <li>Penguji melanjutkan mengisi detail laporan klaim lalu tekan submit.</li> </ol>
Hasil yang Diharapkan	True: Klaim laporan ditemukan berhasil dibuat dan tersimpan pada database False: Klaim laporan ditemukan tidak dapat dibuat

Tabel 6.14 menjelaskan kasus uji dari pengujian validasi pada fungsi mengedit profil.

Tabel 6.14 Kasus Uji Validasi Mengedit Profil

	or religion to the state of the
Nomor Kasus Uji	VAL_011
Nama Kasus Uji	Kasus uji validasi mengedit profil
Nomor SKPL	SKPL_09
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat mengedit profil
Prosedur Uji	Penguji masuk ke halaman edit profil
	2. Penguji mengisi <i>form</i> edit profil
WALLE	3. Penguji kemudian tekan submit
Hasil yang Diharapkan	True : Edit profil berhasil dan tersimpan pada database
RAYKWILL	False : Edit profil gagal

Tabel 6.15 menjelaskan kasus uji dari pengujian validasi pada fungsi lupa password.

Tabel 6.15 Kasus Uji Validasi Lupa Password

Nomor Kasus Uji	VAL_012
Nama Kasus Uji	Kasus uji validasi lupa password
Nomor SKPL	SKPL_10
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat mengirim ulang password
Prosedur Uji	<ol> <li>Penguji masuk ke halaman utama login</li> </ol>
TAZKI BKS	2. Penguji memilih menu lupa password
2511/37 AS	3. Penguji mengisi form pencari email
ALERS HAVE	4. Penguji melanjutkan tekan submit.
	5. Jika email valid, tekan submit kembali
Hasil yang Diharapkan	True : Password berhasil dikirim kepada email
JE	False : Data email tidak ditemukan, dan password tidak dapat dikirim

Tabel 6.16 menjelaskan kasus uji dari pengujian validasi pada fungsi blokir kendaraan.

Tabel 6.16 Kasus Uji Validasi Blokir Kendaraan

ruber 0.10 Rubus of Vandusi blokii Kenduruan		
Nomor Kasus Uji	VAL_013	
Nama Kasus Uji	Kasus uji validasi blokir kendaraan	
Nomor SKPL	SKPL_11_1	
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat memblokir kendaraan yang telah dilaporkan	
Prosedur Uji	<ol> <li>Penguji masuk ke halaman petugas keamanan</li> <li>Penguji memilih menu list laporan masuk</li> <li>Penguji memilih laporan kehilangan kemudian tekan tombol lakukan tindakan</li> </ol>	
Hasil yang Diharapkan	Sistem mengubah status kendaraan dari aktif menjadi blokir	

Tabel 6.17 menjelaskan kasus uji dari pengujian validasi pada fungsi cabut blokir kendaraan.

Tabel 6.17 Kasus Uji Validasi Cabut Blokir Kendaraan

Nomor Kasus Uji	VAL_014	
Nama Kasus Uji	Kasus uji validasi cabut blokir kendaraan	
Nomor SKPL	SKPL_11_2	
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat mencabut blokir pada kendaraan	
Prosedur Uji	<ol> <li>Penguji masuk ke halaman petugas keamanan</li> <li>Penguji memilih menu list blokir kendaraan</li> <li>Penguji memilih data kendaraan kemudian tekan</li> </ol>	

	tor	nbol cabut bl	okir		W	ATT
Hasil yang Diharapkan	Sistem menjad		status	kendaraan	dari	blokir

Tabel 6.18 menjelaskan kasus uji dari pengujian validasi pada fungsi memproses laporan kehilangan.

Tabel 6.18 Kasus Uji Validasi Memproses Laporan Kehilangan

Nomor Kasus Uji	VAL_015		
Nama Kasus Uji	Kasus uji validasi memproses laporan kehilangan		
Nomor SKPL	SKPL_12		
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat melakukan proses laporan kehilangan yang telah divalidasi oleh koordinator parkir		
Prosedur Uji	Penguji masuk ke halaman utama petugas keamanan		
	Penguji memilih menu list laporan masuk		
5	3. Penguji memilih laporan yang akan diproses, kemudian tekan tombol lakukan tindakan		
Hasil yang Diharapkan	Sistem dapat memproses laporan kehilangan		

Tabel 6.19 menjelaskan kasus uji dari pengujian validasi pada fungsi menginput kendaraan keluar kampus.

**Tabel 6.19 Kasus Uji Menginput Kendaraan Keluar Kampus** 

Nomor Kasus Uji	VAL_016		
Nama Kasus Uji	Kasus uji validasi menginput kendaraan keluar kampus		
Nomor SKPL	SKPL_13		
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat menginput kendaraan keluar area parkir oleh petugas keamanan		
Prosedur Uji	<ol> <li>Penguji memilih halaman input kendaraan keluar kampus</li> <li>Sistem menampilkan list data kendaraan on gate</li> <li>Penguji memilih data kendaraan kemudian tekan proses</li> </ol>		
	<ul><li>4. Sistem akan menampilkan detail kendaraan yang akan keluar</li><li>5. Penguji menekan tombol submit</li></ul>		
Hasil yang Diharapkan	True : Sistem menampilkan notifikasi kendaraan berhasil keluar area kampus		
ITAS BR	False : Sistem menampilkan notifikasi kendaraan diblokir		

Tabel 6.20 menjelaskan kasus uji dari pengujian validasi pada fungsi mengelola data pengendara.

Tabel 6.20 Kasus Uji Validasi Mengelola Data Pengendara

Tabel 0.20	Kasus Oji validasi Mengelola Data Pengendara		
Nomor Kasus Uji	VAL_017		
Nama Kasus Uji	Kasus uji validasi mengelola data pengendara		
Nomor SKPL	SKPL_14		
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat mengelola data pengendara		
Prosedur Uji	Penguji masuk ke halaman utama koordinator parkir		
	2. Penguji memilih menu pengendara, kemudian tekan list pengendara maka akan tampil list data pengendara		
11/6	3. Jika penguji ingin menambah data pengendara, maka penguji memilih menu tambah pengendara		
5	4. Jika penguji ingin mengubah data pengendara, maka penguji memilih data pengendara yang akan diubah kemudian tekan tombol edit		
	5. Jika penguji ingin menghapus data pengendara, maka penguji memilih data pengendara yang akan dihapus kemudian tekan tombol delete		
Hasil yang Diharapkan	True: Sistem dapat membuat, melihat, mengubah, dan menghapus data pengendara		

Tabel 6.21 menjelaskan kasus uji dari pengujian validasi pada fungsi mengelola data kendaraan.

Tabel 6.21 Kasus Uii Validasi Mengelola Data Kendaraan

Tabel 0.21 Rasus Of Validasi Wellgelola Data Kelidaraan			
Nomor Kasus Uji	VAL_018		
Nama Kasus Uji	Kasus uji validasi mengelola data kendaraan		
Nomor SKPL	SKPL_15		
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat mengelola data kendaraan		
Prosedur Uji	<ol> <li>Penguji masuk ke halaman utama koordinator parkir</li> <li>Penguji memilih menu kendaraan, kemudian tekan list kendaraan maka akan tampil list data kendaraan</li> <li>Jika penguji ingin menambah data kendaraan, maka penguji memilih menu tambah kendaraan</li> <li>Jika penguji ingin mengubah data kendaraan, maka penguji memilih data kendaraan yang akan</li> </ol>		

	diubah kemudian tekan tombol edit
	<ol> <li>Jika penguji ingin menghapus data kendaraan, maka penguji memilih data kendaraan yang akan dihapus kemudian tekan tombol delete</li> </ol>
Hasil yang Diharapkan	Sistem dapat membuat, melihat, mengubah, dan menghapus data kendaraan

Tabel 6.22 menjelaskan kasus uji dari pengujian validasi pada fungsi mengelola data petugas parkir.

Tabel 6.22 Kasus Uji Validasi Mengelola Data Petugas Parkir

Nomor Kasus Uji	VAL_019	
Nama Kasus Uji	Kasus uji validasi mengelola data petugas parkir	
Nomor SKPL	SKPL_16	
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat mengelola data petugas parkir	
Prosedur Uji	<ol> <li>Penguji masuk ke halaman utama koordinator parkir</li> <li>Penguji memilih menu petugas, kemudian tekan list petugas maka akan tampil list data petugas</li> </ol>	
	Jika penguji ingin menambah data petugas, maka penguji memilih menu tambah petugas	
	4. Jika penguji ingin mengubah data petugas, maka penguji memilih data petugas yang akan diubah kemudian tekan tombol edit	
	5. Jika penguji ingin menghapus data petugas, maka penguji memilih data petugas yang akan dihapus kemudian tekan tombol delete	
Hasil yang Diharapkan	Sistem dapat membuat, melihat, mengubah, dan menghapus data petugas parkir	

Tabel 6.23 menjelaskan kasus uji dari pengujian validasi pada fungsi memvalidasi laproan kehilangan.

Tabel 6.23 Kasus Uji Validasi Memvalidasi Laporan Kehilangan

Nomor Kasus Uji	VAL_020		
Nama Kasus Uji	Kasus uji validasi memvalidasi laporan kehilangan		
Nomor SKPL	SKPL_17		
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat melakukan validasi laporan kehilangan yang telah dilaporkan oleh petugas parkir		
Prosedur Uji	Penguji masuk ke halaman utama koordinator parkir		
ITELY VS BL	2. Penguji memilih menu list laporan kehilangan		

	<ol> <li>Penguji memilih laporan yang akan divalidasi, kemudian tekan tombol detail dan muncul form laporan yang dibuat oleh petugas parkir, lalu tekan submit</li> </ol>	
Hasil yang Diharapkan	Sistem dapat melakukan validasi laporan kehilangan	

Tabel 6.24 menjelaskan kasus uji dari pengujian validasi pada fungsi mengelola data area parkir.

