



**UNIVERSITAS INDONESIA**

**SIMULASI DAN ANALISIS SISTEM RFID MENGGUNAKAN  
*MULTI CHANNEL ANTI COLLISION* PROTOKOL**

**SKRIPSI**

**M RIZKI RUSTAM**

**0606074086**

**FAKULTAS TEKNIK  
PROGRAM STUDI TEKNIK ELEKTRO  
DEPOK  
JUNI 2011**



**UNIVERSITAS INDONESIA**

**SIMULASI DAN ANALISIS SISTEM RFID MENGGUNAKAN  
*MULTI CHANNEL ANTI COLLISION* PROTOKOL**

**SKRIPSI**

Diajukan sebagai salah satu syarat memperoleh gelar sarjana teknik

**M RIZKI RUSTAM**

**0606074086**

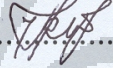
**FAKULTAS TEKNIK  
PROGRAM STUDI TEKNIK ELEKTRO  
DEPOK  
JUNI 2011**

HALAMAN PERNYATAAN ORISINALITAS

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

Nama : M Rizki Rustam

NPM : 0606074086

Tanda Tangan :  .....

Tanggal : Juni 2011

## HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :

Nama : M Rizki Rustam

NPM : 0606074086

Program Studi : Teknik Elektro

Judul Skripsi : Simulasi dan Analisis Sistem RFID Menggunakan *Multi Channel Anti Collision* Protokol

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

### DEWAN PENGUJI

Pembimbing : Dr. Ir. Muhamad Asvial, M.Eng



Penguji : Prof. Dr. Ir. Dadang Gunawan, M.Eng



Penguji : Dr. Ir. Arman Djohan Diponegoro



Ditetapkan di : Depok

Tanggal : Juni 2011

## KATA PENGANTAR/UCAPAN TERIMA KASIH

Puji syukur saya panjatkan kepada 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 Departemen 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) Allah SWT yang telah memberikan kekuatan kepada saya untuk menyelesaikan buku skripsi ini;
- (2) Dr. Ir. Muhamad Asvial, M.Eng selaku pembimbing yang telah yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan saya dalam penyusunan skripsi ini;
- (3) Orang tua dan keluarga saya yang selalu mendoakan yang terbaik untuk saya serta memberikan bantuan dukungan material dan moral.
- (4) Teguh Firmansyah, Syukron Zahri, Teddy Febriansyah, Fauzi Dwi, Rizky ATA, Rhyando, Muhammad Firdaus, Indra W Gumilang dan teman-teman di AMRG yang telah menyemangati, memotivasi, dan membantu saya dalam menyelesaikan skripsi ini.
- (5) Sahabat dan seluruh keluarga besar Civitas Akademika Fakultas Teknik Elektro khususnya angkatan 2006 Universitas Indonesia yang 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, Juni 2011

Penulis

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

---

---

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

Nama : M Rizki Rustam  
NPM : 0606074086  
Program Studi : Teknik Elektro  
Departemen : Teknik Elektro  
Fakultas : Teknik  
Jenis karya : Skripsi

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

**SIMULASI DAN ANALISIS SISTEM RFID MENGGUNAKAN *MULTI CHANNEL ANTI COLLISION* PROTOKOL**

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 : Juni 2011  
Yang menyatakan



( M Rizki Rustam )

## ABSTRAK

Nama : M Rizki Rustam  
Program Studi : Teknik Elektro  
Judul : Simulasi dan Analisis *Throughput* dan Efisiensi Sistem RFID Menggunakan *Multi Channel Anti Collision* Protokol.

*Radio Frequency Identification* (RFID) merupakan sistem identifikasi yang tidak memerlukan kontak secara langsung antara *reader* dan tag. Di dalam RFID terdapat berbagai macam permasalahan salah satunya ialah tumbukan antar *reader*. Masalah tumbukan antar *reader* merupakan salah satu masalah paling penting dalam sistem RFID. Di dalam skripsi ini dibahas tentang simulasi dan analisa data dari sebuah protokol *multi channel anti collision* yang bertujuan untuk mengurangi tumbukan yang terjadi antar *reader* sehingga didapatkan *throughput* dan efisiensi yang tinggi serta waktu tunggu yang kecil. Hasil simulasi menunjukkan bahwa untuk mencapai *throughput* yang tinggi diperlukan *reader* yang banyak sedangkan untuk mencapai efisiensi yang optimal hanya dibutuhkan sedikit *reader*. Untuk waktu tunggu sebuah *reader* dapat berkomunikasi kembali dengan tag setelah sebelumnya berhasil berkomunikasi dengan tag adalah makin besar jumlah *time slot* maka semakin kecil pula waktu tunggu yang dibutuhkan. Oleh karenanya, untuk menghasilkan waktu tunggu yang kecil dibutuhkan *time slot* yang banyak.

Kata kunci:

RFID, *reader*, tag, tumbukan, kanal data, kanal kontrol, *time slot*.

## ABSTRACT

Name : M Rizki Rustam  
Study Program : Teknik Elektro  
Title : Simulation and Analysis RFID Systems Using Multi-Channel Anti Collision Protocol

Radio Frequency Identification (RFID) is an identification system that does not require direct contact between reader and tag. RFID system has many problems one of them is collision between reader. In this paper, simulation and analysis of data from a multi-channel anti-collision protocols to reduce collisions between reader and to obtain a high throughput and efficiency and small waiting time is proposed. The simulation results show that a lot of number of readers are needed to achieve high throughputs and a few number of readers are needed to optimize efficiency. A reader can communicate back to tag after the previous communication to tags is success. The higher number of time slot is required to reduce the waiting time.

Keywords :

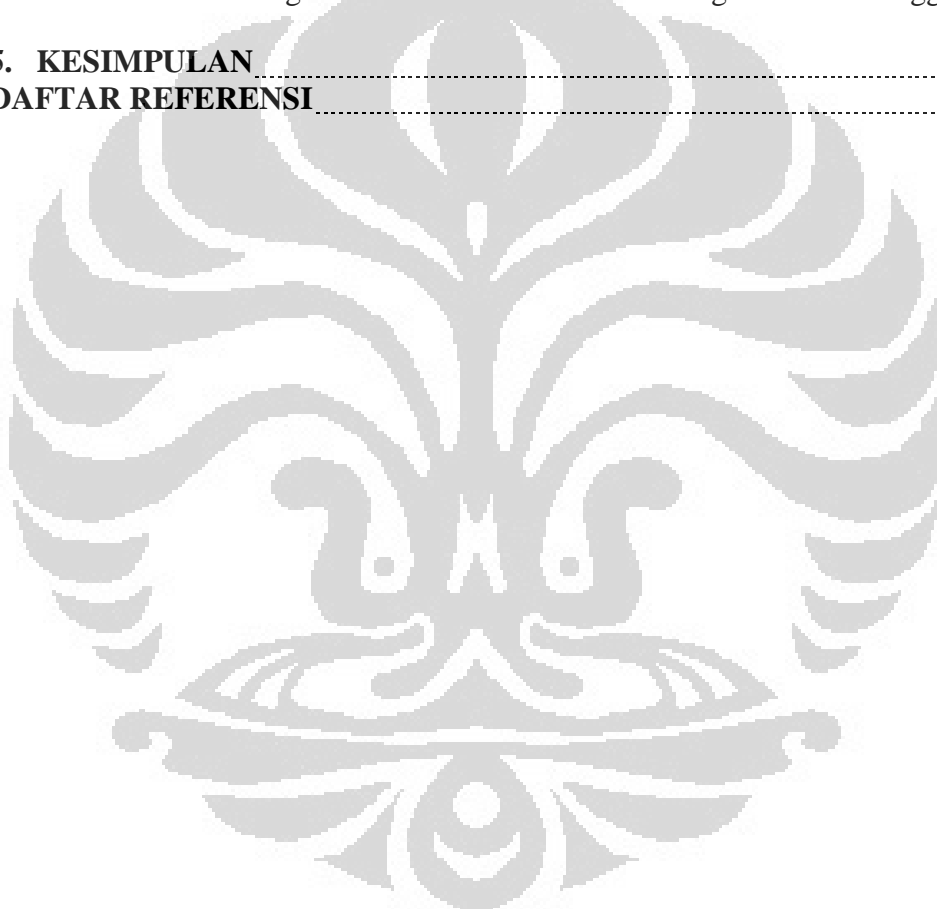
RFID, reader, tag, collision, data channel, control channel, time slot.



## DAFTAR ISI

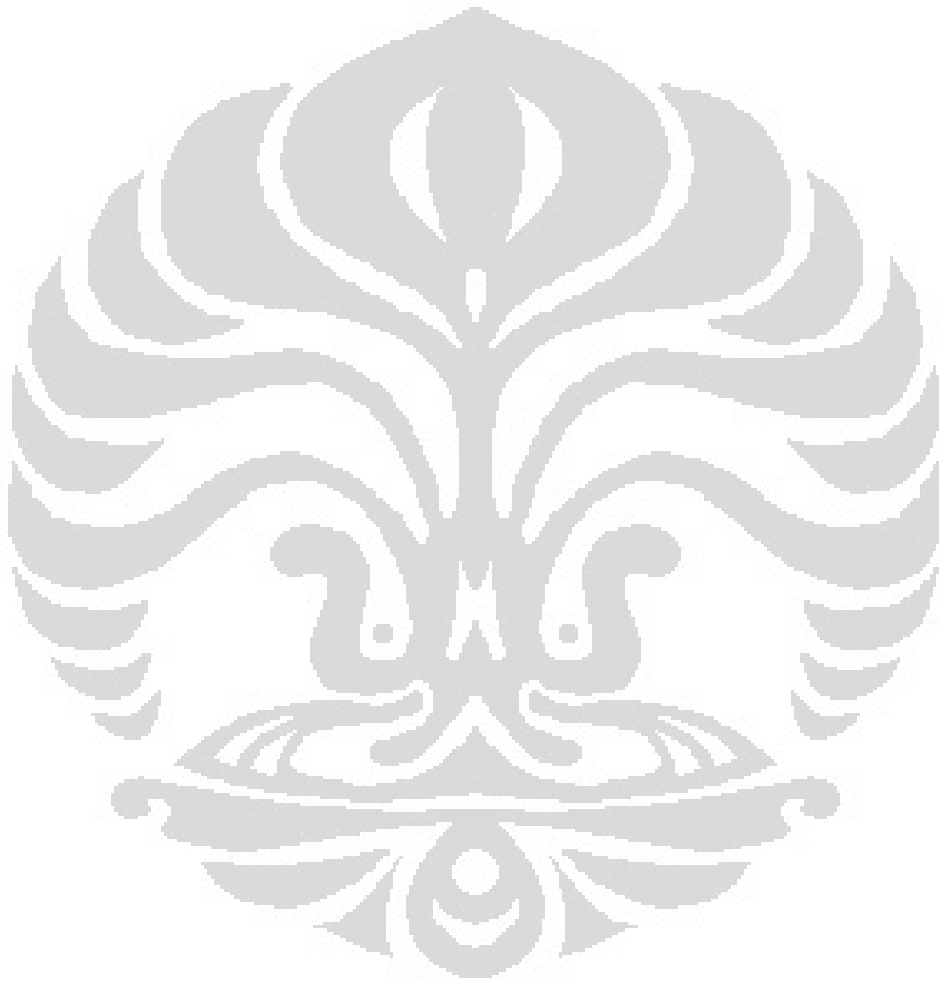
HALAMAN JUDUL.....	i
HALAMAN PERNYATAAN ORISINALITAS.....	iii
LEMBAR PENGESAHAN.....	iv
KATA PENGANTAR/UCAPAN TERIMA KASIH.....	v
LEMBAR PERSETUJUAN PUBLIKASI KARYA ILMIAH.....	vi
ABSTRAK.....	vii
ABSTRACT.....	viii
DAFTAR ISI.....	ix
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xii
DAFTAR LAMPIRAN.....	xiii
<b>1. PENDAHULUAN.....</b>	<b>1</b>
1.1. Latar Belakang.....	2
1.2. Tujuan Penulisan.....	2
1.3. Batasan Masalah.....	2
1.4. Sistematika Penulisan.....	2
<b>2. SISTEM RFID DAN ANTI COLLISION PROTOKOL.....</b>	<b>4</b>
2.1. Sistem RFID.....	4
2.2. Komponen RFID.....	4
2.2.1 Tag RFID.....	5
2.2.2 Reader RFID.....	7
2.2.3 Host Komputer.....	8
2.3. Jenis RFID.....	9
2.3.1 Berdasarkan Frekuensi.....	9
2.3.2 Berdasarkan Kemampuan Baca dan Ditulis.....	10
2.3.3 Berdasarkan Sumber Energi.....	10
2.4. Cara Kerja RFID.....	11
2.5. Tumbukan dalam Sistem RFID.....	13
2.5.1. Tumbukan <i>Reader</i> dengan <i>Reader</i> .....	14
2.5.2. Tumbukan <i>Reader</i> dengan <i>Tag</i> .....	15
2.6. Analisis Performansi pada <i>Multi Channel Anti Collision</i> Protokol Pada RFID.....	16
2.7.1 Throughput.....	16
2.7.2 Effisiensi.....	16
<b>3. PERANCANGAN SIMULASI <i>MULTI CHANNEL ANTI COLLISION</i> PROTOKOL PADA SISTEM RFID.....</b>	<b>18</b>
3.1. Konsep Dasar.....	18
3.2. Algoritma Simulasi.....	18
3.3. Eksponensial <i>Random Backoff</i> .....	22
3.3.1 <i>Truncated Exponential Backoff</i> .....	22
3.3.2 <i>Expected Backoff</i> .....	22
3.4. <i>Backoff Decrement Interval</i> (BDI).....	23
3.5. Variabel Simulasi.....	26
3.6. Kondisi yang Diamati.....	27

3.7. Rancangan Simulasi .....	28
<b>4. HASIL SIMULASI DAN ANALISA DATA</b>	
4.1 Parameter Simulasi .....	29
4.2 Pemodelan Simulasi .....	30
4.3 Keterbatasan Penelitian .....	31
4.4 Hasil Pengolahan Data dan Analisa .....	31
4.4.1 Hubungan Antara Jumlah <i>Reader</i> dengan <i>Throughput</i> .....	31
4.4.2 Hubungan Antara Jumlah <i>Reader</i> dengan Efisiensi .....	32
4.4.3 Hubungan Antara Jumlah <i>Reader</i> dengan Waktu Tunggu .....	33
4.4.4 Hubungan Antara Jumlah <i>Time Slot</i> dengan <i>Throughput</i> .....	34
4.4.5 Hubungan Antara Jumlah <i>Time Slot</i> dengan Efisiensi .....	36
4.4.6 Hubungan Antara Jumlah <i>Time Slot</i> dengan Waktu Tunggu .....	37
<b>5. KESIMPULAN</b> .....	39
<b>DAFTAR REFERENSI</b> .....	40



## DAFTAR TABEL

Tabel 2.1	Kategori Frekuensi RFID.....	9
Tabel 2.2	Sumber Energi RFID.....	11
Tabel 4.1	Parameter Simulasi Sistem.....	29



## DAFTAR GAMBAR

Gambar 2.1	Sistem RFID.....	4
Gambar 2.2	Komponen Dasar Sitem RFID.....	5
Gambar 2.3	RFID <i>Reader</i> dan Komponennya.....	8
Gambar 2.4	Spektrum Frekuensi Radio.....	9
Gambar 2.5	Komunikasi antara <i>Reader</i> dengan Tag.....	11
Gambar 2.6	Proses <i>Inductive Coupling</i> .....	12
Gambar 2.7	Tumbukan <i>Reader</i> dengan <i>Reader</i> .....	14
Gambar 2.8	Tumbukan <i>Reader</i> dengan <i>Tag</i> .....	15
Gambar 3.1	<i>Flowchart</i> Algoritma Simulasi.....	21
Gambar 3.2	Efek Transmisi BDI dari Berbagai Macam <i>Reader</i> .....	25
Gambar 3.3	BDI dari Satu Buah <i>Reader</i> ( <i>R1</i> ).....	25
Gambar 4.1	Grafik Hubungan <i>Throughput</i> dengan <i>Time Slot</i> yang Digunakan oleh Beberapa <i>Reader</i> .....	32
Gambar 4.2	Grafik Hubungan Efisiensi dengan <i>Time Slot</i> yang Digunakan oleh Beberapa <i>Reader</i> .....	33
Gambar 4.3	Grafik Hubungan Waktu Tunggu dengan <i>Time Slot</i> untuk Variasi Jumlah <i>Reader</i> .....	34
Gambar 4.4	Grafik Hubungan <i>Throughput</i> dengan <i>Reader</i> pada Variasi <i>Time Slot</i> Tertentu.....	35
Gambar 4.5	Grafik Hubungan Efisiensi dengan <i>Reader</i> untuk Variasi <i>Time Slot</i> .....	36
Gambar 4.6	Grafik Hubungan Waktu Tunggu dengan <i>Reader</i> untuk Variasi <i>Time Slot</i> .....	38

## BAB 1 PENDAHULUAN

### 1.1. Latar Belakang

Pada saat ini perkembangan teknologi radio berkembang amat cepat. Berbagai macam aplikasi dapat kita saksikan. Mulai dari urusan kesehatan, industri, hingga keamanan tercakup dalam teknologi ini. Teknologi yang dimaksud adalah RFID (*Radio Frequency Identification*). Dengan adanya teknologi ini berbagai kemudahan dapat dirasakan manusia. Banyak alat yang telah diciptakan yang berdasarkan teknologi ini.

