

**IMPLEMENTASI ALGORITMA *SLOTTED ALOHA* SEBAGAI
ANTI-COLLISIONS DALAM MENGIDENTIFIKASI *MULTIPLE*
RFID TAG PADA RFID READER**

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Vynska Amalia Permadi
NIM: 115060901111007



**PROGRAM STUDI TEKNIK INFORMATIKA
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
TAHUN 2016**

PENGESAHAN

IMPLEMENTASI ALGORITMA *SLOTTED ALOHA* SEBAGAI *ANTI-COLLISIONS* DALAM
MENGIDENTIFIKASI *MULTIPLE* RFID TAG PADA RFID READER

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :

Vynska Amalia Permadi
NIM: 115060901111007

Skripsi ini telah diuji dan dinyatakan lulus pada
19 Januari 2016

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Sabriansyah Rizqika Akbar, S.T, M.Eng
NIP: 19820809 201212 1 004

Achmad Basuki, S.T, M.MG, Ph.D
NIP: 19741118 200312 1 002

Mengetahui
Ketua Program Studi Teknik Informatika

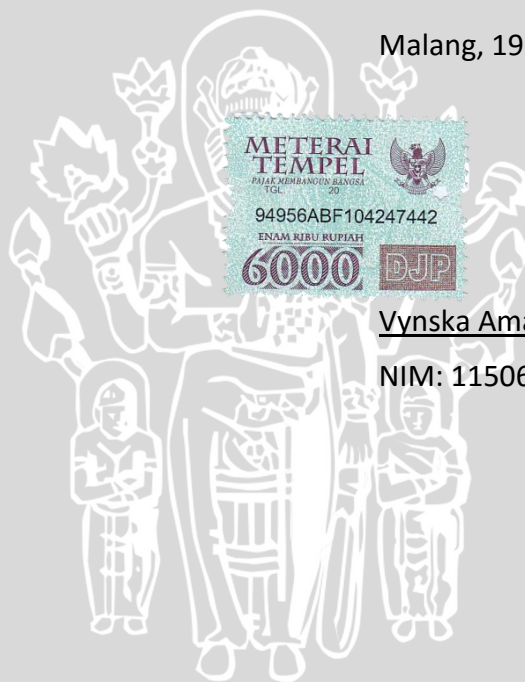
Drs. Marji, M.Si.
NIP: 19670801 199203 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 19 Januari 2016



Vynska Amalia Permadi

NIM: 115060901111007

KATA PENGANTAR

Puji syukur Allah SWT yang telah melimpahkan rahmat, taufik dan hidayah-Nya sehingga laporan skripsi yang berjudul “Implementasi Algoritma *Slotted Aloha* Sebagai *Anti-Collisions* dalam Mengidentifikasi *Multiple RFID Tag* pada *RFID Reader*” dapat terselesaikan.

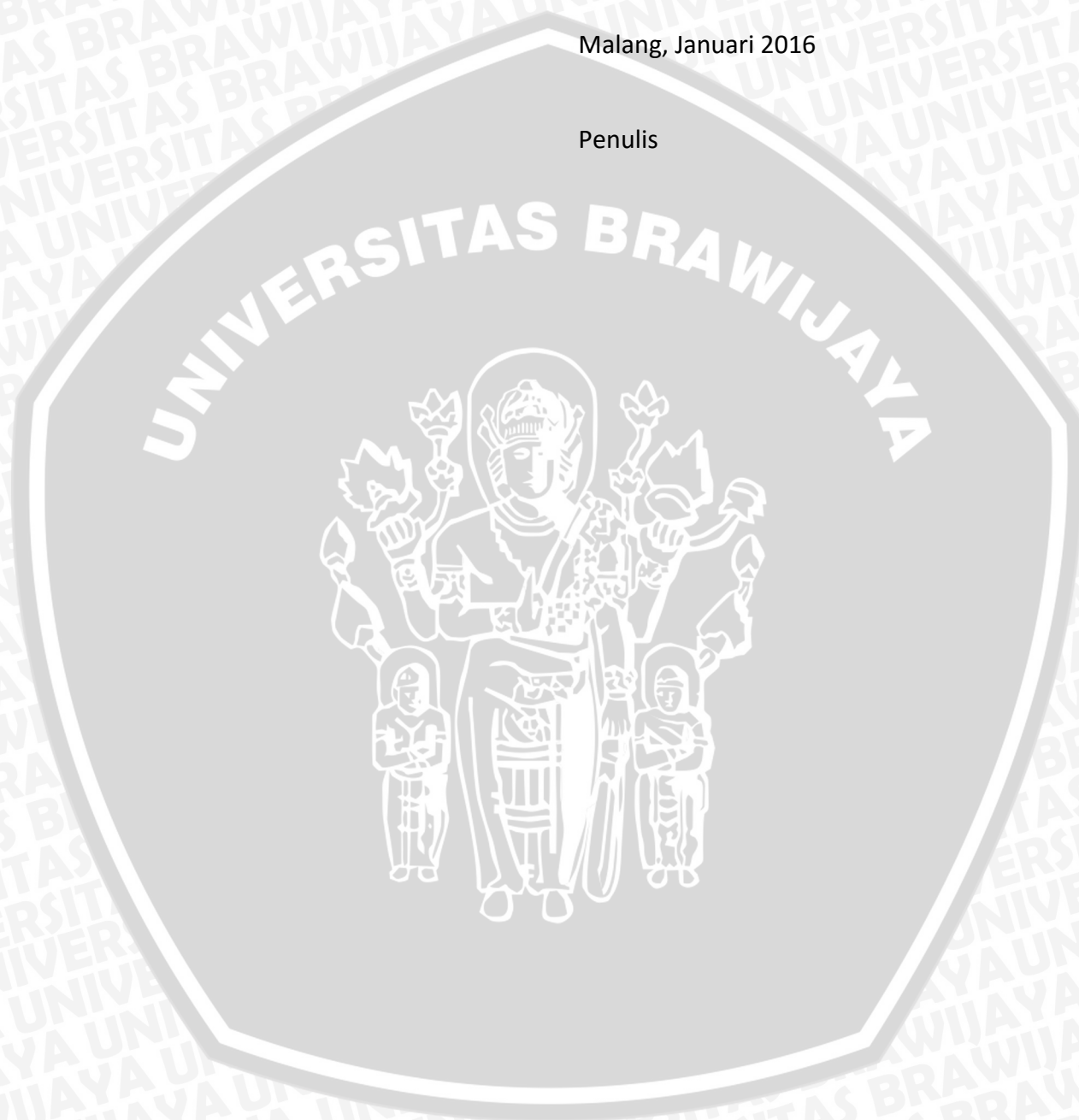
Melalui kesempatan ini, penulis ingin menyampaikan rasa hormat dan terima kasih yang sebesar-besarnya kepada semua pihak yang telah memberikan bantuan baik lahir maupun batin selama penulisan skripsi ini. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan rasa hormat dan terima kasih penulis kepada:

1. Orang tua saya Ir. Devy Permadi, MM dan Lissa Andriyati. Adik saya Muh. Daffa' Lazuardi Permadi. Serta Keluarga yang tiada henti-hentinya memberikan dukungan, doa, semangat, dan kasih sayang demi terselesaikannya skripsi.
2. Bapak Sabriansyah Rizqika Akbar, S.T, M.Eng dan Achmad Basuki, S.T, M.MG, Ph.D selaku dosen pembimbing yang telah berkenan membagi ilmu dan memberikan saran selama pelaksanaan skripsi.
3. Bapak Ir. Sutrisno, M.T., Ir. Heru Nurwasito, M.Kom., Himawat Aryadita, S.T., M.Sc., dan Edy Sasonto, S.Kom., selaku Ketua, Wakil Ketua 1, Wakil Ketua 2 dan Wakil Ketua 3 Fakultas Ilmu Komputer.
4. Bapak Adharul Mutaqqin, S.T., M.T. selaku Ketua Program Studi Sistem Komputer Universitas Brawijaya Malang.
5. Seluruh Dosen Program Studi Sistem Komputer Fakultas Ilmu Komputer Universitas Brawijaya atas kesediaannya membagi ilmu yang bermanfaat kepada penulis.
6. Seluruh civitas akademika Program Studi Sistem Komputer Fakultas Ilmu Komputer Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi di Program Studi Sistem Komputer Universitas Brawijaya dan selama penyelesaian skripsi ini.
7. Teman-teman angkatan 2011 Program Studi Sistem Komputer, terima kasih atas segala bantuannya selama menempuh studi di Program Studi Sistem Komputer Universitas Brawijaya.
8. Semua pihak yang telah membantu dan berbagi ilmu dalam penyelesaian skripsi, yang tidak dapat penulis sebutkan satu per satu.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih banyak kekurangan baik format penulisan maupun isinya. Oleh karena itu, saran dan kritik membangun dari para pembaca senantiasa penulis harapkan guna pengembangan diri. Semoga skripsi ini dapat memberikan manfaat bagi semua pihak, amin.

Malang, Januari 2016

Penulis



ABSTRAK

Teknologi *radio frequency identification* (RFID) saat ini telah banyak digunakan dalam sistem otomatisasi di berbagai bidang. Penggunaan RFID sendiri saat ini telah banyak menggantikan penggunaan *barcode*. Penggunaan RFID mampu menggantikan *barcode* karena kemampuannya yang dapat mengidentifikasi informasi beberapa RFID *tag* dalam waktu yang bersamaan dalam suatu jarak baca tertentu. Di sisi lain, dengan adanya banyak RFID *tag* yang dibaca oleh RFID *reader*, memungkinkan terjadinya kegagalan proses identifikasi, karena akan ada beberapa RFID *tag* yang merespon sinyal dari RFID *reader* dalam waktu yang bersamaan. Respon dari beberapa RFID *tag* yang bersamaan tersebut dapat saling menginterferensi satu sama lain dan memungkinkan tidak terbacanya sinyal yang dikirim oleh masing-masing RFID *tag* secara akurat. Permasalahan tersebut biasanya dikenal sebagai keadaan *tag collisions*. Untuk mengatasi *tag collisions* dibutuhkan suatu algoritma *anti-collisions*. Salah satu algoritma *anti-collisions* yang dapat digunakan adalah algoritma *Slotted Aloha*.

Penelitian ini membahas mengenai implementasi algoritma *Slotted Aloha* untuk meminimalisir terjadinya *tag collisions* dalam suatu sistem RFID. Implementasi dilakukan pada sistem RFID yang terdiri dari Arduino UNO dan *Passive RFID System* 125 KHz. Hasil implementasi algoritma *Slotted Aloha* pada sistem RFID menyatakan bahwa untuk mengidentifikasi RFID *tag* dengan menggunakan algoritma *Slotted Aloha* dibutuhkan iterasi proses identifikasi yang cukup banyak. Banyaknya iterasi proses identifikasi dipengaruhi oleh jumlah RFID *tag* yang diidentifikasi dan jumlah *tag collisions* yang terjadi saat proses identifikasi. Karena algoritma *Slotted Aloha* merupakan algoritma probabilistik, maka hasil identifikasi yang diperoleh merupakan hasil identifikasi yang akurat, meskipun proses identifikasi yang dilakukan lebih lama.

Kata kunci : RFID, *Tag Collisions*, *Slotted Aloha*, Arduino UNO, *Passive RFID System*

ABSTRACT

Radio Frequency Identification (RFID) Technology is now widely used in automation systems in various fields. The use of RFID itself is now largely replaced the use of barcodes. The use of RFID could replace barcodes because of its ability that can identify information from multiple RFID tags at the same time within a certain distance of reading. On the other hand, the presence of many RFID tags identified by RFID reader, enabling the identification process failure, because there will be some RFID tags that respond to signals from the RFID reader at the same time. The response from some of the same RFID tags can be mutually interfere with one another and allow unreadable signals sent by each RFID tag accurately. The problem is usually known as “tag collisions”. To overcome “tag collisions” need an anti-collision algorithm. One of the anti-collision algorithm that can be used is Slotted Aloha algorithm.

This research discusses Slotted Aloha algorithm implementation to minimize “tag collisions” in RFID system. Implementation is done on the RFID system consists of Arduino UNO and Passive RFID System 125 KHz. The results show that considerable iterations required in identification process. The number of iterations in identification process is influenced by the number of RFID tags identified and the number of collisions that occur when the tag identification process. Because Slotted Aloha algorithm is a probabilistic algorithm, the results obtained is the result of an accurate identification, though the identification process is done longer.

Keywords: RFID, Tag Collisions, Slotted Aloha, Arduino UNO, Passive RFID System



DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan	2
1.4 Manfaat.....	2
1.5 Batasan Masalah	2
1.6 Sistematika Pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN	4
2.1 Kajian Pustaka	4
2.2 Dasar Teori.....	4
2.2.1 <i>Radio Frequency Identification (RFID)</i>	4
2.2.2 Arduino UNO.....	9
2.2.3 Arduino IDE	11
2.2.4 <i>Prosedur Multi-Access & Anti-Collisions</i>	13
2.2.5 <i>Algoritma Slotted Aloha</i>	15
BAB 3 METODOLOGI	17
3.1 Perumusan Masalah	17
3.2 Studi Literatur	18
3.3 Perancangan Penelitian	18
3.4 Implementasi	19
3.5 Pengujian	19
3.6 Analisis dan Pembahasan	20

3.7 Pembuatan Laporan Skripsi	20
BAB 4 PERANCANGAN DAN IMPLEMENTASI	21
4.1 Perancangan	21
4.1.1 Analisis Kebutuhan.....	21
4.1.2 Pendefinisian dan Perancangan Penelitian.....	21
4.1.3 Perancangan <i>Hardware</i>	22
4.1.4 Perancangan Komunikasi <i>Hardware</i>	24
4.1.5 Perancangan Program Algoritma <i>Slotted Aloha</i> pada <i>Hardware</i>	25
4.2 Implementasi	25
4.2.1 Spesifikasi <i>Hardware</i>	26
4.2.2 Spesifikasi <i>Software</i>	28
4.2.3 Batasan Penelitian.....	28
4.2.4 Implementasi <i>Hardware</i>	29
4.2.5 Implementasi Komunikasi <i>Hardware</i>	29
4.2.6 Implementasi Algoritma <i>Slotted Aloha</i> pada <i>Hardware</i>	31
BAB 5 PENGUJIAN DAN ANALISIS.....	34
5.1 Pengujian <i>Hardware</i> dan Komunikasi <i>Hardware</i>	34
5.1.1 Tujuan Pengujian.....	34
5.1.2 Prosedur Pengujian	34
5.1.3 Pelaksanaan Pengujian.....	35
5.1.4 Hasil Pengujian	35
5.2 Pengujian Identifikasi <i>RFID Tag</i>	37
5.2.1 Tujuan Pengujian.....	37
5.2.2 Prosedur Pengujian	37
5.2.3 Pelaksanaan Pengujian.....	37
5.2.4 Hasil Pengujian	37
5.3 Pengujian Identifikasi <i>Multiple RFID Tag</i>	39
5.3.1 Tujuan Pengujian.....	39
5.3.2 Prosedur Pengujian	39
5.3.3 Pelaksanaan Pengujian.....	40
5.3.4 Hasil Pengujian	40



5.4 Pengujian Identifikasi <i>Multiple RFID Tag</i> dengan Algoritma <i>Slotted Aloha</i>	44
5.4.1 Tujuan Pengujian.....	44
5.4.2 Prosedur Pengujian	44
5.4.3 Pelaksanaan Pengujian.....	45
5.4.4 Hasil Pengujian	45
5.5 Analisis Hasil Pengujian.....	49
5.5.1 Analisis Hasil Pengujian <i>Hardware</i> dan Komunikasi <i>Hardware</i> ..	49
5.5.2 Analisis Hasil Pengujian Identifikasi <i>RFID Tag</i>	49
5.5.3 Analisis Hasil Pengujian Identifikasi <i>Multiple RFID Tag</i>	49
5.5.4 Analisis Hasil Pengujian Identifikasi <i>Multiple RFID Tag</i> dengan Algoritma <i>Slotted Aloha</i>	50
BAB 6 PENUTUP	54
6.1 Kesimpulan.....	54
6.2 Saran	55
DAFTAR PUSTAKA.....	56



DAFTAR TABEL

Tabel 4.1 Spesifikasi Arduino UNO R3	26
Tabel 4.2 Spesifikasi RFID Reader Seeedstudio Electronic Brick 125 KHz	27
Tabel 4.3 Spesifikasi RFID Card sTag 125 KHz.....	27
Tabel 4.4 Spesifikasi Laptop	28
Tabel 4.5 Daftar Spesifikasi Software	28
Tabel 5.1 Hasil Pengujian Jarak Baca RFID Reader.....	39
Tabel 5.2 Hasil Pengujian Identifikasi 2 RFID Tag pada RFID Reader.....	42
Tabel 5.3 Hasil Pengujian Identifikasi 3 RFID Tag pada RFID Reader.....	42
Tabel 5.4 Hasil Pengujian Identifikasi 4 RFID Tag pada RFID Reader.....	43
Tabel 5.5 Hasil Pengujian Identifikasi 5 RFID Tag pada RFID Reader.....	43
Tabel 5.6 Hasil Pengujian Identifikasi 2 RFID Tag pada RFID Reader Dengan Algoritma Slotted Aloha	47
Tabel 5.7 Hasil Pengujian Identifikasi 3 RFID Tag pada RFID Reader Dengan Algoritma Slotted Aloha	47
Tabel 5.8 Hasil Pengujian Identifikasi 4 RFID Tag pada RFID Reader Dengan Algoritma Slotted Aloha	48
Tabel 5.9 Hasil Pengujian Identifikasi 5 RFID Tag pada RFID Reader Dengan Algoritma Slotted Aloha	48
Tabel 5.10 Analisis Efisiensi Sistem Saat Proses Identifikasi 2 RFID Tag pada RFID Reader dengan Algoritma Slotted Aloha.....	51
Tabel 5.11 Analisis Efisiensi Sistem Saat Proses Identifikasi 3 RFID Tag pada RFID Reader dengan Algoritma Slotted Aloha.....	51
Tabel 5.12 Analisis Efisiensi Sistem Saat Proses Identifikasi 4 RFID Tag pada RFID Reader dengan Algoritma Slotted Aloha.....	52
Tabel 5.13 Analisis Efisiensi Sistem Saat Proses Identifikasi 5 RFID Tag pada RFID Reader dengan Algoritma Slotted Aloha.....	52