Tabel 6.24 Kasus Uji Validasi Mengelola Data Area Parkir

Nomor Kasus Uji	VAL_021	
Nama Kasus Uji	Kasus uji validasi mengelola data area parkir	
Nomor SKPL	SKPL_18	
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat mengelola data area parkir	
Prosedur Uji	<ol> <li>Penguji masuk ke halaman utama koordinato parkir</li> <li>Penguji memilih menu area parkir, kemudiat tekan list area parkir maka akan tampil list data area parkir</li> <li>Jika penguji ingin menambah data area parkir maka penguji memilih menu tambah area parkir</li> <li>Jika penguji ingin mengubah data area parkir maka penguji memilih data area parkir yang akan diubah kemudian tekan tombol edit</li> <li>Jika penguji ingin menghapus data area parkir maka penguji memilih data area parkir yang akan dihapus kemudian tekan tombol delete</li> </ol>	
Hasil yang Diharapkan	Sistem dapat membuat, melihat, mengubah, dan menghapus data area parkir	

Tabel 6.25 menjelaskan kasus uji dari pengujian validasi pada fungsi memvalidasi laproan ditemukan.

Tabel 6.25 Kasus Uji Validasi Memvalidasi Laporan Ditemukan

Nomor Kasus Uji	VAL_022
Nama Kasus Uji	Kasus uji validasi memvalidasi laporan ditemukan
Nomor SKPL	SKPL_19
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat melakukan validasi laporan ditemukan yang telah dilaporkan oleh petugas parkir
Prosedur Uji	Penguji masuk ke halaman utama koordinator parkir
ITAZAS BK	2. Penguji memilih menu list laporan ditemukan

	<ol> <li>Penguji memilih laporan yang akan divalidasi, kemudian tekan tombol detail dan muncul form laporan yang dibuat oleh petugas parkir, lalu tekan submit</li> </ol>	
Hasil yang Diharapkan	Sistem dapat melakukan validasi laporan ditemukan	

# 6.2.1.3 Hasil Pengujian

Tabel 6.26 memaparkan hasil pengujian validasi yang telah dilakukan terhadap setiap kasus uji yang telah didefinisikan pada tahap sebelumnya.

Tabel 6.26 Hasil Pengujian Validasi

No.	Nama Kasus Uji	Hasil yang	Hasil yang Diperoleh	Status
Kasus Uji	, a	Diharapkan	RAW.	
VAL_0 01	Kasus uji validasi masuk area parkir	True : Input dari pengendara untuk masuk area parkir dapat tersimpan dalam sistem	True : Input dari pengendara untuk masuk area parkir tersimpan dalam sistem	Valid
	Q Q	False: input dari pengendara tidak dapat tersimpan dalam sistem	False : input dari pengendara tidak tersimpan dalam sistem	
VAL_0 02	Kasus uji validasi keluar area parkir	True : Input dari pengendara untuk keluar area parkir dapat tersimpan dalam sistem	True : Input dari pengendara untuk keluar area parkir tersimpan dalam sistem	Valid
		False : input dari pengendara tidak dapat tersimpan dalam sistem	False : input dari pengendara tidak tersimpan dalam sistem	
VAL_0 03	Kasus uji validasi keluar area kampus	True : Input dari pengendara untuk keluar area kampus dapat tersimpan dalam sistem	True : Input dari pengendara untuk keluar area kampus tersimpan dalam sistem	Valid
		False : input dari pengendara tidak dapat tersimpan dalam sistem		
VAL_0 04	Kasus uji validasi login	True : Sistem dapat melakukan autentifikasi	True : Sistem melakukan autentifikasi dengan	Valid

