



**UNIVERSITAS INDONESIA**

**PERANCANGAN DAN IMPLEMENTASI SISTEM PARKIR  
BERBASIS RFID DENGAN MENGGUNAKAN ANTARMUKA  
JAVA DAN BASIS DATA MYSQL UNTUK  
DIIMPLEMENTASIKAN PADA LINGKUNGAN PARKIR  
FTUI**

**SKRIPSI**

**ADYTIAWAN ARGA DWITAMA  
04 05 03 0044**

**FAKULTAS TEKNIK  
PROGRAM STUDI TEKNIK ELEKTRO  
DEPOK  
JULI 2009**



**UNIVERSITAS INDONESIA**

**PERANCANGAN DAN IMPLEMENTASI SISTEM PARKIR  
BERBASIS RFID DENGAN MENGGUNAKAN ANTARMUKA  
JAVA DAN BASIS DATA MYSQL UNTUK  
DIIMPLEMENTASIKAN PADA LINGKUNGAN PARKIR  
FTUI**

**SKRIPSI**

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik**

**ADYTIAWAN ARGA DWITAMA**

**04 05 03 0044**

**FAKULTAS TEKNIK  
PROGRAM STUDI TEKNIK ELEKTRO  
DEPOK  
JULI 2009**

## HALAMAN PERNYATAAN ORISINALITAS

Skripsi ini adalah hasil karya saya sendiri, dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar.

Nama : Adytiawan Arga Dwitama

NPM : 0405030044

Tanda Tangan :

Tanggal : 6 Juli 2009

## HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :  
Nama : Adytiawan Arga Dwitama  
NPM : 0405030044  
Program Studi : Teknik Elektro  
Judul Skripsi : Perancangan dan Implementasi Sistem Parkir Berbasis RFID dengan Menggunakan Antarmuka Java dan Basis Data MySQL Untuk Diimplementasikan pada Lingkungan Parkir FTUI

**Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Elektro, Fakultas Teknik, Universitas Indonesia.**

### DEWAN PENGUJI

Pembimbing : Dr. Ir. Kalamullah Ramli, M.Eng. ( )

Penguji : Dr. Ir. Arman Djohan D, M.T. ( )

Penguji : Ir. Arifin Djauhari, M.T. ( )

Ditetapkan di : Depok

Tanggal : 6 Juli 2009

## UCAPAN TERIMA KASIH

Puji syukur kepada Allah SWT atas segala rahmat dan hidayah-Nya sehingga skripsi ini dapat diselesaikan. Shalawat dan salam semoga senantiasa tercurahkan kepada Nabi Muhammad SAW. Ucapan terima kasih ditujukan kepada:

**Dr. Ir. Kalamullah Ramli, M.Eng**

selaku pembimbing skripsi yang telah meluangkan waktunya untuk memberikan bimbingan, saran, pengarahan dan kemudahan lainnya sehingga skripsi ini dapat diselesaikan dengan baik.

Depok, Juli 2009

Penulis

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS  
AKHIR UNTUK KEPENTINGAN AKADEMIS**

---

---

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : Adytiawan Arga Dwitama  
NPM : 0405030044  
Program Studi : Teknik Elektro  
Departemen : Teknik Elektro  
Fakultas : Teknik  
Jenis Karya : Skripsi

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia Hak Bebas Royalti Noneksklusif (Non-exclusive RoyaltyFree Right) atas karya ilmiah saya yang berjudul :

**PERANCANGAN DAN IMPLEMENTASI SISTEM PARKIR BERBASIS  
RFID DENGAN MENGGUNAKAN ANTARMUKA JAVA DAN BASIS  
DATA MYSQL, UNTUK DIIMPLEMENTASIKAN PADA LINGKUNGAN  
PARKIR FTUI**

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (database), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada Tanggal : 14 Juni 2009

Yang Menyatakan

(Adytiawan Arga Dwitama)

## ABSTRAK

Nama : Adytiawan Arga Dwitama  
Program Studi : Teknik Elektro  
Judul : PERANCANGAN DAN IMPLEMENTASI SISTEM PARKIR BERBASIS RFID DENGAN MENGGUNAKAN ANTARMUKA JAVA DAN BASIS DATA MYSQL, UNTUK DIIMPLEMENTASIKAN PADA LINGKUNGAN PARKIR FTUI

Skripsi ini membahas tentang salah satu aplikasi *Radio Frequency Identification (RFID)* pada sebuah sistem parkir dalam perannya sebagai *automatic identification*. Aplikasi ini juga merupakan pembuktian bahwa *RFID* merupakan salah satu alat penghubung antara benda fisik dengan dunia maya (*internet*). Pada sistem ini, *RFID reader* sebagai terminal pembaca *RFID tag* melakukan komunikasi secara *serial* dengan *PC*, dengan standar *RS-232* dan konektor *DE-9*. Dengan menggunakan *Java API* untuk komunikasi serial, maka *RFID* terhubung dengan *PC*. Selanjutnya dengan *Java Database Connector (JDBC)*, aplikasi berbasis bahasa pemrograman *Java* tersebut dapat terhubung ke sebuah sistem basis data, yang dalam skripsi ini digunakan basis data *MySQL*. Sistem parkir yang dihasilkan diharapkan mampu memberikan keamanan yang lebih baik.

Dari sistem yang telah dibuat, dilakukan tiga jenis uji coba. Uji coba tersebut meliputi pengujian alat, seperti tes pembacaan pada *RFID reader* dan *RFID tag*, dan waktu pengaksesan database, dan dilakukan juga pengujian perangkat lunak *JAVA*, dengan cara *debugging*. Hasil pengujian menunjukkan bahwa sistem ini siap diimplementasikan di lapangan, rendah biaya dan dapat diandalkan.

Kata Kunci:  
Sistem parkir, *Java*, *RFID*, *MySQL*

## ABSTRACT

Name : Adytiawan Arga Dwitama  
Study Program : Teknik Elektro  
Title : DESIGNING AND IMPLEMENTATION OF RFID BASED  
PARKING SYSTEM WITH JAVA INTERFACE AND  
MYSQL DATABASE, TO BE IMPLEMENTED ON FTUI  
PARKING AREA

This final assignment discusses about one of the applications of *Radio Frequency Identification (RFID)* on a parking system in its role as automatic identification. This application shows how *RFID* becomes one of the connecting elements between the physical object and the virtual world (the internet). In this system, *RFID reader* as the reader's terminal of an *RFID tag* performs the communication with serial binary data signals with PC, with *RS-232* as the standard and using *DE-9* as a connector. *Java API* is then used for serial communication, resulting *RFID* connected to the PC. Furthermore, with *Java Database Connector (JDBC)*, that application is connected to MySQL database. Parking system which is using this technology to give better quality at it's security.

Three tests have been performed to the system. The tests includes hardware tests, such as reading test between the *RFID reader* and *RFID tag*, and also the database accessing time, and there is also *JAVA* software test that has been performed by *debugging*. And the results shows that this system is ready to be implemented, low costs, and reliable.

Keywords:  
Parking system, *Java*, *RFID*, *MySQL*

## DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PERNYATAAN ORISINALITAS.....	ii
HALAMAN PENGESAHAN.....	iii
UCAPAN TERIMA KASIH.....	iv
HALAMAN PERSETUJUAN PUBLIKASI SKRIPSI.....	v
ABSTRAK.....	vi
DAFTAR ISI.....	viii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xi
<b>1. PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang Skripsi.....	1
1.2 Perumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Sistematika Penulisan Skripsi.....	3
<b>2. TEORI PENDUKUNG RFID, DATABASE DAN JAVA.....</b>	<b>4</b>
2.1 Radio Frequency Identification (RFID).....	4
2.1.1 Definisi RFID.....	4
2.1.2 Dasar Teknologi RFID.....	4
2.1.2.1 Sistem RFID Aktif.....	6
2.1.2.2 Sistem RFID Pasif.....	7
2.1.2.3 Perbandingan Teknis RFID Aktif dan Pasif.....	9
2.1.2.4 Perbandingan Kemampuan Fungsional RFID.....	9
2.1.2.5 Coupling.....	12
2.1.2.6 Faktor yang Mempengaruhi Kinerja RFID.....	14
2.2 Komunikasi Serial.....	14
2.2.1 RS-232.....	16
2.3 Perangkat Lunak (Java) dan sistem Basis Data (MySQL).....	16
2.3.1 Java.....	16
2.3.1.1 Komunikasi Serial dengan Java API.....	17
2.3.2 MySQL.....	18
<b>3. PEMODELAN SISTEM DAN RENCANA UJI COBA.....</b>	<b>19</b>
3.1 Pemodelan dan Konsep Sistem Parkir Berbasis RFID.....	19
3.2 Bagan Sistem Parkir Berbasis RFID.....	20
3.2.1 Sistem Pengenalan.....	21
3.2.2 Sinkronisasi Data dan Sistem Informasi Parkir.....	23
3.3 Alur Kerja Sistem Parkir.....	24
3.4 Tahapan Implementasi.....	26
3.4.1 Implementasi Perangkat Keras.....	26
3.4.1.1 RFID Reader.....	26

3.4.1.2 RFID Tag .....	26
3.4.1.3 Personal Computer (PC) .....	27
3.4.2 Implementasi Perangkat Lunak .....	27
3.5 Rencana Uji Coba .....	28
3.5.1 Pengujian Perangkat Keras .....	28
3.5.2 Pengujian Perangkat Lunak .....	29
<b>4. IMPLEMENTASI SISTEM PARKIR BERBASIS RFID .....</b>	<b>30</b>
4.1 Implementasi Sistem Parkir .....	30
4.1.1 Implementasi Perangkat Keras .....	30
4.1.1.1 RFID Reader .....	30
4.1.2.1 RFID Tag .....	32
4.2 Uji Coba RFID Reader dan RFID Tag .....	33
4.2.1 Percobaan 1 – Pembacaan Tag C .....	34
4.2.2 Percobaan 2 – Pembacaan Tag B .....	34
4.2.3 Percobaan 3 – Pembacaan Tag A .....	35
4.2.4 Percobaan 4 – Pembacaan Tag A dan Tag B .....	35
4.2.5 Percobaan 5 – Pembacaan Tag B dan Tag C .....	36
4.2.6 Percobaan 6 – Pembacaan Tag A dan Tag C .....	36
4.2.7 Analisa Hasil Percobaan .....	37
4.3 Pemrograman dan Uji Coba Perangkat Lunak .....	37
4.3.1 Koneksi Program dengan Serial Port .....	38
4.3.2 Program sebagai <i>Graphical User Interface</i> (GUI) .....	40
4.3.3 Koneksi Program dengan MySQL Database .....	51
4.3.4 Melakukan Query pada Database .....	51
<b>5. KESIMPULAN .....</b>	<b>53</b>
<b>DAFTAR ACUAN .....</b>	<b>54</b>
<b>DAFTAR PUSTAKA .....</b>	<b>55</b>
<b>LAMPIRAN .....</b>	<b>56</b>

## DAFTAR TABEL

Tabel 2.1	Perbandingan teknis RFID Aktif dan Pasif .....	9
Tabel 2.2	Ringkasan kemampuan fungsional teknologi RFID .....	12
Tabel 4.1	Percobaan Tag C .....	34
Tabel 4.2	Percobaan Tag B .....	34
Tabel 4.3	Percobaan Tag A .....	35
Tabel 4.4	Percobaan Tag A dan Tag B .....	35
Tabel 4.5	Percobaan Tag B dan Tag C .....	36
Tabel 4.6	Percobaan Tag A dan Tag C .....	36

## DAFTAR GAMBAR

Gambar 2.1	Komponen Utama Sistem RFID.....	5
Gambar 2.2	Berbagai Sistem yang Menggunakan RFID.....	5
Gambar 2.3	Rangkaian Dalam Tag Aktif.....	6
Gambar 2.4	Contoh Bentuk RFID Tag Aktif.....	7
Gambar 2.5	Rangkaian Dalam RFID Tag Pasif.....	8
Gambar 2.6	Berbagai Bentuk RFID Tag Pasif.....	8
Gambar 2.7	Skema <i>Inductive Coupling</i> .....	13
Gambar 2.7	Skema <i>Propagation Coupling</i> .....	13
Gambar 2.9	Konfigurasi dan Fungsi Pin RS-232 DE-9 <i>female</i> .....	16
Gambar 2.10	Diagram Alir Sistem Kerja Java.....	17
Gambar 3.1	Denah Parkir.....	19
Gambar 3.2	Perangkat Utama.....	21
Gambar 3.3	Koordinasi Reader dan Tag.....	22
Gambar 3.4	Diagram Alir Sistem Parkir.....	25
Gambar 4.1	RFID Starter Kit.....	31
Gambar 4.2	Bentuk RFID reader yang sudah terbungkus.....	32
Gambar 4.3	RFID tag Pasif dengan ukuran ID-1.....	32
Gambar 4.4	Implementasi Perangkat Keras.....	33
Gambar 4.5	GUI 1, Memindai Serial Port.....	39
Gambar 4.6	Tampilan <i>user interface</i> Program.....	40
Gambar 4.7	Menu <i>Add Category</i> .....	41
Gambar 4.8	Tampilan awal <i>Delete Data Window</i> .....	42
Gambar 4.9	Tampilan lanjut <i>Delete Data Window</i> .....	42
Gambar 4.10	Notifikasi hapus data.....	43
Gambar 4.11	Tampilan Menu Konfigurasi.....	44
Gambar 4.12	Notifikasi Keberhasilan Konfigurasi.....	45
Gambar 4.13	Notifikasi Kesalahan Konfigurasi.....	45
Gambar 4.14	GUI 2, Kendaraan Keluar.....	47
Gambar 4.15	Pesan Kesalahan 1.....	47
Gambar 4.16	Pesan Kesalahan 2.....	48
Gambar 4.17	Pesan Kesalahan 3.....	48
Gambar 4.18	Konfirmasi Penyimpanan Data.....	49
Gambar 4.19	Label Keterangan.....	50
Gambar 4.20	Database Error.....	50

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang Skripsi**

Semakin meningkatnya dan semakin majunya teknologi yang ada saat ini terutama dalam bidang elektronika dan telekomunikasi. Salah satu teknologi yang berkembang pesat penggunaannya adalah Radio-Frequency Identification (Pengidentifikasi Gelombang Frekuensi-Radio) atau yang biasa dikenal sebagai RFID. Penggunaan RFID sudah marak digunakan dalam berbagai bidang, mulai dari olahraga, transportasi, inventarisasi, identifikasi, dan penggunaan yang paling banyak adalah pada bidang keamanan.

Walaupun demikian RFID sama sekali bukan sebuah teknologi baru. Pada kenyataannya, RFID berkembang sebagai sebuah teknologi untuk sistem pertahanan dan keamanan nasional. Dimana pada perang dunia kedua RFID digunakan oleh sekutu untuk melacak pem-bom Jerman. Namun saat ini, penggunaan RFID sudah beragam tergantung masalah yang dihadapi pada suatu lokasi tertentu[1].

Kita sadari atau tidak, RFID adalah sebuah bagian penting dalam hidup kita. RFID meningkatkan produktifitas dan kenyamanan. RFID digunakan untuk ratusan hingga ribuan aplikasi seperti pencegahan pencurian kendaraan bermotor, pembayaran biaya tol tanpa harus berhenti, mengatur lalu lintas, indikator masuknya kendaraan ke suatu gedung, otomasi parkir, mengatur akses kendaraan pada komunitas tertentu, kampus – kampus dan bandara, melacak buku – buku perpustakaan, serta pelacakan aset – aset berharga[2].