RFID (*Radio Frequency Identification*) adalah teknologi identifikasi berbasis gelombang radio. Teknologi ini mampu mengidentifikasi berbagai objek secara simultan tanpa diperlukan kontak langsung (atau dalam jarak pendek). RFID dikembangkan sebagai pengganti atau penerus teknologi barcode. Implementasi RFID secara efektif digunakan pada lingkungan manufaktur atau industri dimana diperlukan akurasi dan kecepatan identifikasi objek dalam jumlah yang besar serta berada di area yang luas. RFID bekerja pada HF untuk aplikasi jarak dekat (*proximity*) dan bekerja pada UHF untuk aplikasi jarak jauh (*vicinity*).

Di dalam sistem RFID yang terdiri dari *reader* dan tag, *reader* sesuai dengan namanya berfungsi untuk membaca informasi yang terdapat pada tag yang mana proses pembacaan tersebut dilakukan tanpa kontak langsung. Namun, ketika proses pembacaan tersebut berlangsung banyak permasalahan yang muncul yang menyebabkan *throughput* atau banyaknya informasi yang berhasil dikirim dan diterima oleh *reader* menjadi kecil. Bahkan tidak hanya itu terkadang banyak ditemukan terjadinya kasus dimana tag tidak bisa dibaca oleh *reader*.

Salah satu hal yang menyebabkan hal tersebut adalah tumbukan (*collision*) yang terjadi di dalam sistem RFID. Baik itu antar *reader*, antar tag, maupun antara *reader* dengan tag. Permasalahan ini merupakan permasalahan serius yang harus dicarikan solusinya. Para peneliti telah berupaya untuk membuat protokol yang sederhana dan mudah diimplementasikan tetapi hasilnya tetap baik dalam rangka mengurangi tumbukan yang terjadi di dalam sistem RFID. Seperti jurnal [12] yang menjelaskan upaya pengurangan tumbukan dengan penggunaan *multi*

*channel*. Sedangkan jurnal [13] dan [14] menggunakan protokol yang lebih sederhana dibandingkan [12] untuk mendapatkan *throughput* yang tinggi. Begitu pun juga dengan jurnal [15] yang berupaya mengurangi tumbukan pada *mobile reader*.

Di dalam skripsi ini yang menjadi titik fokus pembahasan adalah tumbukan yang terjadi antar *reader*. Oleh karena itu, penulis membuat sebuah protokol yang mampu menghasilkan *throughput* dan efisiensi yang tinggi. Protokol yang penulis buat adalah *multi channel anti collision*.

## **1.2. Tujuan Penulisan**

Tujuan dari penulisan skripsi ini adalah mempelajari

1. Analisis *throughput* dan efisiensi yang tinggi pada sistem RFID
2. Mengembangkan protokol yang sederhana dan mudah diimplementasikan

## **1.3. Batasan Masalah**

Pembatasan dari skripsi ini adalah mensimulasikan protokol agar didapatkan *throughput* dan efisiensi yang tinggi.

## **1.4. Sistematika Penulisan**

Pembahasan yang dilakukan pada skripsi ini dibagi dalam beberapa tahapan sebagai berikut:

### **Bab I PENDAHULUAN**

Bagian ini terdiri dari latar belakang masalah, tujuan, batasan masalah, dan sistematika penulisan.

### **Bab 2 SISTEM RFID DAN ANTI COLLISION PROTOKOL**

Bagian ini akan berisi penjelasan tentang Sistem RFID, Komponen RFID, Jenis RFID, Cara Kerja RFID, Tumbukan dalam Sistem RFID, dan Analisis Performansi pada *Multi Channel Anti Collision* Protokol.

### Bab 3 PERANCANGAN SIMULASI *MULTI CHANNEL ANTI COLLISION* PROTOKOL PADA RFID

Bagian ini menjelaskan tentang konsep dasar protokol, algoritma simulasi, Eksponensial *Random Backoff* , *Backoff Decrement Interval* (BDI), Variabel Simulasi Kondisi yang Diamati dan Rancangan Simulasi.

### Bab 4 HASIL SIMULASI DAN ANALISA DATA

Bagian ini menjelaskan tentang hasil simulasi menggunakan MATLAB, pembahasan kinerjanya, dan keterbatasan yang terdapat didalamnya.

### Bab 5 KESIMPULAN

Bagian ini berisi kesimpulan dari simulasi dan analisa data.

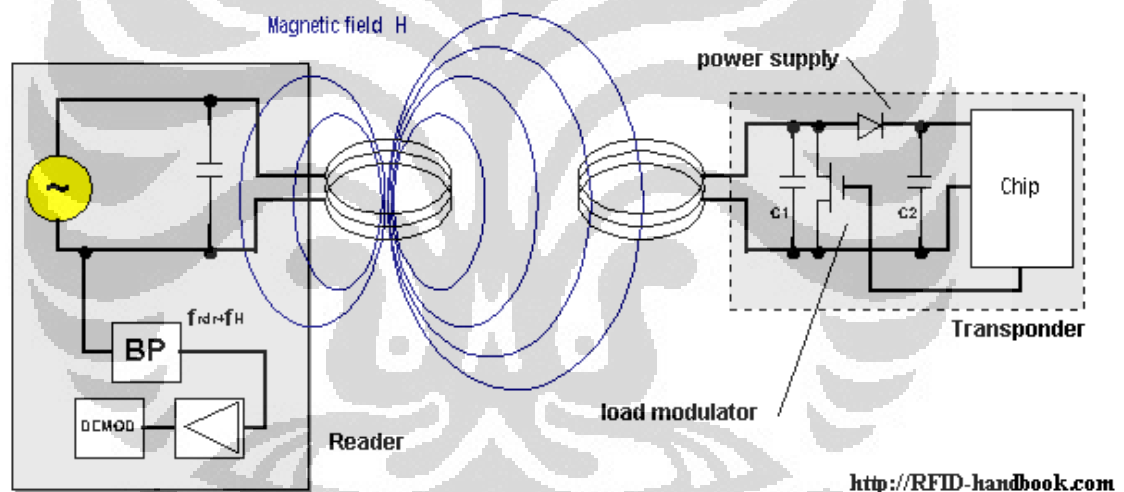


## BAB 2

### SISTEM RFID DAN ANTI COLLISION MAC PROTOKOL

#### 2.1 Sistem RFID

RFID (*Radio Frequency Identification*) adalah teknologi identifikasi berbasis gelombang radio. Teknologi ini mampu mengidentifikasi berbagai objek secara simultan tanpa diperlukan kontak langsung (atau dalam jarak pendek). RFID dikembangkan sebagai pengganti atau penerus teknologi barcode. Implementasi RFID secara efektif digunakan pada lingkungan manufaktur atau industri dimana diperlukan akurasi dan kecepatan identifikasi objek dalam jumlah yang besar serta berada di area yang luas [1].



Gambar 2.1. Sistem RFID [2]

#### 2.2 Komponen RFID

Sebuah sistem RFID menggunakan teknologi komunikasi radio nirkabel untuk mengidentifikasi objek. Ada tiga komponen dasar yang terdapat pada sistem RFID seperti tampak pada Gambar 2.2 [3]:

1. Tag (transponder)

Komponen ini disusun atas chip semikonduktor, antenna dan baterai.

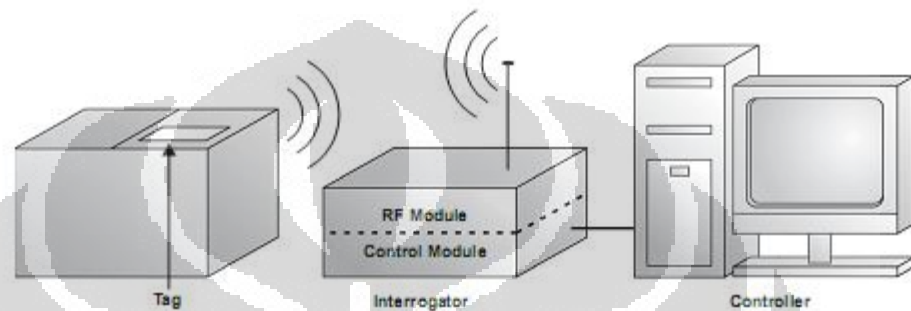
2. Interogator (*reader*)



Komponen ini disusun atas antena, modul elektronik RF, dan modul elektronik kontrol.

### 3. Kontroler (Host)

Komponen ini umumnya berupa personal komputer atau pusat kerja pengelolaan database dan perangkat lunak kontrol (sering disebut juga *middleware*)



Gambar 2.2. Komponen Dasar Sistem RFID [3]

#### 2.2.1 Tag RFID

Label RFID atau yang biasa disebut RFID tag sendiri, pada dasarnya merupakan suatu *microchip* berantena yang disertakan pada suatu unit barang. Dengan piranti ini, perusahaan bisa mengidentifikasi dan melacak keberadaan suatu produk. Seperti halnya *barcode*, yang memiliki *Universal Product Code* (UPC), sebuah tag RFID memiliki *Electronic Product Code* (EPC) berisi identitas produk tersebut, mulai dari nomor seri, tanggal produksi, lokasi manufaktur, bahkan tanggal kadaluarsa. EPC adalah identifikasi produk generasi baru, mirip dengan UPC atau *barcode*. Seperti halnya *barcode*, EPC terdiri dari angka-angka yang menunjukkan kode produsen, produk, versi dan nomor seri. Namun, EPC memiliki digit ekstra untuk mengidentifikasi item yang unik. Ukuran bit EPC yang mencapai 96-bit memungkinkannya secara unik mengidentifikasi lebih dari 268 juta produsen, masing-masing memiliki lebih dari satu juta jenis produk, sementara sisanya masih mencukupi untuk melabel seluruh produk

individualnya. Informasi EPC inilah yang tersimpan di dalam *chip* RFID [4].

RFID tag dapat bersifat aktif atau pasif.

### 1. RFID Pasif

RFID tag yang pasif tidak memiliki *power supply* sendiri. Dengan hanya berbekal induksi listrik yang ada pada antena yang disebabkan oleh adanya frekuensi radio *scanning* yang masuk, sudah cukup untuk memberi kekuatan yang cukup bagi RFID tag untuk mengirimkan respon balik. Sehubungan dengan power dan biaya, maka respon dari suatu RFID yang pasif biasanya sederhana, hanya nomor ID saja. Dengan tidak adanya *power supply* pada RFID tag yang pasif maka akan menyebabkan semakin kecilnya ukuran dari RFID tag yang mungkin dibuat. Beberapa RFID komersial yang saat ini sudah beredar di pasaran ada yang bisa diletakkan di bawah kulit. Pada tahun 2005 tercatat bahwa RFID tag terkecil berukuran 0.4 mm x 0.4 mm dan lebih tipis daripada selembar kertas. Dengan ukuran sekian maka secara praktis benda tersebut tidak akan terlihat oleh mata. RFID tag yang pasif ini memiliki jarak jangkauan yang berbeda mulai dari 10 mm sampai dengan 6 meter. RFID tag yang pasif harganya bisa lebih murah untuk diproduksi dan tidak bergantung pada baterai.

### 2. RFID aktif

RFID tag yang aktif, di sisi lain harus memiliki *power supply* sendiri dan memiliki jarak jangkauan yang lebih jauh. Memori yang dimilikinya juga lebih besar sehingga bisa menampung berbagai macam informasi didalamnya. Jarak jangkauan dari RFID tag yang aktif ini bisa sampai sekitar 100 meter dan dengan umur baterai yang bisa mencapai beberapa tahun lamanya [4].

Label RFID aktif (induktif) terdiri dari tiga bagian [5]:

- *Silicon microprocessor* – Ukuran *chip* ini sangat bervariasi, bergantung pada kegunaannya.

- Metal coil – Terbuat dari kabel tembaga atau aluminium yang akan berhubungan dengan transponder, koil ini berfungsi sebagai antena bagi tag yang akan mengirimkan sinyal kepada *reader*. Jarak baca ditentukan oleh ukuran antena koil yang dapat beroperasi pada frekuensi tertentu.
- *Encapsulating material* – gelas atau materi polimer yang membungkus *chip* dan koil. Label RFID aktif (induktif) menggunakan tenaga dari medan magnet yang diciptakan oleh *reader*. Koil menggunakan tenaga magnet dan berkomunikasi dengan *reader*. Tag tersebut memodifikasi dan mengontrol medan magnet untuk menerima dan mengirim pesan kepada *reader*. Data yang dikirimkan kembali kepada *reader*, diteruskan kepada komputer pusat.

