

**PROTOTYPE PENGENDALI PINTU GARASI
MENGUNAKAN SISTEM MINIMUM ARDUINO MEGA 2560**

SKRIPSI

Oleh :

Alfred Christian Boyong

15.43.061



**PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
WIDYA CIPTA DHARMA
SAMARINDA
2019**

**PROTOTYPE PENGENDALI PINTU GARASI
MENGUNAKAN SISTEM MINIMUM ARDUINO MEGA 2560**

SKRIPSI

*Diajukan Sebagai Salah Satu Syarat
Untuk Mendapatkan Gelar Sarjana Komputer*

Oleh :

Alfred Christian Boyong

15.43.061



**PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
WIDYA CIPTA DHARMA
SAMARINDA
2019**

LEMBAR PENGESAHAN

Skripsi Oleh : Alfred Christian Boyong (15.43.061)

Telah dipertahankan di depan Dewan Penguji

Pada tanggal : 22 JUL 2019

Dewan Penguji :

Ahmad Rofiq Hakim, S.Pd., M.Kom




Andi Yusika Rangan, M.Kom



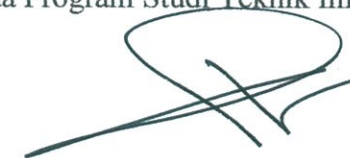
Drs. Azahari, M.Kom



Asep Nurhuda, S.Kom., M.Kom



Mengetahui,
Ketua Program Studi Teknik Informatika



Asep Nurhuda, S.Kom., M.Kom

Mengesahkan,
Ketua STMIK Widya Cipta Dharma



Dr. H. Nursebah, S.Kom., M.Kom

UNIVERSITAS SAMARINDA

JURUSAN TEKNIK INFORMATIKA

FAKULTAS TEKNIK

PROG. JUL 3 S.

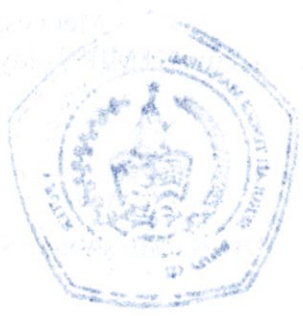
SKRIPSI

PROTOTYPE PENGENDALI PINTU GARASI MENGGUNAKAN SISTEM MINIMUM ARDUINO MEGA 2560

ALFRED CHRISTIAN BOYONG

15.43.061

2019



LEMBAR PERNYATAAN

Saya yang bertanda tangan dibawah ini :

Nama : Alfred Christian Boyong

NIM : 15.43.061

Judul : Prototype Pengendali Pintu Garasi Menggunakan Sistem Minimum Arduino Mega 2560

Menyatakan dan bertanggung jawab dengan sebenarnya bahwa Skripsi ini adalah hasil karya saya sendiri kecuali cuplikan dan ringkasan yang masing-masing telah saya jelaskan sumbernya. Jika waktu selanjutnya ada pihak lain mengkaim bahwa Skripsi ini sebagai karyanya, yang disertai dengan bukti-bukti yang cukup, maka saya siap untuk mendapatkan sanksi akademik yang terkait dengan hal tersebut.

Samarinda, 22 Juli 2019

Yang membuat pernyataan



(Alfred Christian Boyong)

ABSTRAK

Alfred Christian Boyong, 2019, Prototype Pengendali Pintu Garasi Menggunakan Sistem Minimum Arduino Mega 2560. Skripsi Program Studi Teknik Informatika, Sekolah Tinggi Manajemen Informatika dan Komputer Widya Cipta Dharma, Pembimbing(I), Ahmad Rofiq Hakim, S.Pd.,M.Kom., dan Pembimbing (II) Andi Yusika Rangan, M.Kom.

Kata Kunci : Prototype, Garasi, RFID MFRC522, Keypad 4x4, Servo

Keamanan merupakan hal yang penting dalam bidang industri maupun bidang rumah tangga. Seiring dengan perkembangan teknologi yang semakin canggih, khususnya dibidang elektronika analog dan digital. Pada saat yang sama perkembangan teknologi khususnya terhadap ruang penyimpanan kendaraan roda dua dan roda empat seperti garasi juga diperlukan. Prototype pengendali pintu otomatis menggunakan sistem minimum Arduino Mega 2560 ini, menggunakan RFID MFRC522 untuk membaca kartu pengguna menggunakan gelombang radio, Keypad Membrane 4x4 untuk mengetik password dari pengguna dan servo motor untuk berputar sesuai sistem yang berjalan, serta LCD 16x2 untuk menampilkan setiap proses dengan buzzer sebagai bunyi peringatan.

Prototype ini dikembangkan dengan metode pengembangan sistem *Waterfall* (Model Air Terjun) dan alat bantu pengembangan sistem dengan menggunakan Flowchart dan Blok Diagram. *Waterfall* merupakan model yang membangun perangkat lunak berdasarkan struktur yang dimulai dari perencanaan, analisis, desain dan implementasi.

Pada kasus ini maka di rancang pembuatan alat pengendali pintu yang mampu mengendalikan pintu dengan menggunakan *Micro Servo* SG90 dengan RFID MFRC522 sebagai RFID *reader* dan Keypad Membrane Matrix 4x4 sebagai *input password*, serta LCD 16x2 sebagai *output* dan *buzzer* sebagai *output* bunyi.

ABSTRACT

Alfred Christian Boyong, 2019, Garage Door Controller Prototype Using the Arduino Mega 2560 Minimum System. Skripsi Program Informatics Engineering Study, College of Information and Computer Management Widya Cipta Dharma, Mentor (I), Ahmad Rofiq Hakim, S.Pd., M.Kom., and Mentor (II) Andi Yusika Rangan, M.Kom.

Keywords: Prototype, Garage, RFID MFRC522, 4x4 Keypad, Servo

Security is important in the field of industry and in the household sector. Along with the development of increasingly sophisticated technology, especially in the field of analog and digital electronics. At the same time, technological developments especially in the storage space of two-wheeled vehicles and four-wheeled vehicles such as garages are also needed. The automatic door controller prototype uses this Arduino Mega 2560 minimum system, use RFID MFRC522 to read the user's card using radio waves, Keypad Membrane 4x4 for typing passwords from users and servo motors to rotate according to the system running, and 16x2 LCD to display each process with the buzzer as a warning sound.

This prototype was developed with the method of developing a Waterfall system (Model Waterfall) and system development tools using the Flowchart and Block Diagram. Waterfall is a model that builds software based on a structure that starts from planning, analysis, design and implementation.

In this case, it is designed to make a door controller that is able to control the door using Micro Servo SG90 with RFID MFRC522 as RFID reader and Keypad Membrane Matrix 4x4 as input password, and LCD 16x2 as output and buzzer as sound output.

DAFTAR RIWAYAT HIDUP



Alfred Christian Boyong, lahir pada tanggal 9 november 1997 di Sangasanga, Kalimantan Timur. Merupakan anak kedua dari empat besaudara pasangan Bapak Yunus Andrias dan Ibu Maria Dara, Memulai pendidikan Sekolah Dasar pada tahun 2003 di Sekolah Dasar Negeri 013 Sangasanga dan lulus pada tahun 2009. Pada tahun yang sama melanjutkan pendidikan di Sekolah Menengah Pertama Negeri 3 Sangasanga dan lulus pada tahun 2012. Pada tahun yang sama melanjutkan pendidikan di Sekolah Menengah Kejuruan Negeri 1 Sangasanga pada Jurusan Teknik Otomotif dan lulus pada tahun 2015. Kemudian melanjutkan studi pada tahun yang sama di Sekolah Tinggi Manajemen Informatika dan Komputer Widya Cipta Dharma Samarinda pada Program Studi Teknik Informatika Jenjang Strata Satu.

Kegiatan yang pernah dilakukan selama menempuh pendidikan di Perguruan Tinggi adalah melakukan Kegiatan Kuliah Kerja Praktek di Rumah Sakit Umum Daerah Abdul Wahab Sjahranie Samarinda dan melakukan penelitian Skripsi di Sekolah Tinggi Manajemen Informatika dan Komputer Widya Cipta Dharma Samarinda.

KATA PENGANTAR

Puji dan syukur kehadiran Tuhan Yang Maha Esa atas anugerah dan kasih karunia-Nya sehingga penulisan laporan skripsi yang berjudul “**Prototype Pengendali Pintu Garasi Menggunakan Sistem Minimum Arduino Mega 2560**”.

Pada kesempatan ini pula, penulis mengucapkan terima kasih dan penghargaan yang setinggi – tingginya kepada :

1. Bapak Dr. H. Nursobah, S. Kom, M. Kom, selaku Ketua Sekolah Tinggi Manajemen Informatika dan Komputer Widya Cipta Dharma Samarinda.
2. Bapak Asep Nurhuda, M. Kom selaku Ketua Program Studi Teknik Informatika Sekolah Tinggi Manajemen Informatika dan Komputer Widya Cipta Dharma Samarinda.
3. Bapak Ahmad Rofiq Hakim, S. Pd., M. Kom selaku Dosen Pembimbing I atas segala ilmu, arahan, bimbingan dan koreksi yang diberikan pada penulis.
4. Bapak Andi Yusika Rangan, M. Kom selaku Dosen Pembimbing II atas segala ilmu, arahan, bimbingan dan koreksi yang diberikan pada penulis.
5. Bapak Drs. Azahari, M. Kom selaku Dosen Penguji I yang telah memberikan sebagian waktunya untuk menguji dan membahas laporan skripsi ini sehingga dapat terselesaikan dengan baik.

6. Bapak Asep Nurhuda, M. Kom selaku Dosen Penguji II yang telah memberikan sebagian waktunya untuk menguji dan membahas laporan skripsi ini sehingga dapat terselesaikan dengan baik.
7. Seluruh Dosen dan Staff Pengajar STMIK Widya Cipta Dharma dalam memberikan ilmu pengetahuan serta bimbingan yang bermanfaat.
8. Rekan – rekan Teknik Informatika Angkatan 2015 khususnya TI pagi B yang telah memberikan dukungan serta semangat kepada penulis dalam proses penulisan laporan skripsi ini.
9. Seluruh pihak yang secara langsung maupun tidak langsung yang telah membantu penulis dalam menyelesaikan penulisan skripsi ini, yang tidak dapat disebutkan satu persatu.

Penulis menyadari bahwa dalam hasil penulisan skripsi ini masih banyak kelemahan dan kekurangan, tetapi penulis berharap semoga apa yang telah tertulis dalam hasil penelitian ini bermanfaat bagi para pembaca khususnya mahasiswa STMIK Widya Cipta Dharma. Semoga bantuan yang diberikan oleh semua pihak mendapat imbalan dari Tuhan Yang Maha Esa.

Samarinda, 22 Juli 2019

Penulis

Alfred Christian Boyong

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN	ii
LEMBAR PERNYATAAN	iii
ABSTRAK	iv
ABSTRACT	v
DAFTAR RIWAYAT HIDUP	vi
KATA PENGANTAR	vii
DAFTAR ISI	ix
DAFTAR TABEL	xii
DAFTAR GAMBAR	xiii
DAFTAR LAMPIRAN	xiv

BAB I PENDAHULUAN

1.1	Latar Belakang	1
1.2	Rumusan Masalah	2
1.3	Batasan Masalah	2
1.4	Tujuan Penelitian	2
1.5	Manfaat Penelitian	2
1.6	Sistematika Penulisan	3

BAB II TINJAUAN PUSTAKA

2.1	Kajian Empirik	5
2.2	Kajian Teoritik	7
2.2.1	<i>Hardware</i>	7
2.2.2	Garasi	8
2.2.3	Arduino Mega	8
2.2.3.1	Sistem Minimum Mikrokontroler	11
2.2.3.2	Bahasa Pemrograman Arduino.....	11
2.2.3.3	Arduino IDE	16
2.2.4	<i>Matrix Membrane Keypad 4x4</i>	18
2.2.5	RFID	19
2.2.5.1	RFID <i>Tag</i>	19
2.2.5.2	RFID <i>Reader</i>	21
2.2.5.3	Frekuensi Kerja RFID	21
2.2.6	LCD (<i>Liquid Crystal Display</i>)	23
2.2.7	Motor Servo	24
2.2.8	Breadboard	25

2.2.9	<i>Power Supply</i>	25
2.2.10	<i>Kabel Jumper</i>	26
2.2.11	Buzzer	26
2.2.12	Fritzing	27
2.2.13	Metode Pengembangan Sistem	29
2.2.13.1	<i>Waterfall</i>	29
2.2.14	Alat Bantu Pengembangan Sistem	31
2.2.14.1	<i>Flowchart</i>	32
2.2.14.2	Blok Diagram	34
2.2.15	Metode Pengujian	34
2.2.15.1	Pengujian <i>Black Box</i>	35
2.2.15.2	Pengujian <i>White Box</i>	35

BAB III METODE PENELITIAN

3.1	Tempat dan Waktu Penelitian	37
3.2	Teknik Pengumpulan Data	37
3.2.1	Observasi Langsung	37
3.2.2	Studi Pustaka	38
3.3	Metode Pengembangan	38
3.3.1	Analisis	38
3.3.1.1	Analisis Data	38
3.3.1.2	Analisis <i>User</i>	39
3.3.1.3	Analisis Sistem	39
3.3.1.4	Analisis Teknologi	39
3.3.2	Desain (Perancangan)	39
3.3.3	Implementasi	39
3.3.4	Pengujian (<i>Testing</i>)	40
3.3.5	<i>Maintenance</i>	40

BAB IV HASIL PENELITIAN DAN PEMBAHASAN

4.1	Hasil Penelitian	42
4.1.1	Gambaran Umum	42
4.2	Pembahasan	42
4.2.1	Analisis	42
4.2.1.1	Analisis Data	42
4.2.1.2	Analisis <i>User</i>	43
4.2.1.3	Analisis Sistem	43
4.2.1.4	Analisis Teknologi	45
4.2.2	Desain	45
4.2.2.1	Desain Perangkat Keras	46
4.2.2.2	Desain Perangkat Lunak	47
4.2.3	Implementasi	49

4.2.3.1	Perangkat Keras	49
4.2.3.2	Perangkat Lunak	55
4.2.4	<i>Testing</i>	56
4.2.4.1	Pengujian <i>Black Box</i>	58
4.2.4.2	Pengujian <i>White Box</i>	59
4.2.5	<i>Maintenance</i>	63

BAB V PENUTUP

5.1	Kesimpulan	64
5.2	Saran	65

DAFTAR PUSTAKA

DAFTAR TABEL

Tabel 2.1	Kajian Empirik	5
Tabel 2.2	Tabel Spesifikasi Arduino Mega 2560	8
Tabel 2.3	Perbedaan RFID <i>Tag</i> Aktif dan Pasif	20
Table 2.4	Spesifikasi LCD 16 x 2	23
Tabel 2.5	<i>Flowchart</i>	32
Tabel 4.1	Hasil Pengujian Sensor MFRC522 Tanpa Halangan	44
Tabel 4.2	Hasil Pengujian Sensor MFRC522 Dengan Halangan	44
Tabel 4.3	Pengujian <i>Black Box</i>	58
Tabel 4.4	Pengujian <i>White Box</i>	59

DAFTAR GAMBAR

Gambar 2.1	Arduino Mega 2560	11
Gambar 2.2	Arduino Development Environment	16
Gambar 2.3	Matrix Membrane <i>Keypad</i> 4 x 4	18
Gambar 2.4	RFID <i>Tag</i>	19
Gambar 2.5	Modul RFID <i>Reader</i> RC522	21
Gambar 2.6	LCD (<i>Liquid Crystal Display</i>) 16 x 2.....	24
Gambar 2.7	Motor Servo SG90	24
Gambar 2.8	<i>Buzzer</i> Aktif	26
Gambar 2.9	Menu Fritzing	27
Gambar 2.10	Menu <i>Tab</i> Fritzing	28
Gambar 2.11	Model Air Terjun (<i>Waterfall</i>)	29
Gambar 2.12	<i>Pararelisme</i> dalam <i>Waterfall</i>	30
Gambar 4.1	Blok Diagram	46
Gambar 4.2	<i>Flowchart</i> Sistem	48
Gambar 4.3	Desain Pengendali Pintu Otomatis	49
Gambar 4.4	Komponen Alat Pengendali Pintu Otomatis	50
Gambar 4.5	Arduino Mega	51
Gambar 4.6	Keypad Membrane Matrix 4x4	52
Gambar 4.7	RFID MFRC522	53
Gambar 4.8	Servo Motor SG90	53
Gambar 4.9	LCD 16x2 dengan Serial I2C	54
Gambar 4.10	Susunan komponen pada Breadboard	55

DAFTAR LAMPIRAN

- Lampiran 1 Biaya Pembuatan Alat
- Lampiran 2 Lampiran *Coding* Program Arduino IDE
- Lampiran 3 Datasheet Arduino Mega 2560
- Lampiran 4 Datasheet Keypad Membrane 4x4
- Lampiran 5 Datasheet RFID MFRC522
- Lampiran 6 Datasheet LCD 16x2
- Lampiran 7 Datasheet Micro Servo SG90

BAB I

PENDAHULUAN

1.1 Latar Belakang

Teknologi berkembang dengan sangat cepat, perkembangan terus terjadi demi mempermudah pekerjaan manusia dari waktu ke waktu yang membutuhkan mobilitas tinggi dalam melakukan pekerjaan serta otomatisasi sehingga manusia mendapatkan kemudahan dari teknologi tersebut.