VAL_0   Kasus uji validasi menginput kendaraan masuk parkir   False: Sistem dapat menampilkan notifikasi menginput kendaraan masuk area parkir   False: Sistem dapat menampilkan notifikasi menampilkan notifikasi kendaraan diblokir   VAL_0   Kasus uji validasi melihat aktivitas parkir   VAL_0   Kasus uji validasi membuat menampilkan notifikasi membuat laporan kehilangan ditemukan ditemukan   MAL_0   Kasus uji validasi membuat laporan ditemukan ditemukan ditemukan   MAL_0   Kasus uji validasi memampilkan notifikasi keluaraan kehilangan ditemukan ditemukan ditemukan ditemukan ditemukan ditemukan berhasil dibuat   VAL_0   Kasus uji validasi memampilkan notifikasi laporan ditemukan ditemukan ditemukan ditemukan ditemukan ditemukan ditemukan berhasil dibuat   VAL_0   Kasus uji validasi memampilkan notifikasi laporan ditemukan ditemukan ditemukan ditemukan ditemukan ditemukan ditemukan ditemukan berhasil dibuat   VAL_0   Kasus uji validasi memampilkan notifikasi laporan ditemukan ditemukan ditemukan ditemukan ditemukan ditemukan ditemukan berhasil dibuat   VAL_0   Kasus uji validasi menampilkan notifikasi laporan ditemukan ditemukan berhasil dibuat   VAL_0   Kasus uji validasi dibuat   Valid menampilkan notifikasi laporan ditemukan berhasil dibuat   Valid menampilkan notifikasi klaim laporan ditemukan berhasil dibuat   Valid menampilkan notifikasi klaim laporan ditemukan berhasil dibuat   Valid mengedit profil   Valid sistem berhasil dibuat   Valid mengedit profil   Valid sistem berhasil dibuat   Valid sistem berhasil   Valid sistem berhasil dibuat   Valid sistem berhasil dibuat   Valid sistem berhasil dibuat   Valid sistem berhasil   Valid	111/12		TAD PLOK		1 1 1
VAL_0 Kasus uji validasi menginput kendaraan notifikasi kendaraan notifikasi menampilkan notifikasi kendaraan berhasil masuk area parkir False : Sistem dapat menampilkan notifikasi kendaraan ditemukan parkir False : Sistem dapat menampilkan notifikasi kendaraan ditemukan potifikasi kendaraan ditemukan potifikasi kendaraan ditemukan potifikasi kendaraan notifikasi kendaraan notifikasi kendaraan ditemukan parkir False : Sistem dapat menampilkan notifikasi kendaraan perhasil keluar area parkir False : Sistem dapat menampilkan notifikasi kendaraan notifikasi kendaraan perhasil keluar area parkir False : Sistem dapat menampilkan notifikasi kendaraan ditemukan ditemukan ditemukan parkir ditemukan perhasil dibuat Sistem dapat menampilkan notifikasi laporan notifikasi laporan notifikasi laporan ditemukan ditemukan ditemukan ditemukan ditemukan perhasil dibuat Sistem dapat menampilkan notifikasi laporan notifikasi laporan notifikasi laporan notifikasi laporan notifikasi laporan notifikasi kendaraan ditemukan d		MEHEROL	dengan benar	benar	TIVE
menginput kendaraan notifikasi kendaraan berhasil masuk parkir  False: Sistem dapat menampilkan notifikasi kendaraan berhasil parkir  VAL_0 Kasus uji validasi menginput kendaraan diblokir  VAL_0 Kasus uji validasi menginput kendaraan keluar parkir  VAL_0 Kasus uji validasi menginput menampilkan notifikasi kendaraan keluar parkir  VAL_0 Kasus uji validasi menampilkan notifikasi kendaraan keluar area parkir  False: Sistem dapat menampilkan notifikasi kendaraan kendaraan diblokir  VAL_0 Kasus uji validasi melihat aktivitas menampilkan aktivitas keluar masuk kendaraan dalam area parkir  VAL_0 Kasus uji validasi melihat aktivitas keluar masuk kendaraan dalam area parkir  VAL_0 Kasus uji validasi membuat laporan kehilangan kehilangan berhasil dibuat  VAL_0 Kasus uji validasi membuat laporan notifikasi laporan notifikasi laporan kehilangan kehilangan berhasil dibuat  VAL_0 Kasus uji validasi membuat laporan ditemukan ditemukan berhasil dibuat  VAL_0 Kasus uji validasi kalaim laporan ditemukan berhasil dibuat  VAL_0 Kasus uji validasi kalaim laporan ditemukan berhasil dibuat  VAL_0 Kasus uji validasi kalaim laporan ditemukan berhasil dibuat  VAL_0 Kasus uji validasi kalaim laporan ditemukan berhasil dibuat  VAL_0 Kasus uji validasi kalaim laporan ditemukan berhasil dibuat  VAL_0 Kasus uji validasi kalaim laporan ditemukan berhasil dibuat  VAL_0 Kasus uji validasi kalaim laporan ditemukan berhasil dibuat  VAL_0 Kasus uji validasi kalaim laporan ditemukan berhasil dibuat  VAL_0 Kasus uji validasi kalaim laporan ditemukan berhasil dibuat  VAL_0 Kasus uji validasi kalim laporan ditemukan berhasil dibuat  VAL_0 Kasus uji validasi kalim laporan ditemukan berhasil dibuat  VAL_0 Kasus uji validasi kalim laporan ditemukan berhasil dibuat					
Notifikasi kendaraan diblokir  VAL_O Kasus uji validasi menampilkan notifikasi kendaraan keluar parkir  VAL_O Kasus uji validasi menampilkan notifikasi kendaraan berhasil keluar area parkir  VAL_O Kasus uji validasi melihat aktivitas parkir  VAL_O Kasus uji validasi membuat menampilkan laporan ditemukan ditemukan ditemukan berhasil dibuat  VAL_O Kasus uji validasi membuat menampilkan notifikasi laporan kehilangan  VAL_O Kasus uji validasi membuat menampilkan notifikasi laporan kehilangan kehilangan berhasil dibuat  VAL_O Kasus uji validasi membuat menampilkan notifikasi laporan kehilangan berhasil dibuat  VAL_O Kasus uji validasi membuat menampilkan notifikasi laporan kehilangan berhasil dibuat  VAL_O Kasus uji validasi membuat menampilkan notifikasi laporan kehilangan berhasil dibuat  VAL_O Kasus uji validasi membuat menampilkan notifikasi laporan ditemukan berhasil dibuat  VAL_O Kasus uji validasi Sistem dapat klaim laporan ditemukan berhasil dibuat  VAL_O Kasus uji validasi Sistem dapat menampilkan notifikasi klaim laporan ditemukan berhasil dibuat  VAL_O Kasus uji validasi Sistem dapat menampilkan notifikasi klaim laporan ditemukan berhasil dibuat  VAL_O Kasus uji validasi Sistem dapat menampilkan notifikasi klaim laporan ditemukan berhasil dibuat  VAL_O Kasus uji validasi Sistem dapat menampilkan notifikasi klaim laporan ditemukan berhasil dibuat  VAL_O Kasus uji validasi Sistem dapat menampilkan notifikasi klaim laporan ditemukan berhasil dibuat  VAL_O Kasus uji validasi Sistem dapat menampilkan notifikasi klaim laporan ditemukan berhasil dibuat	_	menginput kendaraan	menampilkan notifikasi kendaraan berhasil masuk area parkir False : Sistem dapat	menampilkan notifikasi kendaraan berhasil masuk area parkir False : Sistem	Valid
menginput kendaraan keluar parkir kendaraan keluar parkir False : Sistem dapat menampilkan notifikasi kendaraan berhasil keluar area parkir False : Sistem dapat menampilkan notifikasi kendaraan diblokir Sistem dapat menampilkan aktivitas keluar masuk kendaraan dalam area parkir dalam area parkir Sistem menampilkan aktivitas keluar masuk kendaraan dalam area parkir Sistem menampilkan notifikasi laporan kehilangan berhasil dibuat Sistem dapat menampilkan notifikasi laporan kehilangan berhasil dibuat Sistem dapat menampilkan notifikasi laporan kehilangan berhasil dibuat Sistem dapat menampilkan notifikasi laporan ditemukan ditemukan berhasil dibuat Sistem dapat menampilkan notifikasi laporan ditemukan berhasil dibuat Sistem dapat klaim laporan ditemukan berhasil dibuat Sistem dapat Sistem menampilkan notifikasi klaim laporan ditemukan berhasil dibuat Sistem dapat Klaim laporan ditemukan berhasil dibuat Sistem dapat Sistem menampilkan notifikasi klaim laporan ditemukan berhasil dibuat Sistem dapat Sistem menampilkan notifikasi klaim laporan ditemukan berhasil dibuat Sistem dapat Sistem menampilkan notifikasi klaim laporan ditemukan berhasil dibuat Sistem dapat Sistem menampilkan notifikasi klaim laporan ditemukan berhasil dibuat Sistem dapat Sistem menampilkan notifikasi klaim laporan ditemukan berhasil dibuat Sistem dapat Edit profil pada Valid			notifikasi	notifikasi kendaraan	
menampilkan notifikasi kendaraan diblokir  VAL_0 Kasus uji validasi melihat aktivitas parkir  VAL_0 Kasus uji validasi menampilkan aktivitas keluar-masuk kendaraan dalam area parkir  VAL_0 Kasus uji validasi membuat menampilkan notifikasi laporan kehilangan membuat laporan kehilangan berhasil dibuat  VAL_0 Kasus uji validasi membuat menampilkan notifikasi laporan kehilangan berhasil dibuat  VAL_0 Kasus uji validasi membuat menampilkan notifikasi laporan kehilangan berhasil dibuat  VAL_0 Kasus uji validasi menampilkan notifikasi laporan ditemukan berhasil dibuat  VAL_0 Kasus uji validasi klaim laporan ditemukan berhasil dibuat  VAL_0 Kasus uji validasi klaim laporan ditemukan berhasil dibuat  VAL_0 Kasus uji validasi si klaim laporan ditemukan berhasil dibuat  VAL_0 Kasus uji validasi si klaim laporan ditemukan berhasil dibuat  VAL_0 Kasus uji validasi si klaim laporan ditemukan berhasil dibuat  VAL_0 Kasus uji validasi si klaim laporan ditemukan berhasil dibuat  VAL_0 Kasus uji validasi si klaim laporan ditemukan berhasil dibuat  VAL_0 Kasus uji validasi si klaim laporan ditemukan berhasil dibuat  VAL_0 Kasus uji validasi si klaim laporan ditemukan berhasil dibuat  VAL_0 Kasus uji validasi si klaim laporan ditemukan berhasil dibuat	_	menginput kendaraan	menampilkan notifikasi kendaraan berhasil keluar area parkir	menampilkan notifikasi kendaraan berhasil keluar area parkir	Valid
melihat aktivitas parkir  menampilkan aktivitas keluar-masuk kendaraan dalam area parkir  VAL_0 Kasus uji validasi membuat laporan kehilangan berhasil dibuat  VAL_0 Kasus uji validasi membuat laporan kehilangan berhasil dibuat  VAL_0 Kasus uji validasi membuat menampilkan notifikasi laporan kehilangan berhasil dibuat  VAL_0 Kasus uji validasi membuat menampilkan notifikasi laporan ditemukan berhasil dibuat  VAL_0 Kasus uji validasi si klaim laporan ditemukan berhasil dibuat  VAL_0 Kasus uji validasi klaim laporan ditemukan berhasil dibuat  VAL_0 Kasus uji validasi si sistem dapat menampilkan notifikasi klaim laporan ditemukan berhasil dibuat  VAL_0 Kasus uji validasi sistem dapat menampilkan notifikasi klaim laporan ditemukan berhasil dibuat  VAL_0 Kasus uji validasi Sistem dapat berhasil dibuat		<b>5</b>	menampilkan notifikasi	menampilkan notifikasi kendaraan	
membuat laporan kehilangan berhasil dibuat  VAL_0 Kasus uji validasi memampilkan laporan kehilangan berhasil dibuat  VAL_0 Kasus uji validasi memampilkan notifikasi laporan ditemukan berhasil dibuat  VAL_0 Kasus uji validasi Sistem dapat laporan ditemukan berhasil dibuat  VAL_0 Kasus uji validasi Sistem dapat laporan ditemukan berhasil dibuat  VAL_0 Kasus uji validasi Sistem dapat laporan ditemukan berhasil dibuat  VAL_0 Kasus uji validasi Sistem dapat laporan ditemukan berhasil dibuat		melihat aktivitas	menampilkan aktivitas keluar- masuk kendaraan	aktivitas keluar- masuk kendaraan	Valid
membuat laporan ditemukan berhasil dibuat  VAL_0 Kasus uji validasi klaim laporan ditemukan berhasil ditemukan berhasil dibuat  VAL_0 Kasus uji validasi klaim laporan ditemukan berhasil dibuat  VAL_0 Kasus uji validasi Sistem dapat Sistem menampilkan notifikasi klaim laporan ditemukan laporan ditemukan berhasil dibuat  VAL_0 Kasus uji validasi Sistem dapat Edit profil pada Valid	_	membuat laporan	menampilkan notifikasi laporan kehilangan berhasil	notifikasi laporan kehilangan berhasil	Valid
klaim laporan ditemukan laporan ditemukan laporan ditemukan berhasil dibuat  VAL_0 Kasus uji validasi Sistem dapat Edit profil pada Valid		membuat laporan	menampilkan notifikasi laporan ditemukan berhasil	notifikasi laporan ditemukan berhasil	Valid
	_	klaim laporan	menampilkan notifikasi klaim laporan ditemukan	notifikasi klaim laporan ditemukan	Valid
					Valid

VAL_0 12	Kasus uji validasi lupa password	Sistem dapat mengirim ulang password kepada email yang valid	Sistem berhasil mengirim ulang password kepada email yang valid	Valid
VAL_0 13	Kasus uji validasi blokir kendaraan	Sistem dapat mengubah status kendaraan menjadi blokir	Sistem mengubah status kendaraan menjadi blokir	Valid
VAL_0 14	Kasus uji validasi cabut blokir kendaraan	Sistem dapat mengubah status kendaraan menjadi aktif	Sistem mengubah status kendaran menjadi aktif	Valid
VAL_0 15	Kasus uji validasi memproses laporan kehilangan	Sistem dapat menampilkan notifikasi laporan kehilangan berhasil diproses	Sistem menampilkan notifikasi laporan kehilangan berhasil diproses	Valid
VAL_0 16	Kasus uji validasi menginput kendaraan keluar kampus	True: Sistem dapat menampilkan notifikasi kendaraan berhasil keluar kampus False: Sistem dapat menampilkan notifikasi kendaraan diblokir	True : Sistem menampilkan notifikasi kendaraan berhasil keluar kampus False : Sistem menampilkan notifikasi kendaraan diblokir	Valid
VAL_0 17	Kasus uji validasi mengelola data pengendara	Sistem dapat membuat, melihat, mengubah, dan menghapus data pengendara	Sistem dapat membuat, melihat, mengubah, dan menghapus data pengendara	Valid
VAL_0 18	Kasus uji validasi mengelola data kendaraan	Sistem dapat membuat, melihat, mengubah, dan menghapus data kendaraan	Sistem dapat membuat, melihat, mengubah, dan menghapus data kendaraan	Valid
VAL_0 19	Kasus uji validasi mengelola data petugas parkir	Sistem dapat membuat, melihat, mengubah, dan menghapus data petugas parkir	Sistem dapat membuat, melihat, mengubah, dan menghapus data petugas parkir	Valid
VAL_0 20	Kasus uji validasi memvalidasi	Sistem dapat menampilkan	Sistem dapat menampilkan	Valid

	laporan kehilangan	notifikasi laporan kehilangan berhasil divalidasi	notifikasi laporan kehilangan berhasil divalidasi	
VAL_0 21	Kasus uji validasi mengelola data area parkir	Sistem dapat membuat, melihat, mengubah, dan menghapus data area parkir	Sistem berhasil membuat, melihat, mengubah, dan menghapus data area parkir	Valid
VAL_0 22	Kasus uji validasi memvalidasi laporan ditemukan	Sistem dapat menampilkan notifikasi laporan ditemukan berhasil diklaim	Sistem dapat menampilkan notifikasi laporan ditemukan berhasil diklaim	Valid

#### 6.2.1.4 Analisis

Berdasarkan hasil pengujian validasi pada tabel 6.26, seluruh kasus uji yang didefinisikan telah valid atau berarti seluruh spesifikasi kebutuhan perangkat lunak telah terpenuhi. Sehingga dapat dikatakan sistem pengelolaan parkir menggunakan perangkat NFC berbasis android berhasil lolos uji validasi.

