



**UNIVERSITAS INDONESIA**

**PERANCANGAN DAN EVALUASI  
SISTEM APLIKASI TRANSAKSI BUKU MASUK  
PADA RAK PERPUSTAKAAN  
DENGAN PENGGUNAAN TEKNOLOGI RFID BERBASIS  
TCP/IP DAN JAVA**

**SKRIPSI**

**ZULFIKAR WIDYASENO  
04 05 03 085Y**

**FAKULTAS TEKNIK  
DEPARTEMEN TEKNIK ELEKTRO  
DEPOK  
DESEMBER, 2009**



**UNIVERSITAS INDONESIA**

**PERANCANGAN DAN EVALUASI  
SISTEM APLIKASI TRANSAKSI BUKU MASUK  
PADA RAK PERPUSTAKAAN  
DENGAN PENGGUNAAN TEKNOLOGI RFID BERBASIS  
TCP/IP DAN JAVA**

**SKRIPSI**

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

**ZULFIKAR WIDYASENO  
04 05 03 085Y**

**FAKULTAS TEKNIK  
DEPARTEMEN TEKNIK ELEKTRO  
DEPOK  
DESEMBER, 2009**

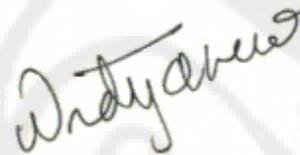
## HALAMAN PERNYATAAN ORISINALITAS

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

Nama : Zulfikar Widyaseno

NPM : 040503085Y

Tanda Tangan :



Tanggal : 30 Desember 2009

## HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :

Nama : Zulfikar Widyaseno

NPM : 040503085Y

Program Studi : Teknik Elektro

Judul Skripsi : Perancangan Dan Evaluasi Sistem Aplikasi Transaksi Buku  
Masuk Pada Rak Perpustakaan Dengan Penggunaan Teknologi  
RFID Berbasis TCP/IP Dan Java

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 : Prof. Dr-ing Ir. Kalamullah Ramli, M.Eng. (Kalamullah)

Penguji : Prof. Dr. Ir. Bagio Budiarto (Bagio)

Penguji : Prima Dewi Purnamasari, S.T. M.Sc. (Prima Dewi)

Ditetapkan di : Depok

Tanggal : 30 Desember 2009

## KATA PENGANTAR

Puji syukur saya panjatkan kehadiran Allah SWT, karena atas berkat dan rahmat-Nya, saya dapat menyelesaikan skripsi ini. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Teknik Program Studi Teknik Elektro pada Fakultas Teknik Universitas Indonesia. Saya menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan skripsi ini, sangatlah sulit bagi saya untuk menyelesaikan skripsi ini. Oleh karena itu, saya mengucapkan terima kasih kepada:

- (1) Bapak Prof. Dr-Ing. Ir. Kalamullah Ramli, M. Eng. selaku dosen pembimbing yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan saya dalam penyusunan skripsi ini;
- (2) Kedua orang tua dan keluarga saya yang telah memberikan bantuan dukungan material dan moral;
- (3) Sahabat-sahabat beserta pihak lain yang baik secara langsung maupun tidak langsung telah banyak membantu saya dalam menyelesaikan skripsi ini.

Akhir kata, saya berharap Allah SWT berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu.

Depok, 30 Desember 2009

Penulis

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

---

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

Nama : Zulfikar Widyaseno

NPM : 0405037057

Program Studi : Teknik Elektro

Departemen : Teknik Elektro

Fakultas : Teknik

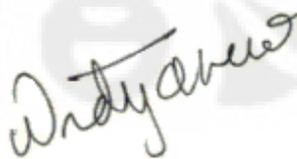
Jenis karya : Skripsi

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

**PERANCANGAN DAN EVALUASI SISTEM APLIKASI TRANSAKSI  
BUKU MASUK PADA RAK PERPUSTAKAAN DENGAN PENGGUNAAN  
TEKNOLOGI RFID BERBASIS TCP/IP DAN JAVA**

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 : 30 Desember 2009  
Yang menyatakan



( Zulfikar Widyaseno )

## ABSTRAK

Nama : Zulfikar Widyaseno  
Program Studi : Teknik Elektro  
Judul : Perancangan Dan Evaluasi Sistem Aplikasi Transaksi Buku  
Masuk Pada Rak Perpustakaan Dengan Penggunaan  
Teknologi RFID Berbasis TCP/IP Dan Java

Otomasi kegiatan identifikasi obyek menjadi salah satu upaya dalam mewujudkan suatu perpustakaan yang modern. Pada skripsi ini, kegiatan tersebut dikhususkan untuk transaksi buku masuk pada rak dimana buku tersebut berada. Modul pengenalan ditempatkan pada masing-masing rak buku berupa pembaca RFID. Jumlah Pembaca RFID yang digunakan dalam perpustakaan bergantung pada banyaknya rak dalam perpustakaan.

Untuk kemudahan komunikasi antar system di dalam perpustakaan, pembaca-pembaca RFID ditempatkan ke dalam jaringan TCP/IP. Pada server, paket data ditangkap dengan suatu aplikasi Java yang membentuk bagian sistem yang menangani proses bisnis kegiatan otomasi perpustakaan. Proses yang dilakukan pada server ditangani Java applet yang diintegrasikan dalam web. Teknologi yang dipakai pada bagian web adalah Java Server Pages (JSP).

Sistem terdiri atas dua bagian, yaitu sisi perangkat keras dan sisi perangkat lunak. Sistem aplikasi dibangun dengan gabungan teknologi RFID, TCP/IP, Java, JSP dan MySQL. Perbedaan dengan sistem RFID lain adalah pada basis komunikasinya yang menggunakan protokol TCP/IP dan *middleware* yang berupa Java applet.

Dari ujicoba didapatkan waktu rata-rata tanggap applet total sebesar 15109286,38  $\mu$ s dan waktu *load* rata-rata total semua halaman web adalah 56787,76  $\mu$ s. Waktu *load* halaman web tidak bergantung pada spesifikasi komputernya melainkan dari bagaimana server menerima dan memproses *request*, serta mem-*forward*-nya ke komputer client. Karakteristik dan sensitivitas pembacaan RFID yang ditemukan dalam pengujian dapat dijadikan pertimbangan dalam pengembangan sistem berbasis RFID berikutnya.

**Kata Kunci :** *RFID, TCP/IP, Java, JSP, Perpustakaan*

## ABSTRACT

Name : Zulfikar Widyaseno  
Study Program : Electrical Engineering  
Title : Design and Evaluation of Application Transaction System of  
Book Incoming to Shelf In Library With TCP/IP Based RFID  
Techology and Java

Automation of object identification activities become one of the efforts in creating a modern library. In this thesis, the activity is focused in detecting the book incoming transactions on the shelf whereby the book is located. The identification module (i.e RFID reader) is placed on each shelf. The number of RFID readers used in the library depends on the number of the shelves in the library.

For the purpose of simple communication between systems in the library, RFID readers are made compatible with TCP/IP network. On the server, the data packet is captured by Java application which builds part of the system that handling the business process of library automation activities. The process performed by the server is handled by Java applet. The technology used for the web is Java Server Pages (JSP).

The systems consists of two parts, namely the hardware and software component. The system is built with the combination of RFID technology, TCP/IP, Java, JSP and MySQL. Differences with similar RFID systems are the basis of communications, which uses TCP/IP protocols, and the Java applet that acts as system middleware.

Results obtained during the tests show the average of total applet response time 15109286.38  $\mu$ s and average of total web load time 56787.76  $\mu$ s. Web page loading time does not depend on computer specifications, but on how the server receives and processes the request, and then forwarding it to the client computer. Characteristics and sensitivity of RFID readings obtained in the tests can be used in the development of the future RFID-based systems.

**Keywords:** *RFID, TCP/IP, Java, JSP, Library*

## DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PERNYATAAN ORISINALITAS.....	ii
HALAMAN PENGESAHAN... ..	iii
KATA PENGANTAR.....	iv
HALAMAN PERSETUJUAN PUBLIKASI SKRIPSI.....	v
ABSTRAK.....	vi
DAFTAR ISI.....	viii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xi
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Tujuan.....	2
1.3 Pembatasan Masalah.....	2
1.4 Sistematika Tulisan.....	2
BAB 2 LANDASAN TEORI.....	3
2.1. Radio Frequency Identification (RFID) .....	3
2.1.1 RFID Tag.....	3
2.1.2 Pembaca RFID.....	6
2.2 WIZ110SR.....	7
2.2.1 Fitur dan spesifikasi.....	8
2.2.2 Diagram Blok.....	9
2.2.3 Mixed mode.....	10
2.3 Teknologi Java.....	11
2.3.1 Applet.....	13
2.4 JavaServer Pages.....	13
BAB 3 PERANCANGAN.....	16
3.1 Skenario.....	16
3.2 Desain.....	17
3.3 Perancangan Perangkat Keras.....	19
3.2.1 Peralatan .....	19
3.2.2 Konfigurasi.....	21
3.4 Perancangan Perangkat Lunak.....	25
3.4.1 Database.....	25
3.4.2 Applet.....	26
3.4.3 Halaman Web.....	30
BAB 4 PENGUJIAN DAN EVALUASI.....	31
4.1 Pengujian Fungsional Sistem.....	31
4.1.1 Pengujian Fungsional Perangkat Keras.....	31
4.1.2 Pengujian Fungsional Perangkat Lunak.....	33
4.2 Pengujian dengan Arah Baca Tertentu.....	37
4.2.1 Pembacaan dengan tag horizontal.....	37
4.2.2 Pembacaan dengan tag vertikal.....	40

4.3 Pengujian Pembacaan dengan Kondisi Khusus.....	43
4.3.1 Pembacaan dengan Interval Waktu.....	44
4.3.2 Pembacaan dengan Dua Tag dibuat berhimpit.....	45
4.3.3 Pembacaan dengan satu tag diam dan satu tag berjalan.....	46
4.3.4 Pembacaan dengan dua reader dibuat berhadapan.....	47
4.4 Pengujian Tanggapan Fungsi Applet.....	48
4.4.1 Pembacaan dua tag pada dua reader yang berbeda.....	50
4.4.2 Pembacaan dua tag pada satu reader.....	52
4.4.3 Pembacaan satu tag pada dua reader.....	53
4.5 Pengujian Halaman Web.....	54
4.5.1 Halaman Tulisan.....	56
4.5.2 Lihat Data.....	57
4.5.3 Tambah Data.....	58
4.5.4 Edit Data.....	58
4.5.5 Update Data.....	59
4.5.6 Delete(1): Konfirmasi.....	60
4.5.7 Delete(2): Hapus.....	61
4.5.8 Hasil Pengujian Waktu load Halaman Web.....	62
4.6 Perbandingan Dengan Sistem RFID Lain Yang Sudah Ada.....	64
4.6 Faktor Yang Patut Diperhatikan Dalam Pengembangan Sistem RFID.....	65
BAB 5 KESIMPULAN.....	67
DAFTAR ACUAN.....	69
DAFTAR PUSTAKA.....	70
LAMPIRAN.....	71

## DAFTAR TABEL

Tabel 2. 1 Struktur Kelas RFID .....	5
Tabel 2. 2 Spesifikasi WIZ110SR.....	9
Tabel 4. 1 Hasil pengujian fungsional sisi perangkat keras .....	33
Tabel 4. 2 Hasil pengujian fungsional sisi perangkat lunak.....	36
Tabel 4. 3 Pembacaan tag horizontal pada arah y+.....	38
Tabel 4. 4 Pembacaan tag horizontal pada arah y-.....	38
Tabel 4. 5 Pembacaan tag horizontal pada arah x+.....	38
Tabel 4. 6 Pembacaan tag horizontal pada arah x-.....	39
Tabel 4. 7 Pembacaan tag vertikal pada arah y+.....	40
Tabel 4. 8 Pembacaan tag vertikal pada arah y-.....	41
Tabel 4. 9 Pembacaan tag vertikal pada arah x+.....	41
Tabel 4. 10 Pembacaan tag vertikal pada arah x- .....	41
Tabel 4. 11 Pembacaan tag dengan interval waktu .....	44
Tabel 4. 12 Hasil pembacaan uji dua tag yang dibuat berhimpit .....	45
Tabel 4. 13 Pembacaan satu tag diam dan satu tag berjalan .....	46
Tabel 4. 14 Pembacaan dua reader dibuat berhadapan .....	48
Tabel 4. 15 Hasil tanggapan pembacaan dua tag pada dua reader yang berbeda.....	51
Tabel 4. 16 tanggapan pembacaan satu tag pada dua reader yang berbeda .....	52
Tabel 4. 17 tanggapan pembacaan dua tag pada satu reader yang berbeda .....	54
Tabel 4. 18 Waktu load halaman utama dan help .....	56
Tabel 4. 19 Waktu load halaman lihat data.....	57
Tabel 4. 20 Waktu load halaman tambah data .....	58
Tabel 4. 21 Waktu load halaman edit data.....	59
Tabel 4. 22 Waktu load halaman update data .....	60
Tabel 4. 23 Waktu load halaman delete(1): konfirmasi.....	61
Tabel 4. 24 Waktu load halaman delete(2): hapus.....	61
Tabel 4. 25 Perbandingan dengan Sistem RFID lain .....	65

## DAFTAR GAMBAR

Gambar 2. 1	Komponen-komponen dalam sistem RFID.....	3
Gambar 2. 2	Contoh Passive Tag dari Texas Instruments Incorporated (kiri) dan Celis Semiconductor Corporation (kanan) .....	4
Gambar 2. 3	Komponen Fisik pembaca RFID.....	7
Gambar 2. 4	Antarmuka WIZ110SR .....	8
Gambar 2. 5	Diagram Blok dari WIZ110SR .....	9
Gambar 2. 6	TCP Client mode pada WIZ110SR.....	10
Gambar 2. 7	Gambaran proses pengembangan perangkat lunak Java.....	11
Gambar 2. 8	"write once, run anywhere" pada Java .....	12
Gambar 2. 9	Alir request/Response flow pada pemanggilan halaman JSP .....	14
Gambar 3. 1	Skema rancangan sistem .....	18
Gambar 3. 2	Tag RFID Pasif Class-1 berbentuk kartu .....	19
Gambar 3. 3	RFID Starter-Kit.....	20
Gambar 3. 4	TCP/IP Module (WIZ110SR) .....	20
Gambar 3. 5	Setting Jumper Pada RFID Starter-Kit.....	21
Gambar 3. 6	Konfigurasi Network Pada WIZ110SR.....	22
Gambar 3. 7	Konfigurasi Serial Pada WIZ110SR .....	22
Gambar 3. 8	Konfigurasi Opsional Pada WIZ110SR .....	23
Gambar 3. 9	Skema Modul Unit Pembaca.....	23
Gambar 3. 10	Penempatan Tag RFID pada Buku.....	24
Gambar 3. 11	Skema Unit Pembaca RFID .....	24
Gambar 3. 12	Diagram relasi database .....	25
Gambar 3. 13	Activity diagram applet.....	27
Gambar 3. 14	Diagram aliran proses data dari RFID reader hingga applet.....	28
Gambar 3. 15	Diagram konsep web .....	30
Gambar 4. 1	Paket data terbaca oleh Wireshark .....	32
Gambar 4. 2	Notifikasi awal ketika applet berhasil dijalankan .....	34
Gambar 4. 3	Notifikasi awal ketika applet gagal dijalankan .....	35
Gambar 4. 4	Contoh notifikasi ketika ada data buku berisi tag rfid masuk.....	36
Gambar 4. 5	Ilustrasi pembacaan tag secara horizontal .....	37
Gambar 4. 6	Ilustrasi pembacaan tag secara vertikal .....	40
Gambar 4. 7	Perkiraan pola radiasi reader .....	42
Gambar 4. 8	Perkiraan pola radiasi tag .....	43
Gambar 4. 9	Ilustrasi pengujian baca dengan interval waktu .....	43
Gambar 4. 10	Ilustrasi pengujian baca dengan dua tag dibuat berhimpit.....	44

Gambar 4. 11 Ilustrasi pengujian baca dengan satu tag diam dan satu tag berjalan, (a) tag diam berada di dekat reader (b) tag berjalan berada di dekat reader .....	45
Gambar 4. 12 Ilustrasi pengujian pembacaan dengan dua reader dibuat berhadapan .....	47
Gambar 4. 13 Ilustrasi pengujian pembacaan dua tag pada dua reader yang berbeda.....	50
Gambar 4. 14 Ilustrasi pengujian pembacaan dua tag pada satu reader yang berbeda.....	52
Gambar 4. 15 Ilustrasi pengujian pembacaan satu tag pada dua reader yang berbeda.....	54
Gambar 4. 16 Grafik hasil pengujian waktu load halaman website dalam satuan waktu ( $\mu$ s) .....	63

## **BAB 1 PENDAHULUAN**

### **1.1 Latar Belakang**

Definisi perpustakaan menurut Undang-undang Nomor 43 Tahun 2007 adalah institusi pengelola koleksi karya tulis, karya cetak, dan/atau karya rekam secara profesional dengan sistem yang baku guna memenuhi kebutuhan pendidikan, penelitian, pelestarian, informasi, dan rekreasi para pemustaka. Perpustakaan berfungsi sebagai wahana pendidikan, penelitian, pelestarian, informasi, dan rekreasi untuk meningkatkan kecerdasan dan keberdayaan bangsa. Serta mempunyai tujuan memberikan layanan kepada pemustaka, meningkatkan kegemaran membaca, serta memperluas wawasan dan pengetahuan untuk mencerdaskan kehidupan bangsa.

Seiring dengan perkembangan zaman, munculah keberadaan perpustakaan yang mengaplikasikan teknologi-teknologi terbaru yang berkembang saat itu. Perpustakaan itu sering disebut sebagai perpustakaan modern. Perpustakaan modern telah didefinisikan sebagai tempat untuk mengakses informasi dalam format apa pun, apakah informasi itu disimpan dalam gedung perpustakaan tersebut atau tidak. Dalam perpustakaan modern ini selain kumpulan buku tercetak, sebagian buku dan koleksinya ada dalam perpustakaan digital (dalam bentuk data yang bisa diakses lewat jaringan komputer).

Salah satu bentuk memodernisasi perpustakaan adalah mengotomasi kegiatan-kegiatan yang berlangsung dalam perpustakaan tersebut. Pada rancangan tugas akhir ini, bentuk otomasi kegiatan yang dikhususkan pada pengenalan literatur-literatur dalam perpustakaan contohnya buku. Selain itu juga perancangan terhadap sistem informasi sederhana yang akan diintegrasikan dengan sistem pengenalan tersebut. Teknologi pengenalan berbasis RFID yang melewatkan datanya melalui jaringan TCP/IP.

## **1.2 Tujuan**

Tujuan penulisan makalah ini adalah merancang dan mengimplementasikan aplikasi otomasi sistem perpustakaan berbasis RFID pada perangkat keras dan Java serta JSP pada Server perpustakaan. Pengambilan data oleh pembaca RFID ditangkap oleh aplikasi Java dan disajikan pada web.

## **1.3 Pembatasan Masalah**

Pada tulisan ini akan dibahas mengenai konsep dan desain pembuatan aplikasi RFID yang terhubung pada jaringan TCP/IP. Kemudian data hasil pembacaan RFID ditangkap oleh aplikasi Java untuk kemudian diproses sedemikian rupa sehingga dapat diakses melalui web yang berbasis JSP.

## **1.4 Sistematika Tulisan**

Sistematika penulisan pada seminar ini ialah sebagai berikut :

### **BAB 1 Pendahuluan**

Terdiri dari latar belakang masalah, tujuan skripsi, batasan masalah dan sistematika penulisan.

### **BAB 2 Landasan Teori**

Membahas mengenai Teknologi RFID, TCP/IP, divais WIZ110SR, Java Applet, dan yang terakhir JavaServer Pages (JSP).

### **BAB 3 Perancangan**

Membahas mengenai perancangan sistem aplikasi secara umum dan secara khusus membahas mengenai perancangan skema perangkat keras serta perangkat lunak sistem ini.

### **BAB 4 Pengujian dan Evaluasi**

Membahas mengenai pengujian-pengujian yang dilakukan beserta evaluasi perbandingan sistem yang dibuat dengan sistem RFID lain yang sudah ada.

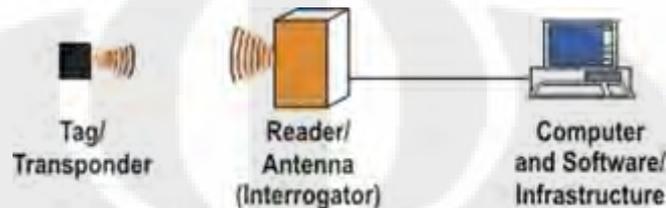
### **BAB 5 Kesimpulan**

Merupakan kesimpulan pembahasan pada penulisan tugas akhir ini.