Akibat dari perkembangan teknologi di era industri modern sekarang ini, berbagai macam teknologi banyak bermunculan mulai dari penemuan baru sampai penyempurnaan dari teknologi sebelumnya. Pada saat yang sama perkembangan teknologi khususnya terhadap ruang penyimpanan kendaraan roda dua dan roda empat seperti garasi juga diperlukan. Mengingat banyaknya kasus pencurian terhadap kendaraan semakin meningkat. Pada umumnya garasi yang ada sekarang kurang memiliki sistem pengedali yang baik dan hanya dikunci dengan seadanya, yang membuat kendaraan menjadi tidak aman karena rentan dengan pencurian setiap saat.

Berdasarkan permasalahan tersebut, maka dibutuhkan teknologi untuk merancang sistem pengedali pintu garasi yang lebih aman dan canggih serta fleksibel, sehingga inovasi tersebut diwujudkan dengan membuat sistem yang dapat mengunci garasi menggunakan *password* yang dikontrol oleh mikrokontroler. Pengembangan sistem ini diharapkan mampu menciptakan sistem

pengendali pintu yang lebih terintegrasi. Dengan menggunakan *keypad* membran 4x4 dan RFID sebagai sistem pengendali, *keypad* dan RFID sensor ini dirancang dengan menggunakan mikrokontroler yang terpasang pada Arduino. Dengan menggabungkan sebuah mikrokontroler, *keypad*, RFID dan komponen lainnya, maka terciptalah sebuah sistem pengendali pintu elektronika.

1.2 Rumusan Masalah

Dari latar belakang dapat dirumuskan masalah penelitian ini yaitu :
“Bagaimana membuat *prototype* pengendali pintu garasi menggunakan sistem minimum Arduino Mega 2560 ?”

1.3 Batasan Masalah

Dengan adanya permasalahan diatas maka ruang lingkup pembahasan ini meliputi :

1. *Input* yang digunakan adalah 4x4 *membrane keypad matrix* dan RFID.
2. Pintu garasi digerakkan oleh Servo Motor SG90 yang dikendalikan dengan *keypad*/RFID.

1.4 Tujuan Penelitian

Adapun tujuan penelitian adalah sebagai berikut:

1. Membuat *prototype* pengendali pintu menggunakan RFID dan *keypad*.
2. Membuat akses menuju ruang tengah menggunakan RFID dan *keypad*.

1.5 Manfaat Penelitian

Adapun manfaat dari penelitian ini diharapkan dari sistem yang sudah dibuat dapat memberikan manfaat kepada pihak-pihak yang terkait, terutama bagi mahasiswa dan pelajar untuk mengembangkan pengetahuan dan memperlebar

referensi pembelajaran tentang elektronika, khususnya pada komunikasi berbasis radio agar dapat diimplementasikan secara nyata pada kehidupan sehari – hari di berbagai aspek kehidupan. Secara umum penelitian ini diharapkan mampu menjadi sumber inspirasi dan informasi sebagai acuan pengembangan penelitian dengan studi literatur yang lebih lengkap.

1.6 Sistematika Penulisan

Sistematika penulisan dalam penelitian ini adalah sebagai berikut :

BAB I PENDAHULUAN

Dalam bab ini menjelaskan mengenai latar belakang masalah, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Berisi tentang uraian mengenai teori-teori dasar yang akan mendukung pembahasan masalah yang berkaitan dengan judul penelitian

BAB III METODE PENELITIAN

Menjelaskan tentang metode penelitian yang dipakai dalam mengadakan penelitian untuk penyusunan Skripsi ini yang meliputi tempat dan waktu penelitian, Teknik Pengumpulan dan Analisa Data.

BAB IV HASIL PENELITIAN DAN PEMBAHASAN

Bab ini menjelaskan analisa sistem yang diusulkan dengan menggunakan flowchart dan blok diagram dari sistem yang akan dibuat, serta pembahasan secara detail tentang alur pembuatan prototipe beserta implementasinya

BAB V PENUTUP

Bab ini berisi kesimpulan dari penelitian dan saran yang berkaitan dengan analisa dan optimalisasi sistem berdasarkan yang telah diuraikan pada bab-bab sebelumnya.

BAB II

TINJAUAN PUSTAKA

2.1 Kajian Empirik

Kajian empirik ini dilakukan guna melihat sistem-sistem yang telah dibangun sebelumnya. Berikut adalah kajian empirik yang telah didapat dari penelitian sebelumnya.

Tabel 2.1 Kajian Empirik

No	Nama	Judul Skripsi	Hasil Penelitian
1	Alvin Dio Pratama (2014)	Perancangan dan Pembuatan Alat Mengukur Tinggi Badan Otomatis berbasis Mikrokontroler	Mengukur tinggi badan pengunjung wahana dan akan memberikan tanda suara apabila melebihi batas yang ditentukan
2	Sri Setyani (2016)	Rancang Bangun Alat Pengaman Brankas Menggunakan RFID (Radio Frequency Identification) Dengan Memanfaatkan E-KTP Sebagai Tag Berbasis Arduino	Mengoptimalkan penggunaan e-KTP sebagai pengganti tag yang digunakan untuk mengakses brankas
3	Saragih Shella Fransiska (2017)	Rancang dan Bangun Sistem Kendali Bak Penampung Air dan Lampu Kamar Mandi Menggunakan Arduino Uno	Mengendalikan bak penampung air dengan menggunakan sistem minimum arduino uno

Sumber : Alvin Dio Pratama (2014) Perancangan dan Pembuatan Alat Mengukur Tinggi Badan Otomatis berbasis Mikrokontroler, Sri Setyani (2016) Rancang Bangun Alat Pengaman Brankas Menggunakan RFID (Radio Frequency Identifiacion) Dengan Memanfaatkan E-KTP Sebagai Tag Berbasis Arduino, Saragih Shella Fransiska (2017) Rancang dan Bangun Sistem Kendali Bak Penampung Air dan Lampu Kamar Mandi Menggunakan Arduino Uno

Menurut Alvin Dio Pratama, 2014, pada penelitiannya yang berjudul Perancangan dan Pembuatan Alat Mengukur Tinggi Badan Otomatis Berbasis Mikrokontroler, pembuatan alat yang mampu mendeteksi tinggi badan secara otomatis dan memberikan sinyal agar pintu masuk wahana terbuka, apabila tinggi badan tidak memenuhi syarat maka alarm berbunyi dan akan menjelaskan bahwa tinggi badan melebihi batas yang di tentukan, di butuhkan empat komponen utama yaitu sensor ultrasonik sebagai *input*, mikrokontroler sebagai pemroses, serta LCD dan alarm sebagai *output*.

Sri Setyani, 2016, pada penelitian yang berjudul Rancang Bangun Alat Pengaman Brankas Menggunakan RFID (*Radio Frequency Identification*) Dengan Memanfaatkan E-KTP Sebagai Tag Berbasis Arduino, pada penelitian ini penulis berusaha untuk mengoptimalkan penggunaan e-KTP sebagai pengganti tag yang digunakan untuk mengakses brankas. Tujuan penggunaan e-KTP dengan alasan e-KTP setiap orang berbeda, sehingga e-KTP yang tidak terdaftar secara otomatis sistem sehingga akan menolaknya dan brankas tidak akan terbuka. Hal ini diharapkan dapat lebih menjamin keamanan dalam pengaksesan brankas itu sendiri

Saragih Shella Fransiska, 2017, pada penelitian yang berjudul Rancang dan Bangun Sistem Kendali Bak Penampung Air dan Lampu Kamar Mandi Menggunakan Arduino Uno. Dari hasil penelitian ini dapat mengendalikan kran pada bak air dan lampu kamar mandi dengan otomatis sehingga pengguna tidak perlu lagi menyalakan atau menghidupkan pakai saklar lampu, membuka kran ataupun menunggu air pada bak air kamar mandi hingga penuh.

Pada penelitian yang berjudul Prototype Pengendali Pintu Garasi Menggunakan Sistem Minimum Arduino Mega 2560. Terdapat sebuah miniatur rumah dengan garasi terkunci yang dapat dibuka dengan memasukkan *password* pada *Keypad* 4x4 dan *ID card* yang dapat diidentifikasi dengan *RFID reader*, servo motor digunakan untuk menggerakkan pintu garasi dan buzzer yang berbunyi apabila *RFID reader* tidak dapat mengenali *ID card* atau *password* yang salah.

Dari penelitian yang telah dipaparkan diatas terdapat beberapa persamaan dalam penelitian, yaitu menggunakan *RFID* dan *Keypad* sebagai *input* dan *buzzer* sebagai *output* peringatan, tetapi dalam penelitian yang telah disebutkan terdapat pula perbedaan dalam penelitian yaitu hasil keluaran setiap penelitian berbeda begitu pula dengan nilai yang akan dimanfaatkan dalam penelitian.

2.2 Kajian Teoritik

Dalam penelitian ini diperlukan suatu konsep dalam merumuskan definisi yang menunjang kegiatan penelitian, baik teori dasar maupun teori umum.

2.2.1 Hardware

Menurut Yudho Yudhanto (2018), *hardware* adalah komponen pembentuk sebuah komputer secara fisik. Perangkat keras komputer atau yang sering disebut *hardware* adalah benda nyata yang bisa disentuh, diraba, dilihat dan fungsinya adalah untuk membantu proses komputerisasi. *Hardware* komputer juga dapat diartikan alat yang dirancang untuk menerima dan mengolah data. Masing - masing *hardware* yang terdapat pada komputer telah mempunyai tugas sendiri-sendiri sehingga menghasilkan sebuah sistem komputer yang utuh dan bekerja dengan baik.

2.2.2 Garasi

Menurut Choirul Amin (2010), Pintu garasi adalah media penghubung aktivitas keluar masuk dari *carport* atau halaman depan kedalam atau keluar menuju garasi. Garasi adalah suatu ruangan untuk menempatkan kendaraan (mobil, motor dll), agar lebih aman dan terlindung dari cuaca luar.

Agar pintu garasi dapat berfungsi dengan baik sekaligus mempunyai daya tahan yang lama maka dibutuhkan beberapa persyaratan tertentu, antara lain pintu garasi harus kokoh dan kuat, aman, serta mempunyai ukuran yang memadai untuk menunjang aktivitas yang terdapat di dalamnya.

2.2.3 Arduino Mega

Menurut Heri Andrianto (2016), *Board* Arduino Mega adalah sebuah *board* Arduino yang menggunakan ic mikrokontroler ATmega 2560. *Board* ini memiliki 54 digital *input/output* (15 buah diantaranya dapat digunakan sebagai *output* PWM), 16 buah *analog input*, 4 UARTs (*universal asynchronous receiver/transmitter*), osilator kristal 16 MHz, koneksi USB, *jack power*, soket ICSP (*In-Circuit System Programming*) dan tombol *reset*. Spesifikasi *board* Arduino mega 2560 dapat dilihat pada table 2.2 berikut ini.

Tabel 2.2 Tabel Spesifikasi Arduino Mega 2560

Spesifikasi	Keterangan
Mikrokontroler	ATMega 2560
Tegangan Operasional	5V
Tegangan Input (rekomendasi)	7-12V
Tegangan Input (limit)	6-20V
PIN Digital I/O	54 (15 buah diantaranya dapat digunakan

	sebagai output PWM)
PIN Analog input	16 (A0 s.d A15)
Arus DC per Pin I/O	40 mA
Arus DC untuk PIN 3.3V	50 mA
Memori Flash	256 KB, 8 KB digunakan untuk bootloader
SRAM	8 KB
EEPROM	4KB
Clock Speed	16 MHz

Sumber : Heri Andrianto, 2016 Arduino Belajar Cepat dan Pemrograman

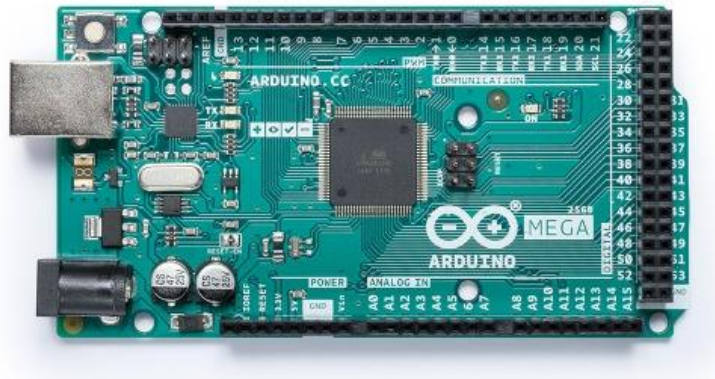
Semua pin digital yang terdapat pada Arduino Mega 2560 dapat digunakan baik sebagai *input* maupun *output* dengan menggunakan fungsi *pinMode()*, *digitalWrite()* dan *digitalRead()*. Tegangan *output* setiap pin adalah 5 Volt. Arus maksimum yang dapat diberikan dan diterima sebesar 40 mA. Pada pin digital ini juga terdapat *internal pull up* resistor sebesar 20-50 KOhm. Beberapa pin memiliki fungsi khusus seperti berikut.

1. Arduino Mega 2560 memiliki kemampuan untuk berkomunikasi dengan komputer, *Board* Arduino lain dan mikrokontroler lainnya. ATmega 2560 memiliki 4 buah UART untuk komunikasi serial TTL. Pin 0 dan 1 terhubung langsung dengan IC ATmega16U2 USB to TTL Serial Chip. IC tersebut merupakan IC *converter* USB ke serial. TTL LED RX dan TX pada *Board* akan menyala saat ada data yang dikirim melalui ATmega16U2 dan koneksi ke komputer melalui USB. Berikut ini *port serial* yang ada pada Arduino Mega2560, yaitu *Port Serial*: pin 0 (RX) dan pin 1 (TX); *Port Serial* 1: pin 19 (RX) dan pin 18 (TX); *Port Serial* 2: pin 17 (RX) dan pin 16 (TX); *Port Serial* 3: pin 15 (RX) dan pin 14

(TX). Pin RX digunakan untuk menerima data serial TTL dan pin (TX) untuk mengirim data serial TTL.

2. *External Interrupts*: pin 2 (*interrupt 0*), pin 3 (*interrupt 1*), pin 18 (*interrupt 5*), pin 19 (*interrupt 4*), pin 20 (*interrupt 3*) dan pin 21 (*interrupt 2*).
3. PWM: pin 2 sampai dengan pin 13 dan pin 44 s.d pin 46. Pin-pin tersebut dapat digunakan sebagai *output* PWM 8 bit.
4. SPI: pin 50 (MISO), pin 51 (MOSI), pin 52 (SCK), pin 53 (SS). Digunakan untuk komunikasi SPI
5. LED: pin 13. Terdapat LED yang terhubung dengan pin 13.
6. TWI: pin 20 (SDA) dan pin 21 (SCL). Pin-pin tersebut dapat digunakan untuk komunikasi TWI. ATmega 2560 juga mendukung komunikasi TWI dan SPI. *Software* Arduino memiliki *wire library* dan *SPI library* untuk mempermudah penggunaan fitur komunikasi TWI dan SPI.
7. AREF: *input* untuk tegangan referensi *input analog*.
8. Reset: digunakan untuk mereset *Board* Arduino Mega

Arduino Mega 2560 juga memiliki 16 buah *input analog* (ADC), yaitu pin A0 s.d A15. Setiap *input* memiliki resolusi sebesar 10bit. Arduino Mega 2560 dilengkapi dengan *resettable polyfuse* yang dapat melindungi *port* USB dari hubungan arus pendek dan kelebihan arus. Meskipun pada setiap komputer sudah terdapat pelindung *internal*, *fuse* ini akan memberikan perlindungan tambahan. Apabila arus yang lewat lebih besar dari 500mA, *fuse* akan otomatis terputus sampai kelebihan arus atau hubungan arus pendek dapat diperbaiki.



.Gambar 2.1 Arduino Mega 2560

Sumber : Heri Andrianto, 2016, Arduino Belajar Cepat dan Pemrograman

2.2.3.1 Sistem Minimum Mikrokontroler

Menurut Bachtiar Effendi (2014), Sistem minimum mikrokontroler adalah sistem elektronika yang terdiri dari komponen – komponen dasar yang dibutuhkan oleh suatu mikrokontroler untuk dapat berfungsi dengan baik. Pada umumnya suatu mikrokontroler membutuhkan dua elemen (selain *power supply*) untuk berfungsi : Kristal Oscilator (XTAL) dan Rangkaian *Reset*. Analogi fungsi kristal adalah jantung pada tubuh manusia. Perbedaannya, jantung memompa darah dan seluruh kandungannya, sedangkan XTAL memompa data. Dan fungsi rangkaian *reset* adalah untuk membuat mikrokontroler memulai kembali pembacaan programnya, hal tersebut dibutuhkan pada saat mikrokontroler mengalami gangguan dalam mengeksekusi program.

2.2.3.2 Bahasa Pemrograman Arduino

Menurut Rafiuddin Syam (2013), *Arduino Board* merupakan perangkat yang berbasis mikrokontroler. *Software* merupakan komponen yang membuat

sebuah mikrokontroler dapat bekerja. Arduino *board* akan bekerja sesuai dengan perintah yang ada dalam perangkat lunak yang ditanamkan kepadanya.