RFID tag juga dapat dibedakan berdasarkan tipe memori yang dimilikinya :

a) *Read / Write* (Baca/Tulis)

Memori baca/tulis secara tidak langsung sama seperti namanya, memorinya dapat dibaca dan ditulis secara berulang-ulang. Data yang dimilikinya bersifat dinamis.

b) *Read only* (Hanya baca)

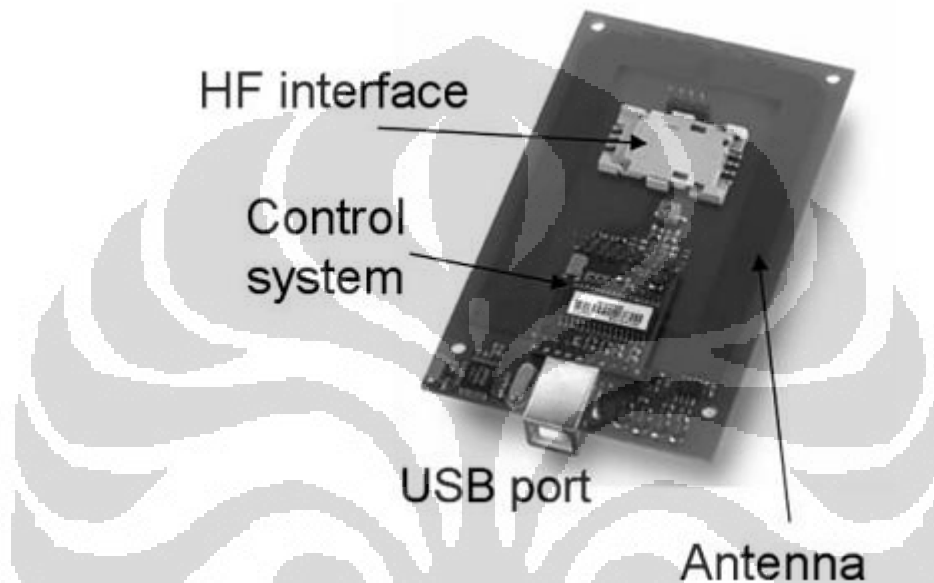
Tipe ini memiliki memori yang hanya diprogram pada saat tag ini dibuat dan setelah itu datanya tidak bisa diubah sama sekali. Data bersifat statis.

### 2.2.2 Reader RFID

*Reader* RFID, terdiri atas RFID-*reader* dan antena yang akan mempengaruhi jarak optimal identifikasi. Terminal RFID akan mengeluarkan gelombang *radio* dan menginduksi label RFID. Gelombang induksi tersebut memiliki kata kunci dan jika dikenali oleh label RFID maka memori dalam label RFID (*ID chip*) akan terbuka. Kemudian label RFID akan mengirimkan kode yang terdapat di memori *ID chip* melalui antena yang terpasang di label. RFID *reader* akan membandingkan kode yang diterima dengan kode kunci yang tersimpan di RFID *reader*. Jika sesuai,

RFID *reader* akan membuka kunci pintu. RFID *reader* akan membuat kode kunci yang baru. Kode baru ini akan disimpan ke memori RFID *reader* dan dikirimkan ke RFID tag yang akan disimpan di memori *ID chip*. Terminal RFID terhubung langsung dengan sistem host komputer [5].

Perhatikan *reader* RFID beserta komponennya yang terdapat pada Gambar 2.3.



Gambar 2.3. RFID *Reader* dan Komponennya [6]

### 2.2.3 Host Komputer

RFID Kontroler atau host komputer merupakan “otak” dari sistem RFID. Mereka digunakan pada jaringan RFID interrogator (*reader*) secara bersama-sama dan untuk memusatkan proses informasi yang ditangkap *reader* dari tag. Kontroler di berbagai jaringan sering disebut sebagai PC atau pusat pengelolaan *database* atau aplikasi *software* [3].

Host komputer merupakan sistem komputer yang mengatur alur informasi dari item-item yang terdeteksi dalam lingkup sistem RFID dan mengatur komunikasi antara label dan *reader*. Host bisa berupa komputer *stand-alone* maupun terhubung ke jaringan LAN / Internet untuk komunikasi dengan server [5].

## 2.3 Jenis RFID

RFID dikelompokkan menjadi beberapa jenis dengan berdasarkan [5]:

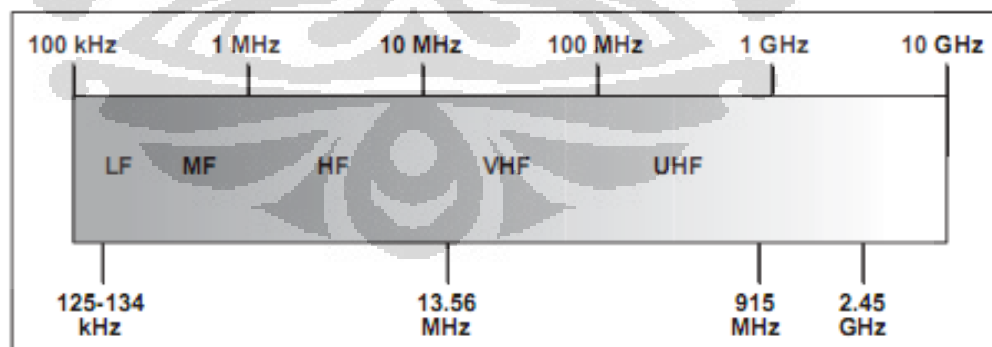
- a. Frekuensi
- b. Kemampuan dibaca dan ditulis
- c. Sumber energi

### 2.3.1 Berdasarkan Frekuensi

Berdasarkan frekuensi yang dipakai menggunakan label RFID maka RFID digolongkan menjadi empat bagian seperti yang ditunjukkan pada Tabel 2.1 dan Gambar 2.4.

Tabel 2.1. Kategori Frekuensi RFID [7]

Kode	Frekuensi	Range	RFID use
LF	<i>Low Frequency</i>	30 kHz to 300 kHz	125kHz
HF	<i>High Frequency</i>	3 MHz to 30 MHz	13,56 MHz
VHF	<i>Very High Frequency</i>	30 MHz to 300 MHz	<i>Not used for RFID</i>
UHF	<i>Ultra High Frequency</i>	300 MHz to 3 GHz	868 MHz, 915 MHz



Gambar 2.4. Spektrum Frekuensi Radio [3]

### 2.3.2 Berdasarkan Kemampuan Baca Tulis

Berdasarkan kemampuan dibaca dan ditulisnya RFID dikelompokkan sebagai berikut [5]:

- a. *Read Only* label berisi nomor unik yang tidak dapat dirubah
- b. *WORM Write Once Read Many* dimungkinkan untuk mengkodekan mengisi untuk pertama kali, dan kemudian data/kode tersebut terkunci dan tidak dapat dirubah.
- c. *Read/Write* dimungkinkan untuk mengisi dan memperbaharui informasi di dalamnya.
- d. Kombinasi *Read-only* dan *Read/Write* : sebagian data tersimpan secara permanent, sebagian sisanya dapat diakses, ditulis, dan diperbaharui datanya.

### 2.3.3 Berdasarkan Sumber Energi

Berdasarkan sumber energi terdapat tiga jenis label RFID dengan penggunaan yang berbeda. Ringkasan sumber energi RFID ditunjukkan pada Tabel 2.2.

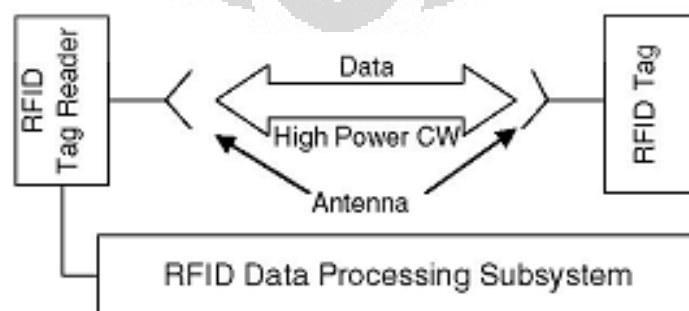
Tabel 2.2. Sumber Energi RFID [7]

Tipe	Aktif	Semi Pasif	Pasif
Karakteristik			
Sumber energi	Baterai pada label	Baterai untuk menjalankan chip. Energi gelombang <i>radio</i> dari <i>reader</i> untuk komunikasi hanya di dalam jangkauan <i>reader</i>	Energi gelombang <i>radio</i> dari <i>reader</i> untuk menjalankan chip dan komunikasi
Ketersediaan	Selalu ada	Rendah	Hanya di dalam

sinyal gelombang <i>radio</i>	100 feet		jangkauan <i>reader</i> , kurang dari 10 feet
Kekuatan sinyal	Tinggi	Rendah	Sangat rendah
Kebutuhan sinyal yang kuat	Sangat rendah		Sangat tinggi
Bidang penerapan	Berguna untuk label barang yang bernilai tinggi untuk discan dalam jarak jauh, misal mobil		Berguna untuk barang yang bervolume tinggi, dan bisa discan dalam arak dekat, misal perdagangan ritel

#### 2.4 Cara Kerja RFID

Sistem RFID terdiri atas komponen pokok yaitu *reader* dan tag. Sistem kerjanya adalah pertama *reader* mengirim sinyal perintah ke tag maka tag akan memproses sinyal perintah tersebut. Ketika tag menerima sinyal perintah tersebut maka ia mendapatkan energi dan rangkaian dalam yang terdapat di dalamnya akan teraktivasi. Di akhir proses, data atau informasi dari tag akan diterima oleh *reader*. Proses diatas dapat ditunjukkan oleh Gambar 2.5

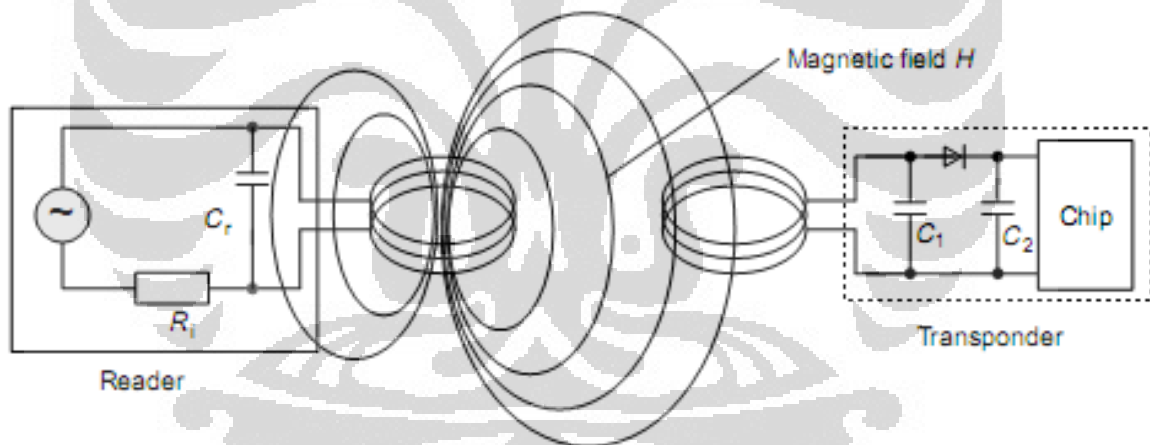


Gambar 2.5 Komunikasi antara *Reader* dengan Tag

Cara kerja dari sistem RFID dibagi menjadi dua bagian, yang pertama adalah proses transmisi *reader* ke tag sekaligus dengan pencatutan terhadap tag. Kemudian yang kedua adalah transmisi dari tag menuju *reader* [9].

a. Proses transmisi dari *reader* ke tag

Proses transmisi dari *reader* ke tag sekaligus pencatutan dilakukan dengan cara *inductive coupling*. *Inductive coupling* biasanya digunakan untuk tag yang bersifat pasif. Pada tag ini terdapat sebuah *microchip* tunggal dan sebuah kumparan yang berfungsi sebagai antena. Pada sistem ini semua daya yang digunakan untuk mengaktifkan mikrochip tag dan untuk mengirimkan sinyal kembali kepada *reader* disediakan oleh *reader*. Antena reader akan membangkitkan medan elektromagnetik yang akan menembus kumparan dari tag. Sistem ini akan lebih jelas seperti pada Gambar 2.6



Gambar 2.6. Proses *Inductive Coupling* [8]

Sebagian medan akan menembus kumparan tag yang letaknya berjauhan dengan *reader*. Dan selanjutnya akan menghasilkan tegangan karena proses induksi. Tegangan ini disearahkan dengan menggunakan diode pada rangkaian *chip*. Kemudian tegangan ini akan digunakan untuk mengaktifkan *chip*. Kapasitor juga dibutuhkan sebagai penghasil frekuensi resonansi yang sesuai dengan frekuensi *reader*.



b. Proses transfer data dari tag ke *reader*

Pada transfer data dari tag ke *reader* juga dilakukan mekanisme yang sama, yaitu menggunakan *inductive coupling*. *Signal feedback* dari tag ke *reader* akan direpresentasikan dengan impedansi pada kumparan antena *reader*. Perubahan impedansi pada antena akan membawa perubahan pada tegangan yang diterima oleh *reader*. Perubahan impedansi tersebut diatur oleh data yang dikirimkan oleh tag. Pengiriman ini disebut dengan *load modulation*. Sinyal yang masuk ke *reader* kemudian disearahkan dan hasilnya adalah berupa amplitudo dari sinyal yang ditransmisikan. Selanjutnya pada sinyal tersebut dilakukan kuantisasi sehingga dapat diketahui level dari data yang dikirimkan tersebut [9].

## 2.5 Tumbukan Dalam Sistem RFID

Setiap *reader* dari jaringan *reader* RFID memiliki keterbatasan zona baca-tulis. Hanya di dalam zona inilah *tag* dapat diakses oleh *reader* dan lebih jauh lagi dapat dibaca maupun ditulis. Zona tersebut di dalam sistem RFID disebut dengan zona interogasi (*interrogation zone*). Luasnya zona interogasi dipengaruhi oleh banyak faktor seperti karakteristik dari antena *reader* transmisi daya *reader*, kondisi kerja *reader*, dan karakteristik dari *tag* itu sendiri dan lain sebagainya. Ketika *reader* dalam jumlah banyak tersebar dalam daerah yang terbatas maka dua atau lebih zona interogasi dari masing-masing *reader* dapat mengalami *overlapped*. Bahkan andai pun zona interogasi tidak mengalami *overlapped*, interferensi pun dapat juga terjadi diantara *reader*. Interferensi tersebut disebabkan oleh frekuensi yang digunakan diantara *tag* dan *reader*. Dengan kata lain, interferensi yang terjadi diantara *reader* disebabkan oleh adanya operasi dari *reader* yang disebut dengan tumbukan (*collision*) *reader*.

Di dalam jurnal [10], tumbukan *reader* dijelaskan untuk pertama kalinya dan tumbukan *reader* dianalisis secara abstrak pada jurnal [11]. Kedua jurnal tersebut menjelaskan cukup detail tentang tumbukan *reader*.

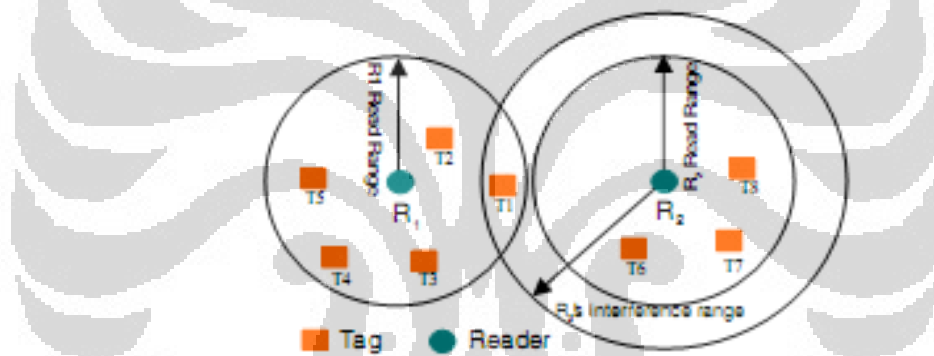
Tumbukan *reader* dapat menyebabkan operasi dalam sistem jaringan RFID menjadi salah sehingga menurunkan tingkat baca (*read rate*) dari sistem RFID. Oleh karena itu, mengurangi tumbukan *reader* menjadi masalah yang sangat

penting untuk dicarikan solusinya. Terlebih lagi dalam kasus tumbukan *reader* pada jaringan *reader* yang bergerak hal tersebut membuat sistem RFID menjadi lebih buruk [12].

