

***STREET PARKING SYSTEM* BERBASIS ANDROID POS
PADA PT. SUKSES JABAR-BANTEN**

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh kelulusan

Jenjang Strata Satu (S1)

Pada program Studi Sistem Informasi

OLEH

Andi Muhammad Isnan

351761002



**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
INDONESIA MANDIRI**

2020

***STREET PARKING SYSTEM* BERBASIS ANDROID POS
PADA PT. SUKSES JABAR-BANTEN**

SKRIPSI

**Diajukan sebagai salah satu syarat untuk memperoleh kelulusan
Jenjang Strata Satu (S1)
Pada program Studi Sistem Informasi**

**OLEH
Andi Muhammad Isnan
351761002**



**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
INDONESIA MANDIRI**

2020

LEMBAR PENGESAHAN

**STREET PARKING SYSTEM BERBASIS ANDROID POS
PADA PT. SUKSES JABAR-BANTEN**

Oleh
Andi Muhammad Isnan
351761002

Skripsi ini telah diterima dan disahkan untuk
Memenuhi persyaratan mencapai gelar

SARJANA SISTEM INFORMASI

Pada
PROGRAM STUDI SISTEM INFORMASI
SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
INDONESIA MANDIRI

Bandung, Februari 2020
Disahkan oleh

Ketua Program Studi,

Dosen Pembimbing,

Moch. Ali Ramdhani, S.T., M.Kom.
NIDN. 0403097701

Moch. Ali Ramdhani, S.T., M.Kom.
NIDN. 0403097701

LEMBAR PERSETUJUAN REVISI
STREET PARKING SYSTEM BERBASIS ANDROID POS
PADA PT. SUKSES JABAR-BANTEN

Oleh
Andi Muhammad Isnan
351761002

Telah melakukan sidang skripsi dan telah melakukan revisi sesuai dengan perubahan dan perbaikan yang diminta pada saat sidang skripsi

Bandung, Februari 2020
Menyetujui

No	Nama Dosen	Keterangan	Tanda Tangan
1.	Moch. Ali Ramdhani, S.T., M.Kom.	Pembimbing	
2.	Hendra Gunawan, S.T., M.Kom.	Penguji 1	
3.	Novi Rukhviyanti, S.T., M.Si.	Penguji 2	

Mengetahui
Ketua Program Studi Sistem Informasi

Moch. Ali Ramdhani, S.T., M.Kom.
NIDN. 0403097701

SURAT PERNYATAAN

Dengan ini saya menyatakan bahwa:

- (1) Naskah Skripsi ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik, baik di Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri maupun perguruan tinggi lainnya.
- (2) Skripsi ini murni merupakan karya penelitian saya sendiri dan tidak menjiplak karya pihak lain. Dalam hal ada bantuan atau arahan dari pihak lain maka telah saya sebutkan identitas dan jenis bantuannya di dalam lembar ucapan terima kasih.
- (3) Seandainya ada karya pihak lain yang ternyata memiliki kemiripan dengan karya saya ini, maka hal ini adalah di luar pengetahuan saya dan terjadi tanpa kesengajaan dari pihak saya.

Pernyataan ini saya buat dengan sesungguhnya dan apabila di kemudian hari terbukti adanya kebohongan dalam pernyataan ini, maka saya bersedia menerima sanksi akademik sesuai norma yang berlaku di Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri

Bandung, Februari 2020
Yang membuat pernyataan

Andi Muhammd Isnan
351761002

ABSTRAK

STREET PARKING SYSTEM BERBASIS ANDROID POS PADA PT. SUKSES JABAR-BANTEN

OLEH

Andi Muhammad Isnan

351761002

Penelitian skripsi ini dilatar belakangi oleh tidak teraturnya pengelolaan parkir-parkir liar yang mengakibatkan minimnya pendapatan pemerintah daerah yang berasal dari retribusi parkir. Hal ini disebabkan pengelolaan parkir masih dilakukan dengan cara manual/konvensional. Adapun tujuan dari penelitian skripsi ini yaitu: membuat aplikasi parkir yang informatif dan membuat aplikasi yang dapat membantu pemerintah untuk pengelolaan parkir liar.

Metode yang digunakan dalam penelitian skripsi ini yaitu: metode pengumpulan data dan metode perancangan sistem. Tahapan dalam metode pengumpulan data terdiri dari: studi pustaka, observasi, dan wawancara. Dalam metode perancangan sistem menggunakan *waterfall* model yang terdiri dari: tahap analisis, tahap perancangan, tahap *coding*, tahap implementasi uji coba dan penggunaan.

Hasil dari penelitian skripsi ini yaitu aplikasi pengelolaan parkir yang informatif. Aplikasi yang dibuat dapat melakukan pengelolaan tarif, pengelolaan petugas, pengelolaan perangkat keras, menunjukkan jumlah pendapatan yang didapat berdasar area parkir.

Kata kunci: Android Pos, *Parking System*

ABSTRACT

STREET PARKING SYSTEM BERBASIS ANDROID POS PADA PT. SUKSES JABAR-BANTEN

WRITED BY

Andi Muhammad Isnan

351761002

This research is motivated by the irregular management of the existing illegal parking lots, It also resulted in the lack of local government revenue from parking retribution. This is because parking management is still used manually / conventionally system. The purpose of this research is: creating an informative parking application and making an application that can help the government to manage illegal parking.

The method used in this research are: data collection method and system design method. The phases in the data collection method consist of: literature study, observation, and interview. In the system design method using the waterfall model, the phases consist of: the analysis phase, the design phase, the coding phase, the implementation phase of the trials and usage.

The results of this research are informative parking management application. The application can manage the fee parking, staffs, hardware management, showing the amount of income based on the parking area, etc.

Keywords: Android POS, Parking System

KATA PENGANTAR

Puji dan Syukur penulis panjatkan kepada **ALLAH SWT** yang telah memberikan rahmat dan karuniaNya sehingga penulis dapat menyelesaikan skripsi yang berjudul “ ***STREET PARKING SYSTEM BERBASIS ANDROID POS PADA PT. SUKSES JABAR-BANTEN***”.

Skripsi ini disusun untuk memenuhi syarat memperoleh gelar sarjana teknik informatika. Skripsi ini diharapkan dapat menjadi bahan bacaan dan pertimbangan bagi pihak yang memerlukannya.

Penulis mengucapkan terimakasih banyak pada pihak yang telah membimbing, membantu, memberikan saran masukan selama proses pengerjaan skripsi ini. Penulis mengucapkan terima kasih khusus kepada:

1. Kepada kedua orang tua saya yang telah sabar dalam memberi semangat dan perhatian sehingga saya dapat menyelesaikan skripsi ini.
2. Bapak Moch. Ali Ramdhani, S.T., M.Kom. sebagai ketua program studi sistem informasi, dosen wali dan juga sebagai pembimbing, yang telah dengan sabar memberi saran, petunjuk, arahan, dan masukan dalam menyelesaikan skripsi ini.
3. Bapak Dr. Chairuddin, Ir., M.M., M.T. sebagai ketua STM IK IM, yang telah memberi saran dan arahan dalam menyelesaikan skripsi ini.
4. Bapak Hendra Gunawan, S.T., M.Kom. dan Ibu Novi Rukhviyanti, S.T., M.Si. sebagai dosen penguji yang memberi saran dan arahan dalam menyelesaikan skripsi ini.

5. Teman-teman di PT. SUKSES JABAR-BANTEN, yakni Juzanda, dan Rahman, yang telah membantu dalam menyelesaikan skripsi ini. Terima kasih bray.
6. Pada Larry Page, Sergey Brin, dan Jimmy Wales. Anda semua adalah pahlawan bagi para pejuang skripsi.
7. Teman-teman program studi informasi yang namanya tidak dapat disebutkan satu-persatu. Kalian luar biasa, yang lagi skripsi semangat kaka.
8. Para dosen penguji, yang telah memberi masukan agar skripsi ini lebih baik.
9. Semua pihak yang telah membantu penulis yang tidak bisa disebutkan satu persatu.

Semoga ALLAH SWT senantiasa memberikan rahmat dan perlindungan pada semua pihak yang telah disebutkan diatas. Skripsi ini masih jauh dari kata sempurna, sehingga penulis dengan senang hati menerima saran dan masukan. Kepada peneliti lain dapat dijadikan pertimbangan dan dibuat pengembangan dengan ruang lingkup yang lebih besar. Sebagai penutup, semoga skripsi ini dapat bermanfaat bagi kita semua.

Bandung, Februari 2020
Penulis,

(ANDI MUHAMMAD ISNAN)

DAFTAR ISI

LEMBAR PENGESAHAN	iii
LEMBAR PERSETUJUAN REVISI	iv
SURAT PERNYATAAN	v
ABSTRAK	vi
ABSTRACT	vii
KATA PENGANTAR	viii
DAFTAR ISI	x
DAFTAR TABEL	xiv
DAFTAR GAMBAR	xvi
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Identifikasi Dan Rumusan Masalah	2
1.2.1. Identifikasi Masalah	2
1.2.2. Rumusan Masalah	2
1.3. Tujuan Penelitian	3
1.4. Batasan Masalah	3
1.5. Metode Penelitian	3
1.5.1. Metode Pengumpulan Data	4
1.5.2. Metode Perancangan Sistem	4
1.6. Sistematika Penulisan	5
BAB II LANDASAN TEORI	6
2.1. Pengertian Retribusi	6
2.1.1. Perbedaan Pajak Dengan Retribusi	7
2.2. Retribusi Parkir Di Tepi Jalan Umum	7
2.2.1. Pengertian Parkir	7
2.2.2. Subjek Dan Wajib Parkir Di Tepi Jalan Umum	8
2.2.3. Tata Cara Pemungutan Parkir Di Tepi Jalan Umum	8

2.3.	Pengertian Sistem	9
2.3.1.	Klasifikasi Sistem	11
2.4.	<i>Point of Sale</i> (POS)	12
2.5.	<i>Unified Modeling Language</i> (UML)	12
2.5.1.	<i>Use Case</i> Diagram.....	12
2.5.2.	<i>Activity</i> Diagram.....	14
2.5.3.	<i>Sequence</i> Diagram.....	16
2.5.4.	<i>Class</i> Diagram.....	16
2.6.	Android.....	18
2.6.1.	SunmiOs.....	18
2.7.	Basis Data.....	18
2.7.1.	Definisi Basis Data.....	18
2.7.2.	<i>Data Definition Language</i> (DDL).....	19
2.7.3.	<i>Data Manipulation Language</i> (DML)	19
2.7.4.	Sistem Basis Data Relasional.....	20
2.8.	<i>DataBase Management System</i> (DBMS)	20
2.8.1.	Fungsi Dan Fasilitas DBMS	20
2.8.2.	PostgreSQL	21
2.9.	Bahasa Pemrograman Yang Digunakan	22
2.9.1.	Kotlin	22
2.9.2.	<i>Hypertext Preprocessor</i> (PHP)	23
2.9.3.	<i>Extensible Markup Language</i> (XML).....	24
2.9.4.	<i>JavaScript</i> (JS)	25
2.10.	Perangkat Lunak Dan Perangkat Bantuan Yang Digunakan	25
2.10.1.	<i>Android Software Development Kit</i> (Android SDK)	25
2.10.2.	<i>Sunmi Software Development Kit</i> (Sunmi SDK).....	25
2.10.3.	<i>Java SE Development Kit</i> (JDK).....	26
2.10.4.	Android Studio	26
2.10.5.	Apache server XAMPP.....	26
2.10.6.	<i>JavaScript Object Notation</i> (JSON).....	27
2.10.7.	Nodejs	27

2.10.8.	<i>Code Igniter</i>	28
2.11.	<i>Waterfall Model</i>	29
2.12.	<i>QR Code</i>	30
2.13.	<i>Electronic Data Capture (EDC)</i>	31
2.14.	<i>Testing</i>	32
2.14.1.	<i>Blackbox Testing</i>	32
2.14.2.	<i>Whitebox Testing</i>	33
BAB III	ANALISA MASALAH DAN PERANCANGAN PROGRAM	34
3.1.	Analisis Sistem Yang Berjalan	34
3.1.1.	<i>Use Case Diagram</i> Sistem Yang Berjalan	34
3.1.2.	<i>Activity Diagram</i> Sistem Yang Berjalan	35
3.1.3.	Diagram Alir Sistem Yang Berjalan	36
3.1.4.	Evaluasi Sistem Yang Berjalan.....	37
3.2.	Perancangan Sistem.....	37
3.2.1.	Gambaran Sistem Usulan.....	37
3.2.2.	Perangkat Keras Yang Akan Digunakan.....	38
3.2.3.	<i>Use Case Diagram</i> Sistem Usulan	40
3.2.4.	Definisi <i>Use Case</i> Dan Aktor Sistem Usulan	41
3.2.5.	<i>Use Case</i> Skenario Sistem Usulan	43
3.2.6.	<i>Activity Diagram</i> Sistem Usulan	65
3.2.7.	<i>Sequence Diagram</i>	75
3.2.8.	<i>Entity Relationship Diagram</i>	83
3.2.9.	Class Diagram	85
3.2.10.	Rancangan Tampilan Sistem.....	86
BAB IV	IMPLEMENTASI DAN UJI COBA	94
4.1.	Implementasi Tampilan	95
4.1.1.	Tampilan Login.....	95
4.1.2.	Tampilan Menu Utama	97
4.1.3.	Tampilan Menu Parkir Masuk	99
4.1.4.	Tampilan Menu Parkir Keluar	101
4.1.5.	Tampilan Menu Cari Karcis.....	102

4.1.6.	Tampilan Menu Kelola Petugas	103
4.1.7.	Tampilan Menu Kelola Area Parkir	105
4.1.8.	Tampilan Menu Kelola Tarif	107
4.1.9.	Tampilan Menu Kelola Device	109
4.1.10.	Tampilan Menu Transaksi	111
4.1.11.	Tampilan Menu Settlement	111
4.2.	Implementasi Basis Data	112
4.3.	Blackbox Testing	116
BAB V	PENUTUP	136
5.1.	Kesimpulan	136
5.2.	Saran	136
DAFTAR PUSTAKA	137
Lampiran-Lampiran	139

DAFTAR TABEL

TABEL: 3. 1. Skenario Parkir.....	35
TABEL: 3. 2. Aktor Sistem Usulan	41
TABEL: 3. 3. Penjelasan Use Case Sistem Usulan	41
TABEL: 3. 4. Skenario Login Petugas Parkir.....	43
TABEL: 3. 5. Skenario Parkir Masuk.....	44
TABEL: 3. 6. Skenario Parkir Keluar	44
TABEL: 3. 7. Skenario Sattlement Setoran	45
TABEL: 3. 8. Skenario Login Admin.....	46
TABEL: 3. 9. Skenario Tambah Petugas Parkir	47
TABEL: 3. 10. Skenario Tambah Area Parkir.....	49
TABEL: 3. 11. Skenario Tambah Device	50
TABEL: 3. 12. Skenario Tambah Tarif	52
TABEL: 3. 13. Skenario Ubah Petugas Parkir.....	53
TABEL: 3. 14. Skenario Ubah Area Parkir	55
TABEL: 3. 15. Skenario Ubah Device.....	57
TABEL: 3. 16. Skenario Ubah Tarif.....	58
TABEL: 3. 17. Skenario Hapus Petugas Parkir	60
TABEL: 3. 18. Skenario Hapus Area Parkir.....	61
TABEL: 3. 19. Skenario Hapus Device	62
TABEL: 3. 20. Skenario Hapus Tarif	63
TABEL: 3. 21. Skenario Lihat Transaksi.....	64
TABEL: 3. 22.Skenario Lihat Settlement.....	64
TABEL: 4. 1. Perangkat Lunak Tambahan Dan Perangkat Keras.....	94
TABEL: 4. 2. Implementasi Basis Data.....	112
TABEL: 4. 3. Pengujian Login Satu	116
TABEL: 4. 4. Pengujian Login Dua	116
TABEL: 4. 5. Pengujian Login Tiga.....	117
TABEL: 4. 6. Pengujian Parkir Masuk	117
TABEL: 4. 7. Pengujian Parkir Keluar Satu	118
TABEL: 4. 8. Pengujian Parkir Keluar Dua	118
TABEL: 4. 9. Pengujian Cari Karcis	119
TABEL: 4. 10. Pengujian Login Website Satu	119
TABEL: 4. 11. Pengujian Login Website Dua	120
TABEL: 4. 12. Pengujian Login Website Tiga.....	120

TABEL: 4. 13. Pengujian Tambah Petugas Satu	121
TABEL: 4. 14. Pengujian Tambah Petugas Dua.....	122
TABEL: 4. 15. Pengujian Ubah Petugas Satu	122
TABEL: 4. 16. Pengujian Ubah Petugas Dua	123
TABEL: 4. 17. Pengujian Hapus Petugas	124
TABEL: 4. 18. Pengujian Tambah Device Satu	125
TABEL: 4. 19. Pengujian Tambah Device Dua.....	125
TABEL: 4. 20. Pengujian Ubah Device Satu.....	126
TABEL: 4. 21. Pengujian Ubah Device Dua	127
TABEL: 4. 22. Pengujian Hapus Device	127
TABEL: 4. 23. Pengujian Tambah Area Satu.....	128
TABEL: 4. 24. Pengujian Tambah Area Dua	129
TABEL: 4. 25. Pengujian Ubah Area Satu	129
TABEL: 4. 26. Pengujian Ubah Area Dua.....	130
TABEL: 4. 27. Pengujian Hapus Area.....	131
TABEL: 4. 28. Pengujian Tambah Tarif Satu.....	131
TABEL: 4. 29. Pengujian Tambah Tarif Dua	132
TABEL: 4. 30. Pengujian Ubah Tarif Satu	133
TABEL: 4. 31. Pengujian Ubah Tarif Dua	133
TABEL: 4. 32. Pengujian Hapus Tarif.....	134
TABEL: 4. 33. Pengujian Laporan Transaksi	135

DAFTAR GAMBAR

GAMBAR: 2. 1. Contoh Use Case Diagram	13
GAMBAR: 2. 2. Contoh simbol pada Use Case Diagram	13
GAMBAR: 2. 3. Contoh simbol pada Activity Diagram	14
GAMBAR: 2. 4. Contoh Activity Diagram	15
GAMBAR: 2. 5. Contoh Sequence Diagram	16
GAMBAR: 2. 6. Contoh Class Diagram	17
GAMBAR: 2. 7. Contoh Simbol Pada Class Diagram	17
GAMBAR: 2. 8. Contoh DDL	19
GAMBAR: 2. 9. Contoh DML	19
GAMBAR: 2. 10. Contoh PostgreSQL	22
GAMBAR: 2. 11. Contoh XML Pada Android Studio	24
GAMBAR: 2. 12. Tampilan Android Studio	26
GAMBAR: 2. 13. Tampilan Apache XAMPP	27
GAMBAR: 2. 14. Contoh Nodejs	28
GAMBAR: 2. 15. Website Resmi CodeIgniter	28
GAMBAR: 2. 16. Waterfall Model (Pressman, 2012)	29
GAMBAR: 2. 17. Contoh QR Code	31
GAMBAR: 2. 18. Contoh Mesin EDC	32
GAMBAR: 3. 1. Use Case Diagram Sistem Yang Berjalan	34
GAMBAR: 3. 2. Diagram Aktifitas Sistem Yang Berjalan	35
GAMBAR: 3. 3. Diagram Alir Sistem Yang Berjalan	36
GAMBAR: 3. 4. Gambaran Sistem Usulan	37
GAMBAR: 3. 5. Perangkat Sunmi V1s	39
GAMBAR: 3. 6. Use Case Diagram Sistem Usulan	40
GAMBAR: 3. 7. Activity Diagram Sistem Usulan Login Petugas Parkir	65
GAMBAR: 3. 8. Activity Diagram Sistem Usulan Login Admin	66
GAMBAR: 3. 9. Activity Diagram Sistem Usulan Parkir Masuk	66
GAMBAR: 3. 10. Activity Diagram Sistem Usulan Parkir Keluar	67
GAMBAR: 3. 11. Activity Diagram Sistem Usulan Sattlement Setoran	67
GAMBAR: 3. 12. Activity Diagram Sistem Usulan Tambah Petugas Parkir	68
GAMBAR: 3. 13. Activity Diagram Sistem Usulan Tambah Area Parkir	68
GAMBAR: 3. 14. Activity Diagram Sistem Usulan Tambah Device	69

GAMBAR: 3. 15. Activity Diagram Sistem Usulan Tambah Tarif.....	69
GAMBAR: 3. 16. Activity Diagram Sistem Usulan Ubah Petugas.....	70
GAMBAR: 3. 17. Activity Diagram Sistem Usulan Ubah Area Parkir.....	70
GAMBAR: 3. 18. Activity Diagram Sistem Usulan Ubah Device.....	71
GAMBAR: 3. 19. Activity Diagram Sistem Usulan Ubah Tarif.....	71
GAMBAR: 3. 20. Activity Diagram Sistem Usulan Hapus Petugas.....	72
GAMBAR: 3. 21. Activity Diagram Sistem Usulan Hapus Area.....	72
GAMBAR: 3. 22. Activity Diagram Sistem Usulan Hapus Device.....	73
GAMBAR: 3. 23. Activity Diagram Sistem Usulan Hapus Tarif.....	73
GAMBAR: 3. 24. Activity Diagram Sistem Usulan Sattlement Admin.....	74
GAMBAR: 3. 25. Activity Diagram Sistem Usulan Laporan Transaksi.....	74
GAMBAR: 3. 26. Sequence Diagram Login Petugas.....	75
GAMBAR: 3. 27. Sequence Diagram Login Admin.....	75
GAMBAR: 3. 28. Sequence Diagram Parkir Masuk.....	76
GAMBAR: 3. 29. Sequence Diagram Parkir Keluar.....	76
GAMBAR: 3. 30. Sequence Diagram Tambah Device.....	77
GAMBAR: 3. 31. Sequence Diagram Tambah Area Parkir.....	77
GAMBAR: 3. 32. Sequence Diagram Tambah Petugas.....	78
GAMBAR: 3. 33. Sequence Diagram Tambah Tarif.....	78
GAMBAR: 3. 34. Sequence Diagram Ubah Area Parkir.....	79
GAMBAR: 3. 35. Sequence Diagram Ubah Device.....	79
GAMBAR: 3. 36. Sequence Diagram Ubah Petugas.....	80
GAMBAR: 3. 37. Sequence Diagram Ubah Tarif.....	80
GAMBAR: 3. 38. Sequence Diagram Hapus Area Parkir.....	81
GAMBAR: 3. 39. Sequence Diagram Hapus Device.....	81
GAMBAR: 3. 40. Sequence Diagram Hapus Petugas.....	82
GAMBAR: 3. 41. Sequence Diagram Hapus Tarif.....	82
GAMBAR: 3. 42. Entity Relationship Diagram Sistem Usulan.....	84
GAMBAR: 3. 43. Class Diagram Sistem Usulan.....	85
GAMBAR: 3. 44. Rancangan Tampilan Login Pada Web.....	86
GAMBAR: 3. 45. Rancangan Tampilan Login Pada POS.....	86
GAMBAR: 3. 46. Rancangan Tampilan Utama Pada Web.....	87
GAMBAR: 3. 47. Rancangan Tampilan Utama Pada POS.....	87
GAMBAR: 3. 48. Rancangan Tampilan Parkir Masuk.....	88
GAMBAR: 3. 49. Rancangan Tampilan Parkir Keluar.....	88
GAMBAR: 3. 50. Rancangan Tampilan Cari Tiket.....	89
GAMBAR: 3. 51. Rancangan Tampilan Menu Petugas.....	89
GAMBAR: 3. 52. Rancangan Tampilan Menu Area Parkir.....	90
GAMBAR: 3. 53. Rancangan Tampilan Menu Device.....	90
GAMBAR: 3. 54. Rancangan Tampilan Menu Tarif.....	91

GAMBAR: 3. 55. Rancangan Tampilan Tambah Petugas	91
GAMBAR: 3. 56. Rancangan Tampilan Tambah Area Parkir	92
GAMBAR: 3. 57. Rancangan Tampilan Tambah Device.....	92
GAMBAR: 3. 58. Rancangan Tampilan Tambah Tarif.....	93
GAMBAR: 4. 1. Tampilan Login Web.....	95
GAMBAR: 4. 2. Tampilan Login Android.....	96
GAMBAR: 4. 3. Tampilan Menu Utama Web	97
GAMBAR: 4. 4. Tampilan Menu Utama Android	98
GAMBAR: 4. 5. Tampilan Parkir Masuk Motor	99
GAMBAR: 4. 6. Tampilan Parkir Masuk Mobil	100
GAMBAR: 4. 7. Tampilan Parkir Keluar	101
GAMBAR: 4. 8. Tampilan Cari Karcis	102
GAMBAR: 4. 9. Tampilan Menu Utama Kelola Petugas.....	103
GAMBAR: 4. 10. Tampilan Menu Tambah Petugas.....	104
GAMBAR: 4. 11. Tampilan Menu Utama Kelola Area Parkir	105
GAMBAR: 4. 12. Tampilan Menu Tambah Area Parkir.....	106
GAMBAR: 4. 13. Tampilan Menu Utama Kelola Tarif	107
GAMBAR: 4. 14. Tampilan Menu Tambah Tarif	108
GAMBAR: 4. 15. Tampilan Menu Utama Kelola Device.....	109
GAMBAR: 4. 16. Tampilan Menu Tambah Device	110
GAMBAR: 4. 17. Tampilan Menu Transaksi	111
GAMBAR: 4. 18. Tampilan Menu Settlement	111

BAB I

PENDAHULUAN

1.1. Latar Belakang

Perkembangan teknologi perangkat keras sekarang ini memang sangat pesat sekali. Sehingga pemanfaatan teknologi dapat digunakan untuk memudahkan pekerjaan. Salah satunya yaitu android *POS*, android *POS* merupakan perangkat *EDC(Electronic Data Capture)* yang berbasis sistem operasi android. Saat ini android *POS* mulai banyak digunakan & dikembangkan untuk membantu pekerjaan dengan mobilitas tinggi. Untuk memaksimalkan manfaat penggunaan teknologi perangkat keras, diperlukan pengembangan perangkat lunak yang terencana dengan baik.

Kita juga sering melakukan kegiatan diluar ruangan menggunakan kendaraan yang kemudian kita memerlukan tempat untuk parkir kendaraan, seperti pada saat mengantar anak sekolah, pergi berbelanja, dll. Permasalahan yang sering terjadi pada tempat parkir liar konvensional antara lain tidak adanya informasi besarnya tarif, pengelolaan parkir masih secara manual, dll. Bahkan sering terjadi juga para penyewa tempat parkir harus membayar dua kali. Kesemerawutan tersebut terjadi dikarenakan parkir liar tidak dikelola dengan baik. Sistem parkir manual biasanya hanya menggunakan karcis untuk setiap kendaraannya bahkan kebanyakan tanpa karcis. Kesemerawutan pengelolaan parkir ini berkaitan dengan penerimaan

pendapatan pemerintah daerah yang mengakibatkan minimnya penerimaan pendapatan pemerintah daerah yang berasal dari retribusi parkir.

Android *POS* dapat diaplikasikan sebagai sistem pengelolaan parkir. Selain itu, tampilan sistem yang dibuat pun dapat mudah digunakan. Sistem yang dibuat dapat mencetak tiket parkir yang menampilkan kapan kendaraan mulai parkir. Para pengguna lahan parkir tidak perlu lagi khawatir akan membayar dua kali dan diharapkan dapat membantu pemerintah dalam mengelola parkir liar.

1.2. Identifikasi Dan Rumusan Masalah

Dari latar belakang yang telah dijelaskan diatas, maka dapat dirumuskan sebagai berikut:

1.2.1. Identifikasi Masalah

- A. Pengelolaan parkir masih dilakukan secara manual/konvensional.
- B. Pemerintah ingin menertibkan parkir liar dan ingin meningkatkan pendapatan dari retribusi parkir.

1.2.2. Rumusan Masalah

- A. Bagaimana membuat aplikasi sistem parkir yang mudah digunakan oleh petugas parkir.
- B. Bagaimana membuat aplikasi sistem parkir yang informatif.

1.3. Tujuan Penelitian

Adapun tujuan penulisan skripsi ini adalah sebagai berikut:

- A. Membuat aplikasi sistem pengelolaan parkir yang informatif.
- B. Membuat aplikasi yang dapat membantu pemerintah untuk pengelolaan parkir liar.

1.4. Batasan Masalah

Dengan mengidentifikasi masalah-masalah yang ada agar lebih terarah dan dapat dipahami dengan mudah, maka perlu dilakukan batasan masalah. Adapun batasan masalah yang dibahas adalah:

- A. Aplikasi yang dibuat hanya mengelola jenis kendaraan motor dan mobil.
- B. Aplikasi tidak membahas masalah teknis yang terjadi, seperti tidak jujurnya petugas parkir dalam penggunaan aplikasi.
- C. Aplikasi ini berbasis client-server.
- D. Tidak membahas keamanan aplikasi & keamanan jaringan.
- E. *Blackbox testing* yang dilakukan hanya uji fungsionalitasnya saja.

1.5. Metode Penelitian

Metode yang digunakan sebagai berikut:

1.5.1. Metode Pengumpulan Data

A. Studi Pustaka

Mempelajari atau mencari bahan referensi baik dari buku, jurnal, dll.

B. Wawancara

Mengajukan pertanyaan-pertanyaan pada pengguna sistem.

C. Observasi

Meninjau langsung kegiatan retribusi parkir dan juga mengumpulkan data yang diperlukan.

1.5.2. Metode Perancangan Sistem

Metode yang digunakan adalah *waterfall* model, dan tahapannya sebagai berikut:

A. Tahap Analisis.

B. Tahap Perancangan.

C. Tahap Coding.

D. Tahap Implementasi dan Uji Coba.

E. Tahap Penggunaan dan Pemeliharaan.

1.6. Sistematika Penulisan

BAB I PENDAHULUAN

Bab ini membahas mengenai latar belakang, identifikasi dan rumusan masalah, tujuan penelitian, batasan masalah, metode penelitian, dan sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini membahas mengenai landasan teori yang digunakan dalam membuat sistem, aplikasi-aplikasi yang digunakan, bahasa pemrograman yang digunakan.

BAB III ANALISA MASALAH DAN PERANCANGAN PROGRAM

Bab ini membahas mengenai konsep dasar dalam pembuatan aplikasi, yang meliputi perancangan input, perancangan output, perancangan basis data, perancangan tampilan aplikasi.

BAB IV IMPLEMENTASI DAN UJI COBA

Bab ini membahas implementasi dari sistem dan hasil uji coba sistem seperti input program maupun hasil output program.

BAB V PENUTUP

Bab ini membahas implementasi dari sistem dan hasil uji coba sistem seperti input program maupun hasil output program.

BAB II

LANDASAN TEORI

2.1. Pengertian Retribusi

Menurut Marihot P. Siahaan (2016) yang dimaksud retribusi adalah salah satu pendapatan asli daerah yang diharapkan dapat menjadi salah satu sumber pembiayaan penyelenggaraan pemerintahan dan juga pembangunan daerah, untuk dapat meningkatkan serta pemeratakan kesejahteraan masyarakat.

Penggolongan retribusi berbeda dengan pajak, karena pada retribusi terdapat imbalan langsung oleh pembayar retribusi. Retribusi daerah menurut Undang-undang Nomor 18 tahun 1997 tentang pajak daerah dan retribusi sebagaimana telah diubah dengan Undang-undang nomor 34 tahun 2000 dan diubah kembali menjadi Undang-undang nomor 28 tahun 2009 tentang pajak daerah dan retribusi daerah dikelompokkan menjadi tiga yaitu:

- A. Retribusi jasa umum, contohnya: retribusi pelayanan sampah, retribusi pelayanan kesehatan, dan retribusi pasar.
- B. Retribusi jasa usaha, contohnya: retribusi terminal, retribusi hotel, dan retribusi parkir.
- C. Retribusi perizinan khusus, contohnya: retribusi trayek angkot dan retribusi IMB (Izin Mendirikan Bangunan).

2.1.1. Perbedaan Pajak Dengan Retribusi

Menurut Dr. Djafar Saidi, SH., MH. (2010). Perbedaan antara pajak dengan retribusi sebagai berikut:

- A. Pada pajak bersifat tidak langsung sedangkan pada retribusi bersifat langsung dan nyata pada individu tersebut.
- B. Lembaga pemungut pajak adalah pemerintah pusat maupun daerah, sedangkan lembaga pemungut retribusi hanya pemerintah daerah saja.
- C. Pajak berasal dari hukum peraturan Undang-undang sedangkan retribusi dapat berasal dari peraturan pemerintah.

2.2. Retribusi Parkir Di Tepi Jalan Umum

2.2.1. Pengertian Parkir

Berikut ini definisi parkir menurut beberapa sumber, sebagai berikut.

- A. Menurut Kepmen Perhub No. 4 Tahun 1994. Parkir adalah keadaan tidak bergerak suatu kendaraan yang tidak bersifat sementara.
- B. Menurut Undang-undang No. 22 tahun 2009 tentang lalu lintas dan angkutan jalan pasal 1. Parkir adalah keadaan kendaraan berhenti atau tidak bergerak untuk beberapa saat dan ditinggalkan pengemudinya.

2.2.2. Subjek Dan Wajib Parkir Di Tepi Jalan Umum

Subjek retribusi terdapat dalam Peraturan Daerah Kota Bandung No 09 Tahun 2010 pasal 4, sebagai berikut:

- A. Subjek retribusi pelayanan parkir adalah orang pribadi atau badan yang menggunakan/menikmati pelayanan jasa umum berupa jasa pelayanan parkir di tepi jalan umum yang disediakan dan ditentukan oleh Pemerintah Daerah.
- B. Wajib retribusi adalah orang pribadi atau badan yang menerima, menggunakan, dan menikmati penyediaan pelayanan parkir di tepi jalan umum yang ditentukan oleh Pemerintah Daerah, termasuk pemungut atau pemotong retribusi.

2.2.3. Tata Cara Pemungutan Parkir Di Tepi Jalan Umum

Tata cara pemungutan retribusi tercantum dalam Undang-undang Nomor 28 Tahun 2009 pasal 160, sebagai berikut:

- A. Retribusi dipungut dengan menggunakan Surat Ketetapan Retribusi Daerah (SKRD) atau dokumen lain yang dipersamakan.
- B. Dokumen lain yang dipersamakan yang dimaksud dapat berupa karcis, kupon, dan kartu langganan.

- C. Dalam hal wajib retribusi tertentu tidak membayar tepat pada waktunya atau kurang membayar, dikenakan sanksi administrasi sebesar 2% setiap bulan dari retribusi yang terutang yang tidak atau kurang dibayar dan ditagih dengan menggunakan STRD.
- D. Penagih retribusi terutang sebagaimana dimaksud pada ayat (3) didahului dengan surat teguran.
- E. Tata cara pelaksanaan pemungutan retribusi ditetapkan dengan peraturan kepala daerah.

2.3. Pengertian Sistem

Menurut Romney dan Steinbart (2014) sistem adalah rangkaian dua atau lebih komponen-komponen yang saling berhubungan, yang saling berinteraksi untuk mencapai sesuatu tujuan

Menurut Mulyadi (2016) sistem adalah suatu jaringan prosedur yang dibuat menurut pola yang terpadu untuk melaksanakan kegiatan pokok perusahaan.

Berdasarkan pengertian diatas dapat diartikan sistem merupakan kumpulan dari komponen-komponen yang saling berkaitan untuk mencapai tujuan dalam suatu kegiatan pokok perusahaan.

Karakteristik sistem:

A. Komponen

Sebuah sistem terdiri dari rangkaian komponen yang saling berinteraksi, bekerjasama membentuk satu kesatuan.

B. Batasan Sistem

Daerah yang membatasi antara sistem yang satu dengan sistem yang lainnya.

C. Lingkungan Luar sistem

Bentuk yang berada diluar ruang lingkup batasan sistem.

D. Penghubung Sistem

Media yang menghubungkan sistem dengan sub sistem.

E. Masukan Sistem

Sesuatu yang dimasukkan kedalam sistem dapat berupa pemeliharaan sistem dan sinyal masukan.

F. Keluaran Sistem

Hasil pengolahan sistem yang dapat berupa keluaran sinyal atau masukan bagi sub sistem.

G. Pengolah Sistem

Bagian sistem yang mengubah masukan menjadi keluaran.

H. Sasaran Sistem.

Tujuan yang ingin dicapai oleh sistem.

2.3.1. **Klasifikasi Sistem**

Menurut Krismiaji (2015) sistem dibagi menjadi empat klasifikasi, dengan keterangannya sebagai berikut:

A. Sistem Tertutup

Sistem yang secara total terisolasi dari lingkungannya. Sistem tertutup tidak memiliki penghubung dengan pihak eksternal. Sehingga sistem ini tidak akan terpengaruh oleh lingkungan yang berada diluar batas sistem.

B. Sistem Relatif Tertutup

Sistem yang berinteraksi dengan lingkungannya secara terkendali. Sistem ini memiliki penghubung dengan lingkungannya dan mengendalikan pengaruh lingkungan terhadap proses yang dilakukan sistem.

C. Sistem Terbuka

Sistem yang berinteraksi dengan lingkungannya tanpa ada kendali. Sistem ini mendapat masukan dan keluaran dari lingkungannya.

D. Sistem Umpan Balik

Sistem yang menggunakan sebagian keluaran menjadi masukan sebagian untuk proses selanjutnya.

2.4. *Point of Sale (POS)*

Menurut beepos.id (2019) POS merupakan sebuah sistem dalam dunia bisnis yang menggunakan perangkat keras dan perangkat lunak. Dua komponen tersebut saling terkait untuk membantu proses transaksi dalam perusahaan. Fungsi *Point of Sale* dalam dunia bisnis sebagai berikut :

- A. Data Lengkap.
- B. Laporan secara *real time*.
- C. Cek persediaan dan perubahan harga secara *real time*.
- D. Efisiensi waktu.