Bahasa pemrograman arduino adalah bahasa pemrograman utama yang menggunakan bahasa pemrograman C untuk membuat program pada arduino. Karena menggunakan bahasa pemrograman C sebagai dasarnya, maka pemrograman arduino memiliki banyak sekali kemiripan, walaupun beberapa hal telah berubah.

Menurut Heri Andrianto (2016), fungsi – fungsi dasar pada bahasa pemrograman arduino :

1. *Setup*

Fungsi *setup()* dipanggil ketika program dijalankan, berfungsi untuk inisialisasi mode *pin* sebagai *input* atau *output* dan inisialisasi serial. Fungsi ini harus ada meski tidak ada instruksi yang ditulis.

2. *Loop*

Setelah memanggil fungsi *setup()*, program yang berada dalam fungsi *loop* akan dieksekusi secara terus menerus, hingga arduino *board* dimatikan (catu daya diputus).

3. *Function*

Fungsi adalah sekumpulan blok instruksi yang memiliki nama sendiri dan blok instruksi ini akan dieksekusi ketika fungsi ini dipanggil. Penulisan fungsi harus didahului dengan tipe fungsi setelah itu nama fungsi dan kemudian parameternya, bila tidak ada nilai yang dihasilkan dari fungsi tersebut, tipe fungsinya adalah *void()*.

4. Kurung Kurawal

Kurung Kurawal ({}) digunakan untuk mengawali dan mengakhiri sebuah fungsi, blok instruksi seperti *loop()*, *void()* dan instruksi pengulangan *for* dan seleksi kondisi *if*.

5. Titik Koma

Kode pada Arduino harus diakhiri dengan tanda titik koma. Biasa digunakan sebagai akhir dari sebuah instruksi.

6. Blok Komentar

Digunakan untuk memberi komentar pada program yang memiliki baris lebih dari satu, biasanya digunakan untuk membantu memahami program yang dibuat. Diawali dengan tanda */** dan diakhiri dengan tanda **/*. Adapun yang ditulis dalam blok komentar ini tidak akan berpengaruh dengan program yang telah dibuat dan tidak akan menghabiskan memori.

7. Komentar Baris

Digunakan untuk memberi komentar per baris program, sama seperti blok komentar, komentar baris tidak akan menghabiskan memori dan tidak berpengaruh pada program.

8. Variabel

Variabel adalah ekspresi yang digunakan untuk mewakili suatu nilai yang digunakan dalam program. Suatu variabel akan menampung nilai sesuai dengan definisi yang telah dibuat.

Variabel perlu didefinisikan terlebih dahulu sebelum digunakan. Variabel didefinisikan sesuai dengan tipe data nilainya seperti *int*, *float*, *long*, *string* dll.

Variabel hanya perlu didefinisikan satu kali saja tetapi nilainya dapat berubah sesuai dengan perhitungan ataupun melalui program.

9. *Array*

Array adalah kumpulan nilai – nilai yang diakses dengan nomor *index*. Setiap nilai dalam *array* dapat dipanggil dengan memanggil nama *array* dan nomor *index* dari nilai tersebut. Nomor *index* dimulai dari 0 (nol). Variabel *array* harus dideklarasikan sebelum dapat digunakan. Untuk mendeklarasikan *array* harus dengan menyatakan tipe *array*, ukuran dan kemudian memberikan nilai pada posisi *index*.

Array sering digunakan pada program *loop* dan sering dikombinasikan dengan suatu *counter* yang digunakan sebagai penunjuk posisi *index* untuk setiap nilai *array*.

10. *Aritmatika*

Operator aritmatika meliputi operasi penjumlahan, pengurangan, pembagian dan perkalian.

11. *Delay*

Instruksi untuk memberi jeda sebelum lanjut ke program selanjutnya. Jeda dalam satuan mili detik.

12. *Konstanta*

Konstanta adalah variabel yang sudah ditetapkan sebelumnya dalam pemrograman arduino. Konstanta digunakan agar program lebih mudah untuk dibaca dan dimengerti.

13. **TRUE / FALSE**

True / False adalah konstanta boolean yang mendefinisikan nilai logika. Nilai *False* dapat didefinisikan sebagai 0 (nol), sedangkan nilai *True* sering didefinisikan sebagai 1, tetapi dalam hal lain dapat didefinisikan sebagai nol.

14. **HIGH / LOW**

Konstanta ini menentukan nilai pin sebagai *HIGH* atau *LOW* dan digunakan ketika membaca atau menulis ke pin digital. *HIGH* didefinisikan sebagai tingkat logika 1 atau *ON* atau 5 Volt, sedangkan *LOW* adalah tingkat logika 0 atau *OFF* atau 0 Volt.

15. **Seleksi If...Else**

Seleksi *if* dan *else* memungkinkan apabila instruksi *if* yang berjalan jika suatu kondisi terpenuhi, apabila tidak terpenuhi maka statement *else* akan dijalankan.

16. **INPUT / OUTPUT**

Konstanta yang sering digunakan pada fungsi `pinMode()` untuk menentukan mode pin digital sebagai input atau output.

17. **For**

Pernyataan *for* digunakan untuk mengulang suatu blok instruksi di dalam kurung kurawal.

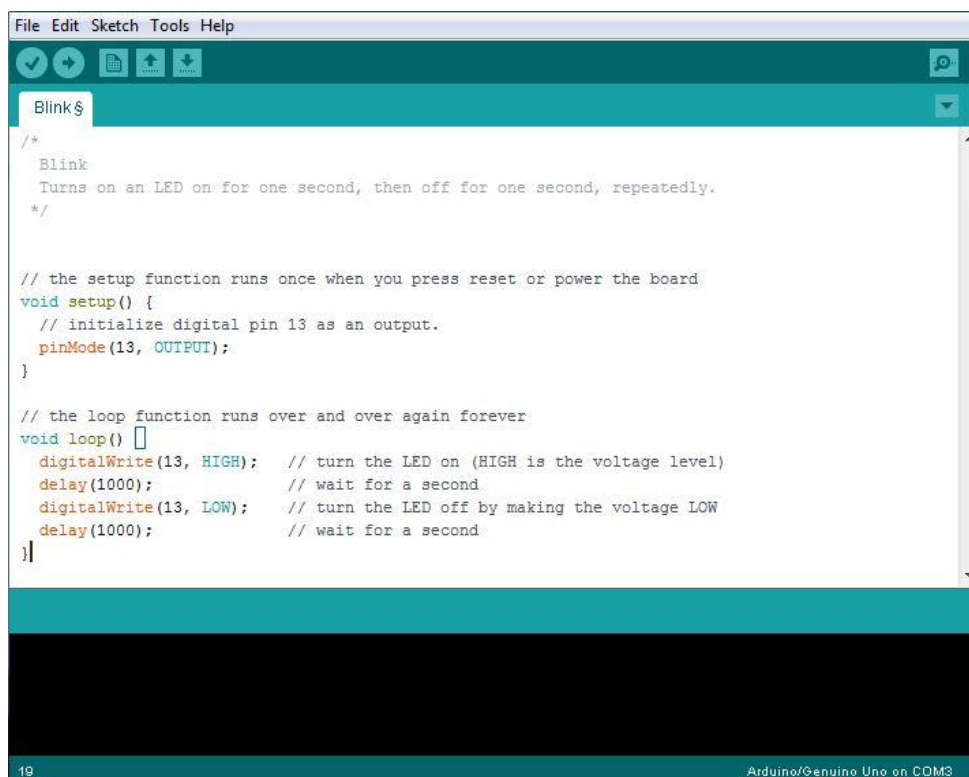
18. **While**

Fungsi *while* akan menjalankan program secara terus menerus hingga suatu kondisi pada fungsi *while* bernilai salah atau *false*.

2.2.3.3 Arduino IDE

Menurut Rafiuddin Syam (2013), *Arduino Development Environment* terdiri dari *editor* teks untuk menulis kode, sebuah area pesan, sebuah konsol, sebuah *toolbar* dengan tombol – tombol untuk fungsi yang umum dan beberapa menu. *Arduino Development Environment* terhubung ke *arduino board* untuk meng-*upload* program dan juga untuk berkomunikasi dengan *arduino board*.

Perangkat lunak yang ditulis menggunakan *Arduino Development Environment* disebut *sketch*. *Sketch* ditulis pada *editor* teks. *Sketch* disimpan dengan file berekstensi *.ino*. Area pesan memberikan informasi dan pesan *error* ketika kita menyimpan atau membuka *sketch*. Konsol menampilkan *output* teks dari *Arduino Development Environment* dan juga menampilkan pesan.



```
File Edit Sketch Tools Help
Blink$
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.
*/


// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}


// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);           // wait for a second
  digitalWrite(13, LOW); // turn the LED off by making the voltage LOW
  delay(1000);           // wait for a second
}
19 Arduino/Genuino Uno on COM3
```


Gambar 2.2 Arduino Development Environment


Sumber : Heri Andrianto, 2016, *Arduino Belajar Cepat dan Pemrograman*


Berikut ini adalah tombol-tombol *toolbar* serta fungsinya :

 *Verify* : Mengecek *error* pada *code* program

 *Upload* : meng-*compile* dan meng-*upload* program ke *Arduino board*.

 *New* : membuat *sketch* baru.

 *Open* : menampilkan sebuah menu dari seluruh *sketch* yang berada di dalam *sketchbook*.

 *Save* : menyimpan *sketch*.

 *Serial Monitor* : membuka serial monitor.

Dalam *arduino* digunakan sebuah konsep yang disebut *sketchbook*, yaitu tempat untuk menyimpan program (*sketch*). *Sketch* yang ada pada *sketchbook* dapat dibuka dari menu *File > Sketchbook* atau dari tombol *open* pada *toolbar*. Ketika pertama kali menjalankan *arduino development environment*, sebuah direktori akan dibuat secara otomatis untuk tempat menyimpan *sketchbook*. Pengguna dapat mengganti lokasi direktori tersebut dari menu *File > Preferences*.

Sebelum meng-*upload* program, pengguna perlu mensetting jenis board dan port serial yang sedang digunakan melalui menu *Tools > Board* dan *Tools > Serial Port*. Pemilihan *board* berguna untuk mengeset parameter (contohnya: kecepatan mikrokontroler dan *baud rate*) yang digunakan ketika meng-*compile* dan meng-*upload sketch*. Setelah memilih *board* dan *port* serial yang tepat, tekan tombol *upload* pada *toolbar* atau pilih menu *file > upload*. *Arduino board* akan me-*reset* secara otomatis dan proses *upload* akan dimulai. Pada kebanyakan *board*, LED RX dan TX akan berkedip ketika program sedang di-*upload*. *Arduino*

development environment akan menampilkan pesan ketika proses *upload* telah selesai, atau menampilkan pesan *error*.

Ketika meng-*upload* program, *arduino bootloader* sedang digunakan, *Arduino bootloader* adalah sebuah program kecil yang telah ditanamkan pada mikrokontroler yang berada pada *arduino board*. *Bootloader* ini mengijinkan kita meng-*upload* program tanpa menggunakan perangkat keras tambahan.

2.2.4 Matrix Membrane Keypad 4x4

Menurut Heri Andrianto (2016), Keypad 4x4 adalah 16 buah saklar yang dibentuk matrik, dengan tujuan penghematan jalur *input/output*, apabila ke 16 saklar tersebut tidak dibentuk matrik maka dibutuhkan 16 jalur *input*, tetapi dengan dibentuk dalam matrik, maka hanya dibutuhkan 8 jalur (4 jalur *input* dan 4 jalur *output*), cara menggunakan *keypad* 4x4 ini dengan metode *scanning*, tiap baris (kolom) diberi logik 0 (0V) secara bergilir, setiap baris (kolom) yang mendapat logik 0, selanjutnya set seluruh kolom dengan logik 1, tahap berikutnya tiap baris (kolom) diuji logiknya, apabila ada baris (kolom) yang terbaca logik 0 (0V), berarti baris tersebut sedang ditekan.



Gambar 2.3 Matrix Membrane Keypad 4x4

Sumber : Heri Andrianto, 2016, *Arduino Belajar Cepat dan Pemrograman*

2.2.5 RFID

Menurut Heru Nugroho (2015), RFID adalah salah satu teknologi Auto-ID. RFID menggunakan media *tag* atau *chips* dan mengirimkan data melalui *frequency* untuk meng-identitaskan suatu produk ke komputer, sehingga data yang direkam adalah data atau data seketika. Teknologi RFID bergantung pada transmisi data nirkabel melalui medan elektromagnetik.

Peranti ini terdiri dari dua bagian. Peranti pertama adalah RFID *reader* yang berfungsi untuk membaca kode dari RFID *tag (label)* dan membandingkan dengan yang ada pada memori *reader*. Sedangkan bagian kedua adalah RFID *tag* yang berfungsi untuk menyimpan kode-kode sebagai pengganti identitas diri. Pada umumnya yang digunakan pada proses implantasi ini adalah RFID *pasif*.

2.2.5.1 RFID Tag

RFID *tag* terdiri dari chip rangkaian sirkuit yang terintegrasi dan sebuah antena. Rangkaian elektronik dari RFID *tag* umumnya memiliki memori. Memori ini memungkinkan RFID *tag* mempunyai kemampuan untuk menyimpan data. Memori pada *tag* dibagi menjadi beberapa sel, untuk menyimpan data *Read Only*. Semua RFID *tag* mendapatkan nomor ID pada saat *tag* tersebut diproduksi.



Gambar 2.4 RFID Tag

Sumber : Heru Nugroho, 2015, Matematika Diskrit dan Implementasinya dalam Dunia Teknologi Informasi

Selain pada RFID *tag* memungkinkan RFID *tag* tersebut dapat ditulis (*Write*) dan dibaca secara berulang. Setiap *tag* dapat membawa informasi yang unik, seperti nomor ID, tanggal lahir, alamat, jabatan dan data lain dari objek yang akan diidentifikasi. Banyaknya informasi yang dapat disimpan oleh RFID *tag* tergantung pada kapasitas memorinya. Semakin banyak fungsi yang dapat dilakukan oleh RFID *tag* maka rangkaian akan semakin kompleks dan ukurannya akan semakin besar. Berdasarkan catu daya, RFID *tag* digolongkan menjadi:

1 Tag Pasif

Tag pasif merupakan RFID *tag* yang tidak memiliki *power supply* sendiri. *Tag* pasif hanya mengandalkan induksi listrik yang ditimbulkan oleh antena karena adanya frekuensi radio *scanning* yang masuk sebagai penyuplai daya bagi RFID *tag* untuk mengirimkan respon balik.

2 Tag Aktif

Tag yang catu dayanya diperoleh dari baterai atau *tag* yang mempunyai *power supply* sendiri, sehingga akan mengurangi daya yang diperlukan oleh RFID *reader* sehingga *tag* dapat mengirimkan informasi dalam jarak yang lebih jauh. Jarak jangkauannya bisa mencapai 10 meter dengan umur baterai hingga beberapa tahun. Untuk mengetahui perbedaan karakteristik umum yang ada pada RFID *tag* aktif dan pasif, berikut dijelaskan pada tabel 2.3 di bawah ini:

Table 2.3 Perbedaan RFID Tag Aktif dan Pasif

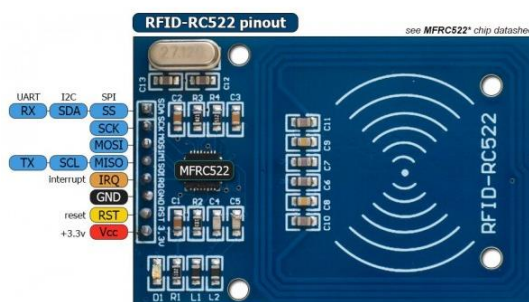
No	Deskripsi	Tag Pasif	Tag Aktif
1	Catu daya	<i>Eksternal</i> (dari <i>reader</i>)	Baterai <i>internal</i>
2	Rentang daya	3 meter atau kurang	100 meter atau lebih
3	Tipe memori	Umumnya <i>read-only</i>	<i>Read-write</i>

4	Usia tag	Mencapai 20 tahun	5 sampai 10 tahun
5	Ketersediaan daya	Hanya pada jangkauan <i>reader</i>	Bersifat kontinyu

2.2.5.2 RFID Reader

RFID *reader* merupakan pembaca RFID *tag* yang kompatibel mampu mengeluarkan gelombang radio dan menginduksi RFID *tag*. Gelombang tersebut berisi *password* dan jika dikenali oleh RFID *tag*, memori RFID *tag* akan terbuka. RFID *tag* akan mengirimkan kode yang terdapat di memori chip melalui antena yang terpasang di RFID *tag*. Selanjutnya RFID *reader* akan membandingkan kode yang diterima dengan kode kunci yang tersimpan.

Sebuah RFID *reader* harus menyelesaikan dua buah tugas, yaitu: menerima perintah dari aplikasi dan berkomunikasi dengan *tag*. RFID *reader* bertugas sebagai penghubung antara aplikasi dengan antena yang akan meradiasikan gelombang radio ke *tag* RFID.



Gambar 2.5 Modul RFID Reader RC522

Sumber : Heru Nugroho, 2015, Matematika Diskrit dan Implementasinya dalam Dunia Teknologi Informasi

2.2.5.3 Frekuensi Kerja RFID

Faktor yang juga harus diperhatikan dalam RFID adalah frekuensi kerja dari sistem RFID. Ada beberapa *band* frekuensi yang digunakan untuk sistem

RFID. Pemilihan frekuensi kerja sistem RFID akan mempengaruhi jarak komunikasi, interferensi dengan frekuensi sistem radio lain, kecepatan komunikasi data, dan ukuran antena.