Salah satu masalah terpenting adalah meningkatkan keamanan pada system parkir kendaraan bermotor. Semakin meningkatnya tingkat kejahatan kendaraan bermotor di Indonesia. Dan modus kejahatan ini tidak hanya terjadi pada kendaraan bermotor yg terletak di luar lapangan parkir. Namun juga sangat banyak ditemukan kasus pencurian kendaraan bermotor ini di areal parkir.

Untuk itu dibutuhkan suatu solusi rendah biaya untuk mengatasi masalah ini. Salah satunya adalah dengan memanfaatkan teknologi RFID. Pada seminar ini

akan dibahas sebuah system parkir dengan pemanfaatan RFID dan sistem basis data berupa MySQL. Sementara untuk sistem antarmuka-nya digunakan bahasa pemrograman Java.

### **1.2 Perumusan Masalah**

Pada penyambungan dan pembentukkan komunikasi antara peripheral atau alat – alat perangkat luar di luar komputer (PC) dengan PC terdapat banyak pilihan, dimana salah satunya adalah menggunakan komunikasi serial melalui port COM pada PC. Pemilihan cara komunikasi tersebut harus tepat guna agar tidak menimbulkan masalah di kemudian harinya.

Kemudian untuk melakukan pengendalian pada peripheral tersebut, dibutuhkan perangkat lunak tambahan sebagai antarmuka pengguna dengan alat. Walaupun Java menjadi pilihan yang baik karena sifatnya yang mudah dikembangkan, namun API yang tersedia masih sangat terbatas untuk platform OS tertentu, sehingga dibutuhkan API tambahan yang berasal dari sumber lain.

Sedangkan untuk melakukan penyimpanan data – data parkir, dibutuhkan suatu sistem basis data. Basis data ini mau tidak mau harus tergabung dengan perangkat lunak yang juga berperan sebagai pengendali dan antarmuka. Dari sini timbul masalah baru lagi tentang bagaimana menghubungkan program berbasis Java dengan basis data.

### **1.3 Batasan Masalah**

Laporan skripsi yang penulis buat terbatas pada :

1. Sistem dengan peripheral berupa RFID receiver (reader) dan RFID transponder (RFID tag).
2. Komunikasi antara RFID receiver dengan PC hanya menggunakan system komunikasi serial dengan standar arsitektur RS232.
3. Proses pengendalian dan komunikasi antara PC dengan RFID receiver menggunakan bahasa pemrograman JAVA.
4. Penggunaan system basis data MySQL, dengan konfigurator-nya menggunakan PHPMyAdmin XAMPP.

5. Koneksi dengan basis data MySQL menggunakan bahasa pemrograman JAVA.
6. Antarmuka pengguna dengan menggunakan bahasa pemrograman JAVA.

#### **1.4 Sistematika Penulisan Skripsi**

Bab I adalah pendahuluan, yang menjelaskan mengenai latar belakang dibuatnya skripsi ini, perumusan masalah yang akan dihadapi dalam pembuatan skripsi dengan tema yang diambil, tujuan dibuatnya skripsi ini, batasan masalah yang akan dibahas pada skripsi ini, dan sistematika penulisan laporan Skripsi.

Bab II adalah landasan teori yang menjelaskan tentang RFID, definisi dari RFID, dasar – dasar teknologi RFID, komunikasi serial dan penerang lunak yang digunakan, yaitu Java dan MySQL.

Bab III adalah penjelasan tentang skema dan permodelan sistem parkir berbasis RFID, antarmuka Java, dan sistem basis data MySQL. Pada bab ini juga dibahas pemodelan dan konsep sistem parkir, pembagian sistem – sistem yang digunakan, alur kerja sistem, rencana implementasi dan rencana uji coba.

Bab IV adalah penjelasan rangkaian implementasi dan uji coba skripsi. Pada bab ini juga terdapat analisa dan pembahasan atas data – data yang telah di uji cobakan.

Bab V adalah penutup dan kesimpulan dari perancangan dan implementasi skripsi ini.

## **BAB II**

### **TEORI PENDUKUNG RFID, DATABASE DAN JAVA**

#### **2.1 Radio Frequency Identification (RFID)**

##### **2.1.1 Definisi RFID**

RFID adalah sebuah istilah umum yang digunakan untuk menjelaskan sebuah alat yang mentransmisikan identitas (dalam bentuk deretan nomor) dari sebuah objek atau manusia secara nirkabel, menggunakan gelombang radio. RFID dikelompokkan sebagai teknologi *automatic identification* (identifikasi otomatis).

Teknologi identifikasi otomatis melingkupi *barcode*, pembaca karakter optikal dan beberapa teknologi biometric, seperti pemindai retina. Teknologi identifikasi otomatis telah digunakan untuk mengurangi jumlah waktu dan tenaga kerja yang dibutuhkan untuk memasukkan data secara manual dan meningkatkan akurasi data[3].

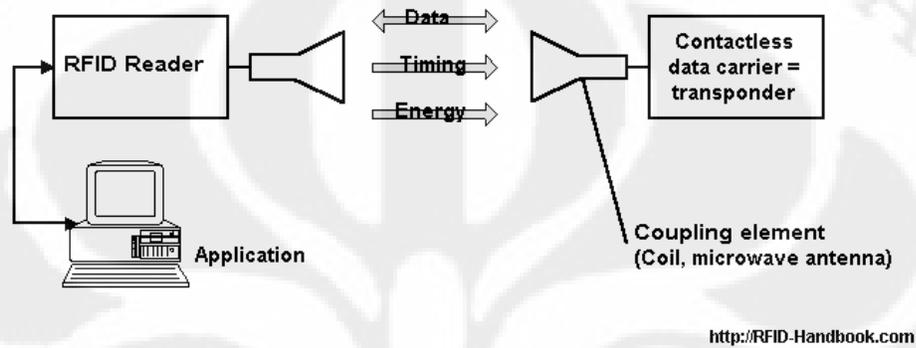
RFID tag terdiri dari sebuah microchip yang tertanam pada sebuah antenna gelombang radio yang ditempelkan pada sebuah substrat (biasanya silikon). Chip tersebut dapat menyimpan data seukuran 2 kilobyte. Sebagai contoh, informasi tentang sebuah barang atau tanggal pengkapalan pada sebuah pabrik, dan sebagainya, dapat dituliskan pada sebuah tag.

Untuk mengambil data yang tersimpan pada RFID tag, dibutuhkan sebuah *reader*. Reader ini adalah sebuah alat yang memiliki satu atau banyak antenna yang memancarkan gelombang radio dan menerima sinyal kembali dari tag. Reader kemudian melewatkan informasi tersebut dalam bentuk digital ke system computer.

##### **2.1.2 Dasar Teknologi RFID**

RFID merupakan sebuah gelombang perubahan baru dalam dunia computer. Secara esensial, RFID adalah teknologi yang mampu menghubungkan sebuah objek ke internet, sehingga objek – objek tersebut dapat dilacak, dan pihak – pihak yang berkaitan dapat berbagi data tentang objek – objek tersebut. Konsepnya sangat sederhana: Letakkan sebuah pemancar – sebuah microchip

dengan sebuah antenna – pada sebuah benda lalu kemudian gunakan *reader* – sebuah alat dengan satu atau lebih antenna – untuk membaca data dari microchip tersebut dengan menggunakan gelombang radio. Reader tersebut kemudian mengirimkan informasi ke computer, sehingga data tersebut dapat digunakan lebih lanjut[4].



Gambar 2.1 – Komponen utama sistem RFID

Ada banyak jenis sistem RFID yang berbeda, dan RFID ini dapat digunakan untuk melakukan penghematan atau meningkatkan efisiensi. Sangat penting untuk memilih jenis system RFID yang tepat untuk beberapa aplikasi tertentu.



Gambar 2.2 – Berbagai sistem yang menggunakan RFID

Hampir semua RFID tag atau transponder menggunakan microchip berbasis silicon untuk menyimpan nomor seri yang unik dan biasanya beberapa informasi tambahan. Ada dua kategori utama dari sistem RFID – system aktif dan pasif. RFID tag pasif tidak memiliki *transmitter*; mereka hanya memantulkan kembali energi (gelombang radio) yang datang dari antena reader. Tag aktif memiliki transmitter sendiri dan sumber daya, biasanya – tapi tidak selalu – sebuah baterai (tag aktif bisa pula mengambil energi dari matahari atau sumber lain). Tag-tag tersebut memancarkan sinyal untuk mengirimkan informasi yang tersimpan dalam microchip. (Ada juga tag semi-pasif yang cocok untuk aplikasi tertentu).

#### 2.1.2.1 Sistem RFID Aktif

Tag aktif digunakan untuk aset – aset besar, seperti muatan kargo yang membutuhkan pelacakan jarak jauh. Tag – tag tersebut biasanya beroperasi pada 455 MHz, 2.45GHz, atau 5.8GHz, dan memiliki jarak pembacaan mulai dari 60 sampai 300 kaki (20 sampai 100 meter).



Gambar 2.3 – Rangkaian dalam Tag Aktif

Lebih jauh lagi, ada dua jenis tag aktif: transponder dan beacon. Transponder aktif terbangun ketika menerima sinyal dari reader. Tag seperti ini digunakan pada pintu tol (pembayaran tarif tol), kontrol titik cek dan sistem lainnya. Ketika sebuah mobil dengan transponder aktif mendekati pintu tol, reader pada pintu tol akan mengirimkan sinyal yang membangkitkan transponder pada mobil. Kemudian transponder itu memancarkan ID uniknya ke reader. Transponder ini mengawetkan waktu hidup baterai karena tag hanya memancarkan sinyal ketika dalam jangkauan readernya.

Beacon kebanyakan digunakan pada sistem pelokasian waktu-nyata (real-time locating systems – RTLS), dimana lokasi sebuah aset yang dipantau harus

**Universitas Indonesia**

diketahui secara tepat. Pada sebuah RTLS, beacon memancarkan sinyal dengan ID uniknya pada interval yang telah diatur sebelumnya (setiap tiga detik sekali, sekali sehari, tergantung seberapa pentingnya mengetahui lokasi aset tersebut pada waktu – waktu tertentu). Sinyal beacon ditangkap paling sedikit oleh 3 antena radar yang diletakan sekitar garis keliling daerah dimana aset tersebut dilacak. RTLS biasanya terdapat di luar ruangan, namun dapat juga digunakan pada pabrik berukuran luas.

Tag aktif memiliki jangkauan pembacaan hingga 300 kaki (100 meter) dan bisa dibaca dengan akurat karena tag – tag tersebut memancarkan sinyal ke reader (beberapa sistem bahkan dapat dipengaruhi oleh hujan). Tag – tag ini memiliki harga yang bervariasi dan relatif mahal, tergantung dari jumlah memori, waktu hidup baterai yang dibutuhkan, apakah tag tersebut memiliki sensor tambahan, dan tingkat kekerasan. Semakin tebal, dan tahan lamanya plastik pembungkus akan meningkatkan harganya.



Gambar 2.4 – Contoh bentuk RFID Tag Aktif

#### 2.1.2.2 Sistem RFID Pasif

RFID tag pasif tidak mempunyai sumber daya dan tidak memiliki pemancar. Harganya lebih murah dari tag aktif dan tidak membutuhkan perawatan berkala, karena alasan inilah perusahaan retail dan pabrik kecil menggunakan tag pasif. Tag pasif memiliki jangkauan baca yang lebih pendek dibandingkan dengan tag aktif (beberapa cm hingga 1 meter)



Gambar 2.5 – Rangkaian dalam RFID Tag Pasif

Transponder RFID pasif memiliki microchip yang tertanam pada sebuah antena. Transponder dapat dikemas dalam berbagai bentuk. Dapat di tempelkan pada sebuah substrat untuk membuat tag, atau diselipkan di antara lapisan adesif dan kertas label untuk membuat label RFID yang dapat dicetak, atau smart label. Transponder dapat pula ditanamkan pada kartu plastik, kunci, dinding container plastik, dan pengemasan khusus tahan panas, dingin atau zat kimia yang merusak. Faktor bentuk yang digunakan ini tergantung pada aplikasinya, namun pengemasan transponder meningkatkan biaya secara signifikan.



Gambar 2.6 – Berbagai bentuk RFID tag pasif

Tag pasif dapat beroperasi pada frekuensi rendah, tinggi, dan UHF. Sistem frekuensi rendah biasanya beroperasi pada 124 kHz, 125kHz atau 135 kHz. Sistem frekuensi tinggi menggunakan 13.56 MHz, dan sistem UHF menggunakan pita frekuensi antara 860 MHz hingga 960 MHz. beberapa sistem juga menggunakan 2.45 GHz dan daerah lain spectrum radio.

Gelombang radio memiliki sifat yang berbeda pada setiap frekuensi tersebut, yang artinya frekuensi yang berbeda tersebut cocok pada aplikasi –

aplikasi tertentu yang berbeda pula. Semakin rendah frekuensinya, maka semakin mudah gelombang tersebut untuk menembus benda – benda tebal, namun jarak transmisinya pun semakin pendek. Sebaliknya, semakin tinggi frekuensi gelombang yang digunakan, maka semakin jauh jarak pancarnya, namun daya tembusnya rendah. Tag berfrekuensi rendah idealnya digunakan untuk aplikasi – aplikasi dimana tag harus dapat dibaca menembus material atau air pada jarak yang dekat.

### 2.1.2.3 Perbandingan Teknis RFID Aktif dan Pasif

Untuk lebih jelasnya, Tabel 2.1 menggambarkan perbandingan teknis antara RFID tag aktif dan pasif.

Tabel 2.1 – Perbandingan teknis RFID Aktif dan Pasif

	<b>RFID Aktif</b>	<b>RFID Pasif</b>
<b>Sumber Daya Tag</b>	Internal di dalam Tag	Energi ditransfer dari Reader ke Tag
<b>Baterai Tag</b>	Ada	Tidak ada
<b>Ketersediaan Daya pada Tag</b>	Kontinyu	Hanya bila Tag berada pada medan Reader
<b>Kuat Sinyal yang dibutuhkan dari Reader ke Tag</b>	Rendah	Tinggi (Karena harus memberikan daya pada Tag)
<b>Kuat Sinyal yang dihasilkan dari Tag ke Reader</b>	Tinggi	Rendah

### 2.1.2.4 Perbandingan Kemampuan Fungsional RFID Aktif dan Pasif

Karena beberapa perbedaan teknis seperti yang telah disebutkan di atas, kemampuan fungsional dari RFID aktif dan pasif sangat berbeda dan harus diperhatikan ketika memilih teknologi yang akan digunakan untuk aplikasi yang spesifik.

#### 1. Jarak Komunikasi

Untuk RFID pasif, jarak komunikasi terbatas oleh dua faktor:

- a) Kebutuhan akan kuatnya sinyal yang harus diterima oleh tag untuk mencatu tag, membatasi jarak antara reader dengan tag, dan
- b) Jumlah daya yang tersedia pada tag terlalu kecil untuk merespon reader, membatasi jarak antara reader dan tag.

Faktor – faktor ini memaksa RFID pasif untuk ber-operasi pada jarak 3 meter atau kurang terhadap reader. Tergantung dari perusahaan pembuatnya dan frekuensi kerjanya, jarak tersebut bisa saja hanya beberapa centimeter. RFID aktif tidak terikat oleh dayanya dan dapat melakukan komunikasi hingga jarak 100 meter atau lebih.