## 6.2.2 Reliability

#### 6.2.2.1 Tujuan

Tujuan dilakukannya pengujian pada karakteristik *reliability* adalah untuk memastikan sistem telah memenuhi kebutuhan non fungsional pada parameter *reliability*.

#### 6.2.2.2 Mekanisme

Karakteristik reliability pada sistem akan dinilai dengan melakukan pengujian terhadap request yang dilakukan oleh user dalam selang waktu tertentu yang dapat diterima oleh sistem. Dalam pengujian ini terdapat skenario pengujian yaitu jumlah user yang diujikan berjumlah 39 user berdasarkan 39 tempat parkir yang tersedia di UB dengan selang waktu 1 menit. Kriteria yang diuji pada karakteristik reliability dengan WAPT yaitu sessions, pages, dan hits.

Karakteristik reliability pada sistem akan dinilai dengan melakukan pengujian terhadap kriteria sessions, pages, dan hits. Sessions merupakan skenario uji pengguna dari melakukan request hingga mendapatkan respons dari sistem, sedangkan pages merupakan halaman yang sukses diunduh, sedangkan hits merupakan request yang sukses diterima sistem.

Pengujian dilakukan dengan memanfaatkan tools wapt (web application performance tools). Setiap user tersebut melakukan request terhadap tiga fitur pada perangkat lunak yang sedang diujikan yaitu login, menginput kendaaraan keluar-masuk area parkir.

## 6.2.2.3 Hasil Pengujian

Hasil pengujian *reliability* dapat dilihat secara lengkap pada lampiran. Tabel 6.27 menjelaskan ringkasan hasil pengujian *reliability*.

Tabel 6.27 Ringkasan Hasil Pengujian Reliability

Kriteria	Berhasil	Gagal	Total
Sessions	353	0	353
Pages	2951	0	2951
Hits	8911	0	8911

Berdasarkan ringkasan hasil pengujian reliability pada Tabel 6.27, seluruh kriteria yang diujikan tidak memiliki kegagalan. Kemudian dapat dimasukkan ke persamaan untuk pencarian nilai reliability.

a. Sessions

$$X = \frac{353}{353} \times 100\% = 100\%$$

b. Pages

$$X = \frac{2951}{2951} \times 100\% = 100\%$$

c. Hits

$$X = \frac{8911}{8911} \times 100\% = 100\%$$

Dari hasil tersebut didapatkan ketiga kriteria yang diujikan yaitu sessions, pages, hits tersebut memiliki nilai 100%.

#### **6.2.2.4** Analisis

Berdasarkan hasil pengujian pada karakteristik *reliability*, didapatkan nilai dari request kepada sistem berdasarkan tiga kriteria pengujian *reliability* sebesar 100%. Dan untuk memenuhi standar *reliability*, diperlukan nilai *reliability* lebih dari sama dengan standar Telcordia yaitu 95% dari kasus uji harus berhasil untuk mendapatkan nilai *reliability* dari sistem. Dengan kata lain, sistem telah memenuhi standar untuk karakteristik *reliability*.

## 6.2.3 Compatibility

#### 6.2.3.1 Tujuan

Tujuan dilakukannya pengujian pada karakteristik compatibility adalah untuk memastikan sistem telah memenuhi kebutuhan non fungsional pada parameter compatibility.

#### 6.2.3.2 Mekanisme

Karakteristik compatibility pada sistem akan dinilai dengan melakukan pengujian pada kompatibilitas antar browser. Pengujian pada karakteristik ini menggunakan tool BrowseEmAll. Sistem diuji menggunakan delapan desktop browser secara virtual dengan BrowseEmAll. Browser yang digunakan adalah

mozilla firefox 37, google chrome 42, internet explorer 8-11, safari 8 dan opera 29. Setiap antarmuka sistem akan diujikan melalui tools tersebut. Jika hasil dari BrowseEmAll tidak menunjukkan adanya error dan warning maka sistem memenuhi karakteristik compatibility. Namun jika terdapat error ataupun warning maka file view harus diperbaiki.

## 6.2.3.3 Hasil Pengujian

Hasil pengujian *compatibility* dapat dilihat secara lengkap pada lampiran. Tabel 6.28 menjelaskan ringkasan hasil pengujian *compatibility*.

Tabel 6.28 Ringkasan Hasil Pengujian Compatibility

URL	Status
http://localhost/parkiran/parkiran/tambah	Valid
http://localhost/parkiran/kendaraan	Valid
http://localhost/parkiran/kehilangan/kehilangan4	Valid
http://localhost/parkiran/parkiran	Valid
http://localhost/parkiran/pengendara	Valid
http://localhost/parkiran/login	Valid
http://localhost/parkiran/kehilangan/tambah	Valid
http://localhost/parkiran/kendaraan/proses_cari	Valid
http://localhost/parkiran/pengendara/tambah	Valid
http://localhost/parkiran/dashboard	Valid

Berdasarkan ringkasan hasil pengujian *compatibility* pada Tabel 6.23, seluruh halaman sistem tidak ditemukan adanya *error* maupun *warning*.

#### **6.2.3.4** Analisis

Berdasarkan hasil pengujian pada karakteristik compatibility, sistem tidak memiliki error dan warning di delapan desktop browser. Dan untuk dapat memenuhi standar compatibility, diperlukan sistem yang tidak memiliki error dan warning jika dioperasikan di berbagai browser. Pada pengujian ini tidak ditemukan error dan warning serta permasalahan-permasalahan yang terjadi pada compactibilit cross browser sehingga dari hasil pengujian tersebut maka sistem telah memenuhi standar untuk karakteristik compatibility.

## 6.3 Pengujian Performa

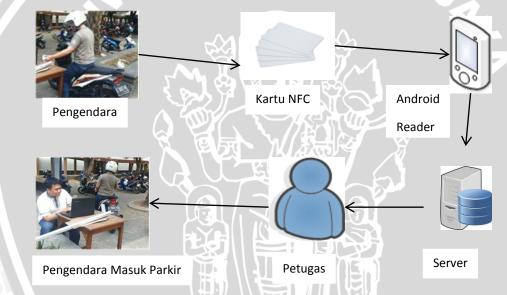
## 6.3.1.1 Tujuan

Tujuan dilakukannya pengujian performa adalah untuk mengukur performa sistem perangkat lunak yang telah dibuat untuk mendapatkan hasil sesuai dengan yang diharapkan.

#### 6.3.1.2 Mekanisme

Pengujian performa dilakukan dengan melakukan uji coba sistem pada lapangan. Pengujian dilakukan dengan melakukan sesuai alur dari perangkat lunak yang telah dibuat. Dalam pengujian akan diukur berapa waktu yang dibutuhkan dalam aktivitas masuk area parkir, keluar area parkir, dan keluar area kampus. Pengujian dilakukan dengan melibatkan 30 pengendara dalam ketiga alur pengujian tersebut.

Pada alur masuk area parkir, pengendara membawa kartu nfc dan mendekatkan pada android reader. Setelah uid muncul maka pengendara memilih menu masuk area parkir. Petugas parkir kemudian mengecek inputan dari pengendara dengan memilih menu input kendaraan masuk parkir. Petugas parkir kemudian memilih data kendaraan dari inputan pengendara kemudian tekan proses. Setelah muncul notifikasi kendaraan berhasil masuk area parkir maka petugas parkir mempersilahkan pengendara untuk masuk area parkir. Gambar 6.7 menjelaskan alur masuk area parkir.



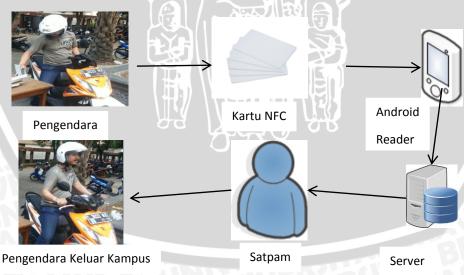
#### Gambar 6.7 Alur Masuk Area Parkir

Pada alur keluar area parkir, pengendara membawa kartu nfc dan mendekatkan pada android reader. Setelah uid muncul maka pengendara memilih menu keluar area parkir. Petugas parkir kemudian mengecek inputan dari pengendara dengan memilih menu input kendaraan keluar parkir. Petugas parkir kemudian memilih data kendaraan dari inputan pengendara kemudian tekan proses. Muncul halaman detail parkir kendaraan lalu petugas membandingkan data pada sistem dengan kendaraan yang digunakan oleh pengendara. Bila cocok maka petugas parkir memilih tombol submit. Setelah muncul notifikasi kendaraan berhasil keluar area parkir maka petugas parkir mempersilahkan pengendara untuk keluar area parkir. Gambar 6.8 menjelaskan alur keluar area parkir.



## **Gambar 6.8 Alur Keluar Area Parkir**

Pada alur keluar area kampus, pengendara membawa kartu nfc dan mendekatkan pada android reader. Setelah uid muncul maka pengendara memilih menu keluar area kampus. Petugas keamanan kemudian mengecek inputan dari pengendara dengan memilih menu input kendaraan keluar kampus. Petugas keamanan kemudian memilih data kendaraan dari inputan pengendara kemudian tekan proses. Muncul halaman detail parkir kendaraan lalu petugas membandingkan data pada sistem dengan kendaraan yang digunakan oleh pengendara. Bila cocok maka petugas keamanan memilih tombol submit. Setelah muncul notifikasi kendaraan berhasil keluar area kampus maka petugas keamanan mempersilahkan pengendara untuk keluar area kampus. Gambar 6.9 menjelaskan alur keluar area kampus.