Berdasarkan frekuensi kerjanya, kecepatan baca tag dan RFID frekuensi yang digunakan oleh sistem RFID ada 4 macam, yaitu:

1 Band LF (*Low Frequency*)

Rentang frekuensi 125 KHz sampai 134 KHz dengan penggunaan jarak pendek, biasanya digunakan untuk sistem identifikasi yang hanya membutuhkan jarak pendek.

2 Band HF (*High Frequency*)

Beroperasi pada frekuensi 13.56KHz dengan pembacaanya hingga kurang lebih 3 meter, pada frekuensi ini cocok digunakan untuk pembacaan pada tingkat item dan banyak digunakan untuk pencocokan barang ditoko, gedung atau pelacakan yang memerlukan dengan kecepatan baca 10 hingga 100 *tag* per detik.

3 Band UHF (Ultra High Frekuensi)

Frekuensi sekitar 868 sampai 956 MHz dengan rentang pembacaan hingga sekitar 9 meter. *Tag* UHF dapat dibaca dengan kecepatan hingga 1000 *tag* per detik. Biasanya digunakan untuk pelacakan barang pada container truk.

4 Gelombang mikro 2,45 GHz

Banyak digunakan untuk pelacakan rantai *supply* dengan jarak pembacaan yang jarak lebih jauh (10m) pada frekuensi banyak mengalami pantulan gelombang dan objek disekitarnya sehingga dapat mengganggu RFID *reader* untuk komunikasi dengan *tag* RFID.

2.2.6 LCD (*Liquid Crystal Display*) 16x2

Menurut Heri Andrianto (2016), LCD (*liquid crystal display*) 16x2 adalah suatu *display* dari bahan cairan Kristal yang pengoperasiannya menggunakan sistem dot matriks. LCD 16x2 dapat menampilkan sebanyak 32 karakter yang terdiri dari 2 baris dan tiap baris dapat menampilkan 16 karakter.

Table 2.4 Spesifikasi LCD (*Liquid Crystal Display*) 16x2

No. Pin	Nama Pin	Deskripsi
1	VSS (Ground)	Catu daya, <i>ground</i> (0V)
2	VDD (5 Volt)	Catu daya positif (+5V)
3	VE (Contrast V)	Untuk kontras tulisan pada LCD
4	RS (Register Select)	<ul style="list-style-type: none"> • High : untuk mengirim data • Low : untuk mengirim instruksi
5	RW (Read/Write)	<ul style="list-style-type: none"> • High : mode membaca data di LCD • Low : mode penulisan ke LCD • Disambungkan dengan LOW untuk pengiriman data ke layar
6	E (Enable)	Mengontrol ke LCD ketika bernilai LOW, LCD tidak dapat diakses
7-14	DB0 - DB7	Data <i>bus line</i> 0 – 7
15	Backlight +	Catu daya layar, <i>positif</i> (+5V)
16	Backlight -	Catu daya layar, <i>negatif</i> (0V)

Modul LCD memiliki karakteristik sebagai berikut:

1. Terdapat 16x2 karakter huruf yang bisa ditampilkan
2. Setiap huruf terdiri dari 5x7 huruf dot-matrix cursor.
3. Terdapat 192 macam karakter.
4. Terdapat 80x8 bit display RAM (maksimal 80 karakter)
5. Memiliki kemampuan penulisan dengan 8 bit maupun dengan 4 bit

6. Dibangun dengan osilator lokal
7. Satu sumber tegangan 5 Volt
8. Otomatis reset saat tegangan dihidupkan
9. Bekerja pada suhu 0°C sampai 50°C



Gambar 2.6 LCD (*Liquid Crystal Display*) 16x2

Sumber : Heri Andrianto, 2016, Arduino Belajar Cepat dan Pemrograman

2.2.7 Motor Servo

Menurut Abdul Kadir (2017), Motor servo mempunyai ciri khas yakni memiliki tiga kabel. Dua kabel digunakan untuk memberikan catu daya kepada motor dan satu kabel lagi digunakan untuk mengontrol arah putar rotor (bagian yang berputar). Umumnya, motor servo dirancang berputar dari sudut 0° hingga 180°. Namun, ada pula jenis motor servo yang dapat berputar 360°. Gambar 2.4 memperlihatkan contoh motor servo berukuran kecil dengan berat hanya 9 gram.



Gambar 2.7 Motor Servo SG90

Sumber : Abdul Kadir, 2017, Pemrograman Arduino dan Android Menggunakan

App Inventor

Tiga kabel di motor servo biasa diberi warna merah (kabel catu daya) dihubungkan ke sumber tegangan DC. Kabel warna coklat (kabel *ground*) dihubungkan ke *ground* dan kabel warna oranye (kabel kontrol) dihubungkan ke pin yang ditujukan untuk mengatur arah putar rotor.

Kekuatan putar motor servo dinyatakan dengan torsi. Semakin tinggi torsinya, semakin kuat motor tersebut untuk memutar suatu beban. Oleh karena itu, motor servo yang digunakan perlu disesuaikan dengan beban yang ingin diputar.

2.2.8 Breadboard

Menurut Firmansyah Saftari (2015), *Breadboard* adalah semacam papan PCB, tetapi tidak memerlukan solder. *Breadboard* banyak digunakan untuk merangkai komponen, karena dengan menggunakan *breadboard*, pembuatan prototipe tidak memerlukan proses solder. Karena sifatnya yang *solderless* alias tidak memerlukan solder sehingga dapat digunakan kembali dan dengan demikian sangat cocok digunakan pada tahapan proses pembuatan *prototype* serta membantu dalam desain sirkuit elektronika.

2.2.9 Power Supply

Power Supply atau dalam bahasa Indonesia disebut dengan Catu Daya adalah suatu alat listrik yang dapat menyediakan energi listrik untuk perangkat listrik ataupun elektronika lainnya. Pada dasarnya *Power Supply* atau Catu daya ini memerlukan sumber energi listrik yang kemudian mengubahnya menjadi energi listrik yang dibutuhkan oleh perangkat elektronika lainnya. Oleh karena itu, *Power Supply* kadang disebut juga dengan istilah *Electric Power Converter*.

2.2.10 Kabel Jumper

Menurut Firmansyah Saftari (2015), Kabel *jumper* adalah suatu istilah kabel yang ber-diameter kecil yang di dalam dunia elektronik digunakan untuk menghubungkan dua titik atau lebih dan dapat juga untuk menghubungkan 2 komponen atau lebih komponen elektronika.

2.2.11 Buzzer

Menurut Andi Dinata (2018), *Buzzer* adalah perangkat elektronik yang mengeluarkan suara monotone. Suara ini dihasilkan oleh getaran mekanis yang diakibatkan oleh arus listrik. Arus listrik yang diterima digunakan untuk proses osilasi yang mengakibatkan getaran antara dua bidang. Getaran ini menimbulkan suara yang dikeluarkan melalui tabung resonansi sehingga terdengar sebuah nada. Karena suara monotone, maka hanya ada satu suara ‘beep’ yang dikeluarkan oleh *buzzer*. *Buzzer* banyak digunakan misalnya pada jam alarm, penanda jarak saat mobil parkir, juga terdapat pada lemari es yang berbunyi pada saat temperature berubah karena kita membuka pintu terlalu lama.



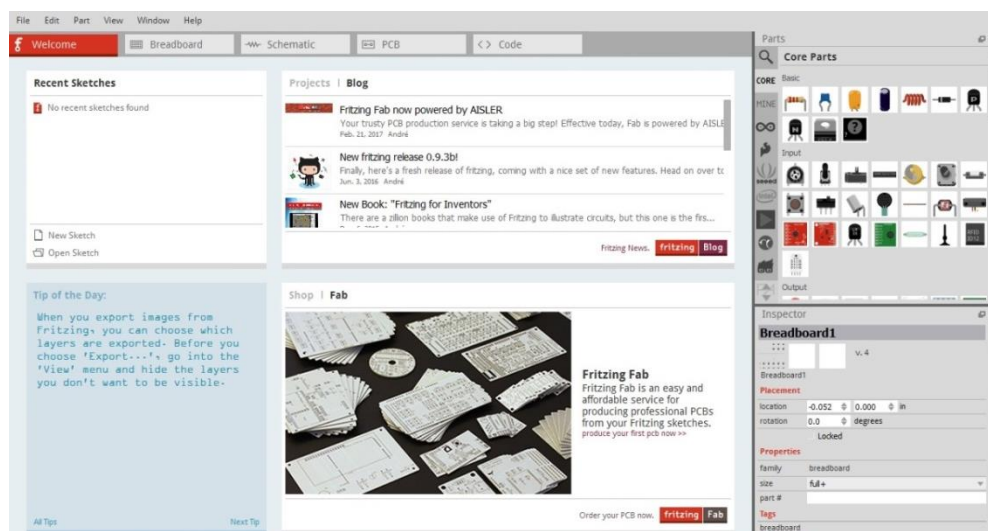
Gambar 2.8 Buzzer Aktif

Sumber : Andi Dinata, 2018, Fun Coding with MicroPython

Buzzer memiliki dua kaki, di mana salah satunya lebih panjang dari yang lainnya. Seperti LED, kaki yang lebih panjang sebagai kutub positif (anoda) dan kaki lebih pendek sebagai kutub negatif (katoda). *Buzzer* dapat berfungsi dengan sumber tegangan listrik 3 volt, sehingga dapat langsung dihubungkan dengan pin GPIO dari mikrokontroler.

2.2.12 Fritzing

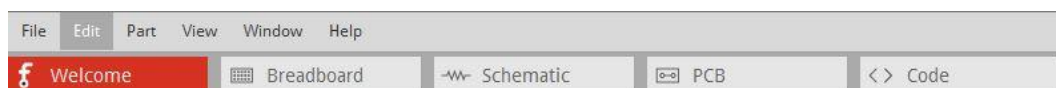
Menurut Heri Andrianto (2016), Fritzing adalah sebuah perangkat lunak gratis dan merupakan sebuah aplikasi *open source* yang didirikan oleh komunitas *online*. Fritzing (Ver 9.0 ke atas) dapat digunakan untuk mendesain PCB dua muka (*double sided*) dan dapat dikirim ke produsen PCB untuk diproduksi massal. Fritzing juga dapat digunakan untuk dokumentasi dan melakukan pemeriksaan desain rangkaian yang dibuat. Fritzing cukup mudah digunakan dan praktis, karena itu banyak digunakan oleh pengembang modul mikrokontroler Arduino, papan tunggal Raspberry-Pi dan sejenisnya.



Gambar 2.9 Menu Fritzing

Sumber : Heri Andrianto, 2016, Arduino Belajar Cepat dan Pemrograman

Fritzing memungkinkan pengguna untuk melihat kabel, pin dan koneksi secara langsung dalam bentuk grafik. Saat ini fritzing adalah program yang menggunakan representasi grafis langsung dari bagian-bagian yang digunakan, sehingga pengguna dapat dengan mudah melihat diagram dan mengetahui bagaimana segala sesuatu terhubung pada *breadboard*.



Gambar 2.10 Menu Tab Fritzing

Sumber : Heri Andrianto, 2016, Arduino Belajar Cepat dan Pemrograman

Fritzing (Ver 09.x) mempunyai 5 menu tab. Menu ***Breadboard***, untuk merancang rangkaian dengan menggunakan papan *breadboard* sebagai tempat peletakan komponen yang digunakan, menu ini sangat membantu dalam visualisasi koneksi secara fisik, sehingga memudahkan perancangan, pemeriksaan koneksi rangkaian maupun untuk tampilan *layout*. Menu ***Schematic***, merupakan menu untuk merancang rangkaian dengan menggunakan simbol komponen elektronika dalam pengoneksiannya, menu ini merupakan menu perancangan rangkaian yang umumnya digunakan dalam berbagai macam perangkat lunak lainnya. Menu ***PCB***, menu ini memperlihatkan pengoneksian antara komponen dalam sebuah papan PCB (satu muka). Fritzing juga menyediakan fasilitas keluaran *file gerber* dan *file PDF* untuk memudahkan pembuatan PCB secara massal. Dua menu lainnya yaitu menu ***Welcome*** merupakan menu yang memperlihatkan aktivitas *file* yang terakhir dibuka dan ada menu yang terhubung dengan blog komunitas fritzing dan pemakaiannya, adapun menu ***Code*** merupakan sarana untuk pembuatan program yang dapat dihubungkan dengan

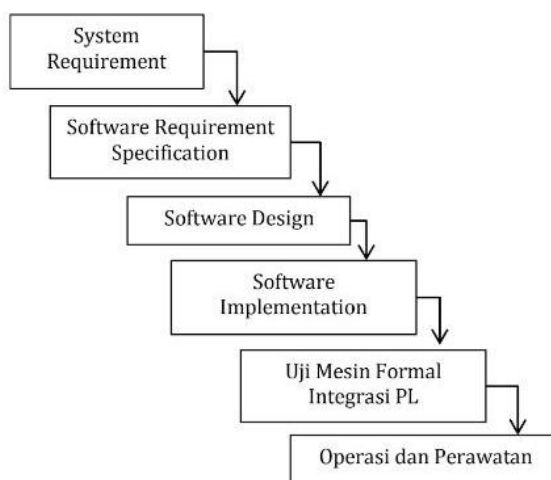
program IDE tertentu, misalnya jika dihubungkan dengan program IDE Arduino, kita dapat membuat program untuk *upload* ke Arduino melalui menu ini.

2.2.13 Metode Pengembangan Sistem

Metode pengembangan sistem adalah sebuah cara yang tersistem atau teratur yang bertujuan untuk melakukan analisa pengembangan suatu sistem tersebut dapat memenuhi kebutuhan, dalam melakukan pengembangan atau perbaikan suatu sistem yang dapat terkomputerisasi harus melakukan langkah – langkah dalam mengimplementasikannya.

2.2.13.1 *Waterfall*

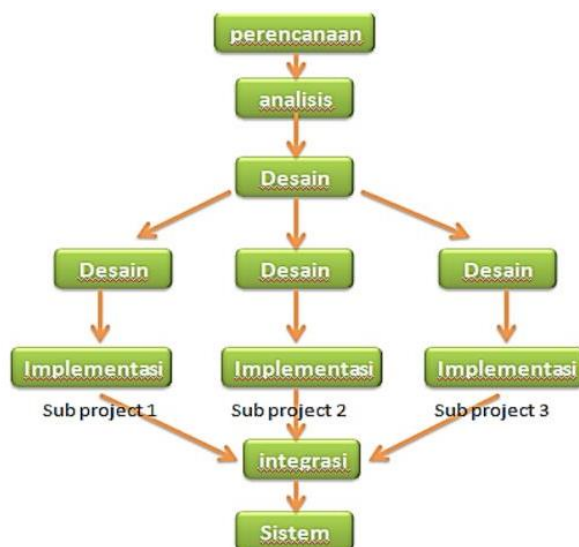
Menurut Yurindra (2017), Karakteristik konsep *Waterfall* merupakan pendekatan yang dikendalikan oleh dokumen. *Waterfall* merupakan model yang membangun perangkat lunak berdasarkan Daur Hidup Perangkat Lunak (SDLC), yaitu model yang mempunyai struktur yang dimulai dari perencanaan, analisis, desain dan implementasi, sehingga tahap pengembangan *waterfall* mempunyai struktur model pengembangan yang disebut dengan *linier* dan *sequential*.



Gambar 2.11 Model Air Terjun (*Waterfall*)

Sumber: Yurindra, 2017, *Software Engineering*.

Dalam *Waterfall* ada proses-proses yang dapat berjalan bersamaan pada waktu tertentu yang sering disebut dengan *pararelisme*.



Gambar 2.12 Pararelisme Dalam *Waterfall*

Sumber: Yurindra, 2017, *Software Engineering*.

Waterfall merupakan salah satu metode dalam SDLC yang mempunyai ciri khas pengerjaan yaitu setiap fase dalam *waterfall* harus diselesaikan terlebih dahulu sebelum melanjutkan ke fase selanjutnya. Artinya fokus terhadap masing-masing fase dapat dilakukan maksimal karena jarang adanya pengerjaan yang sifatnya parallel walaupun dapat saja terjadi pararelisme dalam *waterfall*. Tahapan-tahapan dari metode *waterfall* adalah sebagai berikut:

1 ***Requirement Analysis***

Seluruh kebutuhan *software* harus bisa didapatkan dalam fase ini, termasuk didalamnya kegunaan *software* yang diharapkan pengguna dan batasan *software*. Informasi ini biasanya dapat diperoleh melalui wawancara, survey atau diskusi. Informasi tersebut dianalisis untuk mendapatkan dokumentasi kebutuhan pengguna untuk digunakan pada tahap selanjutnya.

2 *System Design*

Tahap ini bertujuan untuk memberikan gambaran apa yang seharusnya dikerjakan dan bagaimana hasil dari tampilannya. Tahap ini membantu dalam menspesifikasikan kebutuhan *hardware* dan sistem serta mendefinisikan arsitektur sistem secara keseluruhan.

3 *Implementation*

Dalam tahap ini dilakukan pemrograman. Pembuatan *software* dipecah menjadi modul-modul kecil yang nantinya akan digabungkan dalam tahap berikutnya. Selain itu dalam tahap ini juga dilakukan pemeriksaan terhadap modul yang dibuat, apakah sudah memenuhi fungsi yang diinginkan atau belum.