Secara umum ada dua jenis tumbukan pada sisi *reader* yang terjadi pada sistem RFID yaitu, tumbukan *reader* dengan *reader* dan tumbukan *reader* dengan *tag*.

### 2.5.1 Tumbukan Reader dengan Reader

Ketika dua atau lebih *reader* bergerak (*mobile reader*) berada dalam jarak yang berdekatan, mereka akan berusaha membaca tag yang sama pada waktu dan frekuensi yang sama. Akibat hal tersebut, maka akan terjadi tumbukan *reader* dengan *reader* [13].



Gambar 2.7. Tumbukan Reader dengan Reader

Perhatikan Gambar 2.7, T1 berada dalam daerah interferensi *reader* R2. Sinyal refleksi yang diterima *reader* R1 dari *tag* T1 dapat dengan mudah terdistorsi akibat sinyal yang berasal dari *reader* R2. Tumbukan jenis ini dapat terjadi bahkan ketika jarak baca dari dua *reader* tidak *overlap*. Situasi tersebut terjadi dalam sistem RFID ketika ada transmisi yang tidak diinginkan berasal dari sebuah *reader* yang berjarak dekat mengganggu kemampuan *tag* untuk menguraikan sinyal yang diinginkan [14].

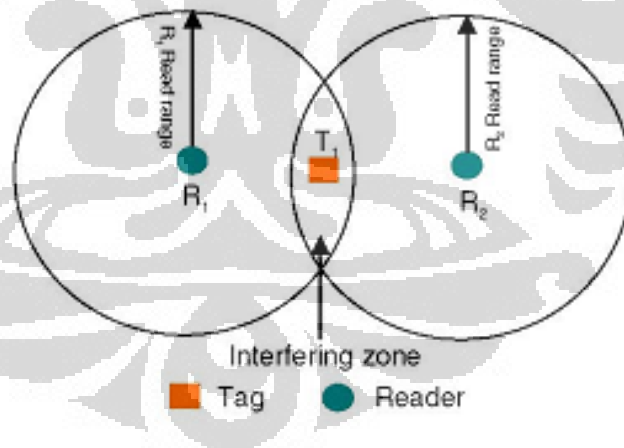
Dari Gambar 2.7 dapat juga dijelaskan bahwa ketika R1 hendak mengakses informasi atau berkomunikasi dengan T1, maka pada saat itu pula R2 menginterferensi R1. Akibat hal tersebut maka terjadilah tumbukan antar *reader* yakni antara R1 dengan R2 sehingga dengan adanya hal

tersebut informasi yang hendak dikirimkan dari T1 ke R1 mengalami distorsi atau dengan kata lain tidak sepenuhnya informasi yang dikirimkan dari T1 menuju R1 dapat dibaca dengan baik oleh R1. Pada gambar tersebut dapat juga kita lihat bahwa meskipun jarak baca (*read range*) dari kedua *reader*, R1 dan R2 tidak *overlap* tetap terjadi tumbukan antar *reader* dikarenakan jarak interferensi (*interference range*) dari R2 yang *overlap* terhadap jarak baca R1.

Jarak interferensi merupakan salah satu karakteristik yang dimiliki hampir setiap *reader* dan pada umumnya lebih besar dibandingkan dengan jarak bacanya.

### 2.5.2 Tumbukan *Reader* dengan *Tag*

Tumbukan *reader* dengan tag terjadi ketika sebuah tag mendengar banyak *reader* berupaya untuk mengakses pada waktu yang sama. Dalam situasi tersebut, tag tidak dapat merespon komunikasi atau akses yang dilakukan oleh *reader*.



Gambar 2.8. Tumbukan *Reader* dengan *Tag*

Pada Gambar 2.8, jarak baca dari dua *reader* R1 dan R2 mengalami *overlap*. Oleh karenanya, sinyal yang berasal dari R1 dan R2 menginterferensi tag T1. Dalam kasus tersebut, T1 tidak dapat mengartikan berbagai sinyal yang datang ke dirinya dan tag juga tidak dapat dibaca baik oleh R1 maupun oleh R2 [15].

Jikalau pada tumbukan *reader* dengan *reader* terjadi *overlap* antara jarak interferensi dengan jarak baca, maka pada tumbukan *reader* dengan tag yang terjadi *overlap* adalah antara jarak bacanya yaitu dalam hal ini adalah jarak baca R1 dengan R2 seperti tampak pada Gambar 2.8. Tumbukan *reader* dengan tag ini dapat terjadi akibat jarak yang amat dekat antara R1 dengan R2. Tumbukan ini menyebabkan tag T1 tidak dapat dibaca baik oleh R1 maupun oleh R2.

## 2.6 Analisis Performansi pada *Multi Channel Anti Collision* Protokol Pada RFID

### 2.6.1 *Throughput*

*Throughput* adalah banyaknya informasi yang berhasil dikirim dan diterima oleh *reader*. atau dengan kata lain *throughput* adalah jumlah rata-rata transmisi paket yang sukses per waktu interval tertentu [16].

Informasi dikatakan sukses dikirim jika ia dikirim oleh *reader* dan sukses diterima oleh semua *tag* di dalam jarak baca misalnya ia tidak mengalami benturan dengan informasi lainnya di dalam jaringan.

Dalam persamaan matematisnya *throughput* dapat digambarkan [15].

$$\textit{Throughput} = \frac{\text{Total informasi yang sukses dikirim oleh reader}}{\text{Total Waktu}} \dots(2.1)$$

### 2.6.2 *Effisiensi*

Effisiensi merefleksikan kemampuan protokol untuk mendeteksi kemungkinan terjadinya tumbukan pada *tag* dan oleh karenanya mencegah transmisi yang tidak diinginkan [15]. Effisiensi dinyatakan dalam persen.

Dalam persamaan matematisnya effisiensi dapat digambarkan.

$$\text{Effisiensi (\%)} = \frac{\text{Total informasi yang sukses dikirim oleh reader}}{\text{Total informasi yang dikirimkan (sukses + gagal) oleh reader}} \dots(2.2)$$

Makin besar jumlah informasi yang sukses dikirimkan, makin besar pula *throughput* dan oleh karenanya makin besar pula jumlah *tag* yang dapat diidentifikasi oleh *reader*. Peningkatan *throughput* mengindikasikan peningkatan dalam tingkat baca sedangkan peningkatan efisiensi mengindikasikan pengurangan tumbukan. Oleh sebab itu, *throughput* dan efisiensi memainkan peran yang penting dalam mendefinisikan keefektifan dari suatu protokol.



### BAB 3

## PERANCANGAN SIMULASI *MULTI CHANNEL ANTI COLLISION* PROTOKOL PADA SISTEM RFID

### 3.1 Konsep Dasar

Jika pada skripsi [9] bertitik fokus pada tumbukan antar tag dan berupaya untuk mengurangi tumbukan antar tag agar diperoleh *throughput* yang tinggi. Konsep dasar dari skripsi ini adalah bertitik fokus pada tumbukan yang terjadi antar *reader* ketika hendak mengakses tag dan berupaya untuk mengurangi tumbukan yang terjadi agar diperoleh *throughput* dan efisiensi yang tinggi.

Pada skripsi ini protokol yang digunakan adalah *multi channel anti collision*, yakni sebuah protokol yang merupakan hasil pengembangan dari [13] [14] dan [15]. Di dalam protokol ini terdapat dua buah kanal yang memainkan peranan yang penting dalam proses pengaksesan tag oleh *reader* yaitu, kanal data dan kanal kontrol. Kanal data merupakan kanal yang dipergunakan oleh *reader* untuk mengakses atau berkomunikasi dengan tag sedangkan kanal kontrol dipergunakan oleh *reader* untuk berkomunikasi dengan *reader* lain. Ketika *reader* sedang atau hendak berkomunikasi dengan tag maka *reader* akan mengirimkan sinyal melalui kanal kontrol kepada *reader* lain yang menandakan bahwa ia sedang atau hendak berkomunikasi dengan tag. Di dalam skripsi ini, penulis hanya memperhatikan sisi *reader*-nya saja karena tag tidak memberikan kontribusi apapun dalam tumbukan yang terjadi antar *reader*.

### 3.2 Algoritma Simulasi

Pada algoritma ini terdapat sinyal yang disebut dengan BEACON. Sinyal ini dibagi menjadi tiga yaitu BROADCAST, BUSY, dan END.

- a. Sinyal BROADCAST merupakan sebuah sinyal yang dikirimkan oleh *reader* melalui kanal kontrol kepada *reader* lain sebagai tanda bahwa *reader* hendak mengakses tag.
- b. Sinyal BUSY merupakan sinyal yang dikirimkan oleh *reader* yang sedang mengakses tag kepada *reader* lain melalui kanal kontrol yang menandakan bahwa ia sedang mengakses tag.

- c. Sinyal END merupakan sinyal yang dikirimkan oleh *reader* yang baru saja selesai mengakses tag.

Pada prinsipnya ketiga sinyal tersebut merupakan sinyal yang sama hanya berbeda istilah saja.

Di dalam algoritma ini, *reader* pertama-tama memasuki fase mendengar (*listening stage/LS*). Filosofinya ialah *reader* mendengar terlebih dahulu sebelum akhirnya berbicara seperti halnya pada skema LBT (*Listen Before Talk*) konvensional, sebuah skema *multiple access* yang berdasarkan pada skema CSMA. *Reader* yang ingin memulai komunikasi dengan tag memasuki fase mendengar (LS) terlebih dahulu selama  $T_{\min}$ . Jika *reader* menerima sinyal BROADCAST atau BUSY maka *reader* memasuki fase menunggu (*waiting stage/WS*). Setelah memasuki fase menunggu maka *reader* akan memasuki mekanisme *random backoff*, sebuah mekanisme untuk mencegah terjadinya tumbukan antar *reader* di kanal kontrol,

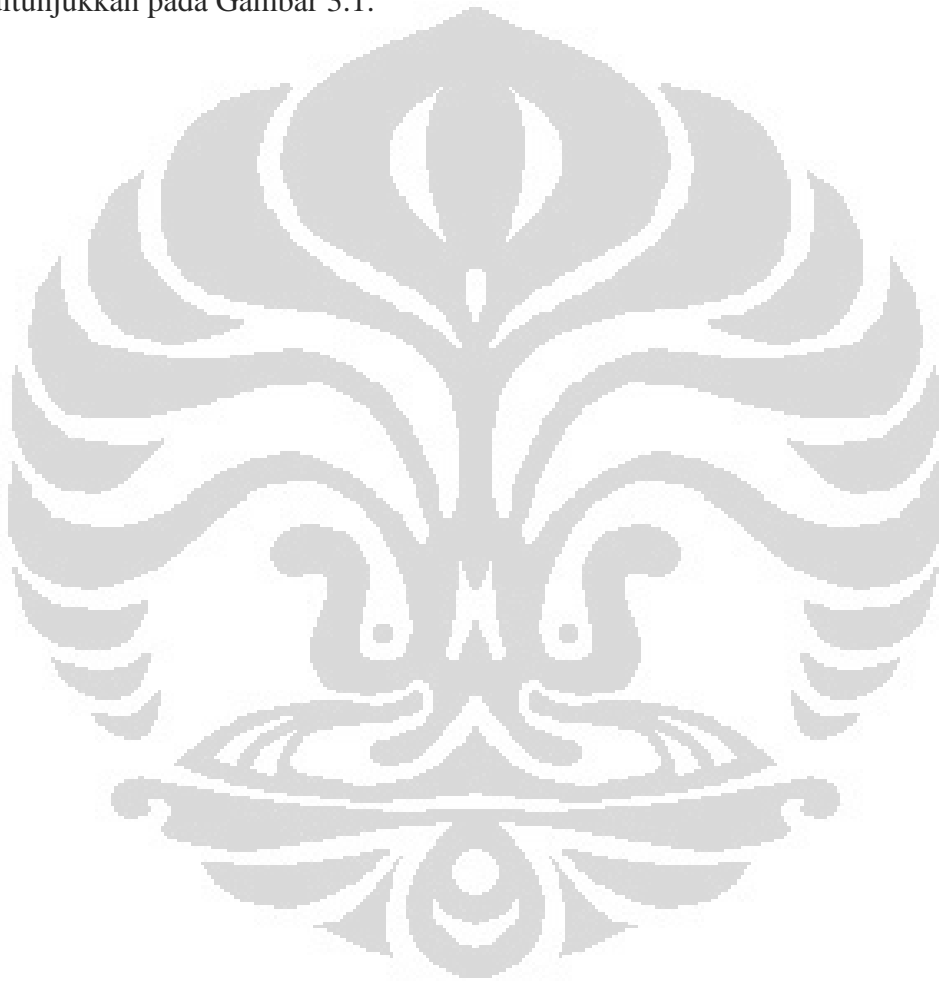
Saat memasuki *random backoff* maka akan terjadi seperti halnya sebuah hitung mundur. Hal ini bertujuan agar antar *reader* tidak terjadi tumbukan. Proses penghitungan mundur ini akan dijelaskan di sub bab 3.4. Untuk penentuan bilangan awal dari proses hitung mundur ini adalah dengan menggunakan mekanisme eksponensial *random backoff* yang nanti akan dijelaskan pada sub bab 3.3. Pada saat proses hitung mundur akan berakhir (mencapai angka nol) maka *reader* akan bersiap mengirimkan sinyal BROADCAST untuk kemudian membaca tag. Akan tetapi, jika belum berakhir dan tiba-tiba menerima sinyal BEACON (BROADCAST/BUSY/END) maka proses hitung mundur akan tertunda beberapa *timeslot* setelah itu *reader* meneruskan proses hitung mundur dan akhirnya bersiap mengirimkan sinyal BROADCAST untuk membaca tag.

Ketika tidak ada sinyal BROADCAST atau BUSY yang diterima dari *reader* lain, maka *reader* akan mengirimkan sinyal BROADCAST melalui kanal kontrol sehingga *reader* yang lain akan menerimanya.

