



TUGAS AKHIR - KI091391

PERANCANGAN SISTEM KEAMANAN TEMPAT PARKIR MENGGUNAKAN SMART GATE

FANDIASA KORUMA KOSTRADA
NRP 5109100054

Dosen Pembimbing I
Ary Mazharuddin S., S.Kom., M.Comp.Sc

Dosen Pembimbing II
Tohari Ahmad, S.Kom., MIT., Ph.D.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2014



FINAL PROJECT - KI091391

***DESIGNING PARKING LOT SECURITY SYSTEM
USING SMART GATE***

**FANDIASA KORUMA KOSTRADA
NRP 5109100054**

**First Advisor
Ary Mazharuddin S., S.Kom., M.Comp.Sc**

**Second Advisor
Tohari Ahmad, S.Kom., MIT., Ph.D.**

**DEPARTMENT OF INFORMATICS
Faculty of Information Technology
Sepuluh Nopember Institute of Technology
Surabaya 2014**

PERANCANGAN SISTEM KEAMANAN TEMPAT PARKIR MENGGUNAKAN SMART GATE

Nama Mahasiswa : Fandiasa Koruma Kostrada
NRP : 5109100054
Jurusan : Teknik Informatika FTIF -ITS
Dosen Pembimbing 1 : Ary Mazharuddin S., S.Kom.,
M.Comp.Sc
Dosen Pembimbing 2 : Tohari Ahmad, S.Kom., MIT., Ph.D.

ABSTRAK

Suatu keamanan memang harus dituntut tingkat keamanan yang lebih tinggi khususnya dalam hal ini adalah sistem keamanan tempat parkir. Adanya gangguan – gangguan dari berbagai sistem tempat parkir dewasa ini dapat menyebabkan turunnya tingkat keamanan dari tempat parkir saat ini. Salah satu gangguan yg pada saat ini sedang heboh adalah kasus pencurian kendaraan dengan memanfaatkan kemudahan untuk memalsukan identitas kendaraan sehingga penjaga tempat parkir terkecoh dan tidak bisa mendeteksi kepemilikan dari kendaraan di tempat parkir tersebut.

Berdasarkan latar belakang tersebut maka penulis merancang sebuah sistem aplikasi berbasis desktop yang diberi nama “Smart Gate” untuk memudahkan para penyeia jasa tempat parkir untuk meningkatkan keamanan di tempat parkir. Pada tugas akhir ini dirancang sebuah aplikasi dengan memanfaatkan 2 buah alat yaitu RFID (Radio Frequency Identification) dan Fingerprint Reader beserta 1 buah alat yaitu NFC (Near Field Communication) yang digunakan sebagai backup apabila terjadi kerusakan atau kesalahan pada alat RFID. Pada Aplikasi ini dibutuhkan masukan berupa informasi mengenai identitas pengendara dan identitas kendaraan dari pengguna jasa parkir dengan menggunakan bantuan dari ke-tiga alat tersebut yang kemudian akan di olah dan disesuaikan dengan database yang sudah tersedia. Aplikasi akan

menampilkan hasil proses matching atau penyamaan antara data masukan dan data yang tersimpan di database.

Dari hasil perancangan aplikasi Tugas Akhir ini didapatkan sebuah aplikasi yang dapat mengurangi tingkat kriminalitas dalam hal pencurian kendaraan bermotor khususnya di tempat parkir.

Kata kunci: Smart Gate, Parking system, RFID, Fingerprint Reader, NFC

DESIGNING PARKING LOT SECURITY SYSTEM USING SMAR GATE

Name : Fandiasa Koruma Kostrada
NRP : 5109100054
Departement : Informatics FTIF -ITS
Supervisor 1 : Ary Mazharuddin S., S.Kom., M.Comp.Sc
Supervisor 2 : Tohari Ahmad, S.Kom., MIT., Ph.D.

ABSTRACT

Nowadays, a lot of crimes happen in the parking area. Therefore we need to improve our secure system especially for parking area. So we can prevent bad guy who wants to steal our car by easily faking our car identity to make the guards deceived and they can't check true owner of the vehicle.

Due to that background, writer design a desktop based application system named "Smart Gate" to ease parking service provider to improve their security in parking area. This final project will design and build application that utilize 2 tools: RFID (Radio Frequency Identification) with Fingerprint Reader and NFC (Near Field Communication) as backup if there is any broken or error in RFID. This application needs input of information about identity of driver and the vehicle in parking area. By using RFID, Fingerprint Reader, and NFC that will be adapted with available database, this application will display matching process which will compare data input and data in database.

This final project will create application that can reduce crime level in stealing vehicle especially in parking area.

Keyword: Smart Gate, Parking system, RFID, Fingerprint Reader, NFC

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN
PERANCANGAN SISTEM KEAMANAN TEMPAT
PARKIR MENGGUNAKAN SMART GATE

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada

Bidang Studi Komputasi Berbasis Jaringan
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

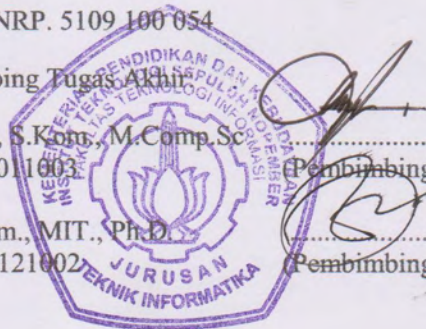
Oleh

Fandiasa Koruma Kostrada

NRP. 5109 100 054

Disetujui oleh Pembimbing Tugas Akhir

1. Ary Mazharuddin S., S.Kom., M.Comp.Sc
NIP: 198106202005011003 (Pembimbing 1)
2. Tohari Ahmad, S.Kom., MIT., Ph.D.
NIP: 197505252003121002 (Pembimbing 2)



SURABAYA

JUNI, 2014

KATA PENGANTAR

Alhamdulillahirabbil'alamin, segala puji bagi Allah SWT, yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis bisa menyelesaikan Tugas Akhir yang berjudul:

“Perancangan Sistem Keamanan Tempat Parkir Menggunakan Smart Gate”

Pengerjaan Tugas Akhir ini merupakan suatu kesempatan yang sangat berharga bagi penulis, karena dengan pengerjaan Tugas Akhir ini, penulis bisa memperdalam, meningkatkan, serta mengimplementasikan apa yang telah didapatkan penulis selama menempuh perkuliahan di Teknik Informatika ITS.

Dengan mengucapkan syukur kepada Allah SWT, dengan ini penulis hendak menyampaikan penghormatan dan terimakasih yang sebesar-besarnya kepada pihak-pihak yang telah memberikan dukungan secara langsung maupun tidak langsung antara lain kepada:

1. Allah SWT, atas limpahan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan tugas akhir dengan baik.
2. Kedua orang tua penulis tugas akhir, bapak Samuji. dan ibu Dyah Setyaning, S.Pd.,M.Pd. yang selalu mencurahkan kasih sayang, dukungan, semangat, dan nasihat kepada penulis. Terima kasih atas petuah papa dan mama yang selalu mengingatkan penulis bahwa ikhtiar dan doa adalah dua hal yang tidak dapat dipisahkan.
3. Kakak penulis, Pegy Pradika Mudayanti, S.Farm., Apt. yang selalu memberikan dukungan dan semangat kepada penulis.
4. Keluarga besar Bapak dan Ibu penulis.
5. Bapak Ibu Guru penulis semasa mengenyam bangku sekolah SD, SMP, dan SMA.

6. Bapak Ary Mazharuddin S., S.Kom., M.Comp.Sc selaku Dosen Pembimbing 1, yang telah memberikan kepercayaan, motivasi, bimbingan, dukungan, nasihat, serta semangat dikala penulis mengalami kesulitan dalam pengerjaan tugas akhir.
7. Bapak Tohari Ahmad, S.Kom., MIT., Ph.D. selaku Dosen Pembimbing 2, yang telah memberikan kepercayaan, motivasi, bimbingan, dukungan, nasihat, serta semangat dikala penulis mengalami kesulitan dalam pengerjaan tugas akhir.
8. Ibu Dr.Eng.Nanik Suciati, S.Kom., M.Kom selaku Kepala Jurusan Teknik Informatika ITS dan bapak Dwi Sunaryono, S.Kom., M.Kom selaku Sekretaris Jurusan Teknik Informatika ITS. Terima kasih memberikan pembelajaran hidup kepada penulis selama penulis menjalani kuliah di jurusan Teknik Informatika ITS.
9. Bapak Ary Mazharuddin Shiddiqie, S.Kom., M.Comp.Sc selaku dosen wali penulis. Terima kasih atas dukungan, nasihat, dan pembelajaran hidup yang diberikan kepada penulis selama penulis menjalani kuliah di jurusan Teknik Informatika ITS.
10. Ibu Erina Letivina Anggraini, S.Kom., M.Kom. selaku dosen pembimbing kerja praktik penulis. Terima kasih atas nasihat, motivasi, dan pembelajaran hidup yang diberikan kepada penulis selama penulis kerja praktik maupun selama menjalani kuliah di jurusan Teknik Informatika, Fakultas Teknologi Informasi ITS.

11. Bapak Radityo Anggoro, S.Kom., M.Sc selaku koordinator tugas akhir jurusan Teknik Informatika ITS. Terima kasih atas semangat, dukungan, kepercayaan, dan evaluasi yang diberikan dikala penulis menyelesaikan tugas akhir.
12. Dosen-dosen jurusan Teknik Informatika ITS yang telah memberikan ilmu, nasihat, semangat, dan pembelajaran hidup kepada penulis selama menjalani kuliah di Jurusan Teknik Informatika ITS.
13. Bapak Yudi Mulyono, Bapak Sugeng, Ibu Suwartani, Bapak Supri, Mbak Fatin, Mbak Icha, Mas Jumali, Mas Hari, dan segenap karyawan tata usaha jurusan Teknik Informatika ITS, terima kasih atas segala pertolongan, pelayanan, dukungan, nasihat, dan petuah yang diberikan kepada penulis selama menjalani kuliah di jurusan Teknik Informatika ITS.
14. Yusuf Kurniawan dan Yanuar teman penulis yang tselantiasa membantu penulis dalam menyelesaikan Tugas Akhir apabila penulis mengalami kesulitan.
15. Sahabat penulis selama tinggal di kota pahlawan, Falah Egy sujana, Adib Nur Ikhwan, Bangkit W, Handhika P, Ahmad Afif, Aulianoor Adi, yang selalu memberikan nasihat dan dukungan satu sama lain. Terima kasih sahabat-sahabat hebatku, semoga kita tetap terus menjaga silaturahmi dan sukses di tempat kerja masing-masing.
16. Sahabat penulis sejak berjuang di jurusan Teknik Informatika ITS, Galih, Hafidh, Uut, Erlangga, Ucup, Irwan, Fadhil, Haki, Reza, Guntur, Imam, Andreyan

yang selalu memberikan nasihat dan dukungan satu sama lain, yang selalu membawa keceriaan kepada penulis selama empat tahun ini.

17. Teman-teman seperjuangan bidang minat NCC, Adon, Gagan, Haki, Guntur, terima kasih atas bantuan, dukungan, dan pembelajaran yang diberikan kepada penulis selama menjalani kuliah pada bidang minat NCC.
18. Kakak-kakak angkatan 2006, 2007, 2008, serta adik-adik angkatan 2010, 2011, dan 2012, terima kasih atas kerja sama dan dukungannya selama ini kepada penulis selama menjalani kuliah .
19. Kepada semua pihak yang belum sempat disebutkan, penulis mengucapkan terima kasih yang sebesar-besarnya atas pertolongan rekan-rekan sekalian dalam membantu penulis agar terselesaikannya tugas akhir ini.

Tentu saja penulis menyadari bahwa Tugas Akhir ini jauh dari kesempurnaan dan masih terdapat beberapa kekurangan. Oleh karena itu, dengan segala kerendahan hati, penulis mengharapkan saran dan kritik dari pembaca.

Bismillahirrahmanirrahim, inilah karya persembahkan penulis hasil dari perkuliahan di jurusan Teknik Informatika ITS selama empat tahun.

Surabaya, Juni 2014

DAFTAR ISI

LEMBAR PENGESAHAN	v
ABSTRAK	vii
KATA PENGANTAR	xi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan	3
1.5 Metodologi.....	4
2. BAB II TINJAUAN PUSTAKA	7
2.1 RFID (<i>Radio Frequency Identification</i>)	7
2.2 NFC (<i>Near Field Communication</i>).....	9
2.3 <i>Fingerprint scanner</i>	10
2.4 <i>.NET</i>	11
2.5 <i>Microsoft Visual C#</i>	12
2.6 <i>MySQL</i>	13
3. BAB III PERANCANGAN DAN DESAIN SISTEM. ..	15
3.1 Deskripsi Umum	15
3.2 Arsitektur Umum Sistem.....	15
3.3 Perancangan Sistem	16
3.3.1 Kasus Penggunaan Mengelola Data Pengguna.....	20
3.3.2 Kasus Penggunaan Melihat Laporan	22
3.3.3 Kasus Penggunaan Menentukan Harga Parkir.....	23
3.3.4 Kasus Penggunaan Mengelola Data <i>Driver</i>	25
3.3.5 Kasus Penggunaan Pengelolaan Data Kendaraan.....	29
3.3.6 Kasus Penggunaan Mengelola Transaksi Masuk	31
3.3.7 Kasus Penggunaan Mengelola Transaksi Keluar.....	34
3.4 Rancangan Antarmuka Aplikasi.....	36
3.4.1 Rancangan Antarmuka Pembuatan Pengguna	36
3.4.2 Rancangan Antarmuka Melihat Laporan.....	37
3.4.3 Rancangan Antarmuka Menentukan Harga.....	37

3.4.4	Rancangan Antarmuka Mengelola Data <i>Driver</i>	38
3.4.5	Rancangan Antarmuka Mengelola Data Kendaraan	41
3.4.6	Rancangan Antarmukan Transaksi Parkir Masuk dan Keluar	43
4.	BAB IV IMPLEMENTASI APLIKASI.....	45
4.1	Lingkungan Implementasi	45
4.1.1	Lingkungan Perangkat Keras	45
4.1.2	Lingkungan Perangkat Lunak	46
4.2	Implementasi perangkat Keras	46
4.3	Implementasi Perangkat Lunak	48
4.3.1	Implementasi Data Pengguna Aplikasi.....	48
4.3.2	Implementasi Data Kendaraan	50
4.3.3	Implementasi Data Driver	52
4.3.4	Implementasi Transaksi Parkir Masuk dan Keluar	54
4.4	Implementasi Pengamanan Dalam Kartu	56
4.5	Implementasi Antarmuka	58
4.5.1	Implementasi Antarmuka Pembuatan Pengguna	58
4.5.2	Implementasi Antarmuka Melihat Laporan	59
4.5.3	Implementasi Antarmuka Menentukan Harga	59
4.5.4	Implementasi Antarmuka Mengelola Data <i>Driver</i>	60
4.5.5	Implementasi Antarmuka Mengelola Data Kendaraan	61
4.5.6	Implementasi Antarmuka Transaksi Parkir Masuk dan Keluar	62
4.6	Implementasi <i>Back up device</i>	63
5.	BAB V UJI COBA DAN ANALISA	65
5.1	Lingkungan Uji Coba	65
5.2	Skenario Uji Coba	66
5.2.1	Uji Coba <i>Login</i> ke Sistem <i>Smart Gate</i>	66
5.2.2	Uji Coba Pengolahan Pengguna	66
5.2.3	Uji Coba Pengolahan Data <i>Driver</i>	69
5.2.4	Uji Coba Pengolahan Data Kendaraan	71
5.2.5	Uji Coba Transaksi Parkir	75
5.2.6	Uji Coba Performa Sistem.....	78
6.	BAB VI KESIMPULAN DAN SARAN	81
6.1	Kesimpulan	81

6.2	Saran	81
7.	DAFTAR PUSTAKA	83
	BIODATA PENULIS	85

DAFTAR TABEL

Tabel 3.1. Kasus Penggunaan.....	17
Tabel 3.2. Pengelola Data Pengguna	20
Tabel 3.3. Melihat Laporan	22
Tabel 3.4. Menentukan Harga Parkir.....	24
Tabel 3.5. Mengelola Data <i>Driver</i>	26
Tabel 3.6. Mengelola Data Kendaraan	29
Tabel 3.7. Mengelola Transaksi Masuk.....	32
Tabel 3.8. Mengelola Transaksi Keluar.....	34
Tabel 5.1. Uji Coba Waktu Sistem Parkir <i>Smart Gate</i>	79

DAFTAR ISI

LEMBAR PENGESAHAN	v
ABSTRAK	vii
KATA PENGANTAR	xi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan	3
1.5 Metodologi.....	4
2. BAB II TINJAUAN PUSTAKA	7
2.1 RFID (<i>Radio Frequency Identification</i>)	7
2.2 NFC (<i>Near Field Communication</i>).....	9
2.3 <i>Fingerprint scanner</i>	10
2.4 <i>.NET</i>	11
2.5 <i>Microsoft Visual C#</i>	12
2.6 <i>MySQL</i>	13
3. BAB III PERANCANGAN DAN DESAIN SISTEM. ..	15
3.1 Deskripsi Umum	15
3.2 Arsitektur Umum Sistem.....	15
3.3 Perancangan Sistem	16
3.3.1 Kasus Penggunaan Mengelola Data Pengguna	20
3.3.2 Kasus Penggunaan Melihat Laporan	22
3.3.3 Kasus Penggunaan Menentukan Harga Parkir.....	23
3.3.4 Kasus Penggunaan Mengelola Data <i>Driver</i>	25
3.3.5 Kasus Penggunaan Pengelolaan Data Kendaraan	29
3.3.6 Kasus Penggunaan Mengelola Transaksi Masuk	31
3.3.7 Kasus Penggunaan Mengelola Transaksi Keluar	34
3.4 Rancangan Antarmuka Aplikasi.....	36
3.4.1 Rancangan Antarmuka Pembuatan Pengguna	36
3.4.2 Rancangan Antarmuka Melihat Laporan.....	37
3.4.3 Rancangan Antarmuka Menentukan Harga.....	37