## 2. Pengumpulan data Banyak-Tag

Sebagai akibat langsung dari terbatasnya jarak komunikasi pada RFID pasif, mengumpulkan data beberapa tag yang terkumpul dan beroperasi secara dinamis sangat sulit dan seringkali tidak dapat diandalkan. Sebagai contoh dari scenario ini adalah sebuah *forklift* membawa paket barang dengan beberapa benda yang dipasangkan tag melewati sebuah pintu dermaga. Mengidentifikasi beberapa tag sekaligus membutuhkan sejumlah banyak komunikasi antara reader dan tag, atau lebih tepatnya sebuah proses beberapa-langkah dimana reader berkomunikasi secara individual dengan setiap tag. Setiap interaksi tersebut membutuhkan waktu, dan potensi terjadinya interferensi bertambah berbanding lurus dengan jumlah tag, lebih jauh lagi membuat durasi operasi bertambah. Karena seluruh operasi pengumpulan data harus selesai pada saat tag masih di dalam jarak jangkauan reader, RFID pasif tidak memenuhi aspek ini. Sebagai contoh, sebuah sistem RFID pasif yang tersedia saat ini membutuhkan waktu lebih dari 3 detik untuk mengidentifikasi 20 tag. Dengan jangkauan komunikasi 3 meter, hal ini membatasi kecepatan dari benda yang tertempel tag tersebut menjadi kurang dari 5 km/jam. RFID aktif, dengan jarak operasi 100 meter lebih, mampu mengumpulkan ribuan tag dari reader tunggal. Sebagai tambahan, tag dapat dibaca dalam kecepatan lebih dari 140 kps dan hasil pembacaannya masih akurat dan dapat diandalkan.

Universitas Indonesia

### 3. Kemampuan Penginderaan

Satu bidang fungsi yang memiliki relevansi besar pada banyak aplikasi *supply chain* adalah kemampuan untuk memantau keadaan lingkungan atau status parameter menggunakan sebuah RFID tag dengan kemampuan penginderaan tertanam. Parameter – parameter yang dimaksud dapat mencakup temperature, kelembaban, guncangan, keamanan, dan deteksi pengrusakan. Karena RFID tag pasif hanya berdaya ketika berada pada jarak dekat dengan rader, tag ini tidak dapat memantau status sensor secara kontinyu. Di samping itu, tag jenis ini terbatas dalam hal melakukan reporting status terbaru hanya ketika mencapai reader. RFID tag aktif secara konstan berdaya, baik ketika berada di jangkauan reader maupun tidak, sehingga dapat secara kontinyu memantau dan menyimpan status sensor, sangat penting pada pengukuran batas temperature dan status segel container. Sebagai tambahan, RFID aktif dapat memberikan daya *clock real-time* internal dan menerapkan stempel tanggal/waktu untuk setiap nilai sensor yang terekam atau suatu kejadian.

### 4. Penyimpanan Data

Baik teknologi RFID aktif dan pasif dapat menyimpan data secara dinamis pada tag. Walaupun demikian, karena keterbatasan daya, RFID pasif biasanya hanya dapat menyediakan ruang penyimpanan data yang kecil, kurang dari 128 byte (1000 bit), tanpa adanya kemampuan pencarian data atau fitur manipulasi data lainnya. Ruang penyimpanan data yang lebih besar dan kemampuan akses data yang canggih membutuhkan tag yang berdaya dalam jangka waktu lama. RFID aktif memiliki fleksibilitas agar tetap berdaya untuk melakukan akses dan pencarian pada ruang data yang lebih besar, demikian juga kemampuannya untuk mentransmisikan paket data yang lebih panjang. RFID tag aktif biasanya memiliki ruang penyimpanan data baca/tulis dinamis dan dapat dicari sebesar 128 Kbytes (1Mb).

Tabel 2.2 – Ringkasan kemampuan fungsional teknologi RFID aktif dan pasif

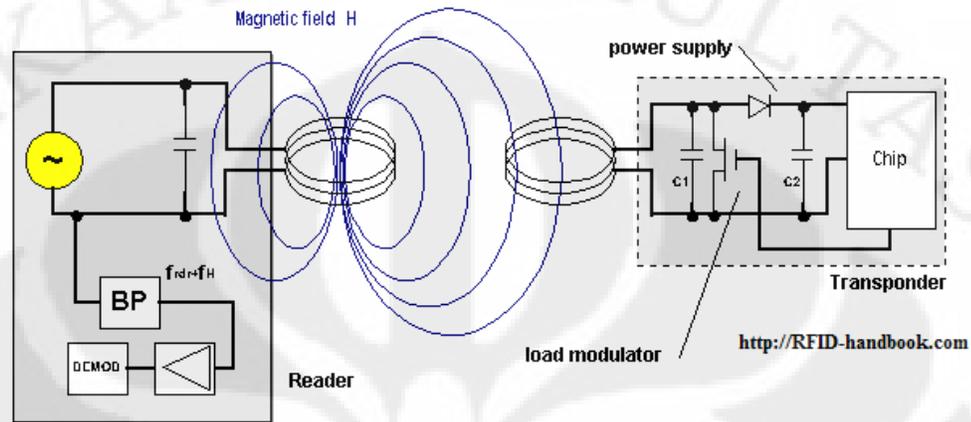
	<b>RFID Aktif</b>	<b>RFID Pasif</b>
<b>Jangkauan Komunikasi</b>	Jangkauan jauh (lebih dari 100 meter)	Pendek ato sangat pendek (kurang dari 3 meter)
<b>Pengumpulan Banyak-Tag</b>	<ul style="list-style-type: none"> <li>• Mengumpulkan 1000 tag pada area 7 are dengan reader tunggal</li> <li>• Mengumpulkan 20 tag yang bergerak dengan kecepatan lebih dari 140kpj</li> </ul>	<ul style="list-style-type: none"> <li>• Mengumpulkan ratusan tag pada jarak 3 meter dengan reader tunggal</li> <li>• Mengumpulkan 20 tag yang bergerak dengan kecepatan kurang dari 5 kpj</li> </ul>
<b>Kemampuan Penginderaan</b>	Mampu secara kontinyu memantau dan merekam input sensor; stempel waktu/tanggal untuk kejadian sensor	Mampu membaca dan mentransfer nilai sensor hanya ketika tag dicatu oleh reader; tanpa stempel waktu/tanggal
<b>Penyimpanan Data</b>	Penyimpanan data baca/tulis yang besar (128KB) dengan kemampuan pencarian dan akses data canggih	Penyimpanan data baca/tulis yang kecil (128 bytes)

#### 2.1.2.5 Coupling

Jarak pembacaan RFID ditentukan oleh banyak faktor, namun satu yang paling penting adalah metode tag pasif dalam mengirimkan data ke reader. Tag dengan frekuensi rendah dan dinggi menggunakan inductive coupling.

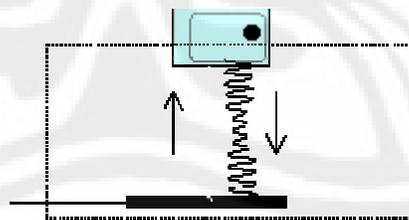
Secara teknis, yang dimaksud dengan inductive coupling adalah perpindahan energi dari satu rangkaian ke rangkaian lain sebagai hasil dari induktansi mutual antara dua lilitan. Pada inductive coupling ini, pada tag dan reader masing – masing mempunyai lilitan. Intinya, lilitan pada antena reader dan tag membentuk sebuah medan elektromagnetik. Tag mendapatkan daya dari medan tersebut, yang kemudian menggunakan daya tersebut untuk menjalankan rangkaian pada chip dan kemudian merubah beban listrik pada antena. Antena reader merasakan perubahan medan magnet ini dan mengkonversi perubahan ini menjadi nilai digital yang dapat dimengerti oleh komputer. Karena lilitan pada

antena tag dan lilitan pada antenna reader harus membentuk medan magnet, tag tersebut harus berada relatif dekat dengan antenna reader, dimana sistem seperti ini terbatas dalam hal jarak pembacaan.



Gambar 2.7 – Skema *Inductive Coupling*

System pasif UHF menggunakan propagation coupling. Antena reader memancarkan energi elektromagnetik (gelombang radio). Tidak ada medan elektromagnetik yang terbentuk. Tag mengumpulkan energi dari antena reader, dan microchip menggunakan energi tersebut untuk mengubah beban pada antena dan memantulkan sinyal yang berubah – ubah. Hal ini disebut backscatter.



Gambar 2.8 – Skema *Propagation Coupling*

Tag UHF dapat berkomunikasi satu – nol dalam tiga cara berbeda. Tag tersebut dapat meningkatkan amplitudo gelombang pantul (ASK), menggeser gelombang hingga keluar dari fasanya (PSK) atau merubah frekuensinya (FSK). Reader menangkap sinyal dan merubah gelombang tersebut menjadi satu dan nol.

Informasi tersebut kemudian diberikan ke komputer yang mengubah data biner menjadi deretan angka (ID) atau data yang tersimpan dalam tag.

#### 2.1.2.6 Faktor yang mempengaruhi kinerja RFID

Ada beberapa faktor yang mampu mempengaruhi kinerja dari RFID. Faktor – faktor ini mempunyai pengaruh yang berbeda tergantung spesifikasi dari RFID yang digunakan. Berikut adalah faktor – faktor tersebut:

1. Frekuensi kerja RFID

Karena sistem yang menggunakan frekuensi rendah dan tinggi menggunakan inductive coupling, ukuran dari reader lebih kecil dan dapat lebih mudah dikontrol. Sistem UHF yang menggunakan propagation coupling lebih sulit dikontrol, karena energi dikirimkan melalui jarak yang jauh. Gelombang dapat memantul dan mencapai tag – tag yang tidak tepat.

2. Media perantara transmisi data

Sistem dengan frekuensi rendah dan tinggi juga bekerja lebih baik daripada sistem UHF di sekitar logam dan air. Gelombang radio tidak memantul pada logam dan menyebabkan kesalahan pembacaan. Dan gelombang tersebut juga lebih baik dalam menembus air, sementara gelombang radio UHF dapat diserap oleh air.

## 2.2 Komunikasi Serial

Komunikasi serial adalah suatu metode komunikasi dengan transmisi data per bit dalam satu waktu melalui sebuah jalur transmisi, kabel atau pun wireless. Komunikasi serial digunakan untuk komunikasi jarak jauh dan kebanyakan dari jaringan komputer, dimana harga kabel dan proses sinkronisasi data menjadi pertimbangan utama. Karakteristik penting pada komunikasi serial, adalah sebagai berikut.

- a. *Baud rate*

Adalah sistem perhitungan untuk komunikasi. *Baud rate* mengindikasikan berapa bit data yang dikirimkan setiap detik, dan pada

*clock cycle*, *baud rate* menunjukkan frekuensi yang digunakan oleh *clock* tersebut.

b. *Data bits*

Adalah perhitungan jumlah dari data bit yang sedang ditransmisikan. Pengiriman data ini pada standarnya adalah 5,7 atau 8 bit, tergantung dari data yang akan ditransmisikan.

c. *Stop bit*

Digunakan untuk mengakhiri komunikasi untuk satu paket, dan juga digunakan untuk menangani error pada clock speed.

d. *Parity*

Adalah bit tambahan yang akan mendeteksi adanya kesalahan pada komunikasi serial. Bit tambahan ini terletak di akhir data yang ditransmisikan. Parity ada beberapa jenis, yaitu even, odd, MARK, dan SPACE parity. Pada even parity, bit parity akan bernilai '1' jika bit '0' pada data yang ditransmisikan berjumlah genap. Pada odd parity, bit parity akan bernilai '1' jika bit '0' pada data yang ditransmisikan berjumlah ganjil. Pada MARK parity, bit parity akan selalu bernilai '1'. Sedangkan pada SPACE parity, bit parity akan selalu bernilai '0'.

Berdasarkan *clock*-nya terdapat 2 macam cara transmisi data serial, yaitu:

a. Transmisi data serial secara sinkron

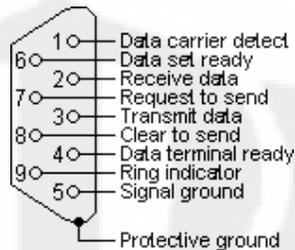
Pada sistem ini, clock dikirim bersamaan dengan data serial, dan komunikasi berlangsung secara half-duplex, dimana pengiriman dan penerimaan data dapat berlangsung dua arah tetapi harus saling bergantian.

b. Transmisi data serial secara asinkron

Pada sistem ini, clock tidak dikirim bersama data serial. Rangkaian penerima data harus membangkitkan sendiri clock pendorong data serial atau harus memiliki baudrate generator. Sistem komunikasi berlangsung secara full-duplex, dimana penerimaan dan pengiriman data dapat berlangsung secara bersamaan.

### 2.2.1 RS-232

RS-232 (Reccomended Standard 232) merupakan standar komunikasi sinyal dengan data serial biner. Saat ini konektor yang umum digunakan untuk standar ini adalah DE-9, dengan konfigurasi pin sebagai berikut.



Gambar 2.8 – Konfigurasi dan fungsi pin RS232 DE-9 *female*

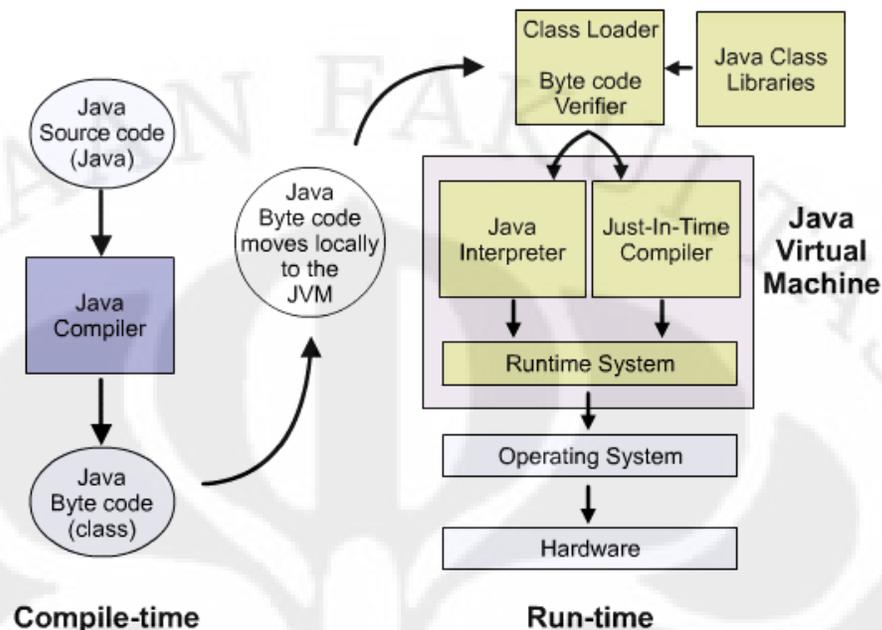
Pada komunikasi serial dengan *port* serial RS232, terdapat 2 pin yang memegang peranan penting dalam aliran informasi data, yaitu TxD dan RxD. TxD adalah pin *Transmit Data* sedangkan RxD adalah pin *Received Data*.