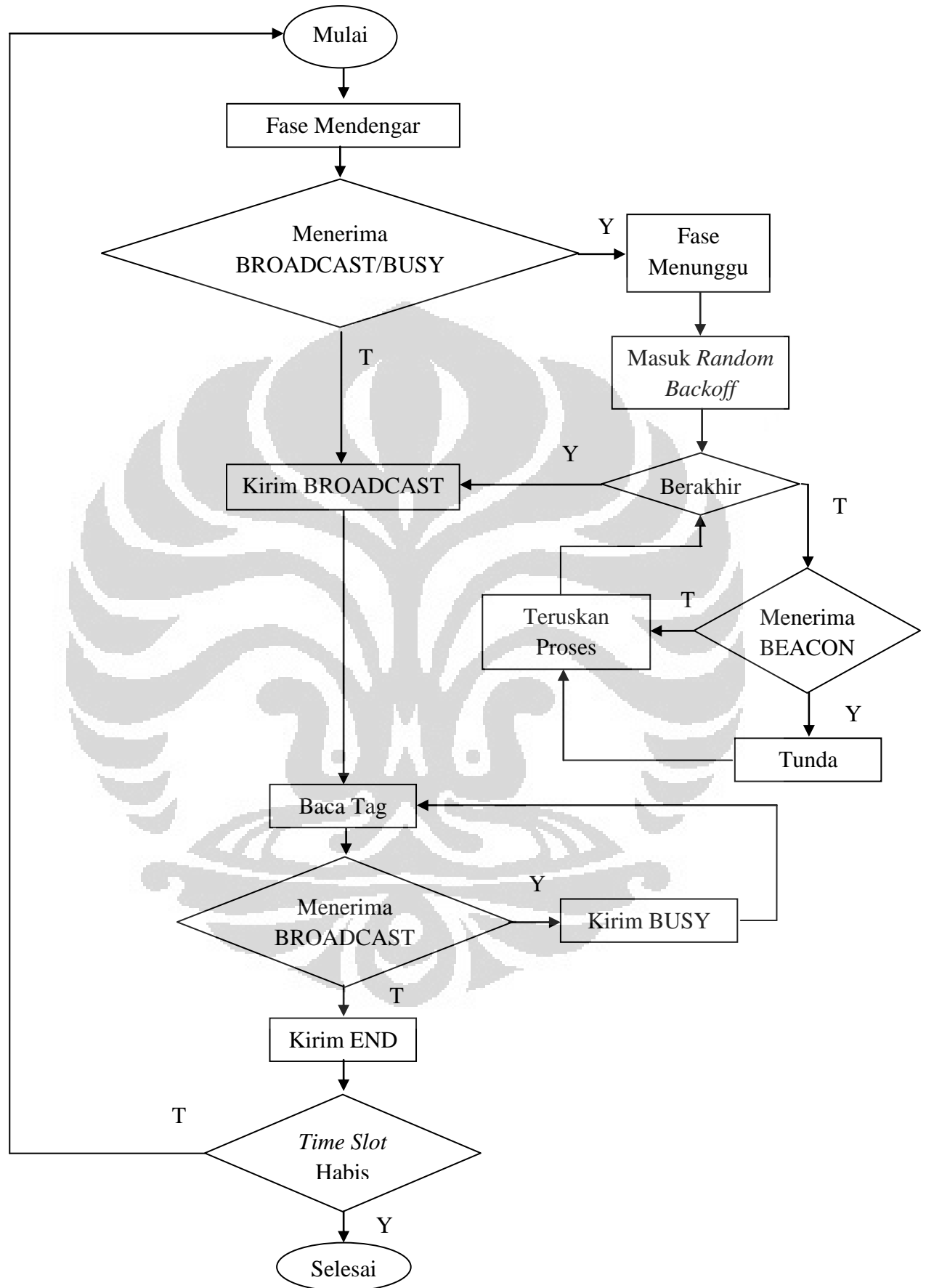
Saat *reader* tidak menerima sinyal BUSY dari *reader* lain maka itu artinya kanal data berada dalam kondisi *idle* sehingga *reader* dapat mengakses tag. Pada saat *reader* sedang mengakses tag, apabila tiba-tiba menerima sinyal BROADCAST dari *reader* lain maka *reader* harus mengirimkan sinyal BUSY

kepada *reader* tersebut sebagai tanda bahwa proses komunikasi dengan tag sedang berlangsung.

Ketika proses komunikasi sudah selesai maka *reader* akan mengirimkan sinyal END melalui kanal kontrol sebagai tanda *reader* lain bisa mengakses tag. Jika *time slot* yang dipakai belum habis maka *reader* akan mengulang kembali usaha untuk berkomunikasi dengan tag namun, jika *time slot* yang dipergunakan telah habis maka proses simulasi telah selesai. *Flowchart* dari algoritma simulasi ditunjukkan pada Gambar 3.1.







Gambar 3.1. Flowchart Algoritma Simulasi

### 3.3 Eksponensial *Random Backoff*

Mekanisme *random backoff* dibutuhkan untuk mencegah kemungkinan terjadinya tumbukan akibat pengaksesan kanal kontrol yang dilakukan oleh banyak *reader* dalam waktu yang sama. *Random backoff* itu sendiri ternyata memiliki banyak jenis. Dalam skripsi ini penulis memilih eksponensial *random backoff*.

#### 3.3.1 *Truncated Exponential Backoff*

Di dalam sistem jaringan komputer, *truncated binary exponential backoff* atau *binary exponential backoff* merujuk kepada algoritma yang digunakan untuk meregangkan atau menguraikan pentransmisi yang berulang-ulang dari beberapa blok data, sehingga sering menjadi bagian jaringan untuk mencegah kepadatan pengiriman data [18].

Contoh dari pentransmisi *frame* yang berulang-ulang adalah yang terdapat pada jaringan *Carrier Sense Multiple Access with Collision Avoidance* (CSMA/CA) dan *Carrier Sense Multiple Sense with Collision Detection* (CSMA/CD). Algoritma pada jaringan ini adalah bagian dari metode akses kanal yang digunakan untuk mengirim data. Di dalam jaringan Ethernet, algoritma ini umumnya digunakan untuk pentransmisi ulang setelah terjadi tumbukan.

Setelah terjadi *c collision* (tumbukan), sejumlah bilangan acak dari *timeslot* antara 0 dan  $2^c - 1$  dipilih. Untuk tumbukan pertama, masing-masing pengirim akan menunggu 0 atau 1 *timeslot*. Kemudian tumbukan kedua, pengirim akan menunggu dari 0 sampai 3 *timeslot*. Lalu tumbukan ketiga, pengirim akan menunggu dari 0 sampai 7 *timeslot* dan seterusnya. Karena jumlah pentransmisi ulang meningkat maka jumlah probabilitas *delay* juga meningkat secara eksponensial [18].

Kata ‘truncated’ secara sederhana merupakan bilangan pembatas yang apabila bilangan tersebut telah tercapai maka penambahan bilangan secara eksponensial akan berhenti.

#### 3.3.2 *Expected Backoff*

*Expected backoff* adalah rata-rata probabilitas dari distribusi seragam [18]. *Expected backoff* dirumuskan sebagai

$$\frac{1}{N+1} \sum_{i=0}^N i \quad (3.1)$$

$$\text{dimana } N = 2^c - 1 \quad (3.2)$$

Sebagai contoh, expected backoff time untuk  $c = 3$ , maka

$$N = 2^c - 1$$

$$N = 2^3 - 1 = 8 - 1$$

$$N = 7$$

Sehingga

$$E(c) = \frac{1}{N+1} \sum_{i=0}^N i$$

$$E(3) = \frac{1}{7+1} \sum_{i=0}^7 i = \frac{1}{8} (0+1+2+3+4+5+6+7) = \frac{28}{8}$$

$$E(3) = 3.5$$

### 3.4 Backoff Decrement Interval (BDI)

*Backoff Decrement Interval* (BDI) merupakan sebuah interval tertentu setelah nilai *backoff* mengalami penurunan [15]. Gambar 3.2 menunjukkan *time line* dari empat buah *reader* dalam sebuah sistem sedangkan Gambar 3.3 menunjukkan transmisi dari *reader* lain (R2, R3, dan R4) yang diletakkan pada *time line reader* R1. Gambar 3.3 juga menunjukkan dengan jelas yang dimaksud dengan BDI.

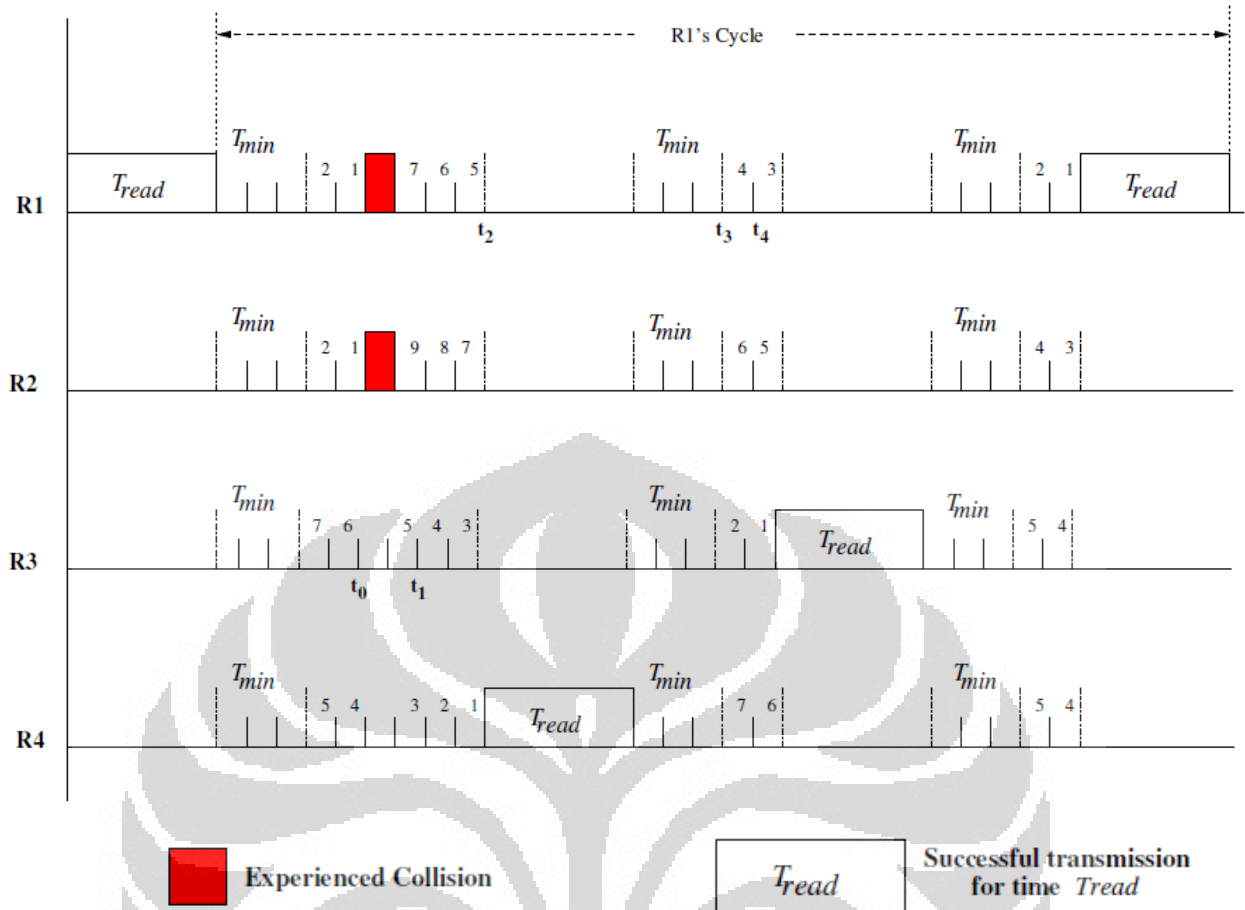
Pada Gambar 3.2 terlihat bahwa tumbukan terjadi ketika *backoff counter* dari R1 dan R2 sama-sama menghasilkan angka 0 (nol). Akibat tumbukan tersebut maka *backoff counter* dari R3 dan R4 terpaksa berhenti sejenak. Setelah tumbukan berakhir maka R3 dan R4 melanjutkan *backoff counter*-nya. Ketika terjadi komunikasi antara *reader* dengan tag maka kondisinya pun serupa dengan diatas, misalnya yang terjadi pada R4. Saat R4 berkomunikasi dengan tag ( $T_{read}$ ) maka *reader* lain menghentikan sejenak *backoff counter* mereka. Setelah  $T_{read}$  berakhir maka *reader* lain dan reader R4 memasuki tahap  $T_{min}$ .  $T_{min}$  ini bisa dikatakan sebuah waktu tunggu/jeda yang dilakukan oleh *reader* setelah adanya sebuah

*reader* yang berkomunikasi dengan tag. Namun, tidak hanya *reader* yang berkomunikasi dengan tag saja yang harus melewati  $T_{min}$  ini tapi semua *reader*.

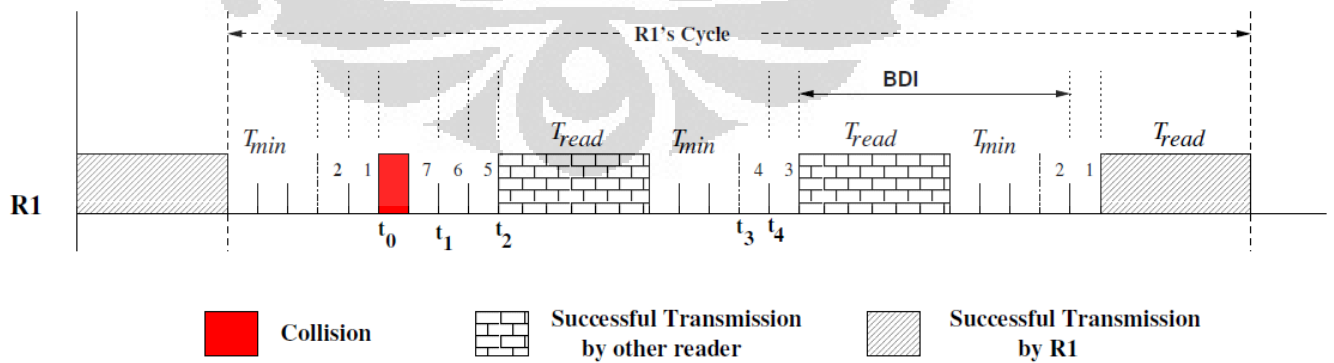
Setelah melewati  $T_{min}$  maka *reader* lain selain R4 melanjutkan *backoff counter*-nya sedangkan R4 melakukan proses *random backoff* untuk menentukan nilai awalan *backoff counter*-nya untuk kemudian nanti bisa berkomunikasi kembali dengan tag.

Pada Gambar 3.2 dan 3.3 satuan yang digunakan tiap kotak-kotak pada gambar tersebut adalah dalam *timeslot*. Sehingga dapat disimpulkan  $T_{read} = 4$  *timeslot* dan  $T_{min} = 3$  *timeslot*.

Di dalam BDI terdapat durasi BDI. Durasi BDI sesaat setelah berkomunikasi dengan tag dan melewati  $T_{min}$  ( $T_s$ ) atau dengan kata lain transmisi yang sukses adalah sama dengan  $T_{read} + T_{min} + 1$ . Sedangkan durasi BDI ketika terjadi tumbukan ( $T_{col}$ ) sama dengan 1 (tumbukan) + 1 (*timeslot* kosong). Sedangkan jika bukan berupa transmisi yang sukses atau tumbukan maka durasi BDI merupakan *timeslot* tunggal [15].