DAFTAR GAMBAR

Gambar 2.1 Perbandingan dari Beberapa Teknologi <i>Automatic Identification System</i>	5
Gambar 2.2 Komponen dan Ilustrasi Komunikasi Dasar pada Sistem RFID	6
Gambar 2.3 Layout <i>Transponder</i>	6
Gambar 2.4 RFID <i>Card Transponder</i>	7
Gambar 2.5 Klasifikasi Sistem RFID Berdasarkan Fungsionalitas.....	7
Gambar 2.6 Prinsip Kerja <i>Master-Slave</i> Antara Aplikasi <i>Software</i> , <i>Reader</i> , dan <i>Transponder</i>	9
Gambar 2.7 Arduino Uno R3	9
Gambar 2.8 Tampilan Software Arduino IDE	12
Gambar 2.9 <i>Broadcast Mode</i>	14
Gambar 2.10 <i>Multi-Access Mode</i>	14
Gambar 2.11 Perbandingan <i>Throughput</i> Algoritma <i>Aloha</i> dan <i>Slotted Aloha</i>	16
Gambar 3.1 <i>Flowchart</i> Pelaksanaan Penelitian	17
Gambar 4.1 Diagram Blok Ilustrasi Implementasi Algoritma <i>Slotted Aloha</i> pada Sistem RFID	22
Gambar 4.2 Diagram Blok Perancangan <i>Hardware</i>	23
Gambar 4.3 Diagram Skematik Arduino UNO R3.	23
Gambar 4.4 Diagram Blok Modul RFID <i>Reader</i> Seeedstudio Electronic Brick 125 KHz	24
Gambar 4.5 <i>Flowchart</i> Perancangan Komunikasi <i>Hardware</i>	24
Gambar 4.6 <i>Flowchart</i> Perancangan Algoritma <i>Slotted Aloha</i> pada <i>Hardware</i> ...	25
Gambar 4.7 Implementasi Board Arduino UNO dan Modul RFID <i>Reader</i> Seeedstudio Electronic Brick 125 KHz	29
Gambar 4.8 <i>Flowchart</i> Komunikasi Serial Antara RFID <i>Reader</i> dan Arduino	30
Gambar 4.9 <i>Source Code</i> Komunikasi Serial Antara RFID <i>Reader</i> dan Arduino....	31
Gambar 4.10 <i>Flowchart</i> Implementasi Algoritma <i>Slotted Aloha</i> pada Arduino UNO	32
Gambar 4.11 <i>Source Code</i> Implementasi Algoritma <i>Slotted Aloha</i> pada Arduino UNO	33
Gambar 5.1 Diagram Blok Pengujian <i>Hardware</i> dan Komunikasi <i>Hardware</i>	34
Gambar 5.2 Konfigurasi Pengujian <i>Hardware</i>	35

Gambar 5.3 Langkah Pengujian Komunikasi <i>Hardware</i>	36
Gambar 5.4 Hasil Pengujian <i>Hardware</i> dan Komunikasinya dalam <i>Serial Monitor</i>	36
Gambar 5.5 Proses Pengujian Identifikasi RFID Tag	38
Gambar 5.6 Hasil Pengujian Identifikasi RFID Tag	38
Gambar 5.7 Proses Pengujian Identifikasi <i>Multiple RFID Tag</i>	41
Gambar 5.8 Hasil Proses Pengujian Identifikasi <i>Multiple RFID Tag</i> pada <i>Serial Monitor</i>	41
Gambar 5.9 Diagram Blok Pengujian Identifikasi <i>Multiple Tag</i> dengan Algoritma <i>Slotted Aloha</i>	44
Gambar 5.10 Proses Pengujian Identifikasi <i>Multiple RFID Tag</i> dengan Algoritma <i>Slotted Aloha</i>	46
Gambar 5.11 Hasil Proses Pengujian Identifikasi <i>Multiple RFID Tag</i> dengan Algoritma <i>Slotted Aloha</i> pada <i>Serial Monitor</i>	46



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Identifikasi adalah konsep umum yang banyak digunakan di berbagai hal termasuk pada sistem komputer. Contohnya, mesin absensi *fingerprint* dapat mengidentifikasi *user* melalui *fingerprint* yang nilainya unik dan berbeda. *automatic identification system* (AIS) merupakan solusi yang saat ini banyak digunakan dalam berbagai bidang, seperti: *asset tracking*, *stock inventory* dan *toll ticketing*. Suatu sistem identifikasi otomatis akan melakukan proses identifikasi yang lebih mudah dan cepat bahkan tidak dibutuhkan kontak fisik dari objek yang akan diidentifikasi.

Barcode merupakan salah satu AIS yang masih banyak digunakan saat ini. Namun, *barcode* masih memiliki beberapa kekurangan, yaitu: hanya dapat mengidentifikasi satu objek dalam satu waktu, hanya dapat menyimpan nilai identifikasi dalam jumlah kecil, dan tidak dapat diidentifikasi jika kotor atau basah.

Radio frequency identification (RFID) adalah sebuah perangkat identifikasi otomatis yang memanfaatkan penggunaan *radio frequency* (RF) untuk mentransfer data. RFID merupakan alternatif AIS yang dapat mengatasi keterbatasan *barcode*. RFID dapat mengidentifikasi beberapa RFID *tag* dalam satu waktu. Namun, adanya banyak RFID *tag* yang dibaca oleh RFID *reader*, memungkinkan terjadinya kegagalan proses identifikasi. Kegagalan proses identifikasi tersebut dikarenakan oleh beberapa RFID *tag* yang merespon sinyal dari RFID *reader* dalam waktu yang bersamaan. Respon dari beberapa RFID *tag* yang bersamaan tersebut dapat saling menginterferensi satu sama lain dan memungkinkan tidak terbacanya sinyal yang dikirim oleh masing-masing RFID *tag*. Permasalahan tersebut biasanya dikenal sebagai *tag collisions*.

Untuk mengatasi permasalahan *tag collisions*, dibutuhkan algoritma *anti-collisions*. Secara umum terdapat dua tipe algoritma, yaitu: probablistik dan deterministik. Algoritma probablistik digunakan untuk permasalahan dengan ruang solusi yang besar, bahkan tak terbatas. Algoritma probablistik akan berusaha menemukan solusi yang terbaik tanpa melebihi batasan waktu yang ditentukan. Solusi yang diberikan belum tentu yang paling optimal, namun dapat diterima kebenarannya. Salah satu contoh algoritma *anti-collisions* dengan tipe probablistik adalah algoritma *Aloha*. Algoritma deterministik hanya memiliki satu langkah pemrosesan, jika tidak ada langkah yang harus dieksekusi maka algoritma dianggap selesai. Algoritma ini akan selalu menghasilkan solusi yang tetap untuk suatu input yang diberikan. Algoritma deterministik digunakan untuk menyelesaikan masalah dengan ruang solusi yang tidak terlalu besar (Hasad, 2011). Contoh penggunaan algoritma deterministik adalah algoritma *Binary Tree*.

Untuk menghindari *tag collisions* dalam proses pembacaan *multiple* RFID *tag* akan diimplementasikan algoritma *Slotted Aloha* pada sistem RFID pasif berbasis

Arduino. Implementasi algoritma *Slotted Aloha* diharapkan dapat mengatasi terjadinya *tag collisions*.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang dipaparkan maka rumusan masalah yang perlu diperhatikan adalah:

1. Bagaimana mengidentifikasi *multiple RFID tag* secara bersamaan.
2. Implementasi algoritma *Slotted Aloha* dalam sistem RFID.
3. Pengujian algoritma *Slotted Aloha* dalam sistem RFID.

1.3 Tujuan

Tujuan dilakukannya penelitian ini sesuai dengan rumusan masalah yang telah diberikan yaitu:

1. Mengimplementasikan algoritma *Slotted Aloha* dalam sistem RFID.
2. Analisis performansi algoritma *Slotted Aloha* dalam sistem RFID.

1.4 Manfaat

Penelitian ini diharapkan dapat mengimplementasikan algoritma *Slotted Aloha* sebagai *anti-collisions* pada sistem RFID. Dengan berhasilnya penerapan algoritma tersebut, identifikasi *multiple RFID tag* dapat dilakukan dengan akurat. Selain itu penelitian ini diharapkan pula dapat menjadi bahan referensi dan tambahan pengetahuan mengenai identifikasi *multiple RFID tag* pada sistem RFID nyata untuk mengantisipasi *tag collisions*.

1.5 Batasan Masalah

Agar pembahasan tidak melebar dari latar belakang dan terfokus dengan apa yang berkaitan dengan penelitian, maka batasan dalam penelitian ini, yaitu:

1. Implementasi algoritma *Slotted Aloha* untuk mengidentifikasi *multiple RFID tag* menggunakan Arduino dan Sistem RFID pasif: Modul RFID reader Seeedstudio Electronic Brick 125 KHz, dan RFID tag card 125 KHz.
2. Performa algoritma *Slotted Aloha* untuk mengidentifikasi *multiple RFID tag* dilihat dari penghitungan efisiensi sistem dan akurasi hasil identifikasi RFID tag.

1.6 Sistematika Pembahasan

Sistematika pembahasan digunakan untuk memberikan gambaran dan uraian dari penulisan skripsi ini secara garis besar, yang meliputi beberapa bab sebagai berikut:

Bab I: Pendahuluan

Bab ini berisi mengenai latar belakang pembuatan skripsi, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika penulisan.

Bab II: Landasan Kepustakaan

Menguraikan tentang referensi dan dasar teori yang mendasari penulisan skripsi: "Implementasi Algoritma *Slotted Aloha* Sebagai *Anti-Collisions* dalam Mengidentifikasi *Multiple RFID Tag* pada *RFID Reader*".

Bab III: Metodologi

Menguraikan dan membahas langkah kerja yang dilakukan dalam penulisan skripsi, yang terdiri dari: perumusan masalah, studi literatur, perancangan penelitian, implementasi, pengujian, analisis hasil, dan pembuatan laporan akhir.

Bab IV: Perancangan dan Implementasi

Membahas perancangan dan implementasi dari penelitian: "Implementasi Algoritma *Slotted Aloha* Sebagai *Anti-Collisions* dalam Mengidentifikasi *Multiple Tag* pada *RFID Reader*".

Bab V: Pengujian dan Analisis

Bab ini berisi pengujian dan analisis terhadap rancangan penelitian: "Implementasi Algoritma *Slotted Aloha* Sebagai *Anti-Collisions* dalam Mengidentifikasi *Multiple RFID Tag* pada *RFID Reader*" yang telah direalisasikan.

Bab VI: Penutup

Bab ini berisi kesimpulan yang diperoleh dari hasil implementasi dan pengujian algoritma *Slotted Aloha* sebagai *anti-collisions* dalam mengidentifikasi *multiple RFID tag* pada *RFID reader* yang dikembangkan dalam skripsi ini, serta saran-saran untuk pengembangan lebih lanjut.

BAB 2 LANDASAN KEPUSTAKAAN

Bab ini berisi landasan kepastakaan yang meliputi kajian pustaka dan dasar teori yang diperlukan. Kajian pustaka membahas penelitian yang telah ada dan yang diusulkan. Dasar teori membahas teori yang diperlukan untuk menyusun penelitian yang diusulkan.

2.1 Kajian Pustaka

Pada jurnal yang berjudul "*Analysis and Simulation of Slotted Aloha-Based RFID Anti-Collisions*" dilakukan pengujian dan analisis terhadap algoritma *Slotted Aloha* untuk mengatasi *tag collisions*. Pengujian algoritma dilakukan dengan mensimulasikan algoritma tersebut pada Matlab. Tujuan dari penelitian tersebut adalah menganalisis perkiraan waktu yang dibutuhkan untuk mengidentifikasi sejumlah RFID *tag* dengan menerapkan algoritma *Slotted Aloha* sebagai *anti-collisions*. Hasil penelitian tersebut adalah pernyataan bahwa dalam suatu proses identifikasi RFID menggunakan algoritma *Slotted Aloha*, waktu yang dibutuhkan untuk menyelesaikan identifikasi RFID *tag* bergantung pada berapa jumlah RFID *tag* yang diidentifikasi dan *interference*. *Interference* pada penelitian tersebut berupa nilai konstan yang diuji sebagai perkiraan nilai interferensi yang ada dalam suatu proses identifikasi RFID.

Pada penelitian ini, penulis mencoba untuk mengembangkan penelitian yang telah dilakukan sebelumnya, yaitu dengan mengimplementasikan algoritma *Slotted Aloha* pada *hardware*. Hasil dari penelitian ini adalah analisis berupa efisiensi sistem dalam proses identifikasi RFID *tag* dengan menggunakan algoritma *Slotted Aloha*. Dengan mengimplementasikan algoritma pada *hardware*, dapat dianalisis bagaimana efisiensi penggunaan algoritma *Slotted Aloha* pada suatu sistem RFID.

2.2 Dasar Teori

Dasar teori berisi teori-teori yang berhubungan dengan perancangan skripsi: "*Implementasi Algoritma Slotted Aloha Sebagai Anti-Collisions dalam Mengidentifikasi Multiple RFID Tag pada RFID Reader*".

2.2.1 Radio Frequency Identification (RFID)

Dasar Penggunaan Sistem RFID

Sistem RFID merupakan salah satu teknologi *automatic identification system* (AIS). Prinsip kerjanya memiliki beberapa kesamaan dengan teknologi AIS lainnya

yaitu *smart card*. Sama seperti pada *smart card*, data identifikasi disimpan pada suatu perangkat elektronik yang disebut *transponder*. Perbedaan dari kedua sistem ini terletak pada penggunaan catu daya dan media pertukaran data. Pada *smart card*, pertukaran data dilakukan menggunakan medan magnet atau gelombang elektromagnetik. Sedangkan RFID menggunakan gelombang radio dan teknik radar yang memiliki kemampuan untuk mendeteksi sumber energi dari pemancar disekitarnya. Pada gambar 2.1 dapat dilihat kelebihan dan kekurangan dari sistem RFID dan perbandingannya dengan teknologi AIS lainnya. *Smart card* dan sistem RFID memiliki kemiripan, namun ada beberapa hal yang membedakan kinerja keduanya, yaitu: pengaruh terhadap kotoran/debu, pengaruh arah posisi identifikasi, dan lain-lain.

System parameters	Barcode	OCR	Voice recog.	Biometry	Smart card	RFID systems
Typical data quantity (bytes)	1-100	1-100	—	—	16-64 k	16-64 k
Data density	Low	Low	High	High	Very high	Very high
Machine readability	Good	Good	Expensive	Expensive	Good	Good
Readability by people	Limited	Simple	Simple	Difficult	Impossible	Impossible
Influence of dirt/damp	Very high	Very high	—	—	Possible (contacts)	No influence
Influence of (opt.) covering	Total failure	Total failure	—	Possible	—	No influence
Influence of direction and position	Low	Low	—	—	Unidirectional	No influence
Degradation/wear	Limited	Limited	—	—	Contacts	No influence
Purchase cost/reading electronics	Very low	Medium	Very high	Very high	Low	Medium
Operating costs (e.g. printer)	Low	Low	None	None	Medium (contacts)	None
Unauthorised copying/modification	Slight	Slight	Possible* (audio tape)	Impossible	Impossible	Impossible
Reading speed (including handling of data carrier)	Low ~4 s	Low ~3 s	Very low >5 s	Very low >5-10 s	Low ~4 s	Very fast ~0.5 s
Maximum distance between data carrier and reader	0-50 cm	<1 cm Scanner	0-50 cm	Direct contact**	Direct contact	0-5-m, microwave

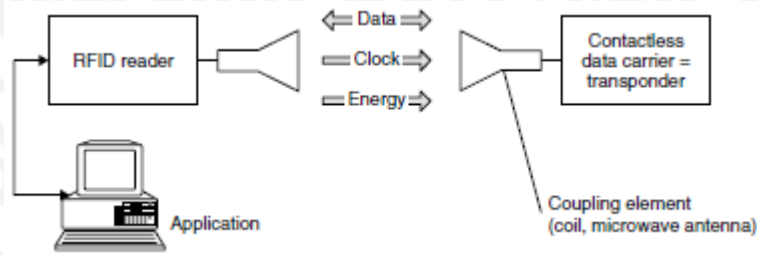
*The danger of 'Replay' can be reduced by selecting the text to be spoken using a random generator, because the text that must be spoken is not known in advance.
**This only applies for fingerprint ID. In the case of retina or iris evaluation direct contact is not necessary or possible.

Gambar 2.1 Perbandingan dari Beberapa Teknologi *Automatic Identification System* (Finkenzyler, 2003)

Sistem RFID terdiri atas 2 komponen, yaitu:

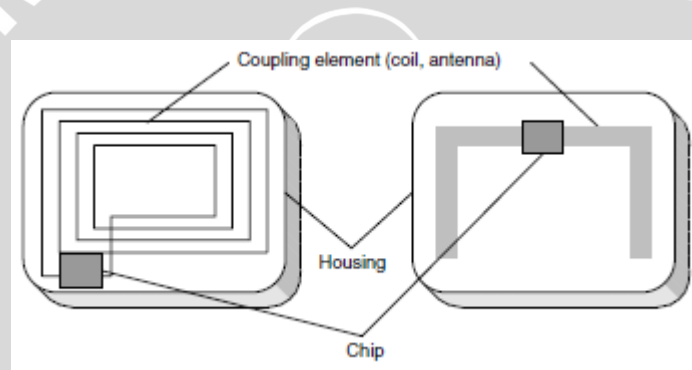
1. *Transponder*. Merupakan perangkat yang diletakkan atau ditanamkan pada objek yang akan diidentifikasi.
2. *Interrogator/Reader*. Merupakan perangkat “penangkap” data yang berfungsi untuk mengidentifikasi objek.

RFID *reader* merupakan sebuah modul *radio frequency* (RF) yang terdiri dari *transmitter* dan *receiver, control unit, dan coupling element*. Beberapa RFID *reader* juga memiliki *interface* tambahan, seperti: RS 232, RS 485, dan lain-lain. *Interface* tersebut digunakan untuk meneruskan data yang diterima ke sistem lainnya, seperti: PC, *robot control system*, dan lain-lain. Komponen dan ilustrasi komunikasi dasar pada sistem RFID dapat dilihat pada Gambar 2.2.



Gambar 2.2 Komponen dan Ilustrasi Komunikasi Dasar pada Sistem RFID (Finkenzeller, 2003)

Transponder merupakan perangkat yang memiliki data identifikasi pada sistem RFID. *Transponder* terdiri dari *coupling element* dan *microchip* elektronik, seperti yang diilustrasikan pada Gambar 2.3.



Gambar 2.3 Layout *Transponder*. Kiri : *Transponder* dengan *Coupling Element* Induktif Berupa *Coil Antenna* ; Kanan : *Transponder* dengan *Coupling Element* Microwave Berupa *Dipolar Antenna* (Finkenzeller, 2003)

Transponder yang tidak memiliki catu daya sendiri akan berada pada keadaan pasif ketika tidak berada di *interrogation zone*. *Transponder* hanya akan aktif saat berada pada *interrogation zone* RFID reader. Catu daya yang dibutuhkan untuk mengaktifkan *transponder* dikirimkan melalui *coupling unit* (tanpa perlu ada kontak secara fisik) dalam bentuk *timing pulse* dan data.

Fitur Sistem RFID

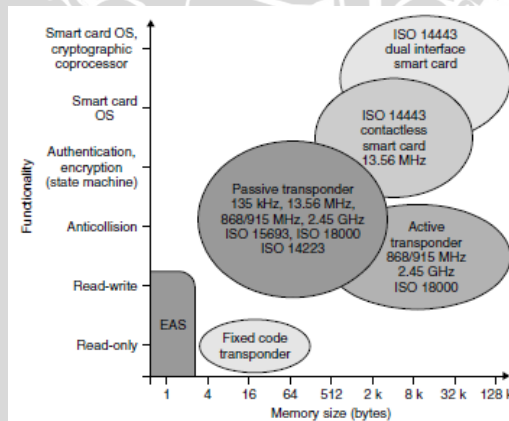
Terdapat berbagai macam bentuk *transponder*, seperti: *key*, *card*, *label*, dan lain-lain. RFID *card* merupakan salah satu *transponder* yang paling banyak digunakan. Bentuknya berupa kartu seukuran kartu kredit atau kartu telepon. Ukurannya kurang lebih 85.72mm x 54.03mm x 0.76mm, seperti pada Gambar 2.4. Salah satu keuntungan dari penggunaan RFID *card* pada sistem RFID induktif adalah *coil area* yang besar, sehingga dapat meningkatkan *range* identifikasi RFID

card. Proses pembuatan RFID tag dilakukan dengan melakukan laminasi transponder pada empat foil PVC. Tiap foil dipanaskan pada tekanan tinggi dan suhu diatas 100°C agar terpasang secara permanen.