**Universitas Indonesia**

## BAB 2 LANDASAN TEORI

### 2.1. Radio Frequency Identification (RFID)



Anonim "What is RFID?" [http://www.aimglobal.org/technologies/rfid/what\\_is\\_rfid.asp](http://www.aimglobal.org/technologies/rfid/what_is_rfid.asp) Akses terakhir: 01 Juni 2009

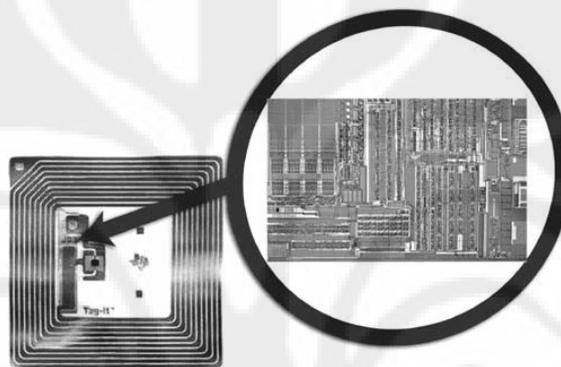
**Gambar 2. 1 Komponen-komponen dalam sistem RFID**

Dalam artian umum, RFID (Radio Frequency Identification) yang berarti mengenali seseorang atau objek menggunakan transmisi frekuensi radio. Teknologi ini dapat digunakan untuk mengidentifikasi, melacak, menyortir atau mendekteksi berbagai macam objek. Komunikasi terjadi antara sebuah *reader* (*interrogator*) dan sebuah *transponder* (chip silikon yang dihubungkan dengan antena) yang biasa disebut *tag*. Tag bisa berjenis aktif (daya dari baterai) atau pasif (daya dari medan *reader*), serta dalam berbagai macam bentuk termasuk *smart-card*. Frekuensi komunikasi yang digunakan bergantung pada seberapa besar aplikasinya, dan berkisar antara 125 kHz sampai 2,45 GHz. [1] Berbeda dengan *bar-code*, RFID tidak memerlukan kontak dalam komunikasinya. Data RFID dapat dibaca melewati tubuh manusia, pakaian dan bahan-bahan non-metal. [2]

#### 2.1.1 RFID Tag

Ada dua komponen utama yang ada dalam tag RFID. Pertama adalah chip silikon kecil atau rangkaian terintegrasi yang mengandung nomor identifikasi

unik. Kedua adalah antena yang dapat mengirim dan menerima gelombang radio. Antena terdiri atas kumparan metal konduktif (Gambar 2) dan chipnya berukuran kurang dari setengah milimeter. Dua komponen ini biasanya disatukan pada tag plastik datar yang dapat ditempel pada benda-benda fisik. Tag-tag ini lumayan kecil dan tipis sehingga bisa ditanam dalam kartu plastik, tiket, label pakaian, dan buku. Dua macam tipe utama tag adalah pasif dan aktif. Tag pasif saat ini diproduksi secara luas karena murah untuk diproduksi. [3]



Einstein, Meyer "Mengenal komponen perangkat keras dari RFID"  
<http://secretatmeyer.blogspot.com/2008/09/mengenal-komponen-perangkat-keras-dari.html> Akses terakhir:  
 02 Desember 2008

**Gambar 2. 2 Contoh Passive Tag dari Texas Instruments Incorporated (kiri) dan Celis Semiconductor Corporation (kanan)**

Tag RFID datang dalam berbagai macam jenis tergantung pada kegunaannya, dan penjenisan ini didefinisikan dalam *RFID Class Structure* oleh Auto-ID Center (kemudian EPC Global) [4]. Struktur dasar tersebut mendefinisikan lima kelas sebagai berikut:

Tabel 2. 1 Struktur Kelas RFID

Kelas	Nama Layer Kelas	Kemampuan
1	Identity Tags	Murni pasif, tag untuk identifikasi
2	Higher Functionality Tags	Murni pasif, identifikasi + beberapa kegunaan lain (baca/tulis memory)
3	Semi-Passive Tags	Tambahan sumber daya baterai
4	Active 'ad hoc' Tags	Komunikasi dengan tag aktif lain
5	Reader Tags	Mampu menyediakan daya untuk dan komunikasi dengan tag lain. Contohnya sebagai reader, mengirim dan menerima gelombang radio

Ward, Matt & van Kranenburg, Rob "RFID: Frequency, standards, adoption and innovation" JISC

Technology and Standards Watch, May 2006

"telah diolah kembali"

Perlu dicatat bahwa terminologi Kelas 1-5 hanya digunakan oleh EPC Global, dan bukan RFID secara umum. Setiap kelas secara berturut-turut dibangun oleh kemampuan yang disediakan layer-bawah. Untuk memahami bagaimana cara bekerjanya akan sangat membantu jika mengetahui terlebih dahulu yang dimaksud dengan pasif, semi-pasif, dan aktif dalam penggunaan sumber dayanya [5].

- Sistem Tag Pasif

Tidak memerlukan sumber daya *on-board* sehingga sistem ini dapat mencari sumber daya dari *reader* untuk menjalankan logika digital dalam chip dan memberikan respon ke *reader*. Sistem ini hanya dapat bekerja ketika ada *reader*. Jarak komunikasinya terbatas pada kebutuhan *reader* untuk membangkitkan sinyal yang kuat untuk memberikan tenaga kepada tag, sehingga terbatas pada jarak reader-ke-tag. Akan tetapi, karena tidak membutuhkan sumber daya yang kontinyu, tag jenis ini mempunyai daur hidup lebih lama selain itu juga lebih murah diproduksi. Sehingga tag

RFID pasif lebih cocok untuk menandai produk-produk individu untuk aplikasi seperti supermarket dan *smart-card*.

- Sistem Tag Semi-pasif

Membutuhkan penggunaan tenaga baterai untuk logika digital dalam chip, tapi masih mengambil tenaga luar untuk komunikasi. Tag semi-pasif jauh lebih handal dan mempunyai jarak baca lebih jauh daripada tag pasif. Tetapi tag jenis ini berumur lebih pendek (karena ketergantungannya pada tenaga baterai, dan lebih mudah pecah, serta lebih mahal).

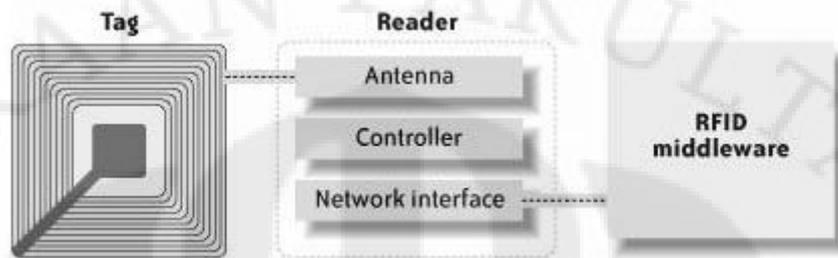
- Sistem Tag Aktif

Mempunyai *transmitter* frekuensi radio yang aktif dan menggunakan baterai untuk menghidupi logika chip dan komunikasi dengan reader. Jarak baca lebih jauh (hingga beberapa kilometer) dan kehandalan meningkat. Tag aktif dapat dibaca ketika bergerak sampai 100 mil/jam (contohnya pada sistem pembayaran jalan-tol) dan mampu membaca sampai seribu tag tiap detik. Tag aktif juga dapat dipasang *built-in sensor* contohnya untuk memonitor temperatur dan melaporkan fluktuasi yang tidak dapat diterima pada produk-produk beku, walaupun dengan begitu harganya juga makin tinggi – sampai £55 (sekitar \$100 atau €80) setiap tagnya [6]. Tag ini juga mempunyai memory yang lebih besar dibandingkan tag pasif, dan karena kapabilitas pemrosesan data yang lebih tinggi, tag ini jauh lebih aman.

### 2.1.2 Pembaca RFID

Pembaca RFID adalah unit *handheld* atau tetap yang dapat menandai tag RFID terdekat dan mendapatkan nomor ID-nya menggunakan komunikasi frekuensi radio (RF). Ketika tag-pasif berada dalam jangkauan *reader*, antena tag menyerap energi yang dipancarkan *reader*, membuat energi tersebut ‘menyalakan’ rangkaian di dalam tag, yang kemudian menggunakan energi tersebut untuk mengirim balik nomor ID dan informasi terkait lainnya.[3]

Ada dua kelas utama pada RFID-reader: *read-only*, contohnya yang beroperasi pada tag pasif EPC Kelas-1, dan *read/write*, yang dapat menuliskan informasi kembali ke dalam tag yang terdapat memori *read/write*. [3]



Bhatt, Himanshu & Glover, Bill "RFID Essentials" O'Reilly, January 2006, ISBN: 0-596-00944-5

**Gambar 2. 3 Komponen Fisik pembaca RFID**

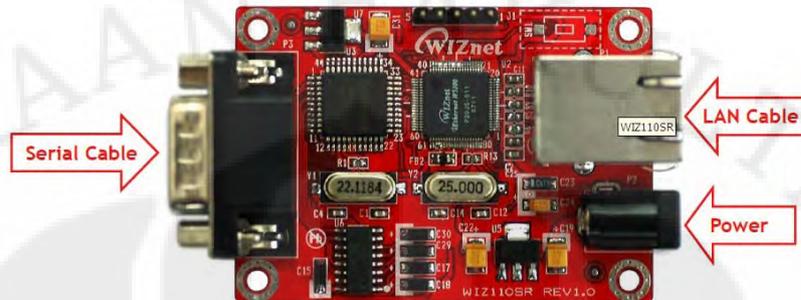
Karena *reader* berkomunikasi dengan tag menggunakan RF, pembaca RFID manapun harus punya satu atau lebih antena. Karena *reader* harus berkomunikasi dengan divais atau *server* lain, *reader* juga harus mempunyai *network interface* atau semacamnya. Contoh *network interface* yang umum adalah *Universal Asynchronous Receiver/Transmitter* (UARTs) untuk komunikasi serial RS232 atau RS485 dan *jack* RJ45 untuk kabel ethernet 10BaseT atau 100BaseT. Beberapa *reader* bahkan sudah punya modul komunikasi *bluetooth* atau *wireless ethernet* didalamnya. Kemudian, untuk implementasi protokol komunikasi dan mengendalikan *transmitter*, setiap *reader* paling tidak mempunyai sebuah microcontroller atau microcomputer. Gambar 2.3 menunjukkan komponen-komponen fisik RFID *reader*. [7]

## 2.2 WIZ110SR

WIZ110SR adalah modul gateway yang mengubah protokol RS232 ke protokol TCP/IP. Alat ini memungkinkan pengukuran jarak jauh, pengaturan dan kontrol suatu divais melalui jaringan yang berdasarkan TCP/IP dengan mengkoneksikan peralatan yang ada dengan antarmuka serial RS232. Dengan kata lain, WIZ110SR adalah konverter protokol yang mengirimkan data yang dikirim peralatan serial sebagai data TCP/IP dan mengkonversi balik data data TCP/IP yang diterima

Universitas Indonesia

melewati jaringan menjadi data serial untuk ditransmisikan kembali ke peralatan tersebut [8].



WIZ110SR User's Manual

**Gambar 2. 4 Antarmuka WIZ110SR**

### 2.2.1 Fitur dan spesifikasi

Berikut adalah fitur-fitur utama yang dimiliki oleh WIZ110SR [8]:

- Sambungan langsung ke divais serial
- Keandalan dan kestabilan sistem dengan menggunakan W5100 Hardware Chip
- Mendukung koneksi PPPoE
- Mendukung konfigurasi serial – dengan “Simple and Easy command”
- Mendukung password untuk keamanan
- *Configuration Tool Program*
- Antarmuka 10/100 Ethernet dan Antarmuka Serial maksimum 230 Kbps
- Mengikuti RoHS

Kemudian spesifikasi dari WIZ110SR ini ditunjukkan dalam tabel berikut [8]:

Tabel 2. 2 Spesifikasi WIZ110SR

Item	Deskripsi
MCU	Mengikuti 8051 (mempunyai internal Flash 62K, 16K SRAM, 2K EEPROM)
TCP/IP	W5100 (Ethernet MAC & PHY Embedded)
Protokol	10/100 Mbps (Auto detection), RJ-45 Connector
Antarmuka Serial	RS232 (DB9)
Sinyal Serial	TXD, RXD, RTS, CTS, GND
Parameter Serial	Parity: None, Even, Odd Data Bits: 7,8 Flow Control: None, RTS/CTS, XON/XOFF Speed: sampai 230Kbps
Tegangan Input	DC 5V
Konsumsi Daya	Dibawah 180mA
Temperatur	0 °C ~ 80 °C (operasi), -40 °C ~ 85 °C (simpan)
Kelembaban	10 ~ 90 %

WIZ110SR User's Manual

“telah diolah kembali”

### 2.2.2 Diagram Blok



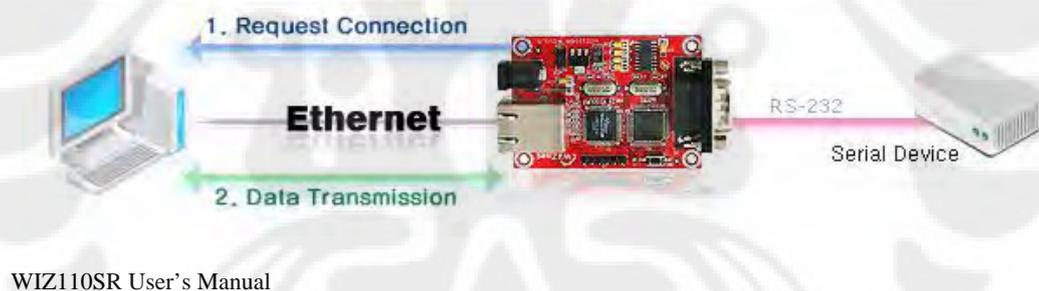
WIZ110SR User's Manual

**Gambar 2. 5 Diagram Blok dari WIZ110SR**

WIZ110SR adalah konverter protokol yang mentransmisikan data yang dikirim peralatan serial ke Ethernet dan mengkonversi balik data TCP/IP yang diterima melalui jaringan menjadi data serial kemabali ke peralatan tersebut. Ketika data diterima melalui port serial, data dikirimkan ke W5100 oleh MCU. Jika ada data yang ditransmisikan melalui Ethernet, data tersebut diterima terlebih dahulu ke buffer internal W5100, kemudian dikirim ke port serial oleh MCU. MCU dalam modul mengendalikan data sesuai dengan nilai konfigurasi yang ditentukan pengguna [8].

### 2.2.3 Mixed mode

WIZ110SR mempunyai tiga macam TCP network mode yaitu client-mode, server-mode, dan mixed-mode. Ini untuk memilih metode komunikasi berdasarkan pada TCP. TCP adalah protokol untuk mengadakan koneksi sebelum komunikasi data, sedangkan UDP hanya memproses komunikasi data tanpa mengadakan koneksi dahulu [8].



**Gambar 2. 6 TCP Client mode pada WIZ110SR**

Pada mode ini, WIZ110SR normalnya beroperasi sebagai TCP Server dan menunggu permintaan koneksi dari *peer*. Akan tetapi, jika WIZ110SR menerima data dari divais serial sebelum koneksi berlangsung, ia akan berubah menjadi client-mode dan mengirim data ke IP server. Sehingga, pada mixed mode, server-mode dioperasikan terlebih dahulu daripada client-mode.

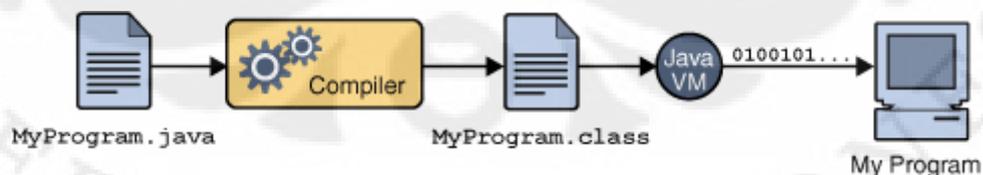
Seperti pada TCP Server mode, Mixed-mode berguna untuk kasus dimana *monitoring center* mencoba mengkoneksikan diri ke divais serial (dimana

WIZ110SR digunakan) untuk mengecek status divais. Sebagai tambahan dari ini, jika ada hal darurat yang terjadi pada divais serial, maka modul akan berubah menjadi Client-mode untuk mengadakan koneksi ke server dan mengantarkan status darurat ke divais.[8]

### 2.3 Teknologi Java

Bahasa pemrograman Java adalah sebuah bahasa pemrograman tingkat tinggi yang dapat dikarakterisasi dengan semua kata-kata berikut ini [9]:

- *Simple* (mudah)
- *Object-oriented* (berorientasi objek)
- *Distributed* (terdistribusi)
- *Interpreted* (dapat diterjemahkan)
- *Robust* (kuat)
- *Secure* (aman)
- *Architectural neutral* (arsitektur netral)
- *Portable* (portabel/mudah dipindahkan)
- *High performance* (performa tinggi)
- *Multithreaded* (bisa di-multithread-kan)
- *Dynamic* (dinamis)

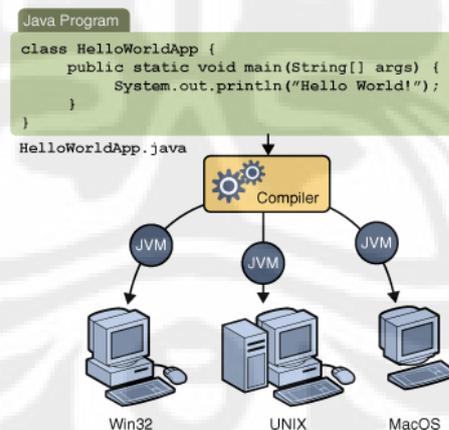


Sun Microsystems, "The Java Tutorial – A Practical Guide for Programmers",  
<http://java.sun.com/docs/books/tutorial> Update terakhir 14 Maret 2008.

**Gambar 2. 7** Gambaran proses pengembangan perangkat lunak Java

Seperti pada kebanyakan bahasa pemrograman, kita biasa mengkompilasi atau menerjemahkan sebuah program agar ia dapat dijalankan di komputer kita. Pada bahasa pemrograman Java, proses kompilasi sebuah program tidak seperti biasanya. Dengan *compiler*, pertama kita menerjemahkan program ke dalam bahasa pertengahan (*intermediate language*) yang dinamakan *Java bytecodes*—sebuah kode-kode yang tidak bergantung pada *platform* (*platform independent*) yang dapat diterjemahkan oleh *interpreter* pada *platform* Java. *Interpreter* menguraikan dan menjalankan setiap instruksi-instruksi *Java bytecodes* pada komputer. Proses kompilasi berlangsung hanya satu kali, interpretasi dilakukan setiap kali program dieksekusi.

*Java bytecodes* membantu terwujudnya "*write once, run anywhere*" atau "tulis sekali, jalan di mana pun". Kita dapat mengkompilasi program kita ke dalam *bytecodes* pada *platform* apapun yang mempunyai *Java compiler*. Kemudian *bytecodes* dapat dijalankan pada semua implementasi *Java VM*. Itu berarti bahwa selama komputer mempunyai *Java VM*, program yang sama yang ditulis dalam bahasa pemrograman Java dapat dijalankan pada Windows 2000, Linux, *workstation* Solaris, atau iMac.



Sun Microsystems, "The Java Tutorial – A Practical Guide for Programmers",  
<http://java.sun.com/docs/books/tutorial> Update terakhir 14 Maret 2008.

**Gambar 2. 8 "write once, run anywhere" pada Java**

### 2.3.1 Applet

Applet adalah program Java khusus dimana *browser* yang mendukung teknologi Java dapat mengunduhnya dan menjalankannya. Applet biasanya menyatu dengan halaman web dan berjalan pada konteks *browser*. Applet merupakan *subclass* dari *class* `java.applet.Applet`, yang menyediakan antarmuka standar antara applet dan lingkungan *browser*. [9]

Daur-hidup applet pada dasarnya terdapat atas empat *method* dalam Applet *class* dimana applet dibangun.[9]

- *init*: *method* ini ditujukan untuk inisialisasi apapun yang dibutuhkan untuk applet. Dipanggil setelah atribut param dari tag applet.
- *start*: *method* ini secara otomatis dipanggil setelah *method* *init*. Dipanggil kapanpun *user* kembali ke halaman yang mengandung applet setelah mengunjungi halaman lain.
- *stop*: *method* ini secara otomatis dipanggil kapanpun *user* pergi dari halaman yang mengandung applet. *Method* ini juga dapat digunakan untuk menghentikan animasi.
- *destroy*: *method* ini hanya dipanggil ketika *browser* ditutup secara normal.

Sehingga applet hanya diinisialisasi sekali, dimulai dan dihentikan berkali-kali dalam daur-hidupnya, dan di-*destroy* hanya sekali saja.

### 2.4 JavaServer Pages

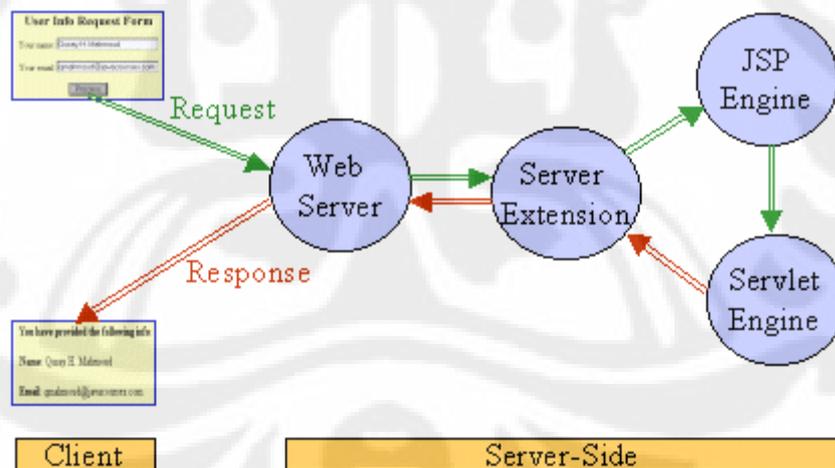
Teknologi JSP merupakan bagian dari teknologi J2EE. Meski demikian halaman JSP dapat dijalankan pada server dengan atau tanpa J2EE. Sepanjang tersedia library dari API JSP maka halaman JSP akan dapat dijalankan.

Halaman JSP merupakan sebuah file teks yang berisikan tag-tag JSP, scriplet untuk membungkus (*encapsulate*) kode Java, content statik (HTML, XML) dan berbagai content lainnya. Pada prinsipnya halaman JSP dapat dianggap sebagai sebuah halaman HTML dengan tambahan content khusus lain. File JSP memiliki ekstensi “.jsp”.

JSP berbedanya dengan JavaScript. JavaScript dijalankan sepenuhnya dalam mesin client, sedangkan JSP dijalankan pada server. JavaScript hanya dapat mengakses *resource* tertentu dalam ruang lingkup di mana dia dipasang (yaitu salam halaman HTML) sementara JSP dapat mengakses semua resource yang ada di dalam aplikasi web.

JSP memisahkan antara presentasi dengan logika bisnis, di mana programmer Java dapat berkonsentrasi pada logika bisnis, sementara programmer web dapat berkonsentrasi pada content logika bisnis. Ini akan memudahkan perubahan presentasi tanpa harus mempengaruhi logika bisnis.