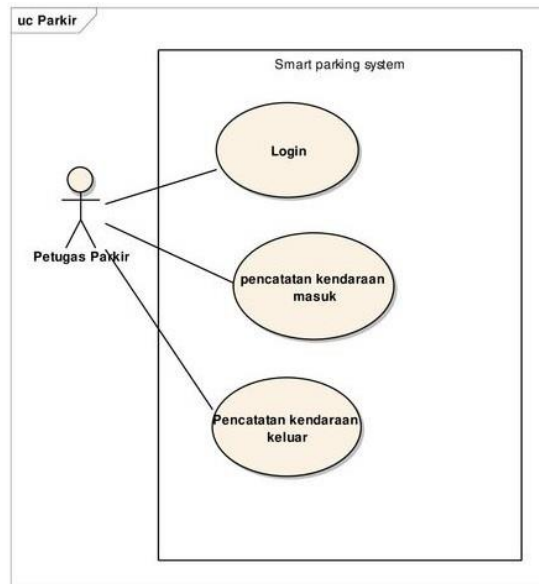
2.5. *Unified Modeling Language (UML)*

Menurut Ambler (2005) UML adalah pemodelan yang digunakan pengembang untuk berkomunikasi mengenai pekerjaan mereka para pengusaha dan pengembang lainnya.

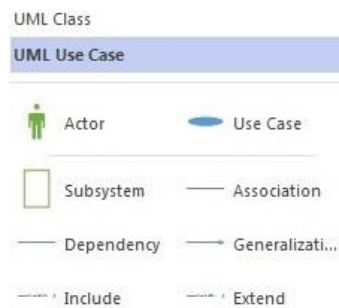
2.5.1. *Use Case Diagram*

Menurut Ambler (2005)”. *Use Case* digunakan untuk sebagai berikut:

- A. Menggambarkan sistem dari sudut pandang pengguna sistem.
- B. Memberikan gambaran dari kebutuhan pengguna sistem.



GAMBAR: 2. 1. Contoh *Use Case* Diagram



GAMBAR: 2. 2. Contoh simbol pada *Use Case* Diagram

Keterangan:

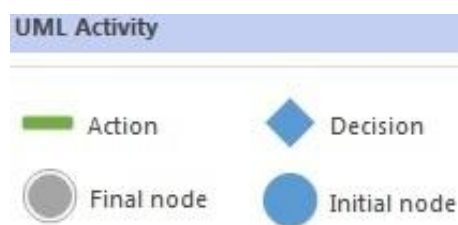
- A. *Actor*: merupakan orang, proses atau sistem yang berinteraksi.
- B. *Use case*: fungsionalitas yang disediakan sistem yang saling berinteraksi.

- C. *Association*: relasi yang menggambarkan interaksi antara *use case* dan *actor*.
- D. *Extend*: menghubungkan antar objek.
- E. *Include*: spesifikasi *use case* secara eksplisit.
- F. *Dependency*: hubungan perubahan elemen yang berpengaruh pada elemen lainnya.
- G. *Subsystem*: menampilkan spesifikasi paket secara terbatas.

2.5.2. Activity Diagram

Menurut Ambler (2005) *Activity Diagram* digunakan untuk sebagai berikut:

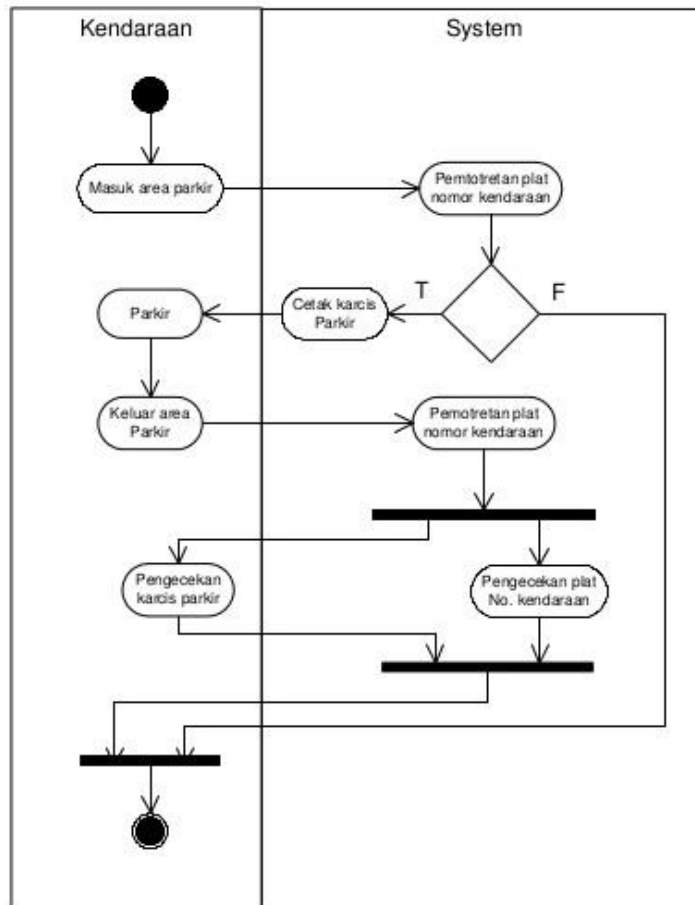
- A. Menggambarkan proses bisnis.
- B. Menggambarkan proses yang terjadi sejara bersamaan.
- C. Menggambarkan alur kerja.



GAMBAR: 2. 3. Contoh simbol pada *Activity Diagram*

Keterangan:

- A. *Action*: posisi dari sistem yang mencerminkan eksekusi dari suatu aksi.
- B. *Decision*: menggambarkan suatu keputusan.
- C. *Initial node*: menggambarkan suatu sistem/objek dibentuk.
- D. *Final node*: menggambarkan suatu sistem/objek diakhiri.

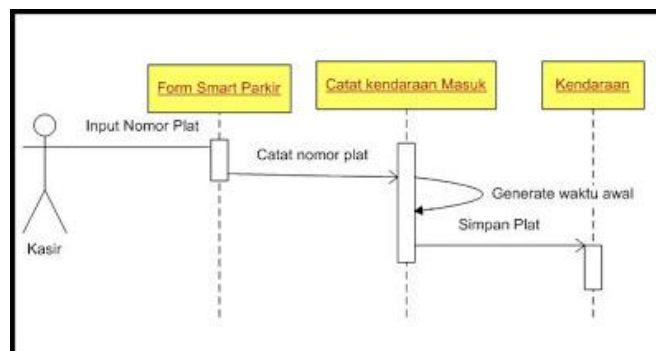


GAMBAR: 2. 4. Contoh *Activity Diagram*

2.5.3. *Sequence Diagram*

Menurut Ambler (2005) *Sequence Diagram* digunakan untuk:

- A. Menggambarkan model yang dinamis.
- B. Menggambarkan aliran fungsionalitas.

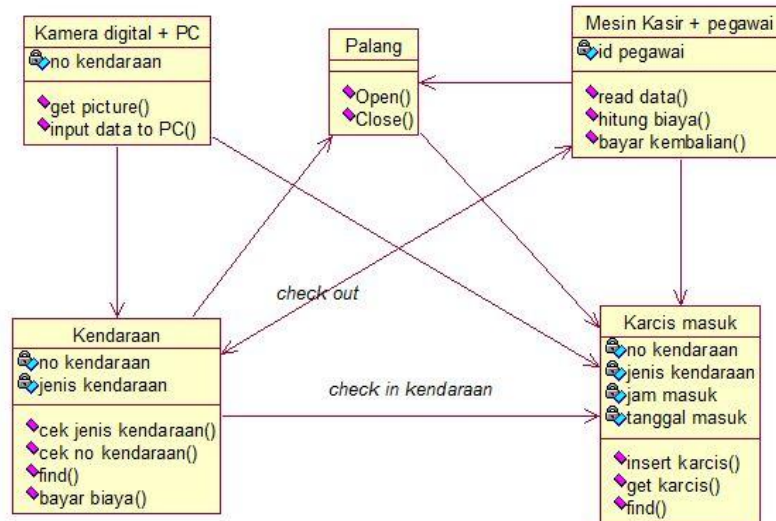


GAMBAR: 2. 5. Contoh *Sequence Diagram*

2.5.4. *Class Diagram*

Menurut Ambler (2005) *Class Diagram* digunakan untuk:

- A. Menggambarkan sistem secara rinci.
- B. Menganalisa kebutuhan



GAMBAR: 2. 6. Contoh *Class Diagram*



GAMBAR: 2. 7. Contoh Simbol Pada *Class Diagram*

Keterangan:

- Class*: Kelas pada struktur sistem.
- Interface*: Sama dengan konsep pemrograman berbasis objek.
- Association*: Relasi antar kelas.
- Directed Association*: Relasi antar kelas yang berarah.
- Dependency*: Relasi antar kelas yang ketergantungan.

F. *Aggregation*: Relasi antar kelas sebagian.

2.6. Android

Menurut Andry (2011) Android merupakan sebuah sistem operasi modifikasi kernel linux yang menyertakan *middleware* dan sejumlah aplikasi utama.

Pada awalnya sistem operasi android dikembangkan di Palo Alto, California, Amerika Serikat oleh Andy Rubin, Rich Miner, Nick Sears, dan Chris White pada tahun 2003. Pada Juli 2005 android diakuisisi oleh Google dan para pengembang android bergabung dengan Google.

2.6.1. SunmiOs

SunmiOs merupakan pengembangan sistem operasi android yang difokuskan pada keperluan bisnis dan tujuan komersial. SunmiOs dikembangkan oleh produsen *smartphone* XIAOMI.

2.7. Basis Data

2.7.1. Definisi Basis Data

- Menurut Jogiyanto (2008) Basis data adalah kumpulan data yang saling berhubungan, tersimpan dalam perangkat keras dan menggunakan perangkat lunak untuk memanipulasinya.

2.7.2. *Data Definition Language (DDL)*

DDL merupakan kumpulan kode perintah dalam basis data untuk membuat (*CREATE*), mengubah (*ALTER*), dan menghapus (*DROP*).

```

48 CREATE TABLE "public"."activity_log" (
49     "id" varchar(100) COLLATE "pg_catalog"."default" NOT NULL,
50     "ip_address" varchar(20) COLLATE "pg_catalog"."default",
51     "path" varchar(255) COLLATE "pg_catalog"."default",
52     "user" varchar(100) COLLATE "pg_catalog"."default",
53     "timestamp" timestamp(6),
54     "request" text COLLATE "pg_catalog"."default",
55     "response" text COLLATE "pg_catalog"."default"
56 );
57 ;

```

GAMBAR: 2. 8. Contoh DDL

2.7.3. *Data Manipulation Language (DML)*

DML merupakan kumpulan kode perintah dalam basis data untuk pengolahan isi data yang meliputi memasukan isi data (*INSERT*), menampilkan data (*SELECT*), hapus isi data (*DELETE*), perbaharui isi data (*UPDATE*).

```

136501 SELECT ('{'||string_agg(to_json(key)||':'||value, ','||')||'}')::json
136502 FROM (
136503     WITH to_merge AS (
136504         SELECT * FROM json_each(merge_data)
136505     )
136506     SELECT *
136507     FROM json_each(data)
136508     WHERE key NOT IN (SELECT key FROM to_merge)
136509     UNION ALL
136510     SELECT * FROM to_merge
136511 ) t;
136512 $BODY$
136513 LANGUAGE sql IMMUTABLE
136514 COST 100;

```

GAMBAR: 2. 9. Contoh DML

2.7.4. Sistem Basis Data Relasional

Menurut Heryanto (2012) sistem basis data relasional adalah basis data yang terdapat model untuk menjelaskan struktur dari basis data tersebut.

2.8. *DataBase Management System (DBMS)*

DBMS merupakan perangkat lunak untuk mengelola basis data, baik melakukan DDL maupun DML ataupun lainnya.

2.8.1. Fungsi Dan Fasilitas DBMS

Menurut Connolly dan Begg (2015), fungsi dan fasilitas DBMS diantaranya sebagai berikut:

- A. Penyimpanan, pengambilan dan pembaruan data.
- B. Katalog data yang dapat diakses oleh pengguna.
- C. Mendukung transaksi.
- D. Layanan pemulihan data.
- E. Konkurensi.
- F. Menyediakan keamanan hak akses.
- G. Integritas data.

2.8.2. PostgreSQL

PostgreSQL merupakan salah satu aplikasi DBMS yang bersifat terbuka. PostgreSQL pertama kali dikembangkan pada tahun 1986 di Universitas California, Berkeley, Amerika Serikat. Berikut ini beberapa tipe data yang terdapat pada PostgreSQL:

- *Network Address Type*

Contohnya: cidr dengan besar ukuran 7-19 bytes

- *Geometric Type*

Contohnya: circle dengan besar ukuran 24 bytes

- *Binary Type*

Contohnya: bytea dengan besar ukuran 4+n bytes

- *Numeric Type*

Contohnya: real dengan besar ukuran 4 bytes

PostgreSQL memiliki kelebihan dengan aplikasi manajemen basis data lainnya, diantaranya sebagai berikut:

A. *Unique Data Type*

PostgreSQL menyediakan tipe data beragam. Selain kita dapat menemukan tipe data yang umum kita juga akan menemukan tipe data seperti Boolean dan geometris.

B. *Multiple-Client* APIs

PostgreSQL mendukung pengembangan aplikasi klien dalam berbagai bahasa pemrograman.

id_kab_kota	jenis_kab_kota	kab_kota	id_provinsi
0244	KAB.	LUWU	020
0245	KAB.	LUWU TIMUR	020
0246	KAB.	LUWU UTARA	020
0247	KAB.	MADIUN	011
0248	KOTA	MADIUN	011
0249	KOTA	MAGELANG	009
0250	KAB.	MAGELANG	009
0251	KAB.	MAGETAN	011
0252	KAB.	MAJALENGA	031
0253	KAB.	MAJENE	016
0254	KOTA	MAKASSAR	020
0255	KAB.	MALANG	011
0256	KOTA	MALANG	011
0257	KAB.	MALINAU	005
0258	KAB.	MALIKU BARAT DAYA	015
0259	KAB.	MALIKU TENGAH	015
0260	KAB.	MALIKU TENGGARA	015
0261	KAB.	MALIKU TENGGARA BARAT	015
0262	KAB.	MAMASA	016
0263	KAB.	MAMBERAMO RAYA	001

GAMBAR: 2. 10. Contoh PostgreSQL

2.9. Bahasa Pemrograman Yang Digunakan

2.9.1. Kotlin

Kotlin merupakan bahasa pemrograman dikembangkan oleh JetBrains pada Juli 2011. Kotlin dirancang dan dikembangkan agar dapat bersinergi dengan bahasa *Java* dan *Java Virtual Machine* (JVM). Pada bulan Oktober 2017 Google secara resmi mendukung bahasa Kotlin sebagai bahasa pemrograman yang dapat digunakan untuk mengembangkan aplikasi android.

Berikut ini beberapa tipe data dan *syntax* pada Kotlin:

A. Tipe Data

```
val alamat: String = "Bandung"
```

```
val tahun: Int=2012
```


B. Konversi Tipe Data

```
val tahun: Int=2012
```

```
tahun.toString()
```

2.9.2. *Hypertext Preprocessor (PHP)*

PHP pertama kali dikembangkan oleh Rasmus Lerdorf pada tahun 1994. PHP banyak digunakan pada sisi *server*. Maksud dari sisi *server* ialah *syntax* dijalankan pada server tetapi terdapat pada halaman HTML biasa. PHP dapat beroperasi pada berbagai sistem operasi Berikut ini beberapa tipe data dan *syntax* pada PHP seperti berikut:

A. Integer

Contoh Syntax:

```
<?php
```

```
    $x=500;
```

```
    var_dump($x); ?>
```

B. String

```
<?php
```

```
    $x="Bandung";
```

```
    echo $x; ?>
```

2.9.3. Extensible Markup Language (XML)

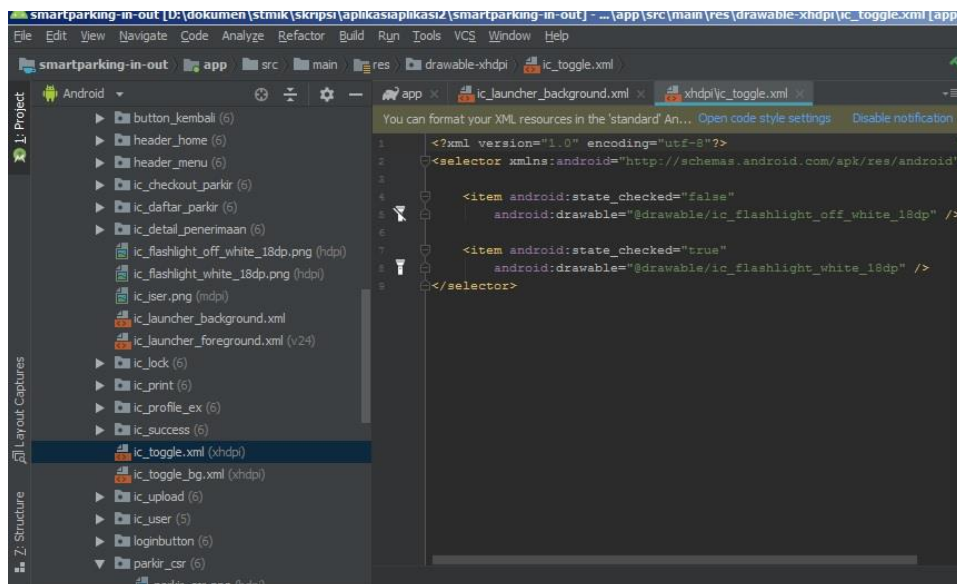
XML merupakan bahasa pemrograman yang berfungsi menyandikan dokumen yang dapat dibaca oleh manusia ataupun mesin. Tujuan dari dikembangannya XML adalah untuk meningkatkan kesederhanaan di internet. XML dapat digunakan sebagai format data dengan tujuan untuk menyimpan dan pertukaran data. Berikut ini beberapa tipe data dan syntax dalam XML:

A. Integer

```
<xs:element name="prize" type="xs:integer"/>
```

B. Decimal

```
<xs:element name="prize" type="xs:decimal"/>
```



GAMBAR: 2. 11. Contoh XML Pada Android Studio

2.9.4. JavaScript (JS)

JS merupakan bahasa pemrograman yang banyak digunakan dari sisi pengguna misalnya pada website agar lebih interaktif. JS pertama kali dikembangkan oleh Brendan Eich dari perusahaan NetScape. Berikut ini beberapa tipe data dan syntax pada JS:

A. Number

```
var length = 16;
```

B. String

```
var lastName = "Johnson";
```

2.10. Perangkat Lunak Dan Perangkat Bantuan Yang Digunakan

2.10.1. Android Software Development Kit (Android SDK)

Android SDK merupakan perangkat *Application Programming Interface* (API) yang diperlukan untuk mengembangkan aplikasi pada platform Android.

2.10.2. Sunmi Software Development Kit (Sunmi SDK)

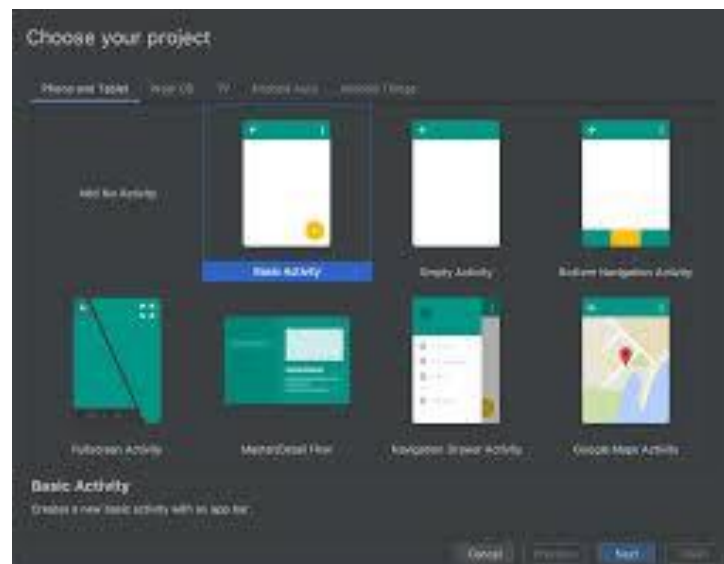
Sunmi SDK merupakan perangkat *Application Programming Interface* (API) yang diperlukan untuk mengembangkan aplikasi pada perangkat keras buatan Sunmi.

2.10.3. Java SE *Development Kit* (JDK)

JDK merupakan perangkat untuk mengkompilasi atau mengubah bahasa Java menjadi Bytecode.

2.10.4. Android Studio

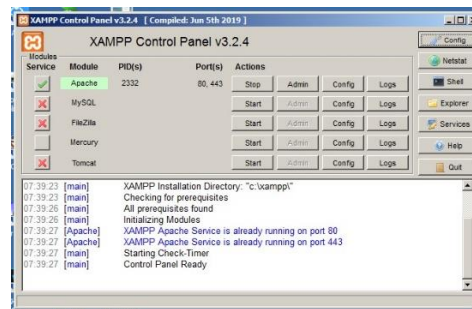
Android studio merupakan alat pengembang atau Integrated Development Environment (IDE) untuk mengembangkan aplikasi android.



GAMBAR: 2. 12. Tampilan Android Studio

2.10.5. Apache *server* XAMPP

Apache server merupakan perangkat untuk mengkompilasi bahasa pemrograman PHP menjadi Bytecode.



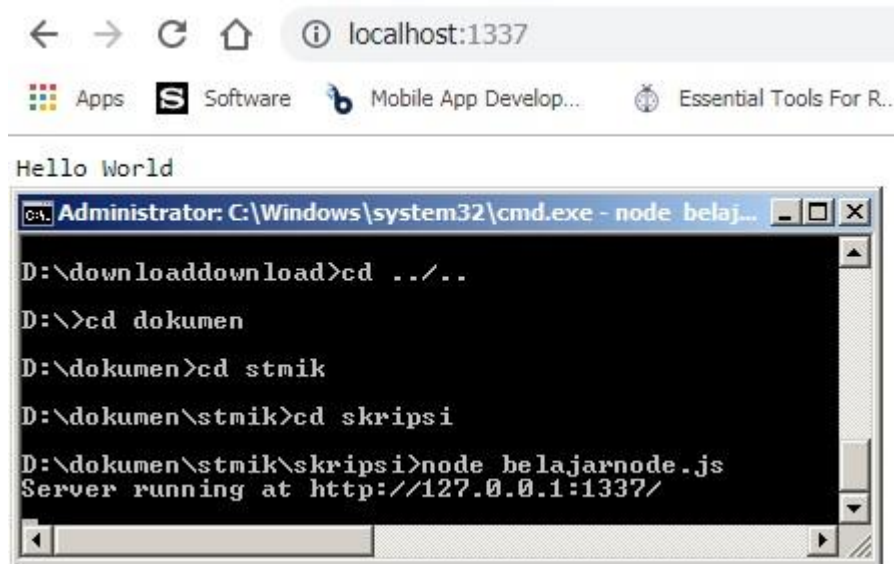
GAMBAR: 2. 13. Tampilan Apache XAMPP

2.10.6. *JavaScript Object Notation (JSON)*

JSON merupakan format data yang digunakan untuk pertukaran dan penyimpanan data. JSON merupakan bagian dari JS. JSON dapat dibaca oleh berbagai bahasa pemrograman.

2.10.7. Nodejs

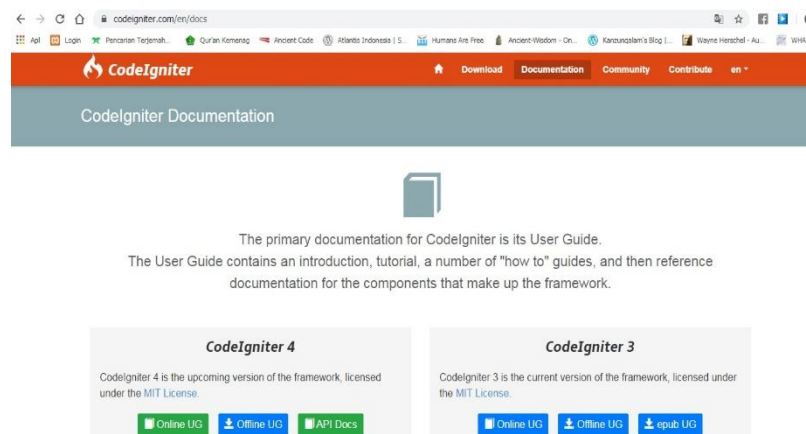
Nodejs merupakan perangkat lunak untuk mengembangkan aplikasi berbasis bahasa pemrograman JS agar dapat berjalan pada sisi server.



GAMBAR: 2. 14. Contoh Nodejs

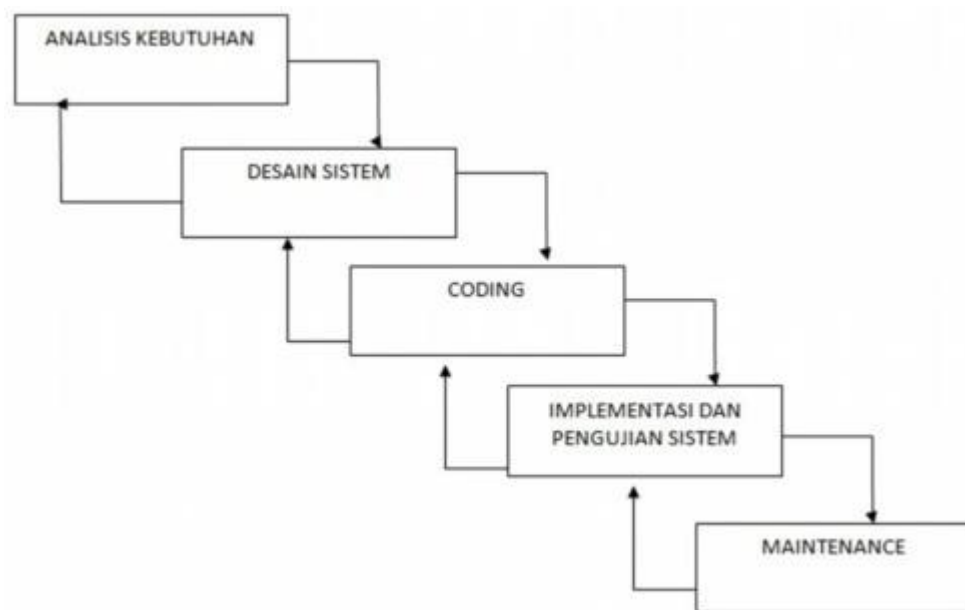
2.10.8. Code Igniter

Code Igniter merupakan sebuah *framework* aplikasi web yang dibuat oleh Rick Ellis pada tahun 2006 dan bersifat terbuka yang digunakan untuk membangun aplikasi web.

GAMBAR: 2. 15. Website Resmi *CodeIgniter*

2.11. Waterfall Model

Menurut Pressman (2012), *Waterfall* model adalah model klasik yang bersifat sistematis, berurutan dalam membuat aplikasi. Model ini juga sering disebut dengan *classic life cycle*. Model ini termasuk kedalam model generic pada rekayasa perangkat lunak dan pertama kali diperkenalkan oleh Winston Royce. Model ini melakukan pendekatan secara sistematis dan berurutan. Disebut dengan *Waterfall* karena tahap demi tahap yang dilalui harus menunggu tahap sebelumnya selesai dan berjalan berurutan. Berikut ini tahapan-tahapan dalam *Waterfall* model :



GAMBAR: 2. 16. *Waterfall* Model (Pressman, 2012)

A. Tahap Analisis

Sebelum memulai suatu pekerjaan, diperlukan komunikasi dengan pengguna untuk memahami dan mencapai tujuan yang diinginkan. Hasil dari komunikasi tersebut adalah inisialisasi proyek.

B. Tahap Desain atau Perancangan

Tahap ini merupakan perancangan dan pemodelan perangkat lunak. Tahap ini membantu dalam menentukan dalam perancangan struktur data, tampilan pengguna, algoritma program, perangkat keras yang digunakan, dll.

C. Tahap Coding

Tahap ini merupakan proses mengubah/menerjemahkan model yang telah dibuat menjadi suatu bahasa pemrograman yang dapat dimengerti oleh mesin.

D. Tahap Implementasi dan Uji Coba

Tahap ini merupakan proses uji coba perangkat lunak yang telah dibangun dengan tujuan mengecek/mencari kesalahan/kegagalan dalam perangkat lunak sebelum digunakan oleh pengguna.

E. Tahap Penggunaan dan Pemeliharaan

Tahap terakhir merupakan penggunaan perangkat lunak pada customer, pemeliharaan perangkat lunak dan juga evaluasi perangkat lunak.

2.12. QR Code

Menurut Jaringanprima.co.id (2019) *QR Code* merupakan singkatan dari *Quick Response Code*. *QR Code* dikembangkan oleh *Denso Corporation*, sebuah perusahaan Jepang yang banyak bergerak di bidang otomotif. Kapasitas data yang

dapat disimpan oleh QR Code sebesar 7089 karakter numerik, 4296 alfanumerik, 2953 *binary*. QR Code dapat dibaca dengan menggunakan kamera atau QR Scanner, serta toleransi kerusakan sampai dengan 30 persen.



GAMBAR: 2. 17. Contoh QR Code

2.13. *Electronic Data Capture (EDC)*

Mesin EDC merupakan salah satu alat otorisasi transaksi pembayaran secara elektronik. Secara umum penggunaan mesin EDC mirip dengan mesin ATM perbedaan utamanya ialah mesin EDC tidak mengeluarkan uang seperti mesin ATM dan mesin EDC lebih mudah dibawa. Jenis jalur komunikasi data pada mesin EDC sebagai berikut :



GAMBAR: 2. 18. Contoh Mesin EDC

2.14. Testing

2.14.1. Blackbox Testing

Menurut scdc.binus.ac.id (2016) *Blackbox testing* merupakan pengujian yang didasarkan pada detail aplikasi seperti tampilan aplikasi, fungsi-fungsi pada aplikasi, dan kesesuaian fungsi dengan proses bisnis yang diinginkan.

Kegiatan tester pada *blackbox testing*, sebagai berikut:

- A. Membuat *test* untuk menguji fungsi yang ada pada aplikasi.
- B. Membuat *test* untuk menguji alur kerja suatu fungsi pada aplikasi.
- C. Mencari *bugs* / kesalahan pada tampilan aplikasi.

2.14.2. *Whitebox Testing*

Menurut scdc.binus.ac.id (2016) *Whitebox testing* merupakan pengujian yang didasarkan pada detail prosedur dan alur logika kode program.

Kegiatan *tester* pada *whitebox testing*, sebagai berikut.

A. Membuat *test* untuk mencari kesalahan pada kode program.

BAB III

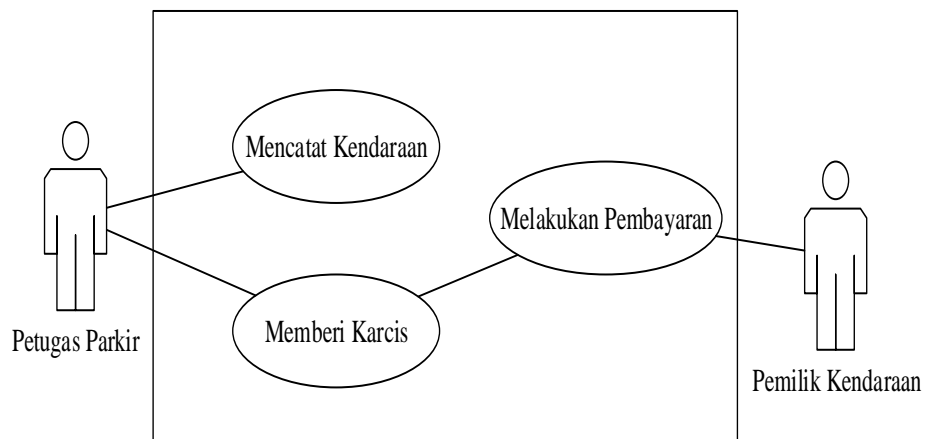
ANALISA MASALAH DAN PERANCANGAN PROGRAM

3.1. Analisis Sistem Yang Berjalan

Sebelum merancang atau membuat suatu perangkat lunak, alangkah baiknya kita terlebih dahulu mengetahui sistem yang sedang berjalan. Hal ini penting untuk mengetahui letak kekurangan atau kelemahan dari sistem yang sedang berjalan.

3.1.1. Use Case Diagram Sistem Yang Berjalan

Diagram *Use Case* merupakan pemodelan sistem yang bertujuan menggambarkan hubungan yang terjadi pada aktor didalam menyampaikan informasi parkir. Diagram *Use Case* yang sedang berjalan sebagai berikut :



GAMBAR: 3. 1. Use Case Diagram Sistem Yang Berjalan

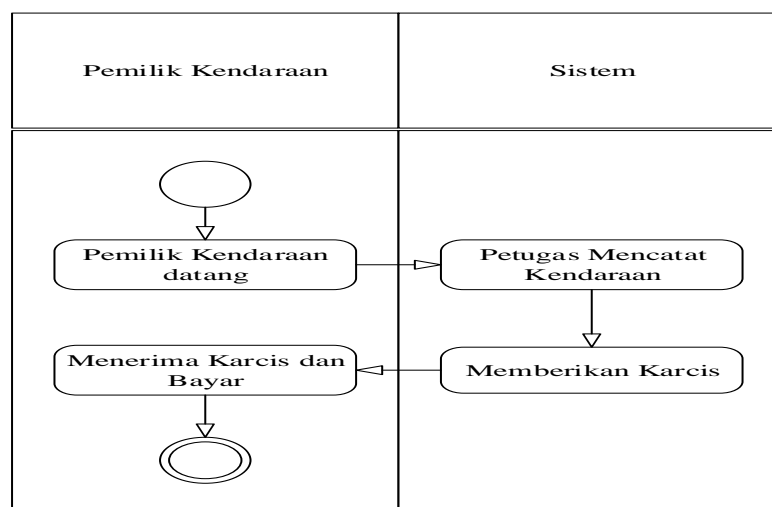
A. Skenario *Use Case*

TABEL: 3. 1. Skenario Parkir.

Nama : Parkir	
Aktor : Pemilik Kendaraan	
Pemilik Kendaraan	Sistem
1. Pemilik Kendaraan Datang.	2. Petugas Menacatat Kendaraan.
3. Bayar Parkir.	4. Memberikan Karcis.

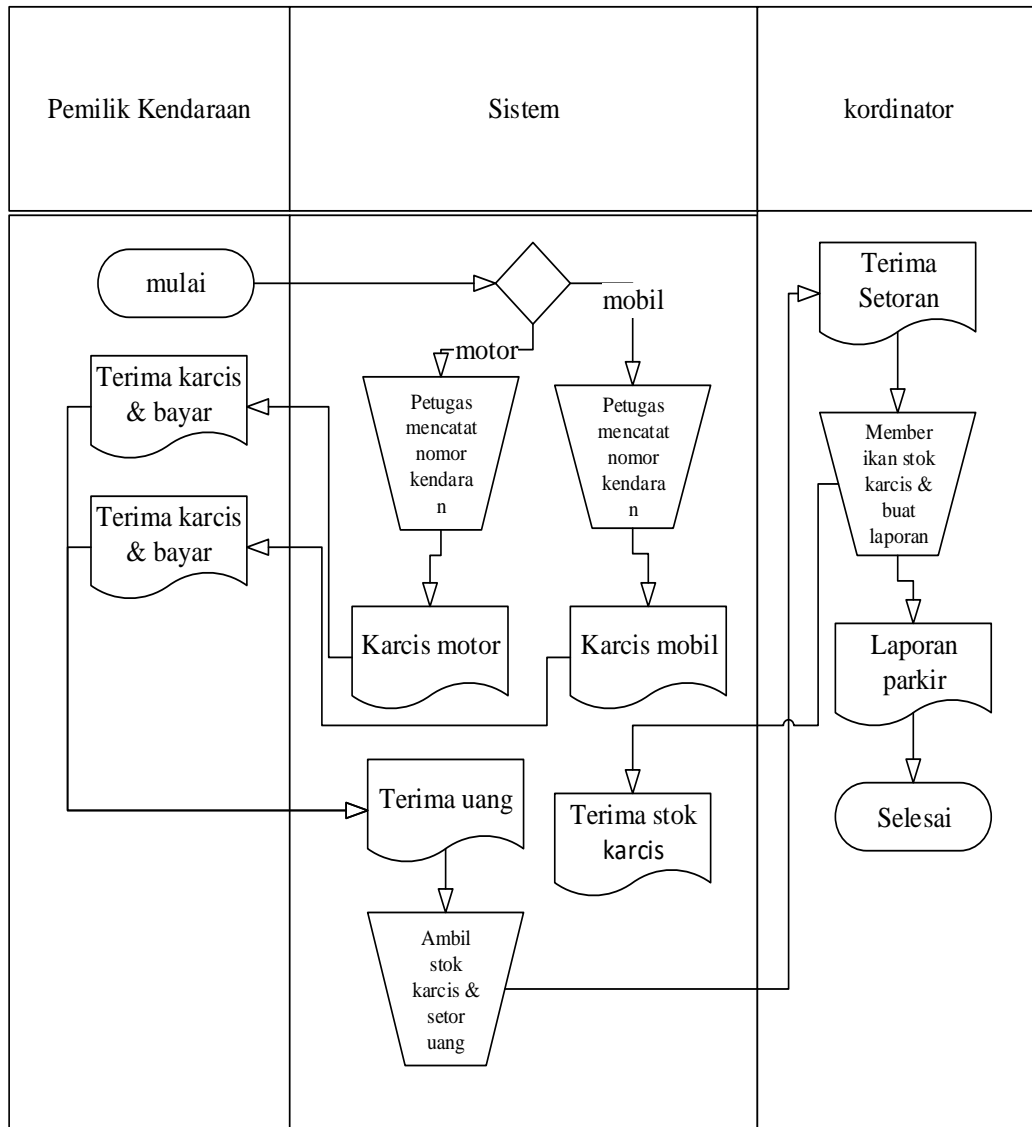
3.1.2. *Activity Diagram* Sistem Yang Berjalan

Diagram aktifitas merupakan pemodelan sistem yang bertujuan menggambarkan proses-proses yang terjadi dalam sistem. Diagram aktifitas yang sedang berjalan sebagai berikut :



GAMBAR: 3. 2. Diagram Aktifitas Sistem Yang Berjalan

3.1.3. Diagram Alir Sistem Yang Berjalan



GAMBAR: 3. 3. Diagram Alir Sistem Yang Berjalan

3.1.4. Evaluasi Sistem Yang Berjalan

Setelah menganalisa sistem yang berjalan, ternyata terdapat beberapa masalah/kendala sebagai berikut:

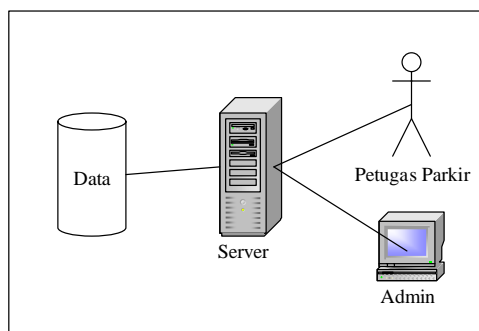
- A. Pencatatan sistem parkir masih manual sehingga tidak ada salinan jika terjadi kehilangan.
- B. Sering terjadi penyewa lahan parkir tidak membayar dikarenakan kehabisan karcis parkir.
- C. Banyak terjadi pemungutan parkir oleh petugas parkir liar.