Gambar 2.4 Kiri : Layout RFID Card dengan Transponder Module dan Antenna (Finkenzeller, 2003) ; Kanan : RFID Card Transponder

Berdasarkan range identifikasi serta ukuran memorinya, terdapat banyak variasi spektrum yang bisa digunakan pada sistem RFID. Dari berbagai variasi spektrum tersebut, dapat dirangkum menjadi dua klasifikasi berdasarkan fungsionalitas tiap spektrum, yaitu: *low-end* dan *high-end system* seperti pada Gambar 2.5.



Gambar 2.5 Berdasarkan Fungsionalitasnya, Sistem RFID Dapat Diklasifikasikan Menjadi Low-End System dan High-End System (Finkenzeller, 2003)

Low-end system terdiri dari *electronic article surveillance systems* (EAS system) dan *read-only transponder*. EAS digunakan untuk memeriksa dan memonitor transponder pada *interrogation zone* RFID reader. *Read-only transponder* memiliki *microchip* yang berisi data identifikasi berupa *serial number* yang “tertanam”, bersifat unik, dan terdiri dari beberapa *bytes*. Jika berada di *interrogation zone*,

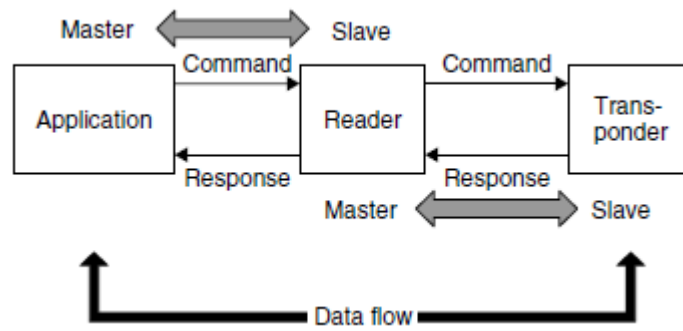
maka *transponder* melakukan *broadcast serial number* secara terus menerus. Komunikasi dari *reader* dan *transponder* merupakan komunikasi satu arah. Pada prakteknya, hanya boleh ada satu *transponder* pada suatu *interrogation zone*. Jika terdapat dua atau lebih *transponder* melakukan transmisi maka terjadi *data collisions*. Karena keterbatasan tersebut, *read-only transponder* cocok diaplikasikan pada proses identifikasi yang membutuhkan nilai identifikasi yang unik.

Fungsi *read-only* tidak hanya dapat beroperasi pada *low-end system*. Semua frekuensi dapat menggunakan fungsi tersebut. Pada *mid-range system*, selain fungsi *read-only*, dapat juga digunakan fungsi *write*, *anti-collisions*, dan *authentication* atau *encryption*. *Transponder* pada *mid-range system* dapat memproses perintah *read* sederhana dan melakukan *write* pada *memory*. Secara umum, *transponder* juga dapat melakukan prosedur *anti-collisions* sehingga beberapa *transponder* yang berada pada *interrogation zone* dapat diidentifikasi tanpa saling menginterferensi satu sama lain. Prosedur kriptologi seperti *authentication* dan *data stream encryption* juga dapat dilakukan pada sistem ini. Sistem ini dapat beroperasi pada semua frekuensi pada sistem RFID.

High-end system digunakan pada sistem yang menggunakan mikroprosesor dan *smart card operating system (smart card OS)*. *High-end system* biasanya menggunakan frekuensi 13,56MHz dan menggunakan standard ISO 14443 untuk komunikasi data antara *transponder* dan *reader*.

Reader

Software application membutuhkan *reader* untuk membaca data dari *transponder* atau menulis data pada *transponder*. Pada operasi *read* dan *write* terhadap *transponder*, digunakan prinsip *master-slave*, seperti yang tercantum pada Gambar 2.6. Semua aktifitas *reader* dan *transponder* dimulai oleh *software application*. *Software application* berperan sebagai *master*, dan *reader* sebagai *slave*. *Reader* hanya akan aktif jika ada perintah *read* atau *write* yang diterima dari *master*. Untuk mengeksekusi perintah yang diterima dari *software application*, *reader* akan melakukan komunikasi dengan *transponder*. Pada bagian ini *reader* berperan sebagai *master* sedangkan *transponder* berperan sebagai *slave*. *Transponder* hanya akan merespon perintah dari *reader* dan tidak dapat aktif tanpa ada perintah dari *reader*. Fungsi utama *reader* adalah untuk mengaktifkan *transponder*, menyusun urutan komunikasi dengan *transponder*, dan mengirimkan data dari *transponder* ke *software application*.



Gambar 2.6 Prinsip Kerja *Master-Slave* Antara Aplikasi *Software*, *Reader*, dan *Transponder* (Finkenzeller, 2003)

2.2.2 Arduino UNO

Arduino UNO adalah *board* berbasis mikrokontroler ATmega328. *Board* ini memiliki 14 digital *input/output* pin (dimana 6 pin dapat digunakan sebagai *output* PWM), 6 *input* analog, 16 MHz kristal osilator, koneksi USB, *jack power*, dan tombol *reset*. Pin-pin ini berisi semua fungsionalitas pendukung yang diperlukan untuk mendukung mikrokontroler.



Gambar 2.7 Arduino Uno R3. Kiri: Sisi Depan; Kanan: Sisi Belakang (arduino, n.d.)

Secara umum Arduino terdiri dari dua bagian, yaitu:

1. *Hardware*. Berupa papan *input/output* (I/O).
2. *Software*. *Software* Arduino meliputi IDE untuk menulis program, *driver* untuk koneksi dengan komputer, dan contoh program dan *library* untuk pengembangan program.

Arduino UNO dapat digunakan melalui koneksi USB atau dengan sebuah catu daya eksternal. Catu daya eksternal (non-USB) dapat diperoleh dari adaptor AC ke DC atau baterai. Adaptor dapat dihubungkan dengan menyambungkan *center-*



positive plug 2,1 mm ke *power jack* pada *board*. Kabel *lead* dari baterai dapat dimasukkan dalam *header/kepala pin ground* (Gnd) dan pin Vin dari konektor *power*.

Board Arduino UNO dapat beroperasi pada catu daya eksternal 6 sampai 20 Volt. Jika disuplai dengan daya yang lebih kecil dari 7 V, *board* Arduino Uno bisa menjadi tidak stabil. Jika menggunakan suplai yang lebih besar dari 12 Volt, *voltage regulator* bisa mengalami kelebihan muatan panas dan membahayakan *board* Arduino UNO. *Range* yang direkomendasikan adalah 7 sampai 12 Volt. (arduino, n.d.)

Pin-pin yang dimiliki Arduino UNO adalah sebagai berikut:

- a) Vin. Tegangan *input* ke *board* Arduino ketika *board* sedang menggunakan sumber catu daya eksternal (seperti 5 Volt dari koneksi USB atau sumber tenaga lainnya yang diatur). Suplai tegangan dapat dilakukan melalui pin ini, atau jika penyuplaian tegangan melalui *power jack*, aksesnya dapat diatur melalui pin ini.
- b) 5V. Pin *output* ini merupakan tegangan 5 Volt yang diatur dari regulator pada *board*. *Board* dapat disuplai dengan salah satu suplai dari DC *power jack* (7-12V), USB connector (5V), atau pin Vin dari *board* (7-12). Penyuplaian tegangan melalui pin 5V atau 3,3V *bypass* regulator, dan dapat membahayakan *board*. Hal tersebut tidak dianjurkan.
- c) 3V3. Sebuah suplai 3,3 Volt dihasilkan oleh regulator pada *board*. Arus maksimum yang dapat dilalui adalah 50 mA.
- d) GND. Pin *ground*.

ATmega328 mempunyai penyimpanan sebesar 32 KB bersifat *non-volatile* yang digunakan untuk menyimpan program yang diupload dari komputer (0,5 KB digunakan untuk *bootloader*). ATmega328 juga mempunyai 2 KB SRAM yang bersifat *volatile* (hilang saat daya dimatikan), biasanya digunakan oleh variabel-variabel di dalam program. Dan 1 KB EEPROM (yang dapat dibaca dan ditulis (RW/read and written)). (arduino, n.d.)

Seluruh pin digital pada Arduino UNO dapat digunakan sebagai *input* dan *output* dengan menggunakan fungsi *pinMode()*, *digitalWrite()*, dan *digitalRead()*. Tiap pin tersebut beroperasi pada tegangan 5 Volt. Setiap pin dapat memberikan atau menerima arus maksimum 40 mA dan mempunyai sebuah resistor *pull-up* (terputus secara default) sebesar 20-50 kOhm. Berikut adalah pembahasan fungsi dari 14 pin yang dimiliki Arduino UNO:

- a) Serial: 0 (RX) dan 1 (TX). Digunakan untuk menerima (RX) dan memancarkan (TX) serial data TTL (*transistor-transistor logic*). Kedua pin ini biasanya dihubungkan ke pin-pin yang sesuai dengan *chip* serial Atmega8U2 USB-to-TTL.

- b) *External Interrupts*: 2 dan 3. Pin-pin ini dapat dikonfigurasi untuk memicu *interrupt* pada program.
- c) PWM: 3, 5, 6, 9, 10, dan 11. Memberikan 8-bit PWM *output* dengan fungsi `analogWrite()`.
- d) SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Pin-pin ini mendukung komunikasi SPI menggunakan SPI *library*.
- e) LED: 13. Terdapat sebuah LED yang terpasang dan terhubung ke pin digital 13. Ketika pin bernilai HIGH maka LED akan menyala, ketika pin bernilai LOW maka LED akan mati.

Arduino UNO juga mempunyai 6 *input* analog, yang diberi label A0 sampai dengan A5. Secara default, 6 *input* analog tersebut mengukur tegangan dari *ground* sampai tegangan 5 Volt, namun dimungkinkan juga melakukan penggantian batas atas dari *rangennya* dengan menggunakan pin AREF dan fungsi `analogReference()`. Selain itu, ada pula pin yang mempunyai fungsi spesifik seperti pin A4 atau SDA dan pin A5 atau SCL. Pin tersebut mendukung komunikasi TWI dengan menggunakan *wire library*. Ada juga beberapa pin lainnya pada *board*, yaitu: AREF yang berfungsi sebagai referensi tegangan untuk *input* analog (penggunaannya disertai dengan `analogReference`), dan *reset* untuk mereset mikrokontroler. (arduino, n.d.)

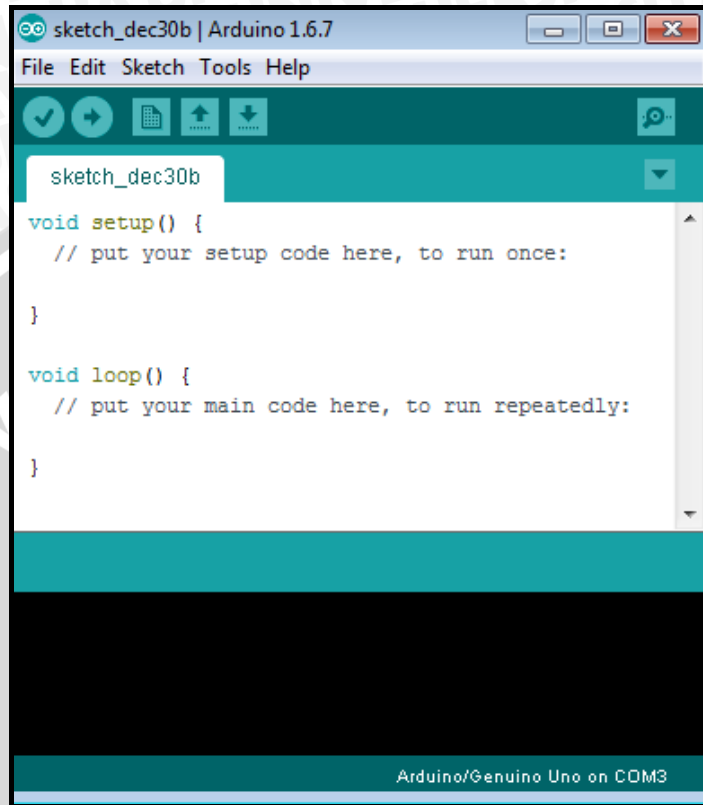
Arduino UNO memiliki beberapa alternatif komunikasi yang dapat digunakan sebagai fasilitas komunikasi antara *board* dan komputer. ATmega328 menyediakan UART TTL (5V) untuk komunikasi serial yang akan aktif pada pin digital 0 (RX) dan 1 (TX). Dibutuhkan *software* Arduino IDE dengan *serial monitor* untuk menampilkan data yang dikirim ke *board* dan dari *board*. *Library* `SoftwareSerial` merupakan *library* yang dapat digunakan untuk komunikasi serial pada pin digital Arduino. (arduino, n.d.)

2.2.3 Arduino IDE

Arduino IDE adalah *software* yang dibuat dengan menggunakan bahasa pemrograman *java* berdasarkan pengolahan `avr-gcc` dan perangkat lunak *open source* lainnya. Arduino IDE terdiri dari:

1. Editor program. Merupakan sebuah *window* yang memungkinkan pengguna menulis dan mengubah program dalam bahasa pemrograman.
2. *Verify/Compiler*. Merupakan sebuah modul yang mengubah kode program menjadi kode biner. Sebuah mikrokontroler tidak akan bisa memahami bahasa pemrograman, karena yang dipahami oleh mikrokontroler adalah kode biner.

3. *Uploader*. Merupakan sebuah modul yang digunakan untuk memuat kode biner dari komputer ke dalam memori mikrokontroler di dalam *board* Arduino.



Gambar 2.8 Tampilan Software Arduino IDE

Pada Gambar 2.8 terdapat *menu bar*, *toolbar* berada dibawahnya, dan sebuah area putih untuk *editing sketch*, area hitam dapat kita sebut sebagai *progress area*, dan bagian paling bawah dapat disebut sebagai *status bar*.

Arduino IDE bisa dijalankan di komputer dengan berbagai macam *platform* karena berbasis *java*. *Source program* yang dibuat untuk aplikasi mikrokontroler menggunakan bahasa pemrograman C/C++ dan dapat digabungkan dengan *assembly*. (arduino, n.d.)

Setiap program Arduino (biasa disebut *sketch*) mempunyai dua buah fungsi yang harus ada, yaitu:

- a) `void setup() { }`

Semua kode didalam kurung kurawal akan dijalankan hanya satu kali ketika program Arduino dijalankan untuk pertama kalinya.

b) `void loop() { }`

Fungsi ini akan dijalankan setelah *setup* (fungsi void *setup*) selesai. Fungsi ini akan dijalankan secara terus menerus sampai catu daya (*power*) dilepaskan. (arduino, n.d.)

Serial digunakan untuk komunikasi antara *board* Arduino, komputer atau perangkat lainnya. *Board* Arduino memiliki minimal satu *port* serial yang berkomunikasi melalui pin 0 (RX) dan 1 (TX) dengan komputer melalui USB. Jika menggunakan fungsi – fungsi ini, pin 0 dan 1 tidak dapat digunakan lagi untuk *input* digital ataupun *output* digital (arduino, n.d.). Terdapat beberapa fungsi serial pada Arduino, antara lain:

a) `Serial.begin ()`

Fungsi ini digunakan untuk transmisi data serial dan mengatur *data rate* dalam *bits per second* (baud). Untuk berkomunikasi dengan komputer gunakan salah satu dari angka ini: 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600, atau 115200.

b) `Serial.available ()`

Fungsi ini digunakan untuk mendapatkan jumlah data *byte* (*characters*) yang tersedia dan membacanya dari *port* serial. Data tersebut adalah data yang telah tiba dan disimpan dalam *buffer* serial yang menampung sampai 64 *bytes*.

c) `Serial.read ()`

Fungsi digunakan untuk membaca data serial yang masuk.

d) `Serial.print ()` dan `Serial.println ()`

Fungsi ini digunakan untuk mencetak data ke *port* serial dalam format text ASCII. Sedangkan fungsi `Serial.println()` sama seperti fungsi `Serial.print ()` hanya saja ketika menggunakan fungsi ini akan mencetak data dan kemudian diikuti dengan karakter *newline* atau *enter*.

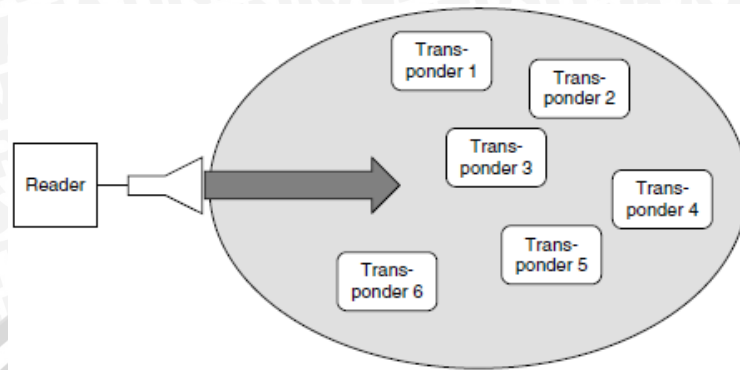
2.2.4 Prosedur *Multi-Access & Anti-Collisions*

Dalam operasi sistem RFID akan ada keadaan dimana beberapa *transponder* berada pada *interrogation zone reader* pada waktu yang bersamaan. Bentuk komunikasi antara *control station*, RFID *reader*, dan sejumlah *transponder* dapat dibagi menjadi dua.

Pada bentuk komunikasi pertama, komunikasi digunakan untuk mentransmisikan data dari *reader* ke *transponder* seperti pada Gambar 2.9. Data yang ditransmisikan oleh *reader* akan diterima secara bersamaan oleh semua

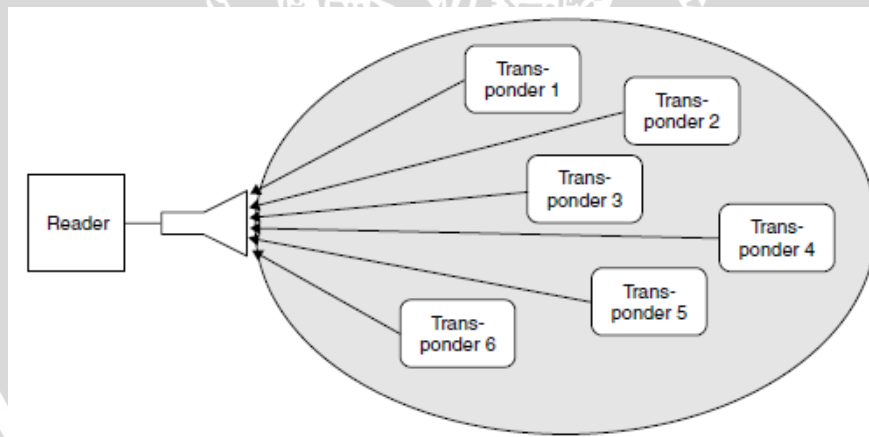
repository.ub.ac.id

transponder yang berada pada *interrogation zone*. Bentuk komunikasi ini disebut *mode broadcast*.