JSP merupakan perluasan dari servlet yang ketika dijalankan JSP maka akan dikompilasi menjadi servlet terlebih dahulu sehingga fungsinya sama seperti servlet, yaitu menerima request dari client dan memberikan response dinamis kepada client. Response dinamis dapat berupa halaman HTML, DHTML, XHTML, atau XML. Protokol yang dipakai untuk mengakses halaman JSP adalah HTTP. Prosesnya sama seperti kita mengakses sebuah halaman HTML. [10]



Mahmoud, Q.H. Servlets and JSP Pages Best Practices March 2003

[http://java.sun.com/developer/technicalArticles/javaserverpages/servlets\\_jsp/](http://java.sun.com/developer/technicalArticles/javaserverpages/servlets_jsp/) Akses terakhir 09 Desember 2009

**Gambar 2. 9 Alir request/Response flow pada pemanggilan halaman JSP**

Pada prinsipnya pemakaian JSP mirip seperti pemakaian servlet. Secara garis besar JSP dipakai sebagai berikut:[10]

- Client mengirimkan request HTTP kepada JSP-container (atau disebut juga JSP-engine).
- JSP-container menentukan class yang mengimplementasikan halaman JSP yang dituju oleh request. (Class ini disebut sebagai *JSP Page implementation class*).
- JSP-container kemudian memanggil salah satu *method* dari class implementasi tersebut untuk menangani request secara dinamis dan menghasilkan response berupa content halaman HTML.
- Output halaman HTML diserahkan kepada JSP-container untuk dikirimkan sebagai response kepada client.

## BAB 3 PERANCANGAN

### 3.1 Skenario

Suatu perpustakaan terdiri atas beberapa rak. Sistem mengotomasi masuknya buku ke dalam rak dengan menggunakan teknologi RFID. Ketika suatu buku ditaruh di dalam rak, user akan mendapat notifikasi bahwa ada buku yang masuk. Dalam notifikasi selain memberitahu nama buku juga memberitahu apakah buku tersebut diletakkan pada rak yang benar. Kemudian dilakukan pencatatan log secara otomatis mengenai buku yang masuk dengan data berupa waktu masuk, judul buku, rak masuk, dan benar atau tidaknya penempatan itu.

Pada setiap rak ditempatkan pembaca RFID yang mengalirkan data hasil pembacaan ke server. Data diterima di server untuk dilakukan pengecekan dan pencatatan kemudian menotifikasi user. Program penerima data tersebut di-embed ke dalam suatu website sederhana. Website tersebut mampu untuk melihat data-data seperti buku, judul buku, penerbit, tag, rak, dan status. Selain melihat data juga dimungkinkan untuk mengubah serta menghapus data-data tersebut. Teknologi yang digunakan pada sisi server adalah menggunakan Java.

Kebutuhan yang diharapkan pada sisi perangkat keras mempunyai spesifikasi sebagai berikut:

- a. **Basis teknologi pengenalan:** modul pengenalan identitas buku berbasiskan pada RFID.
- b. **Pengenalan buku:** merupakan tag RFID yang ditanam atau ditempelkan di dalam buku. Ukuran tag seperti pada kartu kredit biasa dan tipis.
- c. **Real time:** pembaca akan selalu aktif bersiap-siap apabila ada buku yang masuk dalam rak buku.
- d. **Jumlah:** modul pembaca akan berjumlah banyak. Jumlahnya bergantung pada area jarak baca dan banyaknya buku yang dibaca nanti

- e. **Jaringan pembaca:** modul-modul pembaca berada dalam sebuah jaringan yang berujung pada suatu simpul dimana data yang telah dibaca akan dikirimkan ke sana.
- f. **Penempatan:** modul-modul pembaca akan ditempatkan pada masing-masing rak buku.

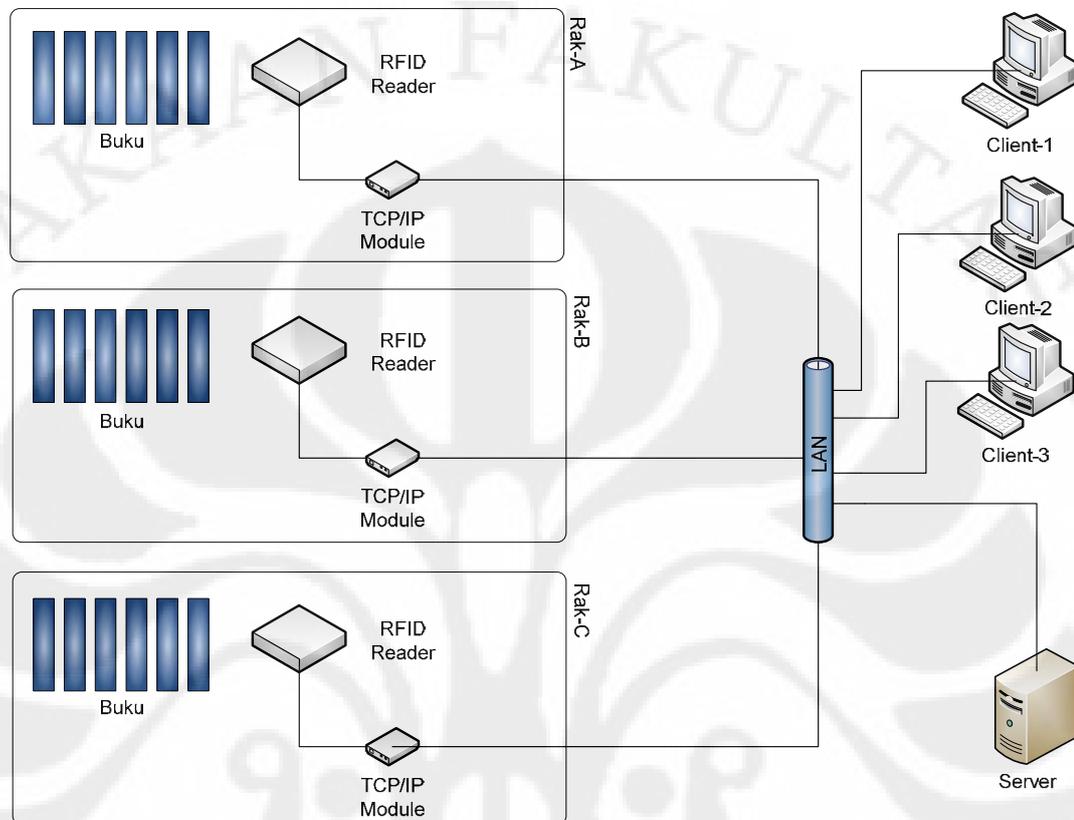
Sedangkan pada sisi perangkat lunak, kebutuhan yang diharapkan adalah:

- a. **Basis teknologi:** menggunakan Java sebagai basis teknologinya.
- b. **Real time:** harus selalu tanggap seandainya terdapat kedatangan data tiba-tiba dari salah satu modul pembaca.
- c. **Notifikasi:** kedatangan data diberitahukan kepada user dalam bentuk notifikasi atau disimpan dalam bentuk log.
- d. **Basis:** aplikasi berbasis web dengan tidak mewajibkan teknologi khusus yang dipakai dalam web tersebut.
- e. **Sistem informasi:** mampu menyimpan, mengedit, menghapus, dan menyajikan informasi-informasi dalam sistem perpustakaan kepada pengguna sistem.
- f. **Penempatan:** sisi perangkat lunak berada dalam sebuah server yang berada dalam jaringan lokal perpustakaan.

### 3.2 Desain

Perancangan sistem dilakukan dari deskripsi kebutuhan yang dijelaskan pada bagian skenario. Hal utamanya adalah sisi perangkat keras menggunakan teknologi RFID sedangkan sisi perangkat lunak menggunakan Java. Pembaca RFID berada pada tiap rak yang berarti akan ada banyak pembaca RFID dalam satu perpustakaan. Perangkat lunak ditempatkan di server. Teknologi yang digunakan di server adalah Java. Selain itu juga terdapat website sederhana. Program pembaca data yang masuk di-*embed* ke dalam website.

Dari deskripsi tadi maka diajukan desain sistem ini sebagaimana arsitekturnya digambarkan pada Gambar 3.1.



**Gambar 3. 1 Skema rancangan sistem**

Dikarenakan jumlah pembaca RFID yang banyak, maka sistem tidak bisa menggunakan protokol komunikasi serial untuk transfer data. Ini disebabkan karena jumlah port serial di server yang jumlahnya terbatas yang tidak mampu menangani jumlah pembaca RFID yang berjumlah belasan atau puluhan. Selain itu juga akan sulit untuk mengidentifikasi pembaca RFID. Oleh karena itu digunakan protokol TCP, sehingga sistem ini dikatakan berbasis IP. Alamat IP digunakan sebagai identifikasi pembaca-RFID. Identifikasi nama rak berdasarkan alamat IP dilihat di database yang ada di server.

Ditambahkan modul pengubah dari protokol serial ke protokol TCP pada setiap pembaca RFID. Kemudian modul tersebut terhubung ke jaringan lokal. Server juga terhubung ke jaringan tersebut. Dapat dikatakan bahwa jaringan tersebut adalah jaringan lokal perpustakaan. Client-1 sampai Client-3 yang tergambar pada

**Universitas Indonesia**

Gambar 3.1 adalah komputer-komputer lain yang berada pada jaringan lokal tersebut. Walaupun berada pada jaringan yang sama, data RFID tidak akan diteruskan ke mereka. Website bisa dibuka melalui mereka tapi hanya sebatas melihat data dan aktivitas lainnya, bukan sebagai server pembaca data.

Pada setiap buku ditempelkan tag RFID sebagai tanda pengenal buku. Dari tag ini diambil data untuk kemudian diteruskan ke server apabila buku tersebut berada dalam area baca pembaca RFID. Setiap rak bisa terdiri atas satu atau lebih pembaca RFID, bergantung pada kemampuan baca alat tersebut.

Unit perangkat lunak penerima data RFID menjadi satu dengan halaman web. Walaupun sama-sama menggunakan teknologi Java, unit penerima data dan website dikembangkan secara berbeda. Hal ini dilakukan dengan pertimbangan sebagaimana dijelaskan nanti. Unit penerima data berupa applet sedangkan website menggunakan JSP.

### 3.3 Perancangan Perangkat Keras

Sisi perangkat keras sistem terdiri atas tiga unit yaitu tag RFID, pembaca RFID dan modul TCP/IP. Susunannya seperti yang digambarkan di skema desain sistem pada Gambar 3.1 dengan tag ditempel ke buku, pembaca RFID dan modul TCP/IP saling terhubung.

#### 3.2.1 Peralatan



**Gambar 3. 2 Tag RFID Pasif Class-1 berbentuk kartu**

Tag berfungsi sebagai pengenal buku, dengan kata lain sebagai nomor identifikasi buku. Jenis tag yang digunakan adalah tag pasif Kelas-1. Tag ini mendapatkan sumber daya dari pembaca kemudian mentransmisikan nomor identifikasinya. Ukuran tag yang digunakan berukuran seperti kartu kredit biasa. Sebenarnya lebih

cocok menggunakan tag berupa label. Namun pemesanan tag berupa label ini harus dalam jumlah yang banyak.



**Gambar 3. 3 RFID Starter-Kit**

Bagian yang kedua adalah pembaca RFID. Di sini divais pembaca yang digunakan adalah RFID Starter-Kit yang berbasis reader tipe ID-12. Pembaca ini mampu membaca tag RFID pasif hingga jarak 12 centimeter. Memang tidak cocok apabila dengan jarak yang pendek itu dipakai untuk aplikasi perpustakaan ini. Namun karena pengerjaan ini sifatnya menghasilkan prototipe sistem, maka sebagai permulaan digunakan pembaca ini. Antarmuka keluaran RFID Starter-Kit adalah port RJ11 dengan format data UART RS-232. Selain format data tersebut dapat juga diatur menjadi protokol UART TTL, Wiegand26, dan Magnet Emulation. Dalam tugas akhir ini dipakai format data keluaran UART RS-232 juga untuk kompatibilitas modul berikutnya.



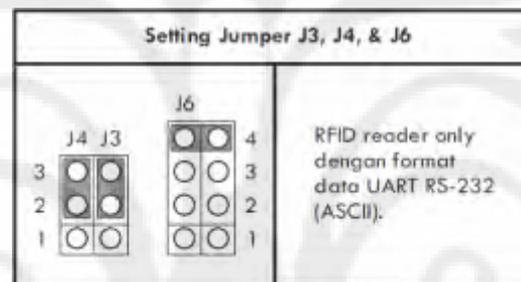
**Gambar 3. 4 TCP/IP Module (WIZ110SR)**

Modul terakhir yang digunakan dalam sisi perangkat keras adalah WIZ110SR. Divais ini dipasang setelah pembaca RFID dan ditempatkan juga dalam rak buku. Guna alat ini adalah mengubah data yang tadinya dalam bentuk serial RS-232 ke dalam bentuk TCP/IP. Dari sekian banyak protokol yang didukung divais ini, yang dipilih adalah protokol TCP. Protokol TCP lebih mementingkan integritas daripada kecepatan penyampaian data. Dengan kata lain protokol ini menjaga data

yang terkirim agar tidak rusak. Karena data yang dikirimkan modul pembaca tidak banyak, bahkan dapat dibuat dalam satu paket saja. Maka protokol ini lebih cocok digunakan pada transmisi data dalam jaringan untuk sistem aplikasi perpustakaan yang dibuat.

### 3.2.2 Konfigurasi

Pembaca RFID dan WIZ110SR diatur sedemikian rupa sehingga data hasil pembacaan dapat dikirimkan melalui jaringan TCP/IP. Data tersebut harus sampai dan terdeteksi oleh server.



**Gambar 3. 5 Setting Jumper Pada RFID Starter-Kit**

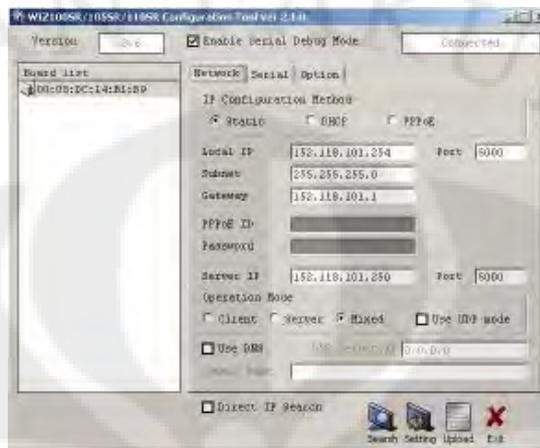
Modul yang pertama kali dikonfigurasi adalah pembaca RFID. Pada modul ini tidak banyak dilakukan konfigurasi. Hanya memastikan bahwa format data keluarannya adalah UART RS-232. Penentuan format ini dimaksudkan agar pada modul TCP/IP data dapat diterima dan dibaca dengan baik.

Kemudian dilakukan konfigurasi pada WIZ110SR. Konfigurasi yang dilakukan di divais ini adalah penentuan alamat IP, alamat IP server, nomor port, jenis protokol yang dipakai, lebar paket, dan konfigurasi sambungan serialnya.

Konfigurasi dilakukan dengan menyambungkan WIZ110SR ke komputer dengan kabel UTP *cross*. Konfigurasi pertama dilakukan dengan menentukan alamat IP divais ini beserta atributnya. Alamat IP dibuat statik, agar memudahkan dokumentasi terutama pada database bahwa alamat IP sekian menunjukkan rak buku yang mana. Nomor port dapat bebas kita tentukan asal tidak *overlap* dengan *well-known ports* yang telah ditentukan IANA. Dari nomor port ini kita set pada

**Universitas Indonesia**

server nantinya sehingga dapat dilakukan koneksi. Mode operasi dipilih Mixed dan tidak mencentang UDP Mode. Secara default WIZ110SR akan menggunakan protokol TCP. Konfigurasi ini dapat dilihat di Gambar 3.6.



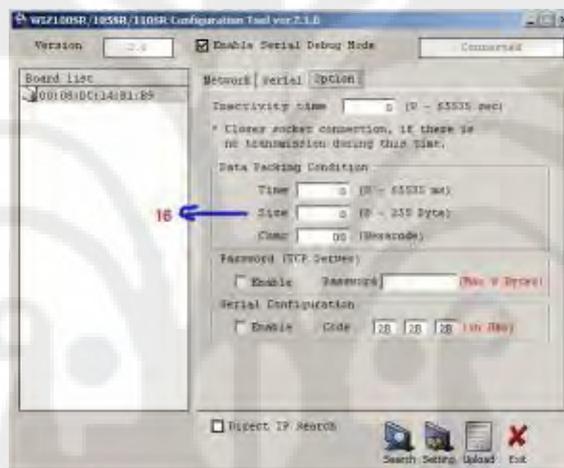
**Gambar 3. 6 Konfigurasi Network Pada WIZ110SR**

Berikutnya adalah menentukan parameter serial untuk antarmuka WIZ110SR. Penentuan ini harus disesuaikan dengan parameter serial yang dimiliki oleh pembaca RFID. Jika bernilai beda maka koneksi data tidak akan sinkron di kedua sisi. Karena pembaca-RFID yang dipakai sudah mempunyai parameter tetap maka kita samakan nilai tersebut pada WIZ110SR. Parameter serial yang dipakai adalah speed: **9600** bps, databit: **8**, parity: **none**, stop bits: **1**, flow control: **none**. Konfigurasi ini dapat dilihat pada Gambar 3.7.



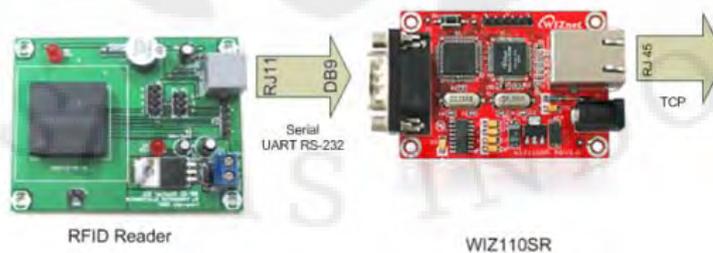
**Gambar 3. 7 Konfigurasi Serial Pada WIZ110SR**

Konfigurasi terakhir pada WIZ110SR adalah menentukan atribut-atribut paket yang akan dikirimkan nanti. Secara default paket yang dikirimkan divais ini mempunyai lebar data 1 byte. Seberapa banyaknya data yang dikirim oleh divais serial akan selalu dikirimkan satu paket per satu byte data. Karena pembaca RFID mengirimkan data sepanjang 16 byte maka pada pengujian didapatkan terdapat 16 paket yang dikirimkan. Terpecahnya data hingga 16 ini beresiko terutama bila ada satu paket yang rusak atau tidak sampai ke server. Sehingga ukuran paket ditentukan sebesar 16 byte dan divais ini mengirim 1 paket per 16 byte. Untuk atribut lain diabaikan karena tidak diperlukan. Untuk mengetahui fungsinya masing-masing dapat mengacu pada buku manual WIZ110SR.



**Gambar 3. 8 Konfigurasi Opsional Pada WIZ110SR**

Setelah semua parameter konfigurasi selesai ditentukan, semuanya di unggah ke MCU dari WIZ110SR. Pada program konfigurasi dilakukan dengan menekan tombol "setting". Untuk mengujinya dapat dilakukan "ping" ke alamat divais ini atau juga menangkap paket data yang dikirimkannya.



**Gambar 3. 9 Skema Modul Unit Pembaca**

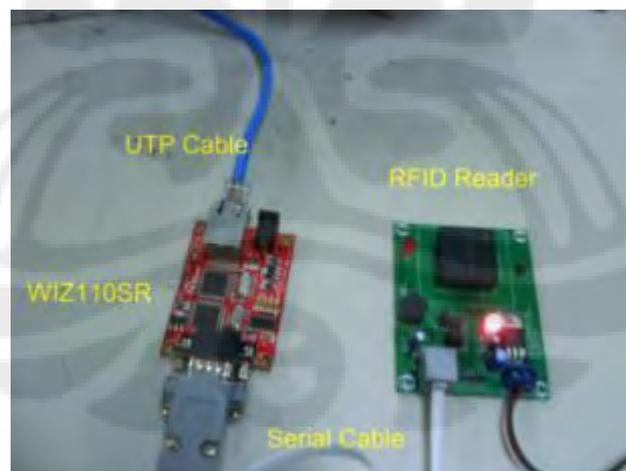
Universitas Indonesia

Konfigurasi berikutnya adalah menghubungkan pembaca RFID ke WIZ110SR. Dari pembaca RFID dihubungkan dengan port RJ11 dan sampai ke WIZ110SR dengan antarmuka DB9. Dari WIZ110SR dihubungkan ke jaringan lokal dengan kabel UTP *straight* menuju *switch*. Pengujian ketersambungan dapat dilakukan dengan program penangkap paket seperti *wireshark* dan membandingkan data paket dengan nomor tag-RFID.



**Gambar 3. 10 Penempatan Tag RFID pada Buku**

Posisi tag-RFID yang akan dipasang pada setiap buku di perpustakaan. Tag yang dipakai sementara berukuran seperti kartu walaupun sebenarnya lebih cocok berbentuk label karena lebih elastis.



**Gambar 3. 11 Skema Unit Pembaca RFID**

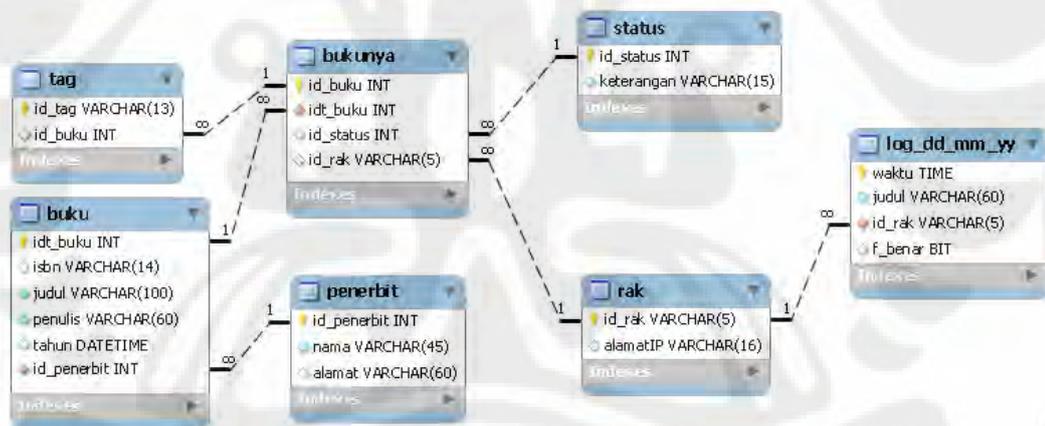
Susunan setiap perangkat pembaca, terdiri atas pembaca RFID yang dihubungkan dengan WIZ110SR melalui kabel serial. Kemudian dihubungkan ke jaringan lokal dengan menggunakan kabel UTP biasa.

