

**PENGENDALI PINTU DENGAN SISTEM KEAMANAN BERBASIS WEB
DAN ANDROID**



ILMA KAMILIA

5215134335

Skripsi Ini Ditulis Untuk Memenuhi Sebagian Persyaratan

Dalam Memperoleh Gelar Sarjana

PROGRAM STUDI PENDIDIKAN TEKNIK ELEKTRONIKA

FAKULTAS TEKNIK UNIVERSITAS NEGERI JAKARTA

2017

HALAMAN PENGESAHAN
PENGESAHAN DOSEN PEMBIMBING

Nama Dosen

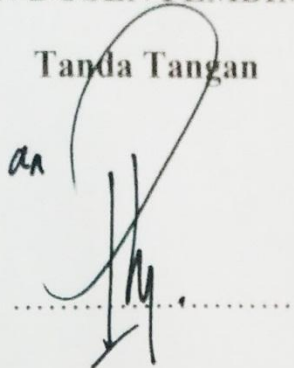
Tanda Tangan

Tanggal

Drs. Jusuf Bintoro, MT.

NIP. 196101081987031003

(Dosen Pembimbing I)

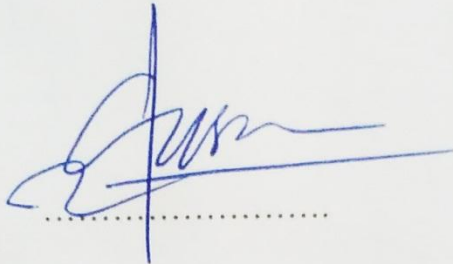
an

.....

21/8/2017
.....

Dr. Muhammad Yusro, MT.

NIP. 197609212001121002

(Dosen Pembimbing II)


.....

18/8/2017
.....

PENGESAHAN PANITIA UJIAN SKRIPSI

Nama Dosen

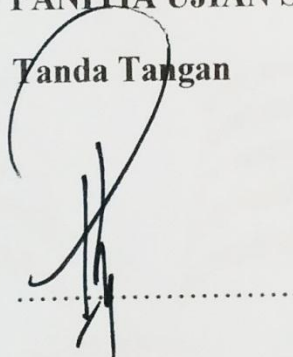
Tanda Tangan

Tanggal

Drs. Pitoyo Yuliatmojo, MT.

NIP. 196807081994031003

(Ketua Penguji)

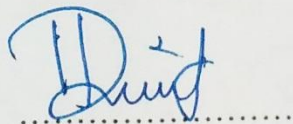

.....

21/8/2017
.....

Diat Nur Hidayat, MT.

NIDK. 8884100016

(Sekretaris)

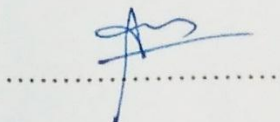

.....

18/8/2017
.....

Aodah Diamah, M.Eng.

NIP. 197809192005012003

(Dosen Ahli)


.....

18 Agustus 2017
.....

Tanggal Lulus : 15 Agustus 2017

HALAMAN PERNYATAAN

Dengan ini saya menyatakan bahwa :

1. Karya tulis skripsi saya ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik sarjana, baik di Universitas Negeri Jakarta maupun di perguruan tinggi negeri lain.
2. Karya tulis ini adalah murni gagasan, rumusan dan penelitian saya sendiri dengan arahan dosen pembimbing.
3. Dalam karya tulis ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka.
4. Pernyataan ini saya buat dengan sesungguhnya dan apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya tulis ini, serta sanksi lainnya sesuai dengan norma yang berlaku di Universitas Negeri Jakarta.

Jakarta, 10 Agustus 2017

Yang Membuat Pernyataan



Ilma Kamilia

5215134335

ABSTRAK

Ilma Kamilia, *Pengendali Pintu Dengan Sistem Keamanan Berbasis Web dan Android*. Skripsi. Jakarta, Program Studi Pendidikan Teknik Elektronika, Fakultas Teknik Universitas Negeri Jakarta, 2017. Dosen Pembimbing: Drs. Jusuf Bintoro, MT dan Dr. Muhammad Yusro, MT.

Penelitian ini bertujuan untuk merancang bangun alat pengendali pintu dengan sistem keamanan berbasis web dan android. Penelitian dilaksanakan di Laboratorium Instrumentasi Fakultas Teknik Universitas Negeri Jakarta pada Januari - Juli 2017.