**Gambar 6.9 Alur Keluar Area Kampus** 

## 6.3.1.3 Hasil Pengujian

Berdasarkan tiga alur pengujian yang telah dijelaskan, didapatkan dua hasil pengeujian yaitu pengujian performa alur keluar-masuk area parkir serta keluar area kampus dan pengujian performa waktu dari alur keluar-masuk area parkir serta keluar area kampus. Pada tabel 6.29 hingga 6.31 akan menampilkan alur keluar-masuk area parkir dan area kampus, pada tabel 6.32 akan menampilkan hasil pengujian performa waktu secara keseluruhan, dan pada tabel 6.33 hingga 6.35 akan menampilkan hasil pengujian performa waktu secara detail.

**Tabel 6.29 Alur Masuk Area Parkir** 

No	Alur
1	Android reader membaca kartu nfc pengendara
2	Android reader menampilkan uid pengendara
3	Pengendara menginputkan menu masuk area parkir
4	Inputan tersimpan dalam sistem
5	Sistem menampilkan inputan dari pengendara
6	Petugas parkir memproses inputan dari pengendara
7	Muncul notifikasi kendaraan berhasil masuk area parkir

**Tabel 6.30 Alur Keluar Area Parkir** 

No	Alur S S S S S S S S S S S S S S S S S S S
1	Android reader membaca kartu nfc pengendara
2	Android reader menampilkan uid pengendara
3	Pengendara menginputkan menu keluar area parkir
4	Inputan tersimpan dalam sistem
5	Sistem menampilkan inputan dari pengendara
6	Petugas parkir memproses inputan dari pengendara
7	Muncul detail parkir pengendara
8	Muncul notifikasi kendaraan berhasil keluar area parkir

**Tabel 6.31 Alur Keluar Area Kampus** 

No	Alur						
1	Android reader membaca kartu nfc pengendara						
2	Android reader menampilkan uid pengendara						
3	Pengendara menginputkan menu keluar area kampus						
4	Inputan tersimpan dalam sistem						
5	Sistem menampilkan inputan dari pengendara						
6	Petugas keamanan memproses inputan dari pengendara						

7	Muncul detail parkir pengendara
8	Muncul notifikasi kendaraan berhasil keluar area kampus

Tabel 6.32 Ringkasan Hasil Pengujian Waktu (dalam detik)

Pengendara	Masuk Area Parkir	Keluar Area Parkir	Keluar Area Kampus
Pengendara 1	10	15	16
Pengendara 2	8	18	12
Pengendara 3	11	10	12
Pengendara 4	12	14	17
Pengendara 5	15	18	15
Pengendara 6	9	16	20
Pengendara 7	8	18	19
Pengendara 8	10	16	19
Pengendara 9	12	18	17
Pengendara 10	11	18	20
Pengendara 11	9	15	18
Pengendara 12	12	19	20
Pengendara 13	107	14	16
Pengendara 14	14	14	17
Pengendara 15	13	15	16
Pengendara 16	11	15	15
Pengendara 17	10	17	18
Pengendara 18	11	20	17
Pengendara 19	14	15	18
Pengendara 20	12	15	15
Pengendara 21	14	16	15
Pengendara 22	10	14	16
Pengendara 23	8	10	12
Pengendara 24	12	18	15
Pengendara 25	10	16	17
Pengendara 26	10	12	16
Pengendara 27	12	15	15
Pengendara 28	14	16	17
Pengendara 29	11	18	16

Pengendara 30	11	14	15
Rata-rata	11.133	15.633	16.366

Tabel 6.33 Ringkasan Hasil Pengujian Waktu Masuk Area Parkir (dalam detik)

Pengendara	Alur 1	Alur 2	Alur 3	Alur 4	Alur 5	Alur 6	Alur 7	Total
Pengendara 1	3	1	2	1	1	1	1	10
Pengendara 2	2	1	1	1	1	1	1	8
Pengendara 3	3	1	3	1	1	1	1	11
Pengendara 4	4	1	3	1	1	1	1	12
Pengendara 5	6	1	4	1	1	1	1	15
Pengendara 6	2	1	2	1	1	1	1	9
Pengendara 7	2	1	1	1	1	1	1	8
Pengendara 8	3	1	2	1	1	1	1	10
Pengendara 9	3	1	4	1	1	1	1	12
Pengendara 10	3	1	3	1	1	1	1	11
Pengendara 11	2	1	2	1	1	1	1	9
Pengendara 12	4	1	3	1	15	1	1	12
Pengendara 13	3	1	2	1	1	1	1	10
Pengendara 14	6	1	3	1	1	1	1	14
Pengendara 15	5	1	3	1	1	1	1	13
Pengendara 16	4 8	1	2	1	1	1	1	11
Pengendara 17	3	1	2	1	1	1	1	10
Pengendara 18	4	1	2	1	1	1	1	11
Pengendara 19	6	1	3	1	1	1	1	14
Pengendara 20	4	1	3	1	1	1	1	12
Pengendara 21	6	1	3-7	1	1	1	1	14
Pengendara 22	3	1	2	1	1	1	1	10
Pengendara 23	2	1	1	1	1	1	1	8
Pengendara 24	4	1	3	1	1	1	1	12
Pengendara 25	3	1	2	1	1	1	1	10
Pengendara 26	3	1	2	1	1	1	1	10
Pengendara 27	4	1	3	1	1	1	1	12
Pengendara 28	7	1	2	1	1	1	1	14
Pengendara 29	4	1	2	1	1	1	1	11
Pengendara 30	3	1	3	1	1	1	1	11

Tabel 6.34 Ringkasan Hasil Pengujian Waktu Keluar Area Parkir (dalam detik)

Pengendara	Alur 1	Alur 2	Alur 3	Alur 4	Alur 5	Alur 6	Alur 7	Alur 8	Total
Pengendara 1	5	1	3	1	1	1	2	1	15
Pengendara 2	6	1	4	1	1	1	3	1	18
Pengendara 3	2	1	1	1	1	1	2	1	10
Pengendara 4	3	1	3	1	1	1	3	1	14
Pengendara 5	6	1	4	1	1	1	3	1	18
Pengendara 6	5	1	3	1	1	1	3	1	16
Pengendara 7	6	1	4	1	1	1	3	1	18
Pengendara 8	5	1	3	1	1	11	3	1	16
Pengendara 9	6	1	3	1	1	1	4	1	18
Pengendara 10	6	1	3	1	1	1	4	1	18
Pengendara 11	5	1	3	1	1	1	2	1	15
Pengendara 12	6	1	4	1	1	1	4	1	19
Pengendara 13	4	1	2	1	1	1	3	1	14
Pengendara 14	3	1	3	1	1	1	3	1	14
Pengendara 15	4 (2)	1	3	1	1	1	3	1	15
Pengendara 16	4	1	3	1	1	1	3	1	15
Pengendara 17	6	1	3	1	1	1	3	1	17
Pengendara 18	8	1	4	1	1	1	3	1	20
Pengendara 19	4	1	3 (4)	1 /)	1	1	3	1	15
Pengendara 20	5	134	3	1	1	1	2	1	15
Pengendara 21	5	1	3	1	1	1	3	1	16
Pengendara 22	4	1	2	1	1	1	3	1	14
Pengendara 23	2	1	1	10	1	1	2	1	10
Pengendara 24	6	1	4	1	1	1	3	1	18
Pengendara 25	5	1	3	1	1	1	3	1	16
Pengendara 26	3	1	2	1	1	1	2	1	12
Pengendara 27	4	1	3	1	1	1	3	1	15
Pengendara 28	5	1	3	1	1	1	3	1	16
Pengendara 29	7	1	3	1	1	1	3	1	18
Pengendara 30	3	1	3	1	1	1	3	1	14

Tabel 6.35 Ringkasan Hasil Pengujian Waktu Keluar Area Kampus (dalam detik)

Pengendara	Alur 1	Alur 2	Alur 3	Alur 4	Alur 5	Alur 6	Alur 7	Alur 8	Total
Pengendara 1	5	1	3	1	1	1	3	1	16
Pengendara 2	3	1	2	1	1	1	2	1	12
Pengendara 3	3	1	2	1	1	1	2	1	12
Pengendara 4	6	1	3	1	1	1	3	1	17
Pengendara 5	5	1	2	1	1	1	3	1	15
Pengendara 6	8	1	3	1	1	1	4	1	20
Pengendara 7	7	1	3	1	1	1	4	1	19
Pengendara 8	6	1	4	1	1	1	4	1	19
Pengendara 9	5	1	3	1	1	1	4	1	17
Pengendara 10	7	1	4	1	1	1	4	1	20
Pengendara 11	6	1	3	1	1	1	4	1	18
Pengendara 12	7	1	4	1	1	1	4	1	20
Pengendara 13	5	1	2	1	1	1	4	1	16
Pengendara 14	5	1	3	1/5	1	11	4	1	17
Pengendara 15	4	1	3	1	1	1	4	1	16
Pengendara 16	4	1	3	1	1	1 (	3	1	15
Pengendara 17	6	1	3	1	1	1	4	1	18
Pengendara 18	5	1	3	1	1	1	4	1	17
Pengendara 19	6	1	3	1	1	1	4	1	18
Pengendara 20	4	1)	3	1	1	1	3	1	15
Pengendara 21	4	1	3	1	1	1	3	1	15
Pengendara 22	4	1	3	1	1	1	4	1	16
Pengendara 23	2	1	2	1	1	1	3	1	12
Pengendara 24	3	1	3	1	1	1	4	1	15
Pengendara 25	5	1	3	1	1	1	4	1	17
Pengendara 26	5	1	2	1	1	1	4	1	16
Pengendara 27	4	1	3	1	1	1	3	1	15
Pengendara 28	5	1	3	1	1	1	4	1	17
Pengendara 29	4	1	3	1	1	1	4	1	16
Pengendara 30	3	1	3	1	1	1	4	1	15

Tabel 6.36 Ringkasan Hasil Pengamatan Waktu Keluar Area Parkir dan Area Kampus (dalam detik)

	am detik	
Pengendara	Keluar Area Parkir	Keluar Area Kampus
Pengendara 1	3	7
Pengendara 2	3	4
Pengendara 3	3	3
Pengendara 4	4	5
Pengendara 5	5	3
Pengendara 6	6	6
Pengendara 7	3	4
Pengendara 8	6	4
Pengendara 9	5	3
Pengendara 10	4	5
Pengendara 11	3	6
Pengendara 12	4	3.^-
Pengendara 13	3	4
Pengendara 14	5	3
Pengendara 15	4	5
Pengendara 16	4	5
Pengendara 17	6	3
Pengendara 18	5	4
Pengendara 19	6	3
Pengendara 20	4	5
Pengendara 21	4	6
Pengendara 22	4///	600
Pengendara 23	3	7
Pengendara 24	3	4
Pengendara 25	5	3
Pengendara 26	5	4
Pengendara 27	4	5
Pengendara 28	5	3
Pengendara 29	4	3
Pengendara 30	3	6

#### 6.3.1.4 Analisis

Berdasarkan pengujian alur keluar-masuk area parkir dan keluar area kampus, alur-alur yang dijalankan oleh pengguna sistem baik oleh pengendara maupun oleh petugas parkir dan petugas keamanan tidak mengalami kendala. Sistem dapat merespon dengan baik aktivitas yang dilakukan oleh pengguna, hal ini ditandai dengan hasil valid pada setiap alur yang dijalankan.