## 2.3 Perangkat Lunak (Java) dan sistem Basis Data (MySQL)

Pada sistem ini, digunakan dua perangkat lunak khusus. Yang pertama adalah perangkat lunak sebagai antar muka pengguna dengan sistem parkir keseluruhan yang dikembangkan dalam bahasa pemrograman Java. Kemudian untuk melakukan penyimpanan dan pengolahan data parkir digunakan sistem Basis Data menggunakan MySQL Database.

### 2.3.1 Java

Java adalah sebuah bahasa pemrograman yang dikembangkan oleh James Gosling, programmer pada Sun Microsystems dan pertama kali diluncurkan pada 1995. Syntax Java banyak diturunkan dari bahasa C dan C++. Aplikasi java dikompilasi dalam bentuk *bytecode* yang dapat dijalankan pada setiap *Java Virtual Machine* (JVM) bagaimanapun arsitektur komputernya[5]. Secara garis besar, sistem kerja dari program berbasis Java dapat dilihat pada Gambar 2.9.



Gambar 2.9 – Diagram alir sistem kerja Java

Program – program java ditulis sebagai *text file* dengan ekstensi ‘\*.java’. Kemudian *Java Compiler* membuat *program file* yang mengandung ‘ByteCode’. File byte-code ini kemudian dikirim ke JVM (pada komputer). JVM kemudian menginterpretasikan byte-code tersebut, satu instruksi sekali waktu. Oleh karena itu untuk setiap OS dan CPU dibutuhkan sebuah JVM untuk menjalankan program – program berbasis Java.

### 2.3.1.1 Komunikasi Serial dengan Java API

Seperti bahasa pemrograman lainnya, Java juga memiliki berbagai rutin API (Application Programming Interface). API merupakan kumpulan rutin – rutin, kelompok data – data, class – class, dan/atau protokol – protokol yang disediakan oleh library pada suatu bahasa pemrograman tertentu untuk mendukung pembuatan sebuah aplikasi.

Komunikasi antara perangkat lunak dengan sistem perangkat keras seperti RFID reader adalah menggunakan komunikasi serial. Yaitu menghubungkan RFID reader tersebut dengan port COM pada PC. Pada bahasa pemrograman Java dengan Java Development Kit (JDK) versi 1.4 telah disediakan

Universitas Indonesia

API khusus untuk melakukan komunikasi serial pada semua OS. Akan tetapi tidak demikian halnya dengan JDK versi 1.6 ke atas. Karena Sun Microsystems tidak lagi menyediakan API komunikasi serial untuk platform Windows.

Namun hal tersebut dapat diatasi, karena Java merupakan bahasa pemrograman tak berbayar, dan pengembangannya tersebar luas, oleh karena itu untuk melakukan komunikasi serial ini dapat digunakan sebuah library khusus yg telah disediakan oleh GiovyNet. Dengan menggunakan GiovyNet 1.3, maka proses komunikasi data melalui port COM pun dapat dilakukan, lengkap dengan parameter – parameter yang dibutuhkan, seperti pemilihan port, set baud rate, dan lainnya.

### 2.3.2 MySQL

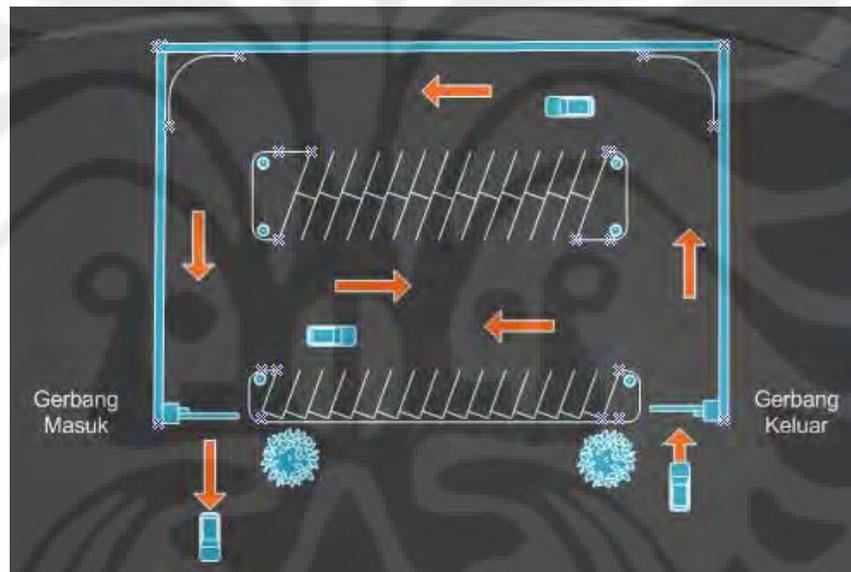
MySQL adalah sebuah *Relational Database Management System* (RDBMS) yang paling banyak digunakan di dunia. MySQL ini dijalankan pada sebuah server dan memberikan akses multi-user ke sejumlah basis data. Seperti namanya, MySQL menggunakan *Structured Query Language* (SQL), yaitu sebuah bahasa basis data komputer yang dirancang untuk mengambil dan melakukan pengaturan data dalam RDBMS, modifikasi dan pembuatan skema basis data.

### BAB III

## PEMODELAN SISTEM PARKIR BERBASIS RFID DENGAN ANTARMUKA JAVA DAN BASIS DATA MYSQL

### 3.1 Pemodelan dan Konsep Sistem Parkir Berbasis RFID

Sistem parkir di suatu kampus membutuhkan suatu jaminan keamanan bagi pengendara atau pemilik kendaraan ketika mereka meletakkan kendaraannya di tempat parkir. Dimana keamanan tersebut juga tidak mengurangi tingkat kenyamanan dari si pengendara, dan tentu saja sistem yang digunakan harus rendah biaya. Asumsikan sebuah lapangan parkir dengan denah seperti Gambar 3.1



Gambar 3.1 – Denah Parkir

Secara umum, lapangan parkir yang berada pada suatu kampus terdiri dari satu pintu gerbang masuk, dan satu pintu gerbang keluar. Dimana di setiap gerbang, idealnya baagi keamanan terdapat satuan pengamanan, dan dengan prosedur yang agak berbelit – belit, seperti pengecekan STNK. Sedangkan biaya parkir tidak terstruktur tarifnya.

Untuk itu dibutuhkan sebuah sistem parkir yang:

1. Mampu memenuhi standar keamanan bagi penggunanya
2. Memiliki prosedur yang mudah dan tidak mengganggu kenyamanan pengguna
3. Biaya implementasi yang tidak terlalu tinggi, namun tetap *high-tech* dan dapat diandalkan
4. Dapat bertahan lama dan dapat diaplikasikan pada bidang yang lebih luas
5. Turut mencerdaskan masyarakat, dalam hal ini pengguna lahan parkir, akan teknologi baru seperti RFID
6. Kenyamanan bagi operator areal parkir dan mempermudah pekerjaan petugas keamanan

Sistem parkir berbasis RFID ini dirancang untuk mengatasi berbagai masalah yang ada pada sistem parkir konvensional. Dimana pada setiap gerbang (gerbang masuk dan gerbang keluar), akan disediakan sebuah RFID reader dan PC untuk mengorganisir proses keluar masuk kendaraan. Untuk mencapai sistem yang rendah biaya dan dinamis, pada gerbang masuk lokasi parkir disediakan sejumlah RFID tag, tentunya dengan ID yang unik.

Secara sederhana, pengendara yang ingin memarkirkan kendaraannya memasuki areal parkir, kemudian pengendara tersebut akan diberikan RFID tag yang telah direcord dalam database. Kemudian apabila si pengendara tersebut akan keluar dari lahan parkir, maka ia pun wajib menunjukkan kembali RFID tag yang sama persis dengan yang ia dapatkan ketika masuk. Dengan sistem seperti ini, maka akan sangat sulit untuk memalsukan RFID tag yang bersesuaian dengan kendaraan yang bersangkutan.

### **3.2 Bagan Sistem Parkir Berbasis RFID**

Perancangan sistem parkir berbasis RFID ini menghasilkan dua bagian penting. Bagian atau sistem yang pertama adalah sistem pengenalan pengendara dan kendaraannya (Embedded System). Kemudian bagian yang kedua adalah proses sinkronisasi data pengendara pada database dan pentarifan (Sistem Informasi).

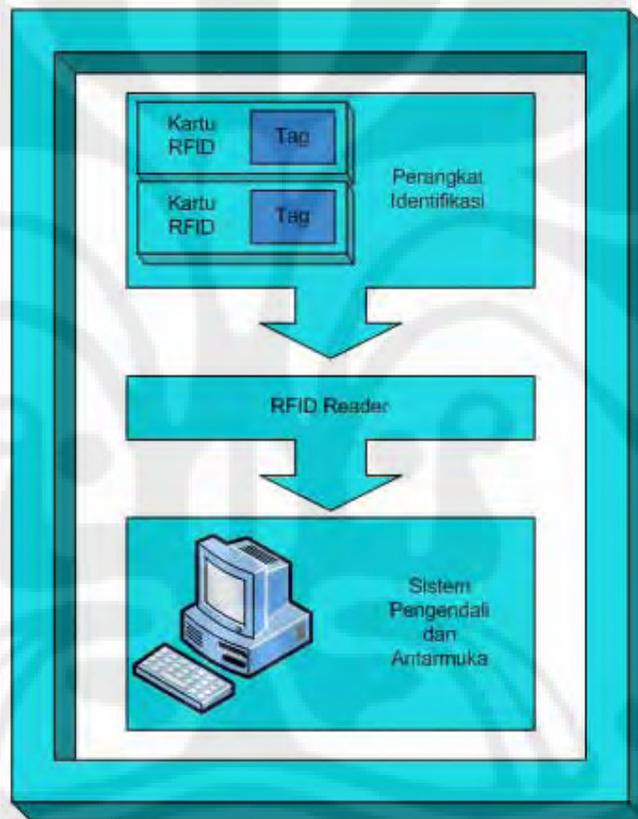
Universitas Indonesia

### 3.2.1 Sistem Pengenalan

Sistem autentikasi ini terbagi menjadi dua bagian, yaitu sistem input data pada gerbang masuk dan sistem autentikasi kendaraan pada gerbang keluarnya.

Pada sistem pengenalan ini, terdiri dari 3 bagian utama, yaitu :

1. RFID reader sebagai pembaca dan pengenalan RFID tag
2. RFID tag sebagai sarana pengenalan
3. PC sebagai pengendali dan antar muka



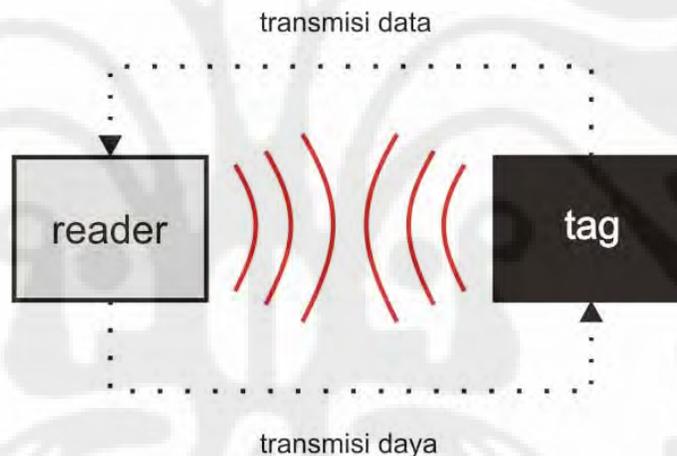
Gambar 3.2 – Perangkat Utama Sistem Pengenalan

Gambar 3.2 menunjukkan berbagai perangkat atau bagian yang digunakan pada sistem pertama yaitu sistem pengenalan beserta hubungan masing – masing perangkat tersebut dengan yang lainnya. Dimana beberapa RFID tag yang akan dibaca ID uniknya, kemudian data tersebut dikirimkan ke sistem pengendali yang merupakan personal computer (PC).

Sistem parkir ini menggunakan reader dengan jarak baca sekitar 12 cm yang akan mendeteksi RFID tag yang didekatkan ke reader tersebut oleh operator parkir. Dalam permodelan sistem parkir ini, konfigurasi dari sistem RFID ini di tentukan sebagai berikut.

1. Menggunakan tag pasif.
2. Frekuensi kerja RFID yang digunakan adalah 125 KHz.

RFID tag pasif digunakan pada sistem ini karena harganya yang ekonomis, bentuknya yang kecil dan mudah dipaketkan dalam sebuah kartu. Selain itu, tag pasif memiliki usia guna yang relatif lebih lama dan sangat cocok untuk aplikasi ini. Karena tag pasif hanya aktif ketika ada transmisi daya (pemicu) dari reader. Gambar 3.3 menunjukkan koordinasi antara RFID reader dengan tag pasif.



Gambar 3.3 – Koordinasi Reader dan Tag

Di dalam PC untuk mempermudah pengendalian, digunakan sebuah software untuk antarmuka antara operator dengan alat – alat tersebut. Sistem ini seperti telah disebutkan di atas, terletak di setiap gerbang (masuk dan keluar) seperti terlihat pada Gambar 3.3.

RFID reader dan basis data pada sistem ini dihubungkan oleh PC, yang merupakan sistem pengontrol, dengan menggunakan jenis komunikasi binari serial. Standar yang digunakan pada komunikasi binari serial ini adalah Recommended Standard 232 (RS232) dengan memakai konektor DE-9.

### 3.2.2 Sinkronisasi Data dan Sistem Informasi Parkir

Bagian kedua dari sistem parkir berbasis RFID ini adalah berupa sinkronisasi dan sistem informasi dari data – data parkir itu sendiri. Bagian ini terdiri dari berbagai komponen pendukung, sebagai berikut:

1. Sistem antar muka yang terdiri dari PC dengan program berbasis Java.
2. Sistem basis data penyimpan data – data pengguna parkir
3. Sistem basis data penyimpan data parkir

Sistem antar muka dengan PC tersebut digunakan untuk melakukan input data parkir dan untuk menampilkan data kendaraan yang tersimpan dalam basis data ketika kendaraan tersebut keluar dari areal parkir.

Basis data yang pertama adalah basis data yang berfungsi untuk menyimpan nomor kendaraan, dan kategori pemilik kendaraan tersebut. Basis data ini merupakan basis data yang bersifat semi-statis, yang artinya data yang digunakan diharapkan tidak berubah pada saat penggunaan sistem ini. Ketika transaksi atau proses keluar masuk kendaraan dilakukan, basis data ini tidak akan berubah karena pengaruh kejadian tersebut. Karena sistem harus melakukan pengecekan terhadap kategori kendaraan keluar – masuk areal parkir terhadap basis data ini. Katakanlah basis data kategori.

Sistem basis data yang kedua adalah sistem yang dinamis. Basis data ini menyimpan data – data kendaraan yang masuk, serta perhitungan statistik dari areal parkir tersebut. Oleh karena itulah data pada basis data ini bersifat dinamis, karena sifatnya yang selalu berubah bila ada kendaraan yang masuk atau keluar. Katakanlah basis data ini adalah basis data parkir.

Sebelum kendaraan masuk ke area parkir, operator akan melakukan input data ke sistem informasi, dengan cara melihat nomor kendaraan pengguna areal parkir tersebut. Kemudian melakukan pencatatan nomor ID dari RFID tag tertentu dengan menggunakan sebuah program yang sudah dirancang khusus. Kedua data tersebut kemudian disimpan ke dalam basis data parkir, beserta waktu masuknya yang secara otomatis tersimpan oleh program.