Penelitian ini menggunakan Metodologi Rekayasa Teknik yang meliputi perancangan sistem, perancangan *hardware*, perancangan *software*, pengujian *hardware* dan *software*, dan analisis pengujian. Penelitian ini terdiri dari 3 subsistem utama, yaitu *input*, proses, dan *output*. Subsistem *input* terdiri dari 2 buah *limit switch* yang digunakan sebagai sensor untuk membuka dan menutup pintu. 4 buah *push button* yang terbagi ke dalam dua jenis sistem yaitu 2 buah *push button* sebagai tombol darurat, dan 2 buah *push button* lainnya sebagai tombol pintu keluar. *Personal computer/ laptop* dan *smartphone android* sebagai penampil pemberi instruksi untuk membuka pintu dan informasi kegiatan sistem. Subsistem proses yaitu Arduino Mega 2560 akan saling mengirimkan data dengan *Go-Language* sebagai *interface* dan *SQLite* sebagai *database*. Subsistem *output* yaitu 2 buah *relay* sebagai *driver solenoid* dan *solenoid* sebagai kunci otomatisnya. 1 buah *buzzer* yang dimanfaatkan sebagai indikator saat pintu terbuka, tertutup atau dalam kondisi tidak aman.

Hasil penelitian ini menunjukkan alat ini dapat melakukan buka dan tutup kunci pintu secara otomatis yang dikendalikan dari jarak jauh melalui *website* dan aplikasi *android* menggunakan *personal computer/ laptop* dan *smartphone android* yang terkoneksi jaringan internet.

Kata kunci : Pintu Rumah, Arduino Mega2560, Website, Android, Solenoid.

ABSTRACT

Ilma Kamilia, *Door Controller With Web-Based Security System and Android*. Essay. Jakarta, Electronic Engineering Education Study Program, Faculty of Engineering, Jakarta State University, 2017. Supervisor: Drs. Jusuf Bintoro, MT and Dr. Muhammad Yusro, MT.

This study aims to design a door control device with web-based security system and android. The research was conducted in Instrumentation Laboratory of Faculty of Engineering, State University of Jakarta in January - July 2017.

This research uses Engineering Methodology which includes system design, hardware design, software design, hardware and software testing, and testing analysis. This research consists of 3 main subsystem, that is input, process, and output. The input subsystem consists of 2 limit switches used as sensors for opening and closing doors. 4 push buttons are divided into two types of system that is 2 pieces push button as the emergency button, and 2 other push button as the exit button. Personal computer / laptop and android smartphone as a giver of instructions to open doors and information system activities. The process subsystem, Arduino Mega 2560, will transmit data with Go-Language as an interface and SQLite as a database. The output subsystem is 2 pieces of relay as solenoid and solenoid drivers as automatic keys. 1 buzzer is used as an indicator when the door is open, closed or in unsafe condition.

The results of this study show that this tool can open and close the door lock automatically controlled remotely through website and android app using personal computer / laptop and android smartphone connected internet network.

Keywords : House Door, Arduino Mega2560, Website, Android, Solenoid.

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT yang telah memberikan kesempatan, kekuatan dan kesehatan untuk bisa menyelesaikan penelitian alat yang berjudul “Pengendali Pintu Dengan Sistem Keamanan Berbasis Web dan Android”.

Kelancaran penulisan penelitian ini selain atas limpahan karunia Allah SWT, juga berkat dukungan pembimbing, orang tua, suami dan kawan-kawan. Untuk itu penulis ingin mengucapkan rasa terima kasih kepada :

1. Drs. Pitoyo Yuliatmojo, M.T, selaku Kepala Program Studi Pendidikan Teknik Elektronika UNJ.
2. Drs. Jusuf Bintoro, M.T, selaku Dosen Pembimbing I Skripsi.
3. Dr. Muhammad Yusro, M.T, selaku Dosen Pembimbing II Skripsi.
4. Seluruh Dosen Elektronika Fakultas Teknik Universitas Negeri Jakarta.
5. Suami saya tercinta, Hikman Sari, S.T
6. Ibu saya tercinta Ibu Sri Mulyanti
7. Ibu dan bapak mertua Ibu Hj. Asiyah dan Bapak Agus
8. Kakak-kakak dan adik yaitu mbak Irna, mbak Islah, kak Ratna dan Rahmad
9. Seluruh teman-teman Pendidikan Teknik Elektronika angkatan 2013 dan adik kelas serta kakak alumni yang telah mensupport saya.
10. Sahabat-sahabat saya Ella Marlina dan Nurul Hidayati, terimakasih kalian telah sabar mendengar keluhan saya selama penulisan skripsi.
11. Seluruh teman-teman kuliah saya Rizky, Gita, Dian, Ana, Ulfah, Desmoon, Rigit, Ghoffar, Meji, Kevlyn, Rizqiana dkk. yang selalu mensupport, berbagi cerita dan keluh kesah serta menemani revisi saya.