3.4.4	Rancangan Antarmuka Mengelola Data <i>Driver</i>	38
3.4.5	Rancangan Antarmuka Mengelola Data Kendaraan	41
3.4.6	Rancangan Antarmukan Transaksi Parkir Masuk dan Keluar	43
4.	BAB IV IMPLEMENTASI APLIKASI.....	45
4.1	Lingkungan Implementasi	45
4.1.1	Lingkungan Perangkat Keras	45
4.1.2	Lingkungan Perangkat Lunak	46
4.2	Implementasi perangkat Keras	46
4.3	Implementasi Perangkat Lunak	48
4.3.1	Implementasi Data Pengguna Aplikasi.....	48
4.3.2	Implementasi Data Kendaraan	50
4.3.3	Implementasi Data Driver	52
4.3.4	Implementasi Transaksi Parkir Masuk dan Keluar	54
4.4	Implementasi Pengamanan Dalam Kartu	56
4.5	Implementasi Antarmuka	58
4.5.1	Implementasi Antarmuka Pembuatan Pengguna	58
4.5.2	Implementasi Antarmuka Melihat Laporan	59
4.5.3	Implementasi Antarmuka Menentukan Harga	59
4.5.4	Implementasi Antarmuka Mengelola Data <i>Driver</i>	60
4.5.5	Implementasi Antarmuka Mengelola Data Kendaraan	61
4.5.6	Implementasi Antarmuka Transaksi Parkir Masuk dan Keluar	62
4.6	Implementasi <i>Back up device</i>	63
5.	BAB V UJI COBA DAN ANALISA	65
5.1	Lingkungan Uji Coba	65
5.2	Skenario Uji Coba	66
5.2.1	Uji Coba <i>Login</i> ke Sistem <i>Smart Gate</i>	66
5.2.2	Uji Coba Pengolahan Pengguna	66
5.2.3	Uji Coba Pengolahan Data <i>Driver</i>	69
5.2.4	Uji Coba Pengolahan Data Kendaraan	71
5.2.5	Uji Coba Transaksi Parkir	75
5.2.6	Uji Coba Performa Sistem.....	78
6.	BAB VI KESIMPULAN DAN SARAN	81
6.1	Kesimpulan	81

6.2	Saran	81
7.	DAFTAR PUSTAKA	83
	BIODATA PENULIS	85

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 2.1. RFID Reader	8
Gambar 2.2. Tag RFID	8
Gambar 2.3. <i>Fingerprint Scanner</i>	10
Gambar 2.4. Arsitektur .NET	11
Gambar 3.1. Arsitektur Sistem	16
Gambar 3.2. Diagram Kasus	19
Gambar 3.3. Diagram Aktivitas Pengolahan Data Pengguna	21
Gambar 3.4. Diagram Aktivitas Memilih Laporan	23
Gambar 3.5. Diagram Aktivitas Menentukan Harga Parkir	25
Gambar 3.6. Diagram Aktivitas Mengelola Data <i>Driver</i>	28
Gambar 3.7. Diagram Aktivitas Mengelola Data Kendaraan	31
Gambar 3.8. Diagram Aktivitas Mengelola Transaksi Masuk	33
Gambar 3.9. Diagram Aktivitas Mengelola Transaksi Keluar	35
Gambar 3.10. Antarmuka Pembuatan Pengguna	36
Gambar 3.11. Antarmuka Melihat Laporan	37
Gambar 3.12. Antarmuka Menentukan Harga Parkir	38
Gambar 3.13. Antarmuka Penambahan Data <i>Driver</i>	39
Gambar 3.14. Antarmuka Penghapusan Data <i>Driver</i>	40
Gambar 3.15. Antarmuka Memerbarui Sidik Jari <i>Driver</i>	41
Gambar 3.16. Antarmuka Penambahan Data Kendaraan	42
Gambar 3.17. Antarmuka Detail Data Kendaraan	43
Gambar 3.18. Antarmuka Transaksi Parkir Masuk dan Keluar	44
Gambar 4.1. Perangkat keras	47
Gambar 4.2. Hak akses Pengguna Super Admin	49
Gambar 4.3. Hak Akses Pengguna Admin	49
Gambar 4.4. Hak Akses Pengguna Petugas	50
Gambar 4.5. Kartu Tag kendaraan	51
Gambar 4.6. <i>Database</i> kendaraan	51
Gambar 4.7. Arsitektur Memori Dalam Tag	52
Gambar 4.8. Perekaman Sidik Jari	53
Gambar 4.9. Relasi Tabel <i>Driver</i> dan Kendaraan	53
Gambar 4.10. <i>Pseudocode</i> Pembacaan Kartu Tag	54
Gambar 4.11. <i>Pseudocode</i> Pembacaan Sidik Jari	55

Gambar 4.12. <i>Pseudocode</i> Penulisan ke Kartu <i>Tag</i>	56
Gambar 4.15. Implementasi Antar Muka Pembuatan Pengguna	58
Gambar 4.16. Implementasi Antarmuka Laporan Parkir	59
Gambar 4.17. Implementasi Antarmuka Harga	60
Gambar 4.18. Implementasi Antarmuka <i>Driver</i>	61
Gambar 4.19. Implementasi Antarmuka Kendaraan	62
Gambar 4.20. Implementasi Antarmuka Parkir	63
Gambar 5.1. <i>Login</i> Pengguna	66
Gambar 5.2. Menu Pendaftaran Pengguna Baru	67
Gambar 5.3. Tampilan Detail pengguna	67
Gambar 5.4. Menu Pencarian Data Pengguna	68
Gambar 5.5. Menu Pendaftaran <i>Driver</i>	69
Gambar 5.6. Menu Detail <i>Driver</i>	69
Gambar 5.7. Menu Perekaman Sidik Jari	70
Gambar 5.8. Menu Pencarian <i>Driver</i>	71
Gambar 5.9. Menu Pendaftaran Kendaraan	72
Gambar 5.10. Menu Detail Kendaraan	72
Gambar 5.11. Menu Pengisian Data ke Kartu	73
Gambar 5.12. Menu Penambahan Kepemilikan Kendaraan	74
Gambar 5.13. Hasil Penambahan Kepemilikan	74
Gambar 5.14. Menu parkir masuk dan keluar.....	75
Gambar 5.15. Hasil Pembacaan Data	76
Gambar 5.16. Keberhasilan Proses Verifikasi	76
Gambar 5.17. Menu Proses Keluar parkir	77
Gambar 5.18. Laporan Riwayat Parkir	78

DAFTAR TABEL

Tabel 3.1. Kasus Penggunaan.....	17
Tabel 3.2. Pengelola Data Pengguna	20
Tabel 3.3. Melihat Laporan	22
Tabel 3.4. Menentukan Harga Parkir	24
Tabel 3.5. Mengelola Data <i>Driver</i>	26
Tabel 3.6. Mengelola Data Kendaraan	29
Tabel 3.7. Mengelola Transaksi Masuk.....	32
Tabel 3.8. Mengelola Transaksi Keluar.....	34
Tabel 5.1. Uji Coba Waktu Sistem Parkir <i>Smart Gate</i>	79

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

1.1 Latar Belakang

Suatu keamanan dewasa ini memang harus dituntut tingkat keamanan yang lebih tinggi khususnya dalam hal ini adalah sistem keamanan tempat parkir. Dikarenakan banyaknya kasus pencurian kendaraan bermotor yang terjadi saat ini, sehingga banyak sekali sistem keamanan tempat parkir yang ditawarkan salah satunya adalah sistem keamanan parkir dengan menggunakan *Smart Gate*. Aplikasi ini menggunakan bantuan 2 buah alat yaitu RFID (*Radio Frequency Identification*) dan *Fingerprint Reader*, beserta 1 buah alat yaitu *NFC Reader (Near Field Comunication)* sebagai alat cadangan atau *backup* apabila RFID mengalami kerusakan atau kegagalan.

Sistem keamanan ini sangat cocok digunakan sebagai media untuk menurunkan angka pencurian kendaraan bermotor yang sering terjadi akibat kelalaian petugas atau masih lemahnya sistem keamanan di tempat parkir. Oleh karena itu dikembangkan seperangkat alat sebagai sistem keamanan parkir tersebut. Sistem keamanan ini diintegrasikan dengan *database* RFID dari setiap kendaraan bermotor dan *database* sidik jari dari pemiliknya. Sistem ini mampu mengecek setiap kendaraan yang masuk maupun keluar dengan cara memasukkan identitas pengguna dan identitas kendaraan (berupa sidik jari dan nomor kendaraan), jika masukkan benar maka menghasilkan sinyal bahwa pintu portal bisa dibuka.

Masukkan pertama kali berasal dari sebuah alat RFID dimana setiap pengguna yang sudah terdaftar memiliki *id card* yang bisa terpasang di kendaraan sebagai masukan identitas nomor kendaraan. Kemudian data tersebut dicocokkan dengan data yang berada di *database*, selanjutnya akan dilakukan pengecekan data tersebut apakah sudah ada di *database* komputer apa belum, jika sudah ada maka masukkan yang ke dua berasal dari sebuah alat *fingerprint reader* sebagai masukan identitas pemiliknya. Apabila

data sidik jari sesuai dengan identitas kendaraan maka akan ada indikator tertentu yang membuktikan data benar. Tetapi jika data pengguna dan data nomor kendaraan belum ada maka diharuskan mendaftar untuk mendapatkan *id card* baru dan memasukkan sidik jari yang nantinya data tersebut disimpan ke dalam *database server*. Kemudian pada bagian akhir proyek ini diharapkan keamanan dan kenyamanan para pengguna jasa parkir bisa lebih terjamin.

1.2 Rumusan Masalah

Rincian permasalahan yang dihadapi dalam pembuatan tugas akhir ini adalah bagaimana agar dapat meningkatkan sistem kerja dari tempat parkir guna tercapainya keamanan, nyaman, dan otomatisasi dari pengguna jasa tempat parkir maupun para pengendara yang menggunakan jasa tempat parkir.

1.3 Batasan Masalah

Permasalahan yang diajukan dalam pembuatan tugas akhir ini akan diberi batasan untuk membatasi lingkup pengerjaan tugas akhir. Batasan-batasan tersebut adalah sebagai berikut:

1. Masukan data merupakan *input* langsung dari pengendara kendaraan bermotor yang berupa masukan dari *id card*, kartu *tag* dan sidik jari yang sudah terdaftar terlebih dahulu.
2. Dalam hal ini PC yang digunakan berjumlah satu, hal ini dikarenakan untuk mempermudah dalam memonitor atau menjaga sehingga biaya yang diperlukan lebih efisien.
3. Proyek akhir ini hanya sebatas simulasi, sehingga output yang dihasilkan hanya berupa indikator tertentu untuk menentukan keberhasilan aplikasi tersebut.
4. Aplikasi dibangun menggunakan bahasa pemrograman C# dan *database mySQL Server*.
5. Aplikasi menggunakan *database* dalam 1 penyedia parkir, sehingga *database* dibangun menggunakan MySQL server di jaringan *localhost*.

6. *Hardware reader* yang digunakan sudah memakai SDK.
7. *Hardware* yang digunakan dapat menjangkau *tag* dengan maksimal 50cm.

1.4 Tujuan

Adapun tujuan dari pengerjaan tugas akhir ini adalah membuat rancang bangun aplikasi sistem keamanan dan pengamanan tempat parkir dengan menggunakan RFID dan *fingerprint reader* berbasis *desktop*. Dimana aplikasi ini nantinya akan mendapat data masukan dari sebuah *id card* dari masing-masing pengendara yang sudah terdaftar yang kemudian di *scan* menggunakan alat RFID sedangkan untuk pengganti *password* pengendara diharuskan memasukan sidik jari melalui *fingerprint reader*. Namun di dalam sistem ini disediakan sistem *link* atau *relation* untuk teman terdekat ataupun keluarga guna untuk mempermudah apabila kendaraan tersebut dipinjam atau digunakan oleh orang lain yang sudah dimasukkan ke dalam daftar relasi tersebut.

Selain untuk meningkat keamanan sistem ini dapat meningkatkan keefisienan waktu dan tenaga untuk para penyedia jasa tempat parkir dan para pengendara ketika akan memasuki tempat parkir. Dengan Sistem ini penyedia layanan dapat mengurangi jumlah tenaga pekerja pada pintu masuk parkir karena pada system ini sudah dilengkapi fitur otomatisasi yang disediakan berdasarkan alat yang digunakan.

Adapun manfaat dari pengerjaan tugas akhir ini adalah sebagai berikut:

1. Meningkatkan keamanan dan kenyamanan pengguna jasa tempat parkir.
2. Mempermudah petugas dalam memonitoring kendaraan karena semua *record* tersimpan ke dalam *database*.
3. Mempermudah pengguna tempat parkir dengan tanpa membawa STNK untuk masuk kedalam lokasi parkir.

4. Mempermudah penanganan petugas untuk pengecekan kendaraan ketika keluar masuk tempat parkir.
5. Menambah tingkat keefisienan waktu pada lokasi parkir karena system dapat melakukan pemrosesan secara cepat.

1.5 Metodologi

Ada beberapa tahap dalam proses pengerjaan tugas akhir ini. Berikut ini adalah tahap-tahap dalam pembuatan tugas akhir.

a. Studi Literatur

Tahap ini merupakan tahap pengumpulan informasi yang diperlukan untuk pengerjaan tugas akhir sekaligus mempelajarinya. Mulai dari pengumpulan literatur, diskusi, serta pemahaman topik tugas akhir antara lain pengambilan *data* dari beberapa alat yang digunakan seperti RFID, *fingerprint* dan NFC yang selanjutnya akan diproses untuk aplikasi keamanan tempat parkir tersebut.

b. Perancangan dan Desain Sistem

Pada tahap merancang suatu sistem terlebih dahulu dilakukan studi literatur dan mempelajari konsep teknologi *software* sehingga dapat dilakukan desain sistem dan desain proses-proses yang ada. Tahap ini merupakan tahap yang paling penting pada bentuk awal atau *prototype* yang akan diimplementasikan.

c. Implementasi

Pada tahap ini dilakukan implementasi rancangan sistem yang telah dibuat. Tahapan ini merealisasikan apa yang terdapat pada tahapan sebelumnya sehingga menjadi sebuah aplikasi dengan simulasi yang digunakan sesuai dengan apa yang telah direncanakan.

d. Uji Coba dan Evaluasi

Pada tahap ini aplikasi yang telah selesai dibuat akan diuji. Pengujian dan evaluasi akan dilakukan dengan melihat kesesuaian dan ketepatan sistem mengambil data dan

mengolahnya untuk *matching* terhadap data yang sudah tersedia, penulisan data informasi ke dalam *tag* RFID atau NFC, dan pembacaan sidik jari pengguna parkir.

e. Penyusunan Laporan Tugas Akhir

Pada tahap ini disusun laporan tugas akhir sebagai dokumentasi pelaksanaan tugas akhir yang mencakup seluruh konsep, teori, implementasi, dan hasil yang telah dikerjakan. Laporan tugas akhir ini akan dibagi menjadi beberapa bab sebagai berikut.

1. Bab I. Pendahuluan yang berisi latar belakang, permasalahan, tujuan, batasan permasalahan, metodologi, dan sistematika penulisan.
2. Bab II. Tinjauan Pustaka mencakupi dasar ilmu yang mendukung pembahasan tugas akhir ini.
3. Bab III. Desain Aplikasi yang akan dibuat pada tugas akhir ini sesuai dengan kebutuhan sistem.
4. Bab IV. Implementasi Aplikasi, akan dilakukan pembuatan aplikasi yang dibangun dengan komponen-komponen yang telah ada sesuai dengan permasalahan dan batasan yang telah dijabarkan pada bab pertama.
5. Bab V. Uji Coba dan Analisis Hasil, akan dilakukan uji coba fungsional dan akurasi berdasarkan parameter-parameter yang telah ditentukan.

[Halaman ini sengaja dikosongkan]

BAB II TINJAUAN PUSTAKA

Pada bab ini akan dibahas mengenai dasar teori yang mendukung penyusunan tugas akhir. Bab ini diawali dengan penjelasan tentang penjelasan dari masing-masing alat yang digunakan seperti RFID, NFC , *Fingerprint Scanner*, dan berbagai macam jenis pemrograman yang digunakan. Bagian-bagian tersebut akan dijelaskan ke dalam bentuk subbab masing-masing.

2.1 RFID (*Radio Frequency Identification*)

RFID adalah sebuah metode untuk mengidentifikasi dengan menggunakan sarana yang disebut label RFID atau *transponder* untuk menyimpan dan mengambil data jarak jauh. Label RFID ini bisa dipasang atau dimasukkan di dalam sebuah produk, hewan atau bahkan manusia dengan tujuan untuk identifikasi menggunakan gelombang radio dan berisi informasi yang disimpan secara elektronik dan dapat dibaca hingga beberapa meter jauhnya.

Sebuah sistem identifikasi frekuensi radio menggunakan *tag* atau label yang dipasang pada objek untuk diidentifikasi. Radio dua arah pemancar-penerima, dimana disebut sebagai pemeriksa atau pembaca, mengirimkan sinyal ke *tag* lalu membaca responnya. Umumnya, pembaca mengirimkan hasil pengamatan tersebut ke sistem komputer yang menjalankan perangkat lunak atau perangkat lunak tengah RFID. Perangkat keras RFID berbentuk seperti pembaca kartu yang memiliki penampang sebesar kartu *tag* pada umumnya seperti pada Gambar 2.1.



Gambar 2.1. RFID Reader

Informasi *tag* disimpan secara elektronik di dalam memori *non-volatile*. *Tag* RFID seperti pada Gambar 2.2. mencakup pemancar dan penerima frekuensi radio yang kecil. Sebuah pembaca RFID mengirimkan sinyal radio yang dikodekan untuk memeriksa *tag*. Lalu, *tag* menerima pesan dan merespon informasi yang diidentifikasinya. Ini mungkin hanya terjadi untuk *tag* dengan nomor seri khusus, atau mungkin untuk sebuah produk yang berkaitan dengan informasi seperti jumlah stok, lot atau nomor tumpak, tanggal produksi, atau informasi spesifik lainnya.



Gambar 2.2. Tag RFID

Sebuah label RFID dapat ditempelkan ke sebuah obyek dan digunakan untuk melacak dan mengelola inventaris, aset, orang, dan lain-lain. Sebagai contoh, label RFID bisa ditempelkan di mobil, peralatan komputer, buku-buku, ponsel, dan lain-lain.

RFID menawarkan keunggulan dibandingkan sistem manual atau penggunaan kode batang. Label dapat dibaca jika melewati dengan dekat pembaca label, bahkan jika pembaca tertutup oleh objek atau tidak terlihat sekalipun. Label dapat dibaca di dalam sebuah wadah, karton, kotak atau lainnya. Label RFID dapat membaca ratusan pada satu waktu, sedangkan kode batang hanya dapat dibaca satu per satu [1].