Kemudian pada saat kendaraan tersebut keluar, kartu RFID tag yang dibawa oleh pengguna areal parkir tersebut kembali dibaca oleh reader dan program. Kemudian program tersebut memunculkan data-data dari pengguna areal

**Universitas Indonesia**

parkir tersebut, sementara operator melakukan pencocokan antara data-data dengan kendaraan tersebut.

Program ini juga sekaligus melakukan pentarifan terhadap kendaraan tersebut. Pentarifan diatur berdasarkan kategori pemilik kendaraan tersebut. Kemudian tarif dihitung berdasarkan lama parkir kendaraan tersebut.

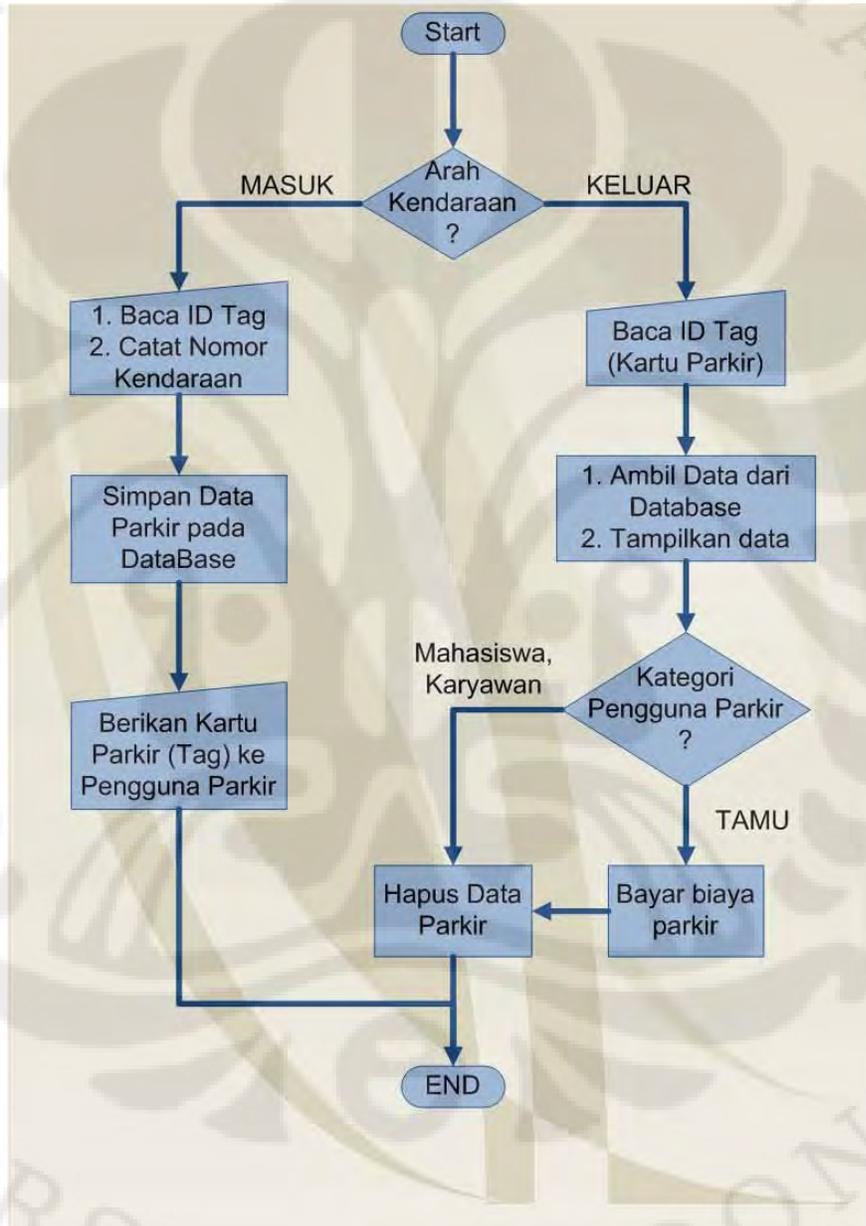
### 3.3 Alur Kerja Sistem Parkir

Sistem parkir dan pentarifannya yang berbasis RFID dan basis data ini memiliki tahapan kerja sebagai berikut:

1. Ketika ada calon pengguna lahan parkir yang datang, operator PC di gerbang masuk mendekati 1 buah RFID tag ke reader, agar ID uniknya dapat terbaca, dan kemudian dikirimkan ke PC dengan komunikasi binary serial. ID unik tersebut ditampilkan pada perangkat lunak yang berperan sebagai antar – muka.
2. Operator mencatat nomor kendaraan calon pengguna lahan parkir tersebut pada perangkat lunak, dan kemudian melakukan input ke basis data.
3. Perangkat lunak menyimpan data – data pengguna lahan parkir yang baru masuk pada basis data. Data – data yang disimpan meliputi ID unik tag, nomor kendaraan, kategori pengguna lahan parkir tersebut, dan jam masuk.
4. Kemudian tag tersebut diberikan kepada pengguna lahan parkir, sebagai kartu tanda parkir yang unik dan tidak boleh dihilangkan.
5. Ketika ada pengguna areal parkir yang akan keluar areal parkir, pengguna tersebut menyerahkan kartu parkir yang ia dapatkan ketika masuk yang berupa tag, kepada operator.
6. Kemudian operator meletakkan tag tersebut dekat reader, kembali reader membaca ID unik dari tag tersebut, dan kemudian mengirimkan datanya ke PC. ID unik tersebut kemudian diolah oleh perangkat lunak tersebut, dan mengambil data – data dari basis data dengan ID unik tag yang sesuai. Kemudian data – data tersebut dirampilkkan pada perangkat lunak tersebut.

Universitas Indonesia

7. Kemudian perangkat lunak tersebut juga melakukan pengecekan kategori pengguna areal parkir tersebut, dan melakukan pentarifan dengan kategori yang sesuai.
8. Apabila kartu tag hilang, maka akan dilakukan prosedur tambahan.



Gambar 3.4 – Diagram Alir Sistem Parkir

### 3.4 Tahapan Implementasi

Sistem parkir ini dapat dibagi menjadi dua komponen utama, yaitu dari sisi perangkat keras (*hardware*) yang berupa *peripheral* RFID dan PC, dan sisi perangkat lunak (*software*) yang berupa sistem antar-muka dan sistem informasi.

#### 3.4.1 Implementasi Perangkat Keras

Perangkat keras utama yang digunakan dalam sistem ini adalah RFID reader, RFID tag, dan PC.

##### 3.4.1.1 RFID reader

Yang pertama diperhatikan adalah pemilihan RFID reader. Karena sistem ini masih menggunakan sistem manual (dengan bantuan manusia) dalam proses pembacaan RFID tag, maka spesifikasi reader yang dibutuhkan tidak harus terlalu bagus juga. Dengan jarak baca sekitar 12cm, sudah cukup untuk diimplementasikan.

Pemasangan RFID reader juga disertai dengan pengkabelan untuk catu daya dan komunikasi serial antara reader dengan PC. Untuk penghematan dan kecepatan transfer data, maka jarak antara reader dengan PC diutamakan tidak terlalu jauh  $\pm 3$  meter.

RFID reader dipasang di pintu keluar dan masuk areal parkir. Dengan faktor pemilihan di atas, akan didapatkan sebuah sistem dengan perangkat keras RFID reader rendah biaya, karena spesifikasinya yang rendah namun tetap dapat diandalkan.

Untuk pengembangan selanjutnya, RFID reader ini bisa saja sudah dalam bentuk terpaket dengan tampilan fisik yang bagus dan mudah diletakkan dimana saja.

##### 3.4.1.2 RFID tag

Implementasi yang kedua adalah pemilihan tag sebagai kartu parkir. Karena tag ini juga berfungsi sebagai kartu parkir, maka hal yang harus diperhatikan adalah bentuk dan ukuran fisiknya. Tag yang digunakan baiknya berbentuk tipis seperti kartu ATM.

Universitas Indonesia

Sedangkan untuk jenis tag-nya, yang digunakan adalah tag pasif, dengan daya peng-aktif dari reader. Sehingga usia sumber daya pada reader bisa berumur panjang. Selain itu harga tag seperti itu relatif murah.

#### 3.4.1.3 Personal Computer

PC pada sistem ini berfungsi sebagai pengendali dan jembatan penghubung antara perangkat keras dan perangkat lunak serta sistem informasi. PC menerima data dari reader melalui serial port (COM Port), serta menampilkan informasi melalui display ke user.

Selain itu PC juga berfungsi sebagai basis data dari data – data parkir. Oleh karena itu antara PC yang digunakan sebagai operator dan basis data harus dihubungkan melalui jaringan komputer.

Ketiga perangkat keras tersebut disusun hingga membentuk suatu sistem. Dimana RFID reader dihubungkan ke PC melalui serial port dengan menggunakan kabel coaxial dengan standar RS-232, dan konektor DE-9. Kemudian, PC dihubungkan ke sebuah jaringan komputer, baik itu PC untuk pengendali, maupun PC yang berfungsi sebagai terminal basis data.

#### 3.4.2 Implementasi Perangkat Lunak

Tahapan selanjutnya adalah implementasi perangkat lunak. Perangkat lunak yang diimplementasikan terbagi ke dalam dua kategori, yaitu implementasi perangkat lunak pada pengendali dan perangkat lunak basis data.

Pada sistem pengendali, perangkat lunak yang diimplementasikan berupa sistem antar-muka sebagai jembatan komunikasi antara RFID reader, user, dan basis data. Oleh karena itu, perangkat lunak ini harus berupa program yang dapat melakukan komunikasi secara serial ke peripheral (RFID), dan mampu melakukan sambungan ke basis data.

Dalam hal ini dipilih program berbasis Java disebabkan oleh beberapa hal. Yang pertama adalah, fleksibilitas Java yang dapat berjalan di hamper segala *platform* OS, karena teknologi *Java Virtual Machine* (JVM). Alasan yang kedua adalah, API yang tersedia pada Java sangat lengkap, termasuk API untuk berkomunikasi serial dengan peripheral dan jembatan penghubung dengan

**Universitas Indonesia**

berbagai program basis data seperti Apache Derby atau MySQL. Kemudian yang ketiga, Java merupakan bahasa pemrograman tak berbayar, sehingga biaya pengembangannya sangat rendah.

Setelah melakukan pemilihan program dengan menggunakan Java, implementasi perangkat lunak pada PC pengendali ini juga harus memenuhi beberapa parameter. Parameter yang pertama adalah *Graphical User Interface* (GUI), karena PC pengendali digunakan sebagai antar-muka antara pengendali dengan sistem, maka implementasi ini sangat penting. Dimana user harus dapat memahami dengan baik apa yang ditampilkan oleh program. Kemudian parameter berikutnya adalah dokumentasi program yang baik. Dokumentasi ini akan sangat berguna untuk melakukan pengembangan lanjutan terhadap program dan sistem ini.

Implementasi perangkat lunak berlanjut pada pemilihan sistem basis data yang digunakan. Sistem basis data yang paling banyak digunakan saat ini adalah MySQL, dengan sistem *query* menggunakan SQL yang sesuai dengan fitur – fitur yang ada pada Java. Sehingga proses komunikasi antara program pengontrol dengan sistem basis data dapat berjalan dengan lancar, baik dari segi aplikasi maupun pengembangan sistemnya.

### **3.5 Rencana Uji Coba**

Karena sistem ini terbagi menjadi dua komponen utama seperti yang dijelaskan pada sub-bab sebelumnya, maka rencana pengujian pun juga dapat dibagi menjadi dua kategori, pengujian pada sistem informasi dan perangkat lunak, dan pengujian pada sisi perangkat kerasnya.

#### **3.5.1 Pengujian perangkat keras**

Rencana pengujian yang pertama adalah melakukan uji coba RFID reader. Parameter – parameter yang diuji-cobakan adalah jarak baca reader dengan tag, kebenaran pembacaan ID pada tag, dan uji coba pembacaan pada tag yang bertumpuk. Secara tidak langsung, pengujian – pengujian tersebut juga menguji kemampuan reader dalam berkomunikasi serial dengan PC pengendali. Hasil uji

coba ini akan menentukan apakah reader yang digunakan layak untuk digunakan pada sistem ini atau tidak.

Kemudian uji coba yang kedua adalah pengkabelan pada jaringan komputer. Parameter – parameter yang diukur adalah waktu peng-akses-an basis data oleh PC pengendali pada jarak tertentu. Hasil dari pengujian ini akan menentukan jarak ideal untuk meletakkan PC basis data.

### 3.5.2 Pengujian perangkat lunak

Pengujian perangkat lunak meliputi pengujian terhadap program antarmuka pada PC pengendali dan pengujian terhadap basis data. Pada perangkat lunak, parameter yang diuji adalah keberadaan *bug* pada program. Uji coba yang dilakukan adalah dengan cara melakukan *debugging* terhadap program tersebut. Hasil dari pengujian ini adalah sebuah program dengan versi yang lebih baru dan sebagai landasan untuk pengembangan program.

Sedangkan uji coba pada sisi basis data memiliki parameter ketahanan dan kecepatan akses. Uji coba yang dilakukan untuk mengetahui parameter tersebut adalah peng-akses-an database oleh program – program pada PC pengendali secara serempak, lalu kemudian diukur kecepatan aksesnya. Hasilnya akan diketahui, perbandingan spesifikasi PC basis data dengan jumlah peng-akses basis data tersebut, yang akan mempengaruhi jumlah gerbang masuk-keluar parkir tersebut.

Uji coba pada sisi perangkat lunak juga dilakukan dengan melakukan pengukuran kecepatan akses basis data tergantung dari spesifikasi PC yang digunakan.

## **BAB IV**

### **IMPLEMENTASI SISTEM PARKIR BERBASIS RFID DAN UJI COBA**

#### **4.1 Implementasi Sistem Parkir**

Implementasi sistem parkir berbasis RFID ini terbagi menjadi dua bagian utama. Pertama adalah implementasi alat (perangkat keras), dan yang kedua adalah implementasi perangkat lunak yang berupa program antar-muka dan basis data.

##### **4.1.1 Implementasi Perangkat Keras**

Perangkat keras yang diimplementasikan berupa RFID reader, RFID tag, dan PC, yang masing – masing akan ditempatkan pada gerbang masuk-keluar sebagai pengendali dan identifikasi kartu parkir, dan ditempatkan sebagai pusat basis data yang akan diakses baik oleh pengendali pada gerbang masuk maupun gerbang keluar.

###### **4.1.1.1 RFID Reader**

RFID reader yang digunakan pada implementasi sistem parkir ini adalah modul RFID starter kit buatan innovative-electronics. Gambar 4.1 menunjukkan modul dari RFID starter kit tersebut. RFID reader ini memiliki spesifikasi sebagai berikut.

- Berbasis RFID reader ID-12 dengan frekuensi kerja 125 kHz untuk kartu berformat EM4001 / sejenis.
- ID-12 dapat membaca kartu RFID pasif bentuk ISO card hingga jarak 12 cm.
- Kompatibel dengan varian RFID reader lainnya, antara lain: ID-2, ID-10, dan ID-20.
- Mendukung varian RFID reader / writer, antara lain: ID-2RW, ID-12RW, dan ID-20RW.