Berdasarkan pengujian performa waktu, sistem yang ada saat ini pengendara membutuhkan waktu rata-rata 4,2 detik untuk keluar area parkir dan 4,4 detik untuk keluar area kampus, bila menggunakan aplikasi NFCPS (NFC Parking System) pengendara membutuhkan waktu rata-rata 15,633 detik untuk keluar area parkir dan 16,366 detik untuk keluar area kampus.

Dari hasil pengujian yang didapatkan dapat dikatakan aplikasi NFCPS (NFC Parking System) lebih lambat menjalankan proses bisnis parkir bila dibandingkan dengan proses bisnis parkir pada saat ini. Hal ini dikarenakan beberapa gangguan yang terjadi seperti pembacaan kartu nfc oleh android reader terkadang mengalami gangguan akibat mendekatkan kartu tidak pas dengan sensor reader, pengecekan data kendaraan dengan membandingkan dengan kendaraan yang dibawa diperlukan ketelitian sehingga menyita waktu lebih banyak. Namun dari aspek keamanan sistem dapat berjalan dengan baik karena pengendara yang memiliki kartu nfc yang terdaftar pada sistem dapat menjalankan aktivitas keluar-masuk area parkir dan keluar area kampus, serta petugas yang telah diberi wewenang yang dapat mengakses sistem.

Berdasarkan pengujian yang telah dilakukan, dapat dikatakan sistem telah berjalan dengan baik dan sesuai dengan hasil yang diharapkan. Sistem tidak mengalami gangguan yang berarti, dan sudah mengakomodir kebutuhan parkir sesuai dengan kebutuhan organisasi.

## **BAB 7 PENUTUP**

## 7.1 Kesimpulan

Berdasarkan hasil penelitian skripsi yang telah dilakukan maka dapat ditarik kesimpulan sebagai berikut:

- 1. Perancangan sistem pengelolaan parkir menggunakan perangkat NFC berbasis android telah menghasilkan desain dari proses bisnis pada alur keluar-masuk area parkir dan keluar area kampus.
- Implementasi desain dari proses bisnis pada sistem yang telah disesuaikan dengan kebutuhan organisasi ini telah menghasilkan fitur masuk area parkir, keluar area parkir, dan keluar area kampus dengan menggunakan kartu NFC sebagai identitas pengendara.
- 3. Berdasarkan hasil pengujian, sistem dapat dikatakan memenuhi kebutuhan organisasi dalam hal proses bisnis parkir. Hal ini ditunjukkan pada pengujian white-box, black-box, dan performa yang telah dilakukan telah dilewati dengan baik dan mencapai hasil 100% valid.
- 4. Penggunaan kartu NFC dapat meningkatkan aspek keamanan karena hanya kartu NFC yang terdaftar pada sistem yang dapat digunakan.
- 5. Berdasarkan pengujian performa, proses keluar-masuk area parkir dan area kampus lebih lambat bila dibandingkan dengan proses terdahulu.

## 7.2 Saran

Saran yang dapat diberikan setelah menyelesaikan penelitian skripsi ini yaitu:

- Untuk pengembangan lebih lanjut dapat dikembangkan dengan menggunakan perangkat nfc reader.
- Untuk pengembangan lebih lanjut dapat dikembangkan pada lebih dari satu area parkir agar dapat terintegrasi antara seluruh area parkir pada kampus UB.
- 3. Untuk pengembangan lebih lanjut dapat dikembangkan fitur cek status kendaraan dan cek kapasitas area parkir.
- Untuk pengembangan lebih lanjut dapat dioptimalkan proses keluar-masuk area parkir agar mendapatkan waktu lebih sedikit tanpa mengurangi aspek keamanan.

## **DAFTAR PUSTAKA**

- Abeysinghe, S. (2009). *PHP Team Development: Easy and Effective Team Work using MVC, Agile Development, Source Control, Testing, Bug Tracking, and More*. Birmingham: PACKT Publishing.
- Abodeqa. (2012, 11 09). Browser Compatibility testing- When, Where and how it is done. Retrieved 11 18, 2015, from abodeqa.com: http://www.abodeqa.com/2012/11/09/browser-compatibility-testing-when-where-and-how-it-is-done/
- Al-Safadi, L. A., & Garcia, R. A. (2012). ISO9126 Based Quality Model for Evaluating B2C e-Commerce Applications - A Saudi Market Perspective. IJCIT, 3(2), 8-15.
- Ashimima. (n.d.). Kegunaan NFC pada Smartphone untuk Apa. Retrieved Maret 20, 2015, from http://ashimima.com/kegunaan-nfc-pada-smartphone-untuk-apa/
- Asthana, A., & Olivieri, J. (2009). Quantifying Software Reliability and Readiness. (pp. 1-6). Naples: IEEE.
- Atmodjo, M. W. (2005). *Restoran dan Segala Permasalahannya*. Yogyakarta: ANDI.
- Broke, J. (1996). SUS A Quick and Dirty Usability Scale. *Usability Evalution in Industry, 189*(194), 4-7.
- Cha, B., & Kim, J. (2013). Design of NFC Based Micro-payment to Support MD Authentication and Privacy fot Trade Safety in NFC Applications.
- Coleman, D., Ash, D., Lowther, B., & Oman, P. (1994). Using Metrics to Evaluate Software System Maintainability. *Computer, 27*(8), 44-49.
- Darat, D. J. (1996). *Pedoman Teknis Penyelenggaraan Fasilitas Parkir*. Jakarta: Departemen Perhubungan.
- Display, L. (2015, April). Penerapan Sistem Kartu Parkir, Sudahkah Menyelesaikan Masalah Keamanan?
- Essbach, J. (2012). An Approach to a Decentral Mobile Payment System using NFC and the German Eid-Card.
- Fahmy, S., Haslinda, N., Roslina, W., & Fariha, Z. (2012). Evaluating the Quality of Software in e-Book Using the ISO 9126 Model. *International Journal of Control and Automation*, 5(2), 115-122.
- Famoco. (2015). *RFID vc NFC: What's the difference?* Retrieved November 13, 2015, from famoco.com: http://www.famoco.com/rfid-vc-nfc-whats-the-difference/

- Fuati, L. N. (2014). Analisis Biaya dan Manfaat pada Perencanaan Sistem Keluar-Masuk Kendaraan dengan menggunakan barcode di Universitas Brawijaya
- Gargenta, M. (2011). Learning Android. Sebastopol: O'Reilly Media.
- Hobbs, F. D. (1995). *Penyelenggaraan dan Teknik Lalu Lintas Edisi Kedua*. Yogyakarta: Gadjah Mada University Press.
- International Standards Office. (2000). ISO/IEC FDIS 9126-1: Quality Model. Geneva: International Organization for Standardization / International Electrotechnical Commission.
- International Standards Office. (2002). ISO/IEC TR 9126-2: External Metrics. Geneva: International Organization for Standardization / International Electrotechnical Commission.
- Jamwal, D. (2010). Analysis of Software Quality Models for Organizations. International Journal of Latest Trends in Computing, 1(2), 19-23.
- Jauhari. (2015, April). Wawancara dengan Kepala Divisi Parkir UB. (H. Akbar, Interviewer)
- Juntunen, A., Luukkainen, S., & Tuunainen, V. (2010). Deploying NFC Technology for Mobile Ticketing Services.
- Kompasiana, T. (2012, Maret 19). *Tren Baru Transaksi dengan NFC*. Retrieved Maret 20, 2015, from teknologi.kompasiana.com: http://teknologi.kompasiana.com/gadget/2012/03/19/tren-baru-transaksi-dengan-nfc-443346.html
- Kristiani, D. (2013, Mei 31). Fungsi NFC pada Smartphone. Retrieved Januari 12, 2015, from teknologi.kompasiana.com: http://teknologi.kompasiana.com/gadget/2013/05/31/fungsi-nfc-pada-smartphone-560842.html
- Kurniawan, F. (2010). Sistem Informasi Pelayanan Parkir Yang Dilengkapi Dengan Kamera. *PENS-ITS*.
- Laube, J. (2015). How to Turn Your Good Restaurant into a Great Business.

  Retrieved Maret 10, 2015, from http://www.restaurantowner.com/public/How-to-Turn-Your-Good-Restauran-tinto-a-Great-Business.cfm
- Lee, H., Hong, W., Kao, C., & Cheng, C. (2014). A User-friendly Authentication Solution using NFC Card Emulation on Android.
- Mazzocchi, A. (2014). OPEN-NPP: an open source library to enable P2P over NFC.
- Ngai, E., Suk, F., & Lo, S. (2008). Development of an RFID-based Sushi Management System. *International Journal of Production Economics*, 112(2), 630-645.
- Nielsen, J. (1993). Usability Engineering. San Diego: Academic Press.