4 *Integration and Testing*

Di tahap ini dilakukan penggabungan modul-modul yang sudah dibuat dan dilakukan pengujian ini dilakukan untuk mengetahui apakah *software* yang dibuat telah sesuai dengan desainnya dan masih terdapat kesalahan atau tidak.

5 *Operation and Maintenance*

Tahap ini merupakan akhir dalam model *waterfall*. *Software* yang sudah jadi dijalankan serta dilakukan pemeliharaan. Pemeliharaan ini termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya. Perbaikan implementasi pada unit sistem dan peningkatan jasa sistem sebagai kebutuhan baru.

2.2.14 Alat Bantu Pengembangan Sistem




Dalam mengembangkan suatu sistem, pemrogram sistem perlu membuat visualisasi dari rancangan tersebut. Perlunya pembuatan visualisasi dikarenakan


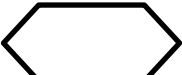
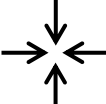
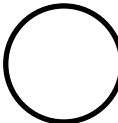

akan sulit sekali mengawasi apakah suatu sistem telah bekerja dengan baik tanpa adanya cetak biru yang bisa dilihat dari sistem tersebut. Karenanya diperlukan alat bantu pengembangan sistem dalam membuat visualisasi dari sistem.

2.2.14.1 Flowchart

Menurut Indra Yatini (2010), *flowchart* adalah diagram yang menunjukkan alur data melalui program atau sistem penanganan informasi dan operasi – operasi yang dikenakan pada data pada titik – titik yang penting disepanjang jalur. *Flowchart* menggunakan anotasi dan lambing, misalnya segi empat, belah ketupat dan oval untuk menyatakan sebagai operasi. Garis dan ujung panah menghubungkan lambing – lambing tersebut untuk menunjukkan arah arus data dari satu titik ke titik lain. Sebagai diagram grafis yang menunjukkan program atau sistem lainnya, *flowchart* berguna sebagai sarana pembantu untuk menunjukkan bagaimana bekerjanya program yang diusulkan dan sebagai sarana untuk memahami operasi – operasi sebuah program.

Tabel 2.5 Flowchart

SIMBOL	NAMA SIMBOL	KETERANGAN
	Terminator	Digunakan untuk menandai awal dan akhir suatu <i>flowchart</i> , simbol ini biasanya diberi label mulai dan selesai.
	<i>Input / Output</i>	Digunakan untuk mempresentasikan fungsi I/O yang membuat sebuah data dapat dimasukan (<i>input</i>) atau ditampilkan (<i>output</i>)
	Percabangan / Keputusan	Digunakan untuk melakukan percabangan, yaitu pemeriksaan terhadap suatu kondisi.

	Proses / Penugasan	Digunakan untuk melakukan proses dari suatu <i>input</i> atau <i>output</i> maupun operasi lainnya.
	<i>Preparation</i>	Digunakan untuk memproses inisialisasi/pemberian harga awal (Misalnya dalam melakukan iterasi)
	Arah aliran	Digunakan untuk menghubungkan setiap langkah dalam <i>flowchart</i> dan menunjukkan arah aliran diagram
	Konektor <i>On Page</i>	Digunakan untuk menghubungkan satu langkah dengan langkah lain dalam <i>flowchart</i> dengan keadaan <i>on page</i> . <i>On page</i> digunakan untuk menghubungkan satu langkah dengan langkah lain dalam satu halaman.
	Konektor <i>Off Page</i>	Digunakan untuk menghubungkan suatu langkah dengan langkah lain dalam halaman yang berbeda.

Sumber : Indra Yatini, 2010 Flowchart, Algoritma dan Pemrograman

Flowchart (bagian alir) merupakan bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem. Bagan ini menjelaskan urutan dari prosedur-prosedur yang ada didalam sistem. Bagan alir sistem menunjukkan apa yang dikerjakan di sistem. Bagan alir sistem digambarkan dengan menggunakan simbol-simbol yang telah ditentukan.

Flowchart banyak digunakan untuk menghubungkan struktur menyeluruh dan aliran sistem ke pengguna akhir. Karena *flowchart* dapat menawarkan tampilan fisik yang berperan penting dalam keterkaitan *hardware* dan data media. *Flowchart* merupakan diagram alir yang menggambarkan suatu urutan logika dari suatu prosedur pemecahan masalah.

Beberapa kegunaan *flowchart*, yaitu :

1. *Flowchart* membantu (mempermudah) programmer dalam mendesain program, sebagai alat verifikasi dan sekaligus untuk dokumentasi program.
2. Dalam proses mendesain, *flowchart* dapat membantu memecahkan persoalan yang cukup kompleks kedalam serangkaian instruksi.
3. Dalam proses verifikasi, *flowchart* lebih mudah diperiksa oleh seorang *quality control* (QC) daripada langsung memeriksa *source code* (instruksi – instruksi) program, atau *flowchart* dapat mempermudah pekerjaan QC tersebut dalam pemeriksaan kualitas program.
4. *Flowchart* dapat digunakan sebagai dokumen spesifikasi proses dalam pembuatan *Data Flow Diagram*.

2.2.14.2 Blok Diagram

Diagram dari sebuah sistem, dimana bagian utama atau fungsi yang diwakili oleh blok dihubungkan dengan garis, yang menunjukkan hubungan dari blok. Banyak dalam dunia rekayasa dalam desain *hardware*, desain elektronik, *software* desain dan proses aliran diagram.

2.2.15 Metode Pengujian

Pengujian sistem merupakan bagian yang sangat penting dalam siklus pembangunan perangkat keras. Pengujian dilakukan untuk menjamin kualitas dan juga mengetahui kelemahan dari perangkat keras. Pengujian sistem dilakukan untuk mengetahui efektifitas dari sistem yang digunakan selain memberikan kesempatan kepada *user* untuk mengoperasikan dan melakukan pengecekan terhadap laporan yang dihasilkan melalui sistem yang telah dibuat.

2.2.15.1 Pengujian *Black Box*

Menurut Pressman (2012), pengujian *black-box* disebut juga uji perilaku, berfokus pada syarat fungsional perangkat lunak. Artinya, teknik pengujian *black-box* memungkinkan seseorang untuk memperoleh serangkaian kondisi masukan yang akan sepenuhnya menjalankan semua persyaratan fungsional untuk sebuah program. Pengujian *black-box* bukanlah alternatif dari metode *white-box*. Sebaliknya, metode ini adalah sebagai pendekatan pelengkap untuk menemukan sebuah perbedaan kelas dari kesalahan – kesalahan pada metode *white-box*.

Pengujian *black-box* bertujuan untuk menemukan kesalahan pada kategori berikut :

1. Fungsi yang salah atau tidak benar.
2. Kesalahan tampilan antarmuka
3. Kesalahan pada struktur data atau akses database eksternal
4. Kesalahan perilaku atau kinerja
5. Kesalahan inisialisasi dan penghentian.

2.2.15.2 Pengujian *White Box*

Menurut Pressman (2012), pengujian *white-box*, adalah metode uji coba desain yang menggunakan struktur kontrol desain prosedural yang digambarkan sebagai bagian dari desain tingkat-komponen untuk mendapatkan kasus uji. Dengan menggunakan metode pengujian *white-box*, perancang sistem dapat melakukan *test case* untuk memberikan jaminan bahwa :

1. Semua jalur independen pada suatu modul ditelusuri minimal 1 kali
2. Melaksanakan semua keputusan logis pada sisi benar dan salah.

3. Semua *loop* dieksekusi pada batas yang tercantum dan batas operasionalnya.
4. Struktur data internal digunakan agar validitas terjamin.

BAB III

METODE PENELITIAN

Metode penelitian adalah rangkaian cara terstruktur atau sistematis yang digunakan oleh para peneliti dengan tujuan mendapatkan jawaban yang tepat atas apa yang menjadi pertanyaan pada objek penelitian. Atau secara mudahnya arti metode penelitian adalah upaya untuk mengetahui sesuatu dengan rangkaian sistematis. Adapun penelitian yang penulis lakukan melalui beberapa tahapan sebagai berikut:

3.1 Tempat dan Waktu Penelitian

Kegiatan penelitian ini dilakukan di STMIK Widya Cipta Dharma Samarinda. Penelitian ini dilaksanakan selama satu bulan dimulai tanggal 1 Februari 2019 s/d 1 Maret 2019. Pengumpulan data dan literatur – literatur yang berhubungan dengan program dilaksanakan di STMIK Widya Cipta Dharma Samarinda.

3.2 Teknik Pengumpulan Data

Untuk menjelaskan dan memudahkan dalam rangkaian penelitian menghindari terjadinya kesalahan, maka metode penelitian yang diterapkan peneliti didasarkan kepada metode – metode penelitian yang sudah umum, yaitu :

3.2.1 Observasi (Pengamatan Langsung)

Adapun observasi atau pengamatan dilakukan langsung di STMIK Widya Cipta Dharma, dimana peneliti turun langsung untuk mengamati dan menganalisa

suatu rancangan juga peralatan yang digunakan dalam menghasilkan model atau desain

3.2.2 Studi Pustaka

Dalam melakukan penelitian menggunakan literatur – literatur yang berhubungan berkenaan dengan materi penelitian. Data yang diperoleh berupa konsep atau teori – teori yang dapat menunjang penelitian dan untuk penulisan laporan, sehingga penelitian dan penulisan skripsi ini tidak menyimpang dari ketentuan yang ada. Pengolahan data yang dapat digunakan dan diketahui secara jelas dan benar. Pada akhirnya semua data yang diperoleh dapat digunakan untuk mendukung pengembangan alat.

3.3 Metode Pengembangan

Pengembangan sistem pada pembuatan alat ini menggunakan metode *Waterfall* (Model Air Terjun). Metode *Waterfall* (Model Air Terjun) menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian dan tahap pendukung (*support*).

3.3.1 Analisis

Untuk memulai pembuatan suatu sistem yang baru, harus dimulai dari awal dengan cara mengumpulkan informasi yang dibutuhkan. Maka analisis adalah suatu hal yang perlu dilakukan pada awal pembuatan sistem.

3.3.1.1 Analisis Data

Analisis data adalah rangkaian kegiatan dalam mengelompokan data berdasarkan variabel dan jenis responden, data di kelompokkan berdasarkan

variabel dan seluruh responden, menyajikan data tiap variabel yang diteliti, melakukan perhitungan untuk menjawab rumusan masalah.

3.3.1.2 Analisis User

Analisis *user* untuk menentukan kepada siapa sistem ditujukan dan kebutuhan apa yang user butuhkan.

3.3.1.3 Analisis Sistem

Analisis sistem adalah analisis mengenai segala macam hal yang sekiranya akan memengaruhi sistem secara keseluruhan, baik itu faktor *internal* maupun *eksternal* yang dapat menjadi kendala pada alat sistem kontrol ini.

3.3.1.4 Analisis Teknologi

Analisis teknologi ini bertujuan untuk menganalisa kebutuhan teknologi yang akan dipakai pada pembuatan *prototype* pengendali pintu ini, adapun analisis teknologi ini meliputi teknologi perangkat keras dan teknologi perangkat lunak serta tambahan alat lainnya yang dibutuhkan pada saat pembuatan sistem.

3.3.2 Desain (Perancangan)

Analisis desain sistem bertujuan untuk memberikan gambaran tentang sistem yang diusulkan. Alat bantu yang dipakai dalam desain sistem ini adalah :

1. Blok Diagram.
2. *Flowchart* / bagan alir.

3.3.3 Implementasi

Setelah melakukan analisa dan desain, maka dilakukan implementasi yaitu pembangunan sistem. Dalam hal ini difokuskan pada pembuatan program dan penggunaannya. Adapun bagian – bagian yang dibangun dalam sistem ini adalah :

1. *Prototype.*
2. Program.

3.3.4 Pengujian (*Testing*)

Pada tahap ini yaitu pengujian sistem (*testing*) melakukan pengujian sistem yang telah dibangun. Pengujian ditujukan untuk menemukan kesalahan – kesalahan pada sistem dan memastikan sistem yang telah dibangun sesuai dengan apa yang direncanakan sebelumnya. Dalam membangun sistem pendeteksi benda. Metode yang digunakan dalam pengujian sistem ini yaitu :

1. *Black Box Testing*

Black box testing menyinggung uji coba yang dilakukan pada *interface software*. Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak. Pengujian ini memungkinkan analisis sistem memperoleh kumpulan kondisi *input* yang akan mengerjakan seluruh keperluan fungsional program. Untuk mempermudah dalam proses pengujian maka perlu dibuatnya suatu tabel pengujian yang digunakan sebagai acuan dalam pengujian sistem tersebut.

2. *White Box Testing*

White box metode pengujian dengan cara melihat kedalam modul program untuk meneliti kode program yang ada. Jika ada modul yang tidak menghasilkan *output* yang tidak sesuai dengan yang diharapkan, maka baris – baris program tersebut akan diperiksa dan diperbaiki.

3.3.5 *Maintenance*

Maintenance yaitu mengoperasikan program di lingkungannya dan melakukan pemeliharaan. Biasanya ini merupakan fase siklus hidup yang paling

lama. Pemeliharaan mencakup koreksi dari berbagai *error* yang tidak ditemukan pada tahap-tahap sebelumnya, melakukan perbaikan atas implementasi unit sistem dan pengembangan layanan sistem.

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

4.1 Hasil Penelitian

4.1.1 Gambaran Umum

Prototype pengendali pintu otomatis ini dibangun untuk mengendalikan pintu otomatis, dengan *input* berupa *keypad* dan RFID serta hasil keluaran *buzzer* dan LCD 16x2, pengendali pintu tersebut akan membuka serta menutup pintu dengan servo motor.

Dengan menggunakan kemajuan teknologi, proses pengendali pintu yang awalnya dilakukan secara manual dapat diganti dengan alat digital, yakni dengan menaruh keypad dan RFID, lalu mikrokontroler akan memproses validasi inputan pada sistem minimum Arduino Mega. Apabila *input* yang diberikan pengguna benar maka servo akan membuka atau menutup pintu dan *buzzer* akan berbunyi sekali, sebaliknya apabila pengguna salah maka servo tidak akan bergerak dan *buzzer* akan berbunyi peringatan.

4.2 Pembahasan

4.2.1 Analisis

Analisis merupakan teknik pemecahan masalah yang menguraikan bagian komponen dengan mempelajari seberapa bagus bagian komponen tersebut bekerja

4.2.1.1 Analisis Data

Analisis data yang digunakan dalam membangun sistem ini agar dapat berjalan antara lain :

1. Datasheet Arduino Mega 2560.
2. Datasheet *Keypad Matrix Membrane* 4x4
3. Datasheet MFRC522
4. Datasheet LCD (*Liquid Crystal Display*) 16x2
5. Datasheet Motor Servo SG90

4.2.1.2 Analisis User

Pada analisis user tahapan yang dilakukan adalah menentukan siapa yang akan menggunakan sistem ini. Target pengguna adalah :

1. Pengguna yang akan membuat alat pengendali pintu garasi, brankas dan palang pintu parkir sederhana.
2. Mahasiswa yang akan melakukan penelitian yang berkaitan dengan RFID dan *keypad*.

4.2.1.3 Analisis Sistem

Pada analisis sistem terdapat beberapa kondisi khusus yang melengkapi alat pengendali pintu ini antara lain :

1. *Input password* pada *keypad* hanya berupa 4 digit angka (*numeric*).
2. Pada MFRC522 maksimum tegangan yang direkomendasikan adalah sebesar 3.3V dan frekuensi RFID *label* yang dapat terdeteksi pada 13,56 MHz.
3. Alat pengendali pintu ini mengeluarkan bunyi berupa *buzzer* apabila validasi password atau RFID pengguna cocok dan *buzzer* berupa peringatan apabila validasi pengguna salah.

Penelitian RFID dilakukan dengan mengukur terbacanya sensor dengan halangan maupun tanpa halangan. Tabel pengujian dibawah ini menunjukkan hasil pengujian sensor RFID.

Tabel 4.1 Hasil Pengujian Sensor MFRC522 Tanpa Halangan

Uji Coba	Jarak dengan sensor (cm)	Status sensor
1	0.5	Terbaca
2	1	Terbaca
3	1.5	Terbaca
4	2	Terbaca
5	2.5	Terbaca
6	3	Terbaca
7	3.5	Tidak Terbaca

Pada tabel 4.1 jarak maksimal yang dapat terbaca oleh sensor tanpa halangan adalah 3 cm. Selebihnya, sensor tidak dapat membaca.

Tabel 4.2 Hasil Pengujian Sensor MFRC522 Dengan Halangan

Uji Coba	Jarak dengan sensor (cm)	Status sensor
1	0.5	Terbaca
2	1	Terbaca
3	1.5	Terbaca
4	2	Terbaca
5	2.5	Tidak Terbaca
6	3	Tidak Terbaca
7	3.5	Tidak Terbaca

Sedangkan pada tabel 4.2 jarak maksimal yang dapat terbaca oleh sensor dengan halangan adalah 2 cm. Selebihnya, sensor tidak dapat membaca.