- Mendukung format data ASCII (UART TTL / RS-232), Wiegand26, maupun Magnetic ABA Track2 (Magnet Emulation).
- Dilengkapi dengan buzzer sebagai indikator baca, serta LED sebagai indikator tulis.
- Tersedia jalur komunikasi serial UART RS-232 dengan konektor RJ11 dan kabel serial untuk menghubungkan modul ini ke COM port komputer.
- Dilengkapi regulator tegangan 5VDC yang membutuhkan input catu daya 9 – 12 VDC.



Gambar 4.1 – RFID Starter Kit

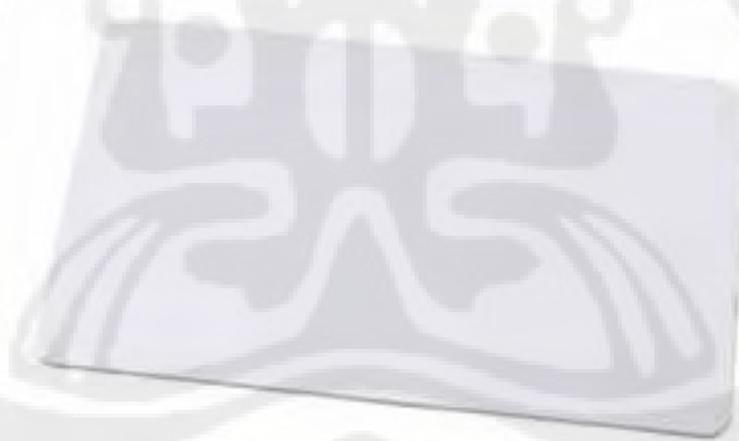
RFID reader yang diimplementasikan dapat berupa reader yang telah dilakukan *packaging*, sehingga bentuknya lebih tipis dan aman dari benturan serta zat cair. Contoh bentuk RFID reader yang telah dibungkus sedemikian rupa itu dapat dilihat pada Gambar 4.2.



Gambar 4.2 – Bentuk RFID reader yang sudah terbungkus

#### 4.1.1.2 RFID Tag

RFID tag yang digunakan pada implementasi sistem ini menggunakan RFID tag pasif, karena beberapa faktor yang telah disebutkan pada bagian pemodelan. RFID tag pasif tersebut berstandar ISO/IEC 7810:2003, dengan ukuran ID-1 (berdimensi  $85.60 \times 53.98$  mm, banyak digunakan pada perbankan dan kartu ID). Bentuknya yang seperti kartu ATM sangat cocok untuk aplikasi ini, Gambar 4.3 menunjukkan bentuk tag tersebut.



Gambar 4.3 – RFID tag Pasif dengan ukuran ID-1

Dengan bentuk seperti ini, RFID tag tersebut dapat dengan mudah diberikan gambar, logo, warna – warna, atau simbol untuk menegaskan keberadaannya sebagai sebuah kartu parkir.

**Universitas Indonesia**

Setelah semua alat tersedia, maka tahapan selanjutnya adalah pemasangan alat – alat tersebut, yaitu menghubungkan RFID Starter Kit dengan PC untuk dapat melakukan komunikasi serial. Gambar 4.4 menunjukkan alat – alat yang telah terpasang.



Gambar 4.4 – Implementasi Perangkat Keras

#### 4.2 Uji Coba RFID Reader dan RFID Tag

Pengujian yang dilakukan terhadap RFID reader adalah dengan melakukan pembacaan tag dengan variasi jumlah tag. Jumlah tag yang digunakan ada 3 dengan spesifikasi pembacaan sebagai berikut.

1. Tag A: 810004168C
2. Tag B: 0103F4CB61
3. Tag C: 0103F4B8CF

#### 4.2.1 Percobaan 1 – Pembacaan Tag C

Tabel 4.1 menunjukkan percobaan dengan pembacaan tag C sebanyak sepuluh kali pembacaan.

Tabel 4.1 – Percobaan Tag C

Pembacaan ke-	Hasil Pembacaan
1	0103F4B8CF
2	0103F4B8CF
3	0103F4B8CF
4	0103F4B8CF
5	0103F4B8CF
6	0103F4B8CF
7	0103F4B8CF
8	0103F4B8CF
9	0103F4B8CF
10	0103F4B8CF

#### 4.2.2 Percobaan 2 – Pembacaan Tag B

Tabel 4.2 menunjukkan percobaan dengan pembacaan tag B sebanyak sepuluh kali pembacaan.

Tabel 4.2 – Percobaan Tag B

Pembacaan ke-	Hasil Pembacaan
1	0103F4CB61
2	0103F4CB61
3	0103F4CB61
4	0103F4CB61
5	0103F4CB61
6	0103F4CB61
7	0103F4CB61
8	0103F4CB61
9	0103F4CB61
10	0103F4CB61

#### 4.2.3 Percobaan 3 – Pembacaan Tag A

Tabel 4.3 menunjukkan percobaan dengan pembacaan tag A sebanyak sepuluh kali pembacaan.

Tabel 4.3 – Percobaan Tag A

Pembacaan ke-	Hasil Pembacaan
1	810004168C
2	810004168C
3	810004168C
4	810004168C
5	810004168C
6	810004168C
7	810004168C
8	810004168C
9	810004168C
10	810004168C

#### 4.2.4 Percobaan 4 – Pembacaan Tag A dan Tag B

Tabel 4.4 menunjukkan percobaan dengan pembacaan tag A dan tag B yang ditumpuk sebanyak sepuluh kali pembacaan.

Tabel 4.4 – Percobaan Tag A dan Tag B

Pembacaan ke-	Hasil Pembacaan
1	0103F4CB61
2	0103F4CB61
3	0103F4CB61
4	0103F4CB61
5	0103F4CB61
6	0103F4CB61
7	0103F4CB61
8	0103F4CB61
9	0103F4CB61
10	0103F4CB61

#### 4.2.5 Percobaan 5 – Pembacaan Tag B dan Tag C

Tabel 4.5 menunjukkan percobaan dengan pembacaan tag B dan tag C yang ditumpuk sebanyak sepuluh kali pembacaan.

Tabel 4.5 – Percobaan Tag B dan Tag C

Pembacaan ke-	Hasil Pembacaan
1	0103F4B8CF
2	0103F4B8CF
3	0103F4B8CF
4	0103F4B8CF
5	0103F4B8CF
6	0103F4B8CF
7	0103F4B8CF
8	0103F4B8CF
9	0103F4B8CF
10	0103F4B8CF

#### 4.2.6 Percobaan 6 – Pembacaan Tag A dan Tag C

Tabel 4.6 menunjukkan percobaan dengan pembacaan tag A dan tag C yang ditumpuk sebanyak sepuluh kali pembacaan.

Tabel 4.6 – Percobaan Tag B dan Tag C

Pembacaan ke-	Hasil Pembacaan
1	0103F4B8CF
2	0103F4B8CF
3	0103F4B8CF
4	0103F4B8CF
5	0103F4B8CF
6	0103F4B8CF
7	0103F4B8CF
8	0103F4B8CF
9	0103F4B8CF
10	0103F4B8CF

#### 4.2.7 Analisa Hasil Percobaan

Dari keenam percobaan di atas, dapat kita lihat bahwa pembacaan reader terhadap tag-nya adalah akurat 100%. Hal ini berarti, apabila ada tag yang terletak pada jarak jangkauan reader, maka tag tersebut akan terbaca seperti apa yang tertera pada ID-nya. Sehingga apabila tag yang didekatkan itu tag tunggal, maka ID dari tag tersebut akan terbaca dan masuk ke dalam komputer.

Namun jika tag yang didekatkan lebih dari 1 tag, maka tag yang paling dominan yang akan terbaca. Dalam dua percobaan di atas terlihat bahwa Tag C memiliki kecenderungan untuk dibaca dibandingkan Tag yang lainnya. Sedangkan tag A menjadi yang paling sulit terbaca bila sedang digabungkan dengan tag – tag lainnya.

Dengan akurasi pembacaan dan tidak adanya bit loss pada pentransmisi data dari tag ke reader, dan dari reader ke komputer, maka dapat dikatakan alat tersebut bekerja dengan baik dan memenuhi standar yang diinginkan untuk digunakan pada implementasi sistem parkir ini.

#### 4.3 Pemrograman dan Uji Coba Perangkat Lunak

Rancang bangun dari sistem parkir dengan antarmuka JAVA ini memiliki satu buah program utama yang mempunyai 3 peran sekaligus. Yang pertama adalah perannya sebagai jembatan komunikasi antara PC dengan RFID reader, dalam perannya tersebut, program JAVA menggunakan Giovynet Serial Port sebagai JAVA API untuk komunikasi dengan serial port pada platform Windows. Yang kedua adalah perannya sebagai antarmuka dengan user atau pengendali, untuk melaksanakan perannya tersebut, Swing GUI pada Java dipergunakan untuk menghasilkan tampilan antarmuka yang nyaman dan baik. Peran ketiga adalah sebagai jembatan penghubung antara program dengan basis data MySQL 5.1. Untuk melaksanakan perannya yang terakhir ini, digunakan JDBC API yang dihubungkan dengan MySQL connector/J, dalam hal ini adalah mysql-connector-java versi 5.1.7.

Untuk menjalankan peran – perannya tersebut, pada *constructor* program tersebut diinstansiasikan 1 *object* pengatur koneksi dengan COM Port dan 1 *object*

tampilan antarmuka. Kedua *object* tersebut bernama Com dan GUI. Berikut adalah *constructor* program ini.

```

Reader () throws Exception
{
    //set parameter COM Port
    Parameters param = new Parameters();
    //instansiasi object Com
    comPort = new Com(param);
    //menutup semua Port
    comPort.closePortC(port);
    //scan port yang tersedia pada komputer, lalu
    //ditampilkan pada window setting
    Object[] portList =
        ComPort.getFreeSerialPort().toArray();
    //Instansiasi GUI
    GUI = new UserInterface(this);
    GUI.updatePortList(portList);
    openConnection(); //buka sambungan ke basis data
    scanning(); //mulai memindai port serial
}

```

Pada *constructor* ini yang pertama kali dilakukan adalah membuka sambungan ke serial port (*default* COM1, Baud Rate 9600 bps, Data Bits 8, Stop Bits 1, No Parity), kemudian melakukan pemindaian terhadap jumlah port yang dapat digunakan pada komputer tersebut. Kemudian program membuka sambungan ke basis data dan melakukan pemindaian terhadap inputan dari serial port.

#### 4.3.1 Koneksi Program dengan Serial Port

Koneksi program dengan serial port terbentuk ketika instansiasi sebuah *object* dari class Com (Class import dari API Giovynet serial port) terbentuk. Dalam hal ini koneksi terbentuk pada saat program mulai dijalankan (terdapat pada *constructor*).

Setelah koneksi tersambung, maka program akan mulai menjalankan *method* scanning(). *Method* ini merupakan *method* dengan skema *forever loop*,

hal tersebut dilakukan agar program terus melakukan pemindaian terhadap serial port yang telah dibuka, menunggu inputan dari reader. Sehingga ketika operator mengambil tag, dan reader membacanya, reader mengirimkan data biner serial ke komputer dan ditangkap oleh program tersebut.

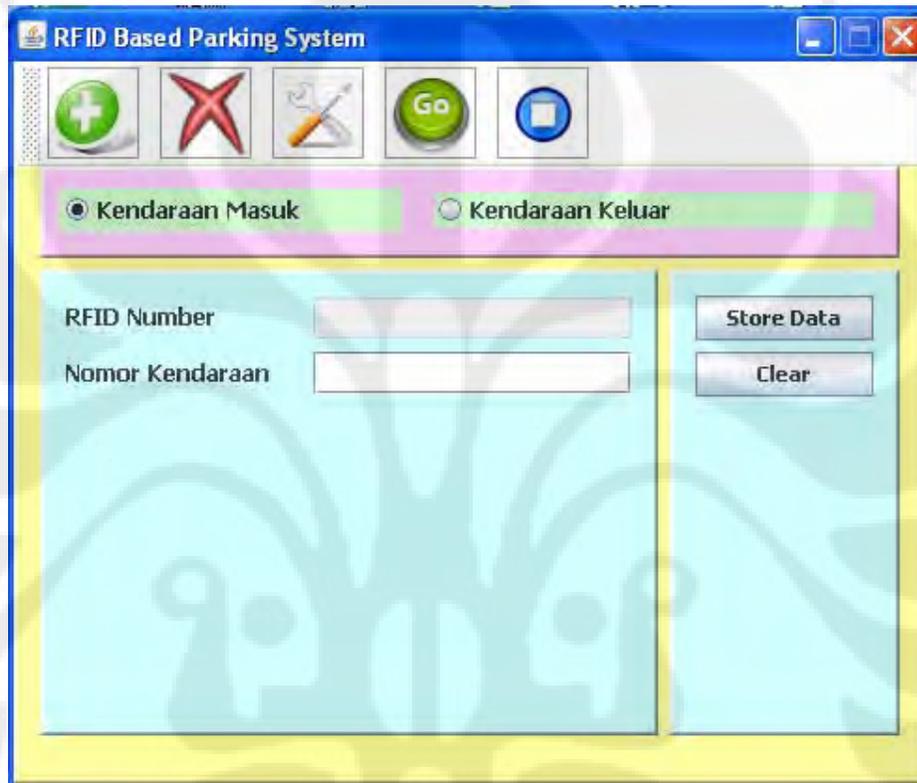
Selain berfungsi untuk memindai inputan dari reader, *method* ini juga berfungsi untuk melakukan konversi data yang dikirim oleh reader. Dimana data yang dikirimkan oleh reader adalah data Hexadecimal, 8 nibble, kemudian dikonversi menjadi bentuk decimal, agar lebih mudah dalam pengamatan ID di basis data. Gambar 4.5 menunjukkan tampilan program ketika program sedang melakukan pemindaian terhadap serial port.



Gambar 4.5 – GUI 1, Memindai Serial Port

#### 4.3.2 Program Sebagai *Graphical User Interface* (GUI)

Berikut ini merupakan tampilan utama dari user interface untuk program *RFID Based Parking System*. Gambar 4.6 menunjukkan tampilan utuh di layar komputer dari Program.



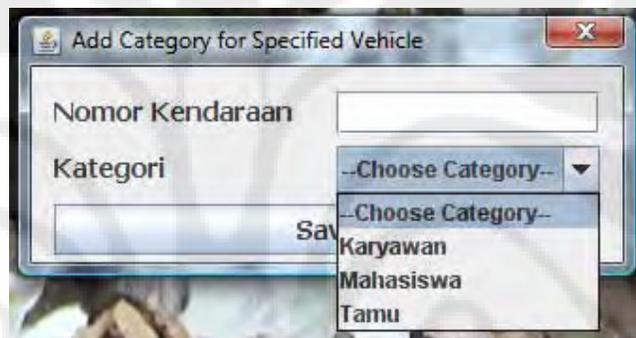
Gambar 4.6 – Tampilan *user interface* Program

Program tersebut terdiri dari 4 panel, tiap panel memiliki tombol dan *field* yang berbeda dengan fungsi yang berbeda juga. Berikut penjelasan masing-masing panel.