3.2. Perancangan Sistem

Perancangan sistem merupakan gambaran sistem yang akan dibuat untuk memperbaiki masalah yang terjadi pada sistem yang berjalan.

3.2.1. Gambaran Sistem Usulan

Sistem usulan yang akan dibangun dapat dilihat pada gambar berikut:



GAMBAR: 3. 4. Gambaran Sistem Usulan

3.2.2. Perangkat Keras Yang Akan Digunakan

A. Perangkat EDC POS

Sunmi V1s, Sunmi V1s pada dasarnya merupakan handphone android yang dimodifikasi untuk tujuan khusus seperti untuk kegiatan bisnis, dll. Berikut ini spesifikasi dari Sunmi V1s :

- Sistem operasi Anroid 6.0
- CPU MTK MT6580 quad core 1.3GHZ
- Memori Internal 8GB
- RAM 1GB LPDDR3
- Layar Display IPS 5.5 inch resolusi 1280 x 720
- Kamera Belakang 5.0 megapixels
- Konektifitas Bluetooth 3.0/4.0, supports iBeacon
- Printer Thermal dengan ukuran kertas 58 mm maksimal diameter 40mm
- Baterai 3.6V/5200 mAH
- Adaptor Output DC 5V/2A Input AC 100240V/0.3A 50/60Hz



GAMBAR: 3. 5. Perangkat Sunmi V1s

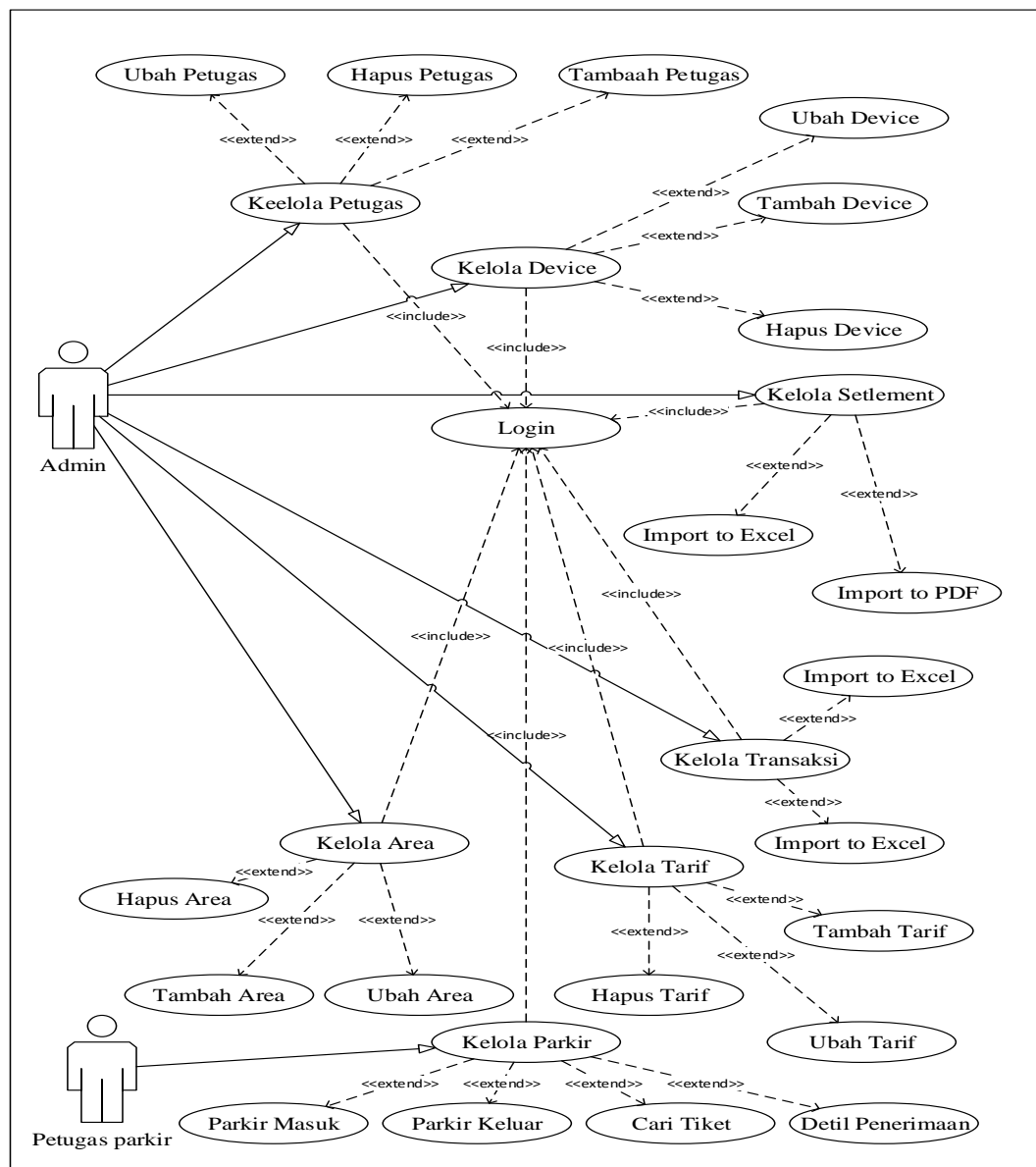
B. Perangkat *Dedicated Server*

Berikut ini spesifikasi perangkat *dedicated server* yang akan digunakan :

- Intel Xeon E3
- Ram 16 GB
- SSD 500 GB
- Bandwith international unmetered 1Gbps

3.2.3. Use Case Diagram Sistem Usulan

Diagram *Use Case* merupakan pemodelan sistem yang bertujuan menggambarkan hubungan yang terjadi pada aktor didalam menyampaikan informasi parkir. Diagram *Use Case* yang sedang berjalan sebagai berikut :



GAMBAR: 3. 6. Use Case Diagram Sistem Usulan

3.2.4. Definisi *Use Case* Dan Aktor Sistem Usulan

TABEL: 3. 2. Aktor Sistem Usulan

Aktor	Penjelasan
Admin.	Orang Yang Bertanggung Jawab Mengelola Para Petugas Parkir, Perangkat Penyewaan Parkir.
Petugas Parkir.	Petugas Operasional Penyewaan Lahan Parkir.
Pemilik Kendaraan.	Orang Yang Menyewa Lahan Parkir.

TABEL: 3. 3. Penjelasan *Use Case* Sistem Usulan

<i>Use Case</i>	Penjelasan
Petugas Parkir	
Login.	Merupakan Pengecekan Hak Penggunaan Sistem.
Parkir Masuk.	Proses Awal Penyewaan Parkir.
Parkir Keluar.	Proses Berakhirnya Penyewaan Parkir.
Cari Tiket.	Proses Pencarian Tiket Hilang.
Settlement.	Proses Penghitungan Jumlah Yang Harus Disetor.

Admin	
Tambah Petugas.	Proses Menambahkan Data Petugas.
Ubah Petugas.	Proses Merubah Data Petugas.
Hapus Petugas.	Proses Menghapus Data Petugas.
Tambah Area Parkir.	Proses Menambah Data Area Parkir.
Ubah Area Parkir.	Proses Merubah Data Area Parkir.
Hapus Area Parkir.	Proses Menghapus Data Area Parkir.
Tambah Device.	Proses Menambah Data Perangkat Yang Digunakan.
Ubah Device.	Proses Merubah Data Perangkat Yang Digunakan.
Hapus Device.	Proses Menghapus Data Perangkat Yang Digunakan.
Tambah Tarif.	Proses Menambah Data Tarif.
Ubah Tarif.	Proses Merubah Data Tarif.
Hapus Tarif.	Proses Menghapus Data Tarif.
Laporan Transaksi.	Proses Melihat Data Pemasukan Tapi Belum Disetor.
Settlement.	Proses Data Pemasukan Yang Telah Disetor.

3.2.5. Use Case Skenario Sistem Usulan

A. Petugas Parkir

TABEL: 3. 4. Skenario *Login* Petugas Parkir

Nama : Login	
Aktor : Petugas Parkir	
Petugas Parkir	Sistem
Skenario Normal	
Kondisi Awal : Belum Login dan Ingin Menggunakan Sistem.	
Kondisi Akhir : Telah Login dan Dapat Menggunakan Sistem.	
1. Masukan Nama Pengguna dan Sandi.	2. Validasi.
	3. Masuk Tampilan Utama.
Skenario Alternatif	
Kondisi Awal : Belum Login dan Ingin Menggunakan Sistem.	
Kondisi Akhir : Petugas Menghubungi Admin Agar Dapat Login.	
1. Masukan Nama Pengguna dan Sandi.	2. Validasi.
	3. Menampilkan Pesan Tidak Sesuai.
	4. Ulangi Langkah Pertama.

TABEL: 3. 5. Skenario Parkir Masuk

Nama : Parkir Masuk	
Aktor : Petugas Parkir	
Kondisi Awal : Telah Login.	
Kondisi Akhir : Pemilik Kendaraan Menerima Karcis.	
Pemilik Kendaraan	Petugas Parkir
1. Parkir Kendaraan.	2. Pilih Jenis Kendaraan.
	3. Foto Kendaraan.
	4. Cetak Karcis.
5. Menerima Karcis.	

TABEL: 3. 6. Skenario Parkir Keluar

Nama : Parkir Keluar	
Aktor : Petugas Parkir	
Pemilik Kendaraan	Petugas Parkir
Skenario Normal	
Kondisi Awal : Telah Login.	
Kondisi Akhir : Pemilik Kendaraan Telah Membayar.	
1. Menyerahkan Karcis.	2. Pindai Karcis.
	3. Hitung.
4. Bayar.	
	5. Cetak Struk.

Skenario Alternatif	
Kondisi Awal : Telah Login.	
Kondisi Akhir : Pemilik Kendaraan Telah Membayar.	
	1. Masuk Menu Pencarian.
	2. Pilih Jenis Kendaraan.
	3. Atur Perkiraan Kendaraan Masuk.
	4. Menampilkan Daftar Kendaraan.
	5. Pilih Kendaraan Yang Akan Dibayar.
	6. Hitung.
7. Bayar.	
	8. Cetak Struk.

TABEL: 3. 7. Skenario *Sattlement* Setoran

Nama : <i>Sattlement</i> Setoran	
Aktor : Petugas Parkir	
Petugas Parkir	Sistem
Skenario Normal	
Kondisi Awal : Telah Login.	
Kondisi Akhir : Petugas Menyetor Pendapatan.	

1. Masuk Menu Settlement.	
2. Pilih Laporkan.	
	3. Tampilkan Pesan Konfirmasi.
	4. Cetak Struk.
5. Menerima Struk.	
Skenario Alternatif	
Kondisi Awal : Telah Login.	
Kondisi Akhir : Petugas Tidak Jadi Menyetor.	
1. Masuk Menu Settlement.	
2. Pilih Laporkan.	
	3. Tampilkan Pesan Konfirmasi.
	4. Kembali Ke Halaman Utama.

B. Admin

TABEL: 3. 8. Skenario *Login* Admin

Nama : Login	
Aktor : Admin	
Admin	Sistem
Skenario Normal	
Kondisi Awal : Belum Login dan Ingin Menggunakan Sistem.	
Kondisi Akhir : Telah Login dan Dapat Menggunakan Sistem.	

1. Masukkan Nama Pengguna dan Sandi.	2. Validasi.
	3. Masuk Tampilan Utama.
Kondisi Alternatif	
Kondisi Awal : Belum Login dan Ingin Menggunakan Sistem.	
Kondisi Akhir : Admin Menghubungi Pengembang Agar Dapat Login.	
1. Masukkan Nama Pengguna dan Sandi.	2. Validasi.
	3. Menampilkan Pesan Tidak Sesuai.
	4. Ulangi Langkah Pertama.

TABEL: 3. 9. Skenario Tambah Petugas Parkir

Nama : Tambah Petugas Parkir	
Aktor : Admin	
Admin	Sistem
Skenario Normal	
Kondisi Awal : Telah Login.	
Kondisi Akhir : Data Petugas Berhasil Ditambah.	
1. Buka Menu Petugas.	
2. Pilih Tambah Petugas.	
3. Isi Kelengkapan Data.	

	4. Validasi.
	5. Menampilkan Pesan Berhasil Disimpan.
Skenario Alternatif 1	
Kondisi Awal : Telah Login.	
Kondisi Akhir : Data Petugas Berhasil Ditambah.	
1. Buka Menu Petugas.	
2. Pilih Tambah Petugas.	
3. Isi Kelengkapan Data.	
	4. Validasi.
	5. Menampilkan Pesan Data Tidak Sesuai.
6. Perbaiki Data.	
	7. Validasi.
	8. Menampilkan Pesan Berhasil Disimpan.
Skenario Alternatif 2	
Kondisi Awal : Telah Login	
Kondisi Akhir : Tidak Jadi Tambah Data Petugas	
1. Buka Menu Petugas.	
2. Pilih Tambah Petugas.	
3. Pilih Batal.	

	4. Tampilkan Menu Petugas.
--	----------------------------

TABEL: 3. 10. Skenario Tambah Area Parkir

Nama : Tambah Area Parkir	
Aktor : Admin	
Admin	Sistem
Skenario Normal	
Kondisi Awal : Telah Login.	
Kondisi Akhir : Data Area Parkir Berhasil Ditambah.	
1. Buka Menu Parkir.	
2. Pilih Tambah Area Parkir.	
3. Isi Kelengkapan Data.	
	4. Validasi.
	5. Menampilkan Pesan Berhasil Disimpan.
Skenario Alternatif 1	
Kondisi Awal : Telah Login.	
Kondisi Akhir : Data Area Parkir Berhasil Ditambah.	
1. Buka Menu Parkir.	
2. Pilih Tambah Area Parkir.	
3. Isi Kelengkapan Data.	
	4. Validasi.

	5. Menampilkan Pesan Data Tidak Sesuai.
6. Perbaiki Data.	
	7. Validasi.
	8. Menampilkan Pesan Berhasil Disimpan.
Skenario Alternatif 2	
Kondisi Awal : Telah Login.	
Kondisi Akhir : Tidak Jadi Tambah Data Area Parkir.	
1. Buka Menu Parkir.	
2. Pilih Tambah Area Parkir.	
3. Pilih Batal.	
	4. Tampilkan Menu Parkir.

TABEL: 3. 11. Skenario Tambah *Device*

Nama : Tambah Device	
Aktor : Admin	
Admin	Sistem
Skenario Normal	
Kondisi Awal : Telah Login.	
Kondisi Akhir : Data Device Berhasil Ditambah.	
1. Buka Menu Device.	

2. Pilih Tambah Device.	
3. Isi Kelengkapan Data.	
	4. Validasi.
	5. Menampilkan Pesan Berhasil Disimpan.
Skenario Alternatif 1	
Kondisi Awal : Telah Login.	
Kondisi Akhir : Data Device Berhasil Ditambah.	
1. Buka Menu Device.	
2. Pilih Tambah Device.	
3. Isi Kelengkapan Data.	
	4. Validasi.
	5. Menampilkan Pesan Data Tidak Sesuai.
6. Perbaiki Data.	
	7. Validasi.
	8. Menampilkan Pesan Berhasil Disimpan.
Skenario Alternatif 2	
Kondisi Awal : Telah Login.	
Kondisi Akhir : Tidak Jadi Tambah Data Device.	
1. Buka Menu Device.	

2. Pilih Tambah Device.	
3. Pilih Batal.	
	4. Tampilkan Menu Device.

TABEL: 3. 12. Skenario Tambah Tarif

Nama : Tambah Tarif	
Aktor : Admin	
Admin	Sistem
Skenario Normal	
Kondisi Awal : Telah Login.	
Kondisi Akhir : Data Tarif Berhasil Ditambah.	
1. Buka Menu Tarif.	
2. Pilih Tambah Tarif.	
3. Isi Kelengkapan Data.	
	4. Validasi.
	5. Menampilkan Pesan Berhasil Disimpan.
Skenario Alternatif 1	
Kondisi Awal : Telah Login.	
Kondisi Akhir : Data Tarif Berhasil Ditambah.	
1. Buka Menu Tarif.	
2. Pilih Tambah Tarif.	

3. Isi Kelengkapan Data.	
	4. Validasi.
	5. Menampilkan Pesan Data Tidak Sesuai.
6. Perbaiki Data.	
	7. Validasi.
	8. Menampilkan Pesan Berhasil Disimpan.
Skenario Alternatif 2	
Kondisi Awal : Telah Login.	
Kondisi Akhir : Tidak Jadi Tambah Data Tarif.	
1. Buka Menu Tarif.	
2. Pilih Tambah Tarif.	
3. Pilih Batal.	
	4. Tampilkan Menu Tarif.

TABEL: 3. 13. Skenario Ubah Petugas Parkir

Nama : Ubah Petugas Parkir	
Aktor : Admin	
Admin	Sistem
Skenario Normal	
Kondisi Awal : Telah Login.	

Kondisi Akhir : Data Petugas Berhasil Diubah.	
1. Buka Menu Petugas.	
2. Pilih Nama Petugas.	
3. Ubah Kelengkapan Data.	
	4. Validasi.
	5. Menampilkan Pesan Berhasil Disimpan.
Skenario Alternatif 1	
Kondisi Awal : Telah Login.	
Kondisi Akhir : Data Petugas Berhasil Diubah.	
1. Buka Menu Petugas.	
2. Pilih Nama Petugas.	
3. Ubah Kelengkapan Data.	
	4. Validasi.
	5. Menampilkan Pesan Data Tidak Sesuai.
6. Perbaiki Data.	
	7. Validasi.
	8. Menampilkan Pesan Berhasil Disimpan.
Skenario Alternatif 2	
Kondisi Awal : Telah Login.	

Kondisi Akhir : Data Petugas Tidak jadi Diubah.	
1. Buka Menu Petugas.	
2. Pilih Nama Petugas.	
3. Ubah Kelengkapan Data.	
4. Pilih Batal.	
	5. Tampilkan Menu Petugas.

TABEL: 3. 14. Skenario Ubah Area Parkir

Nama : Ubah Area Parkir	
Aktor : Admin	
Admin	Sistem
Skenario Normal	
Kondisi Awal : Telah Login.	
Kondisi Akhir : Data Area Parkir Berhasil Diubah.	
1. Buka Menu Parkir.	
2. Pilih Nama Area Parkir.	
3. Ubah Kelengkapan Data.	
	4. Validasi.
	5. Menampilkan Pesan Berhasil Disimpan.
Skenario Alternatif 1	
Kondisi Awal : Telah Login.	

Kondisi Akhir : Data Area Parkir Berhasil Diubah.	
1. Buka Menu Parkir.	
2. Pilih Nama Area Parkir.	
3. Ubah Kelengkapan Data.	
	4. Validasi.
	5. Menampilkan Pesan Data Tidak Sesuai.
6. Perbaiki Data.	
	7. Validasi.
	8. Menampilkan Pesan Berhasil Disimpan.
Skenario Alternatif 2	
Kondisi Awal : Telah Login.	
Kondisi Akhir : Data Area Parkir Tidak jadi Diubah.	
1. Buka Menu Parkir.	
2. Pilih Nama Area Parkir.	
3. Ubah Kelengkapan Data.	
4. Pilih Batal.	
	5. Tampilkan Menu Parkir.

TABEL: 3. 15. Skenario Ubah *Device*

Nama : Ubah Device	
Aktor : Admin	
Admin	Sistem
Skenario Normal	
Kondisi Awal : Telah Login.	
Kondisi Akhir : Data Device Berhasil Diubah.	
1. Buka Menu Device.	
2. Pilih Nomor Device .	
3. Ubah Kelengkapan Data.	
	4. Validasi.
	5. Menampilkan Pesan Berhasil Disimpan.
Skenario Alternatif 1	
Kondisi Awal : Telah Login.	
Kondisi Akhir : Data Device Berhasil Diubah.	
1. Buka Menu Device.	
2. Pilih Nomor Device.	
3. Ubah Kelengkapan Data.	
	4. Validasi.
	5. Menampilkan Pesan Data Tidak Sesuai.
6. Perbaiki Data.	

	7. Validasi.
	8. Menampilkan Pesan Berhasil Disimpan.
Skenario Alternatif 2	
Kondisi Awal : Telah Login.	
Kondisi Akhir : Data Device Tidak jadi Diubah.	
1. Buka Menu Device.	
2. Pilih Nomor Device.	
3. Ubah Kelengkapan Data.	
4. Pilih Batal.	
	5. Tampilkan Menu Device.

TABEL: 3. 16. Skenario Ubah Tarif

Nama : Ubah Tarif	
Aktor : Admin	
Admin	Sistem
Skenario Normal	
Kondisi Awal : Telah Login.	
Kondisi Akhir : Data Tarif Berhasil Diubah.	
1. Buka Menu Tarif.	
2. Pilih Nomor Tarif.	
3. Ubah Kelengkapan Data.	

	4. Validasi.
	5. Menampilkan Pesan Berhasil Disimpan.
Skenario Alternatif 1	
Kondisi Awal : Telah Login.	
Kondisi Akhir : Data Tarif Berhasil Diubah.	
1. Buka Menu Tarif.	
2. Pilih Nomor Tarif.	
3. Ubah Kelengkapan Data.	
	4. Validasi.
	5. Menampilkan Pesan Data Tidak Sesuai.
6. Perbaiki Data.	
	7. Validasi.
	8. Menampilkan Pesan Berhasil Disimpan.
Skenario Alternatif 2	
Kondisi Awal : Telah Login.	
Kondisi Akhir : Data Device Tidak jadi Diubah.	
1. Buka Menu Tarif.	
2. Pilih Nomor Tarif.	
3. Ubah Kelengkapan Data.	

4. Pilih Batal.	
	5. Tampilkan Menu Tarif.

TABEL: 3. 17. Skenario Hapus Petugas Parkir

Nama : Hapus Petugas Parkir	
Aktor : Admin	
Admin	Sistem
Skenario Normal	
Kondisi Awal : Telah Login.	
Kondisi Akhir : Data Petugas Berhasil Dihapus.	
1. Buka Menu Petugas.	
2. Pilih Nama Petugas.	
	3. Menampilkan Pesan Konfirmasi.
	4. Menampilkan Pesan Berhasil Dihapus.
Skenario Alternatif	
Kondisi Awal : Telah Login.	
Kondisi Akhir : Data Petugas Tidak Jadi Dihapus.	
1. Buka Menu Petugas.	
2. Pilih Nama Petugas.	

	3. Menampilkan Pesan Konfirmasi.
	4. Kembali Ke Menu Petugas.

TABEL: 3. 18. Skenario Hapus Area Parkir

Nama : Hapus Area Parkir	
Aktor : Admin	
Admin	Sistem
Skenario Normal	
Kondisi Awal : Telah Login	
Kondisi Akhir : Data Area Parkir Berhasil Dihapus	
1. Buka Menu Parkir.	
2. Pilih Nama Area Parkir.	
	3. Menampilkan Pesan Konfirmasi.
	4. Menampilkan Pesan Berhasil Dihapus.
Skenario Alternatif	
Kondisi Awal : Telah Login	
Kondisi Akhir : Data Area Parkir Tidak Jadi Dihapus	
1. Buka Menu Parkir.	
2. Pilih Nama Area Parkir.	

	3. Menampilkan Pesan Konfirmasi.
	4. Kembali Ke Menu Parkir.

TABEL: 3. 19. Skenario Hapus *Device*

Nama : Hapus Device	
Aktor : Admin	
Admin	Sistem
Skenario Normal	
Kondisi Awal : Telah Login.	
Kondisi Akhir : Data Device Berhasil Dihapus.	
1. Buka Menu Device.	
2. Pilih Nomor Device.	
	3. Menampilkan Pesan Konfirmasi.
	4. Menampilkan Pesan Berhasil Dihapus.
Skenario Alternatif	
Kondisi Awal : Telah Login.	
Kondisi Akhir : Data Device Tidak Jadi Dihapus.	
1. Buka Menu Device.	
2. Pilih Nomor Device.	

	3. Menampilkan Pesan Konfirmasi.
	4. Kembali Ke Menu Device.

TABEL: 3. 20. Skenario Hapus Tarif

Nama : Hapus Tarif	
Aktor : Admin	
Admin	Sistem
Skenario Normal	
Kondisi Awal : Telah Login.	
Kondisi Akhir : Data Tarif Berhasil Dihapus.	
1. Buka Menu Tarif.	
2. Pilih Nomor Tarif.	
	3. Menampilkan Pesan Konfirmasi.
	4. Menampilkan Pesan Berhasil Dihapus.
Skenario Alternatif	
Kondisi Awal : Telah Login.	
Kondisi Akhir : Data Tarif Tidak Jadi Dihapus.	
1. Buka Menu Tarif.	
2. Pilih Nomor Tarif.	

	3. Menampilkan Pesan Konfirmasi.
	4. Kembali Ke Menu Device.

TABEL: 3. 21. Skenario Lihat Transaksi

Nama : Lihat Transaksi	
Aktor : Admin	
Admin	Sistem
Skenario Normal	
Kondisi Awal : Telah Login.	
Kondisi Akhir : Mengetahui Total Transaksi Yang Belum Disetor.	
1. Buka Menu Transaksi.	
2. Pilih Unduh File.	
	3. Merespon Laporan Yang Ingin Diunduh.

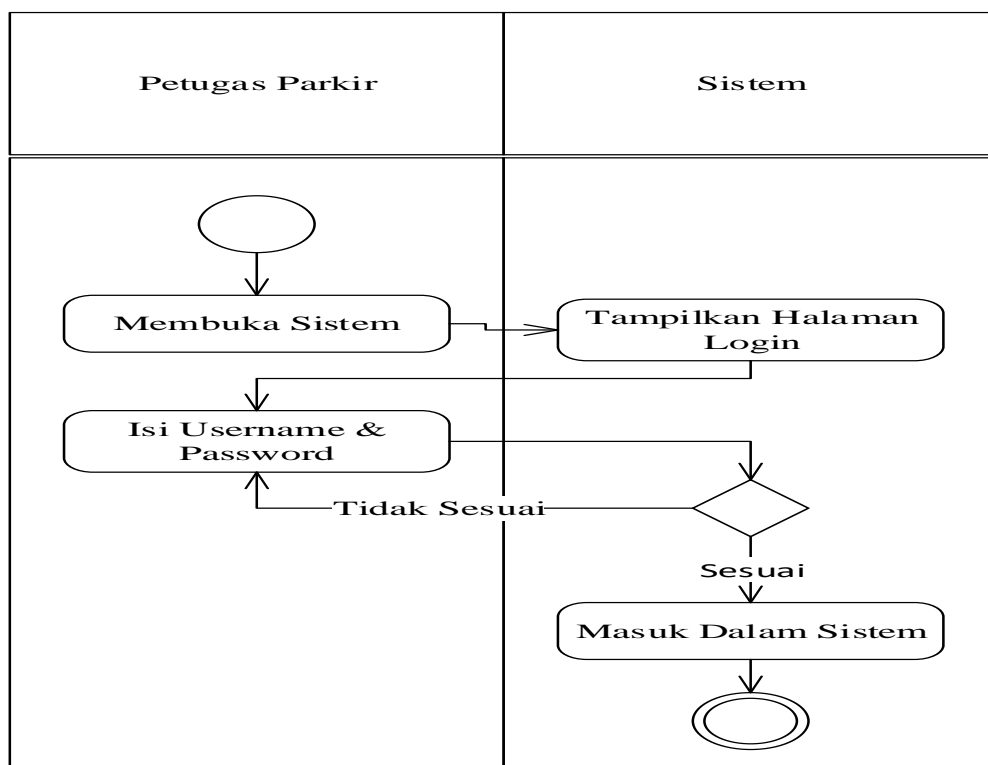
TABEL: 3. 22.Skenario Lihat *Settlement*

Nama : Lihat Transaksi	
Aktor : Admin	
Admin	Sistem
Skenario Normal	
Kondisi Awal : Telah Login.	

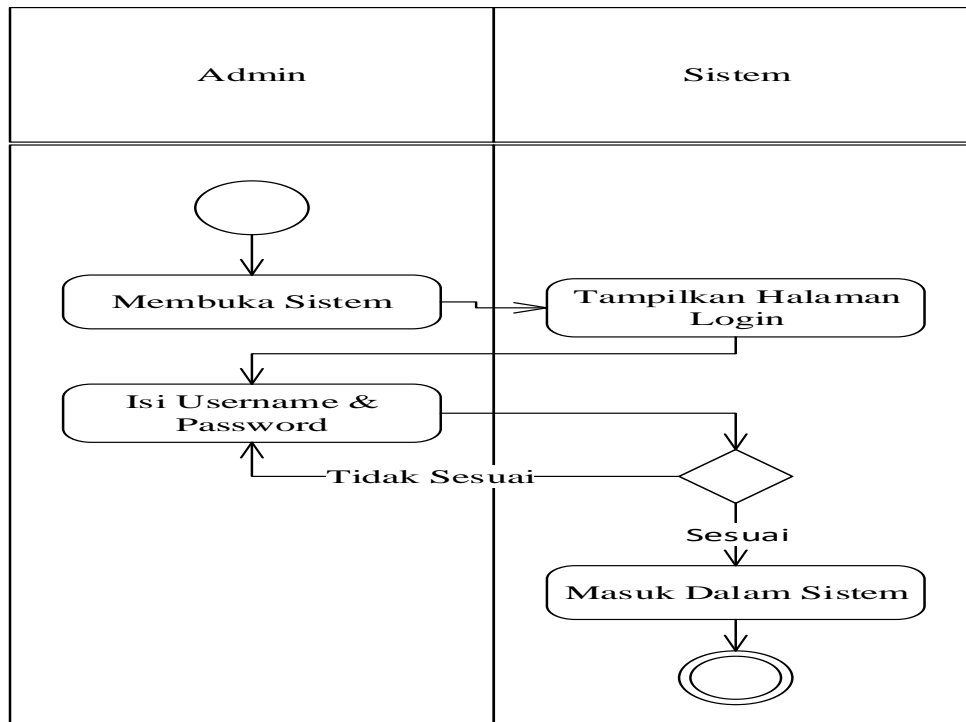
Kondisi Akhir : Mengetahui Total Transaksi Yang Telah Disetor.	
1. Buka Menu Settlement.	
2. Pilih Unduh File.	
	3. Merespon Laporan Yang Ingin Diunduh.

3.2.6. Activity Diagram Sistem Usulan

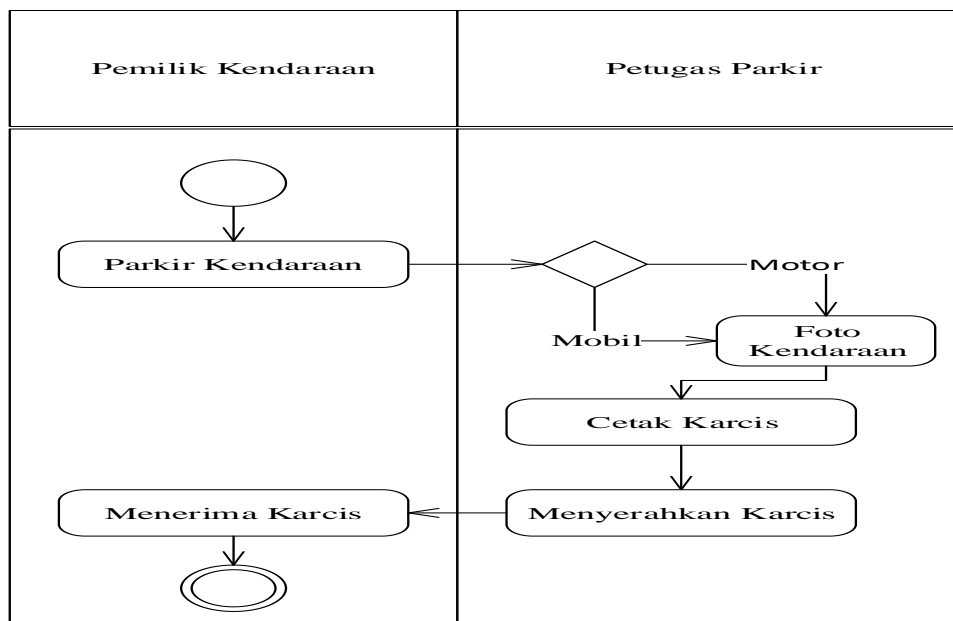
Diagram aktifitas merupakan pemodelan sistem yang bertujuan menggambarkan proses-proses yang terjadi dalam sistem. Diagram aktifitas yang sedang berjalan sebagai berikut :



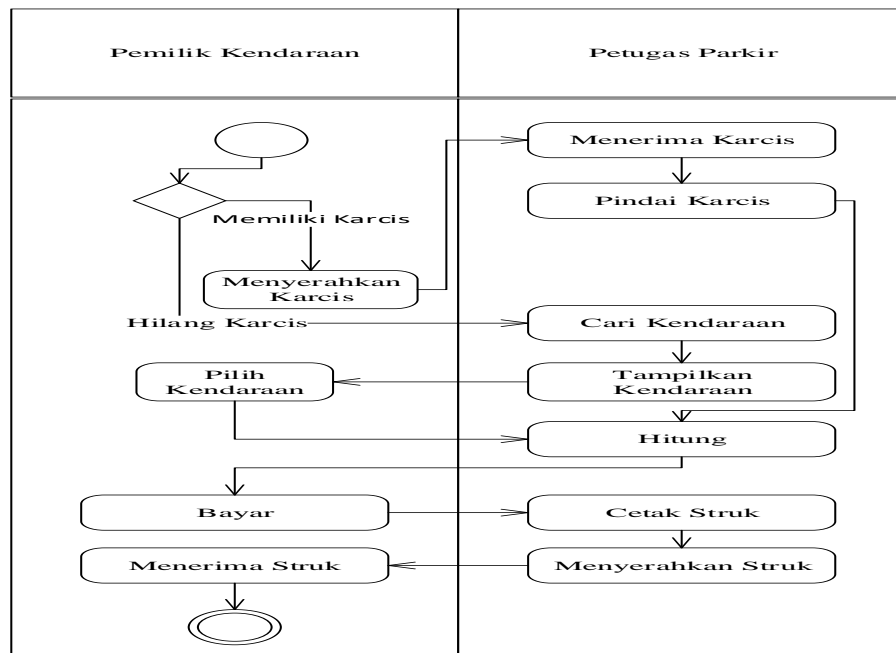
GAMBAR: 3. 7. Activity Diagram Sistem Usulan Login Petugas Parkir



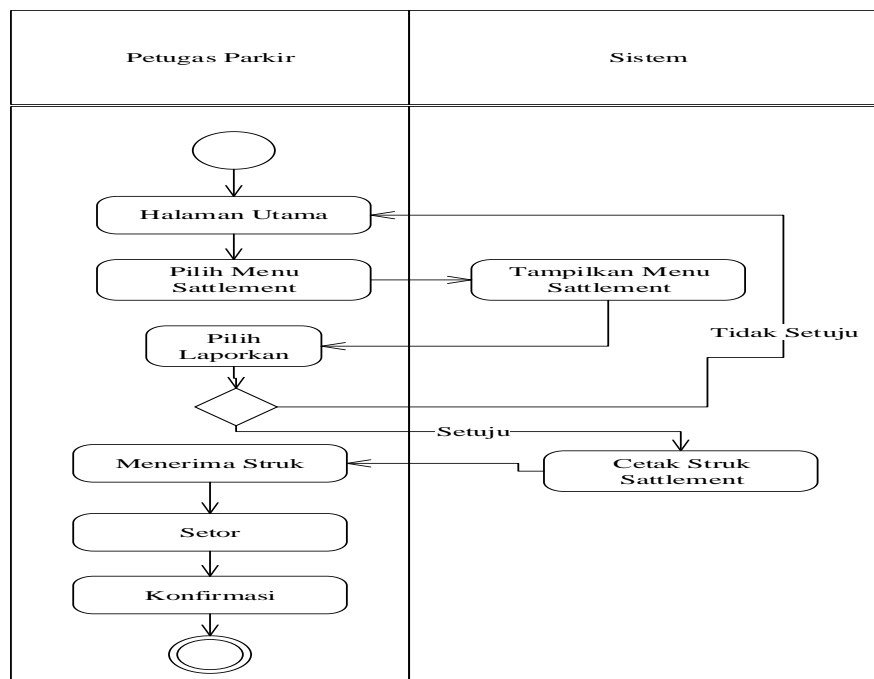
GAMBAR: 3. 8. Activity Diagram Sistem Usulan Login Admin



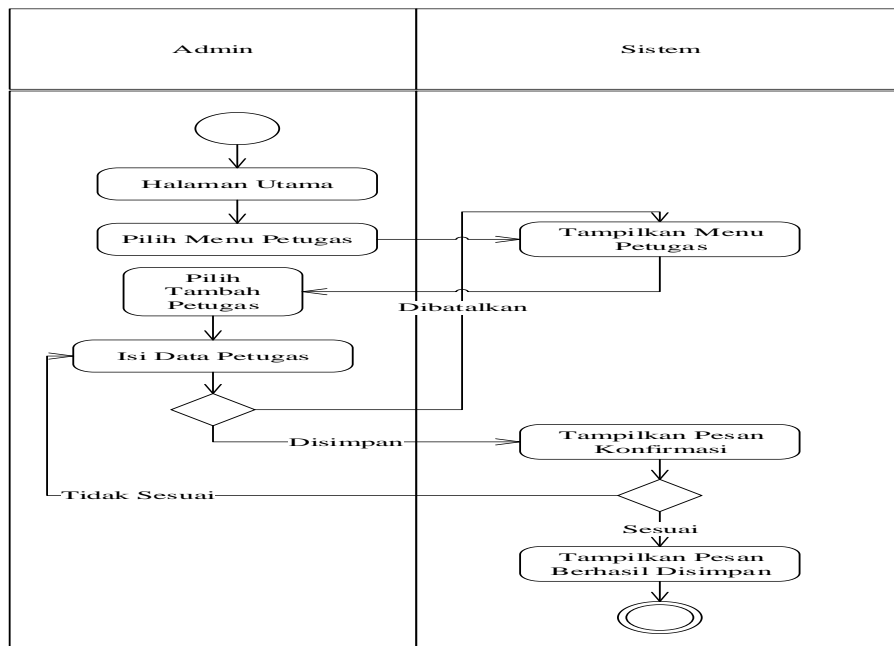
GAMBAR: 3. 9. Activity Diagram Sistem Usulan Parkir Masuk