2.2 NFC (*Near Field Communication*)

Terdapat seperangkat teknologi konektivitas nirkabel berbasis teknologi *Radio Frequency Identification* (RFID) yaitu bernama NFC yang menggunakan induksi medan magnet untuk memungkinkan komunikasi antar perangkat elektronik dalam jarak yang dekat. Teknologi *Near Field Communication* (NFC) dapat membuat pekerjaan lebih mudah dan nyaman bagi penggunaannya di seluruh dunia dalam melakukan transaksi, seperti pertukaran konten digital, hanya menghubungkan perangkat elektronik ke perangkat elektronik yang memiliki teknologi NFC dengan sentuhan, pengguna juga bisa membeli tiket apapun hanya dengan mengaktifkan NFC pada ponsel.

NFC menyediakan media terbaik bagi identifikasi protokol yang memvalidasi secara aman dalam transfer data. Hal ini memungkinkan pengguna untuk mengakses konten digital dan terhubung dengan perangkat elektronik lainnya hanya dengan menyentuh atau membawa perangkat dalam jarak dekat. NFC beroperasi pada pita frekuensi dengan standar berlisensi 13.56MHz dengan jarak lebih dari sekitar 50 mm [2].

NFC Menawarkan kecepatan transfer data 106kbit/s, 212kbit/s dan 424kbit/s. Untuk dua perangkat yang berkomunikasi menggunakan NFC, satu perangkat harus memiliki alat pembaca NFC, yang pada dasarnya adalah sirkuit terintegrasi yang berisi data, terhubung ke antena, dapat dibaca dan ditulis oleh pembaca. Ada dua mode operasi yang dicakup oleh protokol “NFC”, yaitu aktif dan pasif.

2.3 *Fingerprint scanner*

Fingerprint Scanner adalah alat untuk memindai atau mengenali sidik jari seseorang. Seperti diketahui sidik jari tiap-tiap orang tidak ada yang sama di dunia. Ini dapat menjadi penanda alami dari tiap-tiap orang.



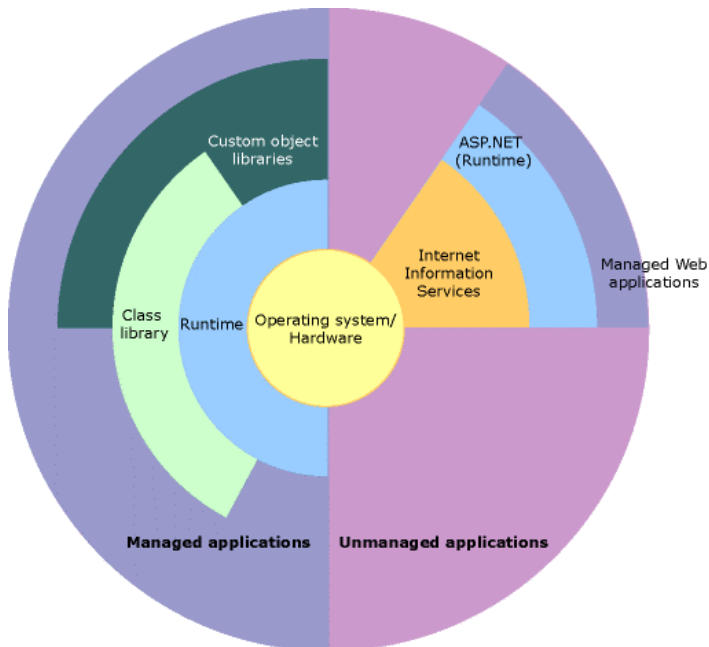
Gambar 2.3. *Fingerprint Scanner*

Seperti Gambar 2.3. pembacaan sidik jari dengan sensor *fingerprint* atau mesin sidik jari terdiri dari beberapa teknik. Pada umumnya sistem pembacaan ini hampir sama proses kerjanya [3]. Pertama kali sidik jari dibaca oleh mesin sidik jari atau sensor *fingerprint*. Hasil *scanning* lalu disimpan dalam format digital pada saat registrasi atau *enrollment* atau pendaftaran sidik jari. Setelah itu, rekaman sidik jari tersebut diproses dan dibuatkan daftar pola fitur sidik jari yang unik.

Pola fitur sidik jari yang unik tersebut kemudian disimpan dalam memori atau *database*. Pola sidik jari yang unik ini disebut dengan istilah *minutiae*. Pada saat identifikasi, pola *minutiae* tersebut kemudian dicocokkan atau dibandingkan dengan hasil scan sidik jari [4]

2.4 .NET

NET *framework* merupakan teknologi dari Microsoft yang mendukung pengembangan dan berjalannya aplikasi dan *web service* generasi terbaru. NET *framework* merupakan teknologi dari Microsoft yang mendukung pengembangan dan berjalannya aplikasi dan *web service* generasi terbaru.



Gambar 2.4. Arsitektur .NET

.NET *framework* didesain untuk memenuhi beberapa objektif antara lain untuk mendukung lingkungan pemrograman berbasis objek yang konsisten, membuat pengembang dapat dengan mudah mengembangkan berbagai aplikasi seperti aplikasi berbasis Windows ataupun aplikasi Web dan membuat standar agar kode yang berbasis .NET dapat terintegrasi dengan kode lain [5].

Pada arsitektur mengenai .NET yang terlihat pada Gambar 2.4. NET *framework* berisi *common language runtime* dan *.NET Framework Class Library*. *Common language runtime* merupakan agen yang mengatur kode pada saat program dieksekusi. Hubungan antara *common language runtime* dan *.NET Framework Class Library* terhadap aplikasi dan sistem secara umum. Kelas kakas bantu yang disediakan oleh .NET *framework* berisi kumpulan kakas bantu berorientasi objek yang bersifat *reusable*. Dengan kakas bantu ini kita dapat mengembangkan aplikasi mulai dari aplikasi berbentuk *command line*, aplikasi dengan *graphical user interface* (GUI) sampai aplikasi seperti ASP.NET dan *Web service* [6]

2.5 Microsoft Visual C#

Microsoft Visual C-Sharp atau yang lebih dikenal dengan Visual C# adalah sebuah bahasa pemrograman yang tidak diragukan lagi dalam proses pengembangan aplikasi berbasis .NET Framework, dimana C# bebas dari masalah kompatibilitas dilengkapi dengan berbagai fitur yang sebagian besar merupakan fitur baru, menarik, dan tentu saja menjanjikan.

C# merupakan sebuah bahasa pemrograman yang berorientasi objek yang dikembangkan oleh Microsoft sebagai bagian dari inisiatif kerangka .NET *Framework*. Bahasa pemrograman ini dibuat berbasiskan bahasa C++ yang telah dipengaruhi oleh aspek-aspek ataupun fitur bahasa yang terdapat pada bahasa-bahasa pemrograman lainnya seperti

Java, Delphi, Visual Basic, dan lain-lain) dengan beberapa penyederhanaan.

Seperti pemrograman Java, C# juga tidak memperbolehkan *multiple inheritance* atau penggunaan *pointer* (pada *safe/managed code*), tetapi C# menyediakan *garbage* memori *collection* pada saat *runtime* dan pada saat pengecekan akses memori [7].

Agar mampu mempromosikan penggunaan besar-besaran dari bahasa C#, Microsoft, yang di dukung oleh Intel Corporation dan Hewlett-Packard, mencoba mengajukan standarisasi terhadap bahasa C#. Akhirnya, pada bulan Desember 2001, standar pertama pun diterima oleh *European Computer manufacturers Association* atau *Ecma International* (ECMA), dengan nomor standar ECMA-334. Pada Desember 2002, standar kedua pun diadopsi oleh ECMA, dan tiga bulan kemudian diterima oleh *International Organization for Standardization* (ISO), dengan nomor standar ISO/IEC 23270:2006 [8].

2.6 MySQL

Sebuah perangkat lunak system manajemen basis data SQL (DBMS) yang dapat *multithread*, dan *multi-user* adalah *MY SQL*. MySQL merupakan implementasi dari *system* manajemen basisdata relasional (RDBMS).

Pada saat ini MySQL merupakan *database server* yang sangat terkenal di dunia, semua itu tak lain karena bahasa dasar yang digunakan untuk mengakses database yaitu SQL. Penggunaan MYSQL sangat mudah dibandingkan dengan yang lain, misalnya dBase atau Clipper karena mereka masih menggunakan perintah-perintah pemrograman murni.

Konektivitas dengan pemrograman selain PHP, MySQL menyediakan bermacam-macam *library* untuk bahasa pemrograman lainnya, sehingga MySQL bisa dipakai dibanyak bahasa pemrograman.

[Halaman ini sengaja dikosongkan]

BAB III

PERANCANGAN DAN DESAIN SISTEM

Pada bab ini akan diuraikan mengenai perancangan sistem perangkat lunak untuk mencapai tujuan tugas akhir.

3.1 Deskripsi Umum

Sistem yang akan dibangun ini adalah perangkat lunak untuk sistem parkir yang menggunakan RFID dan *Fingerprint Scanner* untuk proses otentikasi. Berikut adalah fitur-fitur yang perlu diberikan oleh sistem ini :

1. Sistem dapat mengotentikasi *driver* dan kendaraan yang keluar masuk tempat parkir.
2. Pengguna dapat menetapkan harga parkir.
3. Sistem dapat memberikan laporan yang berisi data-data yang berkaitan dengan sistem parkir.

3.2 Arsitektur Umum Sistem

Untuk memenuhi fitur-fitur yang telah dijelaskan pada Deskripsi Umum Sistem, maka akan dibangun perangkat lunak dengan arsitektur umum seperti pada Gambar 3.1. Gambar tersebut menunjukkan diagram alur dari sistem secara umum.

Sebelum *driver* menuju tempat parkir, apabila belum terdaftar, terlebih dahulu dapat mendaftarkan diri dan kendaraannya lewat melalui sebuah web. Kemudian Sidik jari pengguna dapat direkam ketika telah sampai ditempat parkir dan akan diberi sebuah kartu yang berisi data kendaraan yang telah didaftarkannya.

Untuk masuk ke dalam lahan parkir, *driver* perlu mencocokkan sidik jari dan kartu RFID yang dipunyainya. Setelah kedua data (Sidik Jari dan Kartu RFID) tersebut cocok, maka kartu RFID tersebut akan menyimpan informasi parkir masuk yang dilakukan oleh *driver*.

Untuk proses keluar dari lahan parkir, sama seperti ketika masuk parkir, *driver* perlu mencocokkan sidik jari dan kartu RFID yang dipunyainya. Setelah cocok, pengguna dapat mencetak laporan parkir sebagai bukti pembayaran *driver*.

Admin dapat menggunakan sistem untuk menetapkan tarif parkir yang berlaku. Selain itu admin juga dapat melihat informasi kendaraan dan *driver* pada hari tertentu. Kemudian admin juga dapat mencetak laporan bulanan atau harian.



Gambar 3.1. Arsitektur Sistem

3.3 Perancangan Sistem

Kasus Penggunaan yang digunakan pada sistem ini dapat dilihat pada Gambar 3.2. Dari Gambar tersebut terdapat tujuh kasus penggunaan yang ada dalam sistem ini.

Penjelasan dari masing-masing kasus penggunaan dapat dilihat pada Tabel 3.1. Tiap-tiap kasus penggunaan dijelaskan dengan menggunakan tabel spesifikasi kasus penggunaan pada subbab berikutnya.

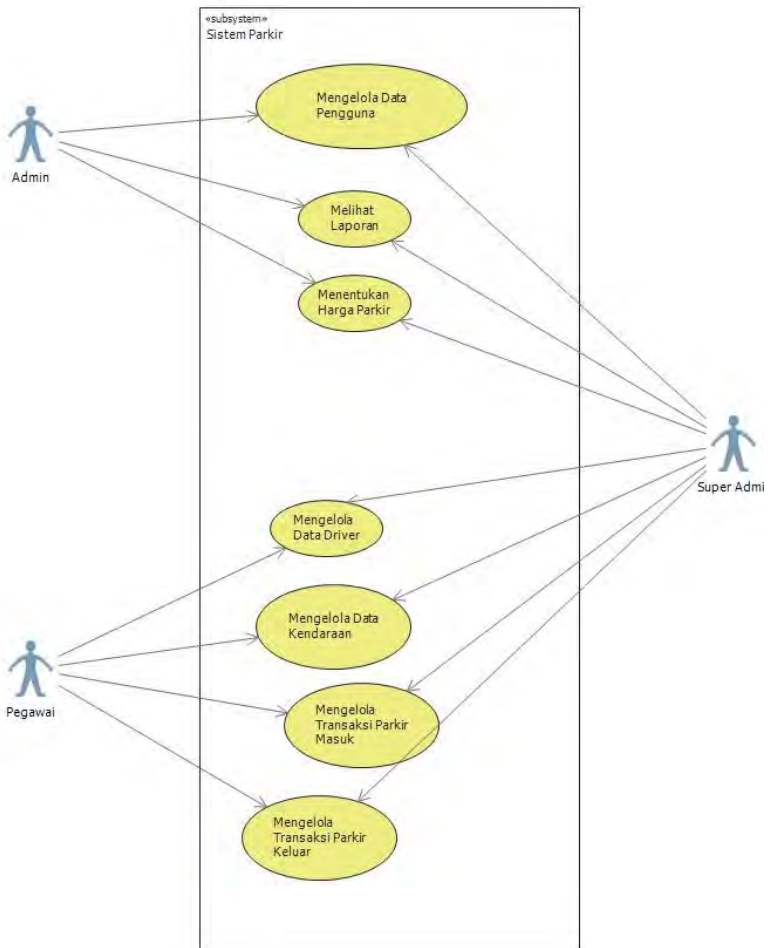
Tabel 3.1. Kasus Penggunaan

No.	Kode	Nama Kasus Penggunaan	Deskripsi
1.	UC-01	Mengelola Data Pengguna	Pengguna dapat membuat pengguna sistem yang baru dan memperbarui data pengguna yang lama.
2.	UC-02	Melihat laporan	Pengguna dapat melihat hasil laporan yang dihasilkan oleh sistem.
3.	UC-03	Menentukan harga parkir	Pengguna dapat mengubah harga parkir.
4.	UC-04	Mengelola Data Driver	Pengguna dapat melakukan penambahan data <i>driver</i> yang baru, menghapus data <i>driver</i> yang ada, memperbarui data <i>driver</i> dan menyimpan rekaman sidik jari <i>driver</i> .
5.	UC-05	Mengelola Data Kendaraan	Pengguna dapat melakukan penambahan data kendaraan yang baru, menghapus

			data kendaraan yang ada, memperbarui data kendaraan dan menyimpan data kendaraan ke dalam kartu.
6.	UC-06	Mengelola Transaksi Parkir Masuk	Pengguna dapat melakukan penambahan transaksi untuk parkir masuk.
7.	UC-07	Mengelola Transaksi Parkir Keluar	Pengguna dapat melakukan penambahan transaksi untuk parkir keluar.

Dari kasus penggunaan yang terdapat pada Tabel 3.1 dapat disimpulkan bahwa pada sistem ini memiliki 7 buah kasus dan 3 aktor yang dapat dijadikan sebagai pengguna aplikasi. Pada Gambar 3.2 merupakan diagram kasus yang terjadi pada sistem, dengan memiliki 3 jenis aktor yang menjadi pengguna aplikasi.

Aktor yang terdapat dalam sistem adalah actor super admin, admin dan pegawai. Masing – masing aktor memiliki fitur – fitur yang berbeda sebagai pengguna. Super admin adalah actor yang mengimplementasikan sebagai pemilik perusahaan yang menyediakan parkir, Admin adalah actor yang memegang kendali laporan transaksi sedangkan pegawai menjadi aktor yang bertugas menjalankan transaksi parkir.



Gambar 3.2. Diagram Kasus

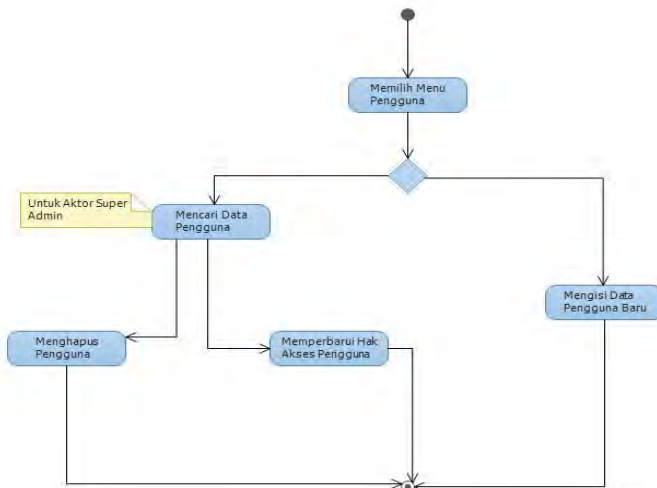
3.3.1 Kasus Penggunaan Mengelola Data Pengguna

Terdapat dua aktor yang dapat mengakses kasus penggunaan ini, yaitu admin dan Super admin. Dalam kasus penggunaan ini, Admin dapat menambahkan pengguna dengan hak akses pegawai. Untuk Super admin dapat menambahkan pengguna dengan hak akses admin ataupun pegawai. Selain itu Super admin dapat menghapus data pengguna yang ada. Rincian kasus penggunaan ini dapat dilihat pada Tabel 3.2, dan diagram aktivitasnya dapat dilihat pada Gambar 3.3

Tabel 3.2. Pengelola Data Pengguna

Nama	Mengelola Data Pengguna.
Nomor	UC-01
Deskripsi	Pengguna dapat membuat pengguna sistem yang baru dan memperbarui data pengguna yang lama.
Tipe	Fungsional
Aktor	Admin dan Super admin.
Kondisi Awal	Pengguna sudah masuk ke dalam sistem.
Alur Normal	<ol style="list-style-type: none"> 1. Sistem menampilkan pilihan menu. 2. Pengguna memilih Menu Pengguna. 3. Sistem menampilkan Menu Pengguna. 4. Pengguna memilih menambah pengguna baru. <ol style="list-style-type: none"> A.1 Pengguna memilih mencari data Pengguna. 5. Sistem menampilkan isian data pengguna baru. 6. Pengguna mengisi data pengguna baru. 7. Sistem menyimpan data pengguna. 8. Kasus penggunaan berakhir.
Alur Alternatif	A.1 Pengguna memilih mencari data Pengguna

	<ol style="list-style-type: none"> 1. Sistem menampilkan daftar pengguna. 2. Pengguna memilih pengguna yang akan diganti hak aksesnya. <ol style="list-style-type: none"> B.1. Pengguna memilih menghapus data pengguna. 3. Sistem menampilkan data pengguna. 4. Pengguna mengganti hak akses. 5. Berlanjut ke alur normal langkah 7. <p>B.1 Pengguna memilih menghapus data pengguna.</p> <ol style="list-style-type: none"> 1. Sistem menghapus data pengguna. 2. Berlanjut ke alur normal langkah 7.
--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------



Gambar 3.3. Diagram Aktivitas Pengolahan Data Pengguna

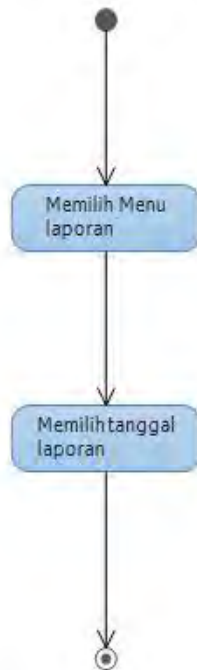
3.3.2 Kasus Penggunaan Melihat Laporan

Pada kasus penggunaan ini, pengguna dapat melihat hasil laporan yang dihasilkan oleh sistem. Isi laporan tersebut mencerminkan data kendaraan yang masuk seperti nomor kendaraan, nama *driver* yang masuk dan keluar, waktu masuk dan keluar, pegawai yang menangani dan jenis kendaraan.