Gambar 2.9 Broadcast Mode : Data yang Ditransmisikan Oleh Reader Akan Diterima Oleh Transponder yang Berada pada Interrogation Zone Secara Bersamaan (Finkenzeller, 2003)

Bentuk komunikasi yang kedua adalah transmisi data dari beberapa *transponder* yang berada pada *interrogation zone* ke *reader*, bentuk komunikasi ini disebut *multi-access mode*. Seperti pada Gambar 2.10.



Gambar 2.10 Multi-Access Mode: Beberapa Transponder Mengirim Data pada Reader Secara Bersamaan (Finkenzeller, 2003)

Setiap *channel* komunikasi memiliki kapasitas *channel* tertentu berupa data *rate* maksimum yang dapat dihandle oleh *channel* dan rentang waktu ketersediannya. Kapasitas *channel* yang tersedia dibagi untuk beberapa *transponder* sebagai jalur pengiriman masing-masing *transponder* ke *reader* tanpa terjadi interferensi (*collisions*).

Sebagai contoh, pada sistem RFID induktif, *reader* berperan sebagai *receiver* data dari semua *transponder* yang berada pada *interrogation zone*. *Reader* akan menyediakan *channel* untuk pengiriman data dari *transponder*. *Data rate* maksimum dapat dihitung dari *bandwidth* efektif dari *antenna* RFID *reader* dan *transponder*.

Permasalahan *multi-access* pada teknologi radio telah lama ada dan diteliti. Beberapa prosedur telah dikembangkan dengan tujuan dapat memisahkan sinyal dari masing-masing *transponder* sehingga tidak saling mempengaruhi satu sama lain. Terdapat empat prosedur dasar yang dapat digunakan, yaitu: *space division multiple access* (SDMA), *frequency domain multiple access* (FDMA), *time domain multiple access* (TDMA) dan *code division multiple access* (CDMA) atau yang biasa dikenal dengan *spread spectrum*. Prosedur dasar tersebut diasumsikan berjalan pada proses pengiriman dan penerimaan data tanpa terjadi *interrupt*. Pembagian *channel* komunikasi akan terus berjalan sampai proses komunikasi berakhir (seperti pada penggunaan komunikasi telepon). Pada *transponder* RFID pembagian *channel* hanya dilakukan sampai kebutuhan identifikasi RFID *tag* selesai, yaitu selama *transponder* berada pada *interrogation zone* *reader*.

Realisasi dari penerapan prosedur *multi-access* pada sistem RFID memiliki banyak tantangan, baik dari sisi *transponder* maupun *reader*. Permasalahan utama adalah kehandalan pencegahan *collisions* antar *transponder* yang akan mengirimkan data pada *receiver*, yaitu RFID *reader*. Dengan adanya *collisions*, bisa jadi akan ada RFID *tag* yang tidak terbaca sehingga dapat menambah pula efektifitas pendeteksian RFID *tag*. Dalam sistem tersebut maka dibutuhkan suatu prosedur teknis berupa algoritma yang dapat memfasilitasi penanganan *multi-access* agar dapat berjalan dengan baik. Algoritma tersebut biasa dikenal dengan algoritma *anti-collisions*.

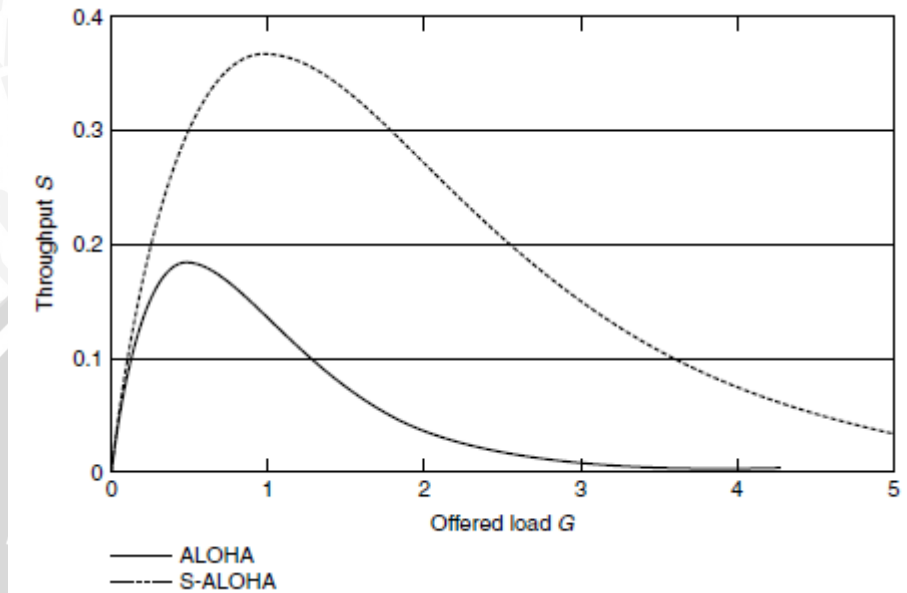
2.2.5 Algoritma Slotted Aloha

Algoritma *Slotted Aloha* merupakan salah satu algoritma yang menyempurnakan algoritma *Aloha* dalam sisi besarnya nilai *throughput*. Pada *Slotted Aloha*, *transponder* hanya diperbolehkan mengirim data ke *reader* pada waktu tertentu (slot). Algoritma ini termasuk dalam kriteria TDMA.

Diasumsikan ukuran paket data yang akan diidentifikasi sama besar, sehingga memiliki durasi transmisi yang sama sebesar τ . Pada algoritma *Aloha*, *collisions* akan terjadi jika dua *transponder* mentransmisikan data pada interval waktu $T \leq 2\tau$. Dengan algoritma *Slotted Aloha*, paket data hanya akan dimulai pada waktu yang sinkron berupa slot waktu sehingga interval *collisions* dapat berkurang menjadi $T = \tau$. Dengan adanya pengurangan interval tersebut, maka nilai *throughput* pada penggunaan algoritma *Slotted Aloha* akan lebih besar dari algoritma *Aloha*. Penghitungan *throughput* dapat dilakukan dengan menggunakan persamaan 2.1 dimana S adalah besarnya *throughput* dan G adalah besar *load*. Pada Algoritma *Slotted Aloha* *throughput* maksimal adalah 36.8% untuk suatu nilai

G, perbandingan besar throughput *Aloha* dan *Slotted Aloha* dapat dilihat pada Gambar 2.11.

$$S = G \cdot e^{-(G)} \quad (2.1)$$

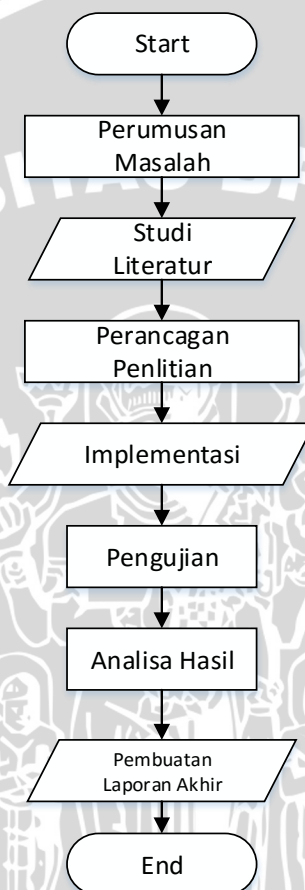


Gambar 2.11 Perbandingan Throughput Algoritma Aloha dan Slotted Aloha (Finkenzeller, 2003)

Algoritma *Slotted Aloha* memiliki tiga *command set* sebagai *anti-collisions*, yaitu: *Request*, *Select*, dan *Read Data*. Jika *transponder* berada pada area *interrogation zone reader*, *transponder* akan mengirimkan *serial number*nya pada *reader*. *Serial number* yang terdeteksi akan diberi nomor slot secara acak. Proses tersebut merupakan bagian dari *Request command*. Selanjutnya, pada *Select command* dilakukan proses pengecekan *serial number* dan slot. Pengecekan dilakukan dengan membandingkan nilai *serial number* dan slot yang dipilih, apakah sama atau sudah dipilih sebelumnya. Pada *Read Data command*, jika tidak terjadi duplikasi dan pemilihan slot yang sama, maka *transponder* akan mengirimkan data identifikasi yang telah disimpan.

BAB 3 METODOLOGI

Bab metodologi membahas alur pelaksanaan penelitian sehingga proses penelitian dapat terarah dengan baik dan sesuai tujuan. Tahap penelitian ditunjukkan dalam bentuk *flowchart* pada Gambar 3.1.



Gambar 3.1 *Flowchart* Pelaksanaan Penelitian

3.1 Perumusan Masalah

Perumusan masalah merupakan tahap menentukan masalah yang dibahas dalam penelitian. Masalah yang dibahas adalah bagaimana mengimplementasikan algoritma *Slotted Aloha* sebagai *anti-collisions* dalam mengidentifikasi *multiple RFID tag* pada *RFID reader*. Hasil penelitian diharapkan dapat digunakan sebagai referensi untuk mengatasi terjadinya *tag collisions* pada proses pembacaan *multiple RFID tag* pada sistem RFID.

3.2 Studi Literatur

Studi literatur dilakukan dengan mengumpulkan dan mempelajari berbagai informasi yang berhubungan dengan RFID *passive system*, algoritma *Slotted Aloha*, Arduino, dan hal-hal yang berhubungan dengan model pemrogramannya.

3.3 Perancangan Penelitian

Perancangan penelitian membahas perencanaan tahap-tahap yang ditempuh dalam mewujudkan penelitian. Tahap-tahap pelaksanaan penelitian yang dilakukan, yaitu: identifikasi kebutuhan (*software* dan *hardware*), dan perancangan proses implementasi

Kebutuhan yang digunakan dalam skripsi ini, yaitu:

1. Kebutuhan *hardware*:
 - Laptop / Komputer PC.
 - Arduino UNO.
 - Modul RFID *reader* Seeedstudio Electronic Brick 125 KHz.
 - 125 KHz RFID *tag*.
2. Kebutuhan *software*:
 - Microsoft Windows 7 Ultimate 32-bit sebagai sistem operasi.
 - Arduino IDE sebagai *Integrate Development Environment*.

Perancangan proses implementasi dilakukan dengan mendefinisikan tahap-tahap yang dilakukan sebelum proses implementasi, yaitu :

1. Perancangan konfigurasi *hardware*

Pada bagian ini dilakukan perencanaan konfigurasi *hardware* yang digunakan. Perencanaan konfigurasi dilakukan dengan menganalisis kebutuhan kinerja *hardware* sehingga *hardware* dapat berjalan sesuai kebutuhan.

2. Perancangan konfigurasi komunikasi *hardware*

Pada bagian ini dilakukan perencanaan konfigurasi komunikasi *hardware* yang akan digunakan. Perencanaan konfigurasi dilakukan dengan menganalisis alur perpindahan data yang dibutuhkan. Dengan dilakukannya perancangan komunikasi *hardware*, diharapkan perpindahan data antar *hardware* dapat berjalan sesuai kebutuhan.

3. Pembuatan program algoritma *Slotted Aloha*

Berdasarkan studi literatur dan analisis kebutuhan *hardware* serta komunikasinya, dilakukan pembuatan program algoritma *Slotted*

Aloha. Program mencakup inisialisasi *hardware*, perpindahan data antar *hardware*, dan algoritma *Slotted Aloha*.

3.4 Implementasi

Dalam tahap ini, dilakukan implementasi berdasarkan studi literatur dan perancangan. Proses implementasi memiliki beberapa tahap, yaitu :

- a. Implementasi dan konfigurasi *hardware*. Pada tahap ini komponen *hardware* diaplikasikan sesuai dengan perancangan. Konfigurasi *hardware* memastikan *hardware* telah terhubung dan dapat berkomunikasi dengan baik. *Hardware* yang digunakan adalah Arduino UNO dan RFID reader 125 KHz.
- b. Implementasi *software*. *Software* pendukung digunakan untuk menunjang jalannya *hardware*. *Software* yang digunakan adalah Arduino IDE sebagai *software* pemrograman Arduino. Arduino diprogram untuk menerima dan memproses data identifikasi RFID tag dari RFID reader. Selain itu Arduino juga diprogram untuk menjalankan algoritma *Slotted Aloha*.

Dari kombinasi implementasi diatas, diharapkan penelitian dapat berjalan sesuai dengan tujuan penelitian, yaitu:

1. RFID reader dapat menerima dan menyampaikan data identifikasi *multiple* RFID tag pada Arduino.
2. Arduino dapat memproses data hasil identifikasi RFID tag dan menerapkan algoritma *Slotted Aloha* sebagai *anti-collisions*.
3. Proses identifikasi RFID tag ditampilkan pada *serial monitor* Arduino.
4. Dapat diketahui proses identifikasi yang dibutuhkan untuk mengidentifikasi *multiple* RFID tag dengan algoritma *Slotted Aloha*.

3.5 Pengujian

Tahap ini membahas pengujian penerapan algoritma *Slotted Aloha* pada Arduino UNO dan RFID reader 125 KHz dan mengevaluasi hasil pengujian. Pengujian dilakukan dengan mengacu pada perancangan penelitian, yaitu:

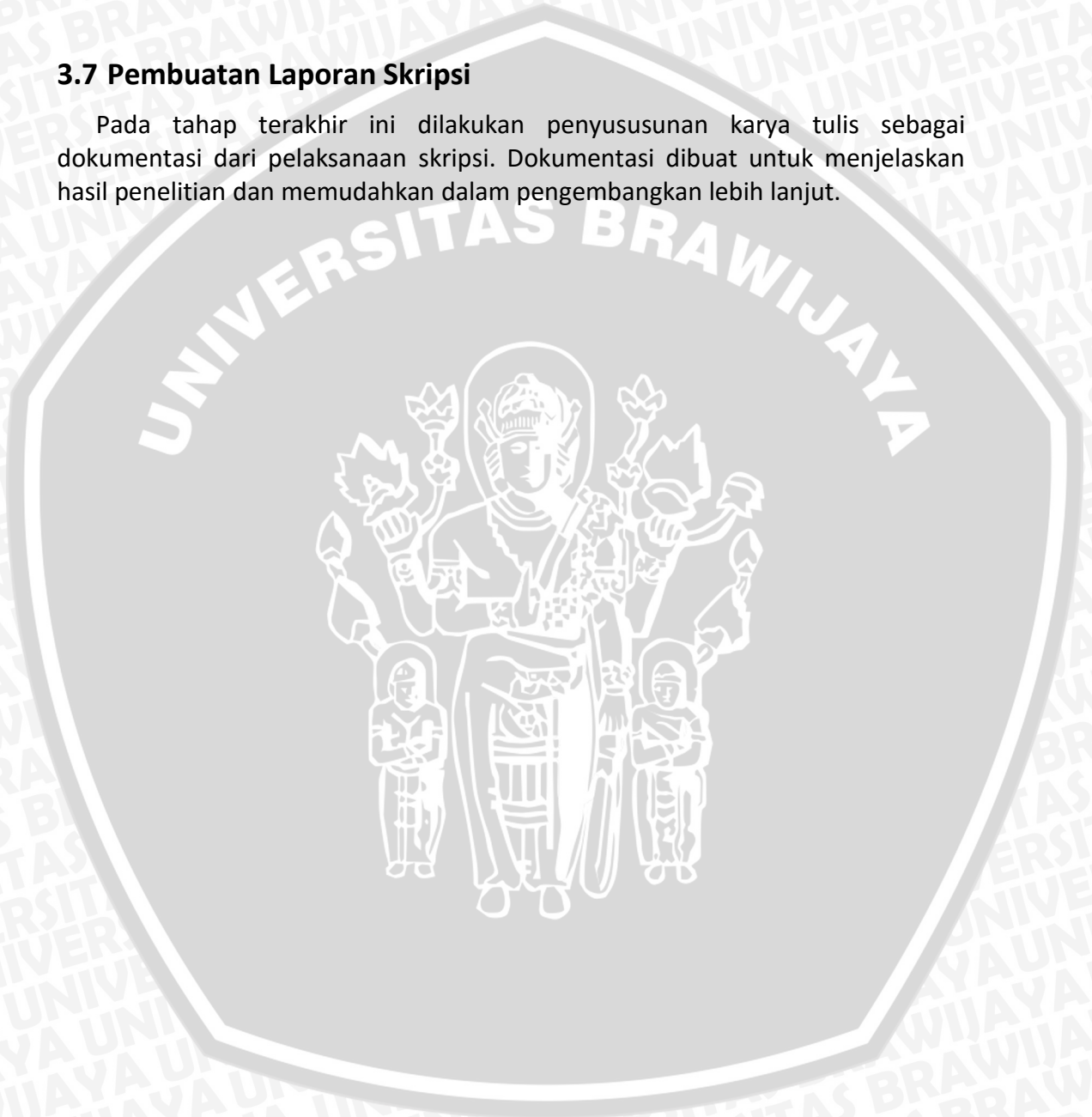
- a. Pengujian *hardware* dan komunikasi data dari RFID reader ke Arduino.
- b. Pengujian identifikasi RFID tag.
- c. Pengujian identifikasi *multiple* RFID tag.
- d. Pengujian identifikasi *multiple* RFID tag dengan algoritma *Slotted Aloha*.

3.6 Analisis dan Pembahasan

Setelah semua tahap penelitian dilakukan dan diuji secara keseluruhan, dilakukan analisis untuk menarik kesimpulan dari penelitian. Hasil pengujian dianalisis untuk menentukan apakah penelitian sudah berjalan baik sesuai dengan yang diharapkan.

3.7 Pembuatan Laporan Skripsi

Pada tahap terakhir ini dilakukan penyusunan karya tulis sebagai dokumentasi dari pelaksanaan skripsi. Dokumentasi dibuat untuk menjelaskan hasil penelitian dan memudahkan dalam mengembangkan lebih lanjut.



BAB 4 PERANCANGAN DAN IMPLEMENTASI

Bab ini berisi perancangan dan implementasi penelitian. Perancangan membahas mengenai sistematika rancangan yang dilakukan untuk penelitian dan implementasi membahas tentang pengimplementasian penelitian untuk tercapainya tujuan penelitian.

4.1 Perancangan

Perancangan penelitian dilakukan untuk merencanakan tahap-tahap yang harus ditempuh dalam melakukan penelitian. Tahap-tahap pelaksanaan penelitian yang harus dilakukan diantaranya identifikasi kebutuhan yang akan digunakan dalam penelitian (berupa *software* maupun *hardware*), dan perancangan proses implementasi. Dengan dilakukannya perancangan, diharapkan dapat mempermudah dalam melakukan langkah-langkah penelitian.

4.1.1 Analisis Kebutuhan

Analisis kebutuhan dalam perancangan bertujuan untuk mendefinisikan kebutuhan-kebutuhan penelitian. *Hardware* dan *software* dianalisis sesuai dengan perannya dalam implementasi penelitian untuk mempermudah implementasi penelitian.