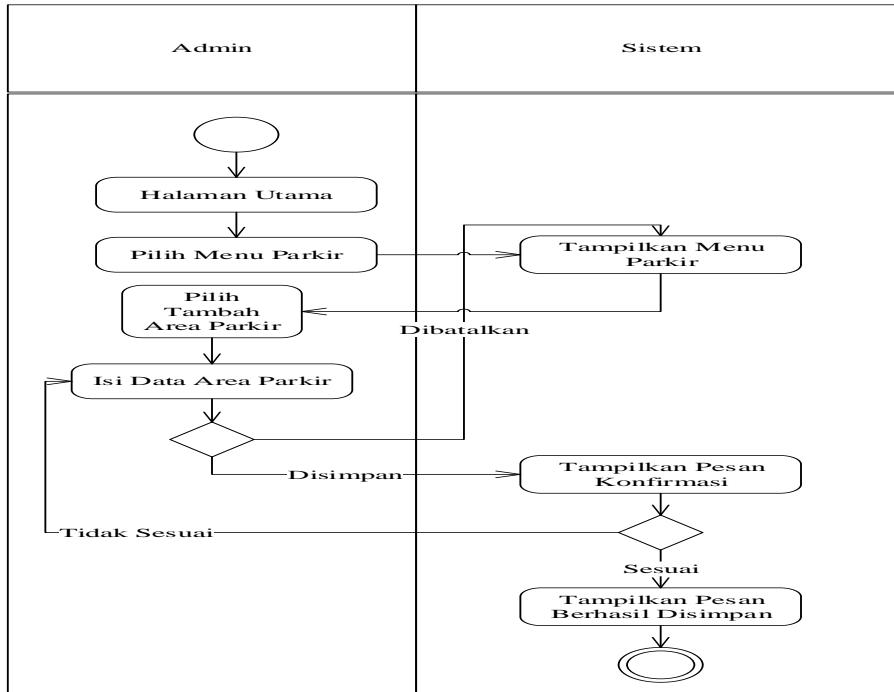
GAMBAR: 3. 10. Activity Diagram Sistem Usulan Parkir Keluar



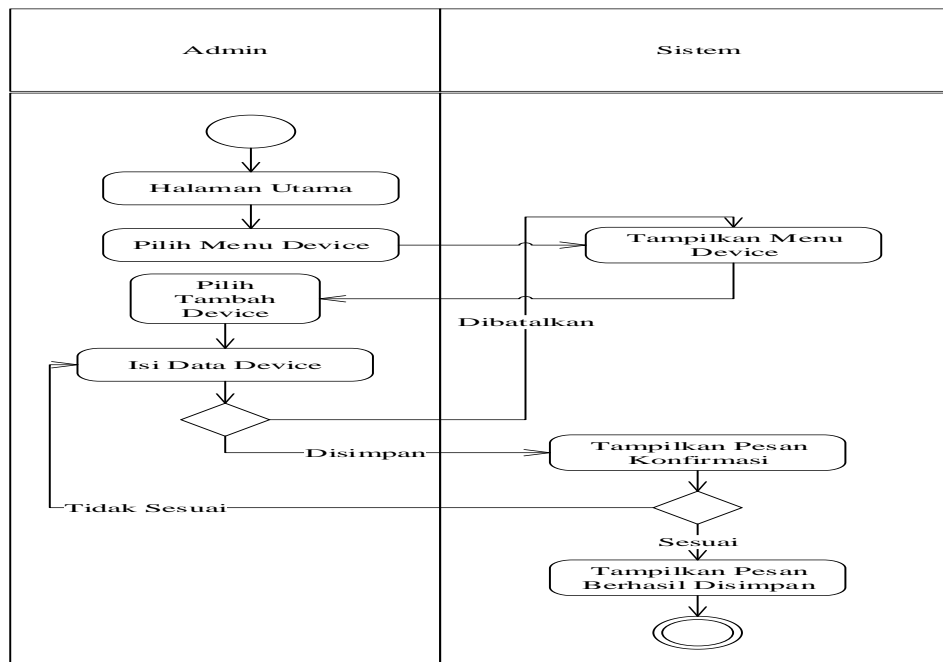
GAMBAR: 3. 11. Activity Diagram Sistem Usulan Sattlement Setoran



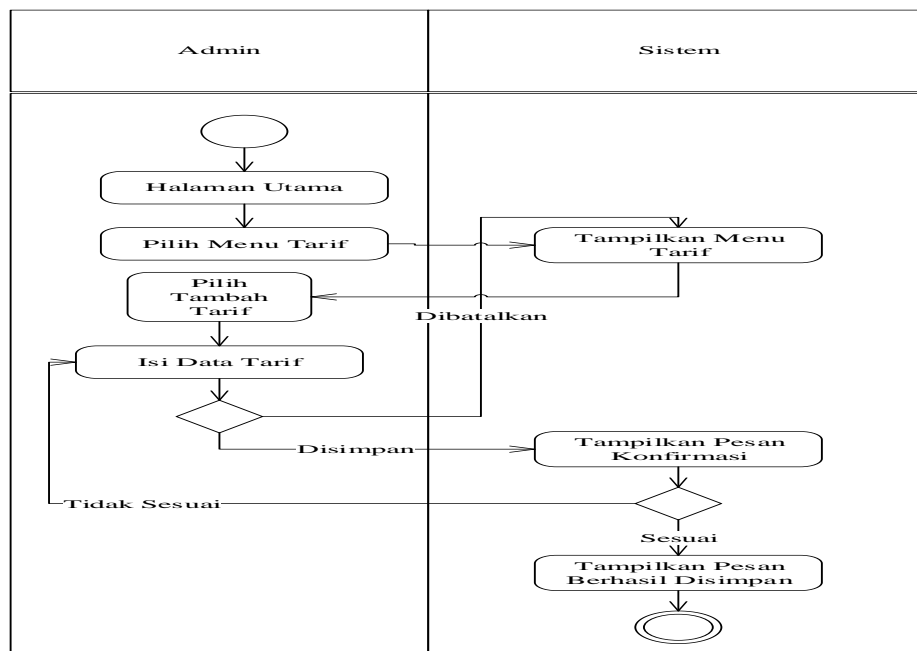
GAMBAR: 3. 12. Activity Diagram Sistem Usulan Tambah Petugas Parkir



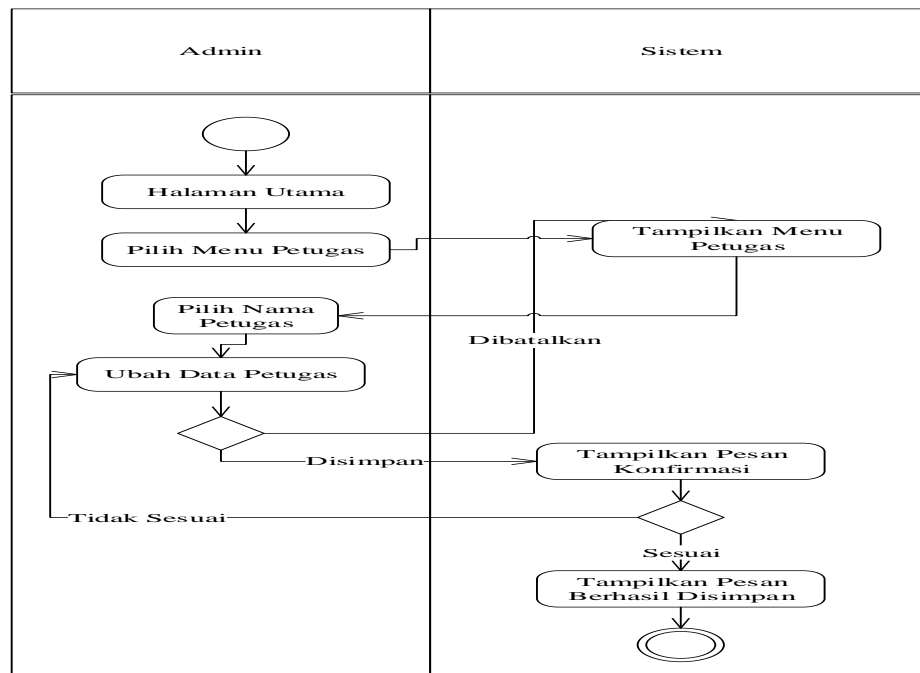
GAMBAR: 3. 13. Activity Diagram Sistem Usulan Tambah Area Parkir



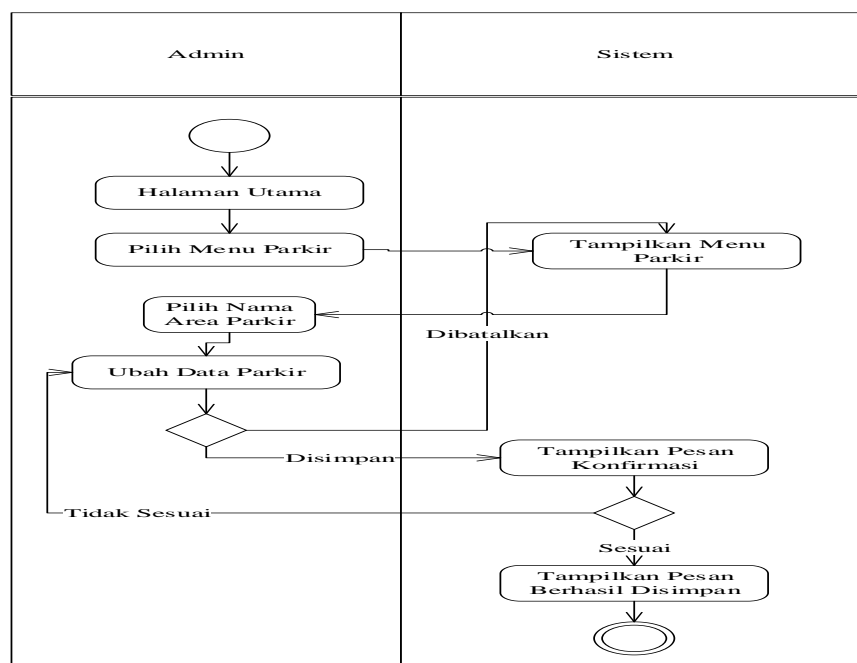
GAMBAR: 3. 14. Activity Diagram Sistem Usulan Tambah Device



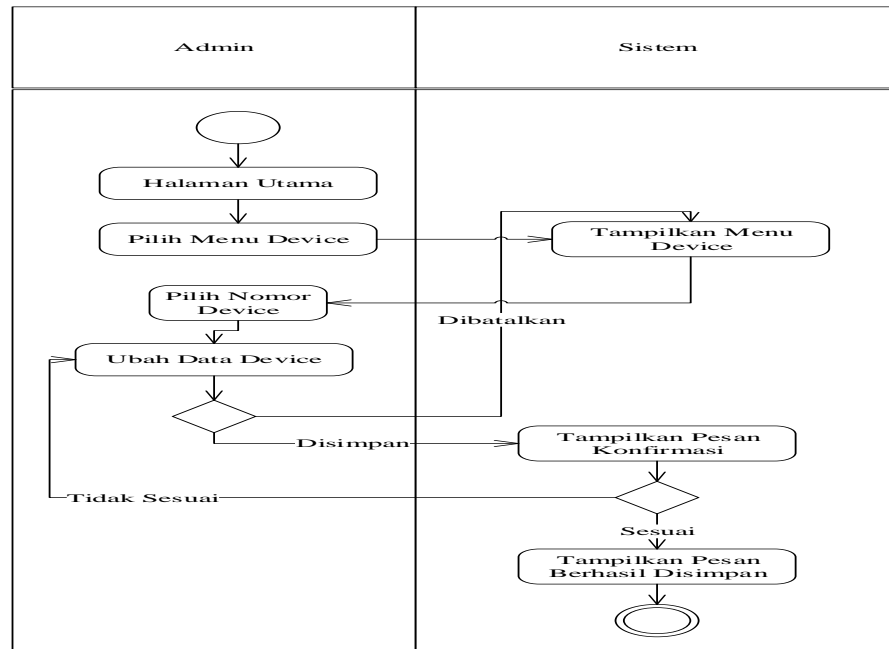
GAMBAR: 3. 15. Activity Diagram Sistem Usulan Tambah Tarif



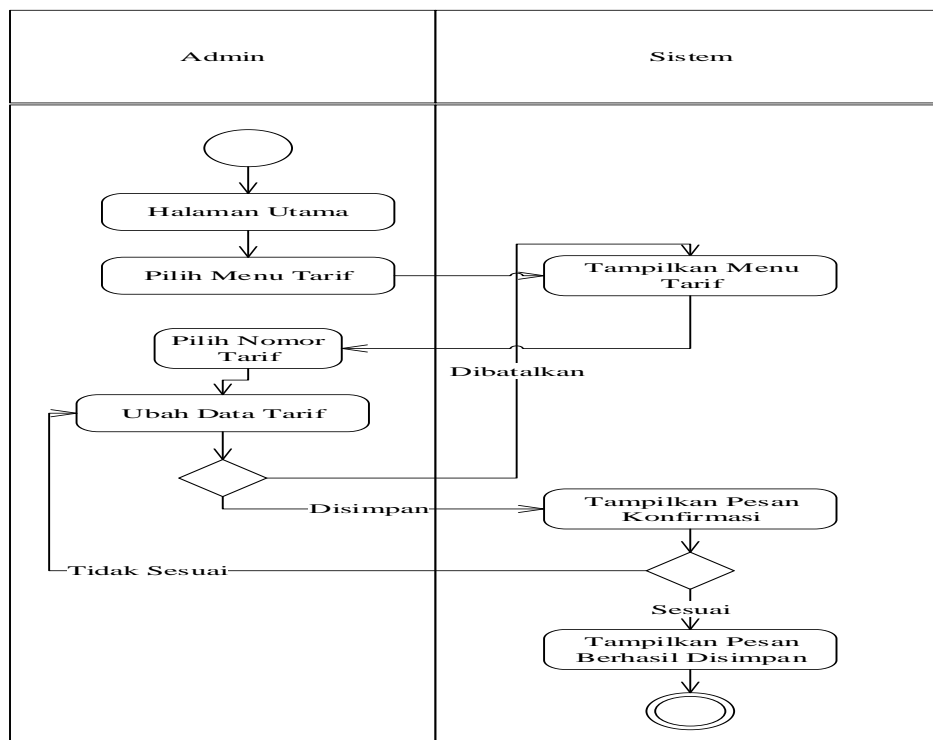
GAMBAR: 3. 16. Activity Diagram Sistem Usulan Ubah Petugas



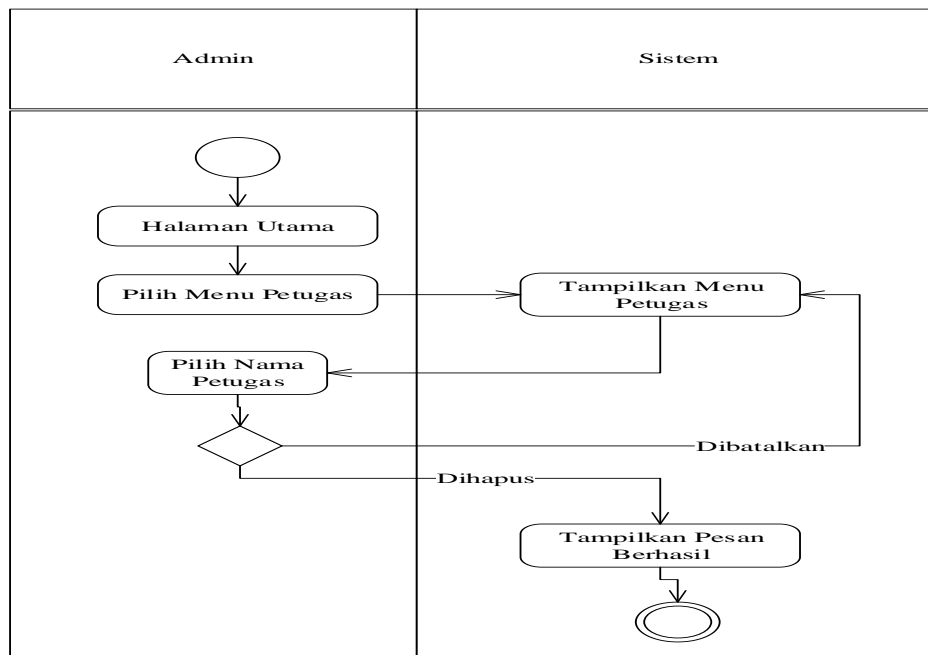
GAMBAR: 3. 17. Activity Diagram Sistem Usulan Ubah Area Parkir



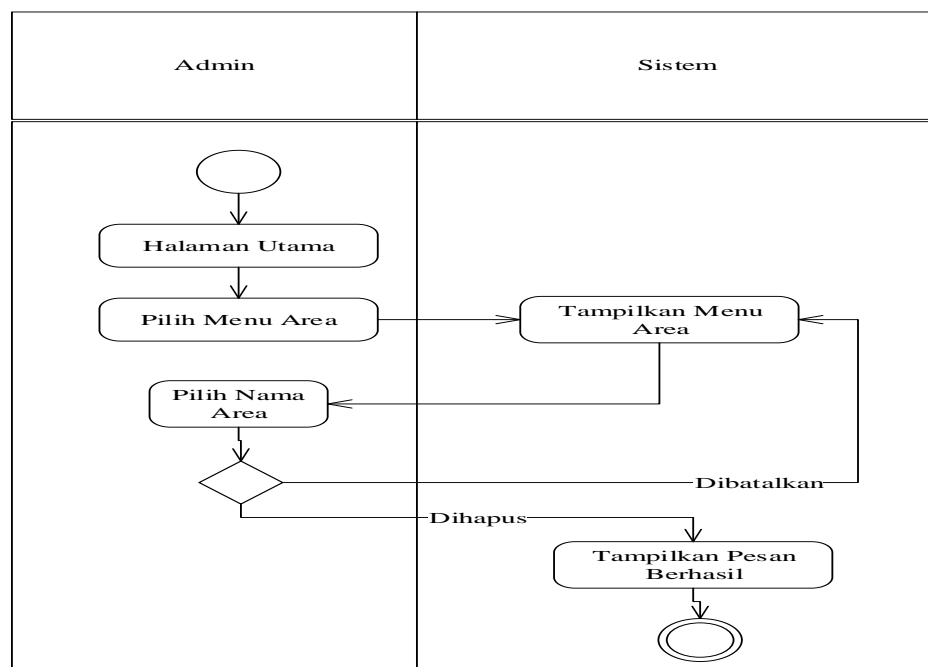
GAMBAR: 3. 18. Activity Diagram Sistem Usulan Ubah *Device*



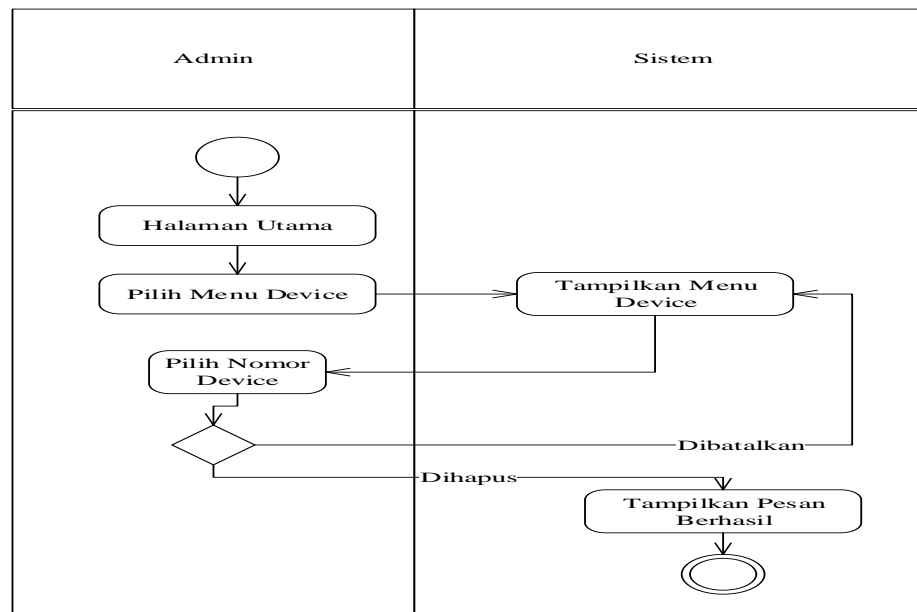
GAMBAR: 3. 19. Activity Diagram Sistem Usulan Ubah Tarif



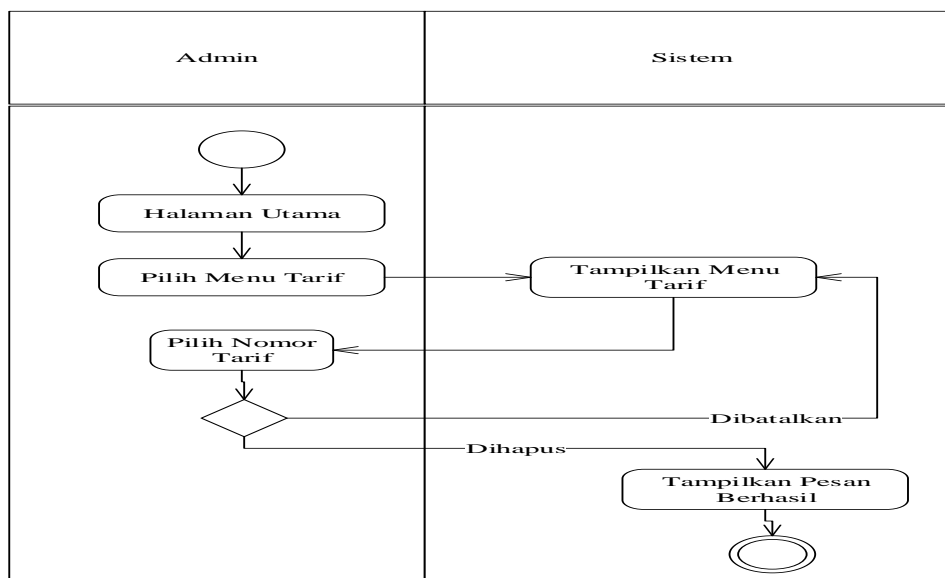
GAMBAR: 3. 20. Activity Diagram Sistem Usulan Hapus Petugas



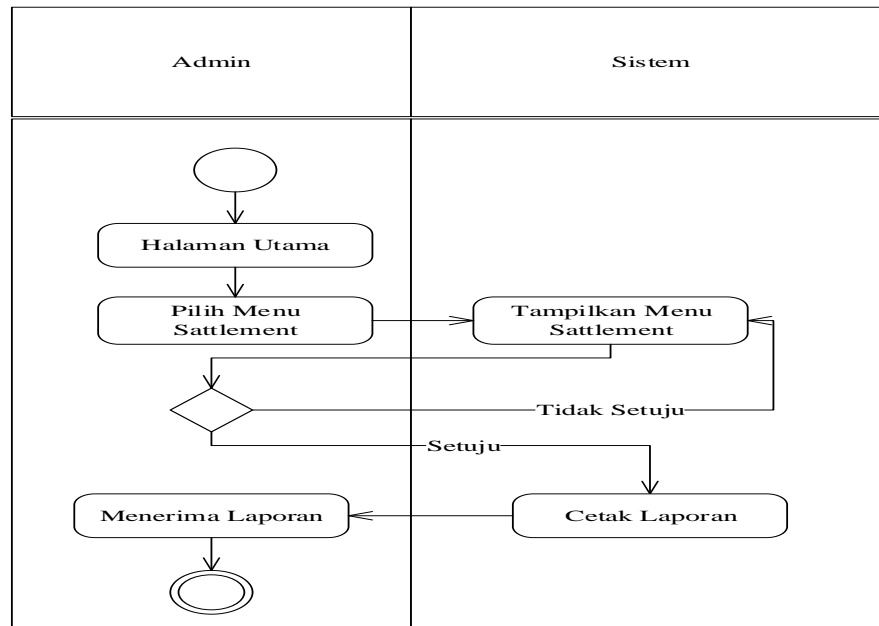
GAMBAR: 3. 21. Activity Diagram Sistem Usulan Hapus Area



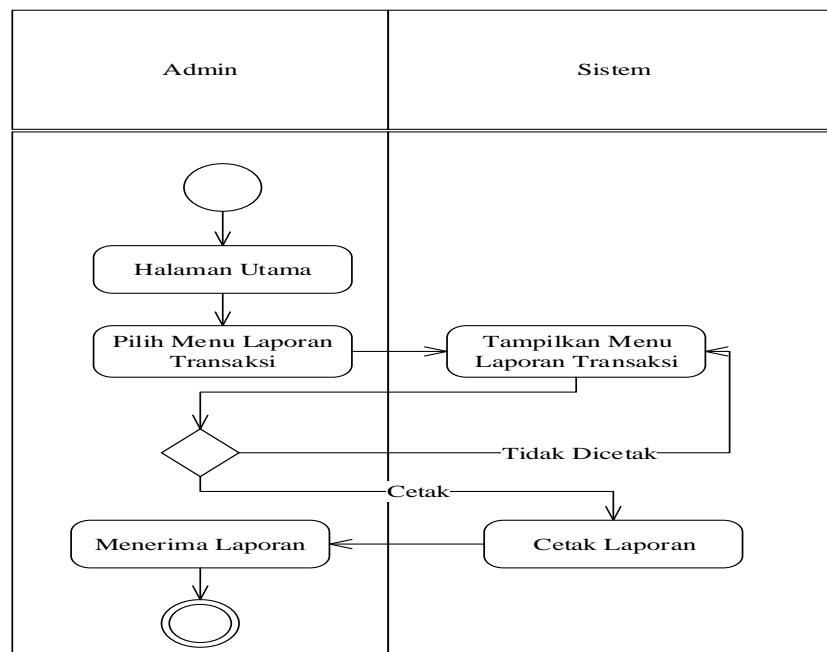
GAMBAR: 3. 22. Activity Diagram Sistem Usulan Hapus Device



GAMBAR: 3. 23. Activity Diagram Sistem Usulan Hapus Tarif



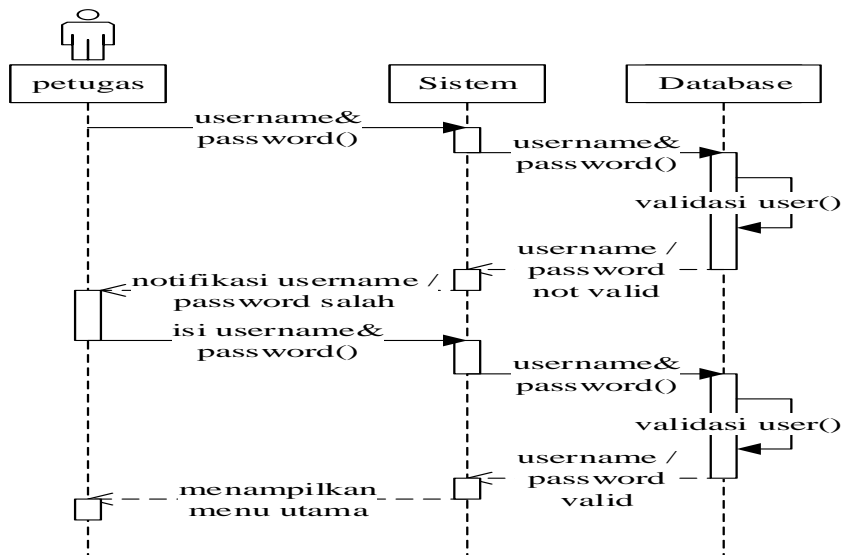
GAMBAR: 3. 24. Activity Diagram Sistem Usulan Sattlement Admin



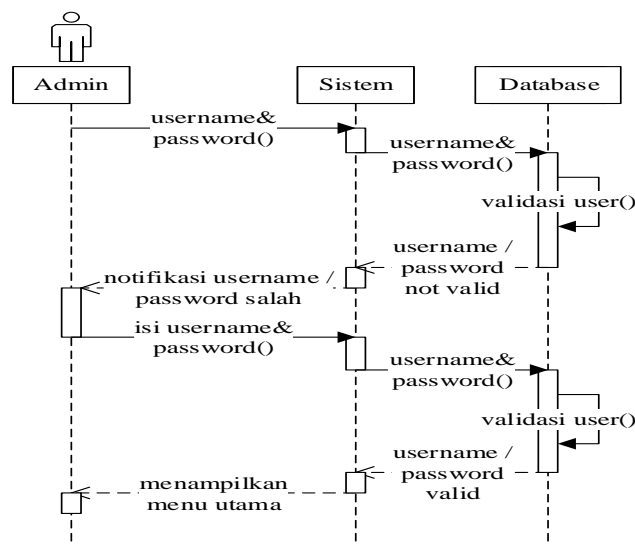
GAMBAR: 3. 25. Activity Diagram Sistem Usulan Laporan Transaksi

3.2.7. Sequence Diagram

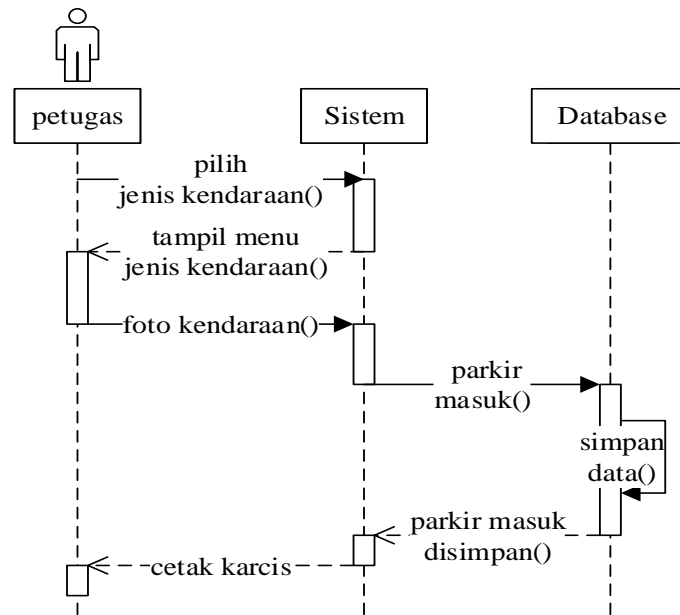
Sequence diagram menjelaskan urutan proses yang dilakukan sistem untuk mencapai tujuan *Use Case*.



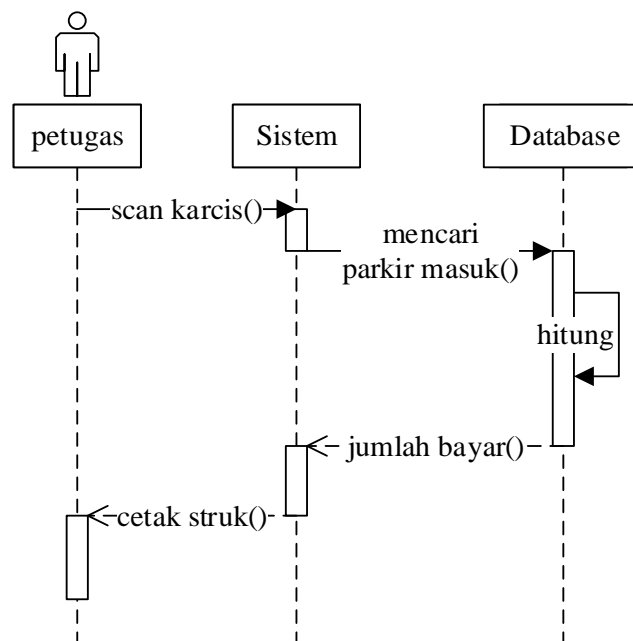
GAMBAR: 3. 26. Sequence Diagram Login Petugas



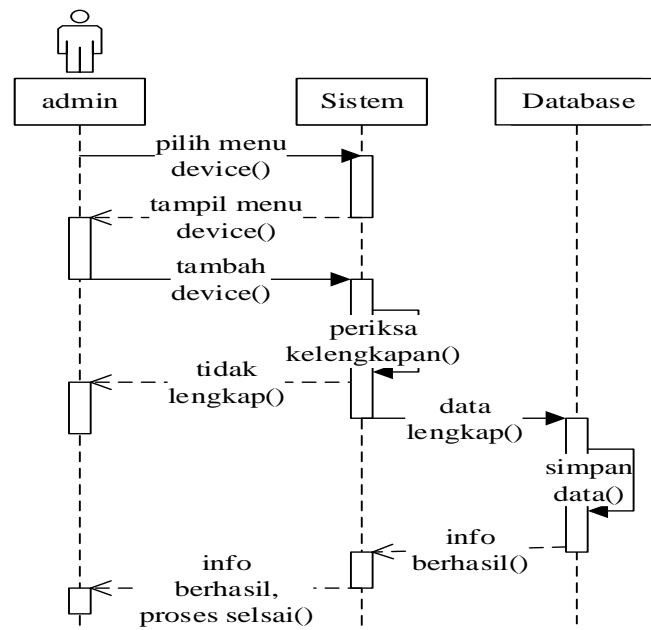
GAMBAR: 3. 27. Sequence Diagram Login Admin



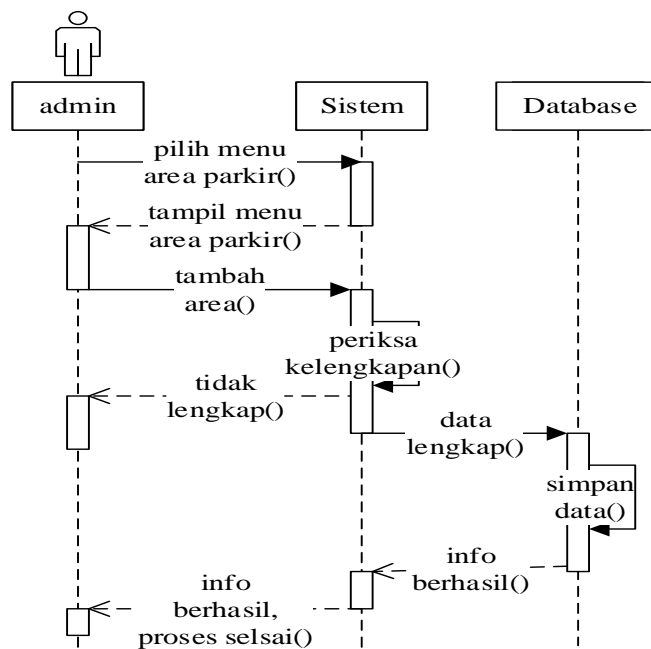
GAMBAR: 3. 28. *Sequence* Diagram Parkir Masuk



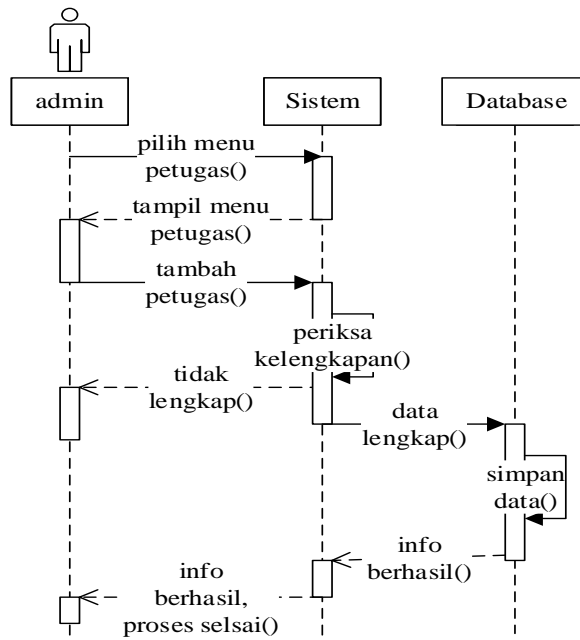
GAMBAR: 3. 29. *Sequence* Diagram Parkir Keluar



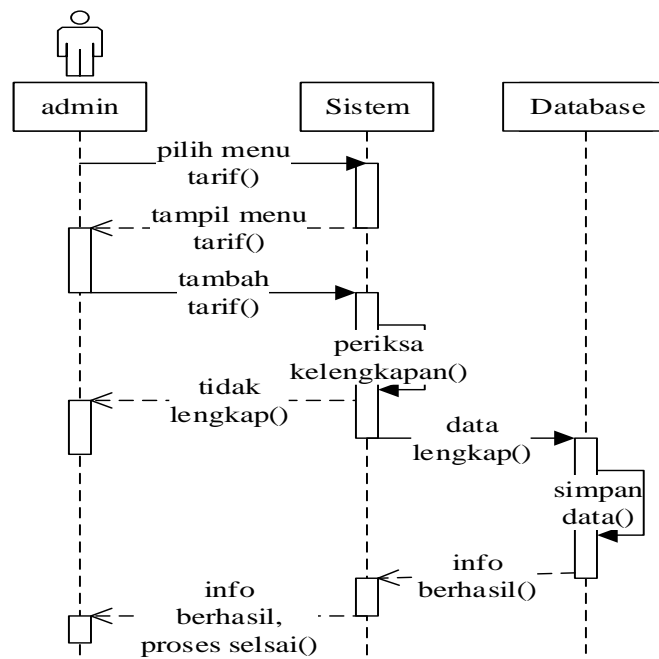
GAMBAR: 3. 30. *Sequence Diagram Tambah Device*



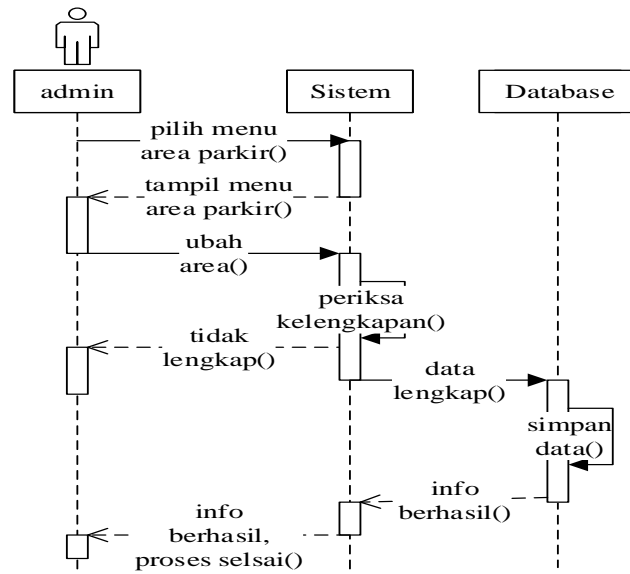
GAMBAR: 3. 31. *Sequence Diagram Tambah Area Parkir*