Kesimpulan dari *table* diatas adalah jarak maksimum sensor RFID dapat membaca kartu tanpa halangan 3 Cm sedangkan untuk membaca kartu dengan halangan berkisar 1-5 Cm sampai dengan 2 Cm

4.2.1.4 Analisis Teknologi

Analisis teknologi yang dibutuhkan untuk pengendali pintu otomatis antara lain :

1. Perangkat Keras
 - 1) Arduino Mega R3 2560
 - 2) MFRC522
 - 3) Membrane Matrix Keypad 4x4
 - 4) LCD 16x2 dengan Serial I2C
 - 5) Motor Servo SG90
 - 6) *Buzzer*
 - 7) Breadboard
2. Perangkat Lunak
 - 1) Sistem Operasi Windows 7 32-bit
 - 2) Arduino Development Environment
 - 3) Fritzing

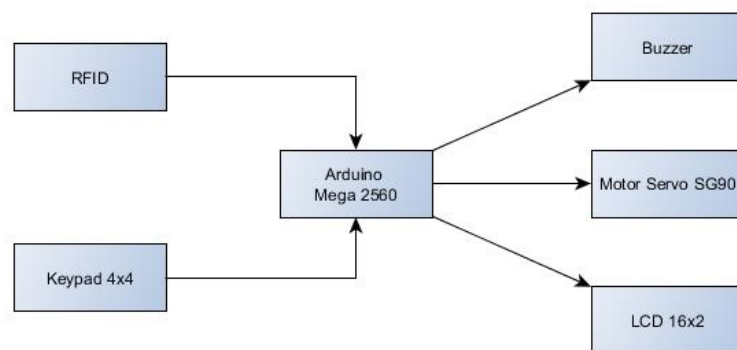
4.2.2 Desain

Desain merupakan sebuah proses untuk mendefinisikan sesuatu yang akan dikerjakan dengan menggunakan teknik yang bervariasi serta didalamnya melibatkan deskripsi mengenai arsitektur serta detail komponen dan juga keterbatasan yang akan dialami proses pengerjaannya.

4.2.2.1 Desain Perangkat Keras

Perancangan alat merupakan langkah yang amat penting dalam merealisasikan sistem yang sesungguhnya. Tujuan perancangan ini adalah untuk merencanakan perangkat keras (*hardware*) sesuai dengan spesifikasi dan cara kerja dari sistem yang hendak dibuat, sehingga diharapkan dapat mengefisienkan waktu, biaya dan tenaga.

1. Blok Diagram



Gambar 4.1 Blok Diagram

Untuk membuat pengendali pintu otomatis, pertama perlu menghubungkan RFID *reader* dan *keypad* sebagai media *input* dari pengguna ke mikrokontroler, kemudian memasang LCD 16x2 dengan Serial I2C sebagai tampilan dari *input* pengguna. Motor servo akan mengendalikan pintu untuk dibuka sampai sudut yang telah ditentukan. Setelah pintu terbuka maka *buzzer* akan mengeluarkan bunyi dan menampilkan pesan di LCD 16x2. Fungsi setiap alat adalah :

1. Arduino Mega 2560 : Sebagai mikrokontroler dan sebagai pusat mengolah data yang keluar dan masuk secara bersamaan.
2. RFID MFRC522 : Sensor yang dapat mengidentifikasi seseorang atau obyek dengan menggunakan frekuensi transmisi radio.

3. Keypad Membrane Matrix 4x4 : Alat yang berfungsi untuk menerima masukkan karakter dari pengguna.
4. Servo Motor : Alat yang berfungsi untuk menentukan posisi sudut dari pintu, servo motor disini berfungsi untuk mengendalikan pintu sesuai dengan *output* yang ditentukan.
5. Buzzer : Alat ini berfungsi sebagai komponen elektronika yang dapat mengubah sinyal listrik menjadi getaran suara.
6. LCD 16x2 dengan Serial I2C : Sebagai media *output* untuk menampilkan hasil dari setiap proses yang sedang berjalan

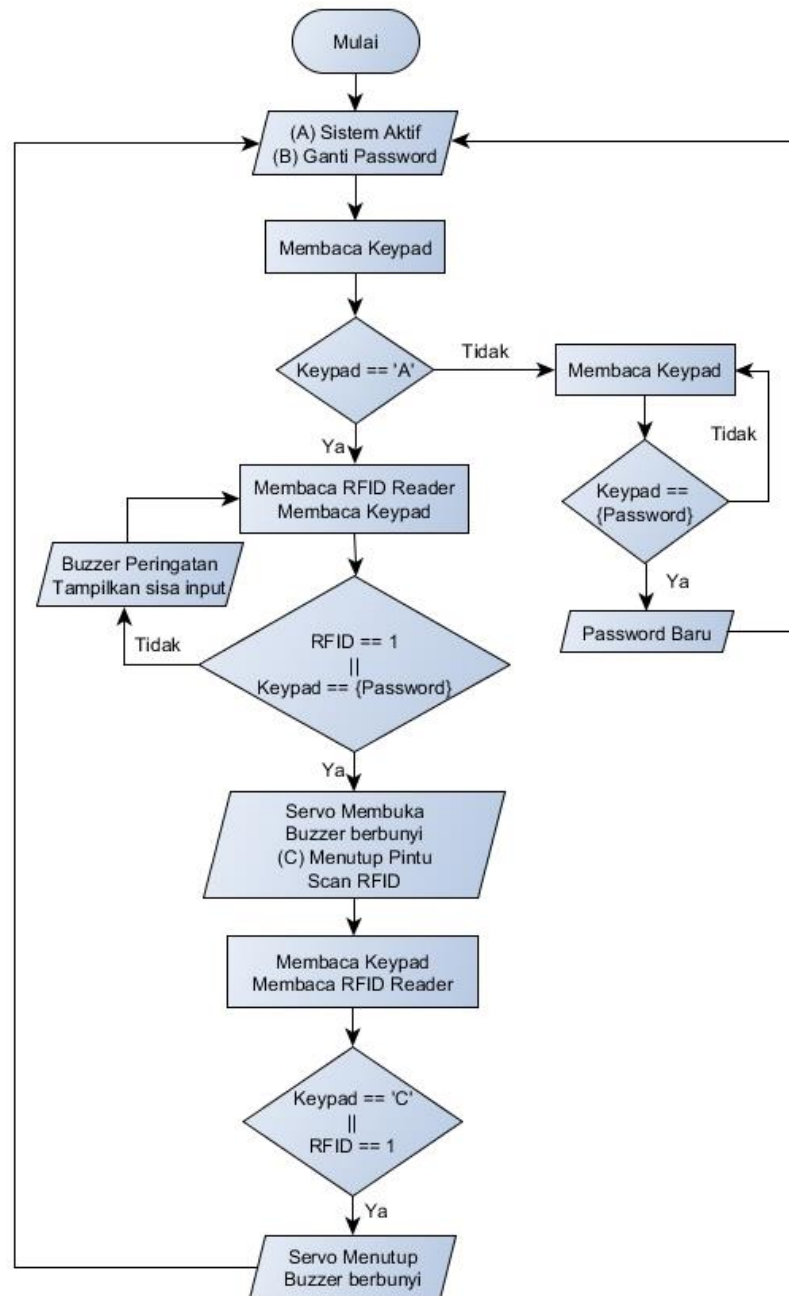
4.2.2.2 Desain Perangkat Lunak

Desain perangkat lunak merupakan sebuah proses untuk mendefinisikan sesuatu yang akan dikerjakan dengan menggunakan teknik yang bervariasi serta didalamnya melibatkan deskripsi mengenai arsitektur serta detail komponen dan juga keterbatasan yang akan dialami dalam proses pengerjaannya.

1. *Flowchart* Sistem

Flowchart sistem pada pengendali pintu otomatis hal pertama adalah menampilkan menu pilihan awal untuk mengaktifkan sistem atau mengganti *password*. Apabila pengguna menekan *keypad* (A) maka sistem akan aktif dan sistem akan menunggu data masukkan pengguna dari *keypad* atau kartu dari RFID, apabila *password* atau kartu tidak sesuai, maka sistem akan mengeluarkan bunyi peringatan dari *buzzer*, sebaliknya jika data sesuai maka *buzzer* akan berbunyi, servo akan membuka pintu sampai terbuka dan LCD akan menampilkan pilihan untuk menutup pintu. Jika *input* pengguna adalah (C) atau RFID cocok

maka servo akan menutup pintu, *buzzer* berbunyi dan LCD akan menampilkan menu awal. Apabila pengguna memilih menu (B) ganti *password*, maka pengguna wajib masukkan *password* lama dan mengganti dengan *password* yang baru, apabila ganti *password* berhasil maka sistem akan kembali ke menu awal.



Gambar 4.2 *Flowchart* Sistem

4.2.3 Implementasi

Setelah melakukan analisa dan desain, maka dilakukan implementasi. Dalam hal ini difokuskan pada pembuatan program dan penggunaannya.

4.2.3.1 Perangkat Keras

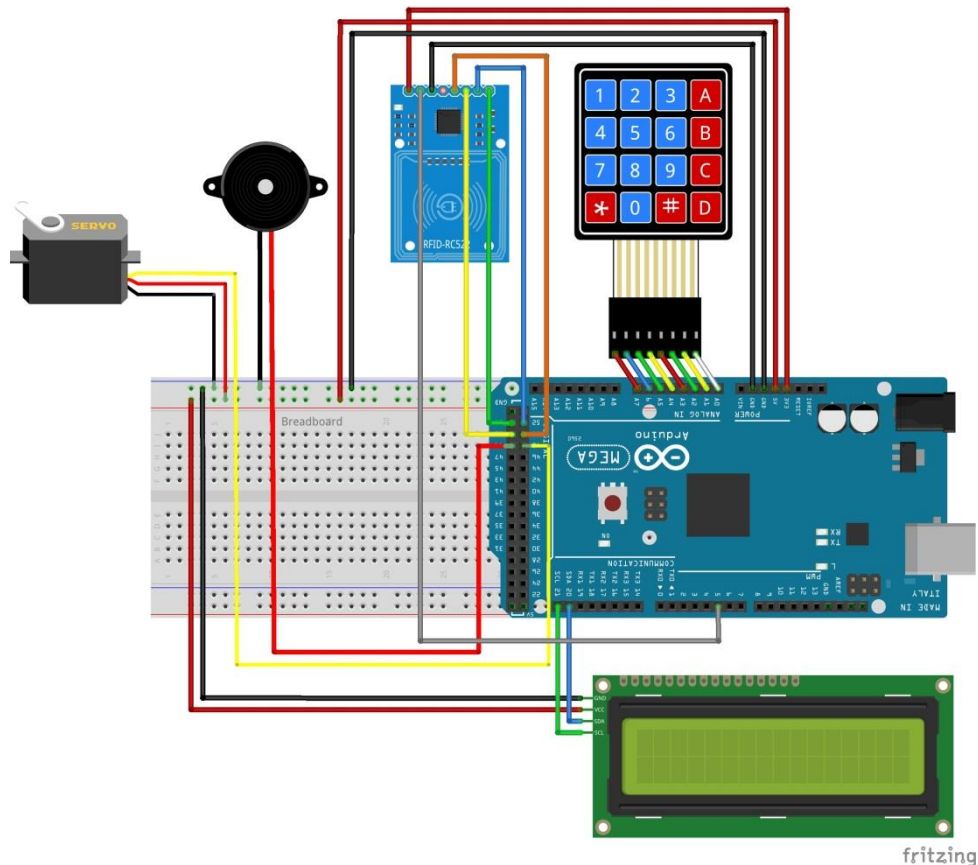
Perangkat keras yang digunakan pada alat pengendali pintu ini adalah Arduino Mega R3, RFID MFRC522, Keypad membrane matrix 4x4, motor *servo* SG90, *buzzer* dan LCD 16x2 dengan Serial I2C.



Gambar 4.3 Desain Pengendali Pintu Otomatis

Arduino Mega R3 sebagai perangkat untuk program yang telah dibuat, RFID MFRC 522 sebagai sensor identifikasi kartu menggunakan frekuensi radio, Keypad membrane matrix 4x4 sebagai masukkan *password*, Motor servo sebagai penggerak pintu otomatis, LCD 16x2 dengan serial I2C sebagai *output* dari proses

yang berjalan serta *buzzer* sebagai bunyi peringatan dan komponen - komponen lain yang digunakan.



Gambar 4.4 Komponen Alat Pengendali Pintu Otomatis

Pada gambar 4.4 Dapat dijelaskan komponen – komponen yang terdapat pada *prototype* pengendali pintu Garasi ini yang terdiri dari :

1. Sistem Minimum Arduino Mega

Pada blok ini, mikrokontroler yang digunakan adalah Arduino Mega R3 yang menggunakan mikrokontroler ATmega 2560. Arduino Mega ini memiliki 54 pin digital (15 pin dapat digunakan sebagai *output* PWM), 16 *input analog*. Dengan memori *flash* sebesar 256 Kb (8 Kb digunakan sebagai *bootloader*). Adapun pin yang digunakan untuk membangun sistem pengendali pintu otomatis adalah sebagai berikut :

Pin 49 : digunakan untuk *pin buzzer*

Pin 48 : digunakan untuk *pin servo motor*

Pin 5 : digunakan untuk *pin RST* pada MFRC522

Pin 20 : digunakan untuk *pin serial komunikasi I2C SDA* pada LCD 16x2

Pin 21 : digunakan untuk *pin serial komunikasi I2C SCL* pada LCD 16x2

Pin 50 : digunakan untuk *pin serial komunikasi SPI MISO* pada MFRC522

Pin 51 : digunakan untuk *pin serial komunikasi SPI MOSI* pada MFRC522

Pin 52 : digunakan untuk *pin serial komunikasi SPI SCK* pada MFRC522

Pin 53 : digunakan untuk *pin serial komunikasi SPI SDA/SS* pada MFRC522

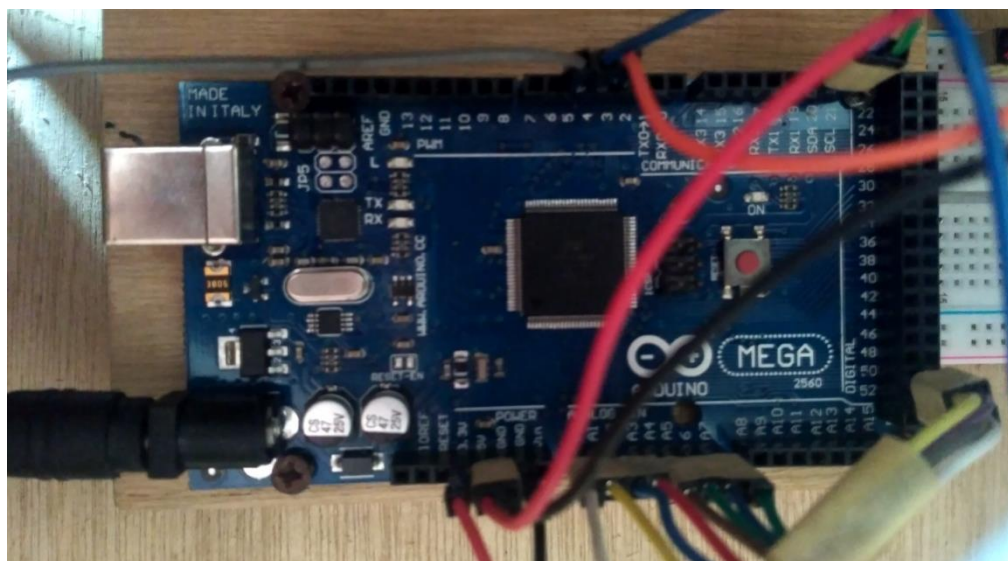
Analog Pin A0 - A3 : digunakan untuk *pin kolom Keypad Membran Matrix 4x4.*

Analog Pin A4 - A7 : digunakan untuk *pin baris Keypad Membran Matrix 4x4.*

Pin Ground : digunakan untuk ground sensor pada Arduino Mega.

Pin 3.3V : digunakan untuk memberikan tegangan listrik 3.3V pada MFRC 522

Pin 5V : digunakan untuk memberikan tegangan listrik 5V pada sensor.



Gambar 4.5 Arduino Mega

2. *Keypad Membrane Matrix 4x4*

Keypad Membrane Matrix 4x4 yang digunakan memiliki 8 buah pin *input* 4 pin pertama untuk baris dan 4 pin selanjutnya digunakan untuk kolom. Memiliki 16 buah *input* yang terdiri dari angka numerik 0 – 9, serta karakter *, #, A, B, C dan D.



Gambar 4.6 *Keypad Membrane Matrix 4x4*

3. **RFID MFRC522**

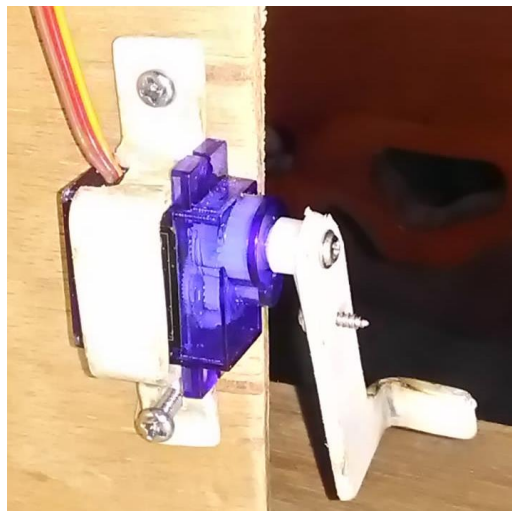
RFID MFRC522 mempunyai tegangan maksimal yang disarankan sebesar 3.3V. MFRC522 menggunakan serial SPI sebagai serial komunikasi antara sensor dengan Arduino Mega dengan pin 50 terhubung dengan MISO Arduino, pin 51 terhubung dengan MOSI Arduino, pin 52 terhubung dengan SCK Arduino, pin 53 terhubung dengan SDA/SS Arduino Mega, sedangkan untuk Reset pin terhubung dengan pin 5 pada Arduino, pin ground dengan pin 3.3V terhubung langsung dengan pin tegangan pada Arduino.