### 3.4 Perancangan Perangkat Lunak

Sisi perangkat lunak, sebagaimana pada sisi perangkat keras terdiri atas tiga bagian. Bagian pertama adalah database dengan menggunakan teknologi MySQL. Berikutnya adalah applet yang dikembangkan dengan Java. Terakhir adalah website yang dibuat dengan menggunakan JSP, suatu teknologi web yang merupakan turunan dari Java.

#### 3.4.1 Database

Database terdiri atas enam tabel utama yaitu bukunya, buku, penerbit, tag, status, dan rak ditambah tabel log yang jumlahnya bertambah. Masing-masing tabel mempunyai relasi antar tabel seperti ditunjukkan pada Gambar 3.12 dimana relasinya berbentuk *one-to-many*. Database menggunakan *engine* InnoDB yang mempunyai fitur hubungan relasi antar tabel.



**Gambar 3. 12 Diagram relasi database**

Tabel “buku” berisikan data-data buku seperti ISBN, judul buku, penulis, penerbit, serta tahun terbitnya. Primary key berupa *integer* yang dibuat auto-increment berguna sebagai referensi oleh tabel lain. Dalam tabel ini penerbit direferensikan ke tabel penerbit karena pertimbangan bahwa beberapa buku bisa mempunyai satu penerbit yang sama. Dalam tabel “penerbit” sendiri terdiri atas nama id\_penerbit sebagai primary key kemudian nama penerbit dan alamatnya.

Tabel “bukunya” merupakan tabel yang merepresentasikan satu unit buku yang ada di perpustakaan. Perbedaan dengan tabel “buku” adalah pada tabel tersebut merupakan representasi dari satu data buku bukan satu unit buku. Dalam perpustakaan kemungkinan bahwa beberapa buku mempunyai judul, penulis, dan penerbit yang sama cukup besar. Jika membuat beberapa buku itu didata pada tabel “buku” akan kurang efisien. Oleh karena itu buku-buku tersebut didata pada tabel “bukunya”. Tabel ini berisikan data berupa `idt_buku` sebagai sebagai pengenalan judul, penulis dan penerbit, status, dan rak dimana buku tersebut disimpan. Semuanya mereferensi pada tabel-tabel lain.

Tabel “tag” berisikan ID tag yaitu data tag yang berjumlah 12 dari 16 karakter yang ada pada setiap tag RFID. Karakter-karakter tersebut menunjukkan ID dari tag, sedangkan karakter lain sebagai penunjuk bahwa data tersebut adalah data RFID.

Tabel “status” hanya berisi nama status saja seperti ada, dipinjam, atau rusak. Tabel “rak” sendiri berisi ID rak yang menunjukkan nama rak pada perpustakaan dan alamat IP yang diambil dari alamat IP pembaca RFID. Data yang diterima oleh applet akan merujuk pada tabel ini untuk menentukan bahwa dimana rak yang dimasuki oleh buku.

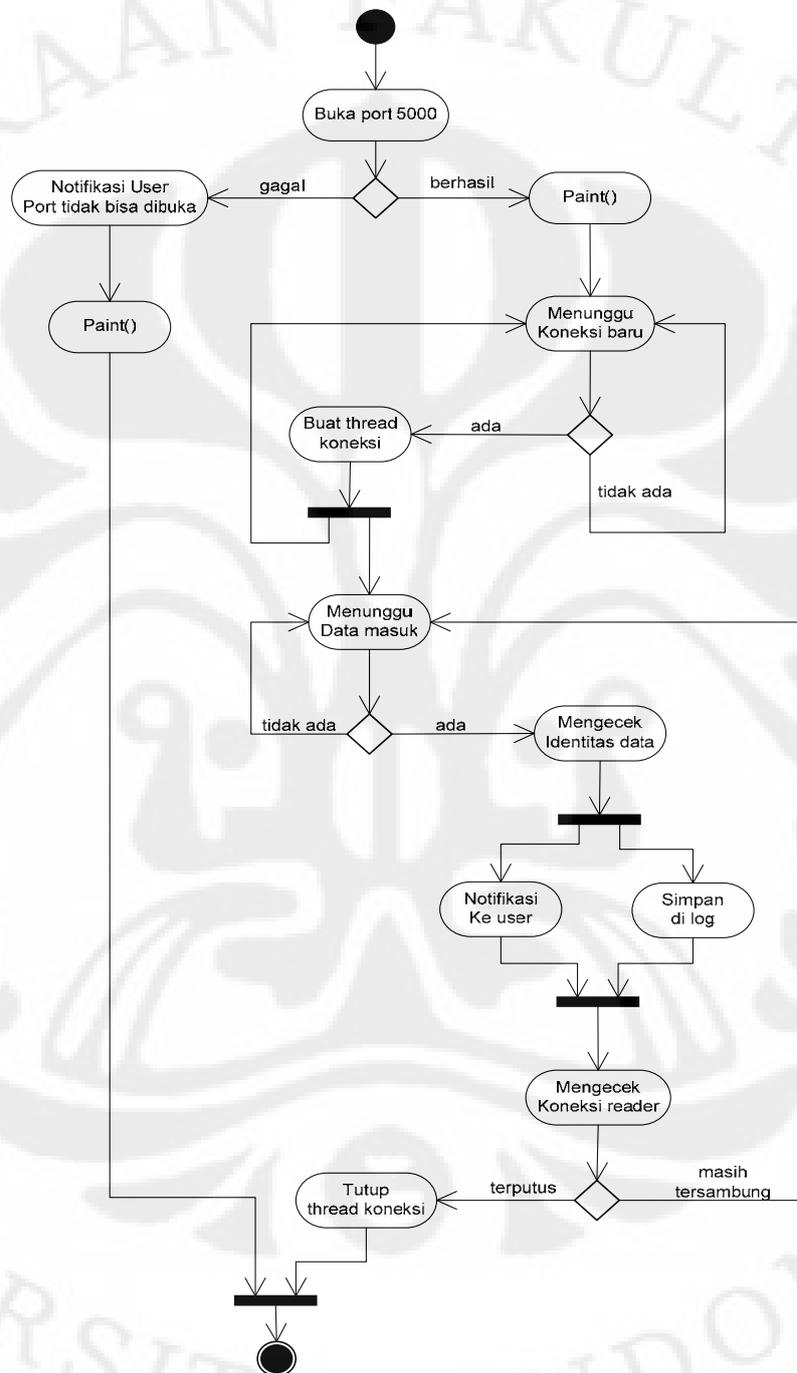
Tabel “log” adalah tabel yang berisi catatan buku-buku yang telah masuk. Tabel ini terdiri atas waktu masuk, judul buku masuk, rak dimana buku tersebut, dan flag yang menunjukkan apakah buku tersebut telah masuk ke rak yang benar atau belum. Setiap hari tabel log akan dibuat baru. Skema tabel yang digambar pada Gambar 3.12 hanyalah representasi tabel log tersebut.

#### 3.4.2 Applet

Applet merupakan bagian sistem di sisi perangkat lunak yang berfungsi sebagai Socket Server yang menerima sambungan dari pembaca RFID, pengecekan data RFID, penyimpanan dalam log, serta notifikasi ke user bila ada buku yang masuk.

Alasan penggunaan applet adalah bahwa jika unit penerima data RFID benar-benar disatukan dengan web, maka web tersebut tidak akan selesai me-load

halaman dimana *script* itu berada. Oleh karena itu dibutuhkan unit yang berjalan secara paralel dimana halaman web tidak perlu menunggu proses pada unit penerima data selesai.



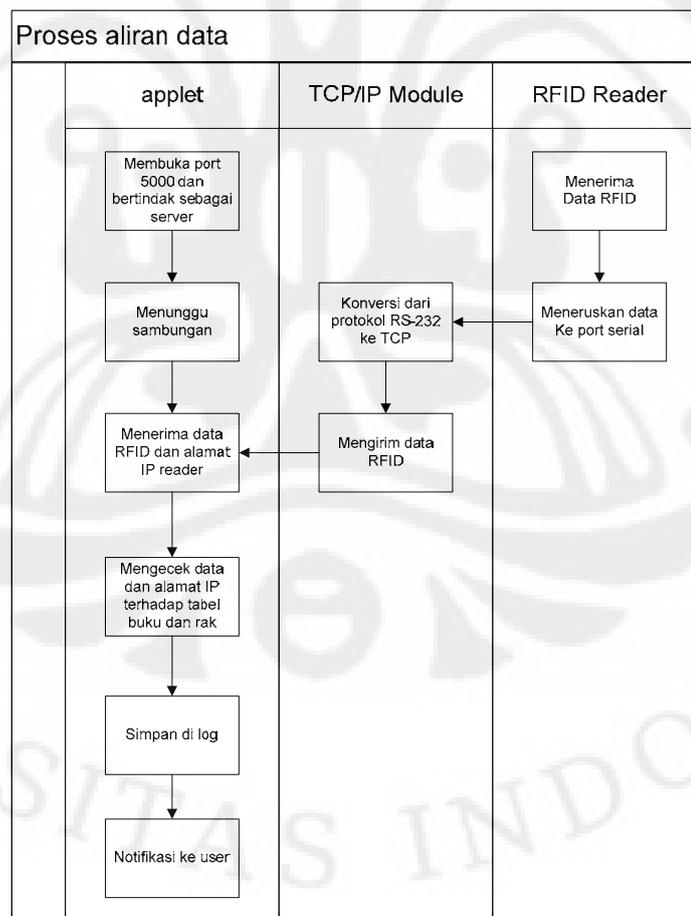
**Gambar 3. 13 Activity diagram applet**

Pertama-tama applet membuka port 5000 di server, port default dari modul TCP/IP. Kemudian applet berlaku sebagai socket server dengan nomor port itu.

**Universitas Indonesia**

Pembukaan port ini dilakukan pada bagian inisialisasi sehingga tidak perlu dilakukan berulang-ulang. Jika port berhasil dibuka, maka applet memberitahu user dan memberitahu juga jika gagal. Jika gagal maka applet akan selesai dan tidak berfungsi walaupun ada data yang masuk. Jika berhasil maka applet akan melakukan fungsi-fungsi berikutnya.

Ketika method `paint()` sudah dijalankan maka applet akan menunggu koneksi yang datang dari unit pembaca RFID. Koneksi akan ditunggu terus sampai masuk sambungan baru. Ketika ada koneksi baru maka dibuat thread koneksi baru sekaligus dibuat lagi thread untuk menunggu apabila ada koneksi baru yang datang. Setiap sambungan dari satu unit pembaca RFID ditangani oleh satu thread koneksi di dalam applet. Hal ini bertujuan agar data dari pembaca RFID yang berbeda dapat diterima secara bersamaan tanpa harus menunggu sambungan dari pembaca lain selesai dahulu.



**Gambar 3. 14 Diagram aliran proses data dari RFID reader hingga applet**

Universitas Indonesia

Thread akan menunggu terus sampai ada data masuk. Jika ada data masuk maka data beserta alamat IP pengirim data tersebut dicek untuk mendapatkan judul buku, id buku, rak dimana buku itu masuk, dan apakah buku tersebut telah masuk ke rak yang benar. Kemudian data tersebut disimpan dalam log, baik di database maupun dalam bentuk file. Selain itu juga user akan mendapat notifikasi mengenai masuknya buku tersebut.

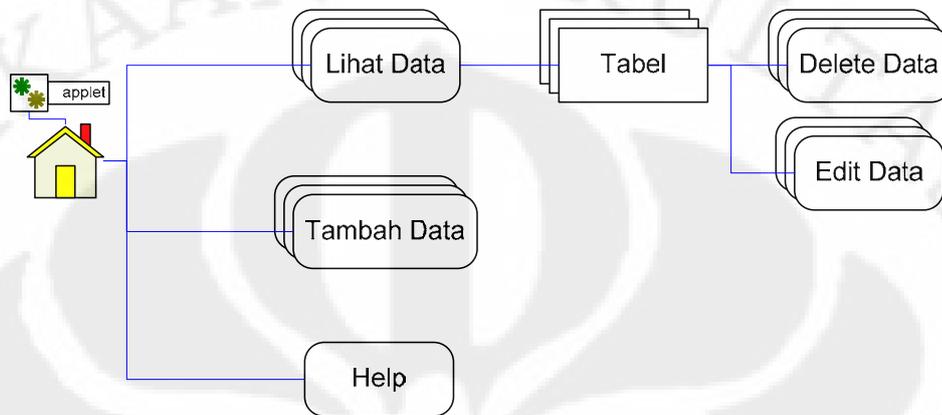
Koneksi akan di cek apakah masih ada atau tidak. Jika masih ada maka proses akan kembali menunggu masuknya data. Jika sudah terputus atau ada sesuatu hal lain maka koneksi akan ditutup kemudian thread koneksi itu akan diberhentikan. Alur logika applet tergambar pada *activity diagram* pada Gambar 3.13 sedangkan bagaimana aliran data mulai dari pembaca RFID hingga applet digambarkan pada Gambar 3.14.

Applet secara default dibatasi oleh banyak hal untuk keamanannya. Beberapa pembatasan yang berkenaan dengan desain sistem ini adalah tidak bolehnya applet mengadakan koneksi socket pada port tertentu dan alamat-alamat biasanya. Selain itu applet juga tidak boleh mengakses atau membuat file. Oleh karena itu, *policy* yang dikenakan oleh JRE ke applet harus kita tambahkan dahulu. Dalam implementasi didaftarkan alamat-alamat IP *reader* yang diperbolehkan mengadakan koneksi socket ke applet beserta nomor port aksesnya. Untuk modifikasi file, diberikan applet diberikan hak untuk melakukan akses apapun terhadap file.

Alasan pengenaan batasan-batasan oleh Java terhadap applet adalah agar applet tidak sembarang digunakan untuk memodifikasi komputer dimana applet tersebut dijalankan. Mengingat bahwa applet sebenarnya adalah layaknya program Java biasa yang berjalan dengan bantuan *browser*. Walaupun pemberian akses khusus kepada applet untuk mengadakan koneksi socket dan akses file berarti membuat celah keamanan, namun untuk dalam desain applet ini tetap dilaksanakan. Hal ini karena penulis sendiri yang memprogram isi applet tersebut dan penulis

bertanggung jawab apabila ada kode dalam applet tersebut yang membahayakan komputer atau server dimana applet tersebut dijalankan.

### 3.4.3 Halaman Web



**Gambar 3. 15 Diagram konsep web**

Website dibuat dengan menggunakan teknologi JSP dan dibuat secara sederhana. Applet di-embed hanya pada halaman utama website ini untuk mencegah tumpang tindih socket server penerima data RFID yang difungsikan applet. Website ini mempunyai fungsi untuk melihat menambah data, mengedit dan menghapus data-data yang ada pada tabel-tabel database kecuali tabel log. Logika dasar masing-masing halaman adalah halaman JSP dengan menggunakan bean. Jadi pada setiap halaman terdapat *class* Java yang menangani request-request pada halaman tersebut. Diagram konsep website digambarkan pada Gambar 3.15.

## BAB 4 PENGUJIAN DAN EVALUASI

### 4.1 Pengujian Fungsional Sistem

Pengujian fungsional sistem bertujuan untuk menguji apakah fungsi-fungsi dari sistem bekerja sesuai dengan desain dari sistem ini. Parameter yang dijadikan acuan adalah bisa atau tidaknya sistem menangani permintaan atau memberikan hasil yang sesuai. Sehingga hal yang lebih detail tidak termasuk dalam pengujian ini. Pengujian ini terdiri atas dua bagian yaitu pengujian pada perangkat keras dan pengujian pada perangkat lunak.

#### 4.1.1 Pengujian Fungsional Perangkat Keras

Pengujian fungsional perangkat keras dilakukan untuk menjawab pertanyaan-pertanyaan berikut:

1. Apakah pembaca-RFID dapat membaca tag?
2. Apakah Modul Converter berfungsi dengan baik?
3. Apakah data RFID berhasil dikirim melalui jaringan?
4. Sampaiakah data RFID tersebut ke server?

Sehingga pengujian fungsional perangkat keras ini dilakukan mulai dari pembacaan tag RFID hingga sampainya paket data ke server. Hasil pengujian didapat dari memonitor modul-modul perangkat keras melalui server pada jaringan lokal. Proses monitor ini tentunya menggunakan bantuan perangkat lunak tapi bukan merupakan sisi perangkat lunak sistem. Perangkat lunak yang dipakai adalah:

1. **Wireshark 0.99.8** untuk menangkap paket yang masuk ke network interface server.
2. **HyperTerminal** untuk mendapatkan bentuk data RFID melalui jaringan dengan mengadakan koneksi Soket ke modul converter.

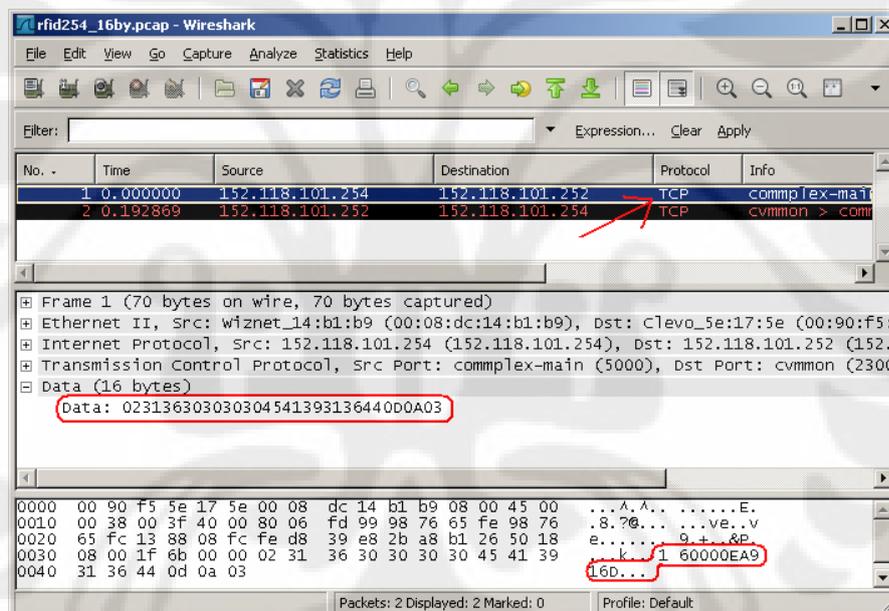
Tapi sebelum itu perangkat keras terutama modul converter haruslah terlebih dahulu dikonfigurasi seperti yang telah dijelaskan pada bagian perancangan di bab sebelumnya.

Pembacaan tag-RFID oleh pembaca-RFID dilakukan dengan berhasil. Ketika tag didekatkan pembaca memberikan tanda dengan membunyikan buzzer yang ada di pembaca. Selain itu data yang terbaca dari tag diteruskan ke keluaran pembaca dimana interface-nya menggunakan protokol RS-232. Keluaran tidak sempat dimonitor dari port tersebut melainkan dilihat setelah data tersebut dilewatkan oleh modul converter. Contoh data yang terbaca di HyperTerminal adalah sebagai berikut:

●160000EA916D

▼

Karena pembacaan ini sekaligus dengan menggunakan modul converter maka sekarang kita lihat bagaimana bentuk paket data yang diterima dengan Wireshark:



**Gambar 4. 1** Paket data terbaca oleh Wireshark

Data yang terbaca oleh Wireshark sama dengan data yang diterima pada HyperTerminal. Dengan begitu hasil pengujian sisi perangkat keras disajikan dalam Tabel 4.1. Pada tabel tersebut terlihat bahwa fungsi-fungsi pada bagian perangkat keras bekerja dengan baik.

Tabel 4. 1 Hasil pengujian fungsional sisi perangkat keras

Deskripsi	Keterangan
Reader membaca tag	Ya
Modul converter RS232-TCP/IP berfungsi	Ya
Paket TCP berisi data RFID berhasil dikirim melalui jaringan	Ya
Data RFID terbaca dengan baik di server	Ya

#### 4.1.2 Pengujian Fungsional Perangkat Lunak

Pengujian fungsional sisi perangkat lunak berguna untuk mengetahui apakah fitur-fitur yang dirancang telah berfungsi. Prosedur yang dilakukan dalam pengujian berusaha menjawab pertanyaan-pertanyaan sebagai berikut:

1. Apakah rancangan database bekerja sesuai dengan hubungan relasinya?
2. Apakah fungsi-fungsi penampilan dan modifikasi pada website bekerja dengan baik?
3. Apakah website me-load applet?
4. Apakah applet memberikan notifikasi awal?
5. Sesuainkah notifikasi awal yang diberikan?
6. Jika berhasil benarkah port 5000 yang terbuka?
7. Berhasilkah koneksi socket terbentuk dari *reader* dan applet?
8. Apakah koneksi dengan lebih dari satu reader dimungkinkan?
9. Samakah data RFID yang dikirim *reader* dengan yang diterima applet?
10. Apakah pengecekan data terhadap database berhasil?
11. Tersimpankah data buku yang masuk tersebut ke dalam log?
12. Apakah user diberi notifikasi?

Pengecekan fungsi database dilakukan dengan cara mengutak-atik data-data dalam tabel yang mempunyai referensi dengan tabel lainnya. Kebanyakan dari hubungan relasi tabel adalah *one-to-many*. Modifikasi-modifikasi yang dilakukan untuk menguji dilakukan dengan bantuan *phpmyadmin*, suatu alat bantu administrasi database yang berbasis web.

Salah satu pengujian adalah dengan menghapus satu *entry* di tabel “buku” dimana mempunyai *constraint* ON DELETE CASCADE ON UPDATE CASCADE. Seharusnya ketika *entry* pada tabel “buku” dihapus, maka semua *entry* yang mereferensi pada tabel “bukunya” akan terhapus mengingat tipe *constraint* antar keduanya. Dan hasilnya adalah semua *entry* pada tabel “bukunya” yang mereferensi *entry* yang dihapus pada tabel “buku” terhapus juga. Berarti pengujian relasi database memberikan hasil yang sesuai dengan rancangan.

Fitur modifikasi tabel-tabel yang dilakukan dengan website juga berjalan dengan baik. Kecuali tabel log, semua tabel dapat dilihat, ditambahkan, diubah, maupun dihapus data-datanya. Untuk hasil pengujian yang lebih detail mengenai halaman web dapat dilihat pada sub-bab 4.5. Applet yang diletakkan pada halaman utama website juga dapat berjalan dengan baik dengan syarat tertentu. Syarat tersebut adalah browser dimana applet dijalankan mempunyai *plugin* Java yang memungkinkan browser menjalankan applet.