Pengguna yang terlibat dalam kasus penggunaan ini adalah Admin dan Super admin. Rincian kasus penggunaan dapat dilihat pada Tabel 3.3. dan diagram aktifitasnya dapat dilihat pada Gambar 3.4.

Tabel 3.3. Melihat Laporan

Nama	Melihat Laporan.
Nomor	UC-02
Deskripsi	Pengguna dapat melihat laporan yang dihasilkan oleh sistem.
Tipe	Fungsional
Aktor	Admin dan Super admin.
Kondisi Awal	Pengguna sudah masuk ke dalam sistem.
Alur Normal	<ol style="list-style-type: none"> 1. Sistem menampilkan pilihan menu. 2. Pengguna memilih Menu Laporan. 3. Sistem menampilkan Menu Laporan. 4. Pengguna mengisi tanggal awal dan akhir untuk transaksi yang akan dibuatkan laporannya. 5. Sistem menampilkan hasil laporan transaksi. 6. Kasus penggunaan berakhir.



Gambar 3.4. Diagram Aktivitas Memilih Laporan

3.3.3 Kasus Penggunaan Menentukan Harga Parkir

Pada kasus penggunaan ini, pengguna dapat menetapkan harga parkir yang berlaku. Pengguna dapat menambah daftar harga parkir, mengubah harga parkir yang berlaku berdasarkan daftar harga parkir, dan menghapus data harga parkir yang tidak dipakai.

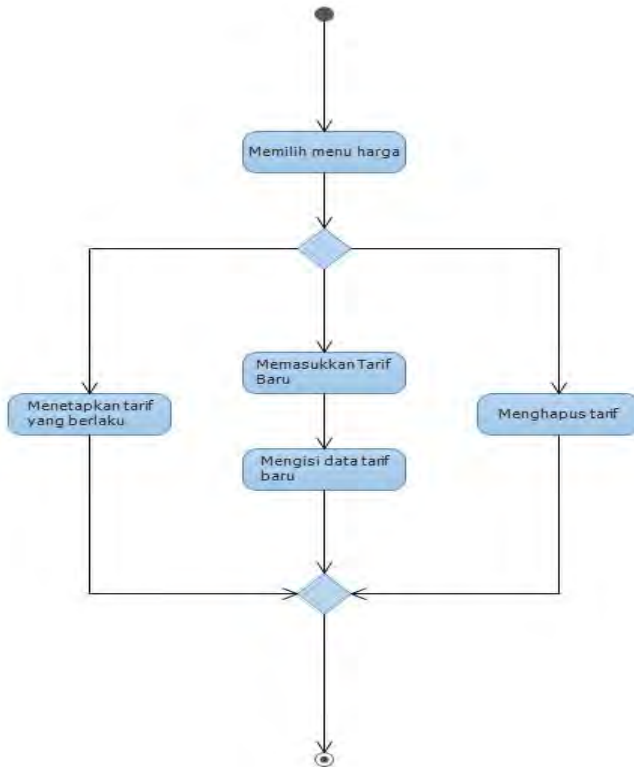
Harga parkir yang perlu ditetapkan antara lain, harga parkir selama satu jam pertama, harga parkir tiap jam setelah satu jam pertama, dan harga parkir maksimal. Harga parkir sepeda motor dapat berbeda dengan harga parkir mobil.

Pengguna yang terlibat dalam kasus penggunaan ini adalah Admin dan Super admin. Rincian kasus penggunaan

dapat dilihat pada Tabel 3.4. dan diagram aktifitas dapat dilihat pada Gambar 3.5.

Tabel 3.4. Menentukan Harga Parkir

Nama	Menentukan harga parkir.
Nomor	UC-04
Deskripsi	Kasus penggunaan ini digunakan untuk menetapkan tarif parkir yang berlaku.
Tipe	Fungsional
Aktor	Admin dan Super admin.
Kondisi Awal	Pengguna sudah masuk ke dalam sistem.
Alur Normal	<ol style="list-style-type: none"> 1. Sistem menampilkan pilihan menu. 2. Pengguna memilih Menu harga. 3. Sistem menampilkan daftar harga. 4. Pengguna menetapkan harga parkir yang berlaku. <ul style="list-style-type: none"> A.1 Pengguna memilih menambah daftar harga parkir. A.2 Pengguna memilih menghapus harga parkir. 5. Sistem menyimpan data harga. 6. Kasus penggunaan berakhir.
Alur Alternatif	<p>A.1 Pengguna memilih menambah daftar harga parkir.</p> <ol style="list-style-type: none"> 1. Sistem menampilkan isian data harga parkir. 2. Pengguna mengisi data harga parkir. 3. Berlanjut ke alur normal langkah 5. <p>A.2 Pengguna memilih menghapus harga parkir.</p> <ol style="list-style-type: none"> 1. Sistem menyimpan daftar harga 2. Berlanjut ke alur normal langkah 6.



Gambar 3.5. Diagram Aktivitas Menentukan Harga Parkir

3.3.4 Kasus Penggunaan Mengelola Data *Driver*

Pada kasus penggunaan ini, pengguna dapat melihat data *driver* yang telah terdaftar sebagai pelanggan parkir. Tidak hanya itu pengguna juga dapat menambahkan, menyunting,

dan menghapus data *driver* yang diinginkan. Selain itu pengguna juga dapat merekam sidik jari dari *driver* tersebut.

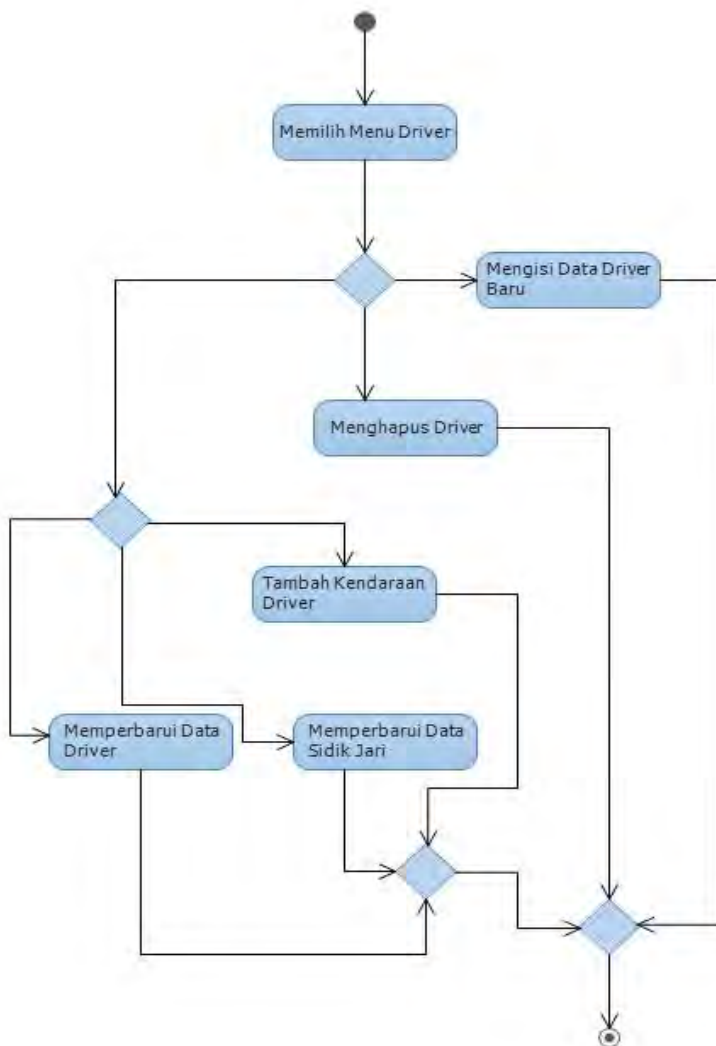
Data *driver* yang disimpan antara lain nomor Identitas penduduk(KTP), dan data-data pribadi seperti nama, alamat, tanggal lahir, pekerjaan. Selain itu data *driver* yang disimpan juga berisi rekaman sidik jari jempol kanan dari *driver*.

Pengguna yang terlibat dalam kasus penggunaan ini adalah pegawai dan super admin. Rincian kasus penggunaan ini dapat dilihat pada Tabel 3. 5. dan diagram aktivitasnya dapat dilihat pada Gambar 3.6.

Tabel 3. 5. Mengelola Data *Driver*

Nama	Mengelola Data <i>Driver</i> .
Nomor	UC-04
Deskripsi	Kasus penggunaan ini digunakan untuk melakukan manajemen pendaftaran <i>driver</i> pelanggan parkir
Tipe	Fungsional
Aktor	Pegawai dan Super admin.
Kondisi Awal	Pengguna sudah masuk ke dalam sistem.
Alur Normal	<ol style="list-style-type: none"> 7. Sistem menampilkan pilihan menu. 8. Pengguna memilih menu <i>driver</i>. 9. Sistem menampilkan menu <i>driver</i>. 10. Pengguna memilih menambah <i>driver</i> baru. <ol style="list-style-type: none"> A.1 Pengguna memilih menghapus data <i>driver</i>. A.2 Pengguna memilih memperbarui data <i>driver</i>. A.3 Pengguna memilih menambah kendaraan <i>driver</i>.

	<p>A.4 Pengguna memilih memperbarui rekaman sidik jari <i>driver</i>.</p> <ol style="list-style-type: none"> 11. Sistem menampilkan isian data pengguna baru. 12. Pengguna mengisi data <i>driver</i> baru. 13. Sistem menyimpan data <i>driver</i>. 14. Kasus penggunaan berakhir.
<p>Alur Alternatif</p>	<p>A.1 Pengguna memilih menghapus data <i>driver</i>.</p> <ol style="list-style-type: none"> 4. Sistem menghapus data <i>driver</i>. 5. Berlanjut ke alur normal langkah 8. <p>A.2 Pengguna memilih memperbarui data <i>driver</i>.</p> <ol style="list-style-type: none"> 3. Sistem menampilkan isian data <i>driver</i>. 4. Pengguna mengisi data <i>driver</i>. 5. Berlanjut ke alur normal langkah 7. <p>A.3 Pengguna memilih menambah kendaraan <i>driver</i>.</p> <ol style="list-style-type: none"> 1. Sistem menampilkan daftar kendaraan. 2. Pengguna memilih kendaraan. 3. Berlanjut ke alur normal langkah 7. <p>A.4 Pengguna memilih memperbarui rekaman sidik jari <i>driver</i>.</p> <ol style="list-style-type: none"> 1. Sistem menampilkan perintah pengisian sidik jari. 2. Pengguna meletakkan jempol kanan <i>driver</i> ke pembaca sidik jari. 3. Sistem menyimpan data sidik jari. 4. Berlanjut ke alur normal langkah 8.



Gambar 3.6. Diagram Aktivitas Mengelola Data *Driver*

3.3.5 Kasus Penggunaan Pengelolaan Data Kendaraan

Pada kasus penggunaan ini, pengguna dapat melihat data kendaraan yang telah terdaftar, menambah data kendaraan baru, memperbarui data kendaraan yang ada, menghapus data kendaraan yang tidak diinginkan, dan merekam data kendaraan ke dalam kartu.

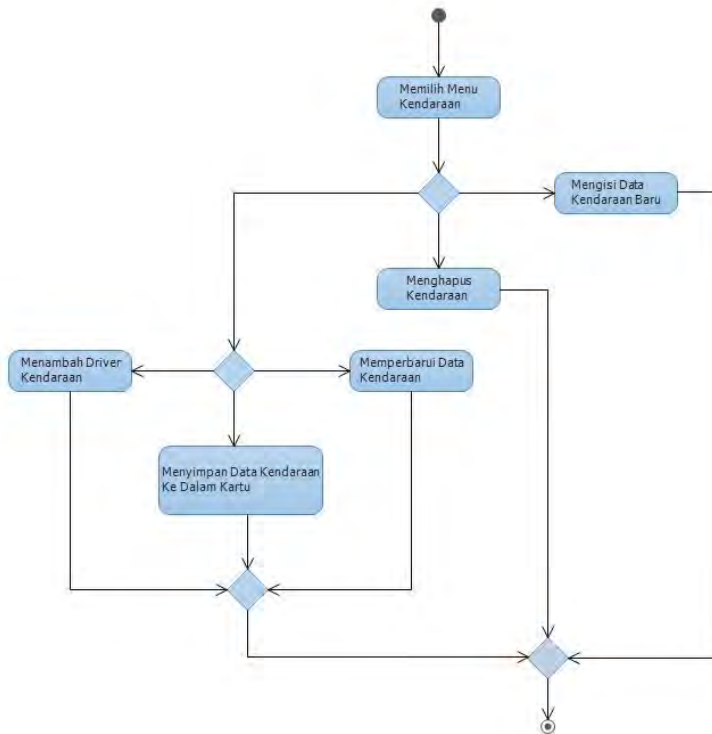
Data kendaraan yang disimpan antara lain nomor polisi dan jenis kendaraan. Jenis kendaraan yang disimpan adalah mobil atau sepeda motor. Kemudian data yang disimpan ke dalam kartu adalah nomor polisi dan id kendaraan.

Pengguna yang terlibat dalam kasus penggunaan ini adalah pegawai dan super admin. Rincian kasus penggunaan ini dapat dilihat pada Tabel 3.6 dan diagram aktivitasnya dapat dilihat pada Gambar 3.7

Tabel 3.6. Mengelola Data Kendaraan

Nama	Mengelola Data Kendaraan.
Nomor	UC-05
Deskripsi	Kasus penggunaan ini digunakan untuk melakukan manajemen pendaftaran kendaraan pelanggan parkir
Tipe	Fungsional
Aktor	Pegawai dan Super Admin.
Kondisi Awal	Pengguna sudah masuk ke dalam sistem.
Alur Normal	<ol style="list-style-type: none"> 1. Sistem menampilkan pilihan menu. 2. Pengguna memilih Menu kendaraan. 3. Sistem menampilkan Menu kendaraan. 4. Pengguna memilih menambah kendaraan baru. <ol style="list-style-type: none"> A.1 Pengguna memilih menghapus data kendaraan.

	<p>A.2 Pengguna memilih memperbarui data kendaraan.</p> <p>A.3 Pengguna memilih menambah <i>driver</i> kendaraan.</p> <p>A.4 Pengguna memilih menyimpan nomor polisi ke dalam kartu.</p> <p>5. Sistem menampilkan isian data kendaraan baru.</p> <p>6. Pengguna mengisi data kendaraan baru.</p> <p>7. Sistem menyimpan data kendaraan.</p> <p>8. Kasus penggunaan berakhir.</p>
<p>Alur Alternatif</p>	<p>A.1 Pengguna memilih menghapus data kendaraan.</p> <ol style="list-style-type: none"> 1. Sistem menghapus data kendaraan. 2. Berlanjut ke alur normal langkah 8. <p>A.2 Pengguna memilih memperbarui data kendaraan.</p> <ol style="list-style-type: none"> 1. Sistem menampilkan isian data kendaraan. 2. Pengguna mengisi data kendaraan. 3. Berlanjut ke alur normal langkah 7. <p>A.3 Pengguna memilih menambah <i>driver</i> kendaraan.</p> <ol style="list-style-type: none"> 1. Sistem menampilkan daftar <i>driver</i>. 2. Pengguna memilih <i>driver</i>. 3. Berlanjut ke alur normal langkah 7. <p>A.4 Pengguna memilih menyimpan nomor polisi ke dalam kartu.</p> <ol style="list-style-type: none"> 1. Pengguna meletakkan kartu ke pembaca kartu. 2. Sistem menyimpan nomor polisi kendaraan ke dalam kartu. 3. Berlanjut ke alur normal langkah 8.



Gambar 3.7. Diagram Aktivitas Mengelola Data Kendaraan

3.3.6 Kasus Penggunaan Mengelola Transaksi Masuk

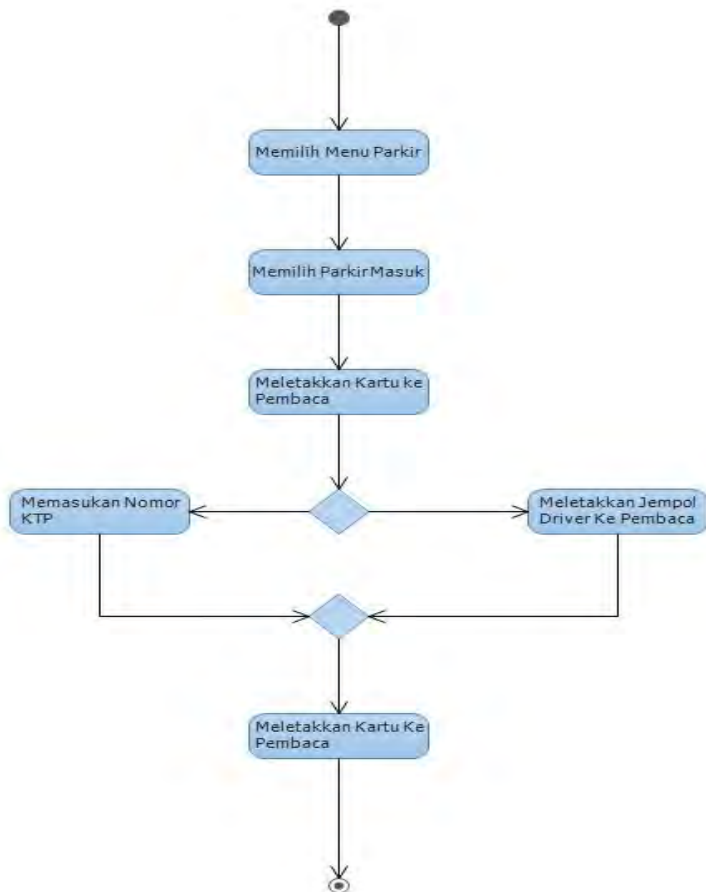
Pada kasus penggunaan ini, pengguna dapat memasukkan data-data kendaraan dan *driver*, yang akan masuk ke areal parkir, ke dalam sistem. Selain data kendaraan dan data *driver*, tanggal dan jam masuknya kendaraan ke area parkir juga disimpan di sistem.