Demikianlah ucapan terima kasih saya selaku penulis. Semoga Allah SWT senantiasa memberikan balasan terbaik atas jasa-jasa yang telah diberikan kalian. Baik di dunia maupun di akhirat. Amin

Jakarta, 10 Agustus 2017

Penulis.

DAFTAR ISI

HALAMAN PENGESAHAN.....	i
HALAMAN PERNYATAAN.....	ii
ABSTRAK.....	iii
ABSTRACT.....	iv
KATA PENGANTAR.....	v
DAFTAR ISI.....	vi
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xii
DAFTAR LAMPIRAN.....	xvi

BAB I PENDAHULUAN

1.1. Latar Belakang.....	1
1.2. Identifikasi Masalah.....	2
1.3. Pembatasan Masalah.....	3
1.4. Rumusan Masalah.....	3
1.5. Tujuan Penelitian.....	4
1.6. Manfaat Penelitian.....	4

BAB II TINJAUAN PUSTAKA

2.1. Kerangka Teoritik.....	6
2.1.1. Pengertian Sistem Pengendali.....	6
2.1.2. Arduino Mega 2560.....	8
2.1.2.1. Bagian-bagian Arduino Mega 2560.....	10

2.1.3. Solenoid.....	15
2.1.4. Limit Switch.....	17
2.1.5. Buzzer.....	18
2.1.6. Push Button.....	19
2.1.7. Relay.....	20
2.1.8. EMS Ethernet Module.....	22
2.1.9. Router.....	22
2.1.10. Modem.....	23
2.1.11. Website.....	24
2.1.12. Sistem Operasi Android.....	26
2.1.13. Arduino IDE.....	27
2.1.14. Go-Language (Golang).....	28
2.1.15. SQLite.....	29
2.2. Simbol Komponen Aktif dan Pasif.....	30
2.2.1. Komponen Aktif.....	30
a. Transistor.....	30
b. Dioda.....	31
c. IC (<i>Integrated Circuit</i>).....	31
2.2.2. Komponen Pasif.....	32
a. Resistor.....	32
b. Kapasitor.....	33
c. Induktor.....	34
2.3. Kerangka Berpikir.....	34
2.2.1. Blok Diagram.....	37
2.2.2. <i>Flowchart</i> Sistem.....	38
2.3. Penelitian Terkait.....	39

BAB III METODOLOGI PENELITIAN

3.1. Tempat dan Waktu Penelitian.....	41
3.2. Alat dan Bahan Penelitian.....	41
3.2.1. <i>Hardware</i>	41
3.2.2. <i>Software</i>	42
3.3. Diagram Alir Penelitian.....	42
3.3.1. Perancangan <i>Hardware</i>	44
3.3.1.1. Menentukan Jenis Pintu.....	44
3.3.1.2. Menentukan Sistem Kendali.....	45
3.3.1.3. Menentukan <i>Solenoid</i>	46
3.3.1.4. Perancangan Rangkaian <i>Driver Solenoid</i>	46
3.3.1.5. Menentukan <i>Limit Switch</i>	47
3.3.1.6. Menentukan <i>Buzzer</i>	47
3.3.1.7. Menentukan <i>Ethernet</i>	48
3.3.1.8. Menentukan <i>Router</i>	50
3.3.1.9. Menentukan <i>Modem</i>	51
3.3.1.10. Menentukan <i>Smartphone</i>	51
3.3.1.11. Menentukan <i>Regulator</i>	52
3.3.1.12. Menentukan Kunci Manual Elektronik.....	53
3.3.1.13. Menentukan <i>Door Closer</i>	53
3.3.1.14. Menentukan <i>Push Button</i>	54
3.3.2. Perancangan <i>Software</i>	55
3.3.2.1. <i>Arduino IDE</i>	55
3.3.2.2. <i>Go-Language (Golang)</i>	55