1. Panel pertama berisikan *Toolbar* dengan 5 buah tombol, dari kiri ke kanan terdapat tombol *Add Category*, *Delete Parking Data*, *Configuration*, *Start Reading from RFID Reader*, *Stop Reading from RFID Reader*.
  - a. Tombol *Add Category* memiliki fungsi untuk menentukan kategori dari pengguna parkir, apakah pemilik kendaraan tersebut adalah Karyawan, Mahasiswa atau Tamu. Menu ini digunakan untuk *me-record* data

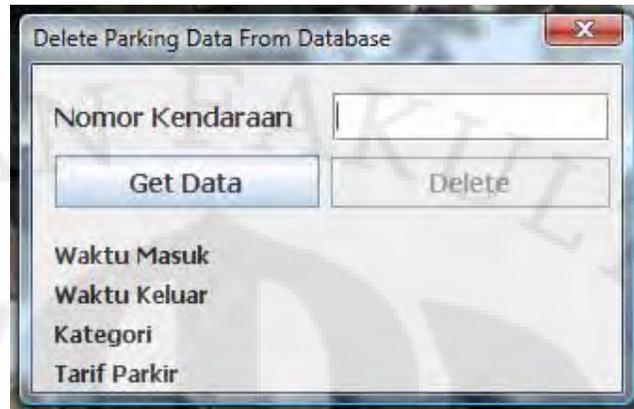
Universitas Indonesia

kategori dari kendaraan dan penggunanya kendaaraannya. Data – data tersebut disimpan dalam tabel “category\_data” pada database. Data tersebut digunakan untuk menentukan kategori pengguna areal parkir yang akan keluar, untuk penentuan tarif parkir. Selain itu, proses recording data ini hanya dilakukan sekali saja di awal pengguna kendaraan tersebut masuk areal parkir. Apabila di lain waktu pengguna tersebut kembali lagi, operator tidak perlu memasukkan kembali data orang tersebut. Gambar 4.7 menunjukkan tampilan dari menu *Add Category*.



Gambar 4.7 – Menu *Add Category*

- b. Tombol *Delete Parking Data* memiliki fungsi untuk menghapus data parkir dari kendaraan yang telah masuk. Metode diperlukan jika kartu parkir (RFID Tag) yang dibawa oleh pengguna lahan parkir tersebut hilang. Tampilan awal ketika mengklik tombol *Delete Parking Data* pada *toolbar* ini ditunjukkan pada Gambar 4.8.



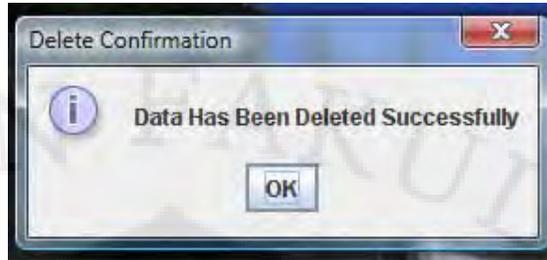
Gambar 4.8 – Tampilan awal *Delete Data Window*

Proses dimulai dengan cara memasukkan nomor kendaraan, kemudian mengklik tombol Get Data akan muncul data-data dari kendaraan tersebut seperti yang ditampilkan pada Gambar 4.9.



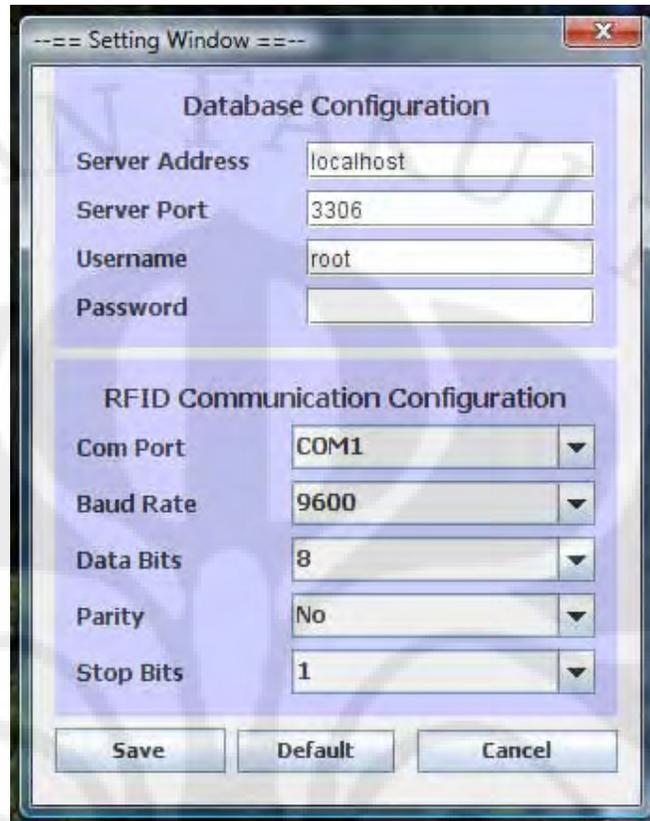
Gambar 4.9 – Tampilan lanjut *Delete Data Window*

Jika tombol delete di-klik, maka data dari kendaraan tersebut akan dihapus dari database, dan akan muncul notifikasi seperti pada Gambar 4.10.



Gambar 4.10 – Notifikasi hapus data

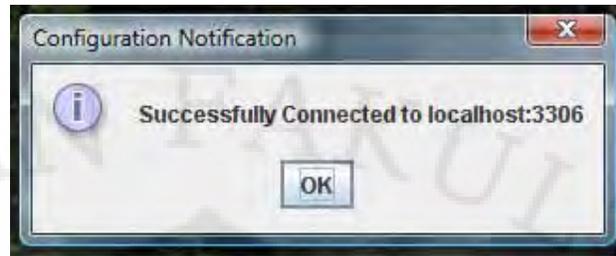
- c. Tombol *Configuration* mengatur konektivitas dari PC ke Database dan dari PC ke RFID Reader. *Panel* sebelah atas digunakan untuk konfigurasi konektivitas ke *Database Server*. *Field* “*Server Address*” dapat diisi dengan alamat IP dari *Database server*, pada contoh di gambar ini, *Server Address* diisi dengan *localhost* (atau 127.0.0.1) karena *Database Server*-nya terdapat pada komputer lokal. Kemudian *field* “*Server Port*” diisi dengan nomor port yang digunakan oleh *database server*. Kemudian *Field Username* dan *Password* diisi dengan *username* dan *password* dari *Database server* yang dituju. Gambar 4.11 menunjukkan tampilan dari menu konfigurasi.



Gambar 4.11 – Tampilan Menu Konfigurasi

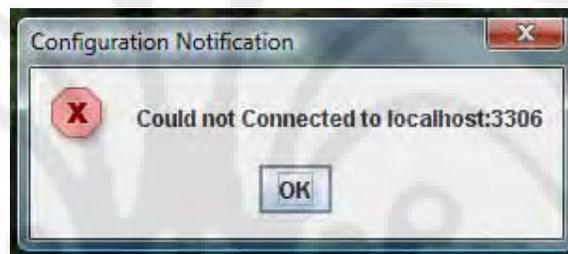
Kemudian untuk *Panel* sebelah bawah digunakan untuk mengatur konektivitas ke RFID Reader, terdiri dari 5 *combo box*, untuk mempermudah pengaturan hubungan dengan reader. *Field* “Com Port” diisi dengan nomor port yang sesuai dengan pada Com Port mana RFID *Reader*-nya terkoneksi dengan PC.

Apabila tombol “Save” di klik dan muncul notifikasi seperti pada Gambar 4.12.



Gambar 4.12 – Notifikasi Keberhasilan Konfigurasi

Notifikasi tersebut menandakan bahwa aplikasi telah terhubung ke *database server* yang dituju dan juga terhubung ke RFID Reader. Apabila setelah mengklik tombol “Save” muncul gambar seperti pada Gambar 4.13.



Gambar 4.13 – Notifikasi Kesalahan Konfigurasi

Notifikasi tersebut menandakan bahwa aplikasi gagal terhubung ke *Database Server*, ini berarti ada kesalahan pengisian pada konfigurasinya.

d. Tombol Start Reading

Tombol ini digunakan untuk memulai melakukan pemindaian serial port untuk menerima input dari reader. Apabila program mulai melakukan pemindaian, maka akan terlihat label yang bertuliskan “*Put The RFID tag near the receiver please...*” seperti pada gambar 4.5 di atas.

e. Tombol *Stop Reading*

Tombol ini digunakan untuk menghentikan proses pemindaian serial port oleh program, sehingga program tidak lagi dapat membaca tag yang didekatkan ke reader. Jika tombol ini di klik maka label yang bertuliskan “*Put The RFID tag near the receiver please...*” akan hilang seperti pada gambar 4.6 di atas.

2. Panel kedua memiliki dua buah *radio button* yang berlabel “Kendaraan Masuk” dan “Kendaraan Keluar”. Penjelasan masing-masing tombol sebagai berikut:

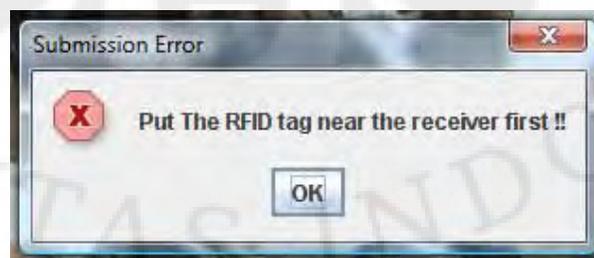
- a. Jika *radio button* “Kendaraan Masuk” diklik maka akan mempengaruhi kedua panel di bawahnya. Semua tombol pada panel sebelah kanan bawah dapat berfungsi dan juga *field* “Nomor Kendaraan” akan di *enable*. Ini artinya, tombol “Kendaraan masuk” digunakan untuk menyimpan data dari kendaraan yang masuk setelah *tag* dibaca oleh *RFID Reader*. Tampilannya sama dengan gambar 4.6.
- b. Jika tombol “Kendaraan Keluar” diklik maka tombol “Store Data” pada panel sebelah kanan bawah akan di non-aktifkan dan Field Nomor Kendaraan akan di-disable . Ini artinya, tombol “Kendaraan Keluar” digunakan untuk menampilkan data dari kendaraan yang keluar setelah tag dibaca oleh *RFID Reader*. Data – data tersebut diambil dari database server tabel “*parking\_data*”. Gambar 4.14 menunjukkan tampilan jika mode kendaraan keluar dipilih:



Gambar 4.14 – GUI 2, Kendaraan Keluar

3. Panel ketiga

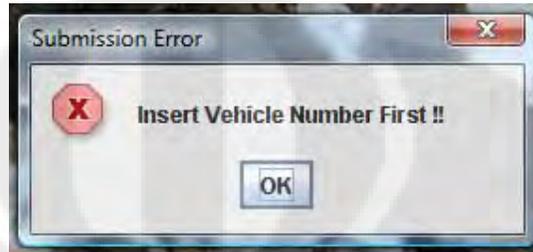
Pada panel ketiga terdapat dua buah tombol yaitu tombol “*Store Data*” dan tombol “*Clear*”. Tombol “*Store Data*” berfungsi untuk menyimpan data dari kendaraan yang masuk setelah *field RFID Number* dan *field Nomor Kendaraan* pada panel keempat terisi dengan data yang sesuai. Apabila *field RFID Number* belum terisi, maka akan terjadi pesan kesalahan seperti yang ditunjukkan pada Gambar 4.15.



Gambar 4.15 – Pesan Kesalahan 1

Universitas Indonesia

Sedangkan apabila *field RFID Number* telah terisi namun *field Vehicle Number* belum terisi, maka akan muncul pesan kesalahan seperti yang ditunjukkan pada Gambar 4.16.



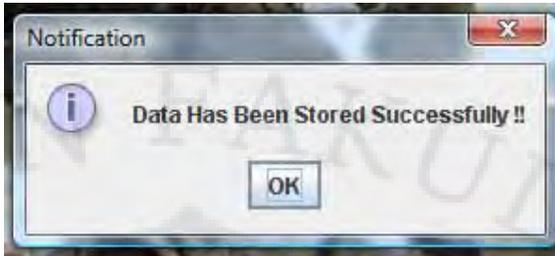
Gambar 4.16 – Pesan Kesalahan 2

Program ini juga dapat melakukan pengecekan apakah nomor kendaraan yang bersangkutan sudah terdaftar di dalam database atau belum. Hal ini sebagai penambah tingkat keamanan sistem parkir, bahwa nomor polisi kendaraan yang akan memasuki areal parkir harus lah nomor yang asli. Gambar 4.17 menunjukkan pesan kesalahan yang muncul apabila nomor kendaraan yang dituliskan sudah berada di dalam database server.



Gambar 4.17 – Pesan Kesalahan 3

Dan jika semua syarat sudah terpenuhi, maka muncul pesan konfirmasi penyimpanan data seperti yang ditunjukkan pada Gambar 4.18.



Gambar 4.18 – Konfirmasi Penyimpanan Data

Tombol “Clear” berfungsi untuk menghapus isi dari *field – field* yang berada pada panel sebelah kiri (panel keempat).

#### 4. Panel keempat

Panel keempat berisi dua buah *text field*, yang pertama yaitu *field* RFID *Number* yang akan berisi Nomor dari RFID *tag* yang didapat setelah *tag* dibaca oleh RFID *Reader*, dan kemudian diteruskan melalui komunikasi serial ke komputer. Kemudian *field* Nomor Kendaraan yang bisa diisi dengan nomor polisi dari kendaraan yang masuk pada mode *Kendaraan Masuk* (*radio button* yang terpilih adalah *Kendaraan Masuk*). Sedangkan pada mode *Kendaraan Keluar* (*radio button* yang terpilih adalah *Kendaraan Keluar*), *text field* tersebut hanya berfungsi sebagai penampil data – data parkir.

Selain kedua *text field* tersebut, panel ini juga memiliki *label – label* tersembunyi yang berfungsi sebagai keterangan atau tampilan dari data parkir pada mode *Kendaraan Keluar*. *Label – label* tersebut hanya akan tampak apabila RFID *tag* dari pengguna areal parkir yang ingin keluar area parkir didekatkan ke reader, kemudian ID-nya digunakan untuk mengambil data – data parkir pada database, maka data – data parkir tersebut akan ditampilkan di *label – label* tersembunyi tersebut. Gambar 4.19 menunjukkan tampilan dari *label – label* tersembunyi tersebut.



Gambar 4.19 – Label Keterangan

Tampilan dari panel tersebut dapat dibersihkan dengan mengklik tombol clear pada panel ketiga. Namun, jika bila nomor ID yang diberikan tidak ditemukan di dalam database, maka akan muncul pesan kesalahan seperti yang ditunjukkan pada Gambar 4.20.



Gambar 4.20 – Database Error

### 4.3.3 Koneksi Program dengan MySQL Database

Koneksi program dengan database dimulai ketika *method* `openConnection()` dijalankan. Berikut adalah penjabaran *method* tersebut.

```
public boolean openConnection()
{
    try
    {
        //memuat class driver database
        //(com.mysql.jdbc.Driver)
        Class.forName(JDBC_DRIVER);
        //membuka sambungan dengan database
        con=DriverManager.getConnection(DATABASE_URL,
            DATABASE_USERNAME, DATABASE_PASSWORD);
        return true;
    }
    catch (Exception exception)
    {
        exception.printStackTrace();
        return false;
    }
}
```

Yang dilakukan oleh *method* ini adalah pertama – tama memuat *class* `driver` untuk database yang digunakan, dalam hal ini adalah MySQL. `Driver` tersebut adalah “`com.mysql.jdbc.Driver`”. Setelah `driver`-nya termuat, maka program mulai melakukan sambungan ke database sesuai dengan `DATABASE_URL`, `DATABASE_USERNAME`, dan `DATABASE_PASSWORD` yang diberikan. Jika usaha membuka sambungan ke database ini berhasil, maka *method* tersebut akan mengembalikan nilai *boolean* `true`. Namun jika gagal, maka *method* ini akan melakukan *exception*.