Pengguna yang terlibat dalam kasus penggunaan ini adalah pegawai dan super admin. Rincian kasus penggunaan ini

dapat dilihat pada Tabel 3.7. dan diagram aktivitasnya dapat dilihat pada Gambar 3.8.

Tabel 3.7. Mengelola Transaksi Masuk

Nama	Mengelola Transaksi Masuk
Nomor	UC-06
Deskripsi	Kasus penggunaan ini digunakan untuk menyimpan data kendaraan dan <i>driver</i> yang masuk ke area parkir.
Tipe	Fungsional
Aktor	Pegawai dan Super Admin.
Kondisi Awal	Pengguna sudah masuk ke dalam sistem.
Alur Normal	<ol style="list-style-type: none"> 1. Sistem menampilkan Menu parkir. 2. Pengguna memilih parkir masuk. 3. Sistem menampilkan menu parkir masuk. 4. Pengguna meletakkan kartu ke pembaca kartu. 5. Sistem menampilkan informasi kendaraan. 6. Pengguna meletakkan jempol kanan driver ke pembaca sidik jari. A.1 Pengguna memasukkan nomor KTP <i>driver</i>. 7. Sistem menampilkan informasi <i>driver</i>. 8. Pengguna meletakkan kartu ke pembaca kartu. 9. Sistem menyimpan data parkir ke kartu. 10. Kasus penggunaan berakhir.
Alur Alternatif	<p>A.1 Pengguna memasukkan nomor KTP driver.</p> <ol style="list-style-type: none"> 1. Kembali ke alur normal langkah 9.



Gambar 3.8. Diagram Aktivitas Mengelola Transaksi Masuk

3.3.7 Kasus Penggunaan Mengelola Transaksi Keluar

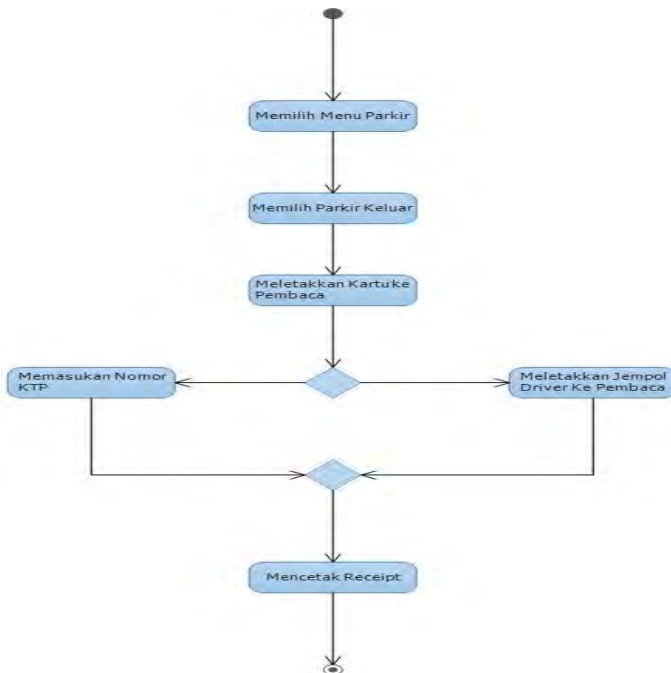
Pada kasus penggunaan ini, pengguna memasukkan data kendaraan dan *driver* yang keluar dari parkir. Setelah memasukkan data kendaraan dan *driver* yang keluar dari parkir, pengguna dapat mencetak *tagihan* yang berisi informasi kendaraan dan *driver*, waktu masuk parkir dan keluar parkir, dan total biaya yang harus dibayar oleh *driver*.

Pengguna yang terlibat dalam kasus penggunaan ini adalah pegawai dan super admin. Detail kasus penggunaan dapat dilihat pada Tabel 3.8 dan diagram aktifitasnya dapat dilihat pada Gambar 3.9.

Tabel 3.8. Mengelola Transaksi Keluar

Nama	Mengelola Transaksi Keluar
Nomor	UC-07
Deskripsi	Kasus penggunaan ini digunakan untuk menyimpan data kendaraan dan <i>driver</i> yang keluar ke area parkir dan mencetak <i>receipt</i> .
Tipe	Fungsional
Aktor	Pegawai dan Super Admin.
Kondisi Awal	Pengguna sudah masuk ke dalam sistem.
Alur Normal	<ol style="list-style-type: none"> 11. Sistem menampilkan pilihan menu. 12. Pengguna memilih Menu parkir. 13. Sistem menampilkan Menu parkir. 14. Pengguna memilih parkir keluar. 15. Sistem menampilkan menu parkir keluar. 16. Pengguna meletakkan kartu ke pembaca kartu. 17. Sistem menampilkan informasi kendaraan.

	<p>18. Pengguna meletakkan jempol kanan <i>driver</i> ke pembaca sidik jari. A.1 Pengguna memasukkan nomor KTP <i>driver</i>.</p> <p>19. Sistem menampilkan informasi <i>driver</i>. 20. Pengguna memilih mencetak <i>receipt</i>. 21. Sistem mencetak <i>receipt</i>. 22. Kasus penggunaan berakhir.</p>
Alur Alternatif	<p>A.1 Pengguna memasukkan nomor KTP <i>driver</i>. 2. Kembali ke alur normal langkah 9.</p>



Gambar 3.9. Diagram Aktivitas Mengelola Transaksi Keluar

3.4 Rancangan Antarmuka Aplikasi

Rancangan antarmuka aplikasi mengulas tentang desain aplikasi beserta penjelasan masing-masing antar muka di setiap menu atau fungsi yang terdapat pada aplikasi *Smart Gate* tersebut.

3.4.1 Rancangan Antarmuka Pembuatan Pengguna

Gambar 3.10. menunjukkan rancangan antarmuka pembuatan pengguna yang dapat menggunakan Sistem ini. Dalam gambar tersebut pengguna perlu memasukkan nomor pengguna baru, nama pengguna, *password* pengguna dan hak akses pengguna tersebut. Tombol “daftar” untuk memasukkan hasil masukkan kedalam *database* dengan catatan data yang diisikan sudah benar.



The image shows a user registration form with the following fields and a button:

- Nomor Pegawai:
- Nama Pegawai:
- Password:
- Hak Akses:
- Daftar:

Gambar 3.10. Antarmuka Pembuatan Pengguna

3.4.2 Rancangan Antarmuka Melihat Laporan

Gambar 3.11. menunjukkan rancangan antarmuka pembuatan laporan yang dihasilkan oleh sistem. Pengguna perlu mengisi terlebih dahulu rentang waktu yang ingin dibuatkan laporan. Setelah itu daftar laporan akan ditampilkan oleh sistem sesuai rentang waktu yang sudah ditentukan sebelumnya.



The image shows a user interface for generating a report. It features two rows of date pickers. The first row is labeled 'Tanggal Awal' (Start Date) and the second row is labeled 'Tanggal Akhir' (End Date). Each label is followed by a 'Datetime Picker' input field with a downward arrow icon. Below these two rows is a single button labeled 'Buat Laporan' (Generate Report).

Gambar 3.11. Antarmuka Melihat Laporan

3.4.3 Rancangan Antarmuka Menentukan Harga

Gambar 3.12. menunjukan rancangan antarmuka yang dipakai untuk menetapkan harga parkir. Seperti yang dapat dilihat pada gambar terdapat tombol “Buat Baru” dengan tombol tersebut dapat menentukan harga baru yg belum dibuat sebelumnya, Di dalam kolom *list* terdapat hasil harga yang sudah dibuat berupa jenis kendaraan, tarif harga per jam, dan maksimal harga dalam 1 hari parkir. Penghapusan daftar bisa dilakukan dengan memilih baris yang akan di hapus kemudian ada pilihan fitur untuk penghapusan atau pengeditan di sebelah kanan daftar.

Jenis Kendaraan v

Buat Baru

Harga Pokok	Harga/Jam	Harga MAX	
Content 1	Content 2	Content 5	<input type="button" value="Set"/> <input type="button" value="Hapus"/>

Gambar 3.12. Antarmuka Menentukan Harga Parkir

3.4.4 Rancangan Antarmuka Mengelola Data *Driver*

Untuk kasus penggunaan mengelola data *driver* terdapat beberapa rancangan antarmuka yang dibuat. Gambar 3.13 menunjukkan isian data *driver* yang akan dipakai, ketika mengisi data *driver* baru atau mengubah data *driver*. Seperti yang terlihat pada Gambar 3.13 data yang perlu masukkan antara lain no KTP, nama, Tgl Lahir dan Tempat Lahir, Pekerjaan dan Alamat. Tombol “Tambah” digunakan untuk memasukkan data kedalam *database* dengan catatan data yang dimasukkan sudah benar.

Data yang dimasukkan kedalam sistem diambil dari beberapa macam *text box* yang sudah disediakan beserta *date picker* yang digunakan untuk menentukan tanggal. Semua data di rekam dengan bentuk *string* dan dimasukkan kedalam *database* setelah semua data dimasukkan. Setelah data masuk maka hasil data yang sudah dimasukkan akan langsung dikeluarkan pada detail *driver*.



The image shows a web form for adding driver data. It consists of several input fields and a button. The fields are labeled as follows:

- No KTP: A text input field.
- Nama: A text input field.
- Tgl Lahir: A dropdown menu labeled "Datetime Picker" with a downward arrow.
- Tempat Lahir: A text input field.
- Pekerjaan: A text input field.
- Alamat: A text input field.

At the bottom left of the form is a button labeled "Tambah".

Gambar 3.13. Antarmuka Penambahan Data *Driver*

Antarmuka untuk melakukan penghapusan data *driver* dapat dilihat pada Gambar 3.14. pengguna terlebih harus mencari *driver* mana yang akan dihapus berdasarkan *filter* yang sudah disediakan berupa jenis kendaraan atau bisa menggunakan kata kunci sesuai huruf yang dimasukkan yang hasilnya ditunjukkan pada tabel. Kemudian cukup pilih hapus baris tersebut. Sedangkan untuk melihat detail dari data *driver* cukup pilih detail di baris itu.

Pada pilihan tombol detail antarmuka akan dibawa ke antarmuka detail *driver* dimana pada antarmuka ini data *driver* ditampilkan dan dapat untuk diperbarui dan sekaligus memasukkan sidik jari.

Berdasarkan ▼

Kata Kunci

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

Gambar 3.14. Antarmuka Penghapusan Data *Driver*

Antarmuka untuk memperbarui Sidik Jari dan menambah daftar kendaraan, dapat dilihat pada Gambar 3.15. Di antarmuka inilah data diri *driver* dan data kendaraan yang bisa dipakai ditampilkan. Untuk memperbarui data diri *driver* cukup menekan tombol yang akan memunculkan antarmuka seperti pada Gambar 3.15. Pada gambar tersebut terdapat tombol ubah, perbarui sidik jari, dan tambah dimana masing masing tombol dapat digunakan untuk mengganti identitas *driver*, memperbarui sidik jari yang belum atau sudah ada, dan untuk menambah jenis kendaraan yang akan digunakan *driver*.

Tabel yang digunakan untuk menampilkan data kendaraan menggunakan data *grid* yang bisa diberikan fungsi tambah dengan menggunakan tombol tambah, kemudian antarmuka akan masuk pada pemiihan data kendaraan.

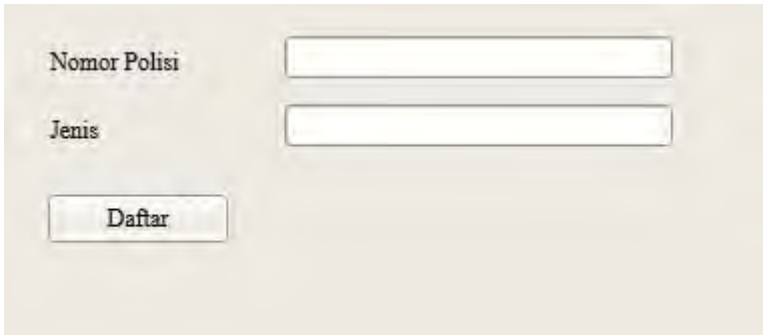


Gambar 3.15. Antarmuka Mememprbarui Sidik Jari *Driver*

3.4.5 Rancangan Antarmuka Mengelola Data Kendaraan

Antarmuka untuk kasus penggunaan mengelola data kendaraan dibagi menjadi beberapa bagian, yaitu : antarmuka penambahan data kendaraan, antarmuka pencarian data kendaraan untuk proses menghapus, dan antarmuka detail data kendaraan untuk proses penambahan data *driver* dan merekam data kendaraan ke kartu.

Antarmuka isian data kendaraan dapat dilihat pada Gambar 3.16. Data yang perlu diisi adalah nomor polisi dan jenis kendaraan. Antarmuka ini akan dipakai untuk menambah data kendaraan yang baru dan memperbarui data kendaraan yang ada.



The image shows a web form with a light beige background. It contains two text input fields. The first field is labeled 'Nomor Polisi' and the second is labeled 'Jenis'. Below these fields is a button labeled 'Daftar'.

Gambar 3.16. Antarmuka Penambahan Data Kendaraan

Pada Gambar 3.16 nomor polisi yang dimasukkan berupa masukan *string* dengan tanpa karakter pemisah atau *space* karena dalam penyimpanan nomor kendaraan yang sudah terenkripsi dalam kartu terlalu panjang dan memori tidak cukup menampung.

Antarmuka pencarian data kendaraan sama persis seperti Gambar 3.17. Pengguna terlebih dahulu memilih *filter* dan kata kunci yang akan digunakan untuk mencari kendaraan dan hasil pencarian dapat dilihat pada tabel di antarmuka tersebut. Ketika ingin menghapus data kendaraan cukup memilih hapus pada baris kendaraan tersebut. Ketika ingin melihat detail data kendaraan dapat memilih detail pada baris kendaraan itu.

Antarmuka detail data kendaraan dapat dilihat pada Gambar 3.17. Di antarmuka ini, terdapat tombol ubah, simpan kartu, dan tambah sehingga pengguna selain dapat melihat detail data kendaraan juga dapat mengubah data kendaraan, menyimpan data kendaraan ke kartu, dan menambah daftar driver yang dapat menggunakan kendaraan tersebut.

Tabel *driver* disebelah kanan sama dengan tabel kendaraan pada detail *driver* sebelumnya. Tabel ini mengimplementasikan kepemilikan *driver* atas kendaraan yang sudah tersimpan pada *database* di table kepemilikan.

The screenshot displays a software interface with two main sections: 'Data Kendaraan' and 'Daftar Driver'.

Data Kendaraan: This section contains a form with four 'Label' text boxes arranged in a 2x2 grid. Below the form are two buttons: 'Ubah' and 'Simpan Data Ke Kartu'.

Daftar Driver: This section features a table with two columns and three rows of data. To the right of the table is a 'Tambah' button.

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

Gambar 3.17. Antarmuka Detail Data Kendaraan

3.4.6 Rancangan Antarmukan Transaksi Parkir Masuk dan Keluar

Transaksi parkir baik untuk masuk parkir ataupun keluar parkir, menggunakan antarmuka yang sama seperti yang dapat dilihat pada Gambar 3.18. Pada antarmuka ini terdapat petunjuk langkah-langkah otentikasi parkir yang diperlukan.

Pada saat awal *driver* masuk atau keluar, *driver* memberikan kartu untuk dibaca dipembaca kartu. Kemudian sistem akan menampilkan data kendaraan yang tersimpan dikartu tersebut. Setelah itu *driver* diharuskan meletakkan jempol kanan di pembaca sidik jari untuk dicocokkan oleh sistem. Apabila terjadi kerusakan pada alat pembaca sidik jari, pengguna dapat menekan tombol untuk menampilkan isian no KTP untuk proses otentikasi.

Ketika dua proses tersebut selesai, untuk parkir masuk, *driver* perlu meletakkan kartu dipembaca kartu untuk proses memasukkan data parkir. Untuk proses keluar parkir, sistem akan mengeluarkan perhitungan biaya parkir yang perlu

dibayar oleh *driver* sesuai lama parkir yang dihitung dari waktu system ketika masuk parkir dan ketika keluar parkir.



Gambar 3.18. Antarmuka Transaksi Parkir Masuk dan Keluar

BAB IV IMPLEMENTASI APLIKASI

Pada bab ini akan dibahas tentang implementasi dari perancangan perangkat keras dan lunak yang telah direncanakan dan dijelaskan pada Bab III. Dalam membangun sistem ini, digunakan beberapa perangkat pendukung, baik perangkat keras maupun perangkat lunak. Lingkungan pembangunan dijelaskan sebagai berikut.

4.1 Lingkungan Implementasi

Aplikasi dibuat pada lingkungan implementasi perangkat keras dan perangkat lunak.

4.1.1 Lingkungan Perangkat Keras

Spesifikasi perangkat keras yang digunakan pada lingkungan pengembangan aplikasi adalah sebagai berikut:

- Laptop Toshiba L745,
 - Windows 8.1 Pro,
 - Processor Intel(R) Core(TM) i5-2410M CPU @ 2.30GHz,
 - RAM DDR3 4,00 GB, dan
 - Sistem Operasi: 64-bit.