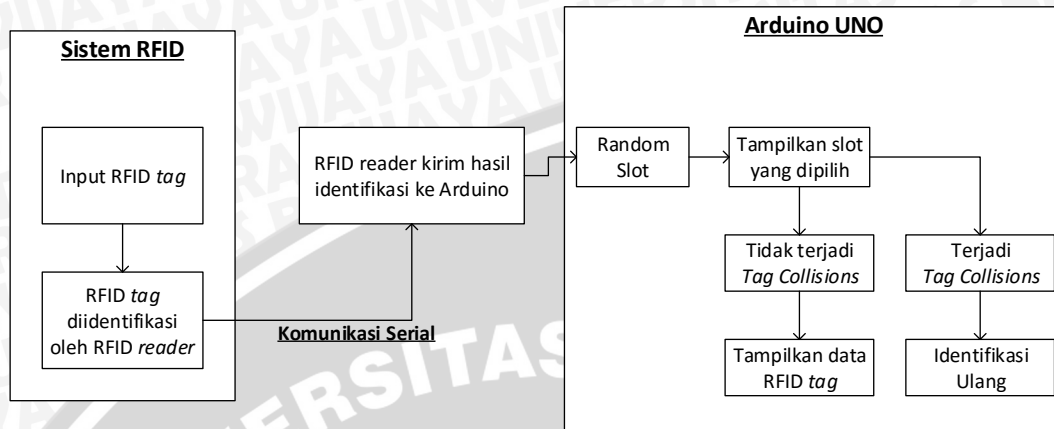
Sistem RFID membutuhkan *reader/interrogator* dan *transponder*. Penelitian ini menggunakan RFID *reader* sebagai *interrogator* dan RFID *card tag* sebagai *transponder*. RFID *reader* dan RFID *tag* yang digunakan, yaitu: RFID *reader* pasif 125 KHz, dan RFID *tag* pasif 125 KHz.

Sebagai pengolah data dan algoritma, dibutuhkan mikrokontroler. Mikrokontroler yang digunakan adalah Arduino UNO. Selain itu, untuk pembuatan program Arduino dibutuhkan pula *software* pemrograman. Pemrograman Arduino dapat dilakukan dengan menggunakan *software* Arduino IDE.

4.1.2 Pendefinisian dan Perancangan Penelitian

Penelitian dilakukan dengan menganalisis algoritma *Slotted Aloha* yang diimplementasikan pada Arduino UNO. Algoritma *Slotted Aloha* digunakan pada sistem RFID sebagai *anti-collisions* untuk mengatasi permasalahan *tag collisions* pada sistem RFID. Algoritma tersebut digunakan untuk pemilihan slot waktu identifikasi RFID *tag* secara berulang hingga semua RFID *tag* teridentifikasi, dan dapat mengatasi *tag collision* pada sistem RFID. Untuk memperjelas perancangan penelitian, telah dirancang diagram blok sebagai kerangka kerja penelitian.

Diagram blok implementasi Algoritma *Slotted Aloha* pada system RFID dapat dilihat pada Gambar 4.1.

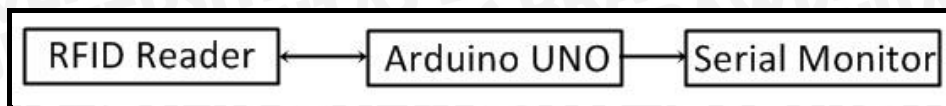


Gambar 4.1 Diagram Blok Ilustrasi Implementasi Algoritma *Slotted Aloha* pada Sistem RFID

Implementasi algoritma *Slotted Aloha* pada sistem RFID dilakukan dengan merangkai seluruh *hardware* yang dibutuhkan agar dapat bekerja dan berkomunikasi sesuai dengan kebutuhan penelitian. *Hardware* yang digunakan adalah RFID reader, RFID tag, dan Arduino UNO. Implementasi algoritma dilakukan pada Arduino UNO. Hasil dari implementasi algoritma *Slotted Aloha* pada *hardware* adalah tampilan ilustrasi pemilihan slot, ilustrasi keadaan *tag collisions*, dan tampilan hasil data identifikasi RFID tag pada serial monitor Arduino.

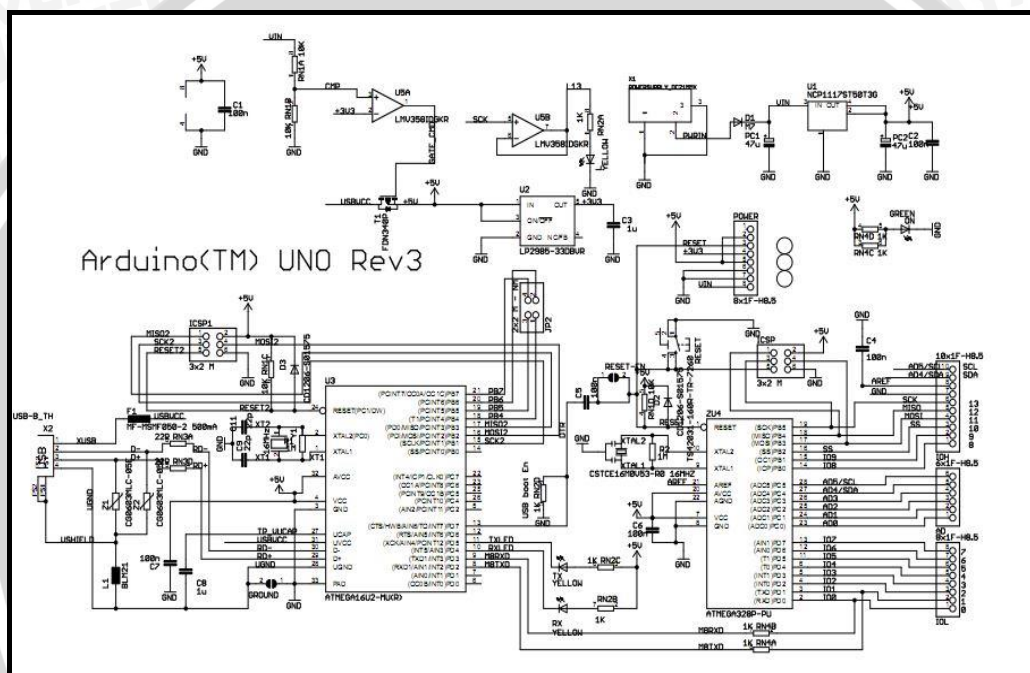
4.1.3 Perancangan *Hardware*

Perancangan *hardware* diawali dengan analisis kebutuhan penelitian yang akan dilakukan. Terdapat tiga *hardware* yang digunakan, yaitu: Arduino UNO, RFID reader 125 KHz, dan RFID tag 125 KHz. Untuk mengidentifikasi RFID tag dibutuhkan RFID reader. Hasil identifikasi yang dilakukan oleh RFID reader kemudian diteruskan ke Arduino. Kemudian, pada Arduino data identifikasi diproses menggunakan algoritma *Slotted Aloha* yang telah diimplementasikan pada Arduino tersebut. Diagram blok perancangan *hardware* dapat dilihat pada Gambar 4.2

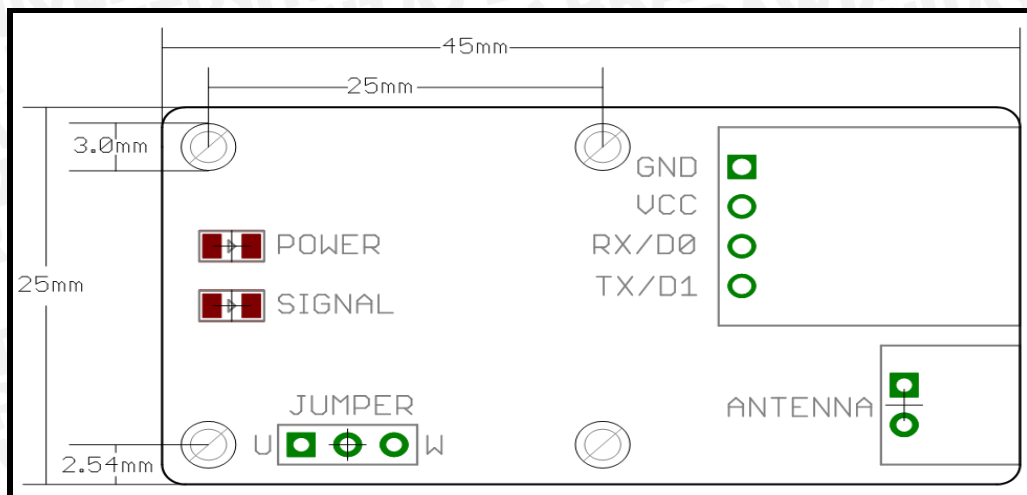


Gambar 4.2 Diagram Blok Perancangan Hardware

Komunikasi antara RFID reader dan Arduino dilakukan secara serial. Pin yang digunakan adalah pin 8 sebagai TX dan pin 9 sebagai RX. Rangkaian skematik dari Arduino UNO dan Blok Diagram modul RFID reader Seedstudio Electronic Brick 125 KHz dapat dilihat pada Gambar 4.3 dan 4.4.



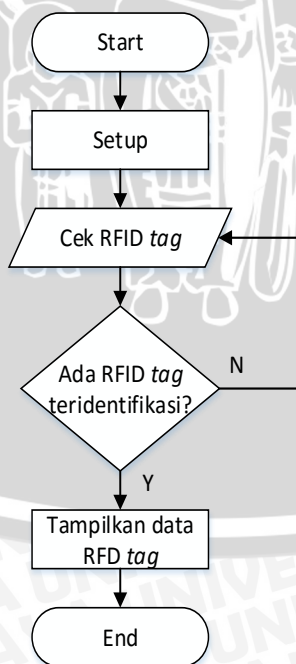
Gambar 4.3 Diagram Skematik Arduino UNO R3. (arduino.cc, n.d.)



Gambar 4.4 Diagram Blok Modul RFID Reader Seedstudio Electronic Brick 125 KHz (SEEDSTUDIO, 2010)

4.1.4 Perancangan Komunikasi Hardware

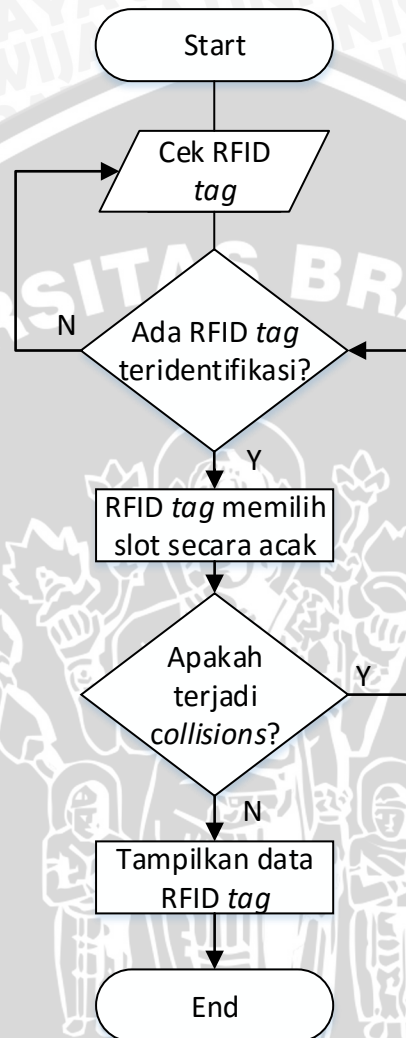
Komunikasi antara Arduino dan RFID reader pada penelitian ini menggunakan komunikasi serial. Dengan menggunakan komunikasi serial, maka komunikasi data dikirimkan melalui kabel serial sebesar satu bit tiap satuan waktu. Maka dari itu, data identifikasi nilai RFID tag akan dikirimkan per bit ke Arduino. Flowchart perancangan komunikasi hardware dapat dilihat pada Gambar 4.5.



Gambar 4.5 Flowchart Perancangan Komunikasi Hardware

4.1.5 Perancangan Program Algoritma *Slotted Aloha* pada *Hardware*

Pada implementasi algoritma *Slotted Aloha*, dilakukan tahap pemrograman *hardware*. *Hardware* yang diprogram untuk mengaplikasikan algoritma *Slotted Aloha* pada penelitian ini adalah Arduino UNO. *Flowchart* perancangan program algoritma *Slotted Aloha* pada Arduino dapat dilihat pada Gambar 4.6.



Gambar 4.6 *Flowchart* Perancangan Algoritma *Slotted Aloha* pada *Hardware*

4.2 Implementasi

Implementasi penelitian membahas mengenai implementasi semua perancangan penelitian yang telah dibuat sebelumnya. Pembahasan pada bagian ini berupa penjelasan langkah penelitian yang dilakukan untuk mengimplementasikan algoritma *Slotted Aloha* sebagai *anti-collisions* pada sistem RFID dan batasan penelitian yang dilakukan.

4.2.1 Spesifikasi Hardware

Perancangan penelitian: “Implementasi Algoritma *Slotted Aloha* Sebagai *Anti-Collisions* dalam Mengidentifikasi *Multiple RFID Tag* pada *RFID Reader*” menggunakan beberapa *hardware*, yaitu: modul *RFID reader* Seedstudio Electronic Brick 125 KHz dan *RFID tag* 125 KHz, serta Arduino UNO. Spesifikasi dari *hardware* tersebut antara lain:

- Perangkat Mikrokontroler

Mikrokontroler yang akan digunakan adalah Arduino UNO. Spesifikasi dari Arduino UNO dapat dilihat pada Tabel 4.1

Tabel 4.1 Spesifikasi Arduino UNO R3 (arduino, n.d.)

<i>Arduino UNO R3</i>	
<i>Microcontroller</i>	ATmega328P
<i>Operating Voltage</i>	5 V
<i>Input Voltage (recommended)</i>	7-12 V
<i>Input Voltage (limit)</i>	6-20V
<i>Digital I/O Pins</i>	14 (of which 6 provide PWM output)
<i>PWM Digital I/O Pins</i>	6
<i>Analog Input Pins</i>	6
<i>DC Current per I/O Pin</i>	20 mA
<i>DC Current for 3.3V Pin</i>	50 mA
<i>Flash Memory</i>	32 KB (ATmega328P) of which 0.5 KB used by bootloader
<i>SRAM</i>	2 KB (ATmega328P)
<i>EEPROM</i>	1 KB (ATmega328P)
<i>Clock Speed</i>	16 MHz
<i>Length</i>	68.6 mm
<i>Width</i>	53.4 mm

- Modul *RFID reader* Seedstudio Electronic Brick 125 KHz dan *RFID tag* 125 KHz

Modul *RFID reader* digunakan untuk mengidentifikasi *RFID card tag*. Spesifikasi dari *RFID reader* Seedstudio Electronic Brick 125 KHz dapat dilihat pada Tabel 4.2

Tabel 4.2 Spesifikasi RFID Reader Seeedstudio Electronic Brick 125 KHz
(SEEDSTUDIO, 2010)

RFID reader Seeedstudio Electronic Brick		
125 KHz		
<i>Pin Name</i>	<i>Function and Note</i>	<i>Rating</i>
<i>GND</i>	<i>Connect to the Host GND</i>	-
<i>VCC</i>	<i>Power Supply</i>	5V
<i>RX / D0</i>	<i>RX – In UART mode, it is unused</i> <i>D0 – In Wiegand mode, it represents Data0</i>	- 5V or 0V
<i>TX / D1</i>	<i>TX – In UART mode, it represents TX data</i> <i>D1 – In Wiegand mode, it represents Data1</i>	5V or 0V 5V or 0V

- RFID tag 125 KHz

RFID tag menyimpan identifikasi unik yang dapat diidentifikasi oleh RFID reader. Spesifikasi dari RFID card tag yang digunakan dapat dilihat pada Tabel 4.3.

Tabel 4.3 Spesifikasi RFID Card Tag 125 KHz

RFID card tag 125 KHz	
<i>Dimensions</i>	86 x 54 x 1.9 mm
<i>Features :</i>	
<ul style="list-style-type: none"> - EM4001 ISO based RFID IC - 125 KHz Carrier - Read Range up to 10 cm - 32-bit unique ID - 64-bit data stream [Header+ID+Data+Parity] 	

- Laptop

Laptop digunakan untuk membuat dan mengupload program ke mikrokontroller. Spesifikasi dari laptop yang digunakan dapat dilihat pada Tabel 4.4.

Tabel 4.4 Spesifikasi Laptop

<i>Laptop Lenovo</i>	
<i>Processors</i>	Intel® Celeron® CPU N2840 @2.16GHz
<i>Memory (RAM)</i>	2.00 GB
<i>Hard Disk</i>	500 GB

4.2.2 Spesifikasi Software

Perancangan penelitian menggunakan beberapa *software* seperti yang ditunjukkan pada Tabel 4.5.

Tabel 4.5 Daftar Spesifikasi Software

<i>Nama Software</i>	<i>Spesifikasi</i>
Sistem Operasi	<i>Windows 7 Ultimate 32-bit</i>
<i>Tools Pemrograman</i>	Arduino IDE
Bahasa Pemrograman	C

4.2.3 Batasan Penelitian

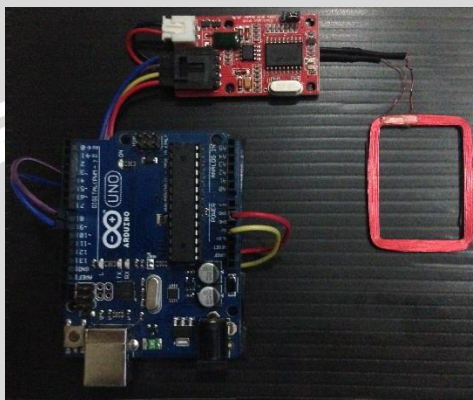
Beberapa batasan dalam penelitian: “Implementasi Algoritma *Slotted Aloha* Sebagai *Anti-Collisions* dalam Mengidentifikasi *Multiple RFID Tag* pada *RFID Reader*” adalah sebagai berikut:

1. *Input* dari implementasi algoritma *Slotted Aloha* pada Arduino dan *RFID reader* adalah hasil identifikasi *RFID reader* terhadap *RFID tag* di sekitarnya.
2. Hasil identifikasi yang telah dilakukan oleh *RFID reader* kemudian dikirimkan ke Arduino melalui komunikasi serial.
3. Algoritma *Slotted Aloha* diterapkan pada Arduino UNO.
4. *Output* dari implementasi algoritma *Slotted Aloha* akan ditampilkan pada *serial monitor*. *Output* yang ditampilkan berupa pemilihan slot dari tiap *RFID tag* yang diidentifikasi, apakah terjadi *tag collisions* atau tidak, dan hasil identifikasi *RFID tag*.
5. Implementasi algoritma *Slotted Aloha* dilakukan untuk mensimulasikan dan menerapkan algoritma *Slotted Aloha* dalam sistem RFID, dan

menganalisis kegunaan algoritma sebagai *anti-collisions* serta efisiensi algoritma dalam proses identifikasi RFID *tag*.