3.3.2.3. <i>SQLite</i>	56
3.3.3. Menentukan Desain Maket Alat.....	59
3.4. Teknik dan Prosedur Pengumpulan Data.....	59
3.5. Teknik Analisis Data.....	61
3.5.1. Pengujian Pada <i>Driver Solenoid</i>	62
3.5.2. Pengujian <i>Regulator</i>	62
3.5.3. Pengujian <i>Limit Swich</i>	63
3.5.4. Pengujian <i>Push Button</i>	63
3.5.5. Pengujian Kecepatan Buka Kunci Pintu Melalui Web dan Aplikasi Android.....	64
3.5.6. Pengujian Kecepatan Buka Kunci Pintu Melalui Tombol Keluar....	65
3.5.7. Pengujian Kecepatan Buka Kunci Pintu Melalui Tombol Darurat.....	66
3.5.8. Komunikasi Web dengan Arduino Mega 2560.....	66

BAB IV HASIL PENELITIAN

4.1. Deskripsi Hasil Penelitian.....	69
4.1.1 Maket Alat.....	69
4.1.2 Tampilan Web dan Android.....	72
4.1.3. Langkah Kerja.....	76
4.2. Analisis Data Penelitian.....	82
4.2.1. Pengujian Pada <i>Driver Solenoid (Relay)</i>	82
4.2.2. Pengujian <i>Regulator</i>	83
4.2.3. Pengujian <i>Limit Switch</i>	85
4.2.4. Pengujian <i>Push Button</i>	86

4.2.5. Pengujian Kecepatan Buka Kunci Pintu Melalui Web dan Aplikasi Android.....	87
4.2.6. Pengujian Kecepatan Buka Kunci Pintu Melalui Tombol Keluar....	89
4.2.7. Pengujian Kecepatan Buka Kunci Pintu Melalui Tombol Darurat...	91
4.2.8. Program Komunikasi Web dengan Arduino Mega 2560.....	93
4.3. Pembahasan.....	97
4.4. Aplikasi Hasil Penelitian..	100

BAB V KESIMPULAN DAN SARAN

5.1. Kesimpulan.....	101
5.2. Saran.....	102

DAFTAR PUSTAKA

LAMPIRAN-LAMPIRAN

DAFTAR TABEL

Tabel 3.1. Spesifikasi Arduino Mega 2560.....	45
Tabel 3.2. Hasil Pengujian <i>Relay Solenoid</i>	62
Tabel 3.3. Hasil Pengujian Regulator.....	62
Tabel 3.4. Hasil Pengujian <i>Limit Switch</i>	63
Tabel 3.5. Hasil Pengujian <i>Push Button</i>	63
Tabel 3.6. Hasil Pengujian Kecepatan Buka Kunci Pintu Melalui Web dan Aplikasi Android.....	64
Tabel 3.7. Hasil Pengujian Kecepatan Buka Kunci Pintu Melalui Tombol Keluar.....	65
Tabel 3.8. Hasil Pengujian Kecepatan Buka Kunci Pintu Melalui Tombol Darurat.....	66
Tabel 4.1. Hasil Pengujian <i>Relay Solenoid</i>	83
Tabel 4.2. Hasil Pengujian <i>Regulator</i>	83
Tabel 4.3. Hasil Pengujian <i>Limit Switch</i>	85
Tabel 4.4. Hasil Pengujian <i>Push Button</i>	86
Tabel 4.5. Hasil Pengujian Kecepatan Buka Kunci Pintu Melalui Web dan Aplikasi Android.....	87
Tabel 4.6. Hasil Pengujian Kecepatan Buka Kunci Pintu Melalui Tombol Keluar.....	89
Tabel 4.7. Hasil Pengujian Kecepatan Buka Kunci Pintu Melalui Tombol Darurat.....	91