Gambar 3.2 Efek Transmisi BDI dari Berbagai Macam Reader [17]



Gambar 3.3 BDI dari Satu Buah Reader (R1) [17]

### 3.5 Variabel Simulasi

Dibawah ini merupakan variabel yang dipergunakan dalam simulasi ini [15]

BDI : *Backoff Decrement Interval*

$P_s$  : Probabilitas BDI yang berisi transmisi yang sukses

$ET_{BDI}$  : Durasi rata-rata BDI

$E_{BDI}$  : Jumlah rata-rata BDI antara dua buah transmisi yang sukses dari sebuah *reader*

$ET_{cycle}$  : Durasi rata-rata antara dua buah transmisi yang sukses dari sebuah *reader*

$T_{read}$  : Durasi maksimum yang dibolehkan untuk sebuah *reader* berkomunikasi dengan tag

$t_{query}$  : *Delay* propagasi pada kanal data

$t_{beacon}$  : *Delay* propagasi pada kanal kontrol

$l_{query}$  : Waktu transmisi dari sebuah *query*

$l_{beacon}$  : Waktu transmisi dari sebuah *beacon*

$Q_{T_{read}}$  : Jumlah *query* yang dikirimkan oleh *reader* dalam  $T_{read}$

$S$  : Sistem *throughput*, yaitu jumlah *query* yang dikirimkan oleh semua *reader* per satuan waktu

Berdasarkan [15] diperoleh persamaan berikut ini:

$$ET_{cycle} = E_{BDI} \times ET_{BDI} + T_{read} \quad (3.3)$$

Asumsikan masing-masing *reader* dibolehkan untuk berkomunikasi dengan tag untuk maksimum  $x$  *beacon* interval mislanya  $T_{read} = x$  *time slot*. Masing-masing *time slot* di dalam  $T_{read}$  dari sebuah *reader* akan terdiri dari satu transmisi *beacon* pada kanal kontrol dan beberapa transmisi (*informasi/query*) pada kanal data. Sehingga menurut [15] diperoleh persamaan:

$$T_{read} = [(t_{query} + l_{query}) \times Q_{T_{read}}] + x(t_{beacon} + l_{beacon}) \quad (3.4)$$

$$Q_{T_{read}} = \left[ \frac{T_{read} - x(t_{beacon} + l_{beacon})}{t_{query} + l_{query}} \right] \quad (3.5)$$

Jumlah rata-rata periode waktu yang sukses ( $T_{read}$ ) dari semua *reader* dalam satu putaran (*cycle*) adalah  $P_s \times E BDI$  dimana  $P_s$  merupakan probabilitas BDI yang berisi transmisi yang sukses. Oleh karenanya, jumlah rata-rata *query* (informasi) yang dikirimkan oleh semua *reader* dalam satu detik atau dengan kata lain merupakan sistem *throughput*. Oleh karena itu, dari [17] persamaannya adalah

$$S = \frac{Q_{T_{read}} \times P_s \times E BDI}{ET_{cycle}} \quad (3.5)$$

Contoh perhitungan:

Berdasarkan Gambar 3.3 dapat diperoleh hasil

$$EBDI = \frac{8}{1} = 8$$

$$ETBDI = \frac{8+1+2+1+1+8+1+8+1}{10} = \frac{31}{10} = 3.1$$

$$ET_{cycle} = \frac{1+2+1+1+4+3+1+1+4+3+1+1}{1} = \frac{23}{1} = 23$$

### 3.6 Kondisi yang Diamati

Kondisi – kondisi yang diamati dalam skripsi ini adalah sebagai berikut:

a) *Throughput*

*Throughput* adalah banyaknya informasi yang berhasil dikirim dan diterima oleh *reader*. atau dengan kata lain *throughput* adalah jumlah rata-rata transmisi paket yang sukses per waktu interval tertentu [16].

b) Efisiensi

Efisiensi merefleksikan kemampuan protokol untuk mendeteksi kemungkinan terjadinya tumbukan pada tag dan oleh karenanya mencegah transmisi yang tidak diinginkan [15].

c) Waktu Tunggu

Merupakan waktu rata-rata dari sebuah *reader* sebelum dapat mengakses kanal yang nilainya sama dengan  $ET_{cycle}$  [17].

### 3.7 Rancangan Simulasi

Proses simulasi yang dilakukan pada skripsi ini adalah dengan menggunakan *tools* MATLAB R2009a yang bekerja pada Windows XP. Dengan MATLAB R2009a ini akan dihitung berapa besar paket yang berhasil di akses oleh *reader* dari tag dan berapa banyak tumbukan yang terjadi antar *reader*. Sehingga nanti akan diketahui berapa besar *throughput* dan effisiensinya.

Pada simulasi ini digunakan 1-20 buah *reader* yang sebarannya secara acak dan 1 buah tag.

Simulasi yang dijalankan pada MATLAB ini mengadopsi sistem matriks. Pertama-tama dilakukan penentuan nilai tiap parameter kemudian melakukan penginisialisasian BDI matriks (`bdiMatrix`). BDI matriks merupakan suatu matriks yang berisikan nilai-nilai *backoff* dari setiap *reader*. Setiap *reader* direpresentasikan dalam sebuah baris pada BDI matriks. BDI matriks ini merupakan algoritma yang sangat penting dalam simulasi ini.

Setelah BDI matriks terbentuk dan terlihat pada *time slot* mana saja terjadi tumbukan, transmisi yang sukses, dan lain sebagainya. Maka langkah selanjutnya adalah menentukan nilai durasi BDI (`bdiLengthSequence`). Nilai durasi BDI ini telah dijelaskan pada sub bab 3.4. Selain untuk mengetahui nilai durasi BDI, perlu diketahui juga *reader* mana saja yang melakukan proses komunikasi dengan tag (`bdiSequenceRead`).

Kemudian setelah diperoleh durasi BDI (`bdiLengthSequence`) maka akan dihitung besar nilai E BDI. Penghitungan E BDI bergantung hasil dari `bdiSequenceRead`. Lalu juga dihitung  $ET_{BDI}$  yang memanfaatkan hasil dari `bdiLengthSequence`.

Setelah E BDI dan  $ET_{BDI}$  diperoleh maka akan didapatkan  $ET_{cycle}$ ,  $QT_{read}$ , dan  $P_s$ . kemudian *throughput* juga akan didapatkan tetapi hasilnya akan dikalikan terlebih dahulu dengan 1000 dan besar *query* dalam satuan byte (`querysize`). Dan akhirnya diperoleh pula besar nilai efisiensi berdasarkan persamaan 2.2.



## BAB 4

### HASIL SIMULASI DAN ANALISA DATA

#### 4.1 Parameter Simulasi

Di dalam simulasi ini terdapat beberapa parameter yang dipergunakan diantaranya:

Tabel 4.1 Parameter Simulasi Sistem

Sistem Parameter	Keterangan	Nilai	Satuan
$T_{min}$	Waktu tunggu sesaat setelah <i>reader</i> berkomunikasi dengan tag	3	<i>time slot</i>
$T_{read}$	Durasi maksimum yang dibolehkan untuk sebuah <i>reader</i> berkomunikasi dengan tag	4	<i>time slot</i>
$T_{col}$	Durasi BDI yang berisi tumbukan	1	<i>time slot</i>
$t_{query}$	<i>Delay</i> propagasi pada kanal data	0.1	ms
$t_{beacon}$	<i>Delay</i> propagasi pada kanal kontrol	0.1	ms
$l_{query}$	Waktu transmisi dari sebuah <i>query</i>	0.1	ms
$l_{beacon}$	Waktu transmisi dari sebuah <i>beacon</i>	0.1	ms
<i>Query size</i>	Ukuran <i>query</i>	10	byte
<i>Truncated</i>	Nilai batas eksponensial <i>random bakoff</i>	10	-
Jumlah <i>reader</i>		1 - 20	-
Jumlah <i>time slot</i>		1 - 3000	-
Jumlah tag		1	-

Pada simulasi ini dipergunakan *time slot* agar lebih mempermudah dalam menjalankan simulasi khususnya dalam hal pewaktuan. Penulis menggunakan 1 *time slot* adalah sama dengan 20 ms. Sedangkan total *time slot* yang digunakan

adalah 3000 *time slot*. Oleh karena itu, total waktu keseluruhan adalah  $20 \cdot 10^{-3} \times 3000 = 60$  detik (1 menit). Nilai variasi *timeslot* ini juga digunakan untuk menganalisa pengaruh banyaknya *timeslot* terhadap *throughput* dan efisiensi yang dihasilkan.

$T_{\min}$ ,  $T_{read}$ ,  $T_{col}$  akan sangat berpengaruh pada saat penghitungan durasi BDI, E BDi dan E  $T_{BDI}$ . Besarnya nilai ketiga parameter tersebut adalah merujuk pada [15].

Parameter  $t_{query}$ ,  $t_{beacon}$ ,  $l_{query}$ ,  $l_{beacon}$  berperan dalam penghitungan jumlah *query* yang dikirimkan oleh *reader* dalam  $T_{read}$  ( $QT_{read}$ ). Nilai-nilainya mengikuti jurnal [19].

Jumlah *reader* yang dipergunakan adalah 1 hingga 20. Nilai variasi jumlah *reader* ini juga digunakan untuk menganalisa pengaruh banyaknya *reader* terhadap *throughput* dan efisiensi yang dihasilkan seperti halnya pada *time slot*. Sebab pada skripsi ini yang akan dilihat adalah tumbukan antar *reader*. Hal ini tentu saja berbeda dengan [9] yang memvariasikan jumlah tag.

Jumlah tag yang digunakan adalah 1 buah sebab seperti yang telah diutarakan diatas, skripsi ini bertujuan untuk mengurangi tumbukan antar *reader*.

## 4.2 Pemodelan Simulasi

Di dalam skripsi ini kita akan mencoba untuk memodelkan seperti apa sistem yang akan dibuat. Perlu diketahui dalam pemodelan ini, *reader* adalah dalam keadaan statis (tidak bergerak). Oleh karenanya, kita akan membuat pemodelan dengan berbagai asumsi. Tujuan asumsi ini adalah agar memudahkan dalam menganalisis sistem. Asumsi-asumsi tersebut adalah berikut [17]:

1. Saturasi. Maksudnya adalah *reader* akan selalu berada di dekat tag sehingga *reader* cenderung akan selalu berkomunikasi dengan tag.
2. Jika salah satu *reader* mengirimkan *beacon*, maka semua *reader* akan menerima *beacon* tersebut.
3. Karena semua *reader* menerima *beacon* yang dikirimkan oleh salah satu *reader* maka untuk berkomunikasi dengan tag tidak bisa untuk lebih dari *reader* (tunggal) atau dengan kata lain tag hanya bisa diakses oleh satu buah *reader*.

4. Penggunaan *time slot*. Seperti yang telah dijelaskan diatas penggunaan *time slot* adalah untuk mempermudah proses simulasi meskipun pada realisasinya di dunia nyata waktu tidaklah mungkin di-slot-kan. Kemudian dilakukan juga sinkronisasi dalam pemodelan ini.

### 4.3 Keterbatasan Penelitian

Simulasi dan penelitian yang telah dilakukan ini memiliki keterbatasan antara lain:

1. *Reader* yang digunakan adalah *reader* yang bersifat statis (tidak bergerak). Akibatnya kita tidak bisa mengetahui bagaimana hasilnya jika *reader* bersifat bergerak dan juga tidak bisa membandingkan pengaruhnya terhadap *throughput* dan efisiensi antara *reader* yang statis dan yang dinamis (*mobile*).
2. Tidak adanya pengaruh frekuensi karena jurnal-jurnal rujukan yang membahas tumbukan antar *reader* yang penulis dapatkan dan telusuri tidak pernah menyertakan landasan teori ataupun persamaan pengaruh frekuensi terhadap sistem. Sehingga penulis tidak bisa menyertakannya dalam simulasi.

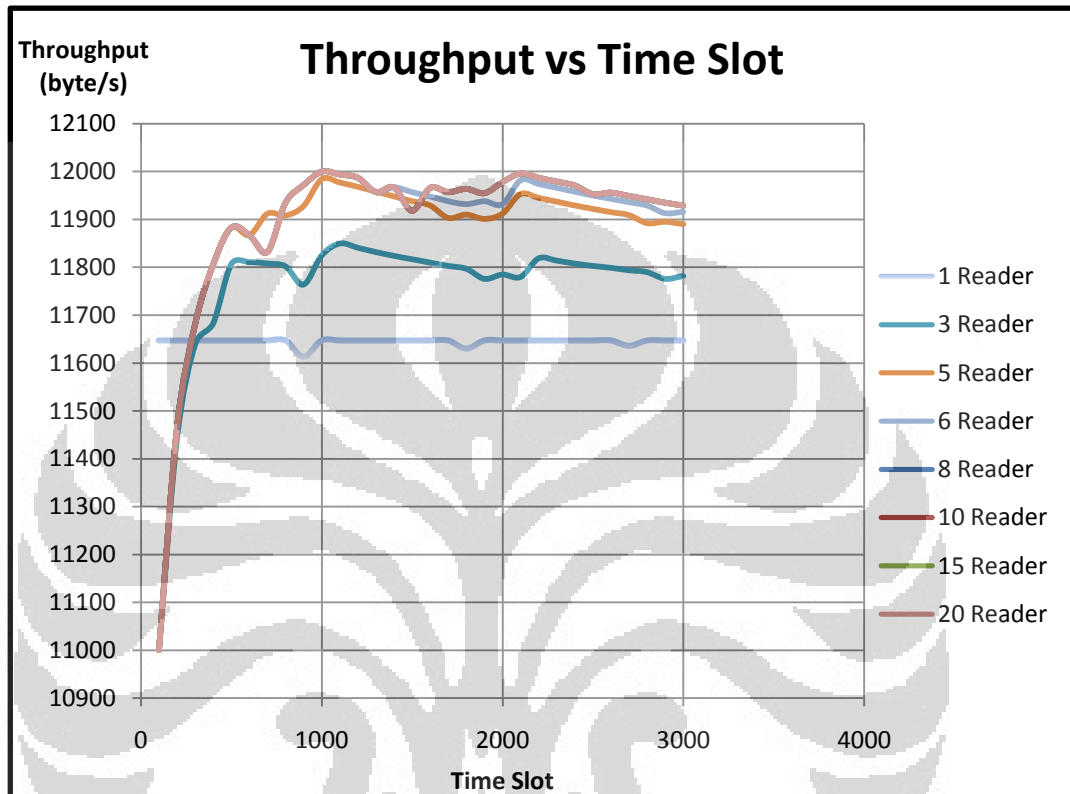
### 4.4 Hasil Pengolahan Data dan Analisa

#### 4.4.1 Hubungan Antara Jumlah *Reader* dengan *Throughput*

Gambar 4.1 dibawah ini adalah grafik hubungan antara *throughput* dengan *time slot* yang digunakan pada jumlah *reader* tertentu. Dapat dilihat bahwa pada saat 1 *reader*, *throughput* yang dihasilkan relatif konstan hanya terlihat terjadi beberapa penurunan saja di *time slot* tertentu. Hal ini disebabkan karena dengan 1 buah *reader* maka informasi yang terdapat pada tag tidak terambil secara maksimal. Sedangkan jika lebih dari 1 buah *reader* maka informasi yang bisa diperoleh dari tag bisa lebih besar.

Makin banyak jumlah *reader* maka makin besar pula nilai *throughput* yang dihasilkan khususnya dari 1 sampai 6 *reader*. Akan tetapi, saat jumlah *reader* lebih dari 6, dalam hal ini 8 hingga 20 *reader* maka dapat dilihat bahwa nilai *throughput* yang dihasilkan sama. Hal ini

menunjukkan bahwa 8 *reader* merupakan jumlah yang paling cukup untuk menghasilkan *throughput* terbaik. Selain itu, 8 hingga 20 *reader* menyebabkan sistem mengalami saturasi (jenuh).