4.2.4 Implementasi *Hardware*

Implementasi *hardware* pada penelitian ini adalah pemasangan *hardware* yang digunakan dalam penelitian. Terdapat dua *hardware* yang digunakan yaitu Arduino UNO sebagai perangkat pengolah data dan algoritma, dan RFID *reader* sebagai pengidentifikasi RFID *tag*. Pengimplementasian *hardware* didokumentasikan seperti pada Gambar 4.7

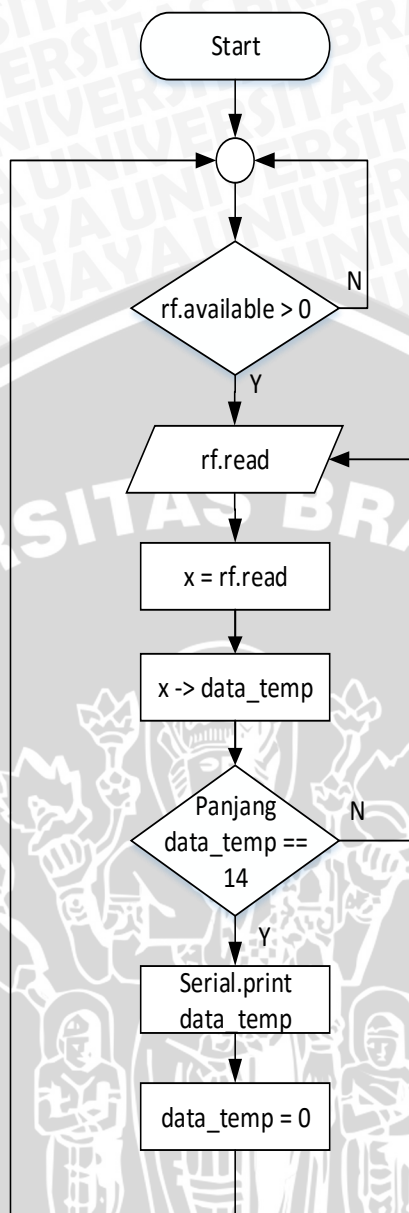


Gambar 4.7 Implementasi Board Arduino UNO dan Modul RFID Reader Seedstudio Electronic Brick 125 KHz

Implementasi *hardware* dilakukan dengan menghubungkan *antenna* RFID *reader* dengan RFID *reader*, kemudian menghubungkan pin TX RFID *reader* pada pin 9 Arduino, pin RX RFID *reader* pada pin 8 Arduino, pin GND pada RFID *reader* pada GND Arduino, dan VCC pada RFID *reader* dengan pin 5V pada Arduino.

4.2.5 Implementasi Komunikasi *Hardware*

Sesuai dengan perancangan, komunikasi *hardware* dilakukan secara serial. Dengan menggunakan komunikasi serial, *board* Arduino akan dihubungkan ke laptop untuk pemrograman dan pengujian. Saat RFID *reader* mendeteksi adanya RFID *tag* disekitarnya, data identifikasi RFID *tag* dikirimkan per bit melalui kabel pin serial Arduino TX dan RX. *Flowchart* dan *source code* komunikasi serial antara RFID *reader* dan Arduino dapat dilihat pada Gambar 4.8 dan 4.9.



Gambar 4.8 Flowchart Komunikasi Serial Antara RFID Reader dan Arduino

```

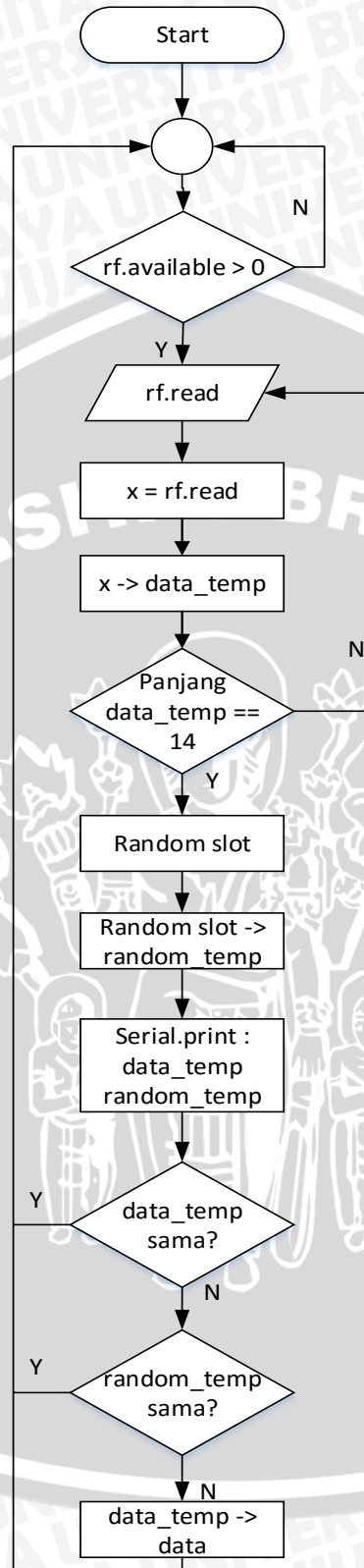
1. #include<SoftwareSerial.h>
2.
3. #define rx 8
4. #define tx 9
5.
6. SoftwareSerial rf(rx,tx);
7.
8. char x;
9. String data_temp;
10.
11. void setup() {
12.     Serial.begin(9600);
13.     Serial.println("Menunggu input RFID Tag.....");
  
```

```
14.
15.     rf.begin(9600);
16. }
17.
18. void loop() {
19.
20.     while(rf.available()>0){
21.         while(data_temp.length()<14){
22.             x=rf.read();
23.             data_temp+=x;
24.         }
25.     }
26.
27.     Serial.print(data_temp);
28.     Serial.println("");
29.     data_temp="";
30. }
31.
```

Gambar 4.9 Source Code Komunikasi Serial Antara RFID Reader dan Arduino

4.2.6 Implementasi Algoritma *Slotted Aloha* pada *Hardware*

Implementasi Algoritma *Slotted Aloha* dilakukan pada Arduino sebagai pengolah hasil identifikasi RFID *tag*. Pengimplementasian algoritma pada *hardware* didokumentasikan berupa *flowchart* dan *source code* seperti pada Gambar 4.10 dan 4.11



Gambar 4.10 Flowchart Implementasi Algoritma Slotted Aloha pada Arduino UNO

```

1. void loop() {
2.     for (i=0; i<2; i++) {
3.         while(data_temp.length()<14){
4.             while(rf.available()>0){
5.                 x=rf.read();
6.                 data_temp+=x;
7.             }
8.         }
9.
10.        random_temp = random(0,2);
11.        Serial.println("");
12.        Serial.println("Hasil Scan : ");
13.        Serial.print(data_temp);
14.        Serial.println("");
15.        Serial.print("Slot : ");
16.        Serial.print(random_temp);
17.        Serial.println("");
18.        if ((String(data_temp)!=String(data0)) &&
19. (String(data_temp)!=String(data1))){
20.            if ((random_temp!=slot[0]) &&
21. (random_temp!=slot[1])){
22.                slot[random_temp] = random_temp;
23.                int panjang_data_temp=data_temp.length();
24.
25.                if (random_temp==0){
26.                    data0=data_temp;
27.                }
28.                else if (random_temp==1){
29.                    data1=data_temp;
30.                }
31.            }
32.            else {
33.                i--;Serial.println("Slot sudah digunakan.
34. Identifikasi ulang..");
35.            }
36.        }
37.        else{
38.            i--;Serial.println("Duplicate ID");
39.        }
40.        data_temp="";
41.
42.    }
43.
44.    Serial.println("");
45.    delay(10);
46.    Serial.println("HASIL IDENTIFIKASI : ");
47.    Serial.println (data0);
48.    delay(10);
49.    Serial.println (data1);
50.
51.    delay(50)
52.    reset();
53. }

```

Gambar 4.11 Source Code Implementasi Algoritma Slotted Aloha pada Arduino UNO

BAB 5 PENGUJIAN DAN ANALISIS

Pada bab ini dilakukan proses pengujian dan analisis dari implementasi yang telah dilakukan. Tujuan dari dilakukannya pengujian adalah untuk mengetahui bahwa semua kebutuhan penelitian telah terpenuhi. Hal-hal yang diuji pada bab ini diantaranya adalah: pengujian *hardware* dan komunikasi *hardware*, pengujian identifikasi RFID *tag*, pengujian identifikasi *multiple* RFID *tag*, dan pengujian identifikasi *multiple* RFID *tag* dengan algoritma *Slotted Aloha*.

Analisis dilakukan untuk mendapatkan kesimpulan dari hasil Implementasi Algoritma *Slotted Aloha* Sebagai *Anti-Collisions* dalam *Mengidentifikasi Multiple* RFID *Tag* pada RFID *Reader*. Proses analisis mengacu pada dasar teori dan sesuai dengan hasil proses pengujian.

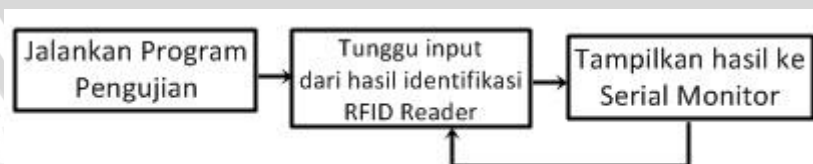
5.1 Pengujian *Hardware* dan Komunikasi *Hardware*

5.1.1 Tujuan Pengujian

Pengujian *hardware* dan komunikasi *hardware* dilakukan untuk mengetahui kinerja *hardware* yang digunakan. Pengujian *hardware* dilakukan untuk mengetahui bahwa semua fungsionalitas *hardware* berjalan dengan baik dan benar.

5.1.2 Prosedur Pengujian

Pengujian *hardware* dilakukan dengan menguji RFID *reader* dan Arduino UNO. Pengujian dilakukan dengan menghubungkan RFID *reader* dengan Arduino. Arduino diprogram untuk menerima hasil identifikasi RFID *tag* dari RFID *reader* dan menampilkan hasil identifikasi pada *serial monitor*. Diagram blok prosedur pengujian dapat dilihat pada Gambar 5.1.



Gambar 5.1 Diagram Blok Pengujian *Hardware* dan Komunikasi *Hardware*

Kebutuhan pengujian :

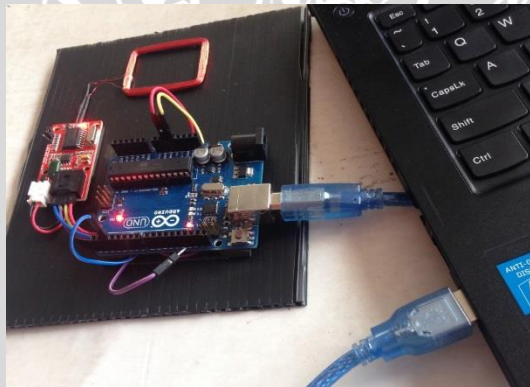
- Laptop.
- *Software* Arduino IDE.
- Arduino UNO.
- RFID reader 125 KHz.
- 2 RFID tag.

5.1.3 Pelaksanaan Pengujian

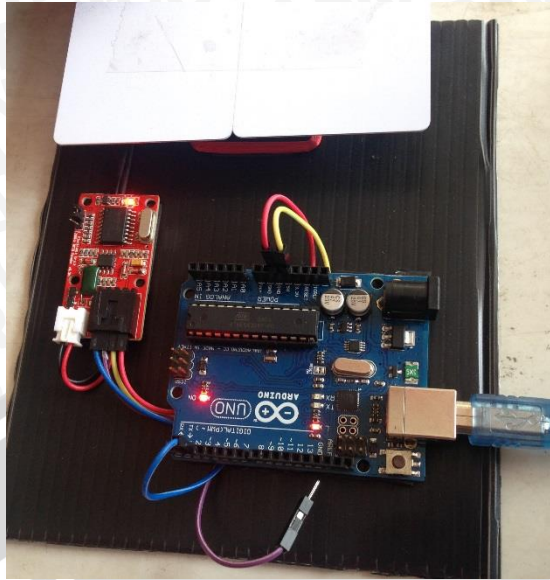
Pengujian *hardware* dan komunikasinya dilakukan dengan menjalankan *hardware* dan program pengujian untuk mengetahui apakah konfigurasi *hardware* sudah berfungsi dengan baik dan benar dan komunikasi serial dapat berjalan dengan baik dan benar. Program pengujian merupakan program sederhana untuk mengidentifikasi RFID tag dan menampilkan hasil identifikasi pada *serial monitor*.

5.1.4 Hasil Pengujian

Hasil pengujian *hardware* dan komunikasinya dapat dilihat pada Gambar 5.2, 5.3, dan 5.4.

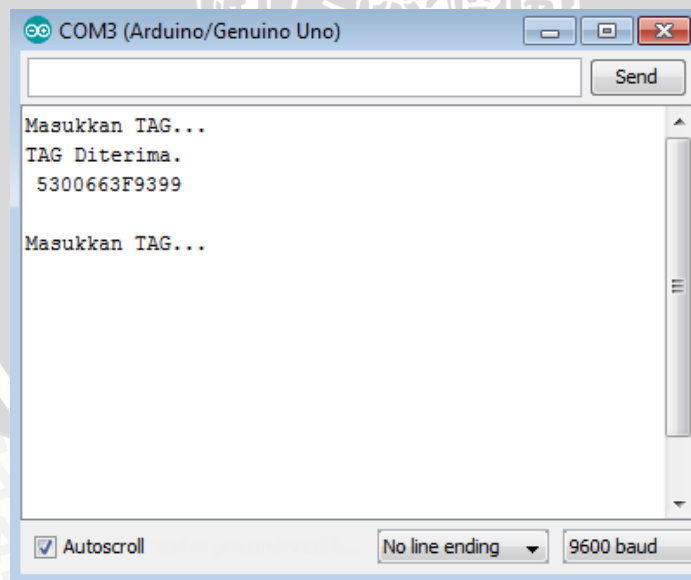


Gambar 5.2 Konfigurasi Pengujian *Hardware*



Gambar 5.3 Salah Satu Langkah Pengujian Komunikasi *Hardware*

Gambar 5.2 merupakan dokumentasi konfigurasi *hardware* yang diuji. RFID *reader* dihubungkan pada Arduino melalui pin 8 dan pin 9 sebagai TX/RX. Arduino dihubungkan pada laptop melalui kabel *USB serial*. Gambar 5.3 merupakan dokumentasi langkah pengujian untuk menguji komunikasi *hardware*. Dari hasil pengujian, diketahui bahwa RFID *reader* dapat mengirimkan hasil identifikasi RFID *tag* pada Arduino dan Arduino dapat mengirimkan hasil identifikasi secara serial pada laptop.



Gambar 5.4 Hasil Pengujian *Hardware* dan Komunikasinya dalam *Serial Monitor*

Gambar 5.3 merupakan dokumentasi hasil pengujian yang telah dilakukan. Berdasarkan hasil pengujian tersebut dapat diketahui bahwa seluruh komunikasi *hardware* dapat dilakukan dengan baik dan benar. Terbukti dengan adanya suksesnya perpindahan data identifikasi RFID *tag* antar *hardware* yang ditunjukkan pada tampilan hasil identifikasi RFID *tag* pada *serial monitor*.

5.2 Pengujian Identifikasi RFID Tag

5.2.1 Tujuan Pengujian

Pengujian identifikasi RFID *tag* dilakukan untuk mengetahui kinerja RFID *tag* yang digunakan. Pengujian identifikasi RFID *tag* dilakukan untuk mengetahui bagaimana proses identifikasi RFID *tag* yang dapat menghasilkan nilai identifikasi yang akurat.

5.2.2 Prosedur Pengujian

Pengujian identifikasi RFID *tag* dilakukan dengan menjalankan program identifikasi RFID *tag*. Pengujian dilakukan untuk mengetahui berapa jarak identifikasi maksimal yang dapat menghasilkan nilai akurat pada proses identifikasi oleh sistem RFID.

Kebutuhan pengujian :

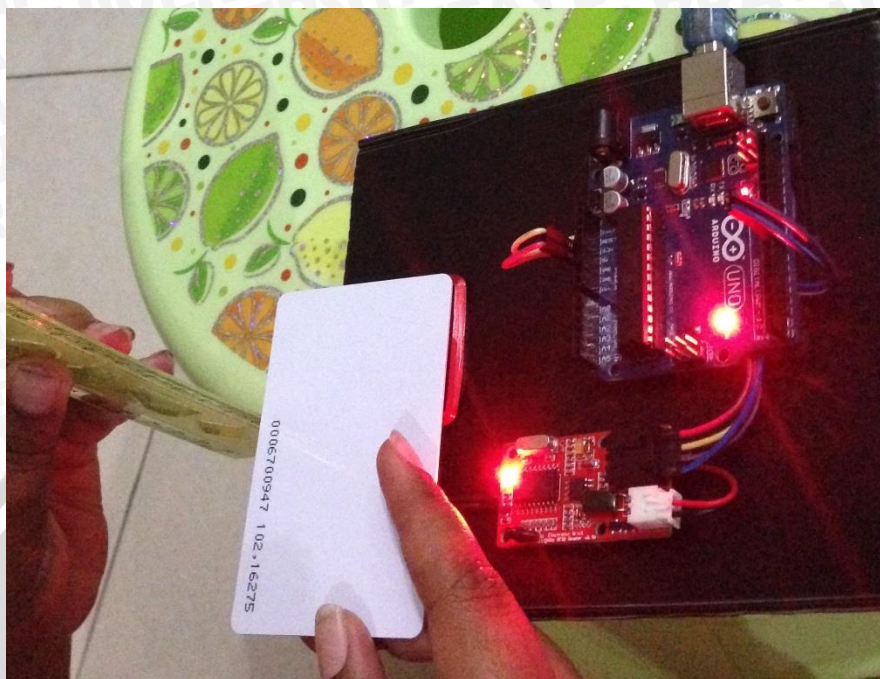
- Laptop.
- *Software* Arduino IDE.
- Arduino UNO.
- RFID *reader* 125 KHz.
- 1 RFID *tag*.

5.2.3 Pelaksanaan Pengujian

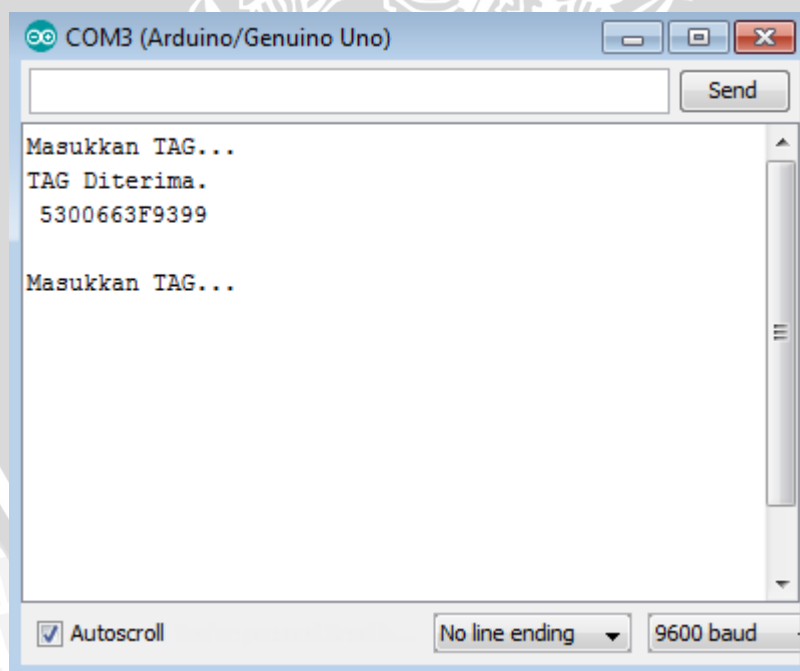
Pengujian dilakukan dengan mendekatkan RFID *tag* pada *antenna* RFID *reader* dalam berbagai jarak. Jarak yang diuji adalah 0-10 cm. Hasil pengujian berupa tabel pengujian jarak identifikasi yang dapat dilakukan oleh sistem RFID. Pengujian identifikasi dilakukan sebanyak dua kali.

5.2.4 Hasil Pengujian

Hasil pengujian Identifikasi RFID *tag* dapat dilihat pada Gambar 5.5, dan 5.6 serta Tabel 5.1.