- Pembaca Sidik Jari
 - Digital Persona U.are.U 4500
 - U.are.U 4500 *Reader* adalah pembaca sidik jari yang memiliki port USB sehingga bisa dihubungkan ke laptop atau komputer. Alat ini memiliki performa membaca sidik jari yang cukup bagus sehingga memungkinkan untuk membaca sidik jari dari tangan yang kering, lembab, ataupun kasar.

- Pembaca Kartu
 - ACR120u
 - ACR120u merupakan pembaca kartu yang tidak hanya dapat membaca data dari kartu tetapi juga dapat menyimpan data ke kartu tersebut. ACR120u dibangun menggunakan teknologi RFID 13.56MHz yang mendukung kartu Mifare dan ISO 14443 A dan B. Jarak operasi maksimal 5 cm tergantung pada tipe *contactless tag*.
 - ACR 122T
 - ACR 122T merupakan pembaca kartu seperti ACR120U hanya ACR 122T merupakan generasi berikutnya yang bisa membaca NFC *tag*.

4.1.2 Lingkungan Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan pada lingkungan perancangan dan pengembangan aplikasi adalah sebagai berikut:

- Sistem Operasi Windows 8.1 Pro,
- MySQL Server,
- Flexcode SDK
- ACR120U SDK
- ACR122U SDK
- M. Office Word, Visio 2013.
- M. Visual Studio 2012

4.2 Implementasi perangkat Keras

Dalam pembuatan aplikasi *Smart Gate* ini dibutuhkan beberapa hardware penunjang diantaranya sebagai berikut :

- 1 buah laptop,
- 2 buah pembaca kartu RFID,

- 1 buah pembaca sidik jari,
- 4 buah kartu *tag* id dengan frekuensi 13.56Mhz, memori 1k
- 4 buah NFC *tag* id dengan frekuensi 13.56Mhz, berbentuk stiker

Dari beberapa perangkat keras diatas seperti pembaca kartu (RFID) dan pembaca sidik terlebih dahulu dihubungkan dengan laptop dengan menggunakan USB seperti Gambar 4.1., karena alat ini bukan berjenis *plug and play* jadi kedua alat ini membutuhkan pengenalan dengan meng-*install* perangkat lunak yang sudah tersedia.



Gambar 4.1. Perangkat keras

Perangkat lunak yang digunakan untuk mengenalkan perangkat keras adalah ACR12U SDK untuk perangkat keras pembaca jartu (RFID) dan Flexcode SDK untuk perangkat keras pembaca sidik jari. *Software* ini cukup diinstall dan mengambill *library* pada Microsoft Visual studio.

Delapan buah kartu *tag* dibutuhkan untuk menyimpan sampel data kendaraan sehingga sistem bisa mengenali informasi detail kendaraan. Dari kartu *tag* yang berbentuk 2 jenis yaitu kartu dan *stiker* digunakan pada alat yang berbeda. *Tag* yang berbentuk kartu digunakan pada kendaraan yang digunakan, sedangkan yang berbentuk *stiker* digunakan pengendara untuk dibawa disimpan sebagai *backup* atau cadangan ketika *tag* berbentuk kartu tidak bisa di baca.

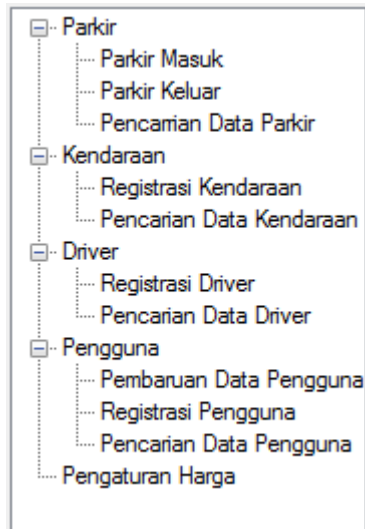
4.3 Implementasi Perangkat Lunak

Pada subbab ini akan dijelaskan tentang proses-proses yang dilakukan oleh aplikasi. Proses-proses yang dilakukan aplikasi mulai dari pembacaan data dari pembaca kartu dan pembaca sidik jari hingga hasil dari proses masuk dan keluar tempat parkir.

4.3.1 Implementasi Data Pengguna Aplikasi

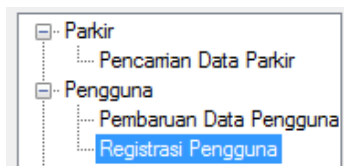
Fitur pengelola data pengguna digunakan untuk pengguna aplikasi yang memiliki hak akses sebagai super admin dan admin, Fitur ini digunakan untuk menambahkan, mengedit, dan menghapus pengguna aplikasi. Macam tipe dalam pendaftaran pengguna ada 3 yaitu super admin, Admin, dan Petugas yang masing-masing mempunyai hak akses yg berbeda seperti berikut :

- **Super Admin**
Super Admin merupakan tipe pengguna yang paling tinggi, yang pada implementasi nyata digunakan oleh pengguna pemilik jasa tempat parkir. Seperti pada Gambar 4.2. semua fitur yang terdapat pada aplikasi *Smart Gate* ini bisa dilakukan oleh super admin



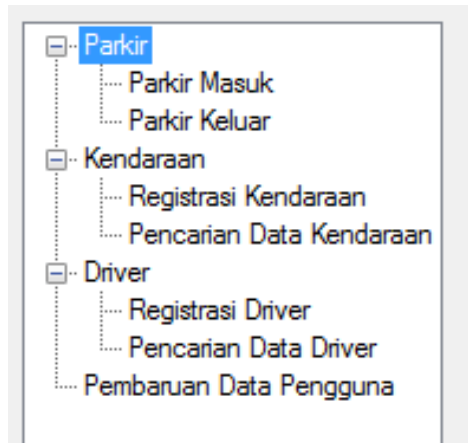
Gambar 4.2. Hak akses Pengguna Super Admin

- Admin
Admin merupakan tipe pengguna dibawah superadmin, tipe pengguna ini diperuntukkan oleh pemilik jasa tempat parkir yang hanya memiliki hak akses untuk mengelola data pengguna dan pencarian transaksi parkir yang sudah. Detail hak akses tipe admin bisa dilihat pad Gambar 4.3.



Gambar 4.3. Hak Akses Pengguna Admin

- **Petugas**
Petugas adalah tipe pengguna yang paling bawah, hak akses yang dimiliki tidak bisa mengelola data pengguna, tetapi memiliki akses untuk melakukan register *driver*, kendaraan dan melakukan transaksi keluar masuk parkir. Pada Gambar 4.4. tipe petugas tidak bisa untuk melihat transaksi yg sudah terjadi.



Gambar 4.4. Hak Akses Pengguna Petugas

4.3.2 Implementasi Data Kendaraan

Pada sub bab ini akan dijelaskan tentang pengambilan data informasi kendaraan. Sistem aplikasi *Smart Gate* perlu mengenali id kendaraan tertentu sebelum kemudian diproses kedalam sistem. Pengenalan terhadap kendaraan digunakan sebuah *tag* RFID yang berupa kartu dan bisa dipasangkan ke masing-masing kendaraan. Bentuk kartu *tag* yang digunakan seperti pada Gambar 4.5.



Gambar 4.5. Kartu Tag kendaraan

Data kendaraan yang dibutuhkan untuk sistem ini berupa informasi nomor polisi dan jenis kendaraan seperti motor atau mobil. Sebelum data tersebut di tulis ke *tag* masing-masing, data tersebut disimpan kedalam *database server*, bentuk penyimpanan kedalam *database* bisa dilihat pada Gambar 4.6. Kemudian kartu *tag* untuk tiap kendaraan yang terdaftar hanya ditulis nomor polisi yang penulisannya sudah di enkripsi menggunakan TDES.

ID_KENDARAAN	NOPOL	STATUS	FOTO	idJenisKendaraan
9	AE4128BA	0	NULL	4
10	DA4354CB	1	NULL	4
11	B1234SE	1	NULL	4
12	L1234SE	1	NULL	3
13	AE4018SE	1	NULL	4

Gambar 4.6. Database kendaraan

Kartu *tag* yang digunakan memiliki arsitektur memori berbentuk *sector* dan *blok* dimana pada satu *sector* memiliki beberapa blok yang masing-masing blok dapat diisi file berbentuk *String* seperti yang terlihat pada Gambar 4.7. Pada proses penulisan data ke dalam kartu *tag* harus diketahui

terlebih dahulu lokasi memori yang akan digunakan untuk penulisan maupun pembacaan, karena jika alamat memori untuk penulisan dan pembacaan tidak sama maka sistem akan *error*.

Sectors (Total 16 sectors. Each sector consists of 4 consecutive blocks)	Data Blocks (3 blocks, 16 bytes per block)	Trailer Block (1 block, 16 bytes)
Sector 0	0x00 ~ 0x02	0x03
Sector 1	0x04 ~ 0x06	0x07
..		
..		
Sector 14	0x38 ~ 0x0A	0x3B
Sector 15	0x3C ~ 0x3E	0x3F

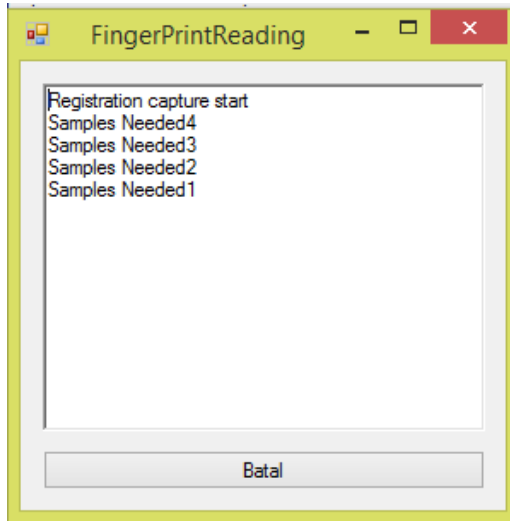
} 1K Bytes

Gambar 4.7. Arsitektur Memori Dalam Tag

4.3.3 Implementasi Data Driver

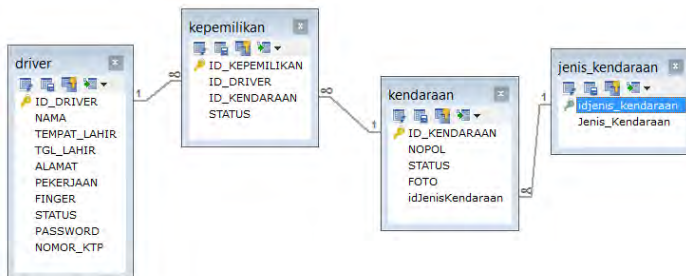
Setiap manusia memiliki bermacam macam karakteristik yang berbeda-beda satu sama lainnya, seperti sidik jari, mata, dan lain lain. Pada sitem *Smart Gate* ini menggunakan sidik jari sebagai pengenalan identitas *driver* kedalam sistem.

Pertama adalah mencatat semua informasi *driver* berupa nomor ktp, nama, tempat tanggal lahir, alamat, pekerjaan dan *signature* dari sidik jari kedalam *database*. Untuk memasukkan data sidik jarinya dibutuhkan perakaman sidik jari menggunakan alat *fingerprint reader* dengan 4 kali rekaman seperti yang terlihat pada Gambar 4.8. 4x masukkan sidik jari disini hanya untuk meningkatkan akurasi dan hanya dilakukan sekali saja ketika akan disimpan ke dalam *database*.



Gambar 4.8. Perekaman Sidik Jari

Pada kasus nyata di tempat parkir, *driver* bisa memiliki lebih dari satu kendaraan pribadi, dan satu kendaraan tersebut juga bisa digunakan oleh beberapa *driver*, maka terbentuk relasi *many to many* antara tabel kendaraan dan tabel *driver*, dengan demikian harus menambahkan satu tabel baru yaitu tabel kepemilikan seperti pada Gambar 4.9.



Gambar 4.9. Relasi Tabel *Driver* dan Kendaraan

4.3.4 Implementasi Transaksi Parkir Masuk dan Keluar

Pada transaksi masuk dan keluar secara implementasi memiliki kemiripan proses yaitu pembacaan data dari kartu *tag* dan pembacaan sidik jari *driver*, yang membedakan adalah pada saat masuk, id parkir akan dituliskan kembali kedalam kartu *tag* untuk mempermudah pencarian pada saat keluar parkir. Pada Gambar 4.10. merupakan proses pembacaan data dari kartu *tag* sehingga didapat id kendaraan.

Pertama kartu *tag* dibaca oleh alat utama yaitu ACR120u apabila kartu tidak sesuai atau tidak ada maka pembacaan dilakukan oleh alat cadangan yaitu ACR122u apabila kartu sudah terbaca dan sesuai, maka id yang dibaca diambil oleh sistem.

```

read tagID from acr120Device
if card not suported
    print "kartu tidak dikenal"
    return
if tagID reading isn't succeed
    read tagID from acr122U device
    if tagID reading failed
        print "fail"
        return
    assign id to tagID
else
    assign id to tagID
  
```

Gambar 4.10. Pseudocode Pembacaan Kartu Tag

Pada Gambar 4.11. adalah *pseudocode* untuk pembacaan sidik jari *driver* ketika akan memasuki tempat parkir. Dengan mencari terlebih dahulu kepemilikan dari kendaraan kemudian sidik jari di cocokan dengan sidik yang sudah tersimpan dengan *database*.

```

fingerprint verification
Set FPver to the Device Class
Add the Device Info into the FPVer

Set milik to the Id Kepemilikan and fingerprint data from
database based on incoming vehicle.

Load milik to the FPVer

Start the verification

if the is no device for fingerprint reading
    print "Tolong Sambungkan Perangkat Pembaca
    Fingerprint"
    return

else if there is no fingerprint Matched
    print "Tidak Ada Fingerprint Yang Cocok"
    return

else if verification failed
    print "Verifikasi gagal"
    return

else if succeed
    stop verification
    return id kepemilikan

```

Gambar 4.11. Pseudocode Pembacaan Sidik Jari

Setelah proses pembacaan kartu *tag* dan sidik jari berhasil, maka id parkir yang mewakili transaksi masuk parkir dimasukkan atau di tulis di kartu *tag* seperti pada Gambar 4.12.

Jneis kendaraan di cari terlebih dahulu dengan id kendaraan, lalu berdasarkan jenis kendaraan maka id harga bisa

ditentukan yang kemudian transaksi baru disimpan ke *database*. Transaksi yang terakhir diambil id parkir yang kemudian ditulis pada kartu *tag* yang sama pada saat pembacaan kartu *tag* sebelumnya.

```

Set vehicleType to type of incoming vehicle
Set hargalD to result from query of the table Price based
on vehicleType
Save the hargalD, Date, and driver of the incoming vehilce
to transaksi table in the DataBase
get IDParking result from query of the last traksaksi
Save the IDParking to the card
if the writing card is success
    print "Verifikasi Berhasil"
else
    print "Verifikasi Gagal"
    delete Parking from Database

```

Gambar 4.12. Pseudocode Penulisan ke Kartu Tag

4.4 Implementasi Pengamanan Dalam Kartu

Triple DES adalah istilah umum untuk *Triple Data Encryption Algorithm* (TDEA atau *Triple DEA*). *Triple DES* mengimplementasikan algoritma *Data Encryption Standard* (DES) tiga kali untuk tiap blok data.

.NET telah menyediakan *class* pembungkus untuk mengakses *Triple DES* versi *Cryptographic Service Provider* (CSP). Untuk menggunakan *Triple DES* milik .NET ini, pertama kita perlu mendefinisikan *key* yang akan dipakai. Dalam implementasi perangkat lunak *Smart Gate* ini, digunakan *key* berbentuk *string password* yang di-*hardcode*. Kemudian *key* tersebut *dienkode* menggunakan UTF8 dan dihitung *hash value*-nya. Hasil *hashvalue* tersebut akan

dijadikan *key* pada *Triple DES*. Detail untuk menghasilkan *key* yang dipakai oleh *Triple DES*.

Setelah berhasil menghasilkan *key* untuk *Triple DES*, selanjutnya adalah proses enkripsi atau deskripsi *string* yang diinginkan. *String* yang akan dienkripsi atau dideskripsi akan diencode menggunakan UTF8 kemudian dengan menggunakan *Class Triple DES Crypto Service Provider* kita dapat memanggil fungsi enkripsi atau deskripsi. Detail proses enkripsi dapat dilihat pada *pseudocode* Gambar 4.. Untuk proses deskripsi dapat dilihat pada Gambar 4..

Setelah proses enkripsi atau deskripsi berhasil, hal yang terakhir dilakukan adalah mengkonversi hasil enkripsi atau deskripsi dari bentuk *byte* ke dalam bentuk *string*.

```
String function Encrypt(Message)
set key to "asa"
set TDESkey to Hash value of the key
set DataToEncrypt to byte value of Message

set Result to encryption result of DataToEncrypt from
TripleDESCryptoServiceProvider using the TDESkey
return Result
```

Gambar 4.13. Pseudocode Enkripsi TDES

```
String function Decrypt(Message)
Set key to "asa"
set TDESkey to Hash value of the key
set DataToDecrypt to byte value of Message

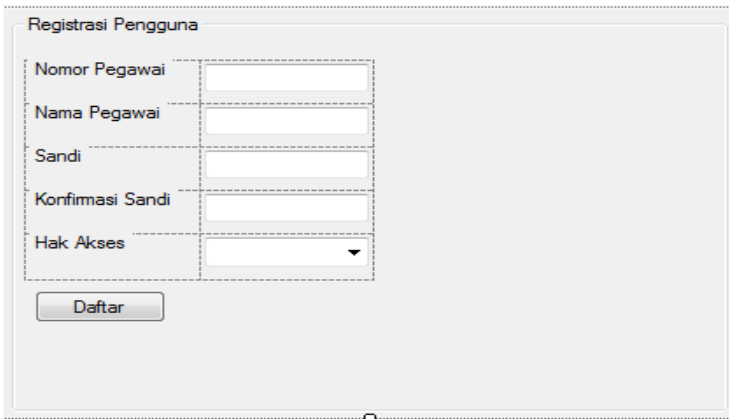
set Result to decryption result of DataToEncrypt from
TripleDESCryptoServiceProvider using the TDESkey
return Result
```

Gambar 4.14. Pseudocode Dekripsi TDES

4.5 Implementasi Antarmuka

Pada sub-bab ini akan dijelaskan mengenai implementasi antarmuka perangkat lunak. Implementasi dilakukan sesuai dengan perancangan yang telah dilakukan pada Bab III.