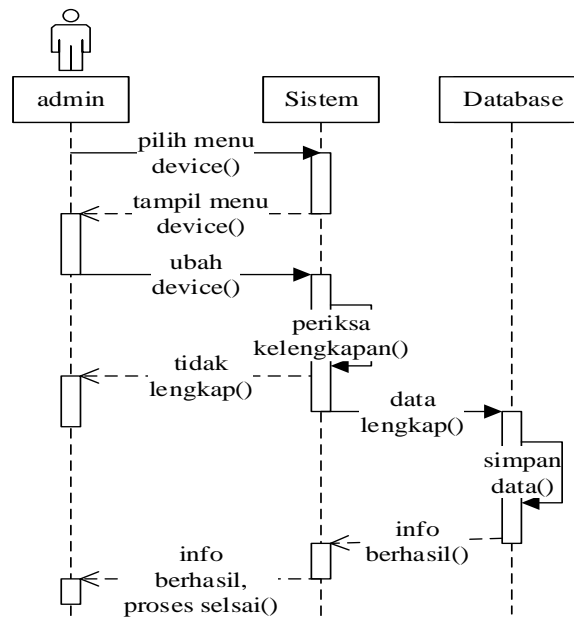
GAMBAR: 3. 32. *Sequence Diagram* Tambah Petugas



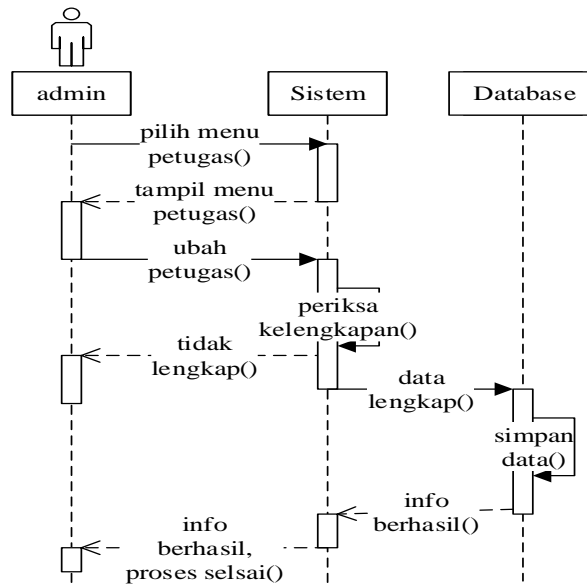
GAMBAR: 3. 33. *Sequence Diagram* Tambah Tarif



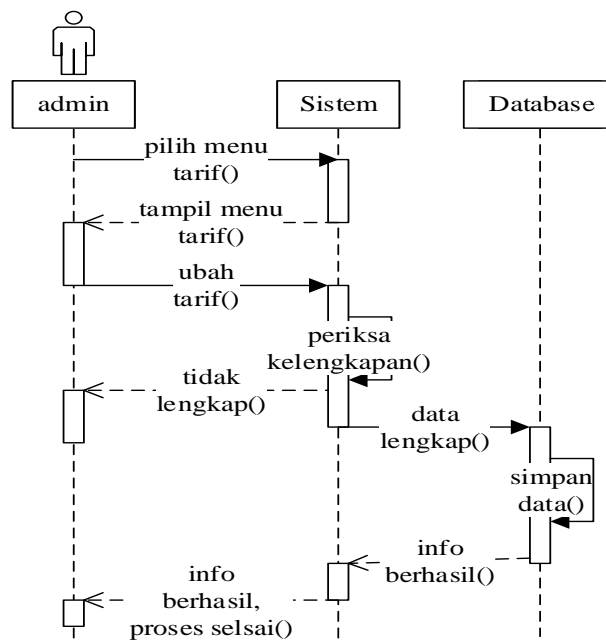
GAMBAR: 3. 34. *Sequence Diagram Ubah Area Parkir*



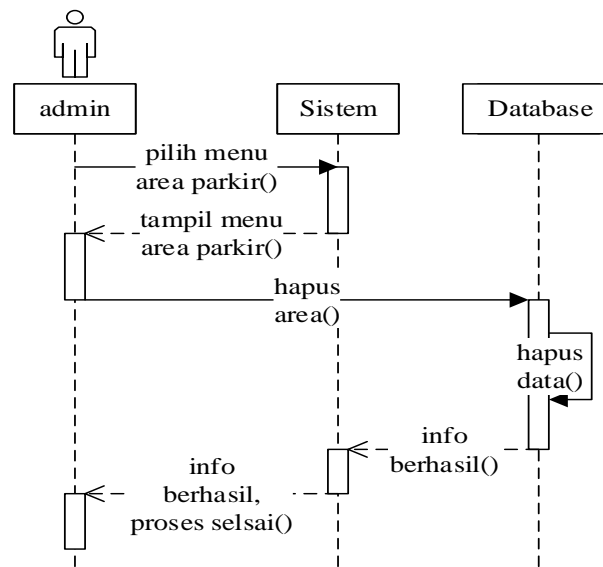
GAMBAR: 3. 35. *Sequence Diagram Ubah Device*



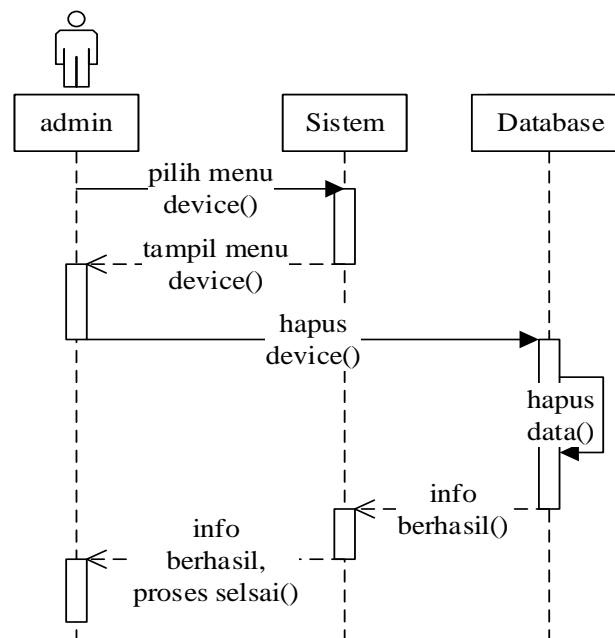
GAMBAR: 3. 36. *Sequence* Diagram Ubah Petugas



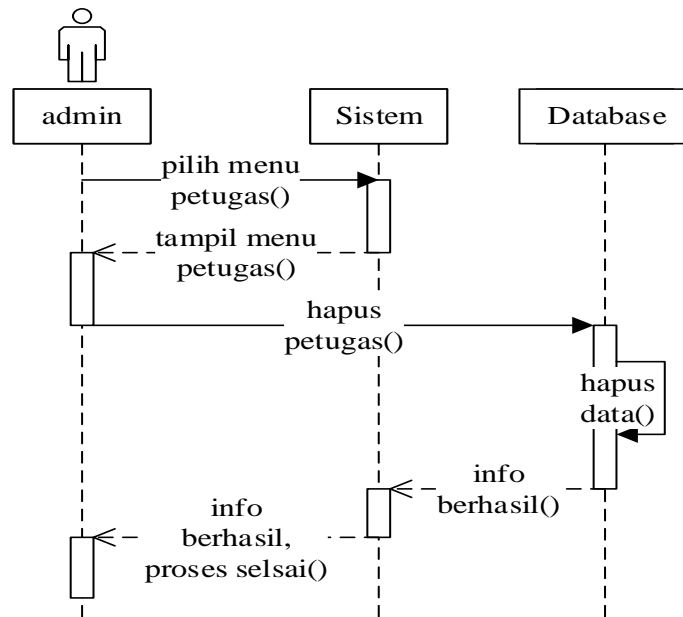
GAMBAR: 3. 37. *Sequence* Diagram Ubah Tarif



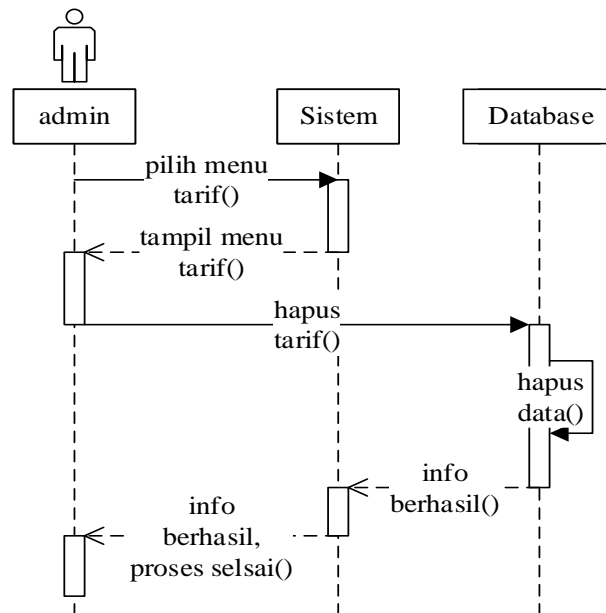
GAMBAR: 3. 38. *Sequence* Diagram Hapus Area Parkir



GAMBAR: 3. 39. *Sequence* Diagram Hapus Device



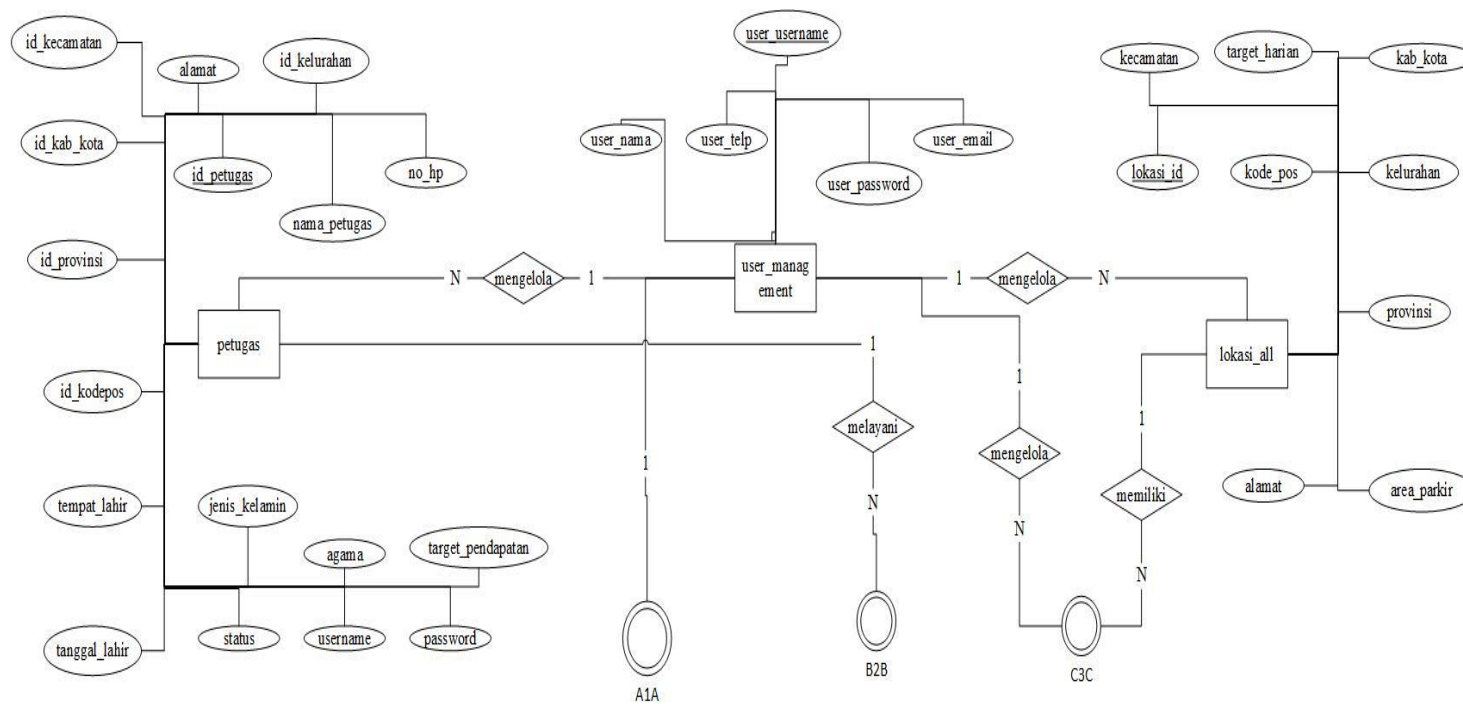
GAMBAR: 3. 40. *Sequence Diagram Hapus Petugas*

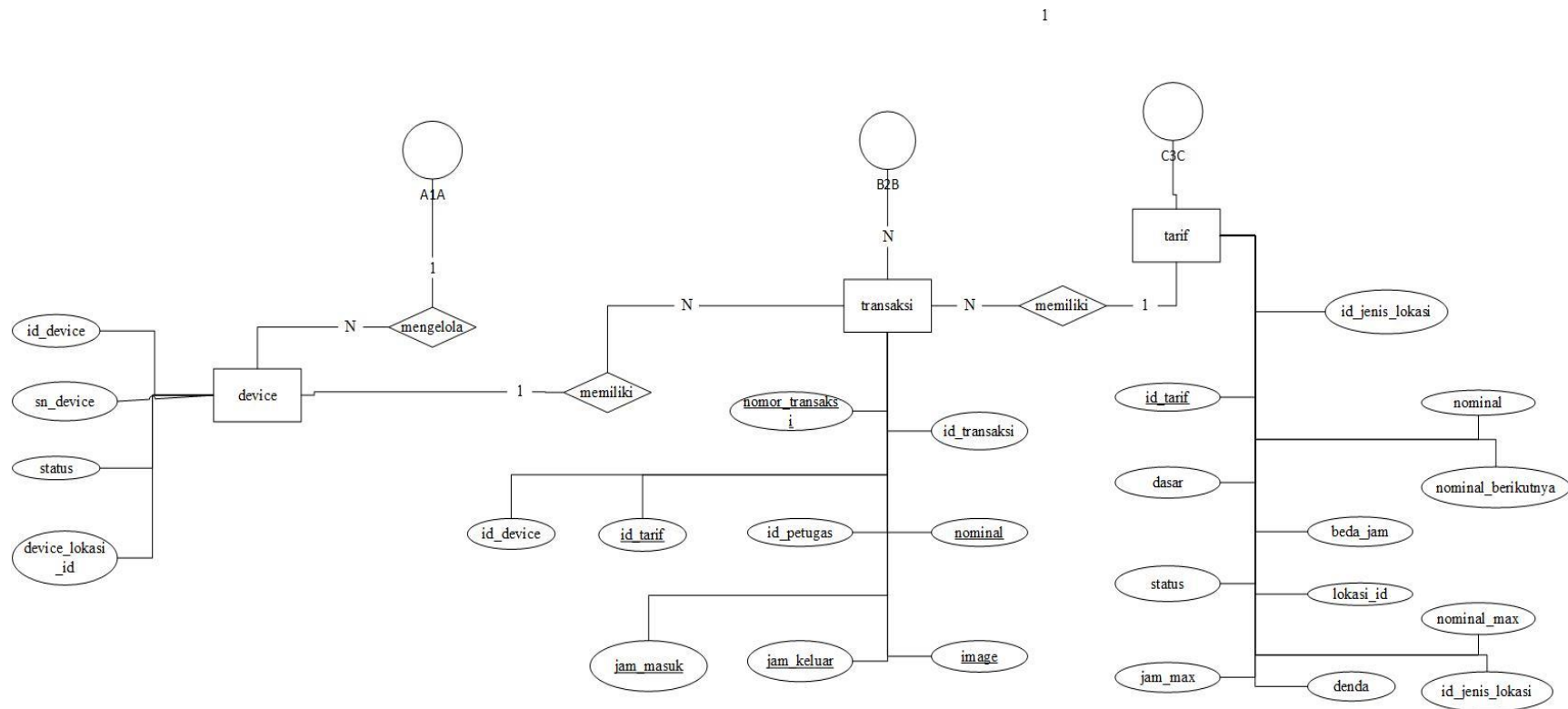


GAMBAR: 3. 41. *Sequence Diagram Hapus Tarif*

3.2.8. Entity Relationship Diagram

Berikut ini *Entity Relationship* diagram sistem usulan:

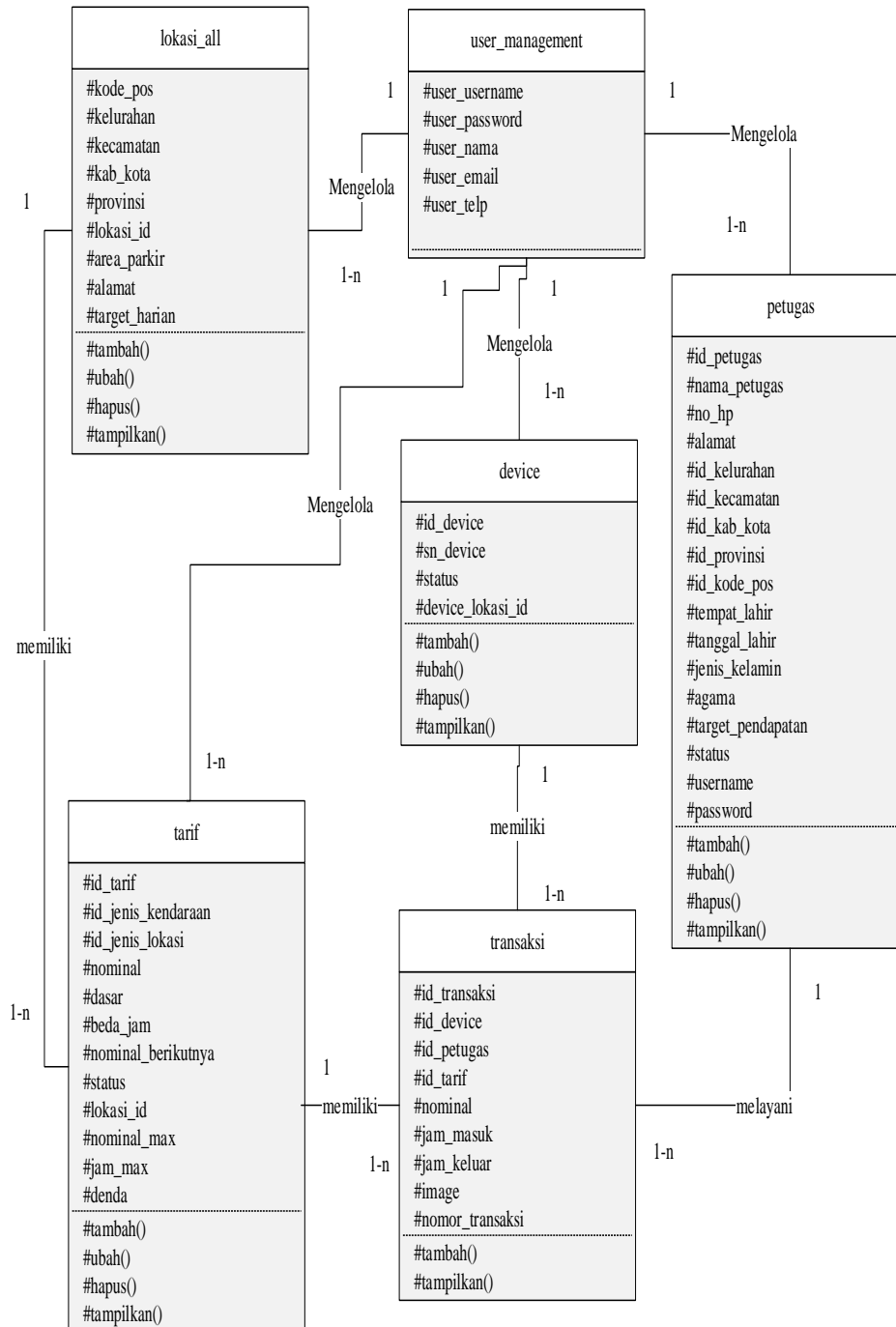




GAMBAR: 3. 42. *Entity Relationship* Diagram Sistem Usulan

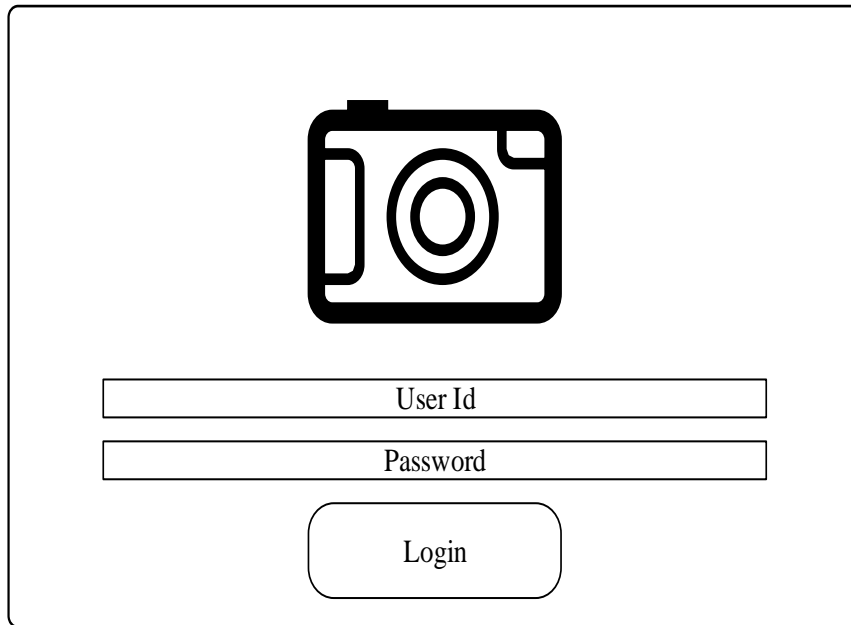
3.2.9. Class Diagram

Berikut ini *Class diagram* sistem usulan :



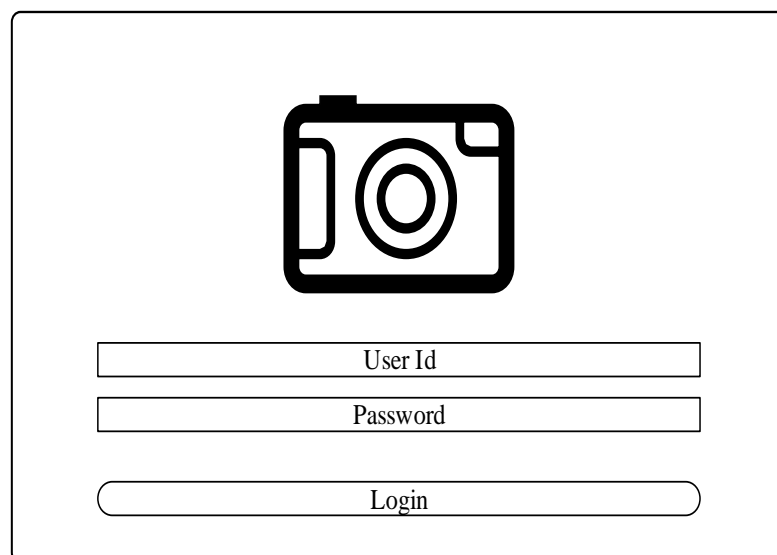
GAMBAR: 3. 43. *Class Diagram* Sistem Usulan

3.2.10. Rancangan Tampilan Sistem



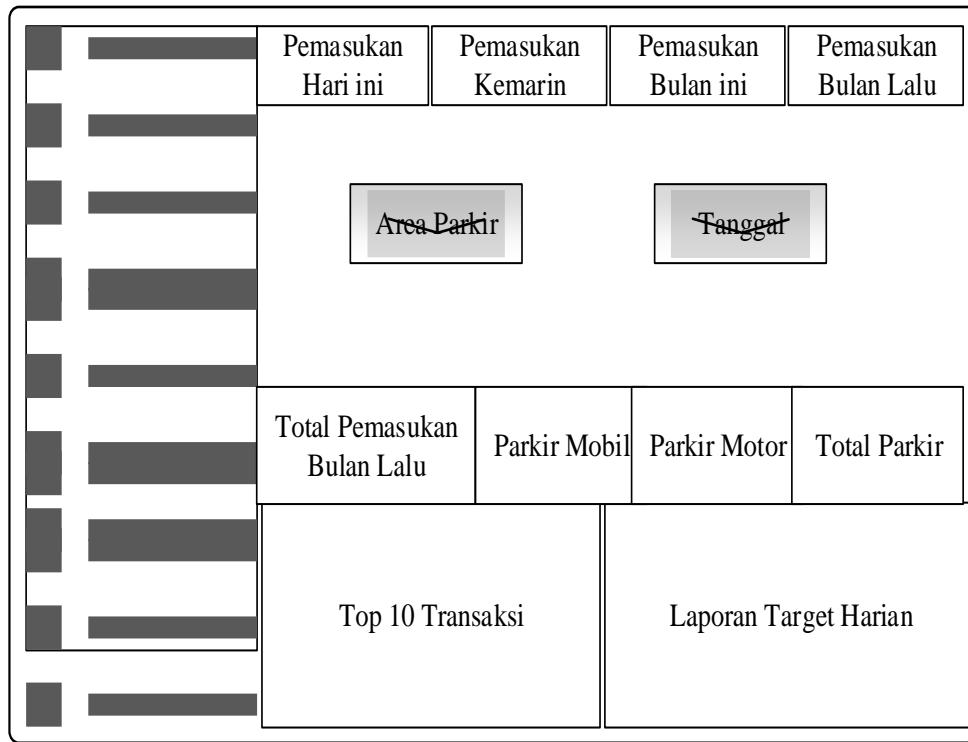
A wireframe diagram of a web login form. At the top center is a camera icon. Below it are two horizontal input fields: the first is labeled "User Id" and the second is labeled "Password". Below the password field is a rounded rectangular button labeled "Login".

GAMBAR: 3. 44. Rancangan Tampilan Login Pada Web

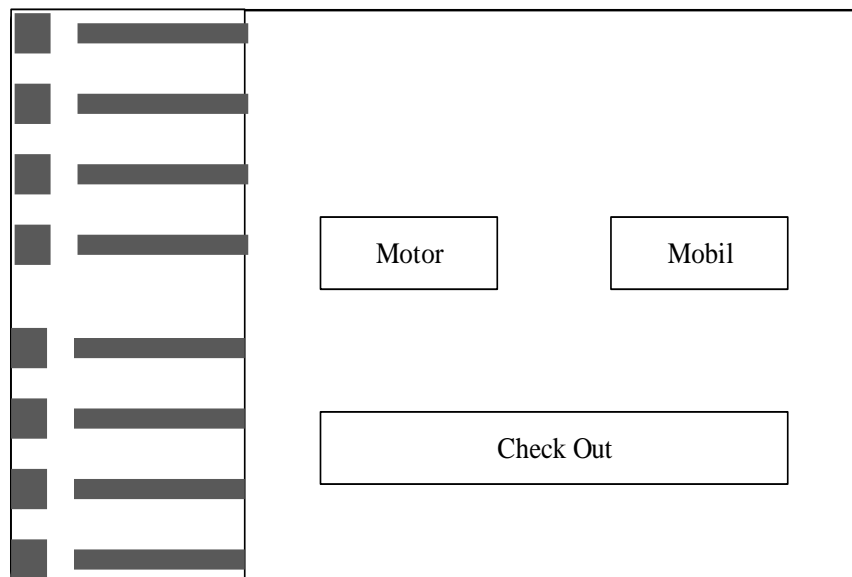


A wireframe diagram of a POS login form. At the top center is a camera icon. Below it are three horizontal input fields: the first is labeled "User Id", the second is labeled "Password", and the third is a rounded rectangular button labeled "Login".

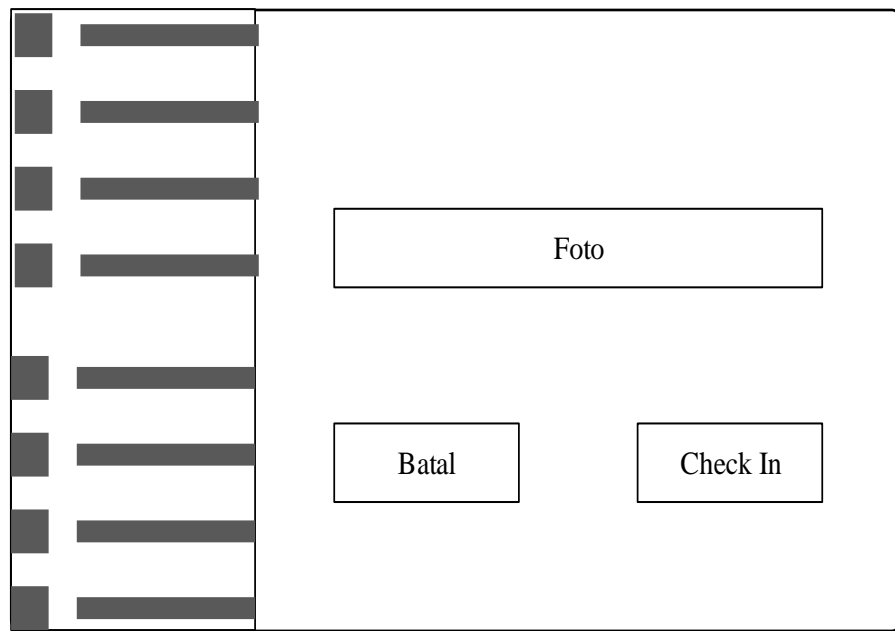
GAMBAR: 3. 45. Rancangan Tampilan *Login* Pada POS



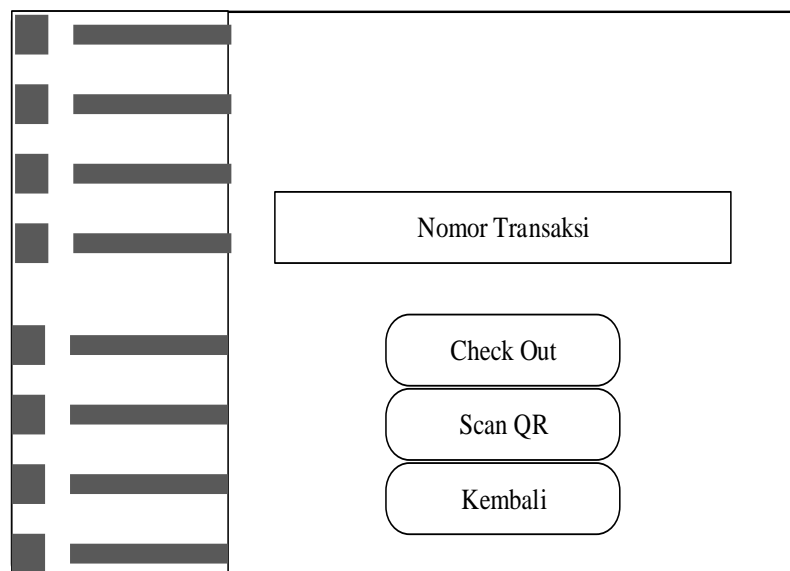
GAMBAR: 3. 46. Rancangan Tampilan Utama Pada Web



GAMBAR: 3. 47. Rancangan Tampilan Utama Pada POS












GAMBAR: 3. 48. Rancangan Tampilan Parkir Masuk



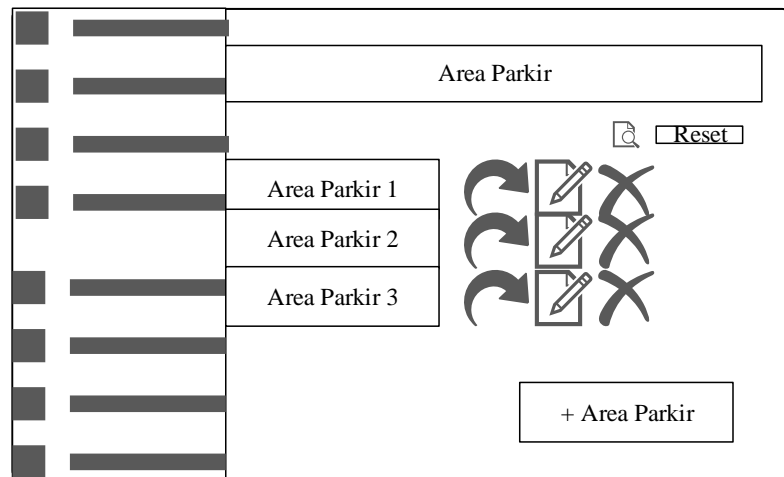
GAMBAR: 3. 49. Rancangan Tampilan Parkir Keluar

	Motor	Mobil
	Tanggal	
	Dari Jam	
	Sampai Jam	
	Batal	Ya

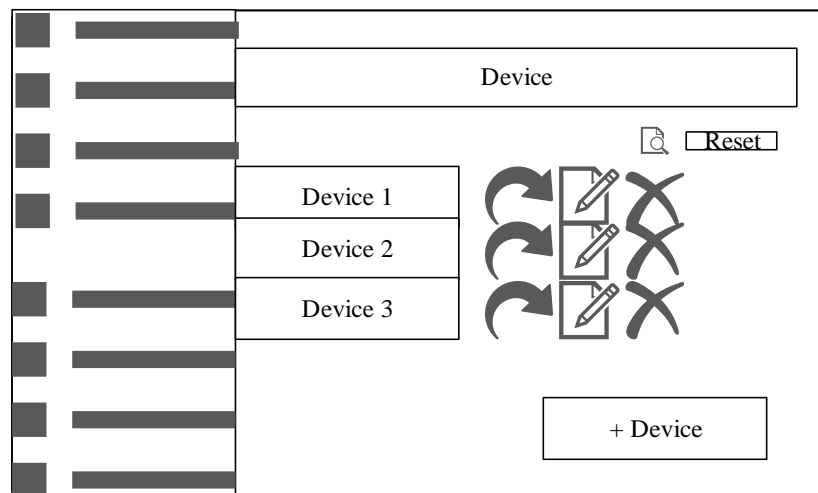
GAMBAR: 3. 50. Rancangan Tampilan Cari Tiket

	Tgl Registrasi	Nama Petugas	No Ktp	Status
				<input type="button" value="Reset"/>
	Petugas 1			
	Petugas 2			
	Petugas 3			
				<input type="button" value="+ Petugas"/>










GAMBAR: 3. 51. Rancangan Tampilan Menu Petugas



GAMBAR: 3. 52. Rancangan Tampilan Menu Area Parkir



GAMBAR: 3. 53. Rancangan Tampilan Menu *Device*

Tarif	
	<input type="text" value="Reset"/>
Tarif 1	  
Tarif 2	  
Tarif 3	  
<input type="button" value="+ Tarif"/>	

GAMBAR: 3. 54. Rancangan Tampilan Menu Tarif

Form Tambah Petugas	
Officer Id :	<input type="text"/>
Nama :	<input type="text"/>
Alamat :	<input type="text"/>
<input type="button" value="Simpan"/>	<input type="button" value="Reset"/> <input type="button" value="Batal"/>

GAMBAR: 3. 55. Rancangan Tampilan Tambah Petugas

		Form Tambah Area Parkir
		Lokasi Id :
		Area Parkir:
		Alamat :
		<input type="button" value="Simpan"/> <input type="button" value="Reset"/> <input type="button" value="Batal"/>

GAMBAR: 3. 56. Rancangan Tampilan Tambah Area Parkir

		Form Tambah Device
		Device id :
		Area Parkir
		Serial Number :
		<input type="button" value="Simpan"/> <input type="button" value="Reset"/> <input type="button" value="Batal"/>

GAMBAR: 3. 57. Rancangan Tampilan Tambah *Device*

		Form Tambah Tarif
		Id tarif :
		Jenis Kendaraan :
		Tarif :
		<input type="button" value="Simpan"/> <input type="button" value="Reset"/> <input type="button" value="Batal"/>

GAMBAR: 3. 58. Rancangan Tampilan Tambah Tarif

BAB IV

IMPLEMENTASI DAN UJI COBA

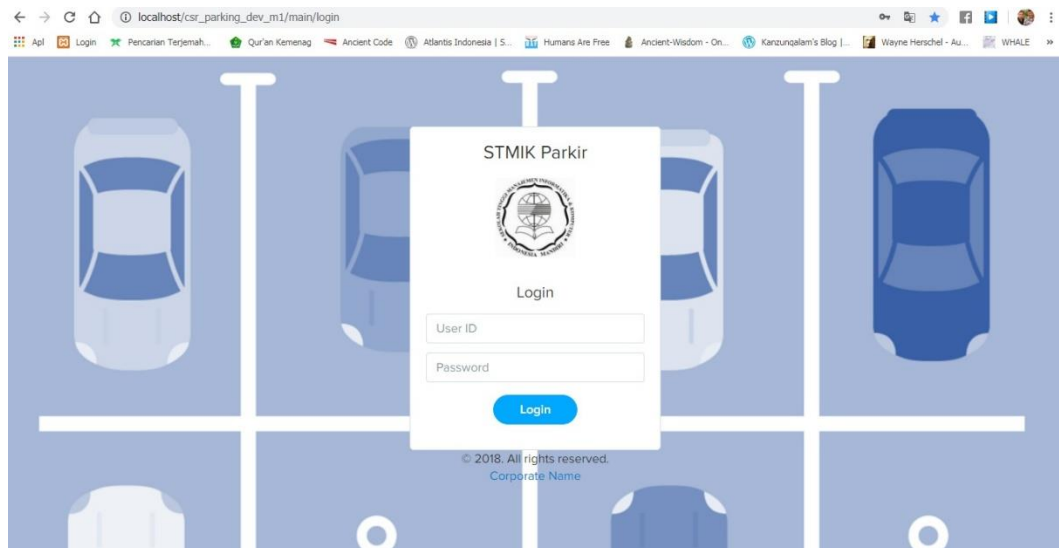
Bab ini akan membahas mengenai implementasi dari rancangan perangkat lunak yang telah dibuat pada bab sebelumnya, dan merupakan tahap selanjutnya pada *waterfall* model. Berikut ini perangkat lunak tambahan dan perangkat keras yang digunakan :

TABEL: 4. 1. Perangkat Lunak Tambahan Dan Perangkat Keras

Perangkat Lunak Tambahan	Perangkat Keras
Node JS	Server atau workstation : Intel Xeon E3 RAM 16 gb SSD 500 gb Unmetered Bandwith 1Gbps
Navicat Premium	
Apache HTTP Server	Sunmi V1S
Visual Studio Code	

4.1. Implementasi Tampilan

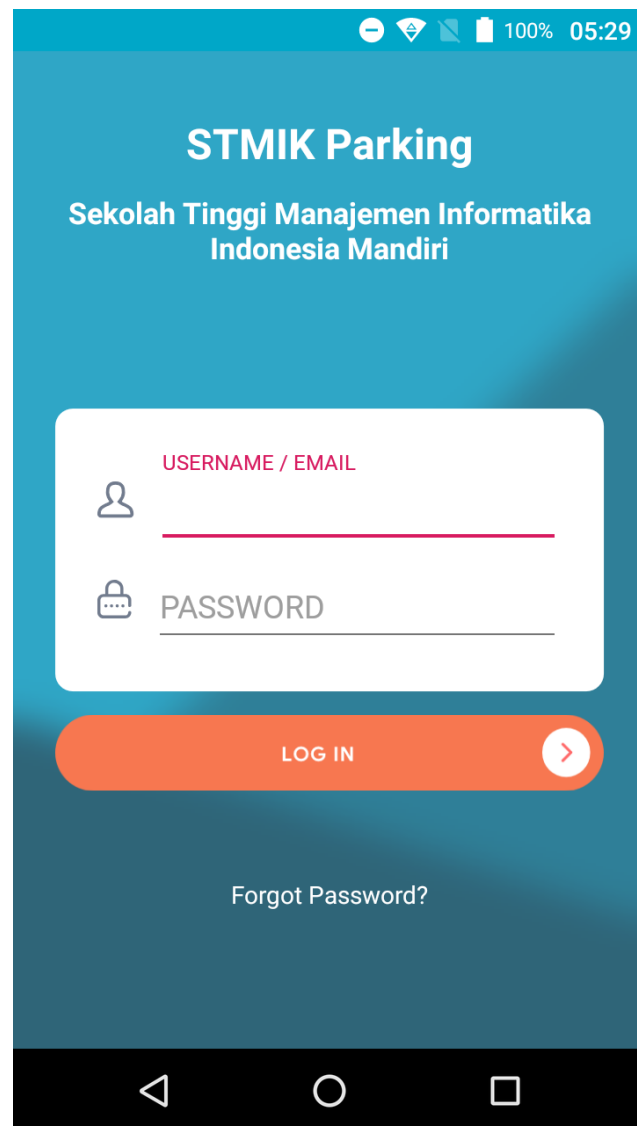
4.1.1. Tampilan Login



GAMBAR: 4. 1. Tampilan *Login* Web

Prosedur untuk login, sebagai berikut:

- A. Isi terlebih dahulu username dan password.
- B. Klik tombol Login.
- C. Apabila username dan password sesuai, maka akan masuk menu utama.