Gambar 5.5 Proses Pengujian Identifikasi RFID Tag



Gambar 5.6 Hasil Pengujian Identifikasi RFID Tag

Gambar 5.5 merupakan dokumentasi proses pengujian jarak baca maksimal yang dapat dilakukan dalam mengidentifikasi RFID tag. Jarak baca yang diuji adalah 0-10cm. Pengujian jarak baca dilakukan dengan menjauhkan RFID tag dari

RFID *reader* sesuai dengan jarak yang diuji. Gambar 5.6 merupakan dokumentasi hasil identifikasi RFID *tag*. Jika RFID *tag* teridentifikasi pada suatu jarak baca tertentu, maka serial monitor akan menampilkan hasil identifikasi RFID *tag* pada *serial monitor*.

Tabel 5.1 Hasil Pengujian Jarak Baca RFID Reader

No	Jarak (cm)										
	0	1	2	3	4	5	6	7	8	9	10
1	√	√	√	√	√	√	√	√	X	X	X
2	√	√	√	√	√	√	√	√	X	X	X

Tabel 5.1 merupakan hasil pengujian jarak baca yang telah dilakukan. Hasil identifikasi RFID *tag* dapat ditampilkan pada *serial monitor* saat berada pada jarak 0-7 cm dari antenna RFID *reader*. Jika melebihi 7 cm, *serial monitor* tidak dapat menampilkan hasil identifikasi RFID *tag*. Berdasarkan hasil pengujian tersebut dapat disimpulkan bahwa jarak baca maksimal yang dapat dilakukan oleh RFID *reader* adalah 7 cm.

5.3 Pengujian Identifikasi *Multiple* RFID *Tag*

5.3.1 Tujuan Pengujian

Pengujian identifikasi *multiple* RFID *tag* dilakukan untuk mengetahui bagaimana proses identifikasi *multiple* RFID *tag* pada RFID *reader*. Dengan dilakukannya pengujian identifikasi *multiple* RFID *tag* diharapkan dapat diketahui berapa kali proses identifikasi *multiple* RFID *tag* harus dilakukan untuk mendapatkan keseluruhan hasil identifikasi *multiple* RFID *tag*. Pengujian pada bagian ini juga akan menjadi dasar perbandingan pada pengujian selanjutnya.

5.3.2 Prosedur Pengujian

Pengujian identifikasi RFID *tag* dilakukan dengan menjalankan program identifikasi RFID *tag*. Pengujian dilakukan untuk mengetahui berapa kali iterasi yang dibutuhkan dalam proses identifikasi *multiple* RFID *tag* untuk mendapatkan keseluruhan hasil identifikasi *multiple* RFID *tag*.

Kebutuhan pengujian :

- Laptop.
- *Software* Arduino IDE.

- Arduino UNO.
- RFID reader 125 KHz.
- 5 RFID tag.

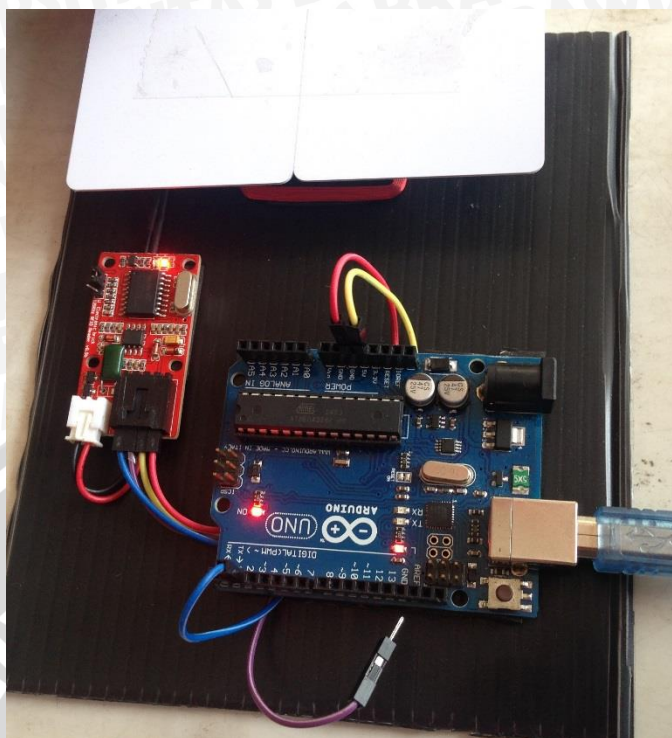
5.3.3 Pelaksanaan Pengujian

Proses pengujian dilakukan beberapa kali dengan menggerakkan RFID tag dua arah diatas *antenna* RFID reader. Jumlah RFID tag yang digunakan dalam pengujian adalah 2, 3, 4, dan 5 RFID tag. Dalam suatu proses identifikasi *multiple* RFID tag dimungkinkan adanya kejadian kegagalan identifikasi RFID tag dalam satu waktu, maka dari itu dibutuhkan iterasi identifikasi sampai semua RFID tag dapat diidentifikasi. Kejadian lain yang mungkin terjadi saat proses identifikasi RFID tag adalah *duplicate ID identification*. *Duplicate ID identification* adalah kejadian duplikasi identifikasi RFID tag dimana RFID tag yang sama merespon sinyal RFID reader dan memberikan nilai identifikasinya lagi pada RFID reader.

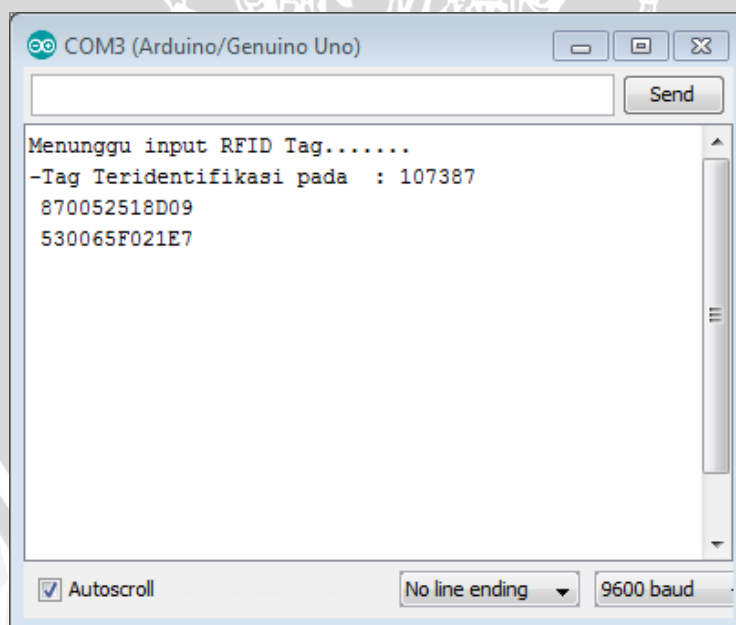
Hasil pengujian ditampilkan dalam tabel yang berisi berapa kali proses iterasi pengujian dilakukan untuk mengidentifikasi *multiple* RFID tag dan berapa kali terjadi *duplicate ID* selama proses identifikasi RFID tag. Banyaknya kejadian *duplicate ID* saat proses identifikasi dapat dilihat pada tampilan *serial monitor* Arduino saat program dijalankan.

5.3.4 Hasil Pengujian

Hasil pengujian Identifikasi *multiple* RFID tag dapat dilihat pada Gambar 5.7, dan 5.8 serta Tabel 5.2, Tabel 5.3, Tabel 5.4, dan Tabel 5.5.



Gambar 5.7 Salah Satu Proses Pengujian Identifikasi *Multiple RFID Tag*



Gambar 5.8 Potongan Hasil Proses Pengujian Identifikasi *Multiple RFID Tag* pada *Serial Monitor*

Gambar 5.7 merupakan dokumentasi proses identifikasi *multiple RFID tag*. Pengujian dilakukan pada beberapa jumlah *RFID tag* yang berbeda, yaitu: 2, 3, 4,

dan 5 RFID tag. Gambar 5.8 merupakan dokumentasi hasil identifikasi *multiple* RFID tag. Jika RFID tag teridentifikasi, maka hasil identifikasinya akan ditampilkan pada *serial monitor*. Apabila RFID tag baru yang teridentifikasi nilainya sama dengan hasil identifikasi RFID tag sebelumnya, maka nilai identifikasi tidak ditampilkan, yang akan ditampilkan adalah notifikasi *duplicate ID identification*.

Tabel 5.2 Hasil Pengujian Identifikasi 2 RFID Tag pada RFID Reader

No	Identifikasi <i>Multiple</i> RFID Tag Berhasil pada Iterasi ke-	Jumlah Kejadian <i>Duplicate ID</i> Saat Proses Identifikasi
1	1	0
2	2	0
3	3	1
4	2	2
5	1	0
6	2	0
7	1	0
8	3	3
9	1	0
10	1	0

Tabel 5.3 Hasil Pengujian Identifikasi 3 RFID Tag pada RFID Reader

No	Identifikasi <i>Multiple</i> RFID Tag Berhasil pada Iterasi ke-	Jumlah Kejadian <i>Duplicate ID</i> Saat Proses Identifikasi
1	5	9
2	1	0
3	1	0
4	2	2
5	2	3
6	4	13
7	4	9
8	2	1
9	1	0
10	2	2

Tabel 5.4 Hasil Pengujian Identifikasi 4 RFID Tag pada RFID Reader

No	Identifikasi <i>Multiple</i> RFID Tag Berhasil pada Iterasi ke-	Jumlah Kejadian <i>Duplicate ID</i> Saat Proses Identifikasi
1	3	7
2	1	2
3	3	5
4	2	2
5	2	2
6	2	3
7	2	2
8	1	0
9	1	0
10	2	4

Tabel 5.5 Hasil Pengujian Identifikasi 5 RFID Tag pada RFID Reader

No	Identifikasi <i>Multiple</i> RFID Tag Berhasil pada Iterasi ke-	Jumlah Kejadian <i>Duplicate ID</i> Saat Proses Identifikasi
1	2	3
2	2	5
3	3	6
4	2	1
5	3	6
6	2	3
7	3	7
8	3	7
9	2	3
10	1	0

Tabel 5.2 sampai dengan 5.5 merupakan hasil pengujian identifikasi *multiple* RFID tag pada sistem RFID. Tabel hasil identifikasi *multiple* RFID tag tersebut didapatkan dari hasil proses identifikasi *multiple* RFID tag pada *serial monitor*. Iterasi identifikasi dihitung dari berapa banyak proses identifikasi yang dibutuhkan untuk mengidentifikasi *multiple* RFID tag, sedangkan *duplicate ID* dihitung dari banyaknya kejadian *duplicate ID identification* selama proses identifikasi berlangsung.

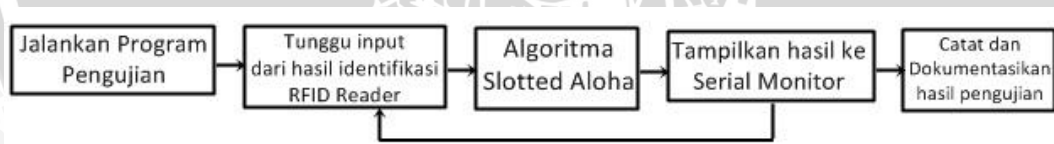
5.4 Pengujian Identifikasi *Multiple RFID Tag* dengan Algoritma *Slotted Aloha*

5.4.1 Tujuan Pengujian

Tujuan dari pengujian algoritma *Slotted Aloha* pada *hardware* adalah untuk mengimplementasikan algoritma *Slotted Aloha* pada *hardware* sistem RFID. Pada pengimplementasian *hardware* di penelitian ini *input*-nya adalah ID *Tag* dan *output* nya adalah hasil identifikasi RFID *tag* dengan menerapkan algoritma *Slotted Aloha* dalam proses identifikasinya. Dengan dilakukannya pengujian algoritma pada *hardware* diharapkan dapat mewakili penelitian untuk mengetahui performa algoritma dalam sistem RFID sesuai dengan kapasitas kemampuan *hardware* yang digunakan.

5.4.2 Prosedur Pengujian

Pengujian identifikasi *multiple RFID tag* dengan algoritma *Slotted Aloha* pada *hardware* dilakukan dengan mengupload program yang telah dibuat pada Arduino IDE ke Arduino UNO. Arduino akan menjalankan program tersebut dan menunggu *input* yang diberikan oleh RFID *reader* secara serial dari hasil identifikasi RFID *tag* yang terbaca oleh RFID *reader*. Jika Arduino telah menerima *input* tersebut maka dilakukan pemrosesan selanjutnya, yaitu penerapan algoritma *Slotted Aloha*. *Output* dari pemrosesan yang dilakukan akan ditampilkan pada *serial monitor* Arduino IDE. Diagram blok prosedur pengujian dapat dilihat pada Gambar 5.9.



Gambar 5.9 Diagram Blok Pengujian Identifikasi *Multiple Tag* dengan Algoritma *Slotted Aloha*

Kebutuhan pengujian :

- Laptop.
- *Software* Arduino IDE.
- Arduino UNO.
- RFID *reader* 125 KHz.
- 5 RFID *tag*.

5.4.3 Pelaksanaan Pengujian

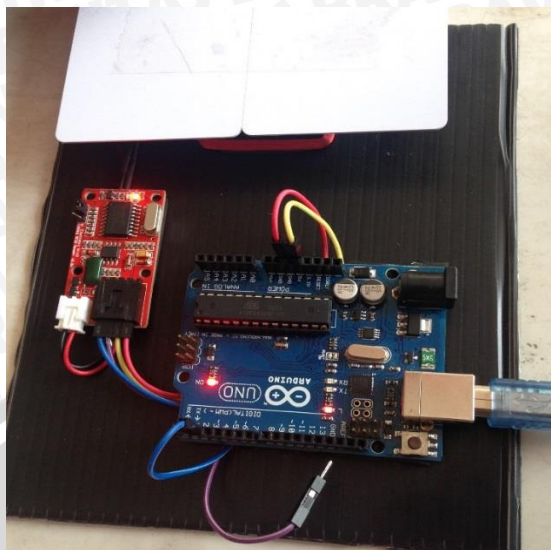
Pengujian dilakukan dengan menggunakan program pengujian yang sama dengan jumlah RFID *tag* 2, 3, 4, dan 5. Pengujian dilakukan dengan menggerakkan RFID *tag* dua arah diatas *antenna* RFID *reader*. Proses pengujian dilakukan beberapa kali.

Dalam suatu proses identifikasi *multiple* RFID *tag* dimungkinkan adanya kejadian kegagalan identifikasi RFID *tag* dalam satu waktu, maka dari itu dibutuhkan iterasi identifikasi sampai semua RFID *tag* dapat diidentifikasi. Kejadian lain yang mungkin terjadi saat proses identifikasi RFID *tag* adalah *duplicate ID identification* dan *tag collisions*. *Duplicate ID identification* adalah kejadian duplikasi identifikasi RFID *tag* dimana RFID *tag* yang sama merespon sinyal RFID *reader* dan memberikan nilai identifikasinya lagi pada RFID *reader*. *Tag collisions* adalah keadaan saat ada beberapa RFID *tag* pada *interrogation zone* RFID *reader* mengirimkan data identifikasinya pada RFID *reader* secara bersamaan. Keadaan ini dapat mengakibatkan kegagalan proses identifikasi. Pada implementasi algoritma *Slotted Aloha* setiap RFID *tag* yang telah diidentifikasi akan memilih slot waktu pengiriman data identifikasi pada RFID *reader*, jika pemilihan slot waktu antar RFID *tag* sama, maka akan terjadi keadaan *tag collisions*.

Hasil pengujian ditampilkan dalam tabel yang berisi berapa kali proses iterasi pengujian dilakukan untuk mengidentifikasi *multiple* RFID *tag*, dan berapa kali terjadi *duplicate ID* serta *tag collisions* selama proses identifikasi RFID *tag*. Banyaknya kejadian *duplicate ID* dan *tag collisions* saat proses identifikasi dapat dilihat pada tampilan *serial monitor* Arduino saat program dijalankan.

5.4.4 Hasil Pengujian

Hasil pengujian Identifikasi *multiple* RFID *tag* dengan algoritma *Slotted Aloha* dapat dilihat pada Gambar 5.10, dan 5.11 serta Tabel 5.6, Tabel 5.7, Tabel 5.8, dan Tabel 5.9.



Gambar 5.10 Salah Satu Proses Pengujian Identifikasi *Multiple RFID Tag* dengan Algoritma *Slotted Aloha*

```

COM3 (Arduino/Genuino Uno)
Menunggu input RFID Tag.....
-----
-Tag Teridentifikasi pada : 4119

Hasil Scan :
51006F03A499
Slot : 1

Hasil Scan :
51006F03A499
Slot : 1
Duplicate ID

Hasil Scan :
5300663F9399
Slot : 1
Slot sudah digunakan. Identifikasi ulang..

Hasil Scan :
5300663F9399
Slot : 0

HASIL IDENTIFIKASI :
5300663F9399
51006F03A499
Waktu Selesai : 6538
Waktu Selisih : 2419

 Autoscroll   No line ending   9600 baud
  
```

Gambar 5.11 Hasil Proses Pengujian Identifikasi *Multiple RFID Tag* dengan Algoritma *Slotted Aloha* pada Serial Monitor

Gambar 5.10 merupakan dokumentasi proses identifikasi *multiple RFID tag* dengan algoritma *Slotted Aloha*. Pengujian dilakukan pada beberapa jumlah RFID tag yang berbeda, yaitu: 2, 3, 4, dan 5 RFID tag. Gambar 5.11 merupakan

dokumentasi hasil identifikasi *multiple RFID tag* dengan algoritma *Slotted Aloha*. Jika *RFID tag* teridentifikasi, maka hasil identifikasinya akan ditampilkan pada *serial monitor*. Apabila *RFID tag* baru yang teridentifikasi nilainya sama dengan hasil identifikasi *RFID tag* sebelumnya, maka nilai identifikasi tidak ditampilkan, yang akan ditampilkan adalah notifikasi *duplicate ID identification*. Apabila *RFID tag* baru yang teridentifikasi nilainya berbeda dengan hasil identifikasi *RFID tag* sebelumnya, namun memilih slot waktu yang sama dengan slot waktu yang dipilih oleh *RFID tag* sebelumnya, maka nilai identifikasi tidak ditampilkan, yang akan ditampilkan adalah notifikasi *tag collisions*.