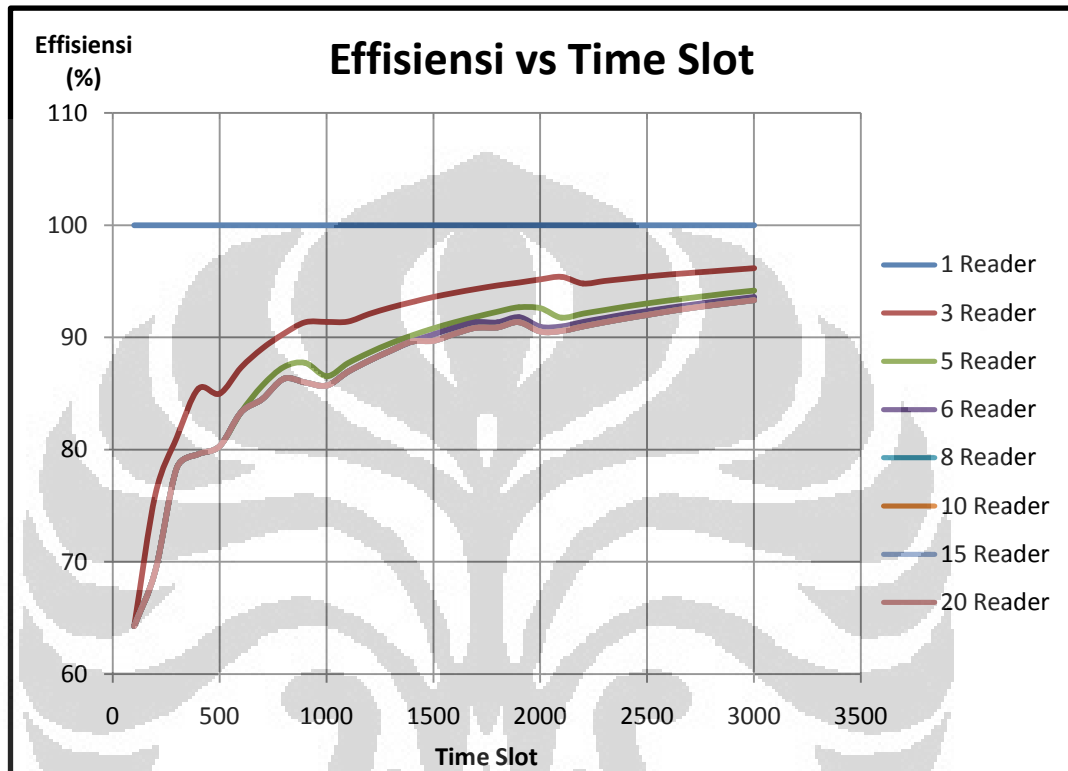
Gambar 4.1 Grafik Hubungan *Throughput* dengan *Time Slot* yang Digunakan oleh Beberapa *Reader*

#### 4.4.2 Hubungan Antara Jumlah *Reader* dengan Efisiensi

Pada Grafik 4.2 dibawah ini dapat dilihat bahwa pada saat jumlah *reader* sama dengan 1 maka efisiensi mencapai angka yang paling ideal. Hal ini sangat wajar karena tidak adanya persaingan dengan *reader* lain dalam berkomunikasi dengan tag sehingga menyebabkan tidak adanya tumbukan dengan *reader* lain.

Dari grafik dibawah ini juga dapat dilihat secara umum bahwa makin banyak jumlah *reader* yang digunakan maka makin kecil pula efisiensi yang dihasilkan sistem. Pada saat jumlah *reader* 8 hingga 20, grafik efisiensi yang dihasilkan adalah berimpit. Hal ini menunjukkan bahwa

mulai jumlah reader sama dengan 8 maka sistem telah mengalami saturasi (jenuh). Hal ini pun juga sama dengan Gambar 4.1 yang menunjukkan bahwa dengan 6 buah *reader* adalah batas maksimal sebelum sistem memasuki kondisi saturasi.



Gambar 4.2 Grafik Hubungan Effisiensi dengan *Time Slot* yang Digunakan oleh Beberapa *Reader*

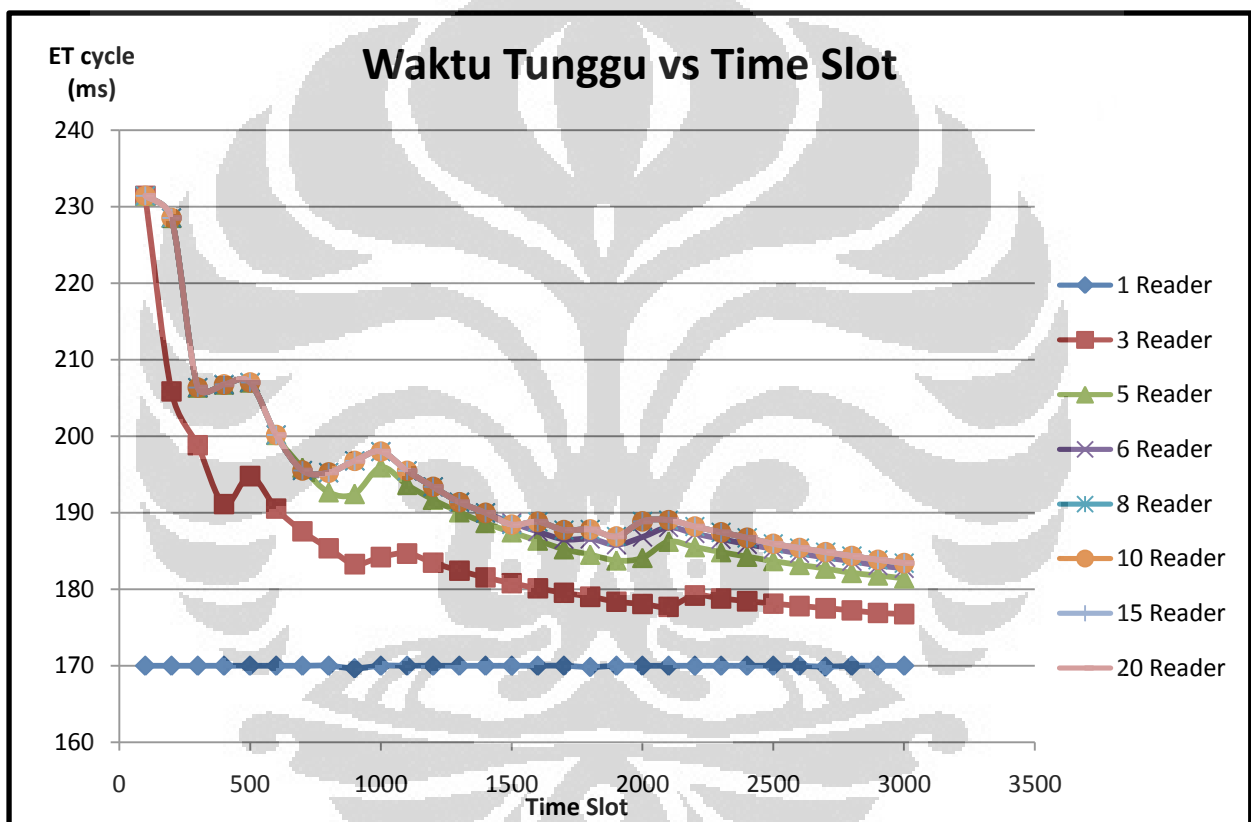
#### 4.4.3 Hubungan Antara Jumlah *Reader* dengan Waktu Tunggu

Pada Gambar 4.3 dibawah ini terlihat bahwa saat jumlah *reader* adalah sama dengan 1 buah maka waktu tunggu adalah relatif sama. Sedangkan grafik dengan jumlah *reader* lebih dari 1 maka hasil waktu tunggu cenderung mengalami penurunan seiring dengan bertambahnya *time slot*. Makin banyak jumlah *reader* maka makin besar pula waktu tunggu. Hal ini disebabkan karena probabilitas sebuah *reader* terselip oleh *reader* lain saat hendak mengakses atau berkomunikasi dengan tag adalah besar. Disamping itu, kita telah sama-sama ketahui bahwa dalam

penelitian ini sebuah *reader* tidak bisa mengakses bersama-sama dengan *reader* lain dalam waktu yang sama.

Pada grafik dibawah ini juga tampak bahwa ketika *time slot*-nya bertambah maka waktu tunggu mengalami penurunan. Hal ini tentu saja sesuai dengan analisis pada grafik sebelumnya.

Dari Gambar 4.3 dibawah ini kita juga bisa melihat bahwa saat jumlah *reader* lebih dari sama dengan 8 maka sistem telah mengalami saturasi.



Gambar 4.3 Grafik Hubungan Waktu Tunggu dengan *Time Slot* untuk Variasi Jumlah *Reader*

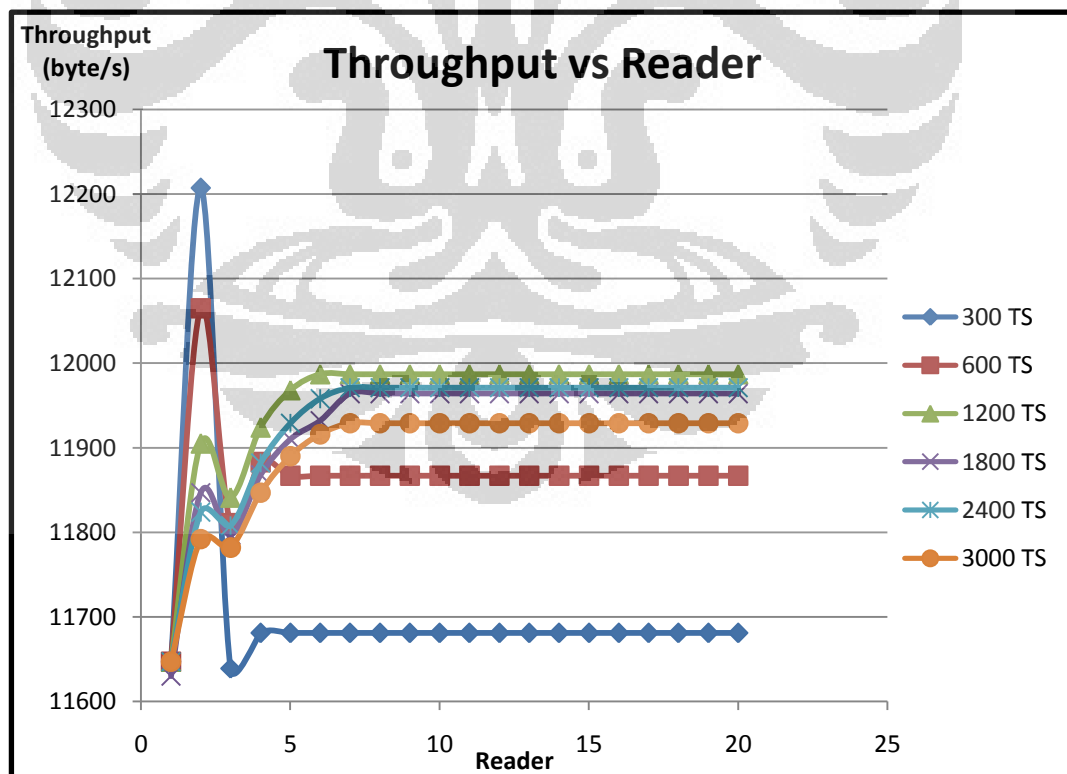
#### 4.4.4 Hubungan Antara Jumlah *Time Slot* dengan *Throughput*

Pada Gambar 4.4 dibawah ini terlihat bahwa pada saat awal-awal grafik yakni, saat jumlah *reader* sama dengan 2 nilai *throughput* tertinggi yang dapat dicapai adalah pada saat *time slot*-nya berjumlah 300 buah. Namun, setelah itu semua variasi *time slot* mengalami penurunan dan

akhirnya tercapai kondisi saturasi. Saat kondisi saturasi justru terjadi sebaliknya, saat 300 *time slot*-lah nilai *throughput* saturasi mendapatkan hasil yang terendah.

Dari grafik dibawah ini dapat disimpulkan bahwa jika ingin menggunakan *reader* dalam jumlah sedikit (kurang dari tiga) maka akan lebih baik untuk menggunakan 300 *time slot* akan tetapi, jika ingin menggunakan *reader* dalam jumlah yang banyak (lebih dari tiga) maka 1200 *time slot* menjadi pilihan terbaik.

Dari Gambar 4.4 juga terlihat bahwa sistem mengalami saturasi untuk berbagai macam variasi *time slot*, yang paling cepat mengalami saturasi adalah saat 300 *time slot* saat *reader* berjumlah 4 buah, kemudian disusul 600 saat *reader* berjumlah 5 buah, dan 1200 *time slot* saat *reader* berjumlah 6 buah. Sedangkan untuk 1800 hingga 3000 *time slot*, saturasi terjadi pada saat yang sama yaitu saat jumlah *reader* adalah sama dengan 7 buah.

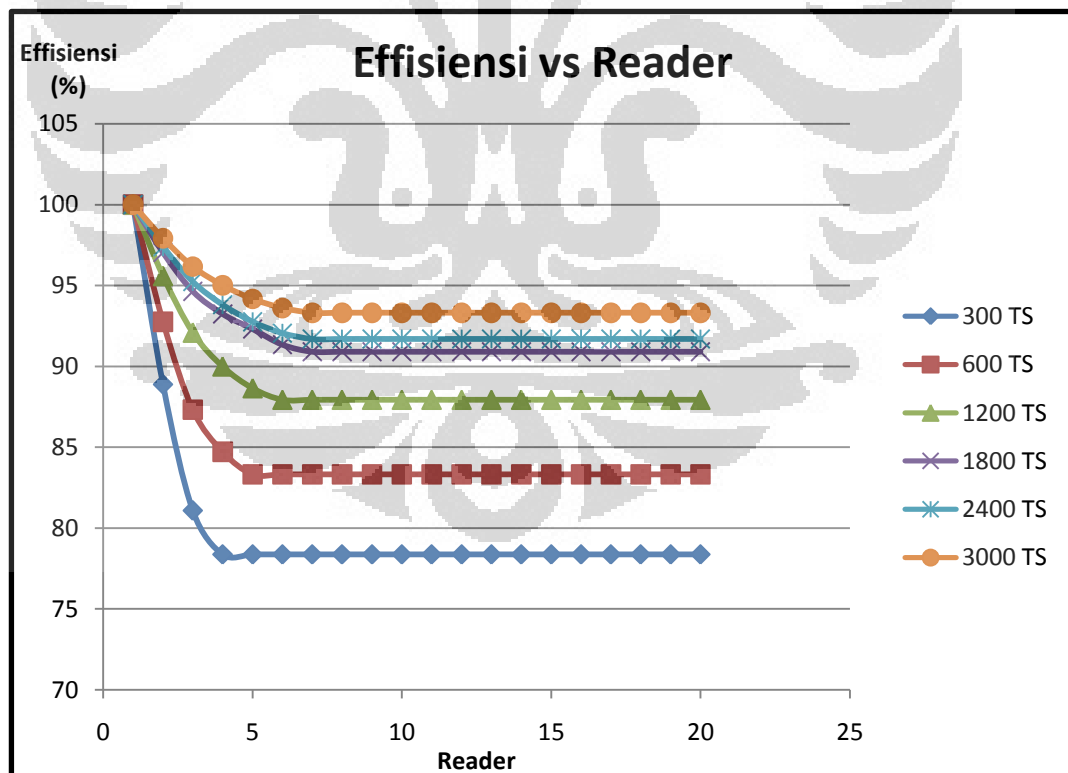


Gambar 4.4 Grafik Hubungan *Throughput* dengan *Reader* pada Variasi *Time Slot* Tertentu

#### 4.4.5 Hubungan Antara Jumlah *Time Slot* dengan Efisiensi

Pada Gambar 4.5 dibawah ini dapat dilihat bahwa saat semua variasi *time slot* menunjukkan hasil yang sama saat jumlah *reader* sebanyak 1 buah akan tetapi, setelah itu terjadi penurunan yang berbeda-beda diantara variasi *time slot* yang ada. Penurunan yang paling tajam adalah saat 300 *time slot*. Sebaliknya, 3000 *time slot* merupakan penurunan yang paling landai dibandingkan variasi *time slot* yang lain.