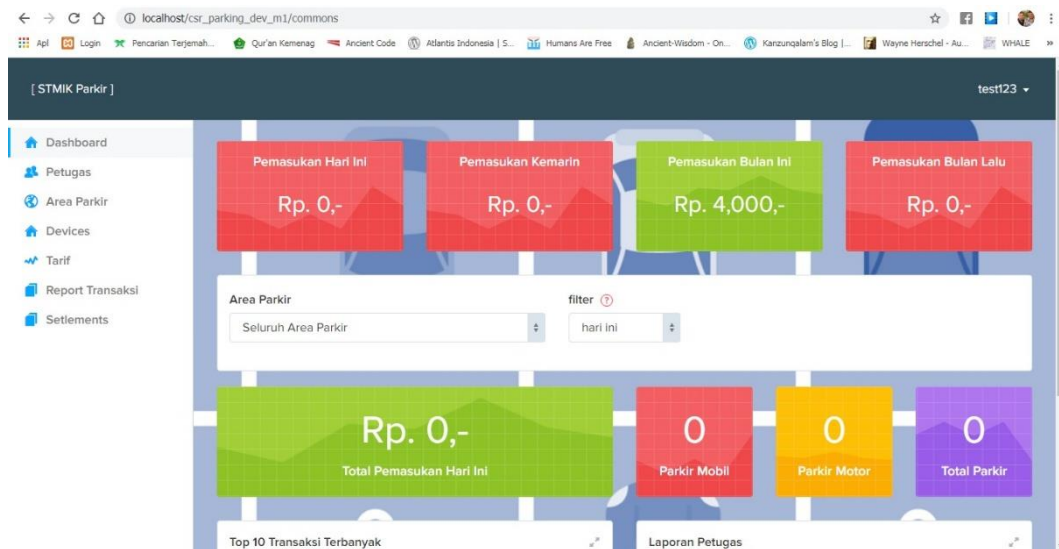


GAMBAR: 4. 2. Tampilan *Login* Android

Prosedur untuk login, sebagai berikut:

- A. Isi terlebih dahulu username dan password.
- B. Klik tombol Login.
- C. Apabila username dan password sesuai, maka akan masuk menu utama.

4.1.2. Tampilan Menu Utama



GAMBAR: 4. 3. Tampilan Menu Utama Web

Keterangan menu sebagai berikut:

- A. Menu petugas, menu yang berfungsi untuk mengelola petugas.
- B. Menu area parkir, menu yang berfungsi untuk mengelola area parkir.
- C. Menu device, menu yang berfungsi untuk mengelola device.
- D. Menu Tarif, menu yang berfungsi untuk mengelola tarif.
- E. Menu report transaksi, menu untuk melihat jumlah pemasukan.
- F. Menu settlement, menu untuk mengelola setoran dari petugas parkir.

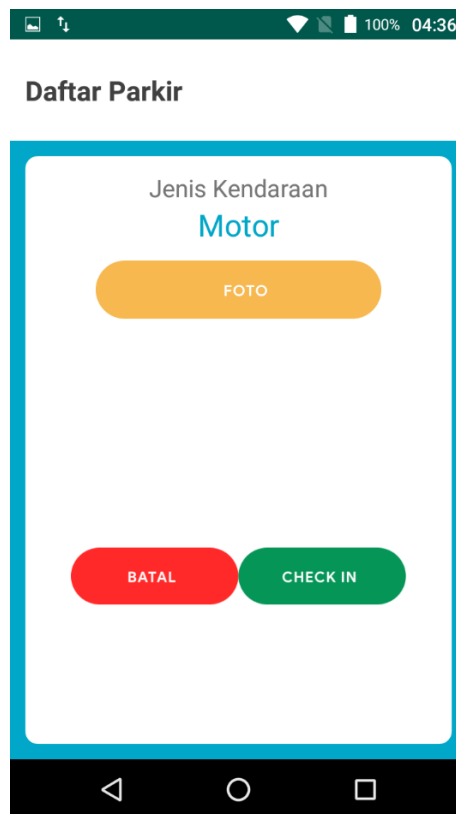


GAMBAR: 4. 4. Tampilan Menu Utama Android

Keterangan sebagai berikut:

- A. Menu motor, menu untuk parkir masuk kendaraan motor.
- B. Menu mobil, menu untuk parkir masuk kendaraan mobil.
- C. Menu *check out*, menu untuk parkir keluar.

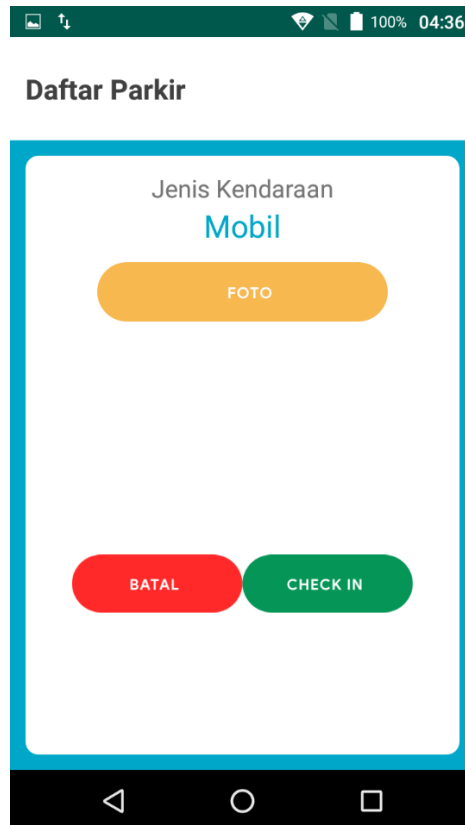
4.1.3. Tampilan Menu Parkir Masuk



GAMBAR: 4. 5. Tampilan Parkir Masuk Motor

Prosedur parkir masuk motor:

- A. Klik tombol foto untuk memfoto nomor plat motor.
- B. Lalu klik tombol *check in*.
- C. Jika ingin membatalkan klik tombol batal.

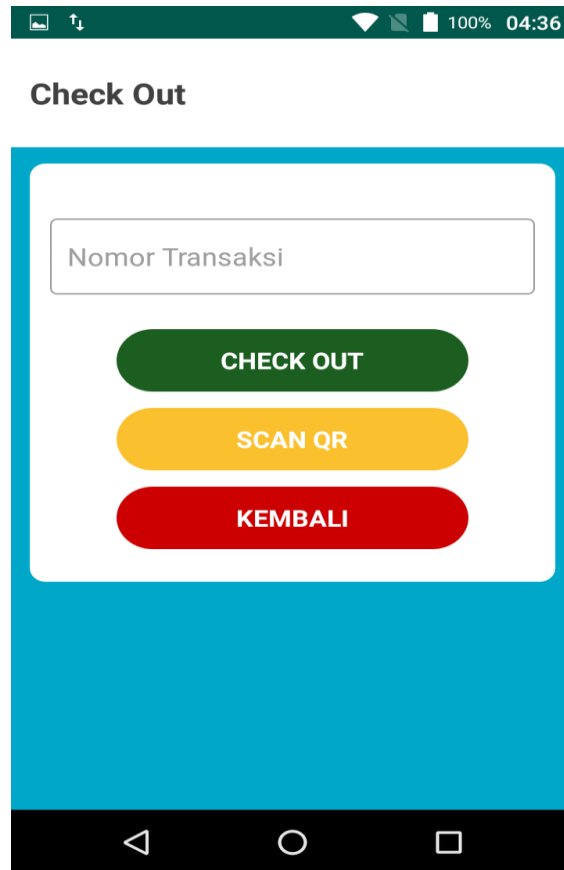


GAMBAR: 4. 6.Tampilan Parkir Masuk Mobil

Prosedur parkir masuk mobil:

- A. Klik tombol foto untuk memfoto nomor plat mobil.
- B. Lalu klik tombol *check in*.
- C. Jika ingin membatalkan klik tombol batal.

4.1.4. Tampilan Menu Parkir Keluar



GAMBAR: 4. 7. Tampilan Parkir Keluar

Prosedur parkir keluar:

- A. Isi nomor transaksi lalu klik tombol *check out*.
- B. Atau klik tombol scan QR.
- C. Jika ingin membatalkan klik tombol kembali.

4.1.5. Tampilan Menu Cari Karcis

The screenshot shows a mobile application interface for searching parking tickets. At the top, there is a status bar with icons for signal, Wi-Fi, battery (100%), and time (04:35). Below the status bar, the title 'Pencarian' is displayed. The main content area is enclosed in a blue border and contains a search form. At the top of the form, there are two tabs: 'MOTOR' (selected) and 'MOBIL'. Below the tabs, there is a text input field labeled 'Tanggal'. Under the heading 'Perkiraan Jam', there are two more text input fields: 'Dari Jam' and 'Sampai Jam'. At the bottom of the form, there are two buttons: a red 'BATAL' button and a green 'YA' button. The entire form is enclosed in a blue border.

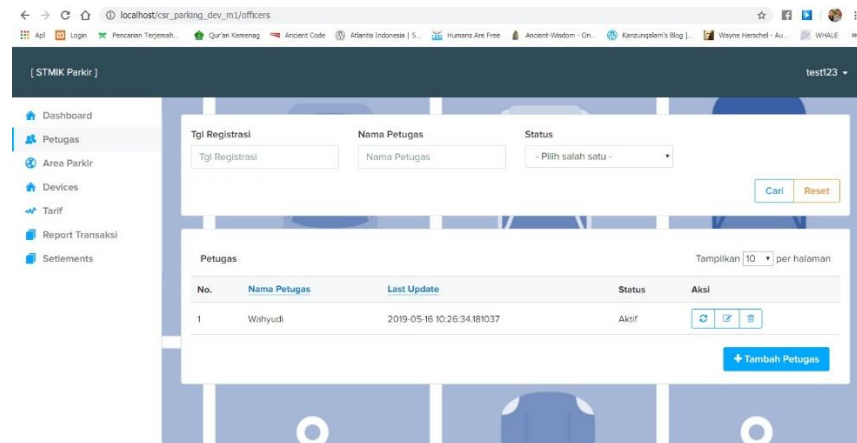
GAMBAR: 4. 8. Tampilan Cari Karcis

Prosedur cari karcis:

- A. Tentukan jenis kendaraan.
- B. Masukkan tanggal parkir, perkiraan awal parkir .
- C. Lalu klik tombol ya.
- D. Jika ingin membatalkan klik tombol batal.

4.1.6. Tampilan Menu Kelola Petugas

A. Tampilan Menu Utama Kelola Petugas



GAMBAR: 4. 9. Tampilan Menu Utama Kelola Petugas

Keterangan sebagai berikut:

- A. Untuk tambah petugas klik tombol tambah petugas.
- B. Untuk mengubah petugas, pilih nama petugas atau mencarinya terlebih dahulu lalu klik tombol ubah .
- C. Untuk menghapus petugas, pilih nama petugas atau mencarinya terlebih dahulu lalu klik tombol hapus.

B. Tampilan Menu Tambah Petugas

The screenshot shows a web browser window displaying the 'Form Tambah Petugas' (Add Officer Form) in a system named 'STMIK Parkir'. The browser address bar shows 'localhost:csr_parking_dev_m1/officers'. The form contains the following fields:

- Officer ID: 0015
- Nama Petugas: [Text Input]
- Alamat: [Text Input]
- Provinsi: [Dropdown Menu]
- Kota: [Dropdown Menu]
- Kecamatan: [Dropdown Menu]
- Kelurahan: [Dropdown Menu]
- Kode Pos: [Text Input]
- Tempat Lahir: [Text Input]
- Tgl Lahir: [Text Input]
- Jenis Kelamin: [Dropdown Menu]
- Agama: [Dropdown Menu]
- No Hn: [Text Input]

The left sidebar menu includes: Dashboard, Petugas, Area Parkir, Devices, Tarif, Report Transaksi, and Settlements. The top right corner of the page indicates the user is 'test123'.

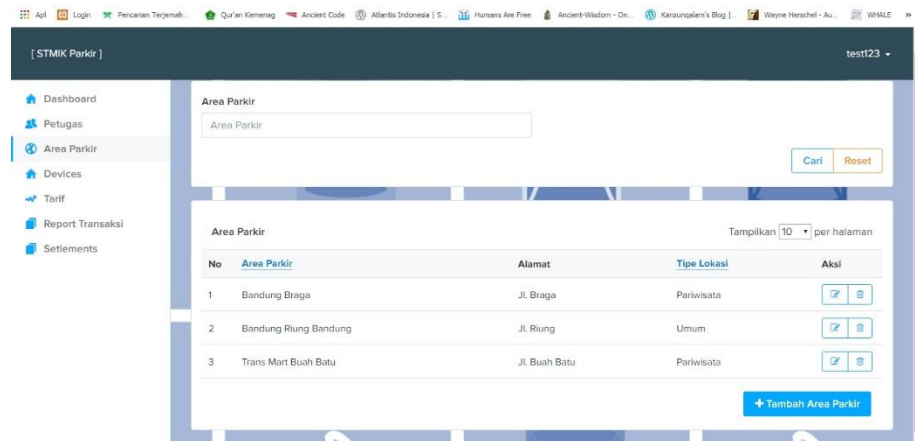
GAMBAR: 4. 10. Tampilan Menu Tambah Petugas

Prosedur tambah petugas sebagai berikut:

- A. Isi form yang telah disediakan lalu klik tombol simpan.
- B. Untuk membatalkan tambah petugas klik tombol batal.

4.1.7. Tampilan Menu Kelola Area Parkir

A. Tampilan Menu Utama Kelola Area Parkir

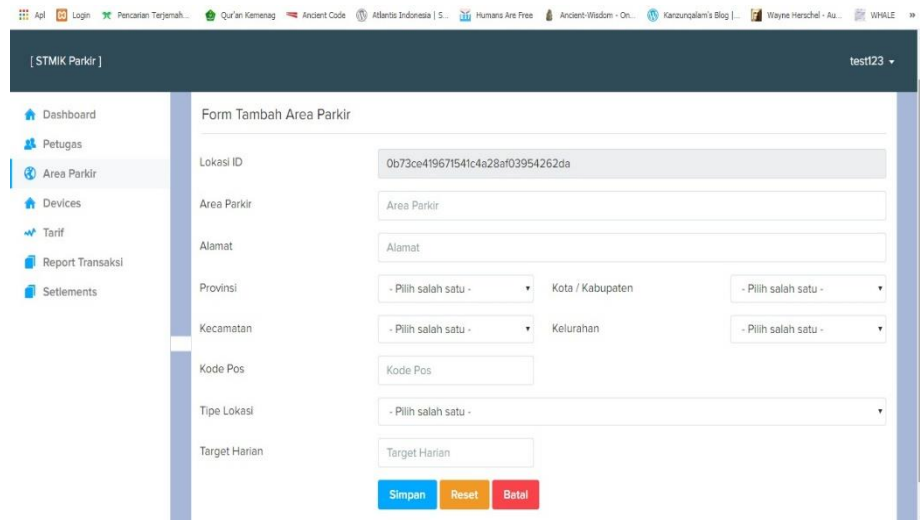


GAMBAR: 4. 11. Tampilan Menu Utama Kelola Area Parkir

Keterangan sebagai berikut:

- A. Untuk tambah petugas klik tombol tambah area parkir.
- B. Untuk mengubah area parkir, pilih area parkir atau mencarinya terlebih dahulu lalu klik tombol ubah .
- C. Untuk menghapus area parkir, pilih area parkir atau mencarinya terlebih dahulu lalu klik tombol hapus.

B. Tampilan Menu Tambah Area Parkir



The screenshot shows a web application interface for adding a parking area. The page title is "Form Tambah Area Parkir". On the left, there is a sidebar menu with the following items: Dashboard, Petugas, Area Parkir (highlighted), Devices, Tarif, Report Transaksi, and Settlements. The main content area contains the following form fields:

- Lokasi ID: 0b73ce419671541c4a28af03954262da
- Area Parkir: Area Parkir
- Alamat: Alamat
- Provinsi: - Pilih salah satu -
- Kota / Kabupaten: - Pilih salah satu -
- Kecamatan: - Pilih salah satu -
- Kelurahan: - Pilih salah satu -
- Kode Pos: Kode Pos
- Tipe Lokasi: - Pilih salah satu -
- Target Harian: Target Harian

At the bottom of the form, there are three buttons: "Simpan" (blue), "Reset" (orange), and "Batal" (red).

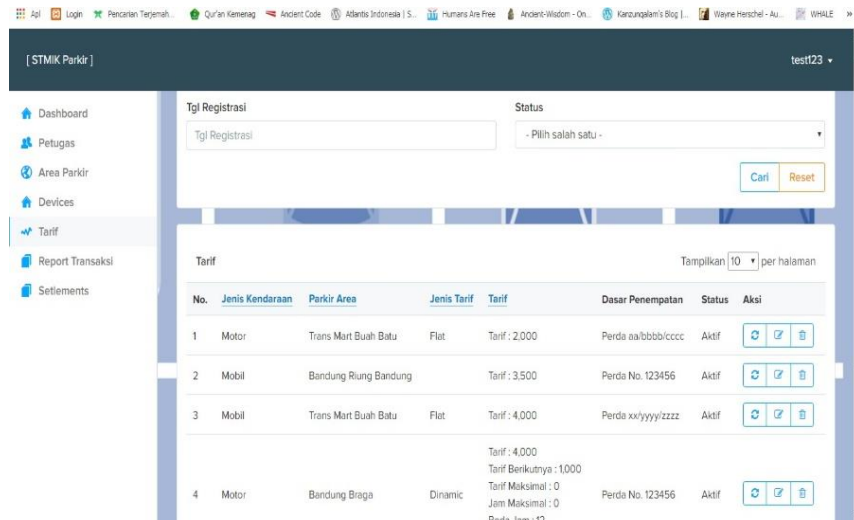
GAMBAR: 4. 12. Tampilan Menu Tambah Area Parkir

Prosedur tambah area parkir sebagai berikut:

- A. Isi form yang telah disediakan lalu klik tombol simpan.
- B. Untuk membatalkan tambah area klik tombol batal.

4.1.8. Tampilan Menu Kelola Tarif

A. Tampilan Menu Utama Kelola Tarif

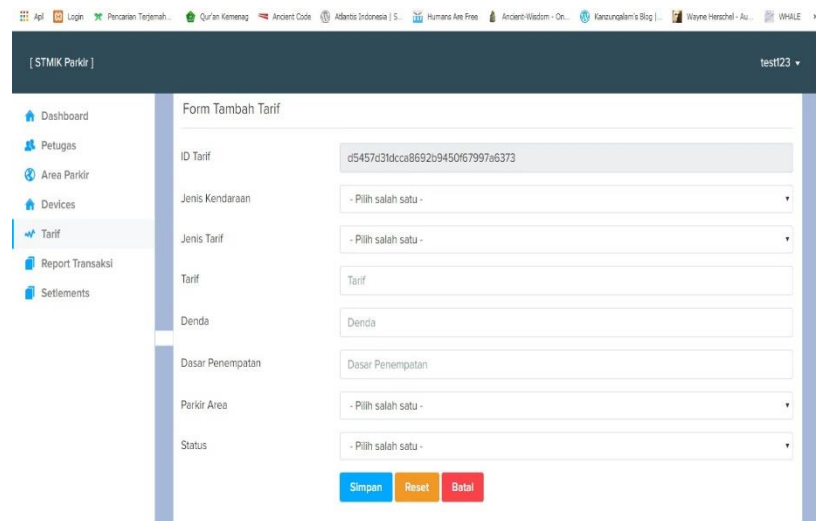


GAMBAR: 4. 13. Tampilan Menu Utama Kelola Tarif

Keterangan sebagai berikut:

- A. Untuk tambah tarif klik tombol tambah tarif.
- B. Untuk mengubah tarif, pilih tarif atau mencarinya terlebih dahulu lalu klik tombol ubah .
- C. Untuk menghapus tarif, pilih tarif atau mencarinya terlebih dahulu lalu klik tombol hapus.

B. Tampilan Menu Tambah Tarif



The screenshot shows a web application interface for adding a parking rate. The page title is "[STMIK Parkir]" and the user is logged in as "test123". The left sidebar contains a menu with the following items: Dashboard, Petugas, Area Parkir, Devices, Tarif (highlighted), Report Transaksi, and Settlements. The main content area is titled "Form Tambah Tarif" and contains the following fields:

- ID Tarif: d5457d31dcca8692b9450f67997a6373
- Jenis Kendaraan: - Pilih salah satu -
- Jenis Tarif: - Pilih salah satu -
- Tarif: Tarif
- Denda: Denda
- Dasar Penempatan: Dasar Penempatan
- Parkir Area: - Pilih salah satu -
- Status: - Pilih salah satu -

At the bottom of the form, there are three buttons: "Simpan" (blue), "Reset" (orange), and "Batal" (red).

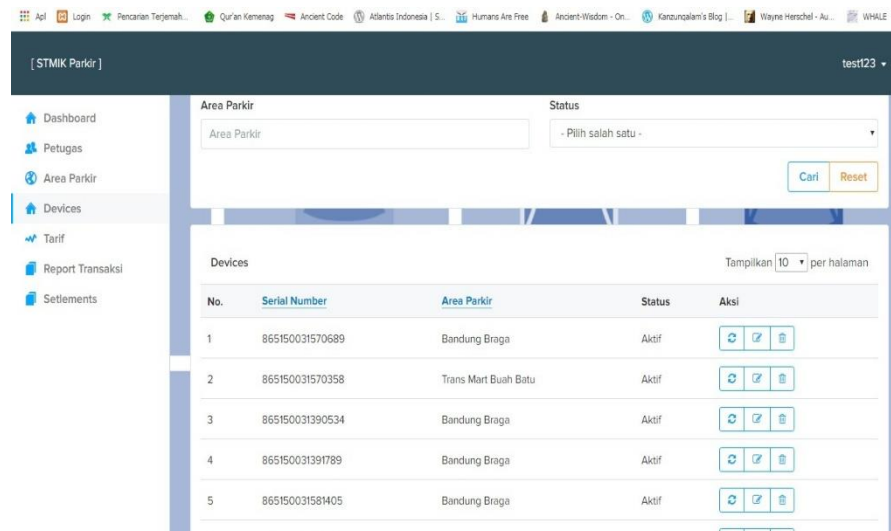
GAMBAR: 4. 14. Tampilan Menu Tambah Tarif

Prosedur tambah tarif sebagai berikut:

- A. Isi form yang telah disediakan lalu klik tombol simpan.
- B. Untuk membatalkan tambah tarif klik tombol batal.

4.1.9. Tampilan Menu Kelola Device

A. Tampilan Menu Utama Kelola Device

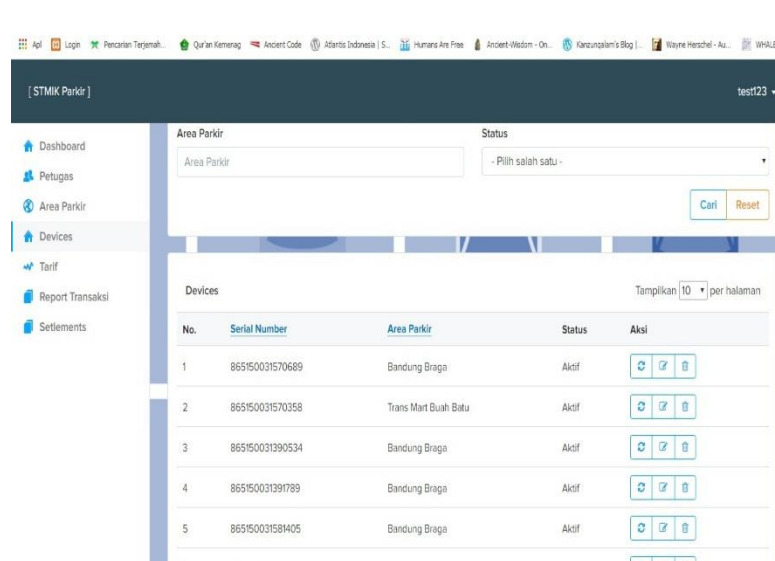


GAMBAR: 4. 15. Tampilan Menu Utama Kelola *Device*

Keterangan sebagai berikut:

- A. Untuk tambah device klik tombol tambah device.
- B. Untuk mengubah device, pilih device atau mencarinya terlebih dahulu lalu klik tombol ubah .
- C. Untuk menghapus device, pilih device atau mencarinya terlebih dahulu lalu klik tombol hapus.

B. Tampilan Menu Tambah Device

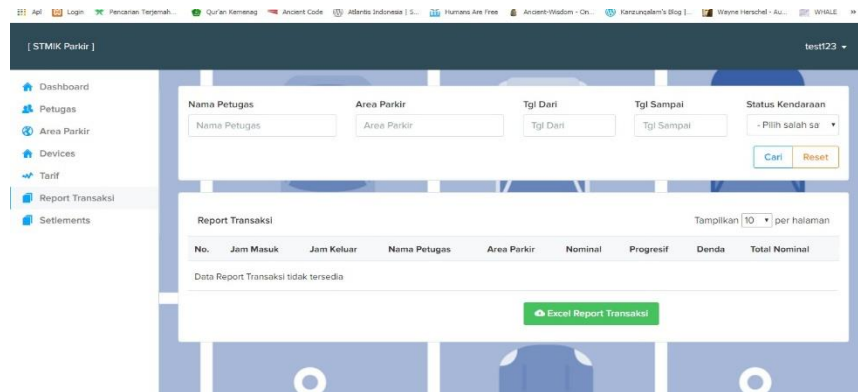


GAMBAR: 4. 16. Tampilan Menu Tambah *Device*

Prosedur tambah tarif sebagai berikut:

- A. Isi form yang telah disediakan lalu klik tombol simpan.
- B. Untuk membatalkan tambah device klik tombol batal.

4.1.10. Tampilan Menu Transaksi

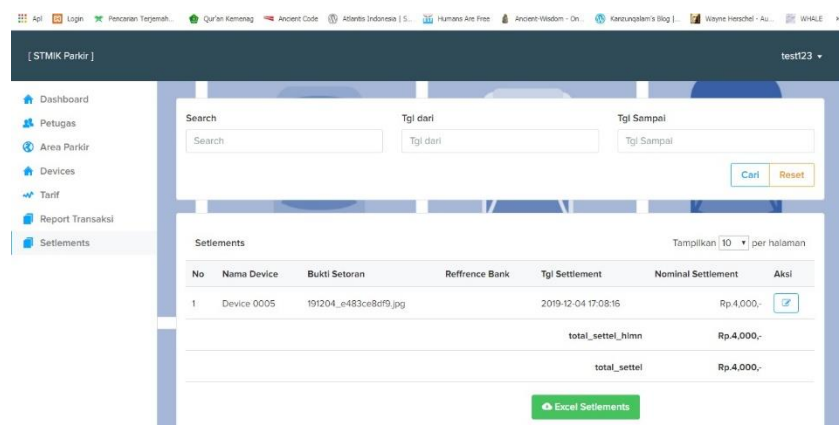


GAMBAR: 4. 17. Tampilan Menu Transaksi

Keterangan sebagai berikut:

- A. Untuk mengunduh transaksi tekan tombol excel report transaksi.

4.1.11. Tampilan Menu Settlement



GAMBAR: 4. 18. Tampilan Menu Settlement

Keterangan sebagai berikut:

- A. Untuk mengunduh settlement tekan tombol excel settlement.

4.2. Implementasi Basis Data

TABEL: 4. 2. Implementasi Basis Data

Nama Tabel	Koding Program
transaksi	<pre>CREATE TABLE "public"."transaksi" ("id_transaksi" varchar(25) COLLATE "pg_catalog"."default" NOT NULL, "id_device" varchar(7) COLLATE "pg_catalog"."default" NOT NULL, "nama_lokasi" text COLLATE "pg_catalog"."default", "id_petugas" varchar(7) COLLATE "pg_catalog"."default" NOT NULL, "id_tarif" varchar(2) COLLATE "pg_catalog"."default" NOT NULL, "nominal" int8, "jam_masuk" timestamp(6) NOT NULL, "jam_keluar" timestamp(6), "metode_bayar" varchar COLLATE "pg_catalog"."default", "image" text COLLATE "pg_catalog"."default", "status" int2, "id_petugas_keluar" varchar(7) COLLATE "pg_catalog"."default", "no_refrensi" varchar(25) COLLATE "pg_catalog"."default", "nomer_transaksi" varchar(16) COLLATE "pg_catalog"."default" NOT NULL, "nominal_tambahan" int8, "total_nominal" int8, "denda" int8);</pre>
device	<pre>CREATE TABLE "public"."device" ("id_device" varchar(32) COLLATE "pg_catalog"."default" NOT NULL,</pre>

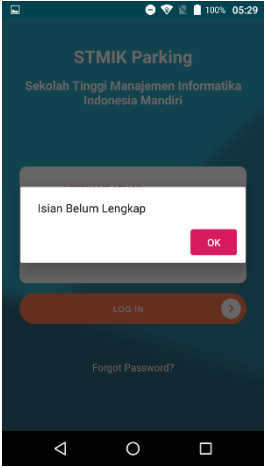
	<pre> "nama_device" varchar(50) COLLATE "pg_catalog"."default", "sn_device" varchar(32) COLLATE "pg_catalog"."default", "alamat_device" text COLLATE "pg_catalog"."default", "id_kelurahan" varchar(32) COLLATE "pg_catalog"."default", "id_kecamatan" varchar(32) COLLATE "pg_catalog"."default", "id_kota_kabupaten" varchar(32) COLLATE "pg_catalog"."default", "id_provinsi" varchar(32) COLLATE "pg_catalog"."default", "id_kode_pos" varchar(64) COLLATE "pg_catalog"."default", "status" varchar(10) COLLATE "pg_catalog"."default", "created_by" varchar(20) COLLATE "pg_catalog"."default", "created_date" timestamp(6), "updated_by" varchar(20) COLLATE "pg_catalog"."default", "updated_date" timestamp(6), "device_lokasi_id" varchar(36) COLLATE "pg_catalog"."default"); </pre>
petugas	<pre> CREATE TABLE "public"."petugas" ("id_petugas" varchar(32) COLLATE "pg_catalog"."default" NOT NULL, "nama_petugas" varchar(200) COLLATE "pg_catalog"."default", "no_ktp" varchar(16) COLLATE "pg_catalog"."default", "no_hp" varchar(16) COLLATE "pg_catalog"."default", "alamat" text COLLATE "pg_catalog"."default", "id_kelurahan" varchar(6) COLLATE "pg_catalog"."default", "id_kecamatan" varchar(5) COLLATE "pg_catalog"."default", "id_kab_kota" varchar(4) COLLATE "pg_catalog"."default", "id_provinsi" varchar(3) COLLATE "pg_catalog"."default", "id_kode_pos" varchar(6) COLLATE "pg_catalog"."default", </pre>

	<pre> "tempat_lahir" varchar(100) COLLATE "pg_catalog"."default", "tanggal_lahir" date, "jenis_kelamin" varchar(10) COLLATE "pg_catalog"."default", "agama" varchar(15) COLLATE "pg_catalog"."default", "target_pendapatan" int8, "status" varchar(10) COLLATE "pg_catalog"."default", "created_by" varchar(20) COLLATE "pg_catalog"."default", "created_date" timestamp(6), "updated_by" varchar(20) COLLATE "pg_catalog"."default", "updated_date" timestamp(6), "username" varchar(32) COLLATE "pg_catalog"."default", "password" varchar(64) COLLATE "pg_catalog"."default", "token" varchar(255) COLLATE "pg_catalog"."default", "last_activity" timestamp(6)); </pre>
tarif	<pre> CREATE TABLE "public"."tarif" ("id_tarif" varchar(32) COLLATE "pg_catalog"."default" NOT NULL, "id_jenis_kendaraan" varchar(32) COLLATE "pg_catalog"."default" NOT NULL, "id_jenis_lokasi" varchar(32) COLLATE "pg_catalog"."default", "nominal" int8 NOT NULL, "dasar" varchar(255) COLLATE "pg_catalog"."default", "beda_jam" int2, "nominal_berikutnya" int4, "created_by" varchar(32) COLLATE "pg_catalog"."default", "created_date" timestamp(0), "updated_by" varchar(32) COLLATE "pg_catalog"."default", "updated_date" timestamp(0), "status" varchar(32) COLLATE "pg_catalog"."default", "lokasi_id" varchar(36) COLLATE "pg_catalog"."default" NOT NULL, </pre>

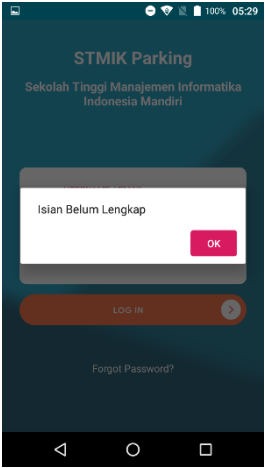
	<pre>"status_dynimc" varchar(36) COLLATE "pg_catalog"."default", "nominal_max" int4, "jam_max" int2, "denda" int4);</pre>
user_management	<pre>CREATE TABLE "public"."user_management" ("user_username" varchar(32) COLLATE "pg_catalog"."default" NOT NULL, "user_password" varchar(255) COLLATE "pg_catalog"."default", "user_nama" varchar(32) COLLATE "pg_catalog"."default", "user_email" varchar(32) COLLATE "pg_catalog"."default", "user_telp" varchar(32) COLLATE "pg_catalog"."default", "role_id" varchar(32) COLLATE "pg_catalog"."default" NOT NULL, "user_token" varchar(255) COLLATE "pg_catalog"."default");</pre>
lokasi_all	<pre>CREATE TABLE "public"."lokasi_all" ("kode_pos" varchar(255) COLLATE "pg_catalog"."default", "kelurahan" varchar(255) COLLATE "pg_catalog"."default", "kecamatan" varchar(255) COLLATE "pg_catalog"."default", "jenis" varchar(255) COLLATE "pg_catalog"."default", "kab_kota" varchar(255) COLLATE "pg_catalog"."default", "provinsi" varchar(255) COLLATE "pg_catalog"."default", "lokasi_id" varchar(36) COLLATE "pg_catalog"."default" NOT NULL, "area_parkir" varchar(255) COLLATE "pg_catalog"."default", "alamat" varchar(255) COLLATE "pg_catalog"."default", "target_harian" int8);</pre>

4.3. Blackbox Testing

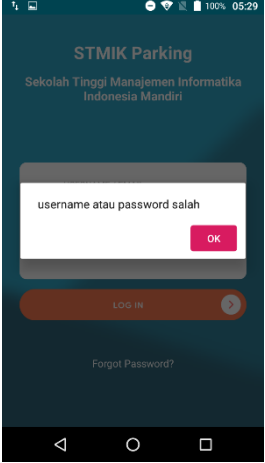
TABEL: 4. 3. Pengujian *Login Satu*

1. Login Aplikasi Android		
Skenario pengujian	Mengosongkan <i>username</i> dan <i>password</i> lalu klik login	
Hasil yang diharapkan	Sistem akan menolak login dan menampilkan pesan notifikasi	
Hasil pengujian	Sistem menolak login dan menampilkan pesan notifikasi	
Kesimpulan	Valid	