Ketika applet pertama kali dijalankan ia akan memberikan notifikasi kepada user apakah applet tersebut berhasil bertindak sebagai socket server pada port 5000 atau tidak. Jika berhasil maka akan muncul notifikasi seperti pada Gambar 4.2. Jika gagal maka notifikasi yang muncul seperti Gambar 4.3. Apabila berhasil maka applet akan berjalan sebagaimana harusnya, sedangkan apabila gagal maka applet tidak berfungsi sama sekali. Kegagalan applet pada bagian ini dikarenakan ada socket-server lain yang menggunakan port tersebut bisa karena applet yang sama ataupun program Java lain. Mengenai apakah program selain Java yang berfungsi sebagai socket server pada port 5000 mengganggu applet ini belum pernah diuji pengaruhnya.



**Gambar 4. 2 Notifikasi awal ketika applet berhasil dijalankan**

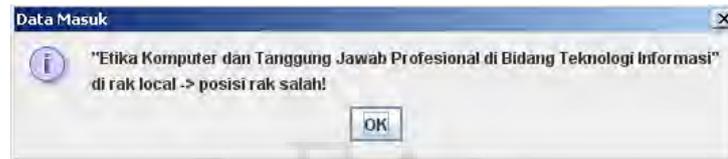


**Gambar 4.3** Notifikasi awal ketika applet gagal dijalankan

Untuk menguji apakah benar-benar port 5000 yang dibuka applet dijalankan program Java sederhana yang berfungsi sebagai socket client yang meminta koneksi pada port tersebut. Jika berhasil maka applet akan menerima data yang dikirim program tersebut, namun jika gagal maka program tersebut akan otomatis tertutup. Dari pengujian didapatkan bahwa antara program tersebut dan applet terjadi koneksi. Data yang dikirim program sama dengan data yang diterima applet. Dengan begitu maka applet memang benar-benar membuka port 5000 dan berfungsi sebagai socket-server pada port tersebut.

Koneksi socket dari pembaca RFID ke applet dicek dengan cara yang mirip dengan pengujian pengiriman data pada sisi perangkat keras. Bedanya, alih-alih menggunakan HyperTerminal, pada pengujian ini digunakan applet itu sendiri. Jika berhasil tersambung maka applet akan mengeluarkan pernyataan pada *console* Java yang memberitahukan bahwa koneksi berhasil. Koneksi dengan lebih dari satu sambungan juga berhasil dilakukan. Hal ini terlihat dari pengujian dengan dua pembaca RFID yang membaca tag secara bersamaan.

Untuk mengecek kesamaan data jika sebelumnya dilakukan dengan menggunakan Wireshark maka disini menggunakan applet juga. Dari hasil penerimaan data dilihat bahwa data yang dikirim pembaca RFID sama dengan data yang diterima applet. Hanya saja karakter yang ditampilkan Java agak berbeda, akan tetapi karena byte karakter tersebut tidak diperhatikan maka pengujian ini memperoleh hasil yang sesuai.



**Gambar 4. 4 Contoh notifikasi ketika ada data buku berisi tag rfid masuk**

Untuk mengetahui apakah dari data RFID tersebut diperoleh data buku yang sesuai dilihat dari notifikasi yang diberikan applet. Jika berhasil maka judul buku yang ditampilkan sama dengan judul buku yang ada *entry* buku yang mempunyai id tag sama dengan data RFID tersebut. Dari hasil pengujian dilihat bahwa applet menotifikasi user judul buku yang sama dengan yang seharusnya. Keberhasilan pencatatan log di database dilihat dengan melihat *entry-entry* tabel log pada hari itu. Sedangkan untuk log berupa file dilihat dari file log yang terbuat. Detail pengujian notifikasi dan pencatatan log dapat dilihat pada sub-bab 4.4. Dengan begitu hasil pengujian sistem pada sisi perangkat lunak disajikan pada tabel 4.2. Pada tabel tersebut terlihat bahwa fungsi-fungsi yang diharapkan bekerja dengan baik.

**Tabel 4. 2 Hasil pengujian fungsional sisi perangkat lunak**

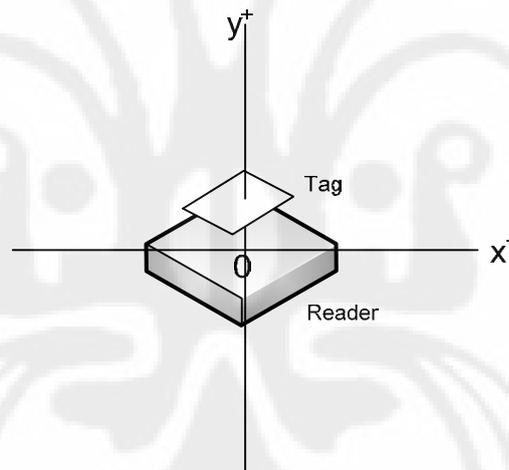
Deskripsi	Keterangan
Rancangan database bekerja sesuai dengan hubungan relasinya	Ya
Fungsi-fungsi penampilan dan modifikasi pada website bekerja dengan baik	Ya
Website me-load applet	Ya
Applet memberikan notifikasi awal	Ya
Notifikasi awal yang diberikan sesuai	Ya
Port 5000 yang terbuka	Ya
Koneksi socket terbentuk dari <i>reader</i> dan applet berhasil	Ya
Koneksi dengan lebih dari satu <i>reader</i> dimungkinkan	Ya
Data RFID yang dikirim <i>reader</i> dengan yang diterima applet sama	Ya
Pengecekan data terhadap database berhasil	Ya
Data buku yang masuk tersimpan ke dalam log	Ya
User diberi notifikasi	Ya

## 4.2 Pengujian dengan Arah Baca Tertentu

Pengujian ini ditujukan untuk menemukan jarak baca efektif dan perkiraan pola radiasi modul pembaca RFID. Pada sistem ini pembaca RFID yang digunakan adalah RFID Starter kit yang berbasis *reader* bertipe ID-12 yang mempunyai kemampuan baca tag pasif sejauh 12 cm.

Pengujian terbagi atas dua metode pembacaan dengan masing-masing melihat keberhasilan baca pada empat arah baca. Arah bacanya adalah  $y^+$ ,  $y^-$ ,  $x^+$ ,  $x^-$  dengan titik pusat berada di tengah-tengah modul pembaca RFID ID-12 yang berwarna hitam itu. Metode pembacaan yang diujikan adalah pembacaan dengan tag dibuat horizontal dan tag dibuat vertikal. Semuanya dilakukan pengambilan data sebanyak sepuluh pembacaan.

### 4.2.1 Pembacaan dengan tag horizontal



**Gambar 4. 5 Ilustrasi pembacaan tag secara horizontal**

Uji pembacaan tag horizontal adalah pengujian jarak baca dengan membuat tag sejajar dengan modul pembaca. Ilustrasi pengujian ini direpresentasikan pada Gambar 4.5. Pengujian dilakukan pada empat arah dan parameter yang dilihat adalah keberhasilan pembaca RFID membaca tag. Keberhasilan baca pembaca RFID ditunjukkan dengan bunyi yang dikeluarkan dan hasil penerimaan data pada HyperTerminal. Hasil dari pengujian ini disajikan pada Tabel 4.3, Tabel 4.4, Tabel 4.5 dan Tabel 4.6.

**Tabel 4. 3 Pembacaan tag horizontal pada arah y+**

Jarak reader vs tag (cm)	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
>12	-	-	-	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-	-	-
4	√	√	√	√	√	√	√	√	√	√
3	√	√	√	√	√	√	√	√	√	√
>3	√	√	√	√	√	√	√	√	√	√

**Tabel 4. 4 Pembacaan tag horizontal pada arah y-**

Jarak reader vs tag (cm)	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
>12	-	-	-	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-	-	-
4	√	√	√	√	-	√	√	√	√	√
3	√	-	√	√	√	√	√	√	√	√
>3	√	√	√	√	-	√	√	√	√	√

**Tabel 4. 5 Pembacaan tag horizontal pada arah x+**

Jarak reader vs tag (cm)	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
>12	-	-	-	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-	-	-
3	-	-	-	-	-	-	-	-	-	-
>3	√	√	-	√	√	√	-	√	√	√

**Tabel 4. 6 Pembacaan tag horizontal pada arah x-**

Jarak reader vs tag (cm)	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
>12	-	-	-	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-	-	-
3	-	-	-	-	-	-	-	-	-	-
>3	√	√	√	√	√	-	-	√	√	√

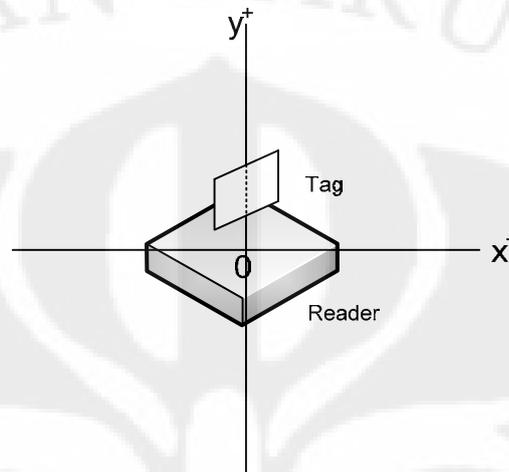
Pada Tabel 4.3 terlihat bahwa pembacaan tag dengan keberhasilan 100% dimulai pada jarak 4 cm. Untuk jarak baca lebih dari itu pembacaan sama sekali tidak berhasil. Untuk Tabel 4.4 yang menunjukkan hasil pembacaan tag horizontal pada arah y- terlihat bahwa karakteristik pembacaan sama dengan Tabel 4.3. Hanya saja terdapat beberapa kegagalan pembacaan. Kemudian baik Tabel 4.5 dan Tabel 4.6 menunjukkan tag berhasil dibaca pada jarak pembacaan kurang dari 3 cm. Walaupun demikian tetap terdapat beberapa tag yang tidak berhasil dibaca.

Hasil pengujian baca pada arah y+ menunjukkan bahwa tag terbaca pada jarak sekitar 4 cm. Untuk jarak diatas itu keberhasilan baca tidak ada sama sekali. Hasil yang sama juga ditunjukkan pada arah baca y-. Hanya saja pada pembacaan y- ini, terdapat beberapa kegagalan baca. Untuk arah x, baik x+ maupun x- memberikan hasil yang mirip. Keberhasilan baca pada arah x dimulai pada jarak kurang dari 3 cm. Untuk jarak lebih dari 3 cm tag tidak terbaca oleh *reader*. Jarak pembacaan yang ditunjukkan dari hasil pengujian memperlihatkan bahwa pembaca RFID tidak benar-benar membaca pada jarak 12 cm.

Uji pembacaan pada empat arah baca untuk tag dibuat sejajar dengan modul pembaca menunjukkan bahwa pembacaan pada arah y lebih jauh daripada pembacaan pada arah x. Namun demikian, walaupun berhasil dibaca keberhasilannya tidak seratus persen. Pada jangkauan jarak baca masih ada tag-tag

yang tidak terbaca. Hal ini diasumsikan bahwa pembaca RFID memerlukan waktu setelah pembacaan suatu tag agar bisa membaca tag yang baru lagi.

#### 4.2.2 Pembacaan dengan tag vertikal



**Gambar 4. 6 Ilustrasi pembacaan tag secara vertikal**

Uji pembacaan dengan tag vertikal adalah pengujian jarak baca dengan membuat tag tegak lurus terhadap modul pembaca RFID. Ilustrasi pengujian ini direpresentasikan pada Gambar 4.6. Sama dengan pengujian tag horizontal, parameter yang dilihat adalah keberhasilan baca pembaca RFID terhadap tag pada arah dan jarak tertentu. Hasil pengujian tag vertikal ini disajikan pada Tabel 4.7, Tabel 4.8, Tabel 4.9 dan Tabel 4.10.

**Tabel 4. 7 Pembacaan tag vertikal pada arah y+**

Jarak reader vs tag (cm)	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
>12	-	-	-	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-	-	-
3	-	-	-	-	-	-	-	-	-	-
>3	-	-	-	-	-	-	-	-	-	-

**Tabel 4. 8 Pembacaan tag vertikal pada arah y-**

Jarak reader vs tag (cm)	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
>12	-	-	-	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-	-	-
3	-	-	-	-	-	-	-	-	-	-
>3	-	-	-	-	-	-	-	-	-	-

**Tabel 4. 9 Pembacaan tag vertikal pada arah x+**

Jarak reader vs tag (cm)	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
>12	-	-	-	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-	-	-
3	√	√	-	√	-	-	-	-	-	-
>3	√	√	√	√	√	√	√	-	√	√

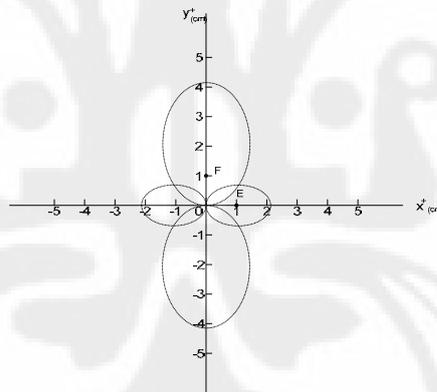
**Tabel 4. 10 Pembacaan tag vertikal pada arah x-**

Jarak reader vs tag (cm)	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
>12	-	-	-	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-	-	-
3	-	-	-	-	-	-	-	-	-	-
>3	√	√	√	√	√	√	√	√	√	√

Tabel 4.7 menunjukkan hasil pembacaan tag vertikal pada array  $y+$ . Pada tabel tersebut terlihat bahwa tag sama sekali tidak ada yang terbaca. Hasil yang sama juga diperlihatkan pada Tabel 4.8. Hasil berbeda ditunjukkan pada Tabel 4.9 dan Tabel 4.10. Terlihat bahwa terdapat keberhasilan baca untuk tag vertikal pada arah  $x$ .

Untuk pembacaan arah  $y$  baik  $y+$  maupun  $y-$ , tag tidak terbaca sama sekali. Hal ini berarti bahwa tag tidak bisa dibaca tegak lurus terhadap modul pembaca. Untuk pembacaan arah  $x$  tag masih dapat terbaca hanya saja untuk jarak kurang dari 3 cm. Walaupun ada keberhasilan baca untuk jarak 3 cm, namun persentasinya tidak besar dan seringkali gagal.

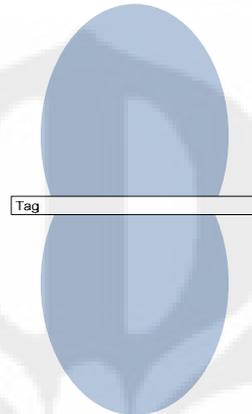
Dari pengujian-pengujian pada sub-bab ini dan sub-bab 4.2.1 didapatkan asumsi bahwa pembaca RFID mempunyai pola radiasi yang tergambar pada Gambar 4.7.



**Gambar 4. 7** Perkiraan pola radiasi *reader*

Jangkauan radiasi paling besar ada pada arah  $y$  dengan kisaran jarak sebesar 4 cm. Untuk jangkauan dengan arah  $x$  mempunyai kisaran jarak sebesar 2 cm. Hal ini berarti bahwa pembacaan optimal pembaca RFID adalah pada arah  $y$ . Kemudian berdasarkan pengujian dengan tag dibuat vertikal menunjukkan bahwa walaupun tag berada dalam jangkauan baca optimal arah  $y$  tetap saja tag tidak terbaca sama sekali. Dari sini timbul asumsi bahwa tag itu sendiri mempunyai pola radiasi RFID sendiri. Perkiraan pola radiasi tag digambarkan pada Gambar

4.8. Hanya saja jangkauannya tidak dapat diketahui dengan pasti. Sehingga dari sini dapat dijelaskan mengapa ketika dengan tag vertikal tidak dapat terbaca pada arah y namun dapat terbaca pada arah x. Kesimpulannya adalah untuk pembacaan optimal RFID baik, radiasi pembaca RFID dan tag harus saling bertemu.

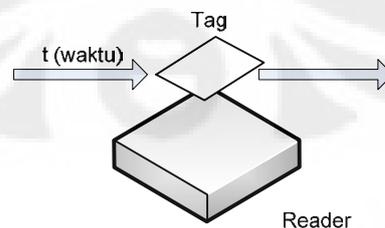


**Gambar 4. 8 Perkiraan pola radiasi tag**

### 4.3 Pengujian Pembacaan dengan Kondisi Khusus

Pengujian ini masih menguji kemampuan baca pembaca RFID seperti pada pengujian sebelumnya. Perbedaannya pada pengujian ini dilakukan pembacaan-pembacaan dengan kondisi khusus. Masing-masing pengujian mempunyai parameter yang berbeda-beda. Tujuan dari pengujian-pengujian ini adalah untuk simulasi keadaan dengan aktivitas pada skenario sistem.

#### 4.3.1 Pembacaan dengan Interval Waktu



**Gambar 4. 9 Ilustrasi pengujian baca dengan interval waktu**

Pengujian ini melihat kemampuan baca pembaca RFID terhadap tag dengan cara biasa yaitu pada arah y+ dengan tag horizontal. Pengujian dilakukan dengan

**Universitas Indonesia**

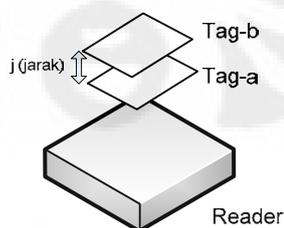
interval waktu antara satu pembacaan tag dengan pembacaan berikutnya. Pembacaan dengan interval waktu 0,5 detik berarti antara pembacaan tag #1 dan tag#2 mempunyai jeda sepanjang 0,5 detik. Begitu pula jeda waktunya untuk tag-tag berikutnya. Parameternya adalah keberhasilan pembaca RFID membaca tag. Ilustrasi pengujian ditunjukkan pada Gambar 4.9. Hasil pengujian ini disajikan pada Tabel 4.11.

**Tabel 4. 11 Pembacaan tag dengan interval waktu**

waktu (detik)	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
0,5	√	-	-	√	-	√	-	√	√	-
1	√	√	√	√	√	√	√	√	-	-
2	√	√	√	√	√	√	√	√	√	√
3	√	√	√	√	√	√	√	√	√	√
>3	√	√	√	√	√	√	√	√	√	√

Pada Tabel 4.11 terlihat bahwa keberhasilan 100% didapat ketika jeda waktu pembacaan 2 detik. Dari data yang diperoleh didapatkan kesimpulan bahwa keberhasilan baca seratus persen ketika interval waktu baca diatas dua detik. Pengujian ini memperlihatkan kenapa terdapat kegagalan baca pada pengujian baca dengan arah dan jarak tertentu di sub-bab 4.2. Sehingga untuk parameter yang diperhatikan untuk pembaca RFID tidak hanya dari jarak dan arah saja tetapi juga interval waktu antara suatu pembacaan dengan pembacaan berikutnya.

#### 4.3.2 Pembacaan dengan Dua Tag dibuat berhimpit



**Gambar 4. 10 Ilustrasi pengujian baca dengan dua tag dibuat berhimpit**

Pengujian ini ditujukan untuk melihat perilaku baca yang dilakukan pembaca RFID ketika dikondisikan terdapat dua tag yang bertumpuk pada jarak tertentu.

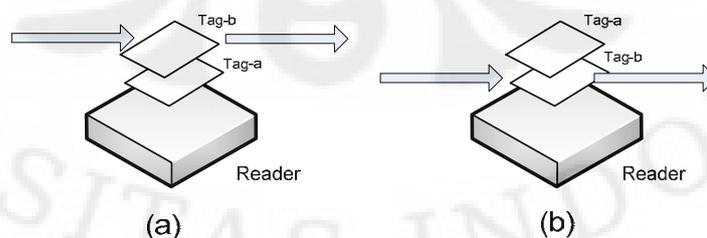
Pengujian ini diilustrasikan pada Gambar 4.10. Parameter yang dilihat adalah tag yang terbaca dari dua tag yang dibuat bertumpuk tersebut. Pengujian ini untuk mensimulasikan kondisi pembacaan ketika ada dua buku yang masuk ke dalam rak dimana terdapat satu pembaca RFID. Jarak baca yang digunakan mulai dari 0 cm sampai 3 cm. Jarak yang lebih dari 3 cm diasumsikan mempunyai data pembacaan yang sama dengan jarak 3 cm. Data hasil pengujian disajikan pada Tabel 4.12.

**Tabel 4. 12 Hasil pembacaan uji dua tag yang dibuat berhimpit**

jarak himpit (cm)	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
0	-	-	-	-	-	-	-	-	-	-
1	a	a	a	a	a	a	a	a	a	a
2	a	a	a	a	a	a	a	a	a	a
3	a	a	a	a	a	a	a	a	a	a

Diperlihatkan pada Tabel 4.12 bahwa pada jarak himpit 0 cm tag tidak ada yang terbaca. Asumsinya adalah pada jarak himpit demikian sinyal yang dikirim oleh kedua tag saling mengganggu sehingga pembaca RFID tidak bisa membaca sinyal dari tag secara benar. Untuk jarak mulai dari 1 cm dan lebih, tag yang terbaca adalah tag yang lebih dekat dengan pembaca RFID dalam hal ini adalah tag-a. Kesimpulan yang didapat adalah ketika ada dua buku yang dimasukkan ke dalam rak dengan satu pembaca RFID maka hanya buku yang posisinya lebih dekat saja yang terbaca, sedangkan yang satunya lagi tidak.

#### 4.3.3 Pembacaan dengan satu tag diam dan satu tag berjalan



**Gambar 4. 11 Ilustrasi pengujian baca dengan satu tag diam dan satu tag berjalan, (a) tag diam berada di dekat reader (b) tag berjalan berada di dekat reader**

Pengujian berikutnya adalah uji baca dengan satu tag diam dan satu tag berjalan. Ilustrasinya digambarkan oleh Gambar 4.11. Dilakukan dua keadaan, pertama ketika tag diam (tag-a) lebih dekat ke pembaca RFID daripada tag berjalan (tag-b) dan yang kedua adalah ketika tag diam lebih jauh dari pembaca RFID daripada tag berjalan. Tujuan pengujian ini adalah untuk mensimulasikan apabila dalam suatu rak dengan satu pembaca RFID dimasuki buku dimana telah ada buku sebelumnya yang berada di sana.