- Opperman, C., & Haneke, G. (2011). Using NFC-enabled Phones for Remote Data Acquistion and Digital Control.
- Orr, E., & Zadik, Y. (2013). *Programming with Codelgniter MVC: Build Feature-rich Web Applications using The Codelgniter MVC Framework*. Birmingham: PACKT Publishing.
- Perdananugraha, G. M. (2010). Rancang Bangun dan Implementasi Sistem Informasi Terintegrasi (Integrated Information System) E-Government Berbasis Open Source. *Teknologi Indonesia*, 33(1), 45-59.
- Pieskä, S., Liuska, M., Jauhiainen, J., & Auno, A. (2013). Intelligent Restaurant System Smartmenu. Budapest: IEEE.
- Presmann, R. S. (2001). *Software Engineering, A Practitioner's Approach Fifth Edition*. New York: McGraw-Hill.
- Rina, & Tyagi, S. (2013). A Comparative Study of Performance Testing Tools. International Journal of Advanced Research in Computer Science and Software Engineering, 3(5), 1300-1307.
- Salonen, V. (2012). *Automatic Portability Testing. Master's Thesis.* Jyväskylä: University Jyväskylä.
- Saparkhojayev, N., Dauitbayeva, A., Nurtayev, A., & Baimenshina., G. (2014). NFC-enabled Access Control and Management System. 1.
- Saputra, S. T., & Harjanti, D. (2013). Pengelolaan dan Pengembangan Usaha pada Restoran Pondok Kemangi di Banjarmasin. *AGORA*, 1(1).
- Septiana, F. T. (2013). Sistem perangkat lunak untuk simulasi keluar masuk dan pembayaran parkir kendaraan berbasis RFID. *Repositori Jurnal Mahasiswa PTIIK UB*.
- Setialana, P. (2014). Pengembangan dan Analisis Kualitas Sistem Informasi Jadwal Akademik Berbasis YII Framework. *Jurnal Pendidikan Teknik Informatika*, 3(7), 1-10.
- Setiawan, D. (2007). Sistem Pengelolaan Parkir dengan Fasilitas Penentuan Lokasi Parkir Terdekat dengan Pintu Keluar Masuk. *ITS*.
- Shimmura, T., Takemaka, T., & Akamatsu, M. (2009). Real-Time Process Management System in a Restaurant by Sharing Food Order Information. (pp. 703-706). Malacca: IEEE.
- Shodiq, F. (2015). Perancangan dan Implementasi Sistem Rekomendasi Penginformasian Tempat Parkir yang Kosong pada Studi Kasus Lahan Parkir Universitas Brawijaya.
- Sigala, M. (2003). Unravelling The Impact of Information and Communication Technologies (ICT) on Restaurant Productivity. *ECIS 2002 Proceedings*, 164.

- Sommerville, I. (2007). *Software Engineering: Eighth Edition*. New York: Addison-Wesley.
- Sukamto, R. A., & Shalahuddin, M. (2013). *Rekayasa Perangkat Lunak: Terstruktur dan Berorientasi Objek.* Bandung: Informatika.
- Taiwar, G. (2013, Oktober 16). *Integration Testing Tips*. Retrieved Maret 23, 2015, from http://university.utest.com/integration-testing-tips/
- Tan, T.-H., & Chang, C.-S. (2010). Development and Evaluation of an RFID-based e-Restaurant System for Customer-centric Service. *Expert Systems with Applications*, *37*(9), 6482-6492.
- Taufik, R. (2014, November 23). *Kelebihan dan Kekurangan QR Code*. Retrieved November 19, 2015, from teukutaufik.com: http://www.teukutaufik.com/2014/11/kelebihan-dan-kekurangan-qr-code.html
- Travo. (2011, Mei 12). *Teknologi QR Code dan Cara Kerjanya*. Retrieved November 13, 2015, from audiovideo.com: http://www.audiovideo.co.id/index.php?option=com\_content&view = article&id=158:terknologi-qr-code-dan-carakerjanya&catid=38:tekno&ltemid=67
- Tullis, T. S., & Stetson, J. N. (2004). A Comparison of Questionnaire for Assessing Website Usability. *Usability Proffessional Association Conference*, 1-12.
- Wahono, R. S. (2006). *Teknik Pengukuran Kualitas Perangkat Lunak*. Retrieved Maret 5, 2015, from http://romisatriawahono.net/2006/06/05/teknik-pengukuran-kualitas-perangkat-lunak/
- Wu, S.-H., & Yang, C. (2013). Promoting Collaborative mobile payment by using NFC-Micro SD technology. 455.
- Zufti, F. (2014). Pengembangan prototipe sistem informasi perparkiran sepeda motor di universitas brawijaya.

# LAMPIRAN A HASIL WAWANCARA KOORDINATOR DIVISI PARKIR UB

Nama: Jauhari

Jabatan : Koordinator Divisi Parkir

## Pertanyaan

- 1. Area parkiran di ub ada berapa?
- 48 area parkir dengan 24 area parkir untuk pengguna roda 2, 9 area parkir untuk pengguna roda 4, dan 15 area parkir pengguna untuk roda 2 dan roda 4.
- 2. Masing-masing parkiran berapa personel yang jaga?
- Tiap area parkir dijaga oleh 2 personel
- 3. Tiap personel shift apa full?
- Dijalankan dengan 2 shift dengan shift pertama anatara jam 07.00-14.00 dan shift kedua antara jam 14.00-21.00
- 4. Berapa petugas parkir yang dimiliki ub?
- 150 orang
- 5. Sticker ub apakah masih berlaku?
- Tidak berlaku
- 6. Karcis ub apakah masih berlaku?
- Tidak berlaku
- 7. Berapa kendaraan yang masuk parkiran dalam sehari?
- Sekitar 18110 kendaraan roda 2 dan roda 4
- 8. Jam berapa saja saat parkiran ramai?
- Jam 09.00 hingga jam 17.00
- 9. Jam berapa saja saat parkiran sepi?
- Jam 07.00 hingga jam 09.00 dan jam 17.00 hingga jam 21.00
- 10. Apakah ada curanmor di parkiran?
- Ada
- 11. Bagaimana modus curanmor di parkiran?
- Mengganti plat nomor kendaraan yang sudah diincar kemudian keluar dengan menunjukkan stnk palsu sesuai dengan plat nomor kendaraan yang diganti
- 12. Bagaimana proses parkir yang ada di ub?
- Untuk masuk area parkir petugas parkir memberikan kartu parkir, untuk keluar area parkir pengendara memberikan kartu parkir dan

menunjukkan stnk, dan untuk keluar area kampus pengendara menunjukkan stnk kepada satpam

- 13. Apakah proses tersebut sudah optimal?
- Belum optimal, dikarenakan masih ada pencurian kendaraan bermotor dalam area UB
- 14. Apa permasalahan yang ada dalam sistem tersebut?
- Jumlah petugas tidak ideal bila dibandingkan dengan luas area parkir, jumlah kartu parkir yang tersedia tidak mencukupi kendaraan yang menggunakan area parkir, masih adanya pencurian kendaraan bermotor, tidak ada pencatatan aktivitas keluar-masuk area parkir
- 15. Apakah ada solusi dari permasalahan tersebut?
- Mendata kapasitas area parkir, mengusulkan penambahan personel untuk memenuhi kebutuhan petugas parkir
- 16. Apa harapan anda mengenai sistem parkir ub?
- Semoga bisa lebih baik dan dapat mengatasi permasalahan yang ada