4.5.1 Implementasi Antarmuka Pembuatan Pengguna



The image shows a user registration form titled "Registrasi Pengguna". The form contains the following fields:

Nomor Pegawai	<input type="text"/>
Nama Pegawai	<input type="text"/>
Sandi	<input type="text"/>
Konfirmasi Sandi	<input type="text"/>
Hak Akses	<input type="text"/>

Below the form is a button labeled "Daftar".

Gambar 4.13. Implementasi Antar Muka Pembuatan Pengguna

Gambar 4.13. menunjukkan implementasi dari antarmuka pembuatan pengguna. Antarmuka ini terdiri dari *field-field* yang perlu diisi oleh pengguna untuk membuat pengguna baru, sesuai dengan rancangan awal maka semua tempat isian data pengguna menggunakan *textbox* yang mengambil data berbentuk *string*. Pemilihan hak akses menggunakan *combobox* yang sudah disediakan pilihan berupa 3 aktor pengguna aplikasi.

4.5.2 Implementasi Antarmuka Melihat Laporan

Pada Gambar 4.14. merupakan implementasi antarmuka untuk mencari daftar parkir yang di cari berdasarkan tanggal yang diinginkan dalam rentang waktu tertentu. Data yang diproses adalah data dari hasil *query* tabel transaksi yang berada di *database* yang ditampilkan sedetail mungkin untuk memperjelas status parkir yang terjadi.



Gambar 4.14. Implementasi Antarmuka Laporan Parkir

4.5.3 Implementasi Antarmuka Menentukan Harga

Pada implementasi antarmuka menentukan harga diberikan *field-field* untuk menentukan harga seperti harga dasar, harga perjam dan harga maksimal. Dari berbagai macam harga yang ditentukan bisa memilih salah satu yang akan dijadikan harga parkir pada saat itu seperti terlihat pada Gambar 4.15.

Harga bisa disesuaikan jenis kendaraan yang sudah ditentukan pada *combobox* dan tariff harga bisa diisi sesuai keinginan penyedia. Dari hasil harga yang sudah ditentukan akan ditampilkan pada data *grid* dan perlu pemilihan pada

tombol yang sudah disediakan sebagai tanda bahwa harga tersebut yang digunakan.

Pengaturan Harga

Jenis Kendaraan

Penambahan Harga

Harga Dasar

Harga / Jam

Harga Maks

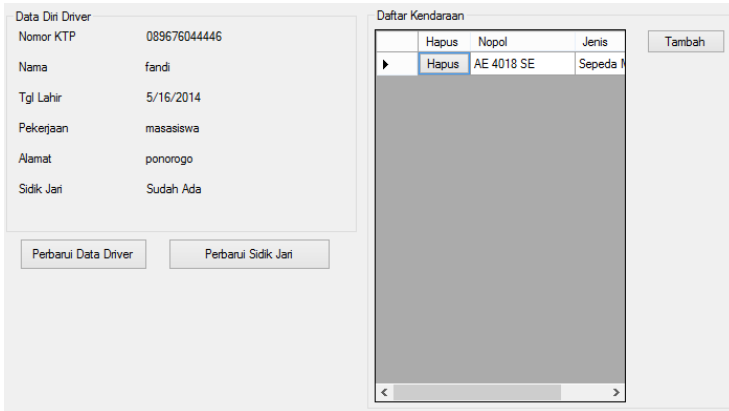
Daftar Harga

	Pilih	Hapus	Harga_Dasar	Harga_per_Jam	Harga_Maks	Status
▶	<input type="button" value="Pilih"/>	<input type="button" value="Hapus"/>	5000	1000	20000	Dipakai
	<input type="button" value="Pilih"/>	<input type="button" value="Hapus"/>	3000	1000	15000	

Gambar 4.15. Implementasi Antarmuka Harga

4.5.4 Implementasi Antarmuka Mengelola Data *Driver*

Pada implementasi antarmuka mengelola data *driver* terdiri dari beberapa bagian seperti data detail dari *driver* yang sudah terdaftar. Pada Gambar 4.16. bisa dilihat ada kolom sebelah kanan terdapat daftar jenis kendaraan yang sudah digunakan oleh *driver*. Selain itu juga terdapat 3 tombol seperti memperbarui data *driver*, perbarui sidik jari, dan tombol tambah untuk menambah kendaraan yang digunakan.



Gambar 4.16. Implementasi Antarmuka *Driver*

4.5.5 Implementasi Antarmuka Mengelola Data Kendaraan

Pada Gambar 4.16 terdapat tombol perbarui sidik jari yang ketika di panggil maka antarmuka akan dibawa ke antarmuka perekaman sidik jari yang membutuhkan 4 kali pembacaan sidik jari. Tombol perbarui data *driver* akan menuju ke antarmuka pendaftaran *driver* yang kemudian *database* diperbarui.

Pada implementasi antarmuka mengelola data kendaraan memiliki antarmuka yang hampir sama dengan mengelola data *driver*. Selain menampilkan data detail kendaraan juga menampilkan daftar *driver* yang menggunakan kendaraannya dikolom bagian kanan. Selain itu seperti pada Gambar 4.17. juga terdapat 3 tombol yang digunakan untuk perbarui data, isi data ke kartu dan tambah *driver* yang akan menggunakannya.

The screenshot displays a software interface for vehicle management. It is divided into two main sections: 'Data Kendaraan' (Vehicle Data) and 'Daftar Driver' (Driver List).

Data Kendaraan:

- Nomor Polisi: AE 4018 SE
- Jenis: Sepeda Motor

Below this section are two buttons: 'Perbarui Data' and 'Isi Data ke Kartu'.

Daftar Driver:

This section contains a table with the following columns: 'Hapus', 'No_KTP', and 'Nama'. A 'Tambah' button is located to the right of the table.

Hapus	No_KTP	Nama
Hapus	089676044446	fandi

The table has a scroll bar at the bottom, indicating it can display more records.

Gambar 4.17. Implementasi Antarmuka Kendaraan

4.5.6 Implementasi Antarmuka Transaksi Parkir Masuk dan Keluar

Desain untuk antarmuka transaksi parkir masuk dan keluar sama persis. Meski proses bisnis ketika transaksi masuk dan keluar terdapat sedikit perbedaan namun tetap dapat menggunakan desain antarmuka yang sama. Tetapi meski memiliki desain antarmuka yang sama, untuk memudahkan implementasi perangkat lunak, maka digunakan antarmuka yang berbeda untuk dua proses bisnis tersebut meskipun desainnya sama persis. Detail implementasi antarmuka dapat dilihat pada Gambar 4.18.

Pada antarmuka transaksi masuk, penggunaan tombol pembacaan kartu *tag* dihilangkan karena dalam antarmuka parkir masuk dibuat otomatis pembacaan kartu dan sidik jari untuk meningkatkan otomatisasi.

Letakkan Kartu di Pembaca

Isi Data Parkir ke Kartu

Data Kendaraan

Nomor Polisi	AE 4018 SE
Jenis	Sepeda Motor

Data Diri Driver

Nomor KTP	089676044446
Nama	fandi
Alamat	ponorogo

Gambar 4.18. Implementasi Antarmuka Parkir

Pada Gambar 4.18 terbentuk setelah *driver* menempelkan kartu pada RFID dan sidik jari nya ke *fingerprint reader*. Maka hasilnya informasi kendaraan dan *driver* akan ditampilkan untuk pengecekan. Kemudian memasukkan data parkirnya ke *database* sekaligus memasukkan id parkir ke dalam kartu *tag*.

4.6 Implementasi *Back up device*

Pada sub bab ini menjelaskan tentang *backup* alat yang digunakan apabila alat utamanya mengalami gagal atau kerusakan. Pada alat pembaca kartu utama menggunakan RFID dan untuk *backup* menggunakan NFC yang memiliki fungsi sama dengan alat utamanya. Namun pada penggunaan *backup* alat digunakan sebuah kode yang ditulis kedalam kartu *tag*, sehingga kartu *tag* hanya bisa dibaca pada alat yang sama. sedangkan pada alat utama kedua yaitu pembaca sidik jari menggunakan masukkan KTP secara manual sebagai *backup* nya.

[Halaman ini sengaja dikosongkan]

BAB V

UJI COBA DAN ANALISA

Pada bab ini akan membahas uji coba dan evaluasi dari sistem yang dibuat. Sistem akan diuji coba secara fungsionalitas dengan menjalankan skenario yang sudah ditentukan.

5.1 Lingkungan Uji Coba

Pada sub-bab ini dijelaskan mengenai gambaran lingkungan yang digunakan untuk melakukan uji coba aplikasi. Uji coba aplikasi ini dilakukan dengan menggunakan sebuah laptop, *RFID reader*, *NFC reader* dan *fingerprint reader* dengan spesifikasi sebagai berikut :

- Laptop Toshiba L745
 - Spesifikasi perangkat keras
 - Processor Intel(R) Core(TM) i5-2410M CPU @ 2.30GHz,
 - RAM 4 GB.
 - Spesifikasi perangkat lunak
 - Windows 8 Ultimate 32-bit sebagai sistem operasi yang digunakan.
 - Visual Studio 2012 untuk mengimplementasikan system.
 - MySQL 5.6.16 sebagai *database* yang digunakan.
- Perangkat Pembaca Kartu
 - Tipe ACR 120U dan Tipe ACR 122T
 - Frekuensi 13.56Mhz
- Perangkat Pembaca Sidik Jari
 - Digital Persona U are U 4500
 - Flexcode SDK

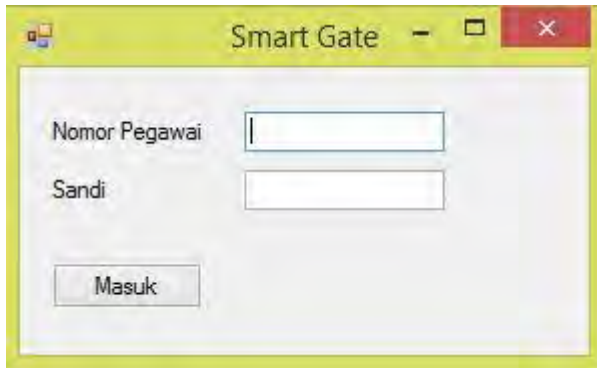
- Perangkat Kartu *Tag*
 - Mifare dengan memori 1K berbentuk kartu
 - Mifare dengan memori 1K berbentuk stiker
 - Frekuensi 13.56Mhz

5.2 Skenario Uji Coba

Uji coba ini dilakukan untuk menguji apakah fungsionalitas yang diidentifikasi pada tahap kebutuhan benar-benar diimplementasikan dan bekerja seperti yang seharusnya.

5.2.1 Uji Coba *Login* ke Sistem *Smart Gate*

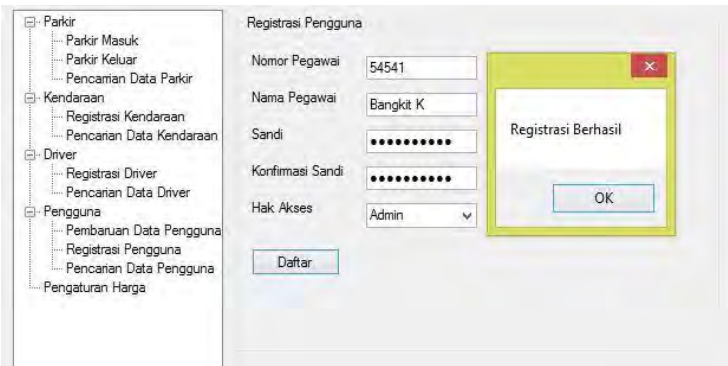
Pada uji coba berikut ini adalah hal yang pertama kali dilakukan untuk memasuki aplikasi dengan *login* oleh pengguna yang sudah terdaftar seperti pada Gambar 5.1. Informasi yang dibutuhkan untuk memasuki sistem dengan menggunakan nomor pegawai dan sandi.



Gambar 5.1. *Login* Pengguna

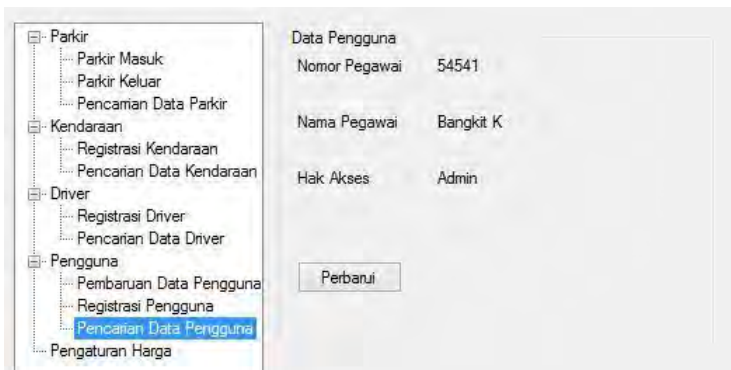
5.2.2 Uji Coba Pengolahan Pengguna

Pada uji coba pengolahan data pengguna ada 3 tahapan yaitu pendaftaran pengguna baru, melihat detail pengguna dan pencarian pengguna.



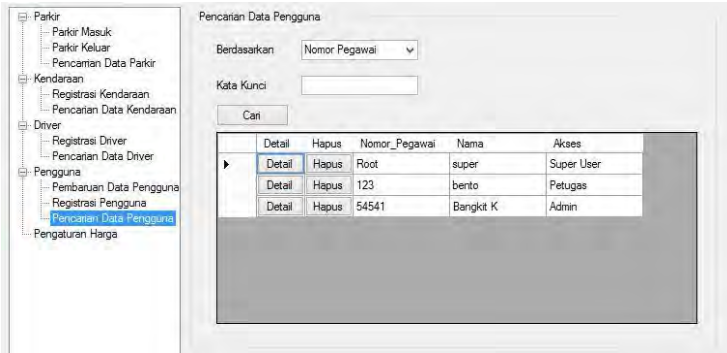
Gambar 5.2. Menu Pendaftaran Pengguna Baru

Pada Gambar 5.2. adalah menu pendaftaran pengguna baru dengan memasukkan identitas yang dibutuhkan beserta hak akses yang akan digunakan untuk pengguna baru tersebut. Setelah data berhasil di proses maka akan keluar *message box* sebagai indikasi bahwa pendaftaran berhasil dilakukan.



Gambar 5.3. Tampilan Detail pengguna

Pada Gambar 5.3. merupakan hasil dari pendaftaran pengguna baru, pada menu ini terdapat tombol perbarui yang digunakan untuk memperbarui data – data yang sudah dimasukkan. Pada saat tombol perbarui ini digunakan maka menu akan kembali seperti pada Gambar 5.2. untuk memasukkan data kembali.



Gambar 5.4. Menu Pencarian Data Pengguna

Menu yang terakhir pada pengolahan data pengguna yaitu menu pencarian pengguna. Seperti pada Gambar 5.4. pencarian bisa berdasarkan nomor pegawai, hak akses, Nama bahkan bisa semua dengan mengosongkan kata kunci. Data yang sudah tersimpan di *database* akan ditampilkan di kolom dengan fitur detail dan hapus yang dapat dilakukan sesuai kebutuhan. Fitur detail apabila dilakukan akan kembali seperti Gambar 5.3. sedangkan fitur hapus akan menghapus pengguna yang sudah tidak bekerja lagi, namun penghapusan tidak sepenuhnya menghapus semua data pengguna hanya mengubah statusnya di dalam *database* untuk membedakan pengguna ini masih aktif atau tidak.

5.2.3 Uji Coba Pengolahan Data *Driver*

Pada pengujian system *driver* terdapat beberapa menu seperti pendaftaran *driver*, perekaman sidik jari, dan pencarian *driver*.

Gambar 5.5. Menu Pendaftaran *Driver*

Gambar 5.6. Menu Detail *Driver*

Pengendara yang belum pernah mendaftar diharuskan mendaftar terlebih dahulu sebelum memasuki parkir, seperti Gambar 5.5. pengendara diminta oleh petugas informasi yang dibutuhkan untuk pendaftaran. Ketika pendaftaran telah usai menu akan berubah menjadi menu detail *driver* seperti pada Gambar 5.6. Pada menu detail ditampilkan informasi *driver*

yang sudah tersimpan beserta fitur untuk memperbarui dan merekam sidik jari *driver*. Fitur perbarui data *driver* jika dilakukan akan kembali ke Gambar 5.5. seperti halnya memasukkan data pengguna baru. Pada Gambar 5.6. diatas terdapat kolom kendaraan disebelah kanan, karena *driver* ini baru mendaftar maka kolom kendaraan masih dalam keadaan kosong karena belum mendaftarkan kendaraan, begitu pula dengan status sidik jari *driver* masih dalam keadaan belum ada karena *driver* belum melakukan perekaman sidik jari.

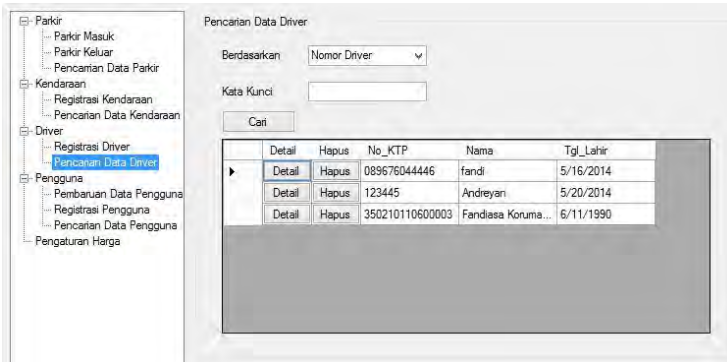


Gambar 5.7. Menu Perekaman Sidik Jari

Pada Gambar 5.7. adalah menu ketika perekaman sidik jari diproses, pada proses perekaman *driver* di minta untuk menempelkan sidik jari 4x sampai keluar tulisan *sample needed* berjumlah 4x di kolom sebelah kiri. Proses yang membutuhkan 4x ini hanya dilakukan ketika awal pendaftaran karena proses ini dibutuhkan untuk alat pembaca sidik jari untuk meningkatkan akurasi. Untuk proses parkir tidak perlu menggunakan 4x pembacaan. Setelah perekaman sidik jari selesai, status sidik jari yang nampak seperti pada Gambar 5.6. berubah menjadi “Sudah Ada”.

Pada menu berikutnya adalah menu pencarian data *driver*. Seperti yang ditampilkan pada Gambar 5.8. menu pencarian kendaraan memiliki fitur yang sama dengan

pencarian pengguna dan kendaraan, hanya yang membedakan adalah tabel data di *database* yang dituju berbeda.