Kondisi pertama adalah ketika buku yang telah ada berada lebih dekat ke pembaca RFID (kondisi-a) sedangkan kondisi kedua adalah ketika buku yang baru dimasukkan yang jaraknya lebih dekat ke pembaca RFID (kondisi-b). Parameter yang dilihat dari pengujian ini adalah tag yang terbaca oleh pembaca RFID. Data tag terbaca tidak mengambil pembacaan inisial atau ketika tag diam baru diletakkan di depan pembaca RFID. Data baru diambil ketika tag berjalan digerak-gerakkan. Data hasil pengujian disajikan pada Tabel 4.13.

**Tabel 4. 13 Pembacaan satu tag diam dan satu tag berjalan**

#	Tag Terbaca	
	a dibawah b (kondisi-a)	a diatas b (kondisi-b)
1	-	b -> a
2	-	b -> a
3	-	b -> a
4	-	b -> a
5	-	b -> a
6	-	b -> a
7	-	b -> a
8	-	b -> a
9	-	b -> a
10	-	b -> a

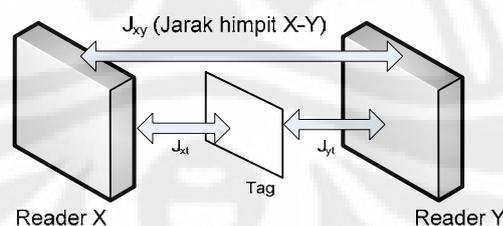
Data yang diperlihatkan pada Tabel 4.13 menunjukkan bahwa pada kondisi-a tag berjalan tersebut sama sekali tidak terbaca. Tag diam yang sudah terbaca pada keadaan inisial pun tidak terbaca lagi. Hal ini berarti ketika pembaca RFID telah membaca suatu tag, maka pembaca RFID tidak dapat membaca tag baru selagi tag

sebelumnya masih berada pada jangkauan baca pembaca RFID. Sehingga apabila dikondisikan pada keadaan sebenarnya, buku yang baru masuk ke dalam suatu rak tidak akan terbaca ketika ada buku lain yang sebelumnya berada lebih dekat ke pembaca RFID.

Untuk kondisi kedua hasilnya berbeda lagi. Tag yang berjalan terbaca oleh pembaca RFID, namun ketika tag tersebut meninggalkan area baca pembaca RFID maka tag yang diam juga terbaca. Artinya ketika suatu buku dimasukkan ke dalam rak dimana buku tersebut lebih dekat jaraknya ketimbang buku lain yang sudah ada kemudian buku tersebut dikeluarkan lagi, maka buku yang sebelumnya diam akan terbaca sebagai buku yang baru dimasukkan.

#### 4.3.4 Pembacaan dengan dua reader dibuat berhadapan

Pengujian terakhir pada uji pembacaan dengan kondisi khusus ini adalah dengan membuat pembaca RFID saling berhadapan pada jarak tertentu. Jarak hadap ( $J_{xy}$ ) yang dipakai adalah 10 cm, 8 cm, 6 cm, dan 4 cm. Kemudian satu tag diletakkan tepat ditengah-tengah kedua pembaca RFID ( $J_{xy}/2$ ). Setelah itu tag tersebut digeser ke arah *reader-x* dan *reader-y* hingga pada jarak tertentu tag tersebut terbaca oleh salah satu *reader*.  $J_{xt}$  adalah jarak antara *reader-x* dan tag dimana titik 0 cm ada di *reader-x*. Sedangkan  $J_{yt}$  adalah jarak antara *reader-y* dan tag dimana titik 0 cm ada di *reader-y*.



**Gambar 4. 12 Ilustrasi pengujian pembacaan dengan dua reader dibuat berhadapan**

Pengujian ini dimaksudkan untuk mensimulasikan perilaku pembacaan dalam satu rak dengan dua pembaca RFID. Parameter yang diambil adalah keberhasilan baca tag ketika berada tepat diantara dua pembaca RFID dan jarak baca tag ketika

tag tersebut digeser-geser ke arah salah satu pembaca RFID. Ilustrasi pengujian ini digambarkan pada Gambar 4.12. Hasil pengujian disajikan pada Tabel 4.14.

**Tabel 4. 14 Pembacaan dua reader dibuat berhadapan**

$J_{xy}$ (cm)	Tag Terbaca pada $J_{xy}/2$	$J_{xt}$ (cm)	$J_{yt}$ (cm)
10	-	5	4
8	-	2,5	3
6	-	2	2
4	-	-	-

Data hasil pengujian pada Tabel 4.14 menunjukkan bahwa ketika berada tepat ditengah-tengah tag tidak terbaca sama sekali. Asumsi penyebabnya adalah karena radiasi kedua pembaca RFID pada titik tersebut mengalami resultan atau saling meniadakan.

Kemudian jarak pembacaan tag yang digeser ke salah satu pembaca RFID mengalami penurunan seiring dengan mengecilnya jarak antar kedua pembaca tersebut. Tag masih terbaca untuk  $J_{xy}$  bernilai 10 sampai 6 cm. Kecuali pada  $J_{xy}$  bernilai 4 cm tag tidak terbaca sama sekali. Sama seperti asumsi sebelumnya, kemungkinan penyebabnya adalah terjadi resultan radiasi kedua pembaca RFID yang membuat tag tidak terbaca.

Kesimpulan dari pengujian ini adalah pada kedua pembaca RFID yang berhadapan terdapat jarak antar pembaca RFID minimum agar tag yang berada diantara kedua pembaca tersebut dapat terbaca.

#### **4.4 Pengujian Tanggapan Fungsi Applet**

Uji tanggapan applet dilakukan untuk menguji applet dengan melihat waktu tanggap applet, keberhasilan baca, serta bagaimana applet tersebut menyimpan data masuk ke dalam log.

Log yang dilihat adalah log yang berupa tabel dalam database dan log yang berupa file. Selain keberhasilan simpan log juga dilihat urutan simpan data.

Maksud dari parameter tag terbaca adalah notifikasi yang dikeluarkan applet terhadap tag tertentu di layar server. Contohnya jika tag-a yang terbaca oleh pembaca RFID maka notifikasi yang keluar adalah notifikasi yang mempunyai data buku yang berkaitan dengan tag-a dengan begitu pada kolom diisi tag terbaca adalah tag-a. Notifikasi yang keluar tidak langsung ditutup sehingga apabila dilakukan lima pembacaan maka menunggu pembacaan tersebut selesai kelimanya baru melihat dan menutup notifikasi. Urutan notifikasi itulah yang dicatat pada kolom tag terbaca.

Parameter waktu yang digunakan menggunakan satuan waktu *millisecond* dan *microsecond*. Keduanya dihasilkan dari fungsi dalam Java yang berbeda. Untuk *microsecond* dihasilkan dari fungsi waktu *nanosecond* dibagi seribu. Penggunaan kedua fungsi dilakukan karena penulis menemukan anomali jika membandingkan nilai waktu antara fungsi yang menghasilkan *millisecond* dan fungsi yang menghasilkan *microsecond*. Pada pengujian ini tidak terlihat anomali tersebut. Baru pada pengujian di sub-bab 4.5 terlihat. Waktu inisial diambil pada setiap thread koneksi mulai dari applet menerima data RFID masuk dan diakhiri setelah applet mencatat data dalam log. Untuk gambaran lebih jelas bagaimana fungsi waktu ditempatkan dapat dilihat sebagai berikut:

```

long t0 = System.currentTimeMillis();
long tn0 = System.nanoTime();
//mendapat data masuk
//mendapat identitas buku yang berkaitan dengan tag
//mencatat dalam log

long tn1 = (System.nanoTime() - tn0)/1000;
long t1 = System.currentTimeMillis() - t0;
//notifikasi keluar + nilai tn1 dan t1

```

t0 dan t1 adalah waktu dalam satuan *millisecond* sedangkan tn0 dan tn1 adalah waktu dalam satuan *microsecond*. Pasangan pengambilan waktu *microsecond*

ditempatkan didalam pasangan pengambilan waktu *millisecond*. Kedua nilai waktu tersebut bersama ditampilkan dalam notifikasi untuk kemudian diisi pada kolom waktu response.

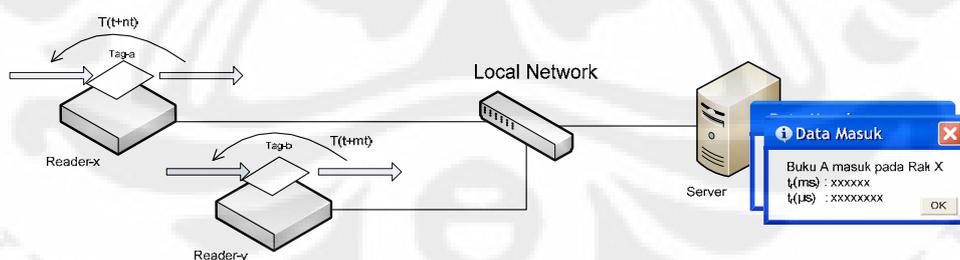
Pengujian dilakukan dalam tiga kondisi dengan masing-masing dilakukan dengan dua keadaan. Pertama secara cepat yaitu dengan jeda waktu antar pembacaan kira-kira 0,3 detik. Kedua secara normal yaitu dengan jeda waktu antar pembacaan kira-kira satu detik. Pada setiap keadaan diambil sepuluh buah data pembacaan.

#### 4.4.1 Pembacaan dua tag pada dua reader yang berbeda

Pengujian applet pertama adalah uji baca dua tag pada dua reader yang berbeda. Urutan pembacaan adalah tag-a dibaca oleh *reader-x* kemudian tag-b dibaca oleh *reader-y* setelah itu dilakukan lagi pembacaan tag-a oleh *reader-x* dan seterusnya. Dengan kata lain urutan baca digambarkan dengan:

ax -> by -> ax -> by -> ax -> by-> ...

Fungsi waktu pembacaan digambarkan pada persamaan  $T(t+nt)$  untuk tag-a dengan  $n = 0,2,4,6,\dots$  kemudian persamaan  $T(t+mt)$  untuk tag-b dengan  $n = 1,3,5,7,\dots$  dengan  $t = 300$  ms untuk pembacaan cepat dan  $t = 1$  s untuk pembacaan normal.



**Gambar 4. 13 Ilustrasi pengujian pembacaan dua tag pada dua reader yang berbeda**

Pengujian ini diilustrasikan pada Gambar 4.13. Hasil pengujian disajikan pada Tabel 4.15.

**Tabel 4. 15 Hasil tanggapan pembacaan dua tag pada dua reader yang berbeda**

#	secara cepat (appr. 300 ms)						secara normal (appr. 1 s)					
	tag terbaca	waktu response		tersimpan di log			tag terbaca	waktu response		tersimpan di log		
		(ms)	( $\mu$ s)	file	db	#urut simpan		(ms)	( $\mu$ s)	file	db	#urut simpa
1	b	22563	22552864	√	-	2	b	174187	174183147	√	√	2
2	b	313	307494	√	√	3	b	204	205018	√	√	3
3	b	31	29917	√	√	4	b	93	89795	√	√	4
4	b	94	84669	√	√	5	b	125	116785	√	√	5
5	b	78	85199	√	√	6	b	94	91729	√	√	6
6	a	19797	19802794	√	√	1	a	132484	132483379	√	√	1
7	a	94	87125	√	√	7	a	94	91421	√	√	7
8	a	47	52001	√	√	8	a	63	52531	√	√	8
9	a	94	85817	√	√	9	a	78	87833	√	√	9
10	a	78	86542	√	√	10	a	31	37439	√	√	10
AVG		4318,9	4317442,2					30745,3	30743908			

Baik pengujian secara cepat maupun normal menunjukkan hasil yang mirip. Terlihat bahwa notifikasi yang keluar adalah notifikasi yang memberi tahu pembacaan data pada tag-b terlebih dahulu ketimbang pembacaan tag-a. Seperti diketahui bahwa pada pengujian tag-a dibaca oleh pembaca RFID terlebih dahulu baru kemudian tag-b. Hal ini menunjukkan notifikasi yang keluar dari hasil pembacaan suatu pembaca RFID akan di-*override* apabila ada pembacaan dari pembaca RFID lain yang datang belakangan. Ini terjadi ketika notifikasi dari thread koneksi dengan pembaca RFID awal belum sempat ditutup dan muncul notifikasi dari thread koneksi pembaca RFID berikutnya. Sehingga pada Tabel 4.15 terlihat bahwa lima data awal menunjukkan pembacaan tag-b baru kemudian pembacaan tag-a. Padahal pembacaan fisiknya dilakukan secara bergantian.

Data waktu menunjukkan bahwa waktu tanggap applet paling besar terjadi pada awal koneksi. Nilainya cukup besar ketimbang pembacaan-pembacaan berikutnya yang rata-ratanya adalah puluhan *millisecond*. Ini berarti pada koneksi awal applet membutuhkan waktu lebih lama untuk menangani koneksi pada thread koneksi dengan pembaca RFID.

Penyimpanan data ke dalam log selalu berhasil untuk log dalam bentuk file namun terjadi kegagalan kecil untuk log dalam database. Alasannya adalah pada tabel log *primary key*-nya yang bertipe waktu tidak memperbolehkan nilai yang sama. Oleh karena itu terjadi kegagalan simpan. Pada nomor urut simpan dalam log terlihat bahwa sebenarnya tag-a (pada data ke-6) terlebih dahulu yang disimpan. Kemudian baru tag-b hingga lima buah dan dilanjutkan empat data tag-a.

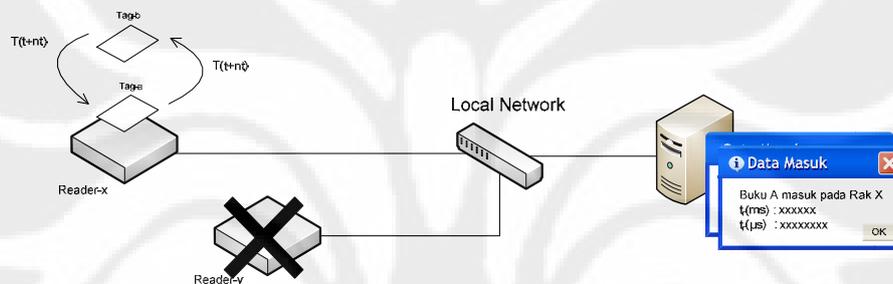
Universitas Indonesia

#### 4.4.2 Pembacaan dua tag pada satu reader

Pengujian berikutnya adalah uji pembacaan dua tag pada satu *reader*. Urutan pembacaan adalah tag-a dibaca oleh *reader-x* kemudian tag-b dibaca oleh *reader-x* juga setelah itu tag-a lagi dibaca oleh *reader-x* dan seterusnya. Urutan baca digambarkan dengan:

ax -> bx -> ax -> bx -> ax -> bx-> ...

Fungsi waktu posisi pembacaan digambarkan pada persamaan  $T(t+nt)$  untuk posisi tag-a dan posisi tag-b dengan  $n = 1,2,3,4,\dots$  dengan  $t = 300$  ms untuk pembacaan cepat dan  $t = 1$  s untuk pembacaan normal.



**Gambar 4. 14 Ilustrasi pengujian pembacaan dua tag pada satu reader yang berbeda**

Pengujian ini diilustrasikan pada Gambar 4.14. Data hasil pembacaan disajikan pada Tabel 4.16.

**Tabel 4. 16 tanggapan pembacaan satu tag pada dua reader yang berbeda**

#	tag terbaca	secara cepat (appr. 300 ms)					secara normal (appr. 1 s)					
		waktu response		tersimpan di log			tag terbaca	waktu response		tersimpan di log		
		(ms)	(µs)	file	db	#urut simpan		(ms)	(µs)	file	db	#urut simpa
1	a	41218	41222379	√	√	1	a	96031	96021634	√	√	1
2	b	109	102034	√	√	2	b	407	418362	√	√	2
3	a	187	188140	√	√	3	a	62	58677	√	√	3
4	b	125	126441	√	√	4	b	359	361101	√	√	4
5	a	172	158826	√	√	5	a	421	435644	√	√	5
6	b	109	112559	√	√	6	b	375	370705	√	√	6
7	a	406	402732	√	√	7	a	32	40039	√	√	7
8	b	219	228841	√	√	8	b	157	157569	√	√	8
9	a	328	325785	√	√	9	a	93	90651	√	√	9
10	b	515	524057	√	√	10	b	94	81777	√	√	10
AVG		4338,8	4339179,4					9803,1	9803615,9			

Hasil pengujian seperti diperlihatkan Tabel 4.16 menunjukkan bahwa keberhasilan simpan dalam log baik dalam bentuk database maupun dalam bentuk file mencapai seratus persen. Walaupun begitu karena menggunakan tabel log yang sama ada kemungkinan ada kegagalan simpan log dalam database apabila ada dua buku yang dimasukkan pada waktu yang sama. Mengingat bahwa *primary key* dalam database tidak boleh ganda. Untuk log dalam bentuk file akan selalu berhasil karena tidak memperhatikan batasan-batasan yang dikenakan seperti dalam database.

Waktu inisial pembacaan bernilai besar hanya pada pembacaan pertama yaitu ketika koneksi antara pembaca RFID dengan server baru dibuat. Pembacaan pada satu pembaca RFID selalu berhasil pada waktu yang cepat karena tag yang digunakan berbeda. Berbeda dengan pengujian pada sub-bab 4.3.1 yang seringkali mengalami kegagalan karena penggunaan tag yang sama.

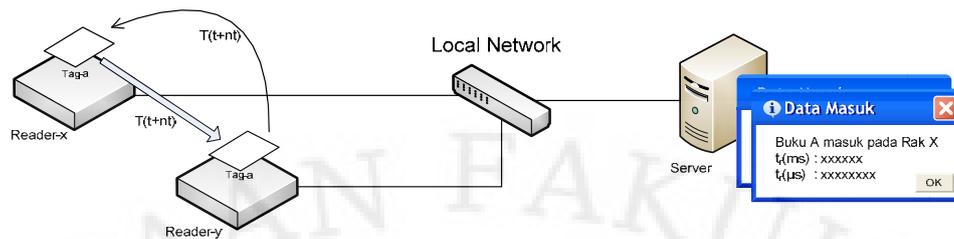
Notifikasi yang keluar juga berurutan berbeda dengan pengujian sebelumnya. Ketika tag dibaca bergantian maka notifikasi yang keluar juga bergantian. Begitu pula nomor urut simpan pada log.

#### 4.4.3 Pembacaan satu tag pada dua reader

Pengujian terakhir adalah uji pembacaan satu tag pada dua *reader*. Urutan pembacaan adalah tag-a dibaca oleh *reader-x* kemudian tag-a dibaca oleh *reader-y* setelah itu tag-a lagi dibaca oleh *reader-x* dan seterusnya. Urutan baca digambarkan dengan:

$$ax \rightarrow ay \rightarrow ax \rightarrow ay \rightarrow ax \rightarrow ay \rightarrow \dots$$

Fungsi waktu posisi pembacaan pada pembaca RFID digambarkan pada persamaan  $T(t+nt)$  untuk posisi tag-a dan posisi tag-b dengan  $n = 1,2,3,4,\dots$  dengan  $t = 300$  ms untuk pembacaan cepat dan  $t = 1$  s untuk pembacaan normal. Pengujian ini diilustrasikan pada Gambar 4.15. Data hasil pengujian disajikan pada Tabel 4.17.



**Gambar 4. 15** Ilustrasi pengujian pembacaan satu tag pada dua reader yang berbeda

**Tabel 4. 17** tanggapan pembacaan dua tag pada satu reader yang berbeda

#	secara cepat (appr. 300 ms)						secara normal (appr. 1 s)					
	tag terbaca	waktu response		tersimpan di log			tag terbaca	waktu response		tersimpan di log		
		(ms)	(μs)	file	db	#urut simpan		(ms)	(μs)	file	db	#urut simpan
1	ay	30578	30577953	√	√	2	ay	185406	185412603	√	√	2
2	ay	359	355383	√	√	3	ay	79	79614	√	√	3
3	ay	297	293207	√	√	4	ay	109	103467	√	√	4
4	ay	78	74630	√	√	5	ay	141	140708	√	√	5
5	ay	391	391862	√	√	6	ay	344	347895	√	√	6
6	ax	34266	34264814	√	√	1	ax	160500	160499613	√	√	1
7	ax	93	85458	√	√	7	ax	343	348749	√	√	7
8	ax	375	375368	√	√	8	ax	343	349235	√	√	8
9	ax	78	66444	√	√	9	ax	344	338895	√	√	9
10	ax	94	100119	√	√	10	ax	312	309714	√	√	10
AVG		6660,9	6658523,8					34792,1	34793049			

Mirip dengan pengujian pada sub-bab 4.4.1, notifikasi yang muncul adalah notifikasi untuk pembacaan pada pembaca RFID yang kedua. Notifikasi pembacaan sebelumnya di-*override* oleh notifikasi pembacaan pada pembaca RFID yang berikutnya. Waktu paling lama juga terjadi ketika koneksi awal antara pembaca RFID dengan server untuk pertama kali. Penyimpanan log semuanya berhasil dengan memperhatikan bahwa log dalam database sewaktu-waktu bisa gagal karena alasan yang dikemukakan pada sub-bab 4.4.1 dan sub-bab 4.4.2. Urut simpan dalam log juga sama dengan urut simpan log pada pengujian di sub-bab 4.4.1.

Dari semua data didapat waktu rata-rata tanggap applet total untuk semua pengujian sub-bab ini adalah 15109286,38 μs.