Gambar 4.7 RFID MFRC522

4. Servo Motor SG90

Servo motor yang digunakan adalah motor servo jenis SG90. Motor servo SG90 bergerak dari 0° sampai dengan 180° . Putaran motor servo ini diatur dari program untuk dapat mengendalikan pintu sesuai dengan perintah pengguna untuk mengaktifkan atau menutup sistem. Pada prototipe ini motor servo diletakkan disamping dinding menyatu dengan kayu triplek sebagai alas. Kabel logika servo dihubungkan dengan pin 4 pada Arduino Mega, sedangkan pin *ground* dengan pin 5V terhubung langsung dengan kabel jumper pada papan *breadboard*.



Gambar 4.8 Servo Motor SG90

5. LCD 16x2 dengan Serial I2C

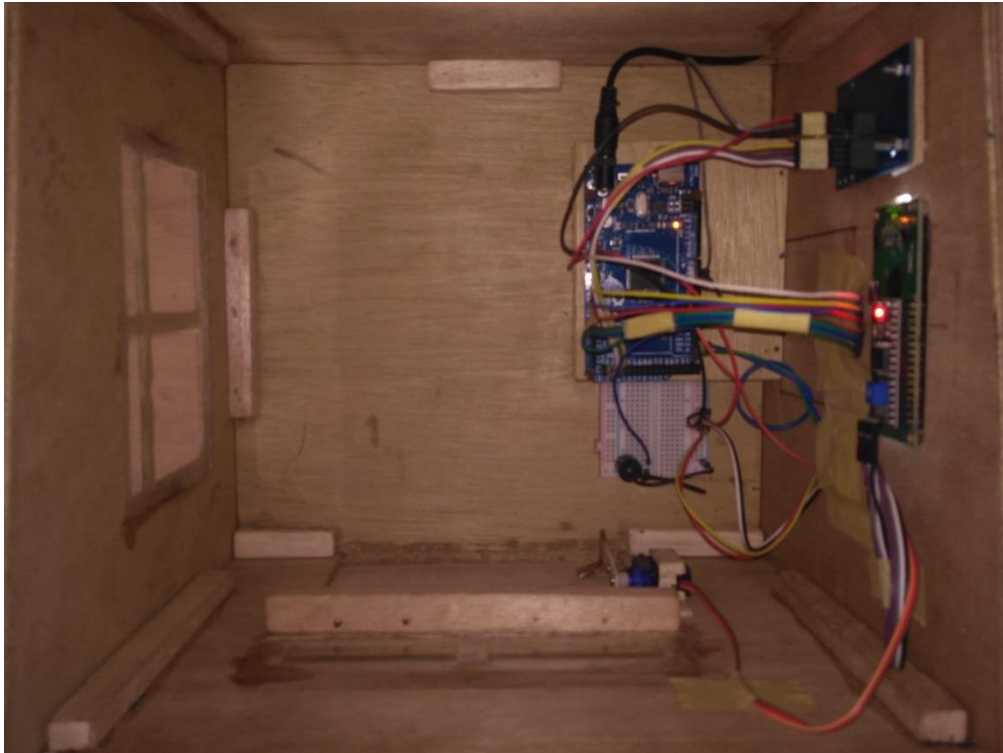
LCD 16x2 yang digunakan pada prototipe ini sudah disolder dengan I2C untuk dapat berkomunikasi dengan Arduino Mega. Penggunaan serial I2C dapat menghemat penggunaan kabel jumper lebih sedikit daripada tidak menggunakan serial I2C. LCD 16x2 dengan serial I2C menghubungkan serial pin SDA dengan digital pin 20 pada Arduino, serial pin SCL terhubung dengan digital pin 21 pada Arduino, sedangkan pin ground dengan pin 5V terhubung langsung dengan kabel jumper pada papan *breadboard*.



Gambar 4.9 LCD 16x2 dengan Serial I2C

6. Breadboard

Breadboard yang digunakan pada prototipe pengendali pintu adalah *breadboard* 830 point, *breadboard* dipasang rata dengan lantai. *Breadboard* untuk catu daya tegangan terhubung langsung dengan kabel jumper dengan pin 5V pada Arduino Mega sedangkan untuk catu daya pin ground *breadboard* terhubung langsung dengan kabel jumper dengan pin *ground* pada Arduino Mega. Arduino Mega dipasang diatas papan *breadboard*. Untuk pemasangan *breadboard* dapat dilihat pada gambar 4.9.



Gambar 4. 10 Susunan komponen pada *Breadboard*

Dapat dilihat pada gambar diatas setiap sensor dan komponen penghubung lainnya terhubung langsung dengan papan *breadboard* dengan menggunakan kabel jumper. Sehingga penggunaan papan *breadboard* pada prototipe ini lebih ringkas dan memudahkan pengembang untuk melakukan perawatan/*maintenance* jika terjadi pengembangan prototipe.

4.2.3.2 Perangkat Lunak

Pada pembahasan ini akan dibahas tentang simulasi alat dan pembuatan program menggunakan aplikasi dari Arduino Uno yaitu *Arduino Development Environment* yang dapat di download langsung dari situs arduino itu sendiri. Dalam aplikasi *Arduino Development Environment* terdapat *sketch* untuk mengetikkan program dan *tool-tool* seperti *verify* atau *compile*, *upload*, *new*, *open* dan *save project*.

4.2.4 *Testing*

Pengujian atau *testing* dilakukan untuk mengetahui apakah sistem telah bekerja dengan baik sesuai dengan apa yang diinginkan. Pengujian pertama yang dilakukan adalah dengan pengujian secara bertahap agar didapatkan hasil yang diharapkan dan yang terakhir pengujian dengan menggunakan metode *black box* dan *white box*. Berikut beberapa pengujian bertahap:

1. **Pengujian Tahap Pertama**, pada pengujian pertama dilakukan uji coba rangkaian dan program sebanyak 10 kali pada RFID MFRC522, tetapi RFID MFRC522 dapat bekerja pada tegangan 3.3V serta mengalami kendala pada kabel jumper dan dilakukan perbaikan pada rangkaian serta sintak program. Setelah dilakukan penggantian kabel jumper dan perbaikan program, RFID MFRC522 dapat bekerja dengan baik.
2. **Pengujian Tahap Kedua**, pengujian kedua dilakukan pada *keypad* membrane matrix 4x4 sebanyak 10 kali. *Keypad* tidak dapat membaca masukan dengan baik pada awalnya. Kemudian dilakukan pengecekan pada kabel jumper dan sambungan kabel, setelah dicek dan menempatkan pin - pin sesuai dengan posisi pada sintak program, *keypad* dapat membaca masukan dari pengguna dengan baik.
3. **Pengujian Tahap Ketiga**, dilakukan uji coba ketiga pada LCD 16x2 dengan serial I2C sebanyak 10 kali. Pada tahap awal pemasangan LCD 16x2 terhubung langsung dengan Arduino Mega, tetapi penggunaan kabel jumper yang terlalu banyak dan boros tempat akhirnya ditambah Serial I2C yang hanya menggunakan 4 kabel pin, dengan 2 pin sebagai serial

komunikasi. Akhirnya pemasangan LCD 16x2 dengan serial I2C dapat menampilkan teks dengan baik dan dapat menghemat lebih banyak ruang.

4. **Pengujian Tahap Keempat**, dilakukan pengujian keempat terhadap servo motor sebanyak 10 kali. Pengujian awalnya dilakukan untuk mengetahui apakah servo dapat mengangkat beban pada pintu *plywood*. Percobaan pertama servo motor dapat mengangkat beban pintu tapi masih terdapat *freezing* pada *gear* sehingga menyebabkan servo motor tidak dapat mengendalikan pintu secara maksimal sampai pada sudut yang telah ditentukan. Akhirnya desain pintu, posisi servo dan sintak pada program dilakukan perbaikan, sehingga servo dapat mengangkat pintu dengan baik.
5. **Pengujian Tahap Kelima**, dilakukan uji coba pada *buzzer* sebanyak 10 kali untuk mengetahui apakah buzzer dapat mengeluarkan suara sesuai dengan frekuensi suara yang ditentukan. Awalnya *buzzer* masih mengeluarkan suara beep yang tidak teratur kemudian dilakukan perbaikan pada sintak program. Buzzer dapat bekerja dengan baik.
6. **Pengujian Tahap Keenam**, dilakukan percobaan secara menyeluruh pada rangkaian alat dan program arduino untuk melihat kemungkinan adanya terjadi kesalahan pada sistem (*bug*). Dan diketahui saat LCD 16x2 menampilkan menu untuk mengganti password, password lama tidak dapat terganti. Kemudian menu untuk menutup kembali pintu tidak dapat berjalan. Akhirnya dilakukan perbaikan sintak program pada *software* Arduino dan perbaikan pada rangkaian alat. Kesalahan sistem tadi dapat diatasi dan berjalan dengan baik.

4.2.4.1 Pengujian *Black Box*

Hasil pengujian yang melakukan uji sistem pada keseluruhan alat pengendali pintu ini adalah *user*. Berikut tabel uji sistem yang di lakukan, antara lain :

Tabel 4.3 Pengujian *Black Box*

No	Uraian	Hasil yang diharapkan	Hasil Pengujian	Ya	Tidak	Jumlah Pengujian
1	Proses	Proses <i>input</i> dan <i>output</i> data dapat ditangani dengan baik oleh mikrokontroler	Data dari pengguna berhasil diproses dengan baik di Mikrokontroler	Ya	-	10
2	Keypad Membrane Matrix 4x4	Keypad dapat menerima input dari pengguna tanpa bermasalah	Keypad berhasil menerima input dari pengguna tetapi keypad sensitif terhadap pemasangan kabel jumper sehingga perlu di lem kertas dengan hati - hati	Ya	-	10
3	RFID MFRC522	RFID MFRC522 dapat membaca data dari pengguna tanpa mengalami masalah pada frekuensi radio	RFID MFRC522 dapat beroperasi pada tegangan 3.3V serta penggunaan kabel jumper juga mempengaruhi RFID dapat membaca data dengan baik.	Ya	-	10
4	Motor Servo SG90	Motor servo dapat berputar mengendalikan pintu tanpa bermasalah.	Motor servo tidak mengalami hambatan untuk mengendalikan pintu jika beban	Ya	-	10

			pada pintu masih tergolong ringan			
5	LCD 16x2 dengan Serial I2C	LCD 16x2 dapat beroperasi sesuai dengan yang diharapkan	Penggunaan serial I2C pada LCD 16x2 sangat membantu pengembang dalam melakukan perawatan atau perbaikan pada prototipe	Ya	-	10
6	Buzzer	Buzzer berbunyi saat merespon setiap proses data dari mikrokontroler	Buzzer dapat berbunyi dengan normal sesuai dengan yang diharapkan	Ya	-	10

4.2.4.2 Pengujian *White Box*

Pengujian pada alat pengendali pintu otomatis meliputi beberapa modul. Pengujian meliputi jenis pengujian, jenis program, hasil yang diharapkan dan hasil pengujian. Modul yang diuji adalah sebagai berikut :

Tabel 4.4 Pengujian *White Box*

No	Uraian	Kode program	Keterangan
1	Membaca kartu dengan UID Byte 118, 139, 18, 26 (kartu yang telah didaftarkan pada sistem)	<pre>#include <SPI.h> #include <MFRC522.h> #define RST 5 #define SS 53 Byte kartu[4]; MFRC522 rfid(SS, RST); SPI.begin(); rfid.PCD_Init(); kartu[0] = 118; kartu[1] = 139;</pre>	RFID MFRC522 terkoneksi pada Arduino Mega dan memvalidasi input dari pengguna

		<pre> kartu[2] = 18; kartu[3] = 26; if (! rfid.PICC_IsNewCardPresent()) return; if (! rfid.PICC_ReadCardSerial()) return; for (byte a = 0; a < 4; a++) tempKartu[a] = rfid.uid.uidByte[a]; if (kunciPintu) { if (memcmp(tempKartu, kartu, 4) == 0) cekValidasi(true); else cekValidasi(false); } else { if (memcmp(tempKartu, kartu, 4) == 0) { lcd.clear(); menu = 'C'; } } rfid.PICC_HaltA(); rfid.PCD_StopCrypto1(); </pre>	
2	Keypad dapat membaca dan mencocokkan password 1234	<pre> #include <Keypad.h> const byte Baris = 4; const byte Kolom = 4; char keyMap[Baris][Kolom] = { {'1', '2', '3', 'A'}, {'4', '5', '6', 'B'}, {'7', '8', '9', 'C'}, {'*', '0', '#', 'D'}}; byte pinBaris[Baris] = {A7, A6, A5, A4}; byte pinKolom[Kolom] = {A3, A2, A1, A0}; Keypad myKeypad = </pre>	Keypad terkoneksi pada Arduino Mega dan dapat membaca masukan dari pengguna.

		<pre>Keypad(makeKeypad(keyMap), pinBaris, pinKolom, Baris, Kolom); String password = "1234"; tombol = myKeypad.getKey(); if (tombol != NO_KEY) { if ((tombol == '0' tombol == '1' tombol == '2' tombol == '3' tombol == '4' tombol == '5' tombol == '6' tombol == '7' tombol == '8' tombol == '9') && sumbuX < 6) { bunyiKeypad(); lcd.setCursor(sumbuX, 1); lcd.print('*'); tempPass += tombol; sumbuX++; } Else if (tombol == '#' && sumbuX == 6) { if (tempPass == password) cekValidasi(true); else cekValidasi(false); } }</pre>	
3	Servo Motor bergerak dengan sudut 0° - 135°	<pre>#include <Servo.h> Servo myServo; myServo.attach(4); for (pos = 0; pos < 135; pos++) { myServo.write(pos); delay(25); } for (pos = 135; pos > 0; pos--) { myServo.write(pos); delay(25); }</pre>	Servo berputar sesuai dengan besaran yang ditetapkan
4	LCD 16x2 dengan serial I2C	<pre>#include <LiquidCrystal_I2C.h></pre>	LCD 16x2 dengan serial I2C terkoneksi

	menampilkan setiap proses dari mikrokontroler	<pre>LiquidCrystal_I2C lcd(0x27, 16, 2); lcd.begin(); lcd.clear(); lcd.setCursor(0, 0); lcd.print("Hallo world");</pre>	dengan baik dan dapat menampilkan setiap proses dari mikrokontroler dengan baik
5	Membaca Buzzer Input = 1000	<pre>#define buzzer 3 pinMode(buzzer, OUTPUT); tone(buzzer, 1000, 1000);</pre>	Buzzer dapat menyala dengan frekuensi sama dengan besaran input
6	Membaca logika program untuk mengaktifkan sistem atau mengganti password	<pre>if (hallo) { lcd.setCursor(0, 0); lcd.print("Memulai sistem"); tone(buzzer, 1500, 1250); delay(1500); hallo = false; lcd.clear(); } else if (!hallo) { if (menu == '\0') { tombol = myKeypad.getKey(); if (kunciPintu) { lcd.setCursor(0, 0); lcd.print("A:Aktif sistem"); lcd.setCursor(0, 1); lcd.print("B:Ganti password"); if (tombol != NO_KEY && tombol == 'A') { lcd.clear(); cdAlarm = true; menu = 'A'; } else if (tombol != NO_KEY && tombol == 'B') { lcd.clear(); mGanti = 0; menu = 'B'; } } } }</pre>	Berhasil membaca logika sistem untuk menampilkan menu mengaktifkan sistem dan mengganti password serta sub sistem di dalamnya.

		<pre> } else if (!kunciPintu) { lcd.setCursor(0, 0); lcd.print("C: Tutup pintu"); lcd.setCursor(0, 1); lcd.print("atau scan RFID"); inputRFID(); if (tombol != NO_KEY && tombol == 'C') { lcd.clear(); menu = 'C'; } } else if (menu == 'A') { AktifkanSistem(); } else if (menu == 'B') { gantiPassword(); } else if (menu == 'C') { KunciSistem(); } } </pre>	
--	--	--	--

4.2.5 Maintenance

Setelah tahap testing atau pengujian telah selesai dan prototipe yang dibuat berjalan dengan baik, maka prototipe selanjutnya akan digunakan pada pengguna. Akan dilakukan instalasi program dan pembelajaran kepada pengguna mengenai cara menggunakan prototipe yang telah dibuat. *Maintenance* akan dilakukan apabila ditemukan adanya kesalahan atau *bug* pada program, *upgrade* alat atau mikrokontroler dan *maintenance* dilakukan untuk menyesuaikan kebutuhan dari pengguna.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan uraian – uraian yang telah dijelaskan pada pembahasan sebelumnya dapat diambil kesimpulan sebagai berikut :

1. Untuk membuat *Prototype* pengendali pintu garasi menggunakan sistem minimum Arduino Mega 2560 ini dibutuhkan 6 komponen utama yaitu, Arduino Mega R3 sebagai mikrokontroler utama perangkat keras, RFID MFRC522 sebagai sensor pengenalan frekuensi radio, Keypad Membrane Matrix 4x4 sebagai memasukkan *password* pengguna, Servo Motor SG90 sebagai penggerak pintu, LCD 16x2 dengan Serial I2C sebagai *output* dari setiap proses yang berjalan dan Buzzer sebagai *output* bunyi terhadap setiap proses yang berjalan.
2. Berdasarkan hasil pengujian maka jarak maksimal yang RFID dapat membaca tanpa halangan adalah 3 cm, sedangkan jarak maksimal yang RFID dapat membaca dengan halangan adalah 2 cm.
3. Berdasarkan hasil dari pengujian yang telah dilakukan dari tahap pertama hingga tahap keenam dapat disimpulkan bahwa alat pengendali pintu ini dapat berfungsi dengan baik, dimana masing – masing alat dapat berfungsi sebagaimana mestinya.
4. Alat Sistem alat pengendali pintu yang dibuat memiliki 2 macam mode yaitu mode (A) Mengaktifkan Sistem untuk memulai sistem dari awal dan

mengunci pintu pada posisi awal dan mode (B) Mengganti Password pilihan pengguna untuk dapat mengganti *password keypad*, pengguna dapat mengganti *password* lama dengan *password* yang baru.