### 4.3.4 Melakukan Query pada Database

*Method* yang digunakan untuk melakukan *query* pada database adalah sebagai berikut.

```
public ResultSet executeQuery(String query)
    throws SQLException
{
    //membuat statement untuk query database
    Statement st = con.createStatement
        (ResultSet.TYPE_SCROLL_SENSITIVE,
         ResultSet.CONCUR_UPDATABLE);
    //jalankan query, hasilnya disimpan di resultSet
    ResultSet resultSet = st.executeQuery(query);
    return resultSet;
}
```

Yang dilakukan oleh *method* ini adalah, pertama kali membuat statement untuk melaksanakan query. *Class* Statement ini merupakan *class* yang terdapat pada *package* “java.sql.” Setelah statement dibuat, maka *query* di-eksekusi dan hasilnya disimpan dalam bentuk ResultSet. *Query* yang di-eksekusi tersebut tertulis dalam SQL.

## **BAB V**

### **KESIMPULAN**

Teknologi RFID, dapat dikembangkan menjadi berbagai macam aplikasi proses identifikasi. Salah satunya adalah sistem parkir yang terhubung dengan dunia internet. Dalam buku ini, telah dilakukan perancangan Sistem Parkir Berbasis RFID dengan menggunakan antarmuka Java dan basis data MySQL, yang merupakan aplikasi yang dapat mempermudah pekerjaan manusia di masa datang. Sistem ini dapat dikembangkan sesuai kebutuhan sehingga mendukung pemanfaatan dalam skala penggunaan yang besar, mencakup lebih banyak pengguna.

Beberapa hal yang dapat disimpulkan dari pemodelan dan implementasi sistem pada skripsi, setelah melalui berbagai rangkaian uji coba ini adalah sebagai berikut.

1. Pemodelan dan *prototype* Sistem Parkir yang berbasis RFID ini siap untuk dirancang menjadi sebuah sistem yang utuh, dan siap untuk dikembangkan lebih lanjut.
2. Dengan penggunaan antarmuka *Java* yang bersifat *open source code*, model dari sistem ini diharapkan pengembangan *prototype* ini nantinya akan menjadi lebih mudah.

Dari hasil yang didapat ini, akan dijadikan bahan studi lebih lanjut untuk pengembangan sistem parkir berbasis RFID dan Web.

## DAFTAR ACUAN

- [1] “The History of RFID Technology”  
<http://www.rfidjournal.com/article/articleview/1338/1/129> (200905.10)
- [2] “Shrouds of Time The history of RFID” (AIM, Association for Automatic Identification and Mobility).
- [3] “What is RFID?”  
<http://www.rfidjournal.com/article/articleview/1339/1/129/> (200905.10)
- [4] “The Basics of RFID Technology”  
<http://www.rfidjournal.com/article/view/1337/1> (200905.10)
- [5] James Gosling, *A brief history of the Green project*. Java.net, [Q1/1998]. April 29, 2007.

## DAFTAR PUSTAKA

- James Gosling, Bill Joy, Guy Steele, and Gilad Bracha, *The Java language specification, third edition*. Addison-Wesley, 2005.
- Committee identification: ISO/IEC JTC /SC 31/WG 4. *Information technology — Radio frequency identification for item management — Unique identification for RF tags*. 2008-06-24
- Savi Technologies. *Active and Passive RFID and Selecting the Right Active Frequency*. 2002.
- Taesu Cheong and Youngil Kim. *RFID Data Management and RFID Information Value Chain Support with RFID Middleware Platform Implementation*. Springer Berlin / Heidelberg: Tuesday, October 11, 2005.
- Shouqin Zhou, Weiqing Ling and Zhongxiao Peng. *An RFID-based remote monitoring system for enterprise internal production management*. Springer Berlin / Heidelberg: Tuesday, April 04, 2006.
- Byunggil Lee and Howon Kim. *A Design of Context aware Smart Home Safety Management using by Networked RFID and Sensor*. Springer Boston: Friday, March 07, 2008.
- Architecture, implementation and application of complex event processing in enterprise information systems based on RFID

## LAMPIRAN

### Lampiran 1: Source Code Reader.java

```
/**
 * @author
 * Adytiawan Arga Dwitama
 * 0405030044
 * Eleectrical Engineering
 * University of Indonesia *
 */

package wbps;

import app.*;
import core.SerialPort;
import java.lang.String;
import java.util.*;
import java.sql.*;
import java.text.*;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;

public class Reader
{
    // GUI for this program
    UserInterface GUI;
```

(Lampiran 1: Source Code Reader.java)

```
// Initialize variable for scanning Port COM
boolean reading = false, portStatus = false, vehicleOut = false;
public String port = "COM1";
public String baudRate = "9600";
public String dataBits = "8";
public String stopBits = "1";
public String parity = "N";
int[] numOfPort;

SerialPort serialPort;
//Array portList;

char data;
int ID_KEY;
int counter=0;
Com comPort;

// Set Up Parameter for Database Connection
// JDBC driver name and database URL
static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
public String DATABASE_URL = "jdbc:mysql://localhost:3306/wbps";
public String DATABASE_USERNAME = "root";
public String DATABASE_PASSWORD = "";

Connection con = null; // manages connection
Statement statement = null; // SQL statement
```

Universitas Indonesia

(Lampiran 1: Source Code Reader.java)

```
static final String INSERT_RECORD = "INSERT INTO  
parking_data(Nomor_RFID, Nomor_Kendaraan" +  
    ", Kategori, Waktu_Masuk, Status) values(?, ?, ?, ?, ?)";  
static final String INSERT_CATEGORY = "INSERT INTO  
category_data(Nomor_Kendaraan, Kategori) values(?, ?)";
```

```
public boolean openConnection()  
{  
    try  
    {  
        Class.forName(JDBC_DRIVER);  
        con = DriverManager.getConnection(DATABASE_URL,  
DATABASE_USERNAME, DATABASE_PASSWORD);  
        return true;  
    }  
    catch (Exception exception)  
    {  
        exception.printStackTrace();  
        return false;  
    }  
}
```

```
public boolean closeConnection()  
{  
    try  
    {  
        con.close();  
        return true;  
    }  
}
```

Universitas Indonesia

(Lampiran 1: Source Code Reader.java)

```
catch ( Exception exception )
{
    exception.printStackTrace();
    return false;
}
}

public ResultSet executeQuery(String query) throws SQLException
{
    Statement st =
con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
                    ResultSet.CONCUR_UPDATABLE);
    ResultSet resultSet = st.executeQuery(query);
    return resultSet;
}

public boolean saveCategory(String vehicleNumber, String category)
{
    try
    {
        ResultSet resultSet = executeQuery("SELECT Nomor_Kendaraan FROM
category_data WHERE" +
        " Nomor_Kendaraan='"+vehicleNumber+"'");
        if (resultSet.next()) {return false;}
        else
        {
            PreparedStatement pstmt =
con.prepareStatement(INSERT_CATEGORY);
            pstmt.setString(1, vehicleNumber);

```

Universitas Indonesia

(Lampiran 1: Source Code Reader.java)

```
pstmt.setString(2, category);
    pstmt.executeUpdate();
    System.out.println("1 row affected");
    return true;
}
}
catch (SQLException s)
{
    System.out.println("SQL Statement is not Executed !!");
    System.out.println(s.fillInStackTrace());
    return false;
}
}

public boolean storeData(String rfidNumber, String vehicleNumber)
{
    try
    {
        ResultSet resultSet = executeQuery("SELECT Nomor_Kendaraan FROM
parking_data WHERE" +
        " Nomor_RFID='"+rfidNumber+"' AND Status='in'");
        if (resultSet.next()) {return false;}
        else
        {
            PreparedStatement pstmt = con.prepareStatement(INSERT_RECORD);
            java.sql.Timestamp timeIn = new java.sql.Timestamp(new
java.util.Date().getTime());
```

Universitas Indonesia

(Lampiran 1: Source Code Reader.java)

```
pstmt.setString(1, rfidNumber);
    pstmt.setString(2, vehicleNumber);
    pstmt.setString(3, searchCategory(vehicleNumber));
    pstmt.setTimestamp(4, timeIn);
    pstmt.setString(5, "in");
    pstmt.executeUpdate();
    System.out.println("1 row affected");
    return true;
}
}
catch (SQLException s)
{
    System.out.println("SQL Statement is not Executed !!");
    System.out.println(s.fillInStackTrace());
    return false;
}
}

public void showInfo(Information information)
{
    Information info = information;
    if (info.match)
    {
        GUI.vehicleNumberTxt.setText(info.vehicleNumber);
        GUI.kategoriLbl.setText(info.kategori);
        GUI.waktuMasukLbl.setText(info.waktuMasuk);
        GUI.waktuKeluarLbl.setText(info.waktuKeluar);
        if (info.tamu)
        {
```

Universitas Indonesia

(Lampiran 1: Source Code Reader.java)

```
GUI.tarifParkirLbl.setText(info.tarifParkir);
GUI.tarifParkirLbl.setVisible(true);
GUI.tarLbl.setVisible(true);
}
GUI.keterangan(true);
}
else
{
    GUI.showNotification("Database Error", "No Matching Pair, Check The
RFID Tag", true);
    GUI.rfidNumberTxt.setText("");
}
}

public void deleteVehicleData(String vehicle, Timestamp time)
{
    try
    {
        ResultSet resultSet = executeQuery("SELECT * FROM parking_data
WHERE Nomor_Kendaraan="
        + vehicle + " " + "AND Status='in'");
        if(resultSet.next())
        {
            do
            {
                resultSet.updateTimestamp("Waktu_Keluar", time);
                resultSet.updateString("Status", "out");
                resultSet.updateString("Keterangan", "Kartu Hilang");
                resultSet.updateRow();
            }
        }
    }
}
```

Universitas Indonesia

(Lampiran 1: Source Code Reader.java)

```

    }while(resultSet.next());
    }
}
catch (SQLException ex)
{
    System.out.println("SQL Statement is not Executed !!");
    System.out.println(ex.fillInStackTrace());
}
}

public Information retrieveInformation(Information info, ResultSet resultSet)
throws SQLException
{
    java.sql.Timestamp timeOut = new java.sql.Timestamp(new
java.util.Date().getTime());
    info.waktuKeluar = timeOut.toLocaleString();
    info.timeOut = timeOut;
    resultSet.updateRow();
    info.kategori = resultSet.getString("Kategori");
    info.waktuMasuk =
resultSet.getTimestamp("Waktu_Masuk").toLocaleString();
    info.tamu = false;
    info.vehicleNumber = resultSet.getString("Nomor_Kendaraan");
    if (resultSet.getString("Kategori").equalsIgnoreCase("Tamu"))
    {
        info.tarifParkir =String.valueOf(((Integer.parseInt(Long.toString(
(timeOut.getTime()
        - resultSet.getTimestamp("Waktu_Masuk").getTime())/1000)))*100);
    }
}

```

Universitas Indonesia

(Lampiran 1: Source Code Reader.java)

```
        info.tamu = true;
    }
    info.match = true;
    return info;
}

public Information getVehicleDataByVehicleNumber(String vehicle)
{
    Information info = new Information();
    try
    {
        ResultSet resultSet = executeQuery("SELECT * FROM parking_data
        WHERE Nomor_Kendaraan="
            + vehicle + " " + "AND Status='in'");
        if(resultSet.next())
        {
            do
            {
                info = retrieveInformation(info, resultSet);
            }while(resultSet.next());
        }
        else
        {
            info.match = false;
            System.out.println("No Matching pair");
        }
    } catch (SQLException ex) {
        System.out.println("SQL Statement is not Executed !!");
    }
}
```

Universitas Indonesia

(Lampiran 1: Source Code Reader.java)

```
        System.out.println(ex.fillInStackTrace());
    }

    return info;
}

public void getVehicleDataById(String rfidNumber)
{
    Information info = new Information();
    try
    {
        ResultSet resultSet = executeQuery("SELECT * FROM parking_data
        WHERE Nomor_RFID="
            + rfidNumber + "' " + "AND Status='in'");
        if(resultSet.next())
        {
            do
            {
                info = retrieveInformation(info, resultSet);
                resultSet.updateTimestamp("Waktu_Keluar", info.timeOut);
                resultSet.updateString("Status", "out");
                resultSet.updateString("Keterangan", "Normal");
                resultSet.updateRow();
            }while(resultSet.next());
        }
        else
        {
            info.match = false;
            System.out.println("No Matching pair");
        }
    }
}
```

Universitas Indonesia

(Lampiran 1: Source Code Reader.java)

```
    }  
  }  
  catch (SQLException ex)  
  {  
    System.out.println("SQL Statement is not Executed !!");  
    System.out.println(ex.fillInStackTrace());  
  }  
  showInfo(info);  
}  
  
public String searchCategory(String vehicleNumber)  
{  
  String category = "";  
  try  
  {  
    ResultSet resultSet = executeQuery("SELECT Kategori FROM  
category_data WHERE Nomor_Kendaraan='"+vehicleNumber+"'");  
    if(resultSet.next())  
    {  
      do  
      {  
        category = (String) resultSet.getObject(1);  
      }while(resultSet.next());  
    }  
    else  
    {  
      category = "Tamun";  
    }  
  }  
}
```

Universitas Indonesia

(Lampiran 1: Source Code Reader.java)

```
}
catch (SQLException ex)
{
    System.out.println("SQL Statement is not Executed !!");
    System.out.println(ex.fillInStackTrace());
}

return category;
}

public void scanning() throws Exception
{

    String ID = "";
    while (true)
    {
        if (reading)
        {
            GUI.infoLabel.setVisible(true);
            try
            {
                data = comPort.receiveSingleChar();
                counter++;

                switch (counter)
                {
                    case 1:
                        if ((int) data != 2)
                        {
```

Universitas Indonesia

```
        counter = 0;
        ID = "";
    }
    else
    {
        System.out.println("Start Reading RFID Transponder!!");
    }
    break;

case 4:
case 5:
case 6:
case 7:
case 8:
case 9:
case 10:
case 11:
    ID = ID + data;
    break;

case 16:

    if ((int) data == 3)
    {
        GUI.rfidNumberTxt.setText("" + Integer.parseInt(ID, 16));
        System.out.println("End of Data");
        if (vehicleOut == true)
        {
            getVehicleDataById (" " + Integer.parseInt(ID, 16));
        }
    }
}
```

```

        else
        {
            System.out.println("End Data Invalid");
        }
        counter = 0;
        ID = "";
        break;
    }
}
catch (Exception ex)
{
    GUI.showNotification("COM Port Error", "Invalid Selected COM
Port !\n" +
        "Make Sure The COM Port's Exist and The Device's Attached
Properly to The Computer", true);
    reading = false;
    ex.printStackTrace();
}
else{GUI.infoLabel.setVisible(false);}
}
}

Reader () throws Exception
{
    Parameters param = new Parameters();
    comPort = new Com(param);
    comPort.closePortC(port);
    Object[] portList = (comPort.getFreeSerialPort()).toArray();
    GUI = new UserInterface(this);
    GUI.updatePortList(portList);
}

```

(Lampiran 1: Source Code Reader.java)

```
openConnection();
scanning();
}

public static void main(String[] args) throws Exception
{
    new Reader();
}
}
```