#### 4.5 Pengujian Halaman Web

Pada uji halaman web parameter yang dilihat adalah waktu *load* halaman dalam satuan waktu *millisecond* dan *microsecond*. Halaman web dikembangkan dengan menggunakan JSP oleh karena itu fungsi Java yang digunakan juga sama dengan fungsi yang dipakai pada pengujian di sub-bab 4.4. Nilai waktu *microsecond* didapat dari fungsi waktu *nanosecond* dibagi 1000. Fungsi waktu inisial ditempatkan sebelum tag <html> dan fungsi waktu akhir setelah tag penutup </html>. Untuk lebih jelasnya berikut adalah gambaran penempatan fungsi waktu pada pengujian ini:

```
<%
long t0 = System.currentTimeMillis();
long tn0 = System.nanoTime();
%>
<html>
    <!-- kode-kode JSP untuk halaman web -->
</html>
<%
long tn1 = (System.nanoTime() - tn0)/1000;
long t1 = System.currentTimeMillis() - t0;
%>
<p>t(mili) : <%=t1%><br />
t(micro): <%=tn1%></p>
```

Semua halaman web diuji baik untuk penampilan tabel, penambahan, pengeditan, dan penghapusan data, serta halaman tulisan seperti halaman utama dan halaman Help. Pengujian dilakukan di tiga komputer yang berbeda dengan spesifikasi sebagai berikut:

##### local

Microsoft XP Professional Version 2002 Service Pack 3  
Genuine Intel® CPU T2500 @ 2.00 GHz  
2.00 GHz, 0.99 GB of RAM

##### client-1

Windows Vista™ Ultimate  
Intel® Core™ 2 Quad CPU Q8200 @ 2.33 GHz 2.33 GHz  
4094 MB of RAM  
64-bit Operating System

##### client-2

MS Windows 2000 5.00.2195 Service Pack 4  
x86 Family 6 Model 8 Stepping 6  
AT/AT Compatible  
260,592 KB RAM

Komputer local adalah server itu sendiri yaitu pengujian dilakukan dengan melihat halaman web secara local. Client-1 adalah komputer yang dianggap cukup canggih yang spesifikasinya dapat dikatakan diatas rata-rata. Client-2 adalah komputer lawas yang masih berfungsi dengan baik dan dengan spesifikasi dibawah rata-rata. Kedua komputer client tersebut masih berada dalam satu jaringan lokal dengan server. Pada pengujian ini bagaimana applet mendapat data RFID tidak diperhatikan. Hal yang dilihat adalah seberapa lama satu halaman di-load.

Web terdiri atas delapan halaman JSP. Dua diantaranya adalah halaman utama yang memuat applet dan halaman help. Keenam halaman lain adalah halaman dengan akses database berdasarkan *request* pada halaman tersebut. Pada enam halaman tersebut masing-masing menangani 6 sampai 7 tabel untuk kemudian kita sebut sebagai sub-halaman. Masing-masing sub-halaman diambil 5 pasang data waktu untuk 3 komputer yang berbeda. Terdapat 37 sub-halaman ditambah 2 halaman tulisan. Sehingga data yang diambil ada 585 pasang data atau 1170 data. Data yang disajikan pada Tabel 4.18 hingga Tabel 4.24 adalah waktu rata-rata *load* sub-halaman dan dua halaman tulisan web. Data-data yang lebih lengkap dilampirkan di bagian lampiran tulisan ini.

#### 4.5.1 Halaman Tulisan

Bagian yang disebut sebagai halaman tulisan terbagi atas dua yaitu halaman utama (*index.jsp*) yang memuat applet dan halaman help (*help.jsp*) yang berisikan tulisan tambahan. Data pada pengujian ini disajikan pada Tabel 4.18.

**Tabel 4. 18 Waktu load halaman utama dan help**

#	local		client-1		client-2		
	(ms)	( $\mu$ s)	(ms)	( $\mu$ s)	(ms)	( $\mu$ s)	
Main	Index	0	1631	0	1353,4	0	1197,4
	Help	0	845,8	0	956,2	0	874,4
	(rata-rata)	0	1238,4	0	1154,8	0	1035,9

Dari data yang terlihat anomali nilai waktu, yaitu nilai *millisecond* seharusnya tidak nol karena nilai *microsecond* menunjukkan hasil ribuan dimana paling tidak

ada 1 *millisecond*. Hal ini menunjukkan bahwa data nilai waktu *microsecond* lebih detail menunjukkan waktu *load* kode halaman maupun kode program. Rata-rata waktu *load* halaman berupa tulisan adalah 1 *millisecond* dengan hasil yang tidak begitu jauh untuk ketiga komputer.

#### 4.5.2 Lihat Data

Halaman lihat data (*show.jsp*) adalah halaman yang berfungsi untuk melihat tabel-tabel database sistem. Fungsi edit dan delete juga melalui halaman ini. Halaman ini menampilkan tabel-tabel bergantung pada *request* terhadap halaman tersebut. Halaman lihat log (*showlog.jsp*) juga termasuk dalam kategori ini hanya saja pada halaman tersebut tidak dimungkinkan melakukan pengeditan dan penghapusan terhadap *entry* pada tabel log.

**Tabel 4. 19 Waktu load halaman lihat data**

	#	local		client-1		client-2	
		(ms)	( $\mu$ s)	(ms)	( $\mu$ s)	(ms)	( $\mu$ s)
Lihat Data	Buku	15,6	15698,4	156	155372,6	15,8	16039
	Data Buku	15,6	15698,4	156	155372,6	15,8	16039
	Penerbit	12,8	10810,6	12,2	8748,4	3	9489
	Tag	18,8	17172,8	6,2	9190,6	31,2	29392,8
	Status	15,6	14365,4	6,4	7206,4	12,4	9899,6
	Rak	9,4	11616,4	0	8306	12,4	9899,6
	Log	37,4	36735,4	12,6	14920	50,2	53803,8
	<b>(rata-rata)</b>	<b>17,89</b>	<b>17442,49</b>	<b>49,91</b>	<b>51302,37</b>	<b>20,11</b>	<b>20651,83</b>

Dari data yang disajikan pada Tabel 4.19 terlihat bahwa terdapat perbedaan waktu yang tidak cukup jauh. Pada waktu rata-rata *load* sub-halaman Buku dan Data Buku untuk client-1 cukup besar sehingga membuat rata-rata totalnya lebih besar. Pada tabel juga terlihat bahwa nilai waktu *millisecond* tidak berbeda jauh dengan nilai waktu *microsecond*. Namun anomali kembali terlihat untuk waktu yang cukup kecil contohnya pada waktu *load* sub halaman Rak untuk client-1 dan sub-halaman Penerbit untuk client-2.

#### 4.5.3 Tambah Data

Halaman tambah data (*insert.jsp*) adalah halaman yang berfungsi untuk menambah data tabel tertentu bergantung pada *request* yang diberikan terhadap halaman ini. Data pengujian untuk halaman ini disajikan pada Tabel 4.20.

**Tabel 4. 20 Waktu load halaman tambah data**

	#	local		client-1		client-2	
		(ms)	( $\mu$ s)	(ms)	( $\mu$ s)	(ms)	( $\mu$ s)
Tambah Data	Buku	56,2	59321,8	59,2	69575,6	109,6	107419,2
	Data Buku	46,8	46677,8	106,2	107178,4	68,6	69525,8
	Penerbit	50	52364,4	137,4	139597,8	118,8	118167
	Tag	68,8	64592,4	150	152809,8	50	51242,2
	Status	62,4	64440	50	52774,6	150	153677,6
	Rak	72	71186	62,4	62884,6	478,4	478204,8
	<b>(rata-rata)</b>	<b>59,37</b>	<b>59763,73</b>	<b>94,20</b>	<b>97470,13</b>	<b>162,57</b>	<b>163039,43</b>

Data waktu *load* yang didapat untuk halaman ini berbeda untuk tiap client. Nilai waktu untuk local paling kecil dan nilai waktu untuk client-2 paling besar. Nilai waktu *millisecond* dan *microsecond* tidak berbeda jauh. Ini berarti untuk nilai waktu yang besar, kedua fungsi mempunyai nilai yang hampir sama.

#### 4.5.4 Edit Data

Halaman edit data (*edit.jsp*) adalah halaman yang berfungsi untuk melakukan perubahan data. Pada halaman ini yang ditampilkan adalah form perubahan data. Jadi pada halaman ini bukan lagi berupa tabel. Halaman ini dipanggil melalui *link* yang berisi *request* terhadap halaman edit ini. Data yang akan diedit bergantung pada *entry* pada tabel apa yang ingin diubah datanya. Data hasil pengujian halaman ini disajikan pada Tabel 4.21.

Tabel 4. 21 Waktu load halaman edit data

#	local		client-1		client-2	
	(ms)	( $\mu$ s)	(ms)	( $\mu$ s)	(ms)	( $\mu$ s)
Buku	6,4	12747,4	15,4	12464,4	6,2	10963,8
Data Buku	12,6	11014	12,4	9311	9,4	9107,2
Penerbit	12,6	8149,2	3	6733,6	3	6323,8
Tag	15,8	9724	9,6	8217,8	0	8141
Status	9,4	8289,6	12,6	6004,8	3,2	6249,6
Rak	15,6	24494	6,4	6516,4	9,6	5969,2
<b>(rata-rata)</b>	<b>12,07</b>	<b>12403,03</b>	<b>9,90</b>	<b>8208,00</b>	<b>5,23</b>	<b>7792,43</b>

Terlihat pada Tabel 4.21 bahwa kali ini nilai rata-rata total waktu paling kecil adalah untuk client-2 sedangkan nilai waktu paling besar untuk local. Jika mengasumsikan dari pengujian pada sub-bab 4.5.3 bahwa komputer yang spesifikasinya dibawah rata-rata yaitu client-2 mempunyai waktu *load* paling besar ternyata menunjukkan hal yang kurang sesuai pada pengujian ini. Sehingga dari sini timbul asumsi bahwa waktu *load* halaman tidak bergantung ada spesifikasi komputernya. Anomali juga terlihat pada pengujian ini karena nilai waktu yang muncul cukup kecil.

#### 4.5.5 Update Data

Update data (update.jsp) adalah halaman yang berfungsi untuk melakukan perubahan data dari form yang di-submit oleh halaman edit (edit.jsp). Dari halaman edit nilai-nilai form akan diteruskan ke halaman update dan dilakukan perubahan kemudian menampilkan kepada user data sebelum dan sesudah perubahan. Data hasil pengujian ini disajikan pada Tabel 4.22.

Tabel 4. 22 Waktu load halaman update data

	#	local		client-1		client-2	
		(ms)	( $\mu$ s)	(ms)	( $\mu$ s)	(ms)	( $\mu$ s)
Update Data	Buku	78	76282,8	147,2	143632,2	171,6	170215,6
	Data Buku	59,6	59542,4	122	122018,4	237,6	242026,4
	Penerbit	62,6	63981,6	224,8	225585,2	134,4	130661,4
	Tag	81,2	82573,2	106,2	107172,6	187,4	186679
	Status	165,8	163961,4	81,4	79315	40,6	41414,2
	Rak	165,4	170327,6	171,4	169356,6	75	71565,4
	(rata-rata)	<b>102,10</b>	<b>102778,17</b>	<b>142,17</b>	<b>141180,00</b>	<b>141,10</b>	<b>140427,00</b>

Nilai waktu yang didapat cukup besar sehingga nilai *millisecond* dan nilai *microsecond* tidak berbeda jauh. Nilai rata-rata total *load* halaman juga tidak berbeda jauh antar ketiga komputer.

#### 4.5.6 Delete(1): Konfirmasi

Halaman delete (*delete.jsp*) ini berfungsi untuk mengkonfirmasi apakah user benar-benar ingin menghapus data yang ia pilih. Halaman *delete.jsp* sebenarnya menangani dua hal yaitu konfirmasi dan melakukan penghapusan. Untuk konfirmasi, yang ditampilkan adalah data-data yang ingin dihapus user. Halaman ini dipanggil melalui *link* yang mengandung *request* yang ada pada halaman lihat data (*show.jsp*). Pada pengujian ini dilihat waktu *load* konfirmasi penghapusan sedangkan untuk penghapusannya sendiri dilakukan pada pengujian berikutnya. data hasil pengujian disajikan pada Tabel 4.23.

Tabel 4. 23 Waktu load halaman delete(1): konfirmasi

	#	local		client-1		client-2	
		(ms)	( $\mu$ s)	(ms)	( $\mu$ s)	(ms)	( $\mu$ s)
Delete(1): Konfirmasi	Buku	12,8	8272,6	3	6842	15,6	6933,6
	Data Buku	9,4	6952,8	3,2	6572,6	6,4	6377
	Penerbit	9,4	8215,6	3	6335,6	3,2	6261
	Tag	6	8456,2	9,4	6854,8	6	6728
	Status	6,4	7912,2	3	6886	9,4	5601,8
	Rak	6,4	8415,6	3	6653	9,6	6111
	<b>(rata-rata)</b>	<b>8,40</b>	<b>8037,50</b>	<b>4,10</b>	<b>6690,67</b>	<b>8,37</b>	<b>6335,40</b>

Nilai waktu yang diperoleh untuk menampilkan konfirmasi penghapusan sangat kecil. Perbedaan nilai waktu juga tidak begitu besar untuk ketiga komputer. Karena halaman ini hanya menampilkan konfirmasi maka dari itu memang waktu *load*-nya tidak besar.

#### 4.5.7 Delete(2): Hapus

Halaman ini sebenarnya adalah halaman yang sama pada pengujian sebelumnya yaitu *delete.jsp*. Sehingga dalam satu halaman mempunyai dua fungsi yaitu untuk konfirmasi dan untuk penghapusan. Dilakukan pembedaan pengujian karena fungsi yang dilakukan berbeda, yang satu hanya untuk konfirmasi dan yang ini untuk menghapus dan menampilkan kepada user data apa yang sudah dihapus. Data hasil pengujian disajikan pada Tabel 4.24.

Tabel 4. 24 Waktu load halaman delete(2): hapus

	#	local		client-1		client-2	
		(ms)	( $\mu$ s)	(ms)	( $\mu$ s)	(ms)	( $\mu$ s)
Delete(2): Hapus	Buku	303	305995,4	106,4	104400	106	105319
	Data Buku	253,2	248040	218,8	216141,4	181	178640
	Penerbit	87,6	88667,4	99,8	101985	65,6	65479
	Tag	46,8	47270,8	65,6	66366,8	109,4	110170,6
	Status	53,2	50889,2	112,4	113328,6	53,2	53848,4
	Rak	46,8	47234,8	103,2	104010,4	71,8	66063,2
	<b>(rata-rata)</b>	<b>131,77</b>	<b>131349,60</b>	<b>117,70</b>	<b>117705,37</b>	<b>97,83</b>	<b>96586,70</b>

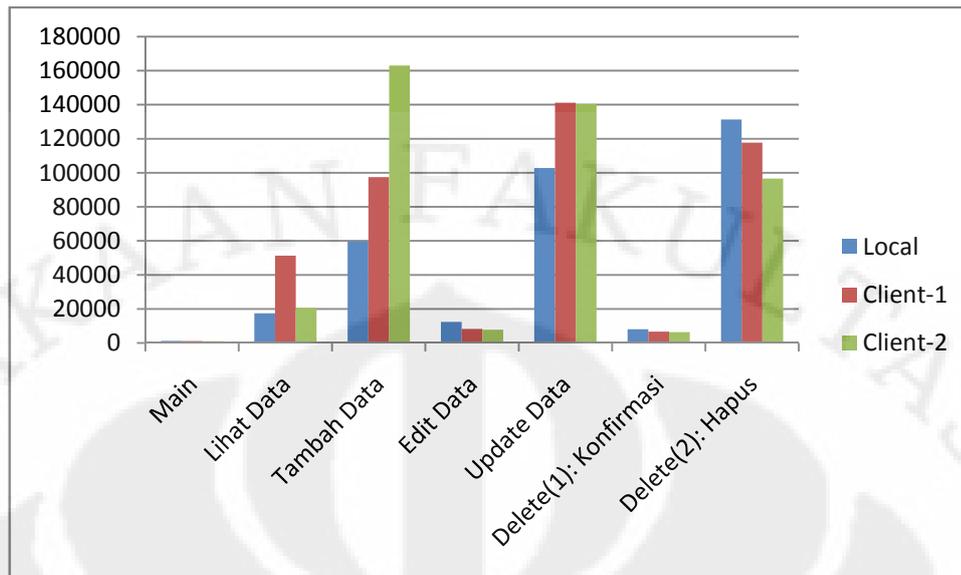
Universitas Indonesia

Berbeda dengan ketika melakukan fungsi konfirmasi, pada fungsi penghapusan dibutuhkan waktu yang lebih besar. Perbedaan tidak begitu jauh untuk ketiga komputer yang digunakan untuk pengujian. Nilai waktu *millisecond* dan *microsecond* juga tidak berbeda.

#### 4.5.8 Hasil Pengujian Waktu *load* Halaman Web

Semua data pengujian untuk semua halaman web yang berjumlah ribuan itu menunjukkan bahwa nilai dari fungsi waktu yang kecil untuk *millisecond* mempunyai keakurasian nilai yang lebih rendah ketimbang dari fungsi waktu dengan satuan *microsecond* yang disini merupakan nilai waktu *nanosecond* dibagi dengan 1000. Oleh karena itu dalam analisis program fungsi waktu yang lebih baik adalah dengan menggunakan fungsi *nanosecond*. Sehingga nilai waktu yang dijadikan acuan adalah nilai dengan satuan waktu tersebut.

Rata-rata total dari waktu *load* halaman ditampilkan dalam bentuk grafik yang ditunjukkan pada Gambar 4.16. Terlihat bahwa waktu *load* untuk halaman berupa tulisan sangat kecil sekali. Waktu *load* yang besar ada pada halaman yang mempunyai fungsi manipulasi data dalam tabel database yaitu penambahan, update dan penghapusan data. Halaman yang mempunyai fungsi penampilan tidak membutuhkan waktu yang lama.



**Gambar 4. 16** Grafik hasil pengujian waktu load halaman website dalam satuan waktu ( $\mu$ s)

Sajian data-data pada pengujian ini baik pada data mentah serta data rata-rata total yang dibuat grafik menunjukkan bahwa waktu *load* halaman web baik yang dilakukan secara local, dengan komputer dengan spesifikasi yang canggih maupun tidak mempunyai nilai yang tidak begitu terpaut jauh. Halaman tulisan mempunyai waktu *load* rata-rata 1143,03  $\mu$ s, halaman Lihat Data 29798,89  $\mu$ s, halaman Tambah Data 106757,76  $\mu$ s, halaman Edit Data 9451,15  $\mu$ s, halaman Update Data 128128,39  $\mu$ s, halaman Delete(1):Konfirmasi 7021,19  $\mu$ s, dan halaman Delete(2): Hapus 115213,89  $\mu$ s. Sehingga dengan demikian waktu *load* halaman-halaman berupa tulisan adalah 1143,03  $\mu$ s, halaman-halaman sajian tabel (Lihat Data, Edit Data, dan Delete(1): Konfirmasi) adalah 154323,74  $\mu$ s, sedangkan untuk halaman-halaman modifikasi tabel (Tambah Data, Update Data, dan Delete(2): Hapus) adalah 116700,01  $\mu$ s.

Sehingga dapat disimpulkan bahwa waktu *load* halaman web tidak bergantung pada spesifikasi komputernya melainkan dari bagaimana server menerima kemudian memproses *request* dan mem-*forward*-nya ke komputer client. Asumsi lain adalah *traffic* dalam jaringan pada saat server dan client saling terkoneksi.

Selain itu jugabeban server untuk pemrosesan *request* dapat diprtimbangan. Namun demikan pengujian beban pada jaringan dan server tidak dilakukan.

#### 4.6 Perbandingan Dengan Sistem RFID Lain Yang Sudah Ada

Untuk membedakan sistem ini dengan sistem RFID lain yang sudah ada dilakukan studi evaluasi perbandingan. Perbandingan dilakukan dengan mempelajari secara singkat karya tulis mengenai sistem berbasis RFID yang penulis ketahui. Karya tulis yang diambil adalah karya tulis yang dibuat oleh beberapa mahasiswa Departemen Teknik Elektro FTUI yang terdokumentasi secara online.

Karya tulis yang dijadikan pembanding adalah:

**FHS2008** – Fadhly H. Saputra, “*Sistem Absensi Menggunakan Teknologi RFID*”.

**HER2008** – Heranudin, “*Rancang Bangun Keamanan Ruangn Menggunakan Radio Frequency Identification (RFID) Berbasis Mikrokontroller AT89C51*”.

**MAH2008** – Mahadhir, “*Rancang Bangun Sistem Identifikasi Kendaraan Pada Akses Masuk Menggunakan Teknologi RFID*”.

**AAD2009** – Adytiawan A. Dwitama, “*Perancangan dan Implementasi Sistem Parkir Berbasis RFID Dengan Menggunakan Antarmuka Java dan Basis Data MySQL Untuk Diimplementasikan Pada Lingkungan Parkir FTUI*”.

**TAZ2009** – Tommy A. Zairi, “*Rancang Bangun Sistem Parkir Otomatis Berbasis Teknologi RFID Reader DL-910 dan Tag Pasif EPC Gen2 Dengan Fitur Sistem Debit Biaya Parkir Via SMS*”.

Sedangkan sistem ini mempunyai kode **ZUW2009**.

**Tabel 4. 25 Perbandingan dengan Sistem RFID lain**

	<b>FHS2008</b>	<b>HER2008</b>	<b>MAH2008</b>	<b>AAD2009</b>	<b>TAZ2009</b>	<b>ZUW2009</b>
<b>Aplikasi Sistem</b>	Sistem Absensi	Sistem Keamanan Ruang	Identifikasi Kendaraan	Sistem Parkir	Sistem Parkir	Sistem Perpustakaan
<b>Basis Komunikasi</b>	Serial	Serial	Serial	Serial	Serial	TCP/IP
<b>Tipe Pembaca RFID</b>	ID-12	PF 5210	PF 5210	ID-12	DL-910	ID-12
<b>Tipe Tag</b>	Passive tag Mifare UL (Ultralight)	N/A	Tag aktif PF-300	Tag pasif ukuran ID-1	Tag pasif EPC Gen2	Tag pasif ukuran ID-1
<b>Hardware Tambahan</b>	Mikrokontroler AT89S52	Mikrokontroler AT89C51, Sensor magnetic, passive infra-red, laser beam	Mikrokontroler Atmega8535, LDR	N/A	Mikrokontroler Atmega8538, Motor Servo	WIZ110SR
<b>Bahasa Pemrograman</b>	Delphi 5	N/A	Visual Basic	Java	Delphi 7	Java, JSP
<b>Tipe Program</b>	Aplikasi	N/A	Aplikasi	Aplikasi	Aplikasi	Applet
<b>Fitur Tambahan</b>	N/A	SMS, Sensor	Sensor	N/A	SMS	Web

Perbandingan antara sistem ini dengan sistem lain ditunjukkan pada Tabel 4.25. Sistem ini adalah sistem yang menggunakan protokol komunikasi TCP/IP. Konversi protokol dari serial ke TCP menggunakan modul WIZ110SR. Pembaca RFID yang digunakan sederhana dengan kemampuan baca maksimal 12 cm. Tag yang dipakai adalah tag pasif berukuran ID-1 (berdimensi 85.60 × 53.98 mm). Program berupa applet agar dapat disatukan ke dalam web yang dikembangkan dengan JSP.