Gambar 5.8. Menu Pencarian *Driver*

Pencarian berdasarkan nomor KTP *driver* atau nama, hasil pencarian akan di tampilkan pada kolom *grid* dan masing-masing memiliki fitur untuk melihat *detail* dan di hapus. Untuk fitur detail akan kembali ke menu seperti Gambar 5.6. dan untuk fitur hapus seperti halnya pengguna tidak sepenuhnya dihapus hanya status di dalam *database* saja yang di rubah.

5.2.4 Uji Coba Pengolahan Data Kendaraan

Pada menu-menu pengolahan data kendaraan memiliki kemiripan dengan pengolahan data pengguna dan *driver* sebelumnya, yaitu terdiri dari menu pendaftaran, detail kendaraan dan pencarian kendaraan, tetapi yang membedakan untuk kendaraan terdapat fitur menulis data kendaraan di kartu dan penambahan *driver* yang akan menggunakan kendaraan.

Pada Gambar 5.9. merupakan menu pendaftaran kendaraan, data yang dibutuhkan hanya nomor kendaraan dan jenis kendaraan berupa mobil atau sepeda motor. Setelah pendaftaran selesai maka menu akan berlanjut ke menu detail kendaraan seperti pada Gambar 5.10.

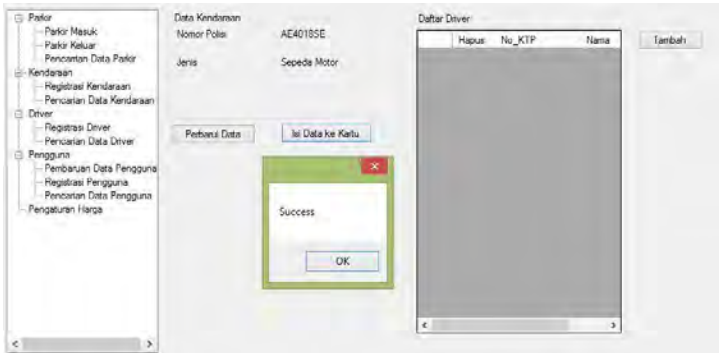


Gambar 5.9. Menu Pendaftaran Kendaraan



Gambar 5.10. Menu Detail Kendaraan

Pada Gambar 5.10. secara tampilan memiliki kemiripan dengan menu *driver*, yang membedakan adalah terdapat fitur untuk menulis data ke kartu yang sudah disediakan. Sebelum penulisan ke kartu pastikan bahwa kartu *tag* sudah di dekatkan dengan alat pembaca kartu. Hasil penulisan akan nampak pada Gambar 5.11.



Gambar 5.11. Menu Pengisian Data ke Kartu

Data yang dituliskan ke dalam kartu adalah berupa id kendaraan dan nomor kendaraan, namun untuk mengantisipasi penggandaan kartu oleh pihak yang tidak bertanggung jawab maka penulisan menggunakan enkripsi dan penentuan lokasi memori yang sudah di sesuaikan oleh pemilik aplikasi atau pemilik parkir.

Pada Gambar 5.11. menunjukkan keberhasilan pengisian data ke kartu *tag*, karena pengisian nomor kendaraan yang memiliki panjang karakter yang banyak maka pada penulisan dalam *tag* membutuhkan 2 blok memori untuk penyimpanan.

Data kendaraan, *driver* sudah terdaftar ke dalam *database*, untuk berikutnya adalah mengatur kepemilikan dari kendaraan dengan *driver* yang bersangkutan. Pada gambar Gambar 5.10. terdapat kolom *driver* disebelah kanan, cukup dengan menambahkan dengan tombol tambah maka akan keluar menu pencarian *driver* seperti pada Gambar 5.12. untuk selanjutnya dipilih sebagai pemilik dari kendaraan tersebut. Kepemilikan ini juga bisa di setting pada menu *detail driver*, dan setelah kepemilikan dipilih maka pada kolom kanan akan berisi daftar *driver* yang sudah terdaftar sebagai pemilik kendaraan seperti pada Gambar 5.13.

Pencarian Data Driver

Berdasarkan

Kata Kunci

	No_KTP	Nama	Tgl_Lahir
<input type="checkbox"/>	089676044446	fandi	5/16/2014
<input type="checkbox"/>	123445	Andreyan	5/20/2014
<input checked="" type="checkbox"/>	350210110600003	Fandiasa Koruma...	6/11/1990

Gambar 5.12. Menu Penambahan Kepemilikan Kendaraan

Parkir

- Parkir Masuk
- Parkir Keluar
- Pencarian Data Parkir

Kendaraan

- Registrasi Kendaraan
- Pencarian Data Kendaraan

Driver

- Registrasi Driver
- Pencarian Data Driver

Pengguna

- Pembaruan Data Pengguna
- Registrasi Pengguna
- Pencarian Data Pengguna
- Pengaturan Harga

Data Kendaraan

Nomor Polisi AE4018SE

Jenis Sepeda Motor

Daftar Driver

	Hapus	No_KTP	Nama
<input type="checkbox"/>	<input type="checkbox"/>	350210110600003	Fandiasa

Gambar 5.13. Hasil Penambahan Kepemilikan

5.2.5 Uji Coba Transaksi Parkir

Pada uji coba transaksi parkir terdapat 2 menu utama yaitu parkir masuk dan parkir keluar, dimana dari kedua fitur parkir tersebut memiliki menu dan fitur yang sama. Dalam menu parkir terdiri dari pembacaan kartu dan diikuti pembacaan sidik jari *driver* seperti yang ditunjukkan pada Gambar 5.14.



Gambar 5.14. Menu parkir masuk dan keluar

Pada Gambar 5.14. tersebut ketika *driver* telah melakukan pembacaan kartu kendaraan sehingga hasil detail kendaraan yang terbaca pada kartu ditampilkan pada layar. Kemudian layar akan menampilkan perintah kepada *driver* untuk memasukkan sidik jarinya ke pembaca sidik jari, tetapi pada fitur parkir ini telah diberikan fitur tambahan apabila terjadi kesalahan pada pembaca sidik jari maka dapat digantikan dengan memasukkan nomor KTP secara manual dengan menekan tombol besar yang sudah disediakan. Alternatif nomor KTP ini digunakan apabila berada pada keadaan terdesak sehingga *driver* tidak bias memasukkan sidik jari, namun harus berada pada pengawasan petugas jaga untuk mengklarifikasi kebenaran KTP *driver*. Keberhasilan proses pembacaan sidik jari akan menampilkan semua informasi detail kendaraan dan *driver* seperti pada gambar Gambar 5.15.



Gambar 5.15. Hasil Pembacaan Data

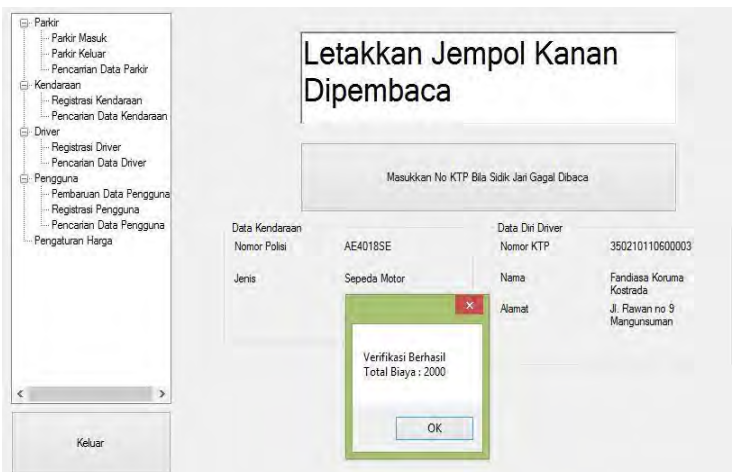


Gambar 5.16. Keberhasilan Proses Verifikasi

Pada Gambar 5.16. merupakan hasil proses verifikasi setelah pembacaan data dari kartu dan sidik jari. Pada proses ini makaterjadi pula proses penulisan id parkir ke dalam kartu sebagai tanda bahwa kartu ini atau kendaraan ini sedang parkir

yang akan digunakan syarat sebagai proses pada keluar parkir berikutnya.

Pada proses keluar parkir hal yang sama dilakukan seperti ketika masuk parkir yaitu dengan pembacaan kartu dan sidik jari, dan data tersebut kemudian di samakan dengan data yang sudah tersimpan di *database* ketika masuk. Keberhasilan proses keluar adalah dengan keluarnya tampilan berupa harga parkir yang sudah ditentukan tarifnya berdasarkan jam dan bias dilihat pada Gambar 5.17.



Gambar 5.17. Menu Proses Keluar parkir

Dalam hal pembukuan atau laporan mengenai segala macam transaksi parkir yang sudah terjadi juga terdapat fitur untuk melakukan pencarian parkir. Dengan fitur ini pemilik jasa parkir dapat melihat riwayat parkir, sekaligus dapat memperhitungkan berapa laba dari parkir yang sudah terjadi seperti ditunjukkan pada Gambar 5.18.

Nopol	Supir_Masuk	Supir_Keluar	Masuk	Keluar	Biaya
AE4128BA	fandi	fandi	5/16/2014 10:34...	5/16/2014 10:35...	2000
AE4128BA	fandi	fandi	5/16/2014 10:41...	5/16/2014 10:41...	2000
L1234SE	asa	asa	5/22/2014 12:42...	5/22/2014 12:43...	5000
AE4018SE	Fandasa Koruma...	Fandasa Koruma...	5/23/2014 7:29 ...	5/23/2014 7:30 ...	2000

Gambar 5.18. Laporan Riwayat Parkir

5.2.6 Uji Coba Performa Sistem

Pada bagian ini dilakukan uji coba performa untuk mengetahui performa dari sistem ketika uji coba pada keadaan yang sebenarnya. Uji coba ini dilakukan untuk mengetahui kecepatan dan keberhasilan sistem dalam menjalankan fungsinya. Terdapat satu skenario uji coba yang dilakukan untuk mengetahui performa sistem ini. Hasil dari skenario uji coba tersebut terdiri dari tiga parameter, yaitu pembacaan id ada tag, pembacaan sidik jari pengendara, dan proses pemasukan transaksi parkir sampai mendapat keberhasilan. Dari masing-masing parameter tersebut juga dibedakan berdasarkan system parkir masuk dan keluar yang memiliki perbedaan proses. Pada table x.x dijelaskan secara keseluruhan hasil uji coba kecepatan sistem dalam melakukan pembacaan dan proses parkir beserta status keberhasilan dari sistem.

Tabel 5. 1. Uji Coba Waktu Sistem Parkir *Smart Gate*

No.	<i>Read_Tag</i> (detik)	<i>Read Fingerprint</i> (detik)	<i>Parking_ in/out</i> (detik)	Keterangan
1.	2	5	5	Berhasil
2.	4	6	6	Berhasil
3.	1	4	5	Berhasil
4.	1	3	5	Berhasil
5.	2	11	8	Berhasil
6.	3	3	6	Berhasil
7.	4	4	6	Berhasil
8.	2	8	5	Berhasil
9.	2	10	6	Berhasil
10.	4	15	5	Berhasil
RataRata	2.5	6.9	5.7	15.1

Pada Tabel 5. 1 merupakan sebuah uji coba efisiensi waktu yang digunakan oleh aplikasi *Smart Gate*. Dalam ujicoba terdapat 10 kali percobaan dengan selisih waktu yang berbeda-beda dan dari keseluruhan hasil uji coba diambil nilai waktu rata-rata dan total keseluruhan rata-rata waktu tiap parameter. Pada hasil yang sudah di ujicoba sistem ini membutuhkan waktu rata-rata 15.1 detik untuk melakukan pengidentifikasi kendaraan maupun *driver* sampai dalam kondisi berhasil dan pintu gerbang terbuka. Dari hasil rata-rata waktu diatas maka sistem ini dapat menghemat waktu identifikasi lebih cepat 20 detik daripada menggunakan system manual yang rata-rata membutuhkan waktu sekitar 35 detik.

[Halaman ini sengaja dikosongkan]

BAB VI KESIMPULAN DAN SARAN

Pada bab ini berisi kesimpulan dan saran yang dapat diambil dari tugas akhir adalah sebagai berikut.

6.1 Kesimpulan

Berdasarkan hasil pengamatan yang telah dilakukan dari tahap perancangan, kemudian implementasi, hingga uji coba aplikasi dapat diambil kesimpulan sebagai berikut:

1. Sistem ini dapat digunakan untuk meningkatkan keamanan dan pengamanan pada lokasi tempat parkir dengan adanya penggunaan sidik jari yang tidak dapat dipalsukan sehingga dapat menekan angka pencurian kendaraan bermotor.
2. Sistem dapat menambah otomatisasi para penyedia jasa tempat parkir maupun para pengendara bermotor dengan adanya sistem serba otomatis dari alat yang digunakan termasuk penentuan harga parkir.
3. Dengan menggunakan sistem ini, maka metode pengolahan lebih efisien dari segi waktu yang hanya membutuhkan rata-rata 15.1 detik dan lebih menghemat 20 detik dari sistem manual maupun sumber daya manusia yang dapat dikurangi pada tempat masuk parkir.

6.2 Saran

Adapun saran-saran yang dapat diberikan untuk pengembangan aplikasi ini berikutnya adalah:

1. Mengingat kemajuan teknologi yang semakin pesat, implementasi aplikasi ini masih jauh dari kesempurnaan untuk meningkatkan keamanan di tempat parkir. Untuk itu diperlukan adanya improvisasi atau penambahan fitur baru untuk mengamankan data yang tersimpan sehingga

sulit untuk diretas oleh pihak yang tidak bertanggung jawab

2. Untuk pengembangan selanjutnya aplikasi ini diharapkan mampu menggunakan alat yang mampu menjangkau kartu dengan jarak yang lebih jauh sehingga bisa diterapkan pada semua jenis kendaraan.
3. Pada *Database* kepemilikan kendaraan dan pengendara yang digunakan untuk kedepannya bisa digunakan secara umum dan bisa digunakan untuk jenis aplikasi yang lain yang mengacu pada database yang sama.

BAB VI KESIMPULAN DAN SARAN

Pada bab ini berisi kesimpulan dan saran yang dapat diambil dari tugas akhir adalah sebagai berikut.

6.1 Kesimpulan

Berdasarkan hasil pengamatan yang telah dilakukan dari tahap perancangan, kemudian implementasi, hingga uji coba aplikasi dapat diambil kesimpulan sebagai berikut:

1. Sistem ini dapat digunakan untuk meningkatkan keamanan dan pengamanan pada lokasi tempat parkir dengan adanya penggunaan sidik jari yang tidak dapat dipalsukan sehingga dapat menekan angka pencurian kendaraan bermotor.
2. Sistem dapat menambah otomasi para penyedia jasa tempat parkir maupun para pengendara bermotor dengan adanya sistem serba otomatis dari alat yang digunakan termasuk penentuan harga parkir.
3. Dengan menggunakan sistem ini, maka metode pengolahan lebih efisien dari segi waktu yang hanya membutuhkan rata-rata 15.1 detik dan lebih menghemat 20 detik dari sistem manual maupun sumber daya manusia yang dapat dikurangi pada tempat masuk parkir.

6.2 Saran

Adapun saran-saran yang dapat diberikan untuk pengembangan aplikasi ini berikutnya adalah:

1. Mengingat kemajuan teknologi yang semakin pesat, implementasi aplikasi ini masih jauh dari kesempurnaan untuk meningkatkan keamanan di tempat parkir. Untuk itu diperlukan adanya improvisasi atau penambahan fitur baru untuk mengamankan data yang tersimpan sehingga

sulit untuk diretas oleh pihak yang tidak bertanggung jawab

2. Untuk pengembangan selanjutnya aplikasi ini diharapkan mampu menggunakan alat yang mampu menjangkau kartu dengan jarak yang lebih jauh sehingga bisa diterapkan pada semua jenis kendaraan.
3. Pada *Database* kepemilikan kendaraan dan pengendara yang digunakan untuk kedepannya bisa digunakan secara umum dan bisa digunakan untuk jenis aplikasi yang lain yang mengacu pada database yang sama.

DAFTAR PUSTAKA

- [1] S. Miles, S. Sarma and J. Williams, RFID Technology and Applications, Cambridge University Press, 2008.
- [2] Identive, "IdentiveNFC," [Online]. Available: <http://www.identivenfc.com/en/what-is-nfc>. [Accessed January 2014].
- [3] T. Harris, "How Stuff Works," 2011. [Online]. Available: <http://computer.howstuffworks.com/fingerprint-scanner.htm>. [Accessed January 2014].
- [4] T. B. R. Santoso, Rancang Bangun Sistem Keamanan Parkir menggunakan pasword, Surabaya, 2006.
- [5] Brain Matics, "Brain Matics," [Online]. Available: <http://brainmatics.com/c-net-programming/>. [Accessed March 2014].
- [6] G. Maulani, "Galih Library," [Online]. Available: <http://www.galih-afm.com/2012/08/mengenal-microsoft-visual-c-sharp-c.html>. [Accessed February 2014].
- [7] Microsoft Inc., "MSDN," September 2013. [Online]. Available: <http://msdn.microsoft.com/en-us/library/zw4w595w.aspx>. [Accessed December 2013].
- [8] C. Nagel, B. Evjen, J. Glynn, C. Watson and M. Skinner, Professional C# 2012 and .NET 4.5, Wrox, 2012.

BIODATA PENULIS



Penulis, Fandiasa Koruma Kostrada dipanggil dengan sapaan Fandi atau Asa merupakan anak kedua dari dua bersaudara, lahir pada tanggal 11 Juni 1990 dikota Ponorogo. Penulis menyelesaikan pendidikan formal di SD Muhammadiyah 1 Ponorogo, SMP negeri 1 Ponorogo, SMA Negeri 1 Ponorogo dan pada tahun 2009 penulis diterima di Institut Teknologi Sepuluh Nopember di Jurusan Teknik Informatika. Penulis memulai untuk berorganisasi semasa SMA dengan mengikuti berbagai kegiatan ekstrakurikuler aktif dan kegiatan diluar akademik yaitu bergabung bersama komunitas Trader Indonesia. Setelah lulus SMA, penulis melanjutkan ke jenjang perkuliahan di Jurusan Teknik Informatika Institut Teknologi Sepuluh Nopember Surabaya (2009-2014) dan mengambil bidang minat NCC (Net Centric Computing).

Penulis dapat dihubungi melalui e-mail di asasuke54@gmail.com atau media sosial seperti Facebook dengan nama account Fandiasa Kostrada atau Twitter di @asasuke54.