Semua variasi *time slot* mengalami saturasi yang berbeda-beda. Untuk 300 *time slot* terjadi saat 4 *reader*, 600 *time slot* terjadi saat 5 *reader*, 1200 *time slot* terjadi saat 6 *reader*, 1800, 2400, dan 3000 *time slot* terjadi saat 7 *reader*. Hal ini menunjukkan bahwa makin banyaknya *time slot* maka akan sangat membantu dalam membuat hasil yang lebih presisi. Sehingga dapat diperoleh efisiensi yang tinggi.



Gambar 4.5 Grafik Hubungan Efisiensi dengan *Reader* untuk Variasi *Time Slot*

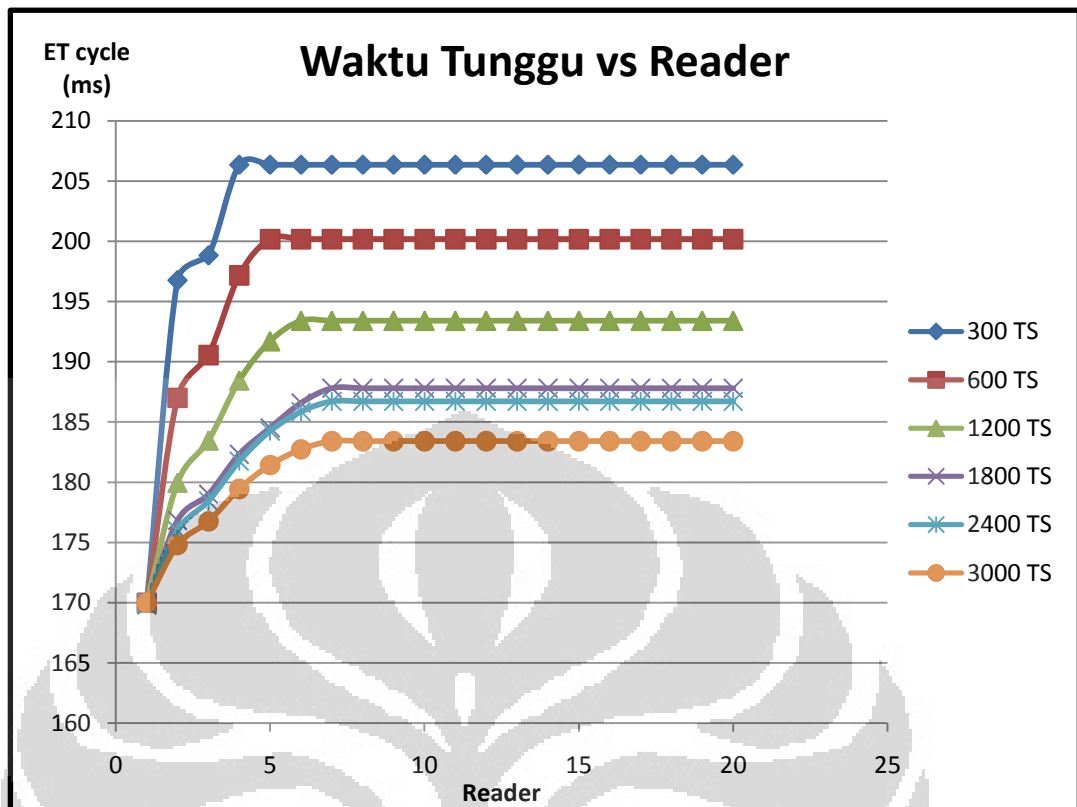


#### 4.4.6. Hubungan Antara Jumlah *Time Slot* dengan Waktu Tunggu

Seperti yang telah dijelaskan sebelumnya pada bab 3 bahwa yang dimaksud dengan waktu tunggu adalah waktu dari sebuah *reader* sebelum dapat mengakses kanal atau dengan kata lain adalah waktu yang dibutuhkan oleh sebuah *reader* untuk dapat berkomunikasi atau mengakses tag kembali.

Dari Gambar 4.6 dibawah ini dapat dilihat secara umum bahwa makin besar jumlah *time slot* maka makin kecil pula harga waktu tunggu ( $ET_{cycle}$ ) dari sebuah *reader* sesaat setelah berkomunikasi dengan tag hingga dapat berkomunikasi kembali dengan tag. Hal ini disebabkan karena dengan makin besarnya *time slot* maka kesempatan dari sebuah *reader* untuk dapat berkomunikasi kembali dengan tag setelah sebelumnya berkomunikasi dengan tag adalah makin besar.

Dapat dilihat pula bahwa sistem mengalami saturasi (jenuh). Makin kecil *time slot* maka makin sedikit pula jumlah *reader* yang dapat dipergunakan untuk mencapai kondisi saturasi. Artinya dengan jumlah *reader* yang kecil dan diimbangi dengan *time slot* yang kecil pula maka sistem makin cepat pula mengalami kondisi saturasi. Sedangkan sebaliknya, jika dengan jumlah *time slot* yang besar maka kondisi saturasi baru tercapai dengan jumlah *reader* yang relatif lebih banyak. Perhatikan grafik dibawah ini pada 300 dan 600 *time slot* kondisi saturasi dicapai saat jumlah *reader* adalah lebih kecil dari sama dengan 5 ( $reader \leq 5$ ) sedangkan pada saat 1200 – 3000 *time slot* kondisi saturasi baru tercapai ketika jumlah *reader* adalah lebih besar dari sama dengan 5 ( $reader \geq 5$ ).



Gambar 4.6 Grafik Hubungan Waktu Tunggu dengan *Reader* untuk Variasi *Time Slot*

## BAB 5

### KESIMPULAN

Berdasarkan hasil simulasi, pengolahan data, dan analisa data maka dapat ditarik beberapa kesimpulan yang dapat menjawab rumusan masalah yang ada, yaitu:

1. Harga *throughput* yang tinggi dapat diperoleh jika *time slot* yang digunakan banyak (lebih dari sama dengan 1000 *time slot*) dan *reader* yang dipergunakan cukup 8 buah *reader*.
2. Harga efisiensi yang tinggi akan diperoleh jika *reader* yang digunakan makin sedikit sebab dengan semakin sedikitnya *reader* maka semakin kecil pula persaingan antar *reader* sehingga tumbukan antar *reader* yang terjadi semakin kecil akibatnya efisiensi makin besar. Dalam penelitian ini jumlah *reader* yang terbaik yang dapat digunakan adalah kurang dari 5 *reader* sedangkan *time slot* terbaik yang dapat digunakan adalah lebih dari 3000 *time slot*.
3. Harga waktu tunggu yang kecil dari sebuah *reader* untuk dapat berkomunikasi dengan tag kembali setelah sebelumnya berhasil berkomunikasi dengan tag adalah dengan memperbesar *time slot* dan memperkecil jumlah *reader*. Dalam penelitian ini hasil terbaik adalah menggunakan 3000 *time slot* dan *reader* yang dipergunakan adalah sebanyak kurang dari 5 buah.

## DAFTAR REFERENSI

- [1] Wedang Ronde. Perpustakaan Masa Depan dengan Teknologi RFID. (26 Oktober 2010). <http://ad71ck.staff.uns.ac.id/archives/132>
- [2] Inovasi Online, RFID Sebuah Teknologi Identifikasi Pengancam Privasi? (26 Oktober 2010).  
[http://io.ppijepang.org/v2/index.php?option=com\\_k2&view=item&id=31:rfid-sebuah-teknologi-identifikasi-pengancam-privasi?](http://io.ppijepang.org/v2/index.php?option=com_k2&view=item&id=31:rfid-sebuah-teknologi-identifikasi-pengancam-privasi?)
- [3] Hunt, V Daniel., Puglia, Albert., & Puglia, Mike. (2007). *RFID A Guide to Radio Frequency Identification*. New Jersey: John Wiley & Sons.
- [4] RFID, Hendri Saputra. (26 Oktober 2010).  
<http://h210189.blog.binusian.org/2009/06/19/rfid/>
- [5] Wahyudi, Ricky Eko.(2010). Sistem Alarm Berbasis RFID Untuk Sistem Keamanan Rumah. *Skripsi Program Studi Teknik Elektro Fakultas Teknik Universitas Indonesia*.
- [6] Roussos, George. (2008). *Networked RFID Systems, Software and Services*. London: Springer.
- [7] Maryono. (2005). *Dasar-Dasar Radio Frequency Identification (RFID)*, Teknologi yang Berpengaruh di Perpustakaan. Media Informasi Vol.XIV No.20. Journal.
- [8] Finkenzeller, Klaus. (2003). *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identifications* (2nd ed.). West Sussex: John Willey & Sons.

- [9] Reza. A, Fauzi Dwi. (2010). Simulasi dan Analisis Delay Pada Sistem RFID Menggunakan Slotted Aloha. *Skripsi Program Studi Teknik Elektro Fakultas Teknik Universitas Indonesia*.
- [10] W.E, Daniel & E.S, Sanjay. (2002). *The Reader Collision Problem*. IEEE SMC.
- [11] Kin Seong Leong., Mun Leng Ng., H.C, Peter. (2005). *The Reader Collision Problem in RFID Systems*. IEEE International Symposium on Microwave, Antenna, Propagation, and EMC Technologies for Wireless Communications Proceedings. 0-7803-9128-4/05, 658-661.
- [12] Hongyue Dai., Shengl Lai., & Hailong Zhu. (2007). *A Multi-Channel MAC Protocol for RFID Reader Networks*. IEEE. 1-4244-1312-5/07, 2093-2096.
- [13] Joshi, Gyanendra & Won Kim, Sung. (2009). *An Efficient MAC Protocol for Throughput Enhancement in Dense RFID System*. IEEE. 978-1-4244-2966-0/09.
- [14] Gyanendra Prasad Joshi., Kazi Abdulla Mamun., & Sun Wong Kim. (2009). *A Reader Anti-collision MAC Protocol for Dense Reader RFID System*. International Conference on Communications and Mobile Computing. 978-0-7695-3501-2/09, 313-316.
- [15] Birari, Shailesh & Iyer, Sridar.(2005). *Mitigating the Reader Collision Problem in RFID Networks with Mobile Readers*. IEEE. 1-4244-0000-7/05, 463-468.
- [16] Sari, Lydia. (2002). Analisa Throughput Multicarrier CDMA ALOHA. *Tesis Program Studi Magister Telekomunikasi Fakultas Teknik Universitas Indonesia*.

- [17] M. Birari, Shailesh. (2005). *Mitigating the Reader Collision Problem in RFID Networks with Mobile Readers*. Desertasi Kanwal Rekhi School of Information Technology Indian Institute of Technology Bombay.
- [18] Exponential Backoff, (26 April 2011). Wikipedia, [http://en.wikipedia.org/wiki/Exponential\\_backoff](http://en.wikipedia.org/wiki/Exponential_backoff)
- [19] Song InChan., Hong SungHyun., & Chang KyungHi. (2009). *An Improved Reader Anti-Collision Algorithm Based on Pulse Protocol with Slot Occupied Probability in Dense Reader Mode*. IEEE. 978-1-4244-2517-4/09.



## Lampiran

### Script simulasi pada MATLAB

```
%menentukan besar T Min dalam satuan timeslot
tMin = 3;
%menentukan besar T Read dalam satuan timeslot
tRead = 4;
%menentukan besar T col dalam satuan timeslot
tCol = 1;
%menentukan besar timeslot dalam milisecond
timeslotLength = 20;
%menentukan nilai Truncate
truncate = 10;
%menentukan nilai propagation delay dan transmission time
tQuery = 0.1;
tBeacon = 0.1;
lQuery = 0.1;
lBeacon = 0.1;
%menentukan besar query dalam satuan byte
querySize = 10;
%menentukan jumlah Reader yang dipakai
nReader = 3;
%menentukan jumlah time slot untuk simulasi
nTimeSlot = 2900;

%inisialisasi BDI Matrix. BDI matrix merupakan suatu matrix yang berisikan
%nilai2 backoff dari setiap reader. Setiap Reader di representasikan dalam
%sebuah baris pada BDI Matrix
bdiMatrix = zeros (nReader,nTimeSlot);
%BDI matrix merupakan suatu matrix yang berisikan
%nilai2 backoff dari setiap reader. Setiap Reader di representasikan dalam
%sebuah baris pada BDI Matrix. Dibawah ini adalah proses pembuatan BDI
%Matrix
for i=1:nReader
bdiMatrix(i,1)= 2^i-1;
c(i)= i;
end

for i = 1 : nTimeSlot
flagC = find(bdiMatrix(:,i)==0);
if size(flagC,1) == 1
c(flagC) = 1;
elseif size (flagC,1) >1
sizeC = size(flagC,1);
for a = 1:sizeC
c(flagC(a)) = c(flagC(a))+1;
if c(flagC(a)) > truncate
c (flagC(a))=1;
end
end
else
c = c;
end

for j =1 : nReader
if bdiMatrix (j,i) == 0
bdiMatrix (j,i+1) = 2^c(j) - 1;
else
bdiMatrix(j,i+1) = bdiMatrix(j,i)-1;
```

```

        end
    end
end

i = 1;
a=0;

%inisialisasi variabel BDI Length Sequence dan BDI Sequence Read
bdiLengthSequence = zeros(1, nTimeSlot);
bdiSequenceRead = zeros (1, nTimeSlot);
%BDI Length Sequence merupakan variabel array, dimana setiap elemennya
%mewakili duration dari BDI (jumlah timeslotnya)
%BDI Sequence Read merupakan variabel array, dimana setiap elemen
%menyatakan Reader mana yang sedang melakukan Reading.
%Dibawah ini adalah proses pembuatan kedua variabel diatas
while a <= nTimeSlot
    flagRead2 = size(find(bdiMatrix(:,i)==0),1);
    if flagRead2 == 1
        bdiLengthSequence(1,i)=tRead + tMin + 1;
        bdiSequenceRead(1,i) = find(bdiMatrix(:,i)==0);
    elseif flagRead2 > 1
        bdiLengthSequence(1,i)=tCol + 1;
        bdiSequenceRead(1,i) = 0;
    else
        bdiLengthSequence(1,i) = 1;
        bdiSequenceRead(1,i) = 0;
    end
    a = sum(bdiLengthSequence);
    i=i+1;
end

%berikut adalah proses untuk menghitung EBDI
b=0;
sumEBDI = 0;
countEBDI = 0;
for i=1:nReader
    b = find(bdiSequenceRead == i);
    bShift = circshift(b, [0,1]);
    bDif=b-bShift-1;
    bDif=bDif(1, [2:size(bDif,2)]);
    sumEBDI = sumEBDI + sum(bDif);
    countEBDI = countEBDI + size(bDif,2);
end

EBDI = sumEBDI/countEBDI;

ETBDI= sum(bdiLengthSequence)/size(find(bdiLengthSequence),2);

ETCYCLE = ((EBDI * ETBDI) + tRead)*timeslotLength

QTREAD = (tRead*timeslotLength- (tRead*(tBeacon +
lBeacon)))/(tQuery+lQuery);

Ps =
size(find(bdiLengthSequence==tRead+tMin+1),2)/size(find(bdiLengthSequence),
2);

S = ((QTREAD * Ps * EBDI)/ETCYCLE)*1000*querySize

```



```
efficiency =  
(size(find(bdiLengthSequence==tRead+tMin+1),2) / (size(find(bdiLengthSequence  
==tRead+tMin+1),2) + (size(find(bdiLengthSequence==tCol+1),2)))) *100
```