5.2 Saran

Adapun saran yang dapat dikemukakan adalah sebagai berikut :

1. Untuk mendapatkan hasil yang lebih baik, RFID reader dapat diganti dengan kualitas yang baik dan high-end.
2. Diharapkan agar sistem ini dapat dijadikan sebagai acuan atau referensi penelitian bagi mahasiswa dibidang informatika.
3. Diharapkan Sistem pengendali pintu otomatis ini bisa terus di kembangkan menjadi lebih kompleks dan lebih baik lagi.
4. Untuk pengembang selanjutnya agar dapat membuat Sistem pengendali pintu otomatis ini dengan *prototype* yang dapat membaca eKTP dan kartu identitas lainnya, serta dapat mengganti kartu RFID.
5. Untuk penelitian lebih lanjut diharapkan agar dapat mengembangkan alat ini dengan menambah alat yang baru dan lebih canggih.

Demikian kesimpulan dan saran yang dapat penulis kemukakan, semoga bermanfaat bagi semua pihak.

DAFTAR PUSTAKA

- Amin, Choirul. 2010. *101 Desain Pintu*. Jakarta: Griya Kreasi
- Dinata, Andi. 2018. *Fun Coding with Micropython*. Jakarta: PT. Elex Media Komputindo
- Effendi, Bachtiar. 2014. *Dasar Mikrokontroler ATmega8535 dengan CAVR*. Yogyakarta: Deepublish.
- Fransiska, Saragih Shella. 2017. *Rancang dan Bangun Sistem Kendali Bak Penampung Air dan Lampu Kamar Mandi Menggunakan Arduino Uno*
- Andrianto, Heri dan Aan Darmawan. 2016. *Arduino Belajar Cepat dan Pemrograman*. Bandung: Informatika.
- Kadir, Abdul. 2017. *Pemrograman Arduino dan Android Menggunakan App Inventor*. Jakarta: Elex Media Komputindo
- Nugroho, Heru. 2015. *Matematika Diskrit dan Implementasinya Dalam Dunia Teknologi Informasi*. Yogyakarta: Deepublish
- Pratama, Alvin Dio. 2014. *Perancangan dan Pembuatan Alat Mengukur Tinggi Badan Otomatis Berbasis Mikrokontroler*.
- Pressman, Roger S. 2012. *Rekayasa perangkat Lunak (Buku Satu)* : Penerbit Andi. Yogyakarta
- Saftari, Firmansyah. 2015. *Proyek Robotik Keren dengan Arduino*. PT. Elex Media Komputindo
- Setyani, Sri. 2016. *Rancang Bangun Alat Pengaman Brankas Menggunakan RFID (Radio Frequency Identification) Dengan Memanfaatkan E-KTP Sebagai Tag Berbasis Arduino*.
- Syam, Rafiuddin. 2013. *Seri Buku Ajar Dasar Dasar Teknik Sensor*. Fakultas Teknik Universitas Hasanuddin
- Yatini, Indra. 2010. *Flowchart, Algoritma dan pemrograman menggunakan bahasa C++*. Yogyakarta: Graha Ilmu.
- Yudhanto, Yudho. 2018. *Panduan Pengantar Belajar Hardware dan Software: Troubleshooting Laptop*.
- Yurindra. 2017. *Software Engineering*. Yogyakarta: Deepublish

LAMPIRAN

LAMPIRAN BIAYA PEMBUATAN ALAT

No	Nama Barang	Harga	Jumlah	Total	
1	Arduino Mega + Kabel USB	Rp180.000	1	Rp	180000
2	Keypad Membrane 4x4	Rp12.000	1	Rp	12000
3	MFRC522 + Label	Rp30.000	1	Rp	30000
4	LCD 16x2	Rp25.000	1	Rp	25000
5	I2C Adapter for LCD 16x2	Rp15.000	1	Rp	15000
6	Micro Servo SG90	Rp35.000	1	Rp	35000
7	Buzzer	Rp5.000	1	Rp	5000
8	Breadboard 830 Points	Rp20.000	1	Rp	20000
9	Power Adapter DC 12V	Rp25.000	1	Rp	25000
10	Jumper Male to Male	Rp20.000	1	Rp	20000
11	Jumper Female to Male	Rp20.000	1	Rp	20000
12	Triplek	Rp120.000	1	Rp	120000
13	Lem Korea	Rp6.000	2	Rp	12000
14	Lem Kertas	Rp15.000	1	Rp	15000
15	Kaca Mika	Rp550.000	1	Rp	550000
Total Biaya				Rp	1.084.000

LAMPIRAN CODING PROGRAM ARDUINO IDE

```
//Nama   : Alfred Christian Boyong
//NIM    : 15.43.061
//Prototype Pengendali Pintu Otomatis Menggunakan Sistem
Minimum Arduino Mega 2560

#include <SPI.h>
#include <MFRC522.h>
#include <LiquidCrystal_I2C.h>
#include <Keypad.h>
#include <Servo.h>

#define pinServo 48
#define pinBuzzer 49
#define RST 5
#define SS 53

boolean hallo, cdAlarm, kunciPintu;
const byte Baris = 4; //Empat Baris
const byte Kolom = 4; //Empat Kolom
char keyMap[Baris][Kolom] = {
    {'1', '2', '3', 'A'},
    {'4', '5', '6', 'B'},
    {'7', '8', '9', 'C'},
    {'*', '0', '#', 'D'}};

byte pinBaris[Baris] = {A7, A6, A5, A4};
byte pinKolom[Kolom] = {A3, A2, A1, A0};

Keypad myKeypad = Keypad(makeKeypad(keyMap), pinBaris,
pinKolom, Baris, Kolom);
LiquidCrystal_I2C lcd(0x27, 16, 2);
Servo myServo;
MFRC522 rfid(SS, RST);

byte sumbuX, x, pos, salahInput, mGanti, kartu[4],
tempKartu[4];
char tombol;
char menu;
String tempPass, password, newPass;

void nilaiAwalVariabel() {
    salahInput = 3;
    x = 2;
    sumbuX = x;
    hallo = true;
    menu = '\0';
    kunciPintu = true;
```

```

}

void setup() {
  nilaiAwalVariabel();
  SPI.begin();
  lcd.begin();
  myServo.attach(pinServo);
  rfid.PCD_Init();
  pinMode(pinBuzzer, OUTPUT);
  password = "1234"; // Password yang disimpan
  kartu[0] = 118;    //ID Kartu Byte 1
  kartu[1] = 139;    //ID Kartu Byte 2
  kartu[2] = 18;     //ID Kartu Byte 3
  kartu[3] = 26;     //ID Kartu Byte 4
  myServo.write(0);
}

void loop() {
  if (hallo) {
    lcd.setCursor(0, 0);
    lcd.print("Memulai sistem");
    tone(pinBuzzer, 1500, 1250);
    delay(1500);
    hallo = false;
    lcd.clear();
  }
  else if (!hallo) {
    if (menu == '\0') {
      tombol = myKeypad.getKey();
      if (kunciPintu) {
        lcd.setCursor(0, 0);
        lcd.print("A:Aktif sistem");
        lcd.setCursor(0, 1);
        lcd.print("B:Ganti password");

        if (tombol != NO_KEY && tombol == 'A') {
          lcd.clear();
          cdAlarm = true;
          menu = 'A';
        }
        else if (tombol != NO_KEY && tombol == 'B') {
          lcd.clear();
          mGanti = 0;
          menu = 'B';
        }
      }
    }
    else if (!kunciPintu) {
      lcd.setCursor(0, 0);
      lcd.print("C: Tutup pintu");
    }
  }
}

```

```

        lcd.setCursor(0, 1);
        lcd.print("atau scan RFID");

        inputRFID();
        if (tombol != NO_KEY && tombol == 'C') {
            lcd.clear();
            menu = 'C';
        }
    }
}
else if (menu == 'A') {
    AktifkanSistem();
}
else if (menu == 'B') {
    gantiPassword();
}
else if (menu == 'C') {
    KunciSistem();
}
else if (menu == 'X') {
    char buff[16];
    byte cd = 60;
    while (cd > 0) {
        lcd.setCursor(0, 0);
        lcd.print("Sistem terkunci");
        lcd.setCursor(0, 1);
        sprintf(buff, "%0.2d Detik", cd);
        lcd.print(buff);

        tone(pinBuzzer, 1000, 500);
        delay(1000); //60 Detik = 1 Menit
        cd --;
    }
    nilaiAwalVariabel();
    lcd.clear();
}
}
}

```

```

void AktifkanSistem() {
    byte cd = 3;
    if (cdAlarm) {
        while (cd > 0) {
            tone(pinBuzzer, 700, 50);
            lcd.setCursor(0, 0);
            lcd.print("Sistem aktif");
            lcd.setCursor(0, 1);
            lcd.print("dalam : ");
            lcd.print(cd);
        }
    }
}

```

```

        lcd.print(" detik");
        cd--;
        delay(1000);
    }
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Sistem aktif!");
    tone(pinBuzzer, 950, 500);
    delay(1000);
    lcd.clear();
    cdAlarm = false;
}
else if (!cdAlarm) {
    lcd.setCursor(0, 0);
    lcd.print("RFID/Password:");

    inputRFID();
    inputKeypad();
    cekInputKeypad();
}
}

void gantiPassword() {
    if (mGanti == 0) {
        lcd.setCursor(0, 0);
        lcd.print("Ganti Password:");

        inputKeypad();
        cekInputKeypad();
    }
    else if (mGanti == 1) {
        lcd.setCursor(0, 0);
        lcd.print("Password baru:");
        inputKeypad();
        if (tombol == '#' && sumbuX == 6) {
            newPass = tempPass;
            if (newPass == password) {
                lcd.setCursor(0, 0);
                lcd.print("Password harus");
                lcd.setCursor(0, 1);
                lcd.print("berbeda");
                tone(pinBuzzer, 450, 125);
                delay(250);
                tone(pinBuzzer, 450, 125);
                delay(1000);
            }
            else {
                mGanti ++;
            }
        }
    }
}

```

```

        lcd.clear();
        tempPass = "";
        sumbuX = x;
    }
}
else if (mGanti == 2) {
    lcd.setCursor(0, 0);
    lcd.print("Ulangi Password:");
    inputKeypad();
    if (tombol == '#' && sumbuX == 6) {
        lcd.clear();
        if (newPass == tempPass) {
            lcd.setCursor(0, 0);
            lcd.print("Password telah");
            lcd.setCursor(0, 1);
            lcd.print("diubah");
            tone(pinBuzzer, 950, 500);
            delay(1000);
            password = newPass;
            menu = '\0';
            mGanti = 0;
        }
        else {
            lcd.setCursor(0, 0);
            lcd.print("Password tidak");
            lcd.setCursor(0, 1);
            lcd.print("sesuai");
            tone(pinBuzzer, 450, 125);
            delay(250);
            tone(pinBuzzer, 450, 125);
            delay(1000);
        }
        lcd.clear();
        tempPass = "";
        sumbuX = x;
    }
}
}

void KunciSistem() {
    tone(pinBuzzer, 450, 500);
    lcd.setCursor(0, 0);
    lcd.print("Menutup pintu");
    for (pos = 135; pos > 0; pos--) {
        myServo.write(pos);
        delay(25);
    }
    lcd.clear();
    lcd.setCursor(0, 0);
}

```



```

    lcd.print("Pintu tertutup!");
    tone(pinBuzzer, 950, 1000);
    delay(1500);

    lcd.clear();
    menu = '\0';
    kunciPintu = true;
}

void inputRFID() {
    if (! rfid.PICC_IsNewCardPresent())
        return;
    if (! rfid.PICC_ReadCardSerial())
        return;
    for (byte a = 0; a < 4; a++)
        tempKartu[a] = rfid.uid.uidByte[a];

    if (kunciPintu) {
        if (memcmp(tempKartu, kartu, 4) == 0)
            cekValidasi(true);
        else
            cekValidasi(false);
    } else {
        if (memcmp(tempKartu, kartu, 4) == 0) {
            lcd.clear();
            menu = 'C';
        }
    }
    rfid.PICC_HaltA();
    rfid.PCD_StopCryptol();
}

void inputKeypad() {
    lcd.setCursor(0, 1);
    lcd.print("> ");

    if (salahInput > 0 && salahInput < 3) {
        lcd.setCursor(9, 1);
        lcd.print("Sisa ");
        lcd.print(salahInput);
        lcd.print("X");
    }

    tombol = myKeypad.getKey();
    if (tombol != NO_KEY) {
        if ((tombol == '0' || tombol == '1' || tombol == '2' ||
tombol == '3' || tombol == '4' ||
        tombol == '5' || tombol == '6' || tombol == '7' ||
tombol == '8' || tombol == '9') && sumbuX < 6) {

```

```

        bunyiKeypad();
        lcd.setCursor(sumbuX, 1);
        lcd.print('*'); //Ubah menjadi '*' untuk password
        tempPass += tombol;
        sumbuX++;
    }
    else if (tombol == 'B' && sumbuX > x) {
        bunyiKeypad();
        lcd.clear();
        tempPass = tempPass.substring(0,tempPass.length()-1);
        sumbuX --;
        for (byte a = x; a < sumbuX; a++) {
            lcd.setCursor(a, 1);
            lcd.print('*');
        }
    }
    else if (tombol == 'C' && sumbuX > x) {
        bunyiKeypad();
        lcd.clear();
        tempPass = "";
        sumbuX = x;
    }
    else if (tombol == 'D') {
        bunyiKeypad();
        lcd.clear();
        tempPass = "";
        sumbuX = x;
        menu = '\0';
    }
}
}

void cekInputKeypad() {
    if (tombol == '#' && sumbuX == 6) {
        if (tempPass == password) //Jika Password Benar
            cekValidasi(true);
        else //Jika Password Salah
            cekValidasi(false);

        tempPass = "";
        sumbuX = x;
    }
}

void cekValidasi(boolean valid) {
    lcd.clear();
    if (valid) {
        lcd.setCursor(0, 0);
    }
}

```

```

if (menu == 'A') {
    lcd.print("Membuka pintu");
    tone(pinBuzzer, 450, 500);
    for (pos = 0; pos < 135; pos++) {
        myServo.write(pos);
        delay(25);
    }
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Pintu terbuka");
    tone(pinBuzzer, 950, 1000);
    delay(1500);

    menu = '\0';
    kunciPintu = false;
}
else if (menu == 'B')
    mGanti++;

    salahInput = 3;
}
else {
    salahInput --;
    tone(pinBuzzer, 450, 125);
    delay(250);
    tone(pinBuzzer, 450, 125);

    if (salahInput == 0) {
        menu = 'X';
        lcd.clear();
    }
}
}

void bunyiKeypad() {
    switch (tombol) {
        case '1':
            tone(pinBuzzer, 697, 100);
            break;
        case '2':
            tone(pinBuzzer, 824, 100);
            break;
        case '3':
            tone(pinBuzzer, 965, 100);
            break;
        case 'A':
            tone(pinBuzzer, 1121, 100);
            break;
        case '4':

```

```
        tone(pinBuzzer, 770, 100);
        break;
    case '5':
        tone(pinBuzzer, 897, 100);
        break;
    case '6':
        tone(pinBuzzer, 1038, 100);
        break;
    case 'B':
        tone(pinBuzzer, 1194, 100);
        break;
    case '7':
        tone(pinBuzzer, 852, 100);
        break;
    case '8':
        tone(pinBuzzer, 979, 100);
        break;
    case '9':
        tone(pinBuzzer, 1120, 100);
        break;
    case 'C':
        tone(pinBuzzer, 1276, 100);
        break;
    case '*':
        tone(pinBuzzer, 941, 100);
        break;
    case '0':
        tone(pinBuzzer, 1068, 100);
        break;
    case '#':
        tone(pinBuzzer, 1209, 100);
        break;
    case 'D':
        tone(pinBuzzer, 1365, 100);
        break;
}
}
```

DATASHEET

ARDUINO MEGA 2560

DATASHEET
KEYPAD MEMBRANE
4X4

DATASHEET

RFID MFRC522

DATASHEET

LCD 16X2

DATASHEET

MICRO SERVO SG90