# **LAMPIRAN B HASIL PENGUJIAN RELIABILITY**

lame									Resu	lt		Comment
ession	error rate for each prof	ile							SUC	CESS		
	or rate for each page								SUC			
	r rate for each page								SUC			
	response time for each	page							SUC			
		,,,,,										
ummar	v											
rofile	Successful sessions	Failed sessions	Successful pages	Failed pages	Successful hits	Failed hits	Other errors	Total F	(Bytes sent	Total KBytes received	Avg re	sponse time,
rofile4	41	0	364	0	1296	0	0	977		36301	0.03(0	.04)
ass/F	ail Criteria											
lame									R	esult		Comn
Session	error rate for each pro	ofile							S	UCCESS		
age e	ror rate for each page								S	UCCESS		
Hits err	or rate for each page								S	UCCESS		
	e response time for eac	ch page							S	UCCESS		
	•											
umma	rv											
rofile	Successful sessions	Failed sessions	Successful pages	Failed pages	Successful hit	s Failed hit	ts Other erro	rs Tot	al KBytes s	ent Total KBytes receiv	red Av	g response ti
rofile1	44	0	377	0	1369	0	0	103	38	38760	0.0	03(0.04)
200/E	ail Critoria										_	
	ail Criteria								Resi	h		Comment
lame		file							Resu			Comment
Name Session	error rate for each pro	file							SUC	CESS		Comment
Name Session Page er	error rate for each pro ror rate for each page	file							SUC	CESS		Comment
Name Session Page er Hits ern	error rate for each pro ror rate for each page or rate for each page								SUC SUC	CESS CESS		Comment
Name Session Page er Hits ern	error rate for each pro ror rate for each page								SUC SUC	CESS		Comment
Name Session Page er Hits ern	error rate for each pro ror rate for each page or rate for each page								SUC SUC	CESS CESS		Comment
Name Session Page er Hits ern Average	error rate for each proi ror rate for each page or rate for each page e response time for each	h page							SUC SUC SUC	CESS CESS CESS		
Name Session Page er Hits ern Average Summa Profile	error rate for each proi ror rate for each page or rate for each page e response time for each ry	h page	Successful pages	Failed pages	Successful hits	Failed hits	Other errors		SUC SUC	CESS CESS CESS Total KBytes received	_	sponse time,
Name Session Page er Hits ern Average Fumma Profile	error rate for each proi ror rate for each page or rate for each page e response time for each ry	h page	Successful pages	Failed pages	Successful hits	Failed hits	Other errors	Total I	SUC SUC SUC	CESS CESS CESS	Avg re 0.04(0	sponse time,
Name Session Page er Hits ern Average Summa Profile	error rate for each proi ror rate for each page or rate for each page e response time for each ry	h page		0		0		814	SUC SUC SUC	CESS CESS CESS Total KBytes received	_	sponse time,
Name Session Page er Hits ern Average umma Profile	error rate for each proi ror rate for each page or rate for each page e response time for each ry	h page		0	995	0	0	814	SUC SUC SUC	CESS CESS CESS Total KBytes received	_	sponse time,
Name Session Page er Hits erro Average umma Profile Profile4	error rate for each proi ror rate for each page or rate for each page e response time for each TV Successful sessions	h page		0	995	0	0	814	SUC SUC SUC	CESS CESS CESS CESS Total KBytes received 26748	_	sponse time,
lame Session Page er Page er Session Session Page er Page er Session S	error rate for each proi ror rate for each page or rate for each page e response time for each TV Successful sessions	Failed sessions		0	995	0	0	814	SUC SUC SUC SUC SUC SUC SUC	CESS CESS CESS CESS Total KBytes received 26748	_	sponse time,
lame Session Page er Vage er V	error rate for each proi ror rate for each page or rate for each page e response time for each ry Successful sessions 43	Failed sessions		0	995	0	0	814	SUC	Total KBytes received 26748	_	sponse time,
Name Session Page er Page er Hits erm Average  Profile Profile4 Name Session Page er	error rate for each proi ror rate for each page or rate for each page e response time for each ry Successful sessions 43	Failed sessions		0	995	0	0	814	SUC	Total KBytes received 26748	_	sponse time,
Name Session Page er Hits erm Average Profile Pass/F Pass/F Page e Hits erm	error rate for each proi ror rate for each page or rate for each page e response time for each  Successful sessions  43  ail Criteria error rate for each pro	Failed sessions  0		0	995	0	0	814	SUC	Total KBytes received 26748	_	sponse time,
Name Session Page er Hits erm Frofile Profile Name Sessior Page e Hits erm	error rate for each provor rate for each page or rate for each page e response time for each successful sessions 43  ail Criteria error rate for each page or rate for each page	Failed sessions  0		0	995	0	0	814	SUC	Total KBytes received 26748	_	sponse time,
Name Session Session Page er Average  Forfile Profile Sessior Page el Hits er Average  Average  Average  Average  Average	error rate for each proper rate for each page or rate for each page e response time for each successful sessions  43  ail Criteria  error rate for each page or rate for each page or rate for each page or rate for each page e response time for each page e response time for each	Failed sessions  0		0	995	0	0	814	SUC	Total KBytes received 26748	_	sponse time,
Name Session Page er Hits erm Frofile Profile Name Sessior Page e Hits erm	error rate for each proper rate for each page or rate for each page e response time for each successful sessions  43  ail Criteria  error rate for each page or rate for each page or rate for each page or rate for each page e response time for each page e response time for each	Failed sessions  0		0	995	0	0	814	SUC	Total KBytes received 26748	0.04(0	sponse time,



Pass/Fail Criteria								
	Name	Result	Comment					
	Session error rate for each profile	SUCCESS						
	Page error rate for each page	SUCCESS						
	Hits error rate for each page	SUCCESS						
	Average response time for each page	SUCCESS						

Summary										
Profile	Successful sessions	Failed sessions	Successful pages	Failed pages	Successful hits	Failed hits	Other errors	Total KBytes sent	Total KBytes received	Avg response time, s
Profile6	44	n	372	n	1027	n	0	799	27805	0.05(0.06)

Pass/Fail Criteria		
Name	Result	Comment
Session error rate for each profile	SUCCESS	
Page error rate for each page	SUCCESS	
Hits error rate for each page	SUCCESS	
Average response time for each page	SUCCESS	

Summary	/											
Profile	Successful sessions	Failed sessions	Successful pages	Failed pages	Successful hits	Failed hits	Other errors	Total KBytes	sent	Total KBytes received	Avg res	ponse time, s
Profile5	45	0	399	0	1107	0	0	868		30140	0.05(0.0	16)
Pass/Fai	l Criteria											
Name									Result		Comment	
Success												

Session error rate for each profile	SUCCESS	
Page error rate for each page	SUCCESS	
Hits error rate for each page	SUCCESS	
Average response time for each page	SUCCESS	

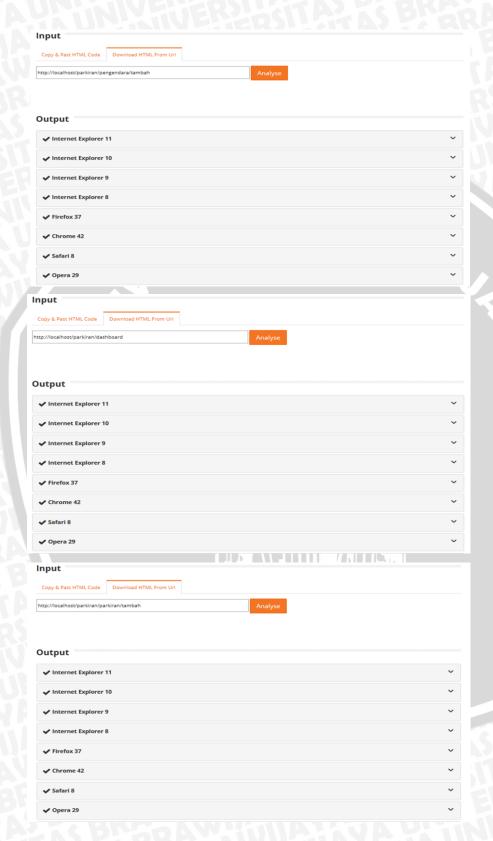
Summar	у										
Profile	Successful sessions	Failed sessions	Successful pages	Failed pages	Successful hits	Failed hits	Other errors	Total KBytes se	nt Total KBytes received	Avg resp	oonse time, s
Profile8	45	0	389	0	1071	0	0	832	28697	0.06(0.0	7)
Pass/Fail Criteria											
ass/Fail	Criteria										
Name									ult		Comment

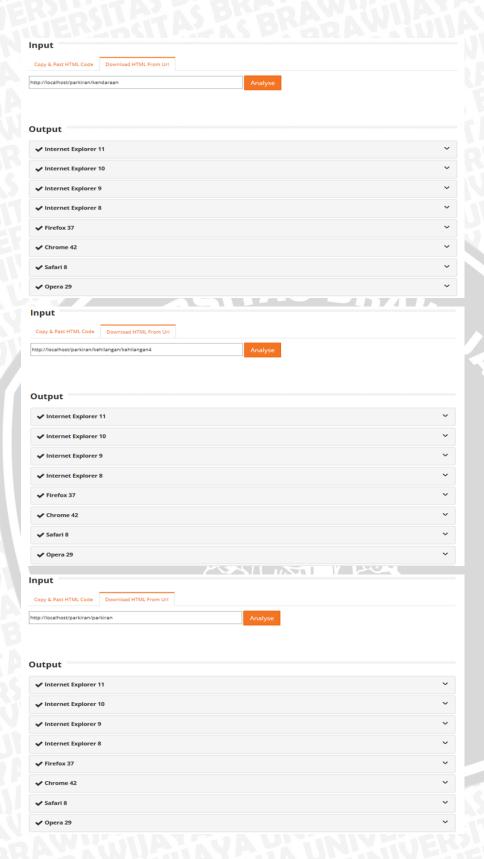
Name	Result	Comment
Session error rate for each profile	SUCCESS	
Page error rate for each page	SUCCESS	
Hits error rate for each page	SUCCESS	
Average response time for each page	SUCCESS	

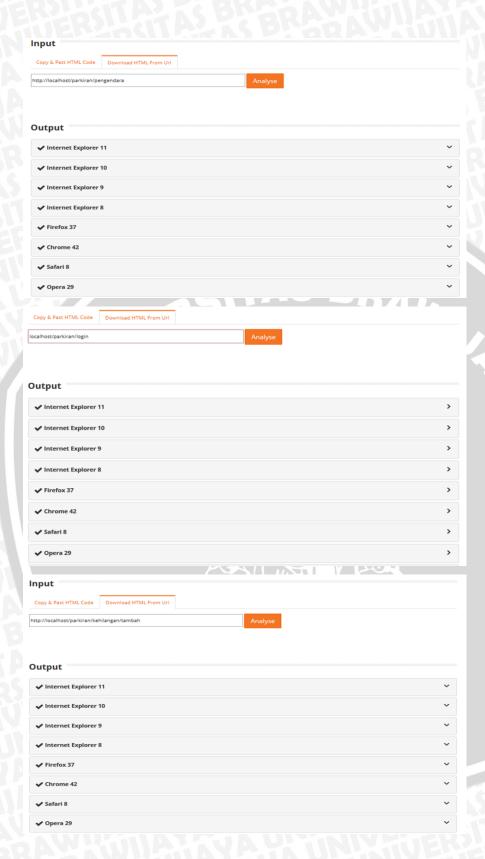
Summary											
	Profile	Successful sessions	Failed sessions	Successful pages	Failed pages	Successful hits	Failed hits	Other errors	Total KBytes sent	Total KBytes received	Avg response time, se
1	Profile7	49	0	411	0	1107	0	0	869	29804	0.05(0.06)



# LAMPIRAN C HASIL PENGUJIAN COMPATIBILITY







Input						
Copy & Past HTML Code	Download HTML From Url					
http://localhost/parkiran/ke	ndaraan/proses_cari		Analyse			
Output						
✓ Internet Explorer	11					~
✓ Internet Explorer	10					~
✓ Internet Explorer	9					~
✓ Internet Explorer	8					~
✓ Firefox 37						~
✓ Chrome 42						~
✓ Safari 8						~
✔ Opera 29						~
	68	19,		17	11/1	