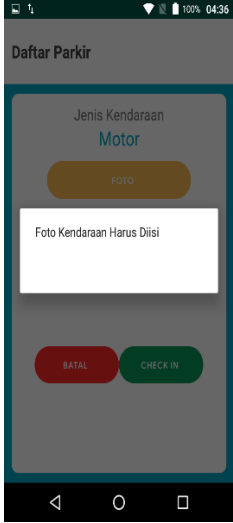
TABEL: 4. 4. Pengujian *Login Dua*

2. Login Aplikasi Android		
Skenario pengujian	Mengisi <i>username</i> atau <i>password</i> saja lalu klik login	
Hasil yang diharapkan	Sistem akan menolak login dan menampilkan pesan notifikasi	
Hasil pengujian	Sistem menolak login dan menampilkan pesan notifikasi	
Kesimpulan	Valid	

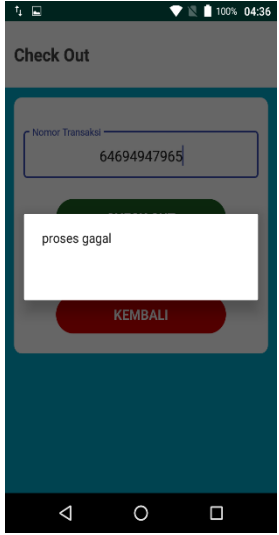
TABEL: 4. 5. Pengujian *Login Tiga*

3. Login Aplikasi Android		
Skenario pengujian	Mengisi <i>username</i> dan <i>password</i> tidak terdaftar lalu klik login	
Hasil yang diharapkan	Sistem akan menolak login dan menampilkan pesan notifikasi	
Hasil pengujian	Sistem menolak login dan menampilkan pesan notifikasi	
Kesimpulan	valid	

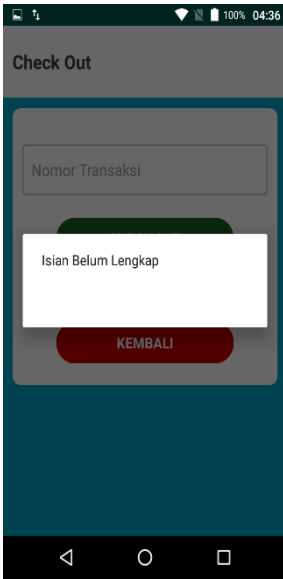
TABEL: 4. 6. Pengujian Parkir Masuk

4. Parkir masuk		
Skenario pengujian	Parkir masuk tapi tidak memfoto kendaraan	
Hasil yang diharapkan	Sistem akan menolak mencetak karcis dan menampilkan pesan notifikasi	
Hasil pengujian	Sistem menolak mencetak karcis dan menampilkan pesan notifikasi	
Kesimpulan	valid	

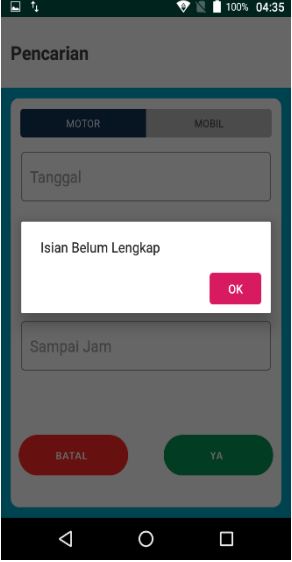
TABEL: 4. 7. Pengujian Parkir Keluar Satu

5. Parkir keluar		
Skenario pengujian	Mengisi dengan kode karcis sembarang	
Hasil yang diharapkan	Sistem akan menolak mencetak struk dan menampilkan pesan notifikasi	
Hasil pengujian	Sistem menolak mencetak struk dan menampilkan pesan notifikasi	
Kesimpulan	valid	

TABEL: 4. 8. Pengujian Parkir Keluar Dua

6. Parkir keluar		
Skenario pengujian	Kode karcis tidak diisi	
Hasil yang diharapkan	Sistem akan menolak mencetak struk dan menampilkan pesan notifikasi	
Hasil pengujian	Sistem menolak mencetak struk dan menampilkan pesan notifikasi	
Kesimpulan	Valid	

TABEL: 4. 9. Pengujian Cari Karcis


7. Cari karcis		
Skenario pengujian	Paramameter pencarian tidak diisi	
Hasil yang diharapkan	Sistem tidak akan menampilkan data parkir dan menampilkan pesan notifikasi	
Hasil pengujian	Sistem tidak menampilkan data parkir dan menampilkan pesan notifikasi	
Kesimpulan	Valid	

TABEL: 4. 10. Pengujian *Login* Website Satu

8. Login Website	
Skenario pengujian	Mengosongkan <i>username</i> dan <i>password</i> lalu klik login
Hasil yang diharapkan	Sistem akan menolak login dan menampilkan pesan notifikasi
Hasil pengujian	Sistem menolak login dan menampilkan pesan notifikasi

Kesimpulan	valid
------------	-------

TABEL: 4. 11. Pengujian *Login* Website Dua

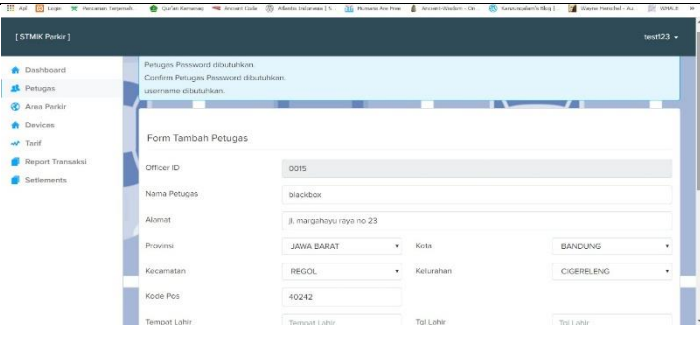
	
9. Login Website	
Skenario pengujian	Mengisi <i>username</i> atau <i>password</i> saja lalu klik login
Hasil yang diharapkan	Sistem akan menolak login dan menampilkan pesan notifikasi
Hasil pengujian	Sistem menolak login dan menampilkan pesan notifikasi
Kesimpulan	Valid

TABEL: 4. 12. Pengujian *Login* Website Tiga

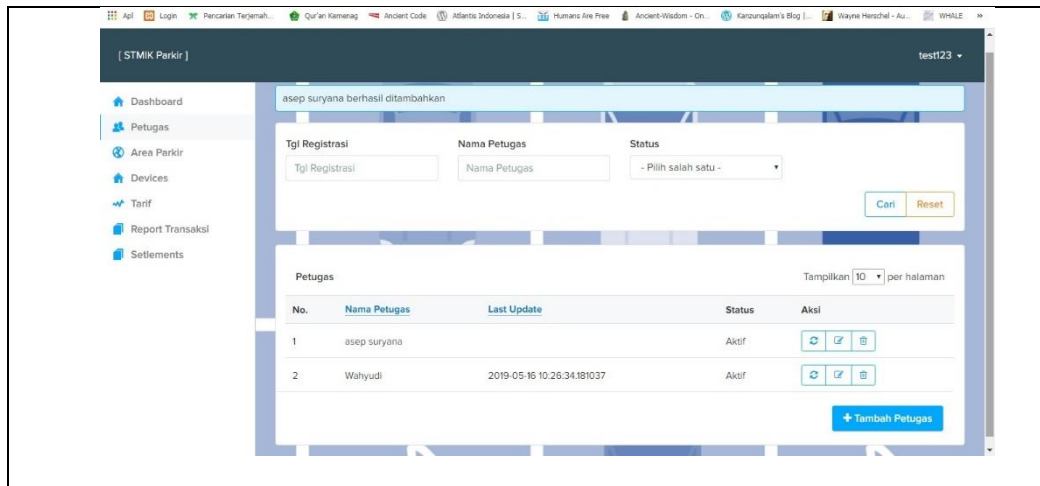
	
10. Login Website	
Skenario pengujian	Mengisi <i>username</i> dan <i>password</i> tidak terdaftar lalu klik login
	login

Hasil yang diharapkan	Sistem akan menolak login dan menampilkan pesan notifikasi
Hasil pengujian	Sistem menolak login dan menampilkan pesan notifikasi
Kesimpulan	Valid

TABEL: 4. 13. Pengujian Tambah Petugas Satu

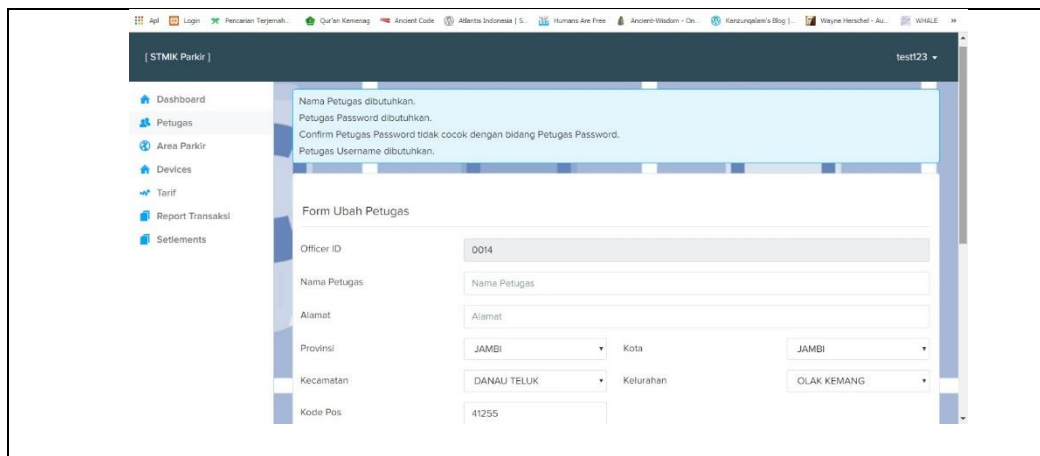
	
11. Tambah petugas	
Skenario pengujian	Tidak mengisi parameter secara benar
Hasil yang diharapkan	Sistem akan menolak menyimpan data dan menampilkan pesan notifikasi
Hasil pengujian	Sistem menolak menyimpan data dan menampilkan pesan notifikasi
Kesimpulan	Valid

TABEL: 4. 14. Pengujian Tambah Petugas Dua



12. Tambah petugas	
Skenario pengujian	Mengisi parameter secara benar
Hasil yang diharapkan	Sistem akan menyimpan data dan menampilkan pesan notifikasi
Hasil pengujian	Sistem menyimpan data dan menampilkan pesan notifikasi
Kesimpulan	Valid

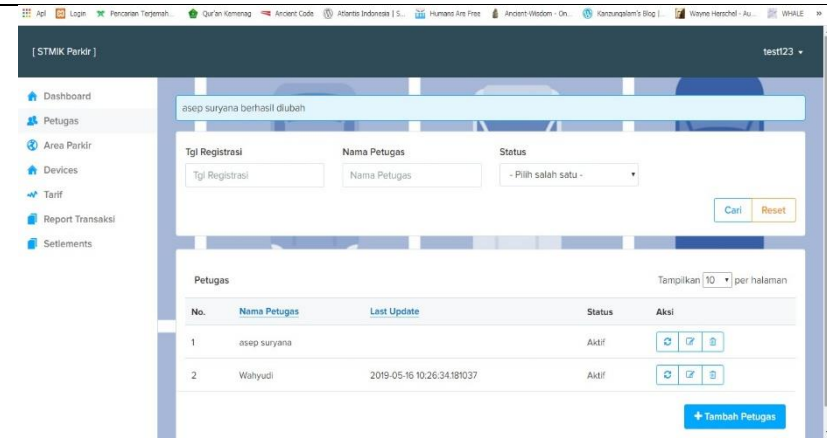
TABEL: 4. 15. Pengujian Ubah Petugas Satu



13. ubah petugas	
Skenario pengujian	Mengubah data petugas lalu tidak mengisi parameter secara benar
Hasil yang diharapkan	Sistem akan menolak menyimpan data dan menampilkan pesan notifikasi
Hasil pengujian	Sistem menolak menyimpan data dan menampilkan pesan notifikasi
Kesimpulan	Valid

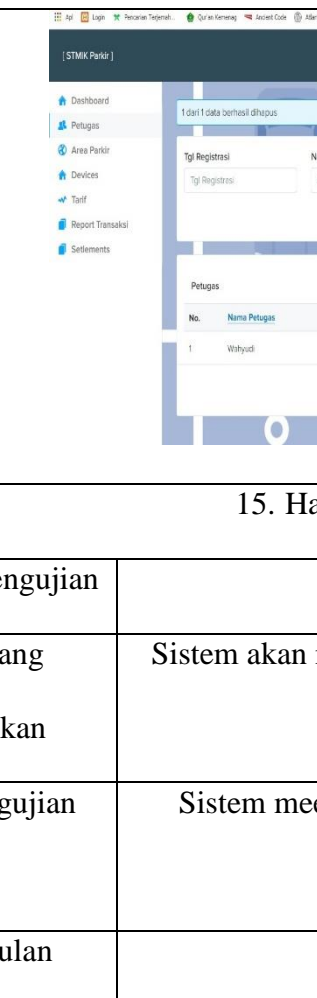
TABEL: 4. 16. Pengujian Ubah Petugas Dua

14. Ubah petugas	
Skenario pengujian	Mengubah data petugas lalu mengisi parameter secara benar
Hasil yang diharapkan	Sistem akan menyimpan data dan menampilkan pesan notifikasi

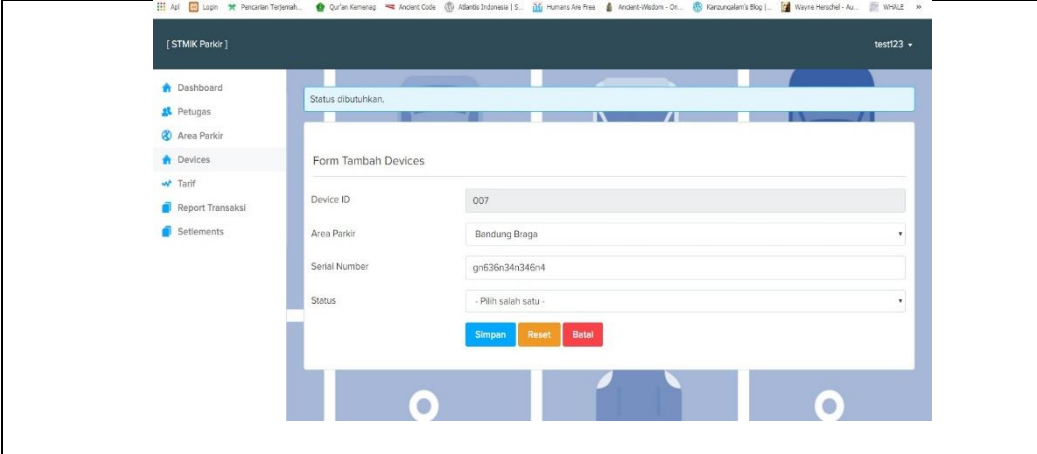


Hasil pengujian	Sistem menyimpan data dan menampilkan pesan notifikasi
Kesimpulan	Valid

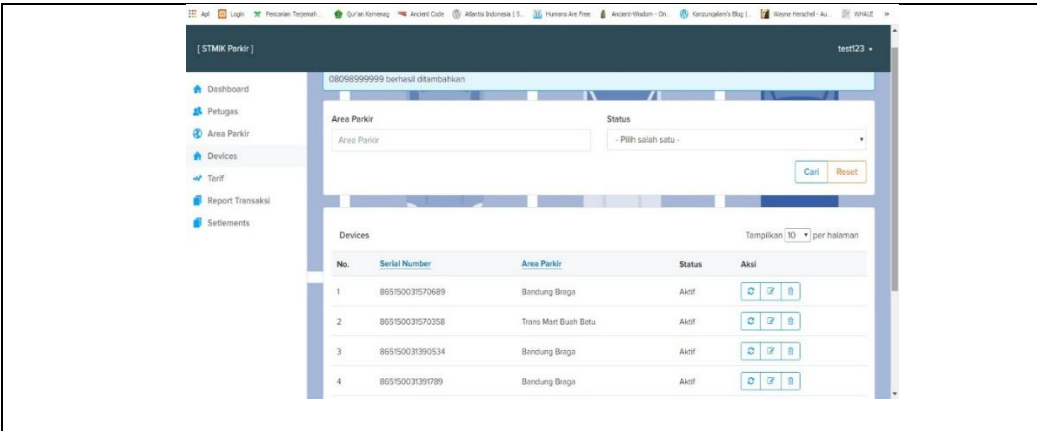
TABEL: 4. 17. Pengujian Hapus Petugas

	
15. Hapus petugas	
Skenario pengujian	Menghapus petugas
Hasil yang diharapkan	Sistem akan menghapus data dan menampilkan pesan notifikasi
Hasil pengujian	Sistem meenghapus data dan menampilkan pesan notifikasi
Kesimpulan	Valid

TABEL: 4. 18. Pengujian Tambah *Device* Satu

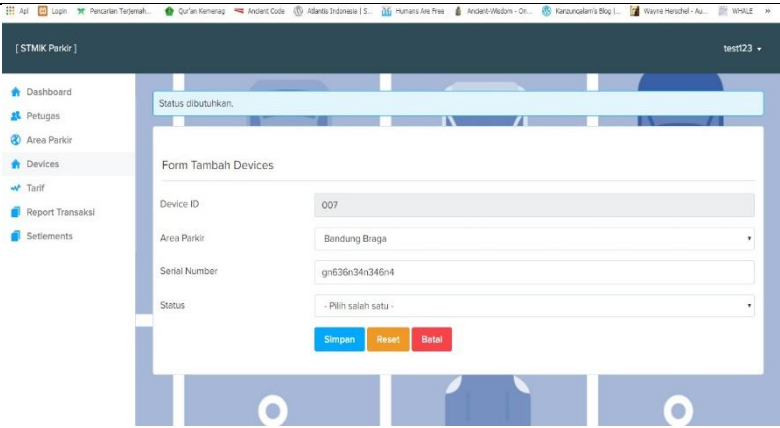
	
16. Tambah device	
Skenario pengujian	Tidak mengisi parameter secara benar
Hasil yang diharapkan	Sistem akan menolak menyimpan data dan menampilkan pesan notifikasi
Hasil pengujian	Sistem menolak menyimpan data dan menampilkan pesan notifikasi
Kesimpulan	Valid

TABEL: 4. 19. Pengujian Tambah *Device* Dua

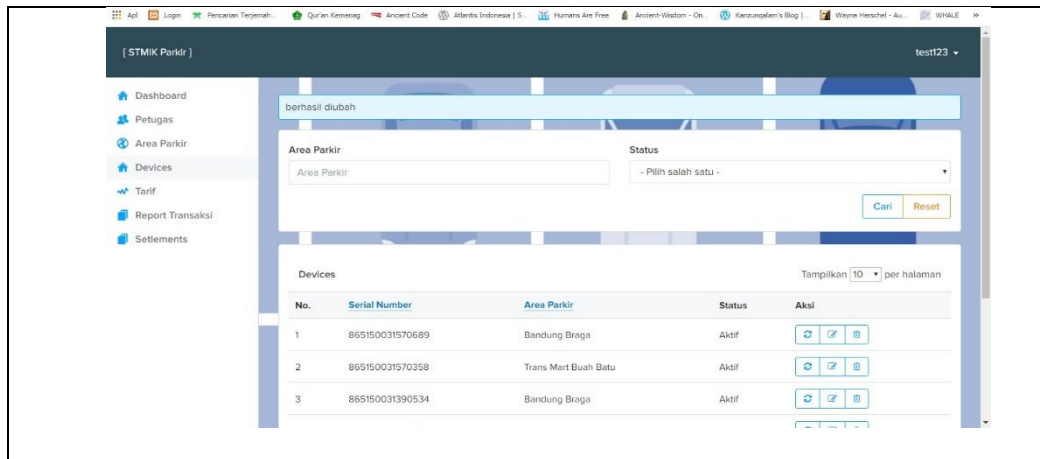
	
17. Tambah device	

Skenario pengujian	Mengisi parameter secara benar
Hasil yang diharapkan	Sistem akan menyimpan data dan menampilkan pesan notifikasi
Hasil pengujian	Sistem menyimpan data dan menampilkan pesan notifikasi
Kesimpulan	Valid

TABEL: 4. 20. Pengujian Ubah *Device* Satu

	
18. Ubah device	
Skenario pengujian	Mengubah data device lalu tidak mengisi parameter secara benar
Hasil yang diharapkan	Sistem akan menolak menyimpan data dan menampilkan pesan notifikasi
Hasil pengujian	Sistem menolak menyimpan data dan menampilkan pesan notifikasi
Kesimpulan	Valid

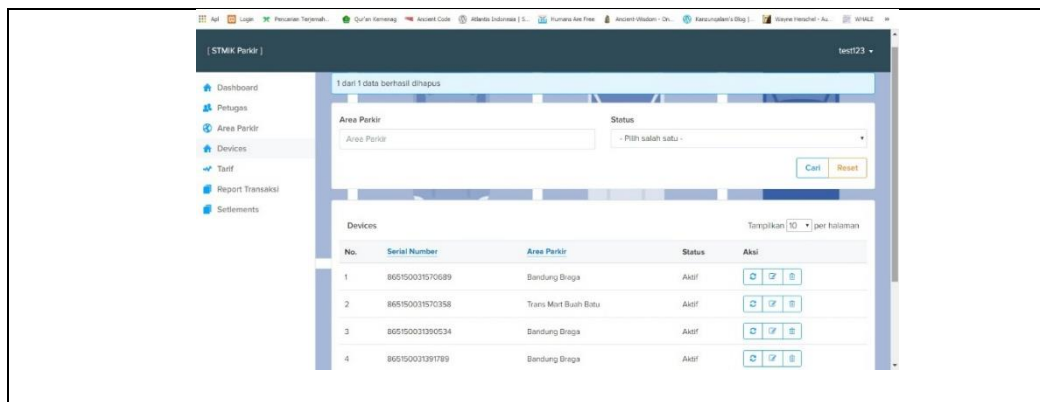
TABEL: 4. 21. Pengujian Ubah *Device* Dua



No.	Serial Number	Area Parkir	Status	Aksi
1	865150031570689	Bandung Braga	Aktif	[Edit] [Hapus]
2	865150031570358	Trans Mart Buah Batu	Aktif	[Edit] [Hapus]
3	865150031390534	Bandung Braga	Aktif	[Edit] [Hapus]

19. Ubah device	
Skenario pengujian	Mengubah data device lalu mengisi parameter secara benar
Hasil yang diharapkan	Sistem akan menyimpan data dan menampilkan pesan notifikasi
Hasil pengujian	Sistem menyimpan data dan menampilkan pesan notifikasi
Kesimpulan	Valid

TABEL: 4. 22. Pengujian Hapus *Device*



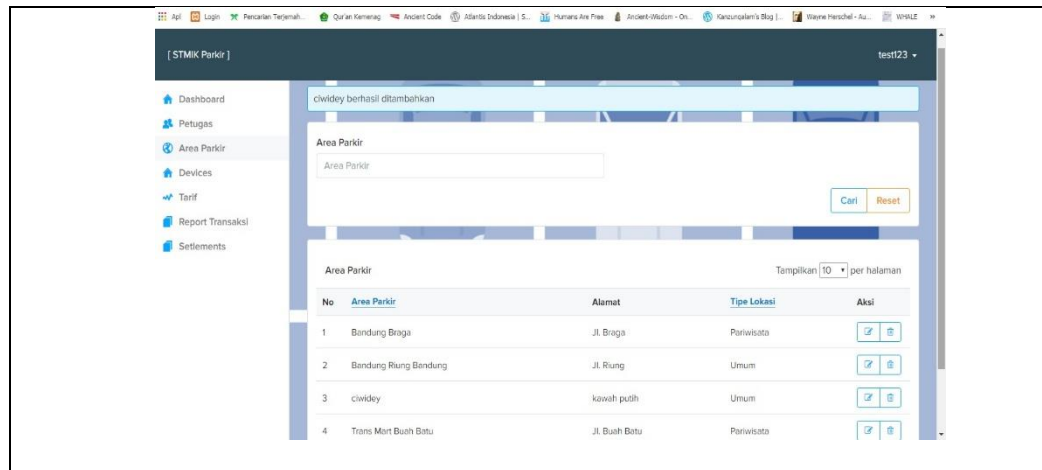
No.	Serial Number	Area Parkir	Status	Aksi
1	865150031570689	Bandung Braga	Aktif	[Edit] [Hapus]
2	865150031570358	Trans Mart Buah Batu	Aktif	[Edit] [Hapus]
3	865150031390534	Bandung Braga	Aktif	[Edit] [Hapus]
4	865150031391789	Bandung Braga	Aktif	[Edit] [Hapus]

20. Hapus device	
Skenario pengujian	Hapus data device
Hasil yang diharapkan	Sistem akan menghapus data dan menampilkan pesan notifikasi
Hasil pengujian	Sistem menghapus data dan menampilkan pesan notifikasi
Kesimpulan	Valid

TABEL: 4. 23. Pengujian Tambah Area Satu

21. Tambah Area	
Skenario pengujian	Tidak mengisi parameter secara benar
Hasil yang diharapkan	Sistem akan menolak menyimpan data dan menampilkan pesan notifikasi
Hasil pengujian	Sistem menolak menyimpan data dan menampilkan pesan notifikasi
Kesimpulan	Valid

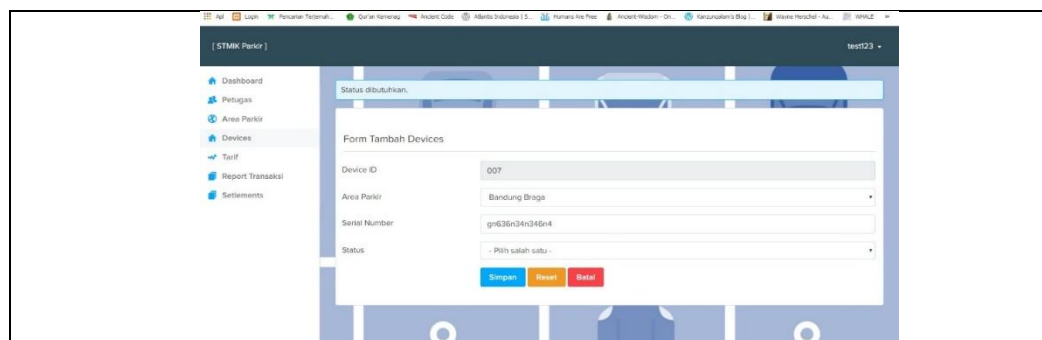
TABEL: 4. 24. Pengujian Tambah Area Dua



22. Tambah area

Skenario pengujian	Mengisi parameter secara benar
Hasil yang diharapkan	Sistem akan menyimpan data dan menampilkan pesan notifikasi
Hasil pengujian	Sistem menyimpan data dan menampilkan pesan notifikasi
Kesimpulan	Valid

TABEL: 4. 25. Pengujian Ubah Area Satu

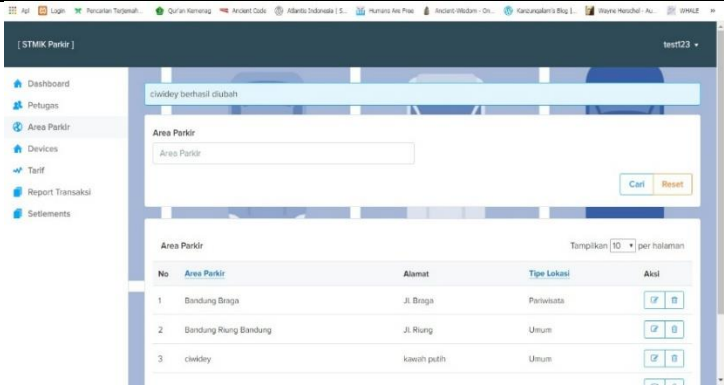


23. Ubah area

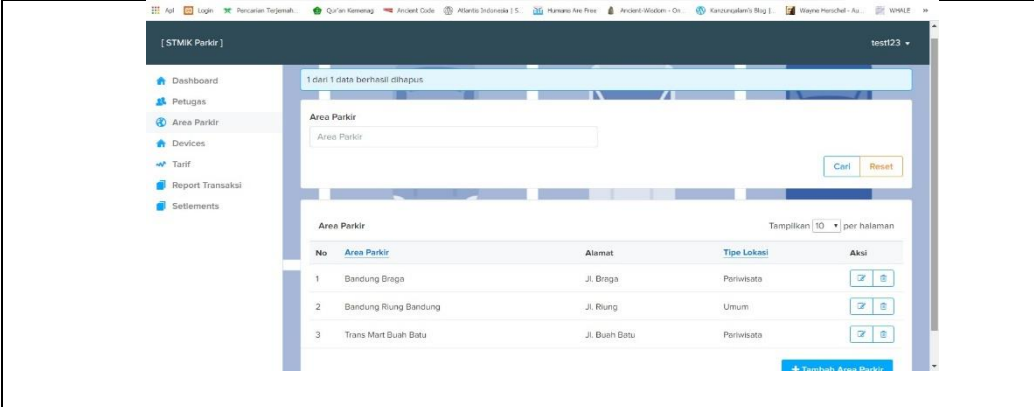
--	--

Skenario pengujian	Mengubah data area lalu tidak mengisi parameter secara benar
Hasil yang diharapkan	Sistem akan menolak menyimpan data dan menampilkan pesan notifikasi
Hasil pengujian	Sistem menolak menyimpan data dan menampilkan pesan notifikasi
Kesimpulan	Valid

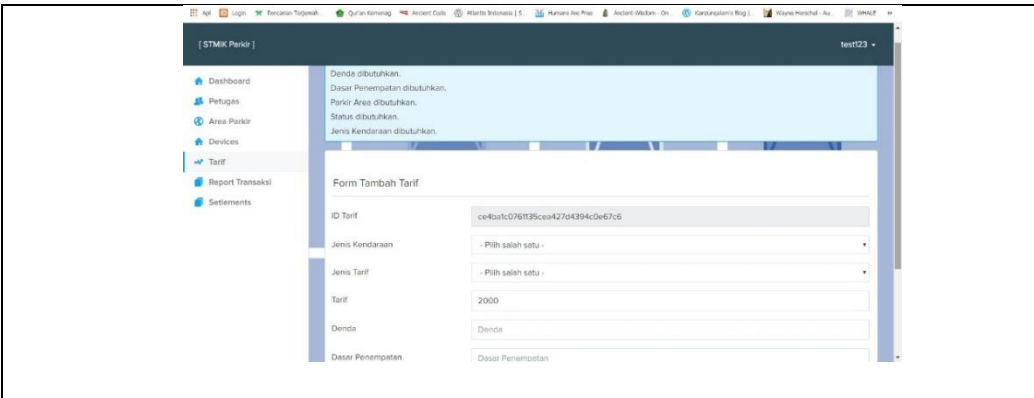
TABEL: 4. 26. Pengujian Ubah Area Dua

	
24. Ubah area	
Skenario pengujian	Mengubah data area lalu mengisi parameter secara benar
Hasil yang diharapkan	Sistem akan menyimpan data dan menampilkan pesan notifikasi
Hasil pengujian	Sistem menyimpan data dan menampilkan pesan notifikasi
Kesimpulan	Valid

TABEL: 4. 27. Pengujian Hapus Area

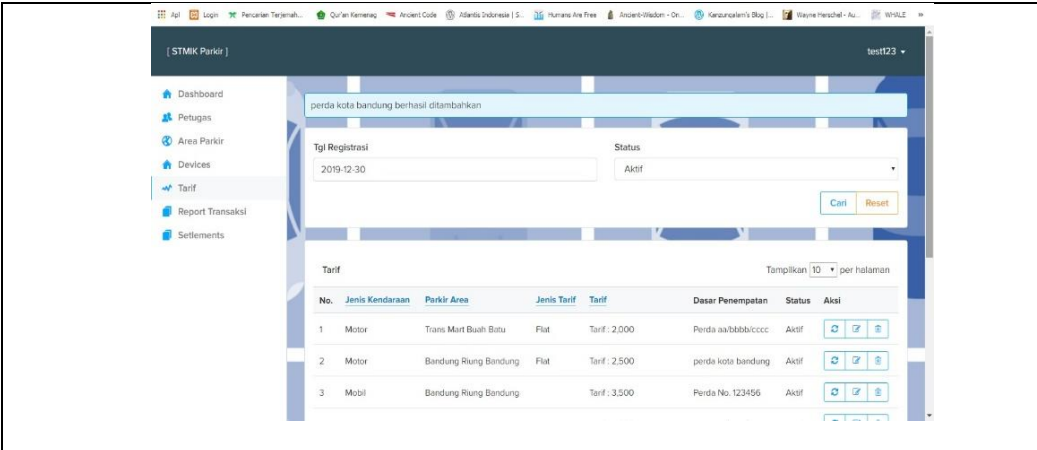
	
<p>25. Hapus area</p>	
<p>Skenario pengujian</p>	<p>Hapus data area</p>
<p>Hasil yang diharapkan</p>	<p>Sistem akan menghapus data dan menampilkan pesan notifikasi</p>
<p>Hasil pengujian</p>	<p>Sistem meghapus data dan menampilkan pesan notifikasi</p>
<p>Kesimpulan</p>	<p>Valid</p>

TABEL: 4. 28. Pengujian Tambah Tarif Satu

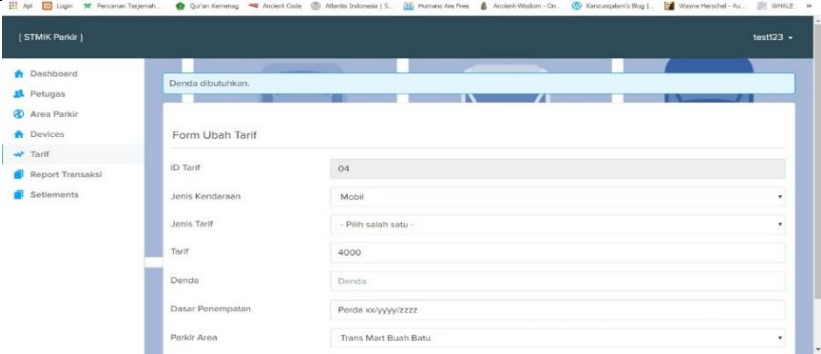
	
<p>26. Tambah tarif</p>	
<p>Skenario pengujian</p>	<p>Tidak mengisi parameter secara benar</p>

Hasil yang diharapkan	Sistem akan menolak menyimpan data dan menampilkan pesan notifikasi
Hasil pengujian	Sistem menolak menyimpan data dan menampilkan pesan notifikasi
Kesimpulan	Valid

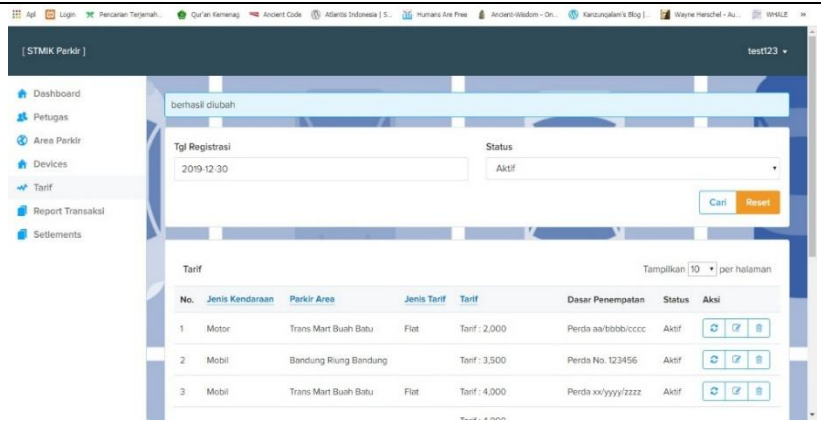
TABEL: 4. 29. Pengujian Tambah Tarif Dua

 <p>The screenshot shows a web application interface for 'STMK Parkir'. At the top, a notification banner reads 'perda kota bandung berhasil ditambahkan'. Below this, there are input fields for 'Tgl Registrasi' (2019-12-30) and 'Status' (Aktif), with 'Cari' and 'Reset' buttons. A table titled 'Tarif' displays the following data:</p> <table border="1"> <thead> <tr> <th>No.</th> <th>Jenis Kendaraan</th> <th>Parkir Area</th> <th>Jenis Tarif</th> <th>Tarif</th> <th>Dasar Penempatan</th> <th>Status</th> <th>Aksi</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Motor</td> <td>Trans Muar Buah Batu</td> <td>Flat</td> <td>Tarif : 2.000</td> <td>Perda aa/bbbbcccc</td> <td>Aktif</td> <td>[Edit] [Hapus] [Tambah]</td> </tr> <tr> <td>2</td> <td>Motor</td> <td>Bandung Riung Bandung</td> <td>Flat</td> <td>Tarif : 2.500</td> <td>perda kota bandung</td> <td>Aktif</td> <td>[Edit] [Hapus] [Tambah]</td> </tr> <tr> <td>3</td> <td>Mobil</td> <td>Bandung Riung Bandung</td> <td></td> <td>Tarif : 3.500</td> <td>Perda No.123456</td> <td>Aktif</td> <td>[Edit] [Hapus] [Tambah]</td> </tr> </tbody> </table>		No.	Jenis Kendaraan	Parkir Area	Jenis Tarif	Tarif	Dasar Penempatan	Status	Aksi	1	Motor	Trans Muar Buah Batu	Flat	Tarif : 2.000	Perda aa/bbbbcccc	Aktif	[Edit] [Hapus] [Tambah]	2	Motor	Bandung Riung Bandung	Flat	Tarif : 2.500	perda kota bandung	Aktif	[Edit] [Hapus] [Tambah]	3	Mobil	Bandung Riung Bandung		Tarif : 3.500	Perda No.123456	Aktif	[Edit] [Hapus] [Tambah]
No.	Jenis Kendaraan	Parkir Area	Jenis Tarif	Tarif	Dasar Penempatan	Status	Aksi																										
1	Motor	Trans Muar Buah Batu	Flat	Tarif : 2.000	Perda aa/bbbbcccc	Aktif	[Edit] [Hapus] [Tambah]																										
2	Motor	Bandung Riung Bandung	Flat	Tarif : 2.500	perda kota bandung	Aktif	[Edit] [Hapus] [Tambah]																										
3	Mobil	Bandung Riung Bandung		Tarif : 3.500	Perda No.123456	Aktif	[Edit] [Hapus] [Tambah]																										
27. Tambah tarif																																	
Skenario pengujian	Mengisi parameter secara benar																																
Hasil yang diharapkan	Sistem akan menyimpan data dan menampilkan pesan notifikasi																																
Hasil pengujian	Sistem menyimpan data dan menampilkan pesan notifikasi																																
Kesimpulan	Valid																																

TABEL: 4. 30. Pengujian Ubah Tarif Satu

	
28. Ubah tarif	
Skenario pengujian	Mengubah data tarif lalu tidak mengisi parameter secara benar
Hasil yang diharapkan	Sistem akan menolak menyimpan data dan menampilkan pesan notifikasi
Hasil pengujian	Sistem menolak menyimpan data dan menampilkan pesan notifikasi
Kesimpulan	Valid

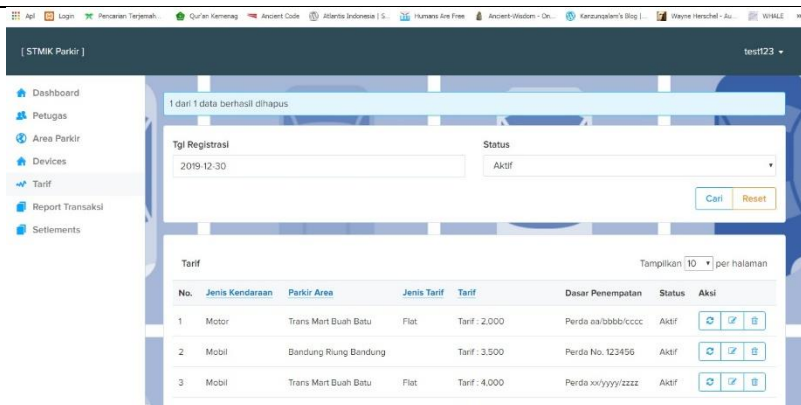
TABEL: 4. 31. Pengujian Ubah Tarif Dua

																																	
<table border="1"> <thead> <tr> <th>No.</th> <th>Jenis Kendaraan</th> <th>Parkir Area</th> <th>Jenis Tarif</th> <th>Tarif</th> <th>Dasar Penempatan</th> <th>Status</th> <th>Aksi</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Motor</td> <td>Trans Mart Buah Batu</td> <td>Flot</td> <td>Tarif : 2.000</td> <td>Perda aa/kbbb/cccc</td> <td>Aktif</td> <td>[Edit] [Hapus] [Tambah]</td> </tr> <tr> <td>2</td> <td>Mobil</td> <td>Bendung Rung Bendung</td> <td></td> <td>Tarif : 3.500</td> <td>Perda No. 123456</td> <td>Aktif</td> <td>[Edit] [Hapus] [Tambah]</td> </tr> <tr> <td>3</td> <td>Mobil</td> <td>Trans Mart Buah Batu</td> <td>Flot</td> <td>Tarif : 4.000</td> <td>Perda xx/yyyy/zzzz</td> <td>Aktif</td> <td>[Edit] [Hapus] [Tambah]</td> </tr> </tbody> </table>		No.	Jenis Kendaraan	Parkir Area	Jenis Tarif	Tarif	Dasar Penempatan	Status	Aksi	1	Motor	Trans Mart Buah Batu	Flot	Tarif : 2.000	Perda aa/kbbb/cccc	Aktif	[Edit] [Hapus] [Tambah]	2	Mobil	Bendung Rung Bendung		Tarif : 3.500	Perda No. 123456	Aktif	[Edit] [Hapus] [Tambah]	3	Mobil	Trans Mart Buah Batu	Flot	Tarif : 4.000	Perda xx/yyyy/zzzz	Aktif	[Edit] [Hapus] [Tambah]
No.	Jenis Kendaraan	Parkir Area	Jenis Tarif	Tarif	Dasar Penempatan	Status	Aksi																										
1	Motor	Trans Mart Buah Batu	Flot	Tarif : 2.000	Perda aa/kbbb/cccc	Aktif	[Edit] [Hapus] [Tambah]																										
2	Mobil	Bendung Rung Bendung		Tarif : 3.500	Perda No. 123456	Aktif	[Edit] [Hapus] [Tambah]																										
3	Mobil	Trans Mart Buah Batu	Flot	Tarif : 4.000	Perda xx/yyyy/zzzz	Aktif	[Edit] [Hapus] [Tambah]																										

29. Ubah tarif	
Skenario pengujian	Mengubah data tarif lalu mengisi parameter secara benar
Hasil yang diharapkan	Sistem akan menyimpan data dan menampilkan pesan notifikasi
Hasil pengujian	Sistem menyimpan data dan menampilkan pesan notifikasi
Kesimpulan	Valid

TABEL: 4. 32. Pengujian Hapus Tarif

30. Hapus tarif	
Skenario pengujian	Menghapus data tarif
Hasil yang diharapkan	Sistem akan menghapus data dan menampilkan pesan notifikasi
Hasil pengujian	Sistem menghapus data dan menampilkan pesan notifikasi
Kesimpulan	Valid



TABEL: 4. 33. Pengujian Laporan Transaksi




31. Laporan Transaksi	
Skenario pengujian	Mengunduh file xls laporan transaksi
Hasil yang diharapkan	File xls berhasil di unduh
Hasil pengujian	File xls berhasil di unduh
Kesimpulan	Valid

BAB V

PENUTUP

5.1. Kesimpulan

- A. Pengelolaan parkir telah menggunakan aplikasi sistem yang dapat membantu menyelesaikan masalah yang terjadi pada sistem parkir manual/konvensional dipinggir jalan.
- B. Aplikasi yang dibuat dapat membantu pemerintah dalam penertiban, pengelolaan parkir liar serta dapat meningkatkan efisiensi pendapatan retribusi parkir.