Tabel 5.6 Hasil Pengujian Identifikasi 2 RFID Tag pada RFID Reader Dengan Algoritma Slotted Aloha

No	Identifikasi <i>Multiple RFID Tag</i> Berhasil pada Iterasi ke-	Jumlah Kejadian <i>Duplicate ID</i> Saat Proses Identifikasi	Jumlah Kejadian <i>Tag Collisions</i> Saat Proses Identifikasi
1	3	1	2
2	3	2	2
3	3	2	2
4	3	1	2
5	4	1	2
6	4	1	2
7	3	1	1
8	3	2	1
9	4	2	1
10	4	2	1

Tabel 5.7 Hasil Pengujian Identifikasi 3 RFID Tag pada RFID Reader Dengan Algoritma Slotted Aloha

No	Identifikasi <i>Multiple RFID Tag</i> Berhasil pada Iterasi ke-	Jumlah Kejadian <i>Duplicate ID</i> Saat Proses Identifikasi	Jumlah Kejadian <i>Tag Collisions</i> Saat Proses Identifikasi
1	9	13	5
2	7	14	4
3	4	7	3
4	4	7	3
5	4	4	4
6	5	10	2
7	9	17	6
8	4	7	1
9	5	7	3
10	6	9	3

Tabel 5.8 Hasil Pengujian Identifikasi 4 RFID Tag pada RFID Reader Dengan Algoritma Slotted Aloha

No	Identifikasi <i>Multiple RFID Tag</i> Berhasil pada Iterasi ke-	Jumlah Kejadian <i>Duplicate ID</i> Saat Proses Identifikasi	Jumlah Kejadian <i>Tag Collisions</i> Saat Proses Identifikasi
1	13	20	8
2	4	0	1
3	9	14	1
4	11	25	4
5	15	21	4
6	5	7	2
7	11	15	10
8	4	1	1
9	18	29	12
10	9	13	3

Tabel 5.9 Hasil Pengujian Identifikasi 5 RFID Tag pada RFID Reader Dengan Algoritma Slotted Aloha

No	Identifikasi <i>Multiple RFID Tag</i> Berhasil pada Iterasi ke-	Jumlah Kejadian <i>Duplicate ID</i> Saat Proses Identifikasi	Jumlah Kejadian <i>Tag Collisions</i> Saat Proses Identifikasi
1	8	33	10
2	11	27	8
3	18	54	11
4	15	40	6
5	12	30	5
6	11	30	5
7	16	41	12
8	22	37	9
9	13	16	8
10	12	21	5

Tabel 5.6 sampai dengan 5.9 merupakan hasil pengujian identifikasi *multiple RFID tag* dengan algoritma *Slotted Aloha* pada sistem RFID. Tabel hasil identifikasi *multiple RFID tag* dengan algoritma *Slotted Aloha* tersebut didapatkan dari hasil proses identifikasi *multiple RFID tag* pada *serial monitor*. Iterasi identifikasi dihitung dari berapa banyak proses identifikasi yang dibutuhkan untuk mengidentifikasi *multiple RFID tag*, sedangkan *duplicate ID* dan *tag collisions* dihitung dari banyaknya kejadian *duplicate ID identification* dan *tag collisions* selama proses identifikasi berlangsung.

5.5 Analisis Hasil Pengujian

5.5.1 Analisis Hasil Pengujian *Hardware* dan Komunikasi *Hardware*

Berdasarkan pengujian yang telah dilakukan pada *hardware*, diketahui bahwa *hardware* telah berfungsi dengan baik dan benar. RFID *tag* yang didekatkan pada *antenna* RFID *reader* dapat diidentifikasi oleh RFID *reader*. Namun ada beberapa hal yang perlu diperhatikan masalah keterbatasan *hardware* yang digunakan, diantaranya: RFID *reader* hanya dapat membaca RFID *tag* dengan jarak tertentu dalam posisi tegak lurus dengan *antenna*, dan dalam proses identifikasi RFID *tag* jika RFID *tag* tidak terbaca (dapat dilihat pada lampu indikator pada RFID *reader*) coba gerakkan RFID *tag* (namun tetap perhatikan jarak baca dari *antenna*). Hasil pengujian komunikasi serial berjalan sesuai dengan tujuan, seperti tampak pada Gambar 5.12 dimana data identifikasi RFID *tag* dapat disampaikan ke Arduino UNO dan dapat ditampilkan pada *serial monitor*.

5.5.2 Analisis Hasil Pengujian Identifikasi RFID *Tag*

Berdasarkan pengujian yang telah dilakukan, Identifikasi RFID *tag* dengan *hardware* *Passive* RFID 125 KHz dapat dilakukan dengan jarak baca maksimal 7 cm tegak lurus dengan *antenna*. Jarak baca RFID *reader* sangat berpengaruh kepada *gain antenna*, ukuran *antenna*, dan jenis RFID *tag*. Untuk memperbesar jarak baca dapat dilakukan dengan memperbesar *gain* dan ukuran *antenna*, dan menggunakan jenis RFID *tag* aktif yang memiliki catu daya sendiri.

5.5.3 Analisis Hasil Pengujian Identifikasi *Multiple* RFID *Tag*

Pengujian identifikasi *multiple* RFID *tag* dilakukan untuk mengetahui bagaimana proses identifikasi *multiple* RFID *tag* tanpa menggunakan algoritma *anti-collisions*. Pada pengujian, jumlah RFID *tag* yang digerakkan dua arah diatas *antenna* RFID *reader* adalah sebanyak 2, 3, 4, dan 5 RFID *tag*. Tujuan pengujian dalam berbagai jumlah RFID *tag* adalah untuk mengetahui berapa kali proses iterasi identifikasi dibutuhkan untuk mengidentifikasi *multiple* RFID *tag* pada RFID *reader*. Dari hasil pengujian dapat disimpulkan bahwa jumlah RFID *tag* dan banyaknya kejadian *duplicate ID identification* berpengaruh pada iterasi yang dibutuhkan untuk mengidentifikasi *multiple* RFID *tag*. Seperti pada yang ditunjukkan pada Tabel 5.2, Tabel 5.3, Tabel 5.4, dan Tabel 5.5 bahwa semakin banyak jumlah RFID *tag* dan jumlah kejadian *duplicate ID identification*, maka dibutuhkan iterasi identifikasi yang semakin banyak untuk mengidentifikasi *multiple* RFID *tag*.

5.5.4 Analisis Hasil Pengujian Identifikasi Multiple RFID Tag dengan Algoritma Slotted Aloha

Pengujian identifikasi *multiple* RFID tag dengan algoritma *Slotted Aloha* dilakukan untuk mengetahui proses identifikasi *multiple* RFID tag dengan algoritma *Slotted Aloha* dan performanya.

Berdasarkan hasil pengujian, dengan menggunakan algoritma *Slotted Aloha* proses identifikasi *multiple* RFID tag dapat dilakukan. Dengan algoritma ini, keadaan *tag collisions* dapat diatasi sehingga pada akhirnya semua RFID tag dapat teridentifikasi. Pada proses identifikasi manual pada pengujian 5.3, dibutuhkan *serial monitor* untuk mengetahui apakah kedua RFID tag telah teridentifikasi atau belum. Algoritma *Slotted Aloha* menawarkan solusi *reliability* sehingga tidak diperlukan *monitoring* secara terus menerus untuk mengetahui apakah semua RFID tag telah terbaca atau belum.

Namun, identifikasi *multiple* RFID tag dengan menggunakan algoritma *Slotted Aloha* membutuhkan proses identifikasi yang lebih lama. Hal ini dikarenakan dibutuhkannya proses identifikasi RFID tag ulang jika keadaan *tag collisions* dan *duplicate ID identification* terjadi.

Sebagai *anti-collisions*, algoritma ini telah dapat mengatasi permasalahan *tag collision*. Jika terjadi *tag collisions* saat RFID tag memilih slot yang sama, maka dibutuhkan proses identifikasi ulang hingga RFID tag dapat seluruhnya teridentifikasi dan memilih slot transmisi yang berbeda. Karena algoritma ini termasuk algoritma probabilistik, maka tujuan dari algoritma ini adalah mendapatkan hasil identifikasi yang seakurat mungkin, meskipun proses identifikasi yang dilakukan lebih lama. Tolak ukur akurasi yang dinilai pada penelitian ini adalah apakah seluruh RFID tag dapat teridentifikasi sesuai dengan jumlahnya. Dengan diterapkannya algoritma ini pada sistem RFID seluruh nilai identifikasi RFID tag yang berada pada *interrogation area* RFID reader dapat diidentifikasi secara akurat namun dibutuhkan iterasi proses identifikasi yang lebih banyak. Banyaknya proses iterasi identifikasi dipengaruhi oleh banyaknya RFID tag yang diidentifikasi, kejadian *duplicate ID identification* dan *tag collisions*. Semakin banyak jumlah RFID tag dan dua keadaan tersebut terjadi, maka akan semakin banyak pula proses iterasi yang dibutuhkan untuk mengidentifikasi *multiple* RFID tag.

Untuk mengidentifikasi performa algoritma *Slotted Aloha* pada sistem RFID, digunakan persamaan 5.1. Pada penelitian ini, hanya digunakan nilai **Rident** dan **Rcoll** untuk menghitung total iterasi yang dibutuhkan dalam mengidentifikasi RFID tag, karena semua slot dianggap tidak ada yang mengalami keadaan *idle*. Hasil analisis efisiensi sistem RFID terhadap penggunaan algoritma *Slotted Aloha* dapat dilihat pada Tabel 5.10, Tabel 5.11, Tabel 5.12, Tabel 5.13, dan Tabel 5.14.

$$SE = \frac{R_{ident}}{R_{tot}} = \frac{R_{idle} + R_{ident} + R_{coll}}{R_{idle} + R_{ident} + R_{coll}} \quad (5.1)$$

(Engels, 2001)

Keterangan :

SE = Efisiensi sistem

R_{ident} = Jumlah RFID tag yang diidentifikasi

R_{tot} = Total iterasi yang dibutuhkan untuk mengidentifikasi RFID tag

R_{idle} = Total kejadian *idle* slot

R_{coll} = Total kejadian tag collisions

Tabel 5.10 Analisis Efisiensi Sistem Saat Proses Identifikasi 2 RFID Tag pada RFID Reader dengan Algoritma Slotted Aloha

No	Identifikasi <i>Multiple</i> RFID Tag Berhasil pada Iterasi ke-	SE
1	3	0.67
2	3	0.67
3	3	0.67
4	3	0.67
5	4	0.5
6	4	0.5
7	3	0.67
8	3	0.67
9	4	0.5
10	4	0.5
Rata-Rata		0.6

Tabel 5.11 Analisis Efisiensi Sistem Saat Proses Identifikasi 3 RFID Tag pada RFID Reader dengan Algoritma Slotted Aloha

No	Identifikasi <i>Multiple</i> RFID Tag Berhasil pada Iterasi ke-	SE
1	9	0.33
2	7	0.43
3	4	0.75
4	4	0.75
5	4	0.75
6	5	0.6
7	9	0.33

No	Identifikasi <i>Multiple</i> RFID <i>Tag</i> Berhasil pada Iterasi ke-	SE
8	4	0.75
9	5	0.6
10	6	0.5
Rata-Rata		0.58

Tabel 5.12 Analisis Efisiensi Sistem Saat Proses Identifikasi 4 RFID *Tag* pada RFID Reader dengan Algoritma *Slotted Aloha*

No	Identifikasi <i>Multiple</i> RFID <i>Tag</i> Berhasil pada Iterasi ke-	SE
1	13	0.31
2	4	1
3	9	0.44
4	11	0.36
5	15	0.27
6	5	0.8
7	11	0.36
8	4	1
9	18	0.22
10	9	0.44
Rata-Rata		0.52

Tabel 5.13 Analisis Efisiensi Sistem Saat Proses Identifikasi 5 RFID *Tag* pada RFID Reader dengan Algoritma *Slotted Aloha*

No	Identifikasi <i>Multiple</i> RFID <i>Tag</i> Berhasil pada Iterasi ke-	SE
1	8	0.63
2	11	0.45
3	18	0.28
4	15	0.33
5	12	0.42
6	11	0.45
7	16	0.31
8	22	0.23
9	13	0.38
10	12	0.42
Rata-Rata		0.39

Berdasarkan hasil penghitungan SE diatas, dapat disimpulkan bahwa besar nilai efisiensi sistem RFID dengan menggunakan algoritma *Slotted Aloha* dipengaruhi oleh banyaknya jumlah RFID *tag* yang diidentifikasi dan banyaknya proses iterasi yang dibutuhkan untuk mengidentifikasi *multiple* RFID *tag*. Semakin banyak jumlah RFID *tag* yang akan diidentifikasi, maka nilai efisiensi sistem akan semakin berkurang.



BAB 6 PENUTUP

Bagian ini memuat kesimpulan dan saran penulis terhadap penelitian yang telah dilakukan.

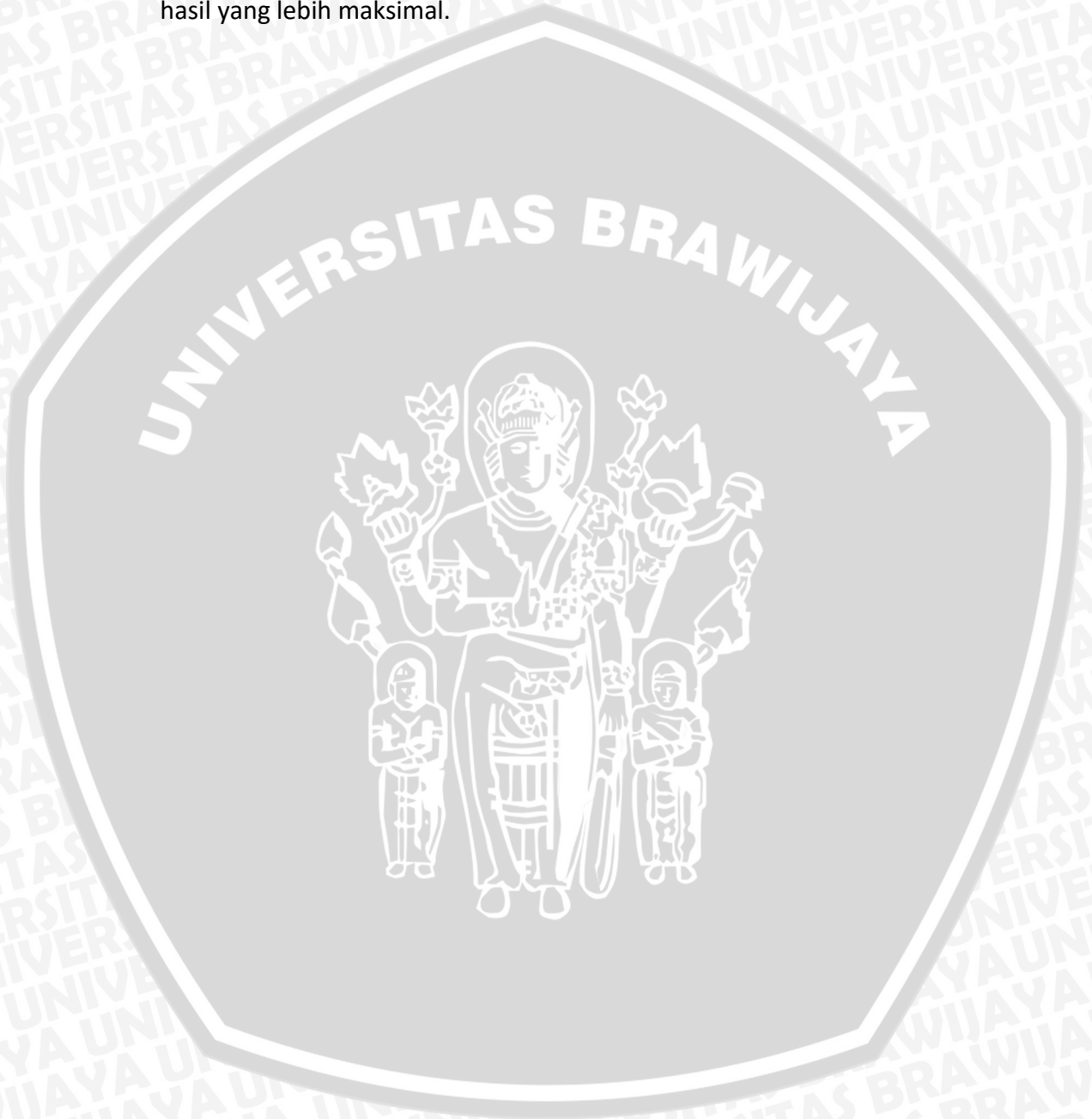
6.1 Kesimpulan

Berdasarkan perancangan, implementasi, dan analisis pengujian yang telah dilakukan maka dapat diambil kesimpulan sebagai berikut:

1. Untuk mengidentifikasi *multiple RFID tag* dibutuhkan algoritma *anti-collisions* agar hasil identifikasi yang didapatkan akurat. Tolak ukur akurasi pada penelitian ini adalah apakah seluruh *RFID tag* dapat teridentifikasi sesuai dengan jumlahnya.
2. Algoritma *Slotted Aloha* dapat diimplementasikan pada Arduino dan sistem RFID dengan baik. Simulasi dapat menampilkan keadaan *collisions* dan dapat mengatasi keadaan tersebut. Dengan diterapkannya algoritma tersebut sebagai *anti-collisions* pembacaan *multiple RFID tag* yang dilakukan bersamaan dapat diidentifikasi dengan baik dan benar.
3. Hasil analisis pada implementasi algoritma *Slotted Aloha* pada *hardware* menyatakan bahwa iterasi identifikasi yang dibutuhkan untuk mengidentifikasi *RFID tag* dengan menggunakan algoritma *Slotted Aloha* masih cukup banyak. Banyaknya iterasi identifikasi dipengaruhi berapa banyak *RFID tag* yang diidentifikasi dan berapa banyak *tag collisions* serta *duplicate ID identification* yang terjadi. Karena algoritma *Slotted Aloha* merupakan algoritma probabilistik, maka hasil identifikasi yang diperoleh merupakan hasil identifikasi yang akurat, meskipun waktu identifikasi yang dilakukan lebih lama. Efisiensi implementasi algoritma *Slotted Aloha* pada sistem RFID dipengaruhi oleh berapa banyak jumlah *RFID tag* yang diidentifikasi dan berapa banyak keadaan *tag collisions* yang terjadi dalam proses identifikasi. Pada proses identifikasi 2 *RFID tag* nilai rata-rata efisiensinya adalah 0.6, pada proses identifikasi 3 *RFID tag* nilai rata-rata efisiensinya adalah 0.58, pada proses identifikasi 4 *RFID tag* nilai rata-rata efisiensinya adalah 0.52, dan pada proses identifikasi 2 *RFID tag* nilai rata-rata efisiensinya adalah 0.39.

6.2 Saran

1. Waktu identifikasi dapat ditingkatkan dengan menggabungkan algoritma *Slotted Aloha* dengan salah satu algoritma deterministik.
2. Penerapan algoritma *Slotted Aloha* pada sistem RFID nyata dengan menggunakan *hardware* yang performanya memadai, agar didapatkan hasil yang lebih maksimal.



DAFTAR PUSTAKA

arduino.cc, n.d. *ARDUINO*.

[Online]

Available at: <https://www.arduino.cc/en/uploads/Main/Arduino Uno Rev3-schematic.pdf>

[Accessed 10 12 2015].

arduino, n.d. *arduino*.

[Online]

Available at: <https://www.arduino.cc/en/Guide/Environment>

[Accessed 10 12 2015].

arduino, n.d. *arduino.cc*.

[Online]

Available at: <https://www.arduino.cc/en/Main/ArduinoBoardUno>

[Accessed 10 12 2015].

Deegala, K. S., 2006. Performance of Slotted Aloha Anti-Collision Protocol for RFID Systems Under Interfering Environments.

Engels, D., 2001. *The Reader Collision Problem*. s.l.:Auto-ID-Center.

Finkenzeller, K., 2003. *RFID Handbuch, English*. 2nd ed. s.l.:Wiley.

Hasad, A., 2011. *Algoritma Optimasi dan Aplikasinya*.

Namboodiri, V. & Gao, L., 2007. Energy-Aware Tag Anti-Collision Protocols for RFID Systems. *Proceedings of the Fifth IEEE International Conference on Pervasive Computing and Communications*, pp. 23-36.

SEEDSTUDIO, 2010. *SEEDSTUDIO*.

[Online]

Available at: www.seeedstudio.com

[Accessed 10 12 2015].

sparkfun.com, n.d. *sparkfun.com*.

[Online]

Available at: <https://www.sparkfun.com/products/8310>

[Accessed 20 12 2015].