Hal yang membuat sistem ini berbeda dengan sistem lain adalah basis komunikasinya yang menggunakan protokol TCP/IP. Kemudian tipe program yang berupa applet yang disatukan ke dalam halaman web. Modifikasi data dalam database sendiri dilakukan dengan menggunakan web tersebut. Fokus utama sistem ini adalah pada applet sehingga web yang dibuat sederhana. Studi tambahan untuk perbandingan dengan sistem RFID lain yang dikembangkan pada institusi lain belum sempat dilakukan.

#### **4.7 Faktor Yang Patut Diperhatikan Dalam Pengembangan Sistem RFID**

Berdasarkan pengujian-pengujian yang dilakukan terhadap sistem yang menggunakan teknologi RFID disimpulkan bahwa terdapat beberapa faktor yang perlu diperhatikan. Faktor-faktor tersebut adalah:

1. Proses bisnis

Dimulai dengan menerjemahkan skenario kerja perangkat RFID. Sehingga didapat pemahaman untuk apa sistem tersebut dipakai, dimana sistem tersebut akan diimplementasikan, dan bagaimana proses yang dilakukan sistem tersebut. Kemudian dari penerjemahan tersebut dapat diketahui gambaran arsitektur sistem yang akan dibuat.

2. Kemampuan pembaca RFID dan tag

Faktor penting yang perlu diperhatikan adalah kemampuan RFID, terutama perangkat pembacanya. Hal ini berkaitan apakah dengan peralatan RFID yang ada dapat bekerja sesuai dengan kebutuhan skenarionya. Beberapa hal yang dijadikan pertimbangan adalah banyak tag yang dapat dideteksi dalam sekali pembacaan, jarak pembacaan antara tag dan pembaca, arah pembacaan, rata-rata error pembacaan, dan lain-lain.

3. Desain perangkat lunak

Perangkat lunak yang dibuat juga memperhatikan apakah sesuai dengan kebutuhan skenario sistem. Penggunaan bahasa pemrograman juga pertimbangan sendiri, apakah dalam bahasa pemrograman tersebut terdapat fitur untuk komunikasi dengan pembaca RFID. Aplikasi tersebut menangani apa dan kira-kira bagaimana *user* berinteraksi dengannya. Informasi yang menyertai tag juga penting. Dari informasi tersebut kita dapat mengetahui identitas barang yang ditempel tag itu. Informasi-informasi tersebut ditempatkan dalam suatu *database*.

4. Kompatibilitas perangkat keras dan perangkat lunak

Data yang dikirimkan pembaca RFID harus sampai ke perangkat lunak dengan benar. Oleh karena itu, komunikasi antara pembaca RFID dan perangkat lunak penting dalam pengembangan sistem. Jangan sampai karena kesalahan protokol komunikasi menyebabkan sistem tidak bekerja sesuai dengan yang diharapkan.

## **BAB 5 KESIMPULAN**

1. Sistem terdiri atas dua bagian yaitu sisi perangkat keras dan sisi perangkat lunak.
2. Sisi perangkat keras berupa sistem RFID yang menggunakan protokol komunikasi data TCP/IP.
3. Sisi perangkat lunak dikembangkan dengan Java applet untuk komunikasi dan pemrosesan data serta JSP untuk website dan MySQL untuk database.
4. Komunikasi antara server dan unit-unit pembaca RFID menggunakan koneksi socket.
5. Modul pembaca RFID ID-12 diperkirakan mempunyai pola radiasi dipole dengan jangkauan radiasi paling jauh pada arah y.
6. Tag RFID juga diperkirakan mempunyai pola radiasi dipole.
7. Pembaca RFID ID-12 memerlukan waktu untuk melakukan pembacaan berikutnya dengan tag yang sama.
8. Ketika dua tag dibuat berhimpit maka tag yang terbaca adalah tag yang lebih dekat ke pembaca RFID
9. Dua tag berhimpit yang jarak sangat kecil hingga 0 cm sama sekali tidak terbaca oleh pembaca RFID.
10. Ketika pembaca RFID telah membaca suatu tag, maka pembaca RFID tidak dapat membaca tag baru selagi tag sebelumnya masih berada pada jangkauan baca pembaca RFID.
11. Pada kedua pembaca RFID yang berhadapan terdapat jarak antar pembaca RFID minimum agar tag yang berada diantara kedua pembaca tersebut dapat terbaca.
12. Notifikasi dari applet yang sudah ada dan belum ditutup akan di-*override* oleh notifikasi data RFID yang masuk dari pembaca RFID yang berbeda.
13. Notifikasi dari applet untuk masuknya data dari pembaca RFID yang sama akan berurutan sesuai dengan urutan pembacaan tag secara fisik.
14. Penyimpanan log lebih baik dalam bentuk file ketimbang dalam bentuk database.

15. Pada pemrograman Java, pengukuran waktu untuk mengukur kode program lebih baik menggunakan fungsi perhitungan waktu yang menggunakan satuan *nanosecond*.
16. Waktu rata-rata tanggap applet total adalah 15109286,38  $\mu$ s.
17. Waktu *load* halaman web tidak bergantung pada spesifikasi komputernya melainkan dari bagaimana server menerima kemudian memproses *request* dan mem-*forward*-nya ke komputer client.
18. Pada web sistem ini, waktu *load* rata-rata total semua halaman adalah 56787,76  $\mu$ s.
19. Hal yang membuat sistem ini berbeda dengan sistem lain adalah basis komunikasinya yang menggunakan protokol TCP/IP. Kemudian tipe program yang berupa applet yang disatukan ke dalam halaman web.
20. Faktor-faktor yang perlu dipertimbangkan dalam pengembangan sistem RFID adalah proses bisnis, kemampuan pembaca RFID dan tag, desain perangkat lunak, serta kompatibilitas antara bagian perangkat keras dan perangkat lunak.

## DAFTAR ACUAN

- [1] Lewis, Steve “a basic introduction to rfid technology and its use in the supply chain” [http://www.ship2save.com/page\\_images/wp\\_printronic\\_rfid\\_supplychain.pdf](http://www.ship2save.com/page_images/wp_printronic_rfid_supplychain.pdf) Akses terakhir Juni 2009
- [2] Anonim “What is RFID?” [http://www.aimglobal.org/technologies/rfid/what\\_is\\_rfid.asp](http://www.aimglobal.org/technologies/rfid/what_is_rfid.asp) Akses terakhir: 01 Juni 2009
- [3] Ward, Matt & van Kranenburg, Rob “RFID: Frequency, standards, adoption and innovation” JISC Technology and Standards Watch, May 2006 [www.rfidconsultation.eu/docs/ficheiros/TSW0602.pdf](http://www.rfidconsultation.eu/docs/ficheiros/TSW0602.pdf) Akses terakhir: 04 Januari 2009
- [4] Engels, D.W., & Sarma, S. E. 2005. *Standardization of requirements within the RFID class structure framework. Technical Report: Auto-ID Center*. Cambridge, Massachusetts. <http://www.autoidlabs.org/uploads/media/AUTOIDLABS-WP-SWNET-011.pdf> Akses terakhir: 04 Januari 2009
- [5] Cheekiralla, S., & Engels, D. W. 2005. *A functional taxonomy of wireless sensor network devices*. Proceedings of the 2nd International Conference on Broadband Networks, pp. 26–33. IEEE. <http://ieeexplore.ieee.org/stampPDF/getPDF.jsp?tp=&arnumber=01589707&isnumber=33500> Akses terakhir 04 Januari 2009
- [6] IDTechEx. 2005. Active RFID: a profitable business. IDTechEx website. 12th December 2005. IDTechEx Ltd: Cambridge, <http://www.idtechex.com/products/en/articles/00000396.asp> Akses terakhir 04 Oktober 2006.
- [7] Bhatt, Himanshu & Glover, Bill “RFID Essentials” O’Reilly, January 2006, ISBN: 0-596-00944-5
- [8] *WIZ110SR User’s Manual* <http://www.wiznet.co.kr> Akses terakhir: 04 Januari 2009
- [9] Sun Microsystems, “The Java Tutorial – A Practical Guide for Programmers”, <http://java.sun.com/docs/books/tutorial> Update terakhir 14 Maret 2008.
- [10] Wijono, S.R., Suharto, B.H., & Wijono, M.S. “Pemrograman Java Servlet dan JSP dengan NetBeans” Penerbit ANDI, Yogyakarta 2007, ISBN: 978-979-29-0077-4.

**DAFTAR PUSTAKA**

Wijono, S.R., Suharto, B.H., & Wijono, M.S. *"Pemrograman Java Servlet dan JSP dengan NetBeans"* Penerbit ANDI, Yogyakarta 2007, ISBN: 978-979-29-0077-4

Hartati, G.S., Suharto, B.H., & Wijono, M.S. *"Pemrograman GUI Swing Java dengan NetBeans 5"* Penerbit ANDI, Yogyakarta 2007 ISBN: 979-763-646-1



## LAMPIRAN

### Data Percobaan Lengkap Pengujian Waktu *load* Halaman Web

#### 1. Halaman Tulisan

	#	local		c1		c2	
		(ms)	( $\mu$ s)	(ms)	( $\mu$ s)	(ms)	( $\mu$ s)
i n d e x	1	0	1901	0	1359	0	1309
	2	0	1147	0	1478	0	1228
	3	0	1603	0	1347	0	1136
	4	0	2267	0	1385	0	1150
	5	0	1237	0	1198	0	1164
	<b>t AVG</b>	0	1631	0	1353,4	0	1197,4
h e l p	1	0	779	0	1206	0	1259
	2	0	1064	0	880	0	778
	3	0	821	0	1100	0	759
	4	0	775	0	807	0	750
	5	0	790	0	788	0	826
	<b>t AVG</b>	0	845,8	0	956,2	0	874,4

## 2. Lihat Data

	#	local		c1		c2	
		(ms)	( $\mu$ s)	(ms)	( $\mu$ s)	(ms)	( $\mu$ s)
B u k u	1	16	16184	703	701392	16	19147
	2	16	15296	31	21264	16	15903
	3	16	15674	15	18639	16	16226
	4	15	16209	15	19006	16	14270
	5	15	15129	16	16562	15	14649
	<b>t AVG</b>	15,6	15698,4	156	155372,6	15,8	16039
D B a u t k a u	1	15	15540	16	24362	15	15506
	2	16	14865	16	14296	16	12964
	3	16	13894	16	14582	16	13142
	4	15	17107	15	14892	16	12910
	5	16	15177	0	13726	78	81012
	<b>t AVG</b>	15,6	15316,6	12,6	16371,6	28,2	27106,8
P e n i e r	1	0	10727	16	7678	0	7083
	2	16	10060	15	9334	0	12015
	3	16	11384	0	8424	0	11745
	4	16	10952	15	9682	0	8085
	5	16	10930	15	8624	15	8517
	<b>t AVG</b>	12,8	10810,6	12,2	8748,4	3	9489
T a g	1	16	17658	16	11638	110	111209
	2	32	18541	15	8667	15	9244
	3	15	17619	0	7076	0	8463
	4	16	15945	0	9491	16	8692
	5	15	16101	0	9081	15	9356
	<b>t AVG</b>	18,8	17172,8	6,2	9190,6	31,2	29392,8
S t a t u s	1	16	11997	16	6336	15	11552
	2	15	15016	16	8370	15	10119
	3	0	8906	0	6863	16	8949
	4	31	27945	0	8154	0	8823
	5	16	7963	0	6309	16	10055
	<b>t AVG</b>	15,6	14365,4	6,4	7206,4	12,4	9899,6
R a k	1	0	12477	0	6948	15	11552
	2	15	10406	0	9470	15	10119
	3	16	10649	0	8304	16	8949
	4	16	12385	0	8579	0	8823
	5	0	12165	0	8229	16	10055
	<b>t AVG</b>	9,4	11616,4	0	8306	12,4	9899,6
L o g	1	78	78911	32	24434	109	119391
	2	62	61996	0	11495	31	39449
	3	16	24880	16	7031	79	77726
	4	16	7822	0	8936	16	22946
	5	15	10068	15	22704	16	9507
	<b>t AVG</b>	37,4	36735,4	12,6	14920	50,2	53803,8

## 3. Tambah Data

	#	local		c1		c2	
		(ms)	( $\mu$ s)	(ms)	( $\mu$ s)	(ms)	( $\mu$ s)
<b>B u k u</b>	1	62	55976	47	49535	235	228269
	2	78	83484	63	75935	78	74060
	3	47	57938	62	73924	78	79145
	4	47	48270	62	76377	78	74638
	5	47	50941	62	72107	79	80984
	<b>t AVG</b>	56,2	59321,8	59,2	69575,6	109,6	107419,2
<b>D B a u t k a u</b>	1	46	43126	78	80299	78	82710
	2	63	56922	62	73208	78	81139
	3	47	44060	94	82364	78	77721
	4	62	67575	219	219678	31	25863
	5	16	21706	78	80343	78	80196
	<b>t AVG</b>	46,8	46677,8	106,2	107178,4	68,6	69525,8
<b>P e b n i e r</b>	1	47	41751	78	78630	79	73377
	2	47	51662	141	149982	78	75297
	3	47	52504	218	223814	78	82510
	4	63	70775	235	231091	47	49279
	5	46	45130	15	14472	312	310372
	<b>t AVG</b>	50	52364,4	137,4	139597,8	118,8	118167
<b>T a g</b>	1	94	90533	63	72026	15	25694
	2	62	53840	78	76935	94	79738
	3	62	52571	78	82130	47	40331
	4	79	70783	234	230987	78	81855
	5	47	55235	297	301971	16	28593
	<b>t AVG</b>	68,8	64592,4	150	152809,8	50	51242,2
<b>S t a t u s</b>	1	93	98549	46	38647	297	299947
	2	63	65634	79	77772	63	72971
	3	47	52724	31	34472	78	74168
	4	62	62755	32	40048	218	220692
	5	47	42538	62	72934	94	100610
	<b>t AVG</b>	62,4	64440	50	52774,6	150	153677,6
<b>R a k</b>	1	94	83833	78	76896	79	76701
	2	47	48939	78	80693	172	168540
	3	78	92066	78	73589	329	327406
	4	94	83007	47	51647	46	56069
	5	47	48085	31	31598	1766	1762308
	<b>t AVG</b>	72	71186	62,4	62884,6	478,4	478204,8

## 4. Edit Data

	#	local		c1		c2	
		(ms)	( $\mu$ s)	(ms)	( $\mu$ s)	(ms)	( $\mu$ s)
B u k u	1	0	13701	15	14411	16	12913
	2	0	12403	15	11362	15	10625
	3	16	12499	15	13987	0	10471
	4	16	12999	16	11261	0	10546
	5	0	12135	16	11301	0	10264
	<b>t AVG</b>	6,4	12747,4	15,4	12464,4	6,2	10963,8
D B a u t k a u	1	15	10236	16	9384	16	10023
	2	16	10501	15	8701	15	8766
	3	16	12838	15	10271	0	8909
	4	16	11147	16	9813	0	9240
	5	0	10348	0	8386	16	8598
	<b>t AVG</b>	12,6	11014	12,4	9311	9,4	9107,2
P e b n i e r	1	0	8346	0	6301	15	7549
	2	16	8321	15	7925	0	6106
	3	16	7909	0	6530	0	5749
	4	15	8414	0	7272	0	6168
	5	16	7756	0	5640	0	6047
	<b>t AVG</b>	12,6	8149,2	3	6733,6	3	6323,8
T a g	1	16	10867	16	6675	0	8109
	2	16	10222	16	8813	0	9327
	3	15	8569	16	8680	0	7573
	4	16	8877	0	6643	0	7201
	5	16	10085	0	10278	0	8495
	<b>t AVG</b>	15,8	9724	9,6	8217,8	0	8141
S t a t u s	1	0	7904	16	5553	0	6928
	2	15	10208	15	5645	0	5597
	3	16	7664	16	5677	0	7398
	4	16	8087	0	6946	0	5527
	5	0	7585	16	6203	16	5798
	<b>t AVG</b>	9,4	8289,6	12,6	6004,8	3,2	6249,6
R a k	1	16	9460	16	5874	0	5908
	2	16	7057	0	5590	16	6941
	3	15	7147	16	8094	16	5582
	4	15	7540	0	7533	16	5786
	5	16	91266	0	5491	0	5629
	<b>t AVG</b>	15,6	24494	6,4	6516,4	9,6	5969,2

## 5. Update Data

	#	local		c1		c2	
		(ms)	( $\mu$ s)	(ms)	( $\mu$ s)	(ms)	( $\mu$ s)
B u k u	1	109	99476	31	32808	609	606383
	2	94	85222	297	287247	62	56635
	3	62	69493	79	78273	31	43213
	4	78	79783	94	85340	78	73229
	5	47	47440	235	234493	78	71618
	<b>t AVG</b>	78	76282,8	147,2	143632,2	171,6	170215,6
D B a u t k a u	1	94	100175	78	81800	422	432067
	2	31	33099	78	74684	47	48492
	3	63	67782	297	295070	406	414904
	4	47	45063	79	82470	141	133813
	5	63	51593	78	76068	172	180856
	<b>t AVG</b>	59,6	59542,4	122	122018,4	237,6	242026,4
P e b n i e r	1	47	51038	297	298079	47	36711
	2	94	93153	234	228342	62	61217
	3	78	74266	297	292164	94	93347
	4	63	57353	218	226512	406	405402
	5	31	44098	78	82829	63	56630
	<b>t AVG</b>	62,6	63981,6	224,8	225585,2	134,4	130661,4
T a g	1	63	65299	78	78353	125	117760
	2	46	60108	78	77022	63	55331
	3	78	76654	219	230838	234	239158
	4	78	77664	78	72855	125	126157
	5	141	133141	78	76795	390	394989
	<b>t AVG</b>	81,2	82573,2	106,2	107172,6	187,4	186679
S t a t u s	1	297	284507	78	73338	31	30118
	2	79	82340	94	83024	16	23261
	3	297	296272	78	80069	47	39030
	4	78	81041	79	80880	31	37133
	5	78	75647	78	79264	78	77529
	<b>t AVG</b>	165,8	163961,4	81,4	79315	40,6	41414,2
R a k	1	109	116526	296	293939	78	77497
	2	62	60516	312	306959	78	79987
	3	125	133096	78	83308	47	38085
	4	422	425289	78	77983	93	82376
	5	109	116211	93	84594	79	79882
	<b>t AVG</b>	165,4	170327,6	171,4	169356,6	75	71565,4

## 5. Delete(1): Konfirmasi

	#	local		c1		c2	
		(ms)	( $\mu$ s)	(ms)	( $\mu$ s)	(ms)	( $\mu$ s)
B u k u	1	16	7735	0	8895	16	8732
	2	16	6977	0	6616	15	6149
	3	16	8024	0	6583	15	6367
	4	16	8025	0	6027	16	6203
	5	0	10602	15	6089	16	7217
	<b>t AVG</b>	12,8	8272,6	3	6842	15,6	6933,6
D B a u t k a u	1	0	5346	0	6372	0	6576
	2	15	7413	0	5935	0	6059
	3	16	5419	16	7265	0	6160
	4	0	7479	0	6863	16	5946
	5	16	9107	0	6428	16	7144
	<b>t AVG</b>	9,4	6952,8	3,2	6572,6	6,4	6377
P e b n i t e r	1	16	7544	15	7139	0	7055
	2	16	8437	0	6033	0	5532
	3	0	7647	0	6166	16	5521
	4	15	9705	0	5594	0	7306
	5	0	7745	0	6746	0	5891
	<b>t AVG</b>	9,4	8215,6	3	6335,6	3,2	6261
T a g	1	0	10270	16	7424	0	6105
	2	15	8222	0	6445	0	7534
	3	0	8032	16	6591	0	5943
	4	15	7961	15	6651	15	6452
	5	0	7796	0	7163	15	7606
	<b>t AVG</b>	6	8456,2	9,4	6854,8	6	6728
S t a t u s	1	0	7882	0	8418	0	5595
	2	0	9417	0	6131	15	5452
	3	16	7255	0	7452	16	5455
	4	16	6982	0	6335	0	6078
	5	0	8025	15	6094	16	5429
	<b>t AVG</b>	6,4	7912,2	3	6886	9,4	5601,8
R a k	1	0	7960	15	6841	16	5816
	2	0	7181	0	5840	0	6057
	3	16	8182	0	6097	16	6102
	4	16	10558	0	7560	16	5445
	5	0	8197	0	6927	0	7135
	<b>t AVG</b>	6,4	8415,6	3	6653	9,6	6111

## 6. Delete(2): Hapus

	#	local		c1		c2	
		(ms)	( $\mu$ s)	(ms)	( $\mu$ s)	(ms)	( $\mu$ s)
B u k u	1	437	439864	31	32481	109	104175
	2	297	303723	63	60282	218	224108
	3	250	257149	297	294920	78	77242
	4	390	392652	78	71812	47	43434
	5	141	136589	63	62505	78	77636
	<b>t AVG</b>	303	305995,4	106,4	104400	106	105319
D B a u t k a u	1	407	393236	313	306422	297	292772
	2	187	189465	328	316794	234	224369
	3	375	369315	266	279127	78	79381
	4	250	240166	94	91477	78	74230
	5	47	48018	93	86887	218	222448
	<b>t AVG</b>	253,2	248040	218,8	216141,4	181	178640
P e b n i t e r	1	296	302043	328	324011	78	74753
	2	31	33545	62	51522	78	76592
	3	32	23342	78	79449	15	23278
	4	63	63933	15	30316	78	76722
	5	16	20474	16	24627	79	76050
	<b>t AVG</b>	87,6	88667,4	99,8	101985	65,6	65479
T a g	1	78	66410	78	74441	78	75485
	2	47	52296	16	21787	78	74794
	3	31	43456	78	79598	297	292607
	4	31	32762	78	77470	16	26054
	5	47	41430	78	78538	78	81913
	<b>t AVG</b>	46,8	47270,8	65,6	66366,8	109,4	110170,6
S t a t u s	1	47	37492	78	79450	78	78854
	2	63	56646	78	76887	16	23781
	3	47	51238	78	74710	47	43937
	4	62	62032	250	254631	78	78188
	5	47	47038	78	80965	47	44482
	<b>t AVG</b>	53,2	50889,2	112,4	113328,6	53,2	53848,4
R a k	1	94	97288	94	80865	172	159764
	2	31	27914	78	79703	31	24434
	3	31	35509	63	75870	78	81747
	4	31	29447	15	22675	47	39923
	5	47	46016	266	260939	31	24448
	<b>t AVG</b>	46,8	47234,8	103,2	104010,4	71,8	66063,2