5.2. Saran

Bagi para peneliti selanjutnya aplikasi ini dapat dikembangkan lagi misalnya, seperti:

- A. Penggunaan Quick Response Code Indonesia (QRIS), sehingga pada proses pembayaran karcis pun dapat menggunakan jasa fintech.
- B. Penambahan keamanan *messaging* ISO8583, sehingga aplikasi dapat melakukan topup melalui bank.

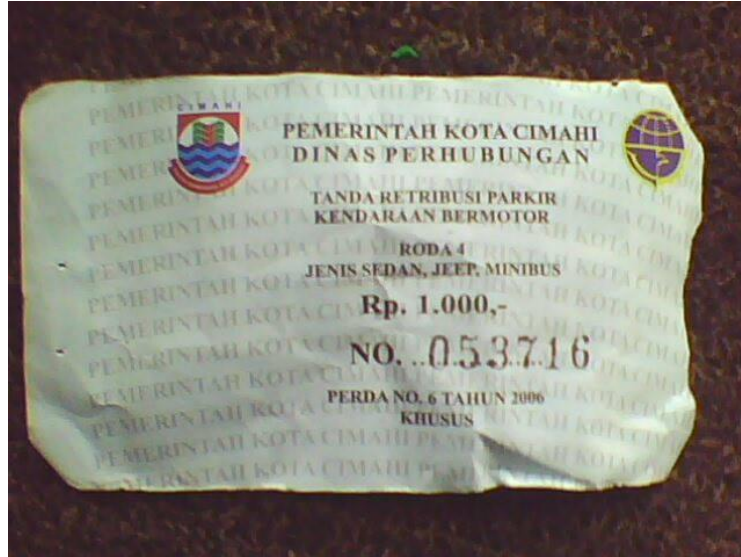
DAFTAR PUSTAKA

- Andry. (2011): *Android A Sampai Z*, Pcplus, Jakarta.
- Ambler, S,W. (2005): *The Elements of UML Style*, Cambridge, 168 halaman.
- Connoly, Thomas. dan Begg, Carolyn. (2015): *Database Systems sixth edition*, Pearson, 1329 halaman.
- Dyah, A,K. (2019): Penerapan Sistem Gerbang Pelayanan Terpadu Nasional Dalam Menunjang Transaksi Daring, *Jurnal Bisnis dan Manajemen Universitas Merdeka Malang*, 8, 56-64.
- Gusrión, D. (2018): Sistem Pembayaran Secara Cashless Pada Koperasi Sekolah Yayasan Igaras, *LPPM UPI YPTK Padang*, 9, 63-72.
- Hayati, A,P. dan Lestaringati, S,I. (2018): Electronic Data Capture Berbasis Mini Pc Raspberry Pi. *Jurnal Teknik Komputer Unikom*, 6, 1-6.
- Heryanto, I. (2012): *Membuat Database Dengan Microsoft Acces Edisi Revisi*, Informatika, 480 halaman.
- Jogiyanto, H. (2008): *Metodologi Penelitian Sistem Informasi*, Andi, 312 halaman.
- Krismiaji. (2015): *Sistem Informasi Akutansi Edisi Keempat*, UPP STIM YKPN, 514 halaman.
- Mulyadi. (2016): *Sistem Informasi Akutansi Edisi 4*, Salemba Empat, 582 halaman.
- Nafaliya, D,H. dan Aji, S. (2018): Role of Support Infrastructure and Information System on Non-Cash Transaction Policies. *Economics Development Analysis Journal*, 6, 224-230.
- Pressman, Roger. (2012): *Rekayasa Perangkat Lunak Edisi 7*, Andi, 782 halaman.
- Romney, M,B, dan Steinbart, P,J. (2014): *Sistem Informasi Akutansi Edisi 13*, Salemba Empat, 906 halaman.
- Safaat, Nazruddin. (2018): *Pemograman Aplikasi Mobile Berbasis Android*, Informatika, 392 halaman.
- Saidi, D. (2010): *Hukum Penerimaan Negara Bukan Pajak*, Raja Grafindo Persada, 196 halaman.
- Siahaan, M, P. (2016): *Pajak Daerah & Retribusi Daerah edisi revisi*, Raja Grafindo Persada, 702 halaman.
- Peraturan Daerah Provinsi Jawa Barat Nomor 05 Tahun 2009, Pemerintah Provinsi Jawa Barat, 87 halaman.
- Peraturan Daerah Kota Bandung Nomor 09 Tahun 2010, Pemerintah Kota Bandung, 17 halaman.
- Undang-undang Nomor 28 Tahun 2009 Tentang Pajak Daerah dan Retribusi Daerah, Republik Indonesia, 124 halaman.
- Undang-undang Nomor 22 Tahun 2009 Tentang Lalu Lintas dan Angkutan Jalan, Republik Indonesia, 203. Halaman
- Keputusan Menteri Perhubungan Nomor 4 Tahun 1994, Republik Indonesia, 13 Halaman.

- Beepos.id (2019): Pengertian Dan Fungsi Dari Point Of Sale, <https://www.beepos.id/blog/pengertian-dan-fungsi-dari-point-of-sale/>. Diakses 7 Nopember 2019.
- Jaringanprima.co.id (2019): Mengenal Qr Code, <https://www.jaringanprima.co.id/id/mengenal-qr-code>. Diakses 7 Nopember 2019.
- Scdc.binus.ac.id (2016): Perbedaan white box testing dan black box testing, <http://scdc.binus.ac.id/himsisfo/2016/10/perbedaan-white-box-testing-dan-black-box-testing/>. Diakses 7 Nopember 2019.

Lampiran-Lampiran

Lampiran 1



(Karcis Parkir Resmi)

Lampiran 2



(Karcis Parkir Tidak Resmi)

Lampiran 3

Index.php

```

<?php
/**
 * *
 * @package CodeIgniter
 * @author EllisLab Dev Team
 * @copyright Copyright (c) 2008 - 2014, EllisLab, Inc.
 (https://ellislab.com/)
 * @copyright Copyright (c) 2014 - 2016, British Columbia Institute of
 Technology (http://bcit.ca/)
 * @license http://opensource.org/licenses/MIT MIT License
 * @link https://codeigniter.com
 * @since Version 1.0.0
 * @filesource
 */
define('ENVIRONMENT', isset($_SERVER['CI_ENV']) ?
$_SERVER['CI_ENV'] : 'production');
switch (ENVIRONMENT)
{
    case 'development':
        error_reporting(-1);
        ini_set('display_errors', 1);
        break;

    case 'testing':
    case 'production':
        ini_set('display_errors', 0);
        if (version_compare(PHP_VERSION, '5.3', '>='))
        {
            error_reporting(E_ALL & ~E_NOTICE &
~E_DEPRECATED & ~E_STRICT & ~E_USER_NOTICE &
~E_USER_DEPRECATED);
        }
        else
        {
            error_reporting(E_ALL & ~E_NOTICE &
~E_STRICT & ~E_USER_NOTICE);
        }
        break;
    default:
        header('HTTP/1.1 503 Service Unavailable.', TRUE, 503);
        echo 'The application environment is not set correctly.';
        exit(1);
}

$system_path = 'system';
$application_folder = 'application';

```

```

$view_folder = "";
if (defined('STDIN'))
{
    chdir(dirname(__FILE__));
}

if (($_temp = realpath($system_path)) !== FALSE)
{
    $system_path = $_temp.DIRECTORY_SEPARATOR;
}
else
{
    $system_path = strtr(
        rtrim($system_path, '\\'),
        '\\',
        DIRECTORY_SEPARATOR.DIRECTORY_SEPARATOR
    ).DIRECTORY_SEPARATOR;
}
if (! is_dir($system_path))
{
    header('HTTP/1.1 503 Service Unavailable.', TRUE, 503);
    echo 'Your system folder path does not appear to be set
correctly. Please open the following file and correct this:
'.pathinfo(__FILE__, PATHINFO_BASENAME);
    exit(3);
}
define('SELF', pathinfo(__FILE__, PATHINFO_BASENAME));
define('BASEPATH', $system_path);
define('FCPATH',
dirname(__FILE__).DIRECTORY_SEPARATOR);
define('SYSDIR', basename(BASEPATH));
if (is_dir($application_folder))
{
    if (($_temp = realpath($application_folder)) !== FALSE)
    {
        $application_folder = $_temp;
    }
    else
    {
        $application_folder = strtr(
            rtrim($application_folder, '\\'),
            '\\',
            DIRECTORY_SEPARATOR.DIRECTORY_SEPARATOR
        );
    }
}

```

```

    }
  }
  elseif
(is_dir(BASEPATH.$application_folder.DIRECTORY_SEPARATOR)
)
  {
    $application_folder = BASEPATH.strtr(
      trim($application_folder, '\\'),
      '\\',
      DIRECTORY_SEPARATOR.DIRECTORY_SEPARATOR
    );
  }
  else
  {
    header('HTTP/1.1 503 Service Unavailable.', TRUE, 503);
    echo 'Your application folder path does not appear to be
set correctly. Please open the following file and correct this: '.SELF;
    exit(3); //
  }
  define('APPPATH',
$application_folder.DIRECTORY_SEPARATOR);
  if ( ! isset($view_folder[0])      &&
is_dir(APPPATH.'views'.DIRECTORY_SEPARATOR))
  {
    $view_folder = APPPATH.'views';
  }
  elseif (is_dir($view_folder))
  {
    if (($_temp = realpath($view_folder)) !== FALSE)
    {
      $view_folder = $_temp;
    }
    else
    {
      $view_folder = strtr(
        rtrim($view_folder, '\\'),
        '\\',
        DIRECTORY_SEPARATOR.DIRECTORY_SEPARATOR
      );
    }
  }
  elseif
(is_dir(APPPATH.$view_folder.DIRECTORY_SEPARATOR))
  {

```



```

        $view_folder = APPPATH.strtr(
            trim($view_folder, '\\'),
            '\\',

            DIRECTORY_SEPARATOR.DIRECTORY_SEPARATOR
        );
    }
    else
    {
        header('HTTP/1.1 503 Service Unavailable.', TRUE, 503);
        echo 'Your view folder path does not appear to be set
correctly. Please open the following file and correct this: '.SELF;
        exit(3); //
    }
    define('VIEWPATH',
        $view_folder.DIRECTORY_SEPARATOR);
require_once BASEPATH.'core/CodeIgniter.php';

```

Module device

```

<div class="box-typical box-typical-padding">
    <h5 class="m-t-lg with-border">Form <?php echo lang('label.' . $this-
>method) . '.' . db_lang('page_title.' . $this->module) ?></h5>
    <?php echo form_open(uri_string(), 'id="form_data" class="crud"
autocomplete="off") ?>
    <div class="row form-group">
        <label class="col-sm-3 form-control-label"><?php echo
db_lang('id_device') ?></label>
        <div class="col-sm-9">
            <?php echo form_input('id_device', $row->id_device ? $row-
>id_device : $id_device, 'class="form-control" id="id_device"
placeholder="'. db_lang('id_device') .'" readonly') ?>
        </div>
    </div>

    <div class="row form-group">
        <label class="col-sm-3 form-control-label"><?php echo
db_lang('device_lokasi_id') ?></label>
        <div class="col-sm-9">
            <?php
            $select = array(" => lang('select.pick')) + $area_parkir;
            echo form_dropdown('device_lokasi_id', $select, $row-
>device_lokasi_id, 'class="form-control" id="device_lokasi_id");
            ?>
        </div>
    </div>

```

```

        <div class="row form-group">
            <label class="col-sm-3 form-control-label"><?php echo
db_lang('sn_device') ?></label>
            <div class="col-sm-9">
                <?php echo form_input('sn_device', $row->sn_device,
'class="form-control" id="sn_device" placeholder="'.
db_lang('sn_device') .'"') ?>
            </div>
        </div>

        <div class="row form-group">
            <label class="col-sm-3 form-control-label"><?php echo
db_lang('status') ?></label>
            <div class="col-sm-9">
                <?php
                $select = array(" => lang('select.pick')) + $status;
                echo form_dropdown('status', $select, $row->status,
'class="form-control" id="status");
            ?>
            </div>
        </div>

        <div class="row form-group">
            <div class="col-sm-9 col-sm-push-3">
                <button type="submit" class="btn btn-primary submit"><?php
echo lang('buttons.save') ?></button>
                <button type="reset" class="btn btn-warning"><?php echo
lang('buttons.reset') ?></button>
                <button class="btn btn-danger cancel"><?php echo
lang('buttons.cancel') ?></button>
            </div>
        </div>
        <?php echo form_close() ?>
</div>

```

Module lokasi

<div class="box-typical box-typical-padding">

```

<h5 class="m-t-lg with-border">Form <?php echo lang('label.' . $this->method) . '': db_lang('page_title.' . $this->module) ?></h5>
<?php echo form_open(uri_string(), 'id="form_data" class="crud" autocomplete="off") ?>
    <div class="row form-group">
        <label class="col-sm-3 form-control-label"><?php echo db_lang('lokasi_id') ?></label>
        <div class="col-sm-9">
            <?php echo form_input('lokasi_id', $row->lokasi_id ? $row->lokasi_id : $lokasi_id, 'class="form-control" id="lokasi_id" placeholder="'. db_lang('lokasi_id') .'" readonly') ?>
        </div>
    </div>

    <div class="row form-group">
        <label class="col-sm-3 form-control-label"><?php echo db_lang('area_parkir') ?></label>
        <div class="col-sm-9">
            <?php echo form_input('area_parkir', $row->area_parkir, 'class="form-control" id="area_parkir" placeholder="'. db_lang('area_parkir') .'"') ?>
        </div>
    </div>

    <div class="row form-group">
        <label class="col-sm-3 form-control-label"><?php echo db_lang('alamat') ?></label>
        <div class="col-sm-9">
            <?php echo form_input('alamat', $row->alamat, 'class="form-control" id="alamat" placeholder="'. db_lang('alamat') .'"') ?>
        </div>
    </div>

    <div class="row form-group">
        <label class="col-sm-3 form-control-label"><?php echo db_lang('provinsi') ?></label>
        <div class="col-sm-3">
            <?php
                $select = array(" => lang('select.pick')) + $provinsi;
                echo form_dropdown('provinsi', $select, $row->provinsi, 'class="form-control" id="provinsi" onchange="get_kota(this.value)");
            ?>
        </div>
        <label class="col-sm-3 form-control-label"><?php echo db_lang('kab_kota') ?></label>
        <div class="col-sm-3">

```

```

        <div id="get_kota">
            <?php
                $select = array(" => lang('select.pick')) + $kota;
                echo form_dropdown('kab_kota', $select, $row->kab_kota,
'class="form-control"                                id="kab_kota"
onchange="get_kecamatan(this.value)");
            ?>
        </div>
    </div>
</div>

    <div class="row form-group">
        <label class="col-sm-3 form-control-label"><?php echo
db_lang('kecamatan') ?></label>
        <div class="col-sm-3">
            <div id="get_kecamatan">
                <?php
                    $select = array(" => lang('select.pick')) + $kecamatan;
                    echo form_dropdown('kecamatan', $select, $row-
>kecamatan, 'class="form-control" id="kecamatan"
onchange="get_kelurahan(this.value)");
                ?>
            </div>
        </div>
        <label class="col-sm-3 form-control-label"><?php echo
db_lang('kelurahan') ?></label>
        <div class="col-sm-3">
            <div id="get_kelurahan">
                <?php
                    $select = array(" => lang('select.pick')) + $kelurahan;
                    echo form_dropdown('kelurahan', $select, $row->kelurahan,
'class="form-control" id="kelurahan" ');
                ?>
            </div>
        </div>
    </div>

    <div class="row form-group">
        <label class="col-sm-3 form-control-label"><?php echo
db_lang('kode_pos') ?></label>
        <div class="col-sm-3">
            <?php echo form_input('kode_pos', $row->kode_pos,
'class="form-control" id="kode_pos" placeholder="'.
db_lang('kode_pos') .'"') ?>
        </div>
    </div>

```

```

        <div class="row form-group">
            <label class="col-sm-3 form-control-label"><?php echo
db_lang('jenis') ?></label>
            <div class="col-sm-9">
                <?php
                $select = array(" => lang('select.pick')) + $jenis_lokasi;
                echo form_dropdown('jenis', $select, $row->jenis,
'class="form-control" id="jenis");
                ?>
            </div>
        </div>

        <div class="row form-group">
            <label class="col-sm-3 form-control-label"><?php echo
db_lang('target_harian') ?></label>
            <div class="col-sm-3">
                <?php echo form_input('target_harian', $row->target_harian,
'class="form-control" id="target_harian" placeholder="'.
db_lang('target_harian') .'"') ?>
            </div>
        </div>

        <div class="row form-group">
            <div class="col-sm-9 col-sm-push-3">
                <button type="submit" class="btn btn-primary submit"><?php
echo lang('buttons.save') ?></button>
                <button type="reset" class="btn btn-warning"><?php echo
lang('buttons.reset') ?></button>
                <button class="btn btn-danger cancel"><?php echo
lang('buttons.cancel') ?></button>
            </div>
        </div>
        <?php echo form_close() ?>
</div>

```

Login android

```

import android.Manifest
import android.annotation.SuppressLint
import android.app.Dialog
import android.content.Context

```

```

import android.content.pm.PackageManager
import android.graphics.Color
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.telephony.TelephonyManager
import android.text.TextUtils
import androidx.appcompat.app.AlertDialog
import androidx.appcompat.app.AlertDialog.*
import androidx.core.app.ActivityCompat
import com.google.gson.Gson
import com.ssp.smartparking.R
import com.ssp.smartparking.data.ApiService
import com.ssp.smartparking.data.request.LoginRequestModel
import com.ssp.smartparking.internal.NoConnectivityException
import com.ssp.smartparking.utils.UtilsPref
import com.ssp.smartparking.utils.UtilsProject
import kotlinx.android.synthetic.main.activity_login.*
import kotlinx.android.synthetic.main.dialog_iddevice.view.*
import kotlinx.coroutines.*
import org.jetbrains.anko.*
import org.kodein.di.Kodein
import org.kodein.di.KodeinAware
import org.kodein.di.android.closestKodein
import org.kodein.di.generic.instance
import kotlin.coroutines.CoroutineContext

class LoginActivity : AppCompatActivity(), CoroutineScope,
KodeinAware {
    override val kodein: Kodein by closestKodein()
    private val apiService : ApiService by instance()
    private lateinit var job: Job
    private var showDialog = false
    override val coroutineContext: CoroutineContext
        get() = job + Dispatchers.Main
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_login)
        job = Job()
        getImei()
        if(UtilsPref.isLoggedIn(baseContext)){
            startActivity<MainActivity>()
            finish()
        }
        buttonLogin.setOnClickListener {
            if(TextUtils.isEmpty(etUsername.text.toString())
            TextUtils.isEmpty(etPassword.text.toString())){

```

```

        alert(getString(R.string.isianbelum lengkap)) {
            positiveButton("Ok") {

                }
            }.show().apply {
                getButton(BUTTON_POSITIVE)?.let { it.textColor =
Color.WHITE }
            }
            return@setOnClickListener
        }
        val params = LoginRequestModel(
            etUsername.text.toString(),
            etPassword.text.toString(),
            UtilsPref.getImei(baseContext)
        )
        val dialog = Dialog(this@LoginActivity,
R.style.DialogBounceAnim)
        dialog setContentView(R.layout.progress_dialog)
        dialog.show()
        launch(UtilsProject.handler) {
            try{
                val response = apiService.authLogin(params).await()
                withContext(Dispatchers.Main){
                    dialog.dismiss()
                    if(response.success == true){
                        UtilsPref.saveUser(baseContext,
Gson().toJson(response.data))
                        UtilsPref.saveUsername(baseContext,
etUsername.text.toString())
                        UtilsPref.saveLogin(baseContext, true)
                        UtilsPref.saveIdDevice(baseContext,
response?.data?.idDevice.toString())
                        startActivity<MainActivity>()
                        finish()
                    }else{
                        alert(
response.alert?.get(0)
                        ?:
getString(R.string.usernameataupasswordsalah)) {
                            positiveButton("Ok") {

                                }
                            }.show().apply {
                                getButton(BUTTON_POSITIVE)?.let { it.textColor
= Color.WHITE }
                            }
                        }
                    }
                }
            }
        }
    }
}

```

```

        }catch (e:NoConnectivityException){
            dialog.dismiss()
            toast(getString(R.string.errorconnection))
        }catch (e:Exception){
            dialog.dismiss()
            toast(getString(R.string.terjadikesalahan))
        }
    }
}

//ini sementara nanti pindah ke menu navigation
buttonLogout.setOnClickListener {
    launch(UtilsProject.handler) {
        val response = apiService.authLogout().await()
        withContext(Dispatchers.Main){

        }
    }
}

private fun getImei(){
    if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.READ_PHONE_STATE)
        != PackageManager.PERMISSION_GRANTED) {
        getRequestPermission()
    }else{
        getImeiId()
    }
}

private fun getRequestPermission(){
    if (ActivityCompat.shouldShowRequestPermissionRationale(this,
Manifest.permission.READ_PHONE_STATE)) {
        AlertDialog.Builder(this@LoginActivity)
            .setTitle("Permission Request")

        .setMessage(getString(R.string.permission_read_phone_state_rationale)
        )

        .setCancelable(false)
        .setPositiveButton(android.R.string.yes) { dialog, which ->
            ActivityCompat.requestPermissions(
                this@LoginActivity,
                arrayOf(Manifest.permission.READ_PHONE_STATE),
                88
            )
        }
    }
}

```



```

        }
        .show()
    } else {
        ActivityCompat.requestPermissions(
            this, arrayOf(Manifest.permission.READ_PHONE_STATE),
            88
        )
    }
}

override fun onRequestPermissionsResult(requestCode: Int,
permissions: Array<out String>, grantResults: IntArray) {
    super.onRequestPermissionsResult(requestCode, permissions,
grantResults)
    if (requestCode == 88) {
        if (grantResults.count() == 1 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
            getImeiId()
        } else {
            getRequestPermission()
        }
    }
}

@SuppressLint("MissingPermission")
private fun getImeiId() {
    val tm = getSystemService(Context.TELEPHONY_SERVICE) as
TelephonyManager
    val deviceId = tm.deviceId
    UtilsPref.saveImei(baseContext, deviceId)
}

override fun onBackPressed() {
    if(!showDialog){
        finish()
    }
}

private fun getIdDevice(){
    showDialog = !showDialog
    val dialog = AlertDialog.Builder(this@LoginActivity,
R.style.myDialog).create()
    val dialogView = layoutInflater.inflate(R.layout.dialog_iddevice,
null)
    dialog.setView(dialogView)
}

```

```

        dialog.setCancelable(false)
        dialog.setCanceledOnTouchOutside(false)
        dialogView.buttonBatal.setOnClickListener {
            dialog.dismiss()
            finish()
        }
        dialogView.buttonYa.setOnClickListener {
            if(dialogView.etIdDevice.text.toString() == ""){
                dialogView.etIdDevice.error = "Id Device Harus Diisi"
            }else{
                showDialog = !showDialog
                dialog.dismiss()
                UtilsPref.saveIdDevice(baseContext,
                dialogView.etIdDevice.text.toString())
            }
        }
        dialog.show()
    }

    override fun onDestroy() {
        super.onDestroy()
        job.cancel()
    }
}
}

```

Sunmi trans

```

package com.sunmi.trans;
import android.os.Parcel;
import android.os.Parcelable;

public class TransBean implements Parcelable {

    private byte type = 0;
    private String text = "";
    private byte[] data = null;
    private int datalength = 0;

    public TransBean(){
        type = 0;
        data = null;
        text = "";
        datalength = 0;
    };
}

```

```
public byte getType() {
    return type;
}

public void setType(byte type) {
    this.type = type;
}

public String getText() {
    return text;
}

public void setText(String text) {
    this.text = text;
}

public byte[] getData() {
    return data;
}

public void setData(byte[] data) {
    if(data != null){
        datalength = data.length;
        this.data = new byte[datalength];
        System.arraycopy(data, 0, this.data, 0, datalength);
    }
}

public TransBean(Parcel source){
    this.type = source.readByte();
    this.datalength = source.readInt();
    this.text = source.readString();
    if(datalength > 0){
        this.data = new byte[datalength];
        source.readByteArray(data);
    }
}

public TransBean(byte type, String text, byte[] data){
    this.type = type;
    this.text = text;
    if(data != null){
        this.datalength = data.length;
        this.data = new byte[datalength];
        System.arraycopy(data, 0, this.data, 0, datalength);
    }
}
```

```

    }
}

@Override
public int describeContents() {
    return 0;
}

@Override
public void writeToParcel(Parcel dest, int flags) {
    dest.writeByte(type);
    dest.writeInt(datalength);
    dest.writeString(text);
    if(data != null){
        dest.writeByteArray(data);
    }
}

public static Creator<TransBean> CREATOR = new
Creator<TransBean>(){

    @Override
    public TransBean createFromParcel(Parcel source) {
        return new TransBean(source);
    }

    @Override
    public TransBean[] newArray(int size) {
        return new TransBean[size];
    }
};
}

```

QR Activity

```

import android.app.Activity

import android.content.Intent

import android.hardware.Camera

import android.os.AsyncTask

```

```
import android.os.Bundle

import android.os.Handler

import android.os.Vibrator

import android.util.Log

import android.view.SurfaceHolder

import android.view.SurfaceView

import android.widget.ToggleButton

import androidx.appcompat.app.AppCompatActivity

import com.ssp.smartparking.R

import com.ssp.smartparking.utils.sunmiutils.FinderView

import com.ssp.smartparking.utils.sunmiutils.SoundUtils

import com.sunmi.scan.Config

import com.sunmi.scan.Image

import com.sunmi.scan.ImageScanner

import com.sunmi.scan.Symbol

class QRCodeActivity : AppCompatActivity(), SurfaceHolder.Callback {

    private var mCamera: Camera? = null

    private var mCameraParameters: Camera.Parameters? = null

    private var mHolder: SurfaceHolder? = null

    private var surface_view: SurfaceView? = null
```

```
private var scanner: ImageScanner? = null//声明扫描器

private var autoFocusHandler: Handler? = null

private var asyncDecode: AsyncDecode? = null

internal var soundUtils: SoundUtils? = null

private var vibrate: Boolean = false

// private String username;

var decode_count = 0

private var finder_view: FinderView? = null

// private TextView textview;

private var torch_button: ToggleButton? = null

override fun onCreate(savedInstanceState: Bundle?) {

    super.onCreate(savedInstanceState)

    setContentView(R.layout.activity_scan_sunmi)

    // mSharedPreferences = getSharedPreferences("auth",
Context.MODE_PRIVATE);

    //

    // final Bundle extras = getIntent().getExtras();

    // username = extras.getString("username");
```

```
init()
}

private fun init() {
    surface_view = findViewById(R.id.surface_view) as SurfaceView
    finder_view = findViewById(R.id.finder_view) as FinderView
    //    textview = (TextView) findViewById(R.id.textview);
    torch_button = findViewById(R.id.torch_button)
    mHolder = surface_view!!.holder

    mHolder!!.setType(SurfaceHolder.SURFACE_TYPE_PUSH_BUFFER
S)

    mHolder!!.addCallback(this)

    scanner = ImageScanner()//创建扫描器

    scanner!!.setConfig(0, Config.X_DENSITY, 2)//行扫描间隔

    scanner!!.setConfig(0, Config.Y_DENSITY, 2)//列扫描间隔

    scanner!!.setConfig(0, Config.ENABLE_MULTILESYMS, 0)//是
否开启同一幅图一次解多个条码,0表示只解一个, 1为多个
```

```

scanner!!.setConfig(0, Config.ENABLE_INVERSE, 0)//是否解反
色的条码

scanner!!.setConfig(Symbol.PDF417, Config.ENABLE, 0)//是否禁
止PDF417码 · 默认开启

autoFocusHandler = Handler()
asyncDecode = AsyncDecode()
decode_count = 0

torch_button!!.setOnCheckedChangeListener { buttonView,
isChecked ->
    if (isChecked) {
        mCameraParameters!!.flashMode =
Camera.Parameters.FLASH_MODE_TORCH
        mCamera!!.parameters = mCameraParameters
    } else {
        mCameraParameters!!.flashMode =
Camera.Parameters.FLASH_MODE_OFF
        mCamera!!.parameters = mCameraParameters
    }
}
}
}

```



```
override fun surfaceCreated(holder: SurfaceHolder) {  
    try {  
        mCamera = Camera.open()  
        mCameraParameters = mCamera!!.parameters  
    } catch (e: Exception) {  
        Log.d("DBG", "surfaceCreated: " + e.message)  
        mCamera = null  
    }  
}  
  
override fun surfaceChanged(holder: SurfaceHolder, format: Int,  
width: Int, height: Int) {  
    if (mHolder!!.surface == null) {  
        return  
    }  
    try {  
        mCamera!!.stopPreview()  
    } catch (e: Exception) {  
    }  
}
```

```
try {  
  
    mCamera!!.setDisplayOrientation(90)//竖屏显示  
  
    mCamera!!.setPreviewDisplay(mHolder)  
  
    mCamera!!.setPreviewCallback(previewCallback)  
  
    mCamera!!.startPreview()  
  
    mCamera!!.autoFocus(autoFocusCallback)  
  
} catch (e: Exception) {  
  
    Log.d("DBG", "Error starting camera preview: " + e.message)  
  
}  
  
}  
  
override fun surfaceDestroyed(holder: SurfaceHolder) {  
  
    if (mCamera != null) {  
  
        mCamera!!.setPreviewCallback(null)  
  
        mCamera!!.release()  
  
        mCamera = null  
  
    }  
  
}  
  
/**  
  
* 预览数据
```

```

        */

        internal var previewCallback: Camera.PreviewCallback =
Camera.PreviewCallback { data, camera ->
    if (asyncDecode!!.isStoped) {
        val parameters = camera.parameters

        val size = parameters.previewSize//获取预览分辨率

        //创建解码图像，并转换为原始灰度数据，注意图片是被旋
转了90度的

        val source = Image(size.width, size.height, "Y800")
        val scanImageRect =
finder_view!!.getScanImageRect(size.height, size.width)

        //图片旋转了90度，将扫描框的TOP作为left裁剪

        source.setCrop(scanImageRect.top, scanImageRect.left,
scanImageRect.height(), scanImageRect.width())

        source.data = data//填充数据

        asyncDecode = AsyncDecode()

        asyncDecode!!.execute(source)//调用异步执行解码

    }
}

```

```
private inner class AsyncDecode : AsyncTask<Image, String,
String>() {
    var isStoped = true
    private set
    private var str: String? = ""

    override fun doInBackground(vararg params: Image): String? {
        isStoped = false
        val sb = StringBuilder()
        val src_data = params[0]//Get grayscale data

        val startMillis = System.currentTimeMillis()

        //Decode, return value is 0 for failure, >0 for success
        val nsyms = scanner!!.scanImage(src_data)

        val endMillis = System.currentTimeMillis()
        val cost_time = endMillis - startMillis

        if (nsyms != 0) {
            println("TEST TEST MASUK")
        }
    }
}
```

```
playBeepSoundAndVibrate()//解码成功播放提示音

decode_count++

//      sb.append("计数: " + String.valueOf(decode_count)

+ ", 耗时: " + String.valueOf(cost_time) + " ms \n" );

val syms = scanner!!.getResults()//获取解码结果

for (sym in syms) {

    //      sb.append("[ " + sym.getSymbolName() + " ]:

" + sym.getResult() + "\n");

    sb.append(sym.result)

    println("CEK RESULT ${sym.result}")

}

}

str = sb.toString()

return str

}

override fun onPostExecute(result: String) {

    super.onPostExecute(result)

    println("CEK RESULT QR $str")

    isStoped = true
```

```
        if (null == str || str == "") {

        } else {

            println("CEK QR ${str}")

            val intent = Intent()

            intent.putExtra("data", str)

            setResult(Activity.RESULT_OK, intent)

            finish()

        }

    }

}

/**

 * 自动对焦回调

 */

internal var autoFocusCallback: Camera.AutoFocusCallback? =

    Camera.AutoFocusCallback { success, camera ->

autoFocusHandler!!.postDelayed(doAutoFocus, 1000) }

//自动对焦

private val doAutoFocus = Runnable {

    if (null == mCamera || null == autoFocusCallback) {
```

```
        return@Runnable
    }

    mCamera!!.autoFocus(autoFocusCallback)
}

private fun initBeepSound() {
    if (soundUtils == null) {
        soundUtils = SoundUtils(this, SoundUtils.RING_SOUND)
        soundUtils!!.putSound(0, R.raw.beep)
    }
}

override fun onResume() {
    // TODO Auto-generated method stub
    super.onResume()
    initBeepSound()
    vibrate = false
}

private val VIBRATE_DURATION = 200L
private fun playBeepSoundAndVibrate() {
    if (soundUtils != null) {
        soundUtils!!.playSound(0, SoundUtils.SINGLE_PLAY)
    }
}
```

```
    }  
    if (vibrate) {  
        val vibrator = getSystemService(VIBRATOR_SERVICE) as  
Vibrator  
        vibrator.vibrate(VIBRATE_DURATION)  
    }  
}  
}
```