DAFTAR GAMBAR

Gambar 2.1. Konfigurasi Sistem Kontrol Loop Terbuka.....	7
Gambar 2.2. Konfigurasi Sistem Kontrol Loop Tertutup.....	7
Gambar 2.3. Berbagai Macam Arduino.....	9
Gambar 2.4. Arduino Mega 2560.....	9
Gambar 2.5. Bagian – bagian dari Arduino Mega 2560.....	10
Gambar 2.6. Hubungan Pin ATmega2560 dengan Pin Arduino Mega 2560.....	14
Gambar 2.7. Bentuk Fisik <i>Solenoid</i>	16
Gambar 2.8. Cara Kerja <i>Solenoid</i>	17
Gambar 2.9. Pergerakan <i>Solenoid</i>	17
Gambar 2.10. <i>Limit Switch</i>	18
Gambar 2.11. <i>Buzzer</i>	19
Gambar 2.12. Rangkaian <i>Push Button</i>	20
Gambar 2.13. <i>Relay</i>	22
Gambar 2.14. <i>EMS Ethernet Module</i>	22
Gambar 2.15. <i>Router</i>	23
Gambar 2.16. <i>Modem</i>	24
Gambar 2.17. <i>Website</i>	25
Gambar 2.18. Logo Android.....	26
Gambar 2.19. Editor Arduino IDE.....	27
Gambar 2.20. Tampilan Golang.....	29
Gambar 2.21. <i>SQLite</i>	30
Gambar 2.22. Transistor.....	31
Gambar 2.23. Dioda.....	31
Gambar 2.24. IC (<i>Integrated Circuit</i>).....	32

Gambar 2.25. Resistor.....	33
Gambar 2.26. Kapasitor.....	33
Gambar 2.27. Induktor.....	34
Gambar 2.28. Blok Diagram.....	37
Gambar 2.29. <i>Flowchart</i>	38
Gambar 3.1. Diagram Alir Metode Penelitian Alat.....	42
Gambar 3.2. Pintu <i>Swing</i>	44
Gambar 3.3. Arduino Mega 2560.....	46
Gambar 3.4. <i>Solenoid NC</i>	46
Gambar 3.5. Rangkain <i>Driver Solenoid</i>	47
Gambar 3.6. <i>Limit Switch</i>	47
Gambar 3.7. <i>Passive Piezoelektrik Buzzer</i>	48
Gambar 3.8. <i>EMS Ethernet Module</i>	49
Gambar 3.9. Router TP-Link TL-MR3020.....	50
Gambar 3.10. <i>Modem</i>	51
Gambar 3.11. <i>Smartphone Android Ice Cream Sandwich</i>	52
Gambar 3.12. <i>Regulator</i>	52
Gambar 3.13. Kunci Manual Elektronik.....	53
Gambar 3.14. <i>Door Closer Manual</i>	54
Gambar 3.15. Tombol Keluar.....	54
Gambar 3.16. Tombol Darurat.....	54
Gambar 3.17. Tampilan Arduino IDE	55
Gambar 3.18. <i>Go-Language (Golang)</i>	56
Gambar 3.19. <i>SQLite Database</i>	56
Gambar 3.20. Tampilan Halaman <i>Login</i>	57
Gambar 3.21. Diagram Alir Halaman <i>Login</i>	58

Gambar 3.22. Maket Alat.....	59
Gambar 4.1. Penampakan Keseluruhan Maket Alat.....	69
Gambar 4.2. Penampakan Pintu dari Sisi Depan.....	70
Gambar 4.3. Penampakan Pintu dari Sisi Belakang Pintu (Dalam Rumah)	70
Gambar 4.4. <i>Limit Switch</i>	71
Gambar 4.5. Box Panel Kontrol.....	71
Gambar 4.6. Halaman <i>Login</i> dari <i>Website</i>	72
Gambar 4.7. Aplikasi Pengendali Pintu (<i>Smart Door</i>)	72
Gambar 4.8. Halaman <i>Login</i> Pada Aplikasi Android.....	73
Gambar 4.9. Halaman Ucapan Selamat Datang Pada Aplikasi Android.....	73
Gambar 4.10. Menu <i>Realtime</i>	74
Gambar 4.11. Menu <i>Email</i>	74
Gambar 4.12. Menu <i>User</i>	75
Gambar 4.13. <i>Application Exit</i>	75
Gambar 4.14. Alamat URL Website.....	76
Gambar 4.15. Aplikasi Pengendali Pintu (<i>Smart Door</i>)	76
Gambar 4.16. Tampilan Halaman <i>Login</i>	77
Gambar 4.17. Menu <i>Realtime</i>	77
Gambar 4.18. Pintu Terbuka.....	78
Gambar 4.19. Buka Pintu dari Tombol Keluar.....	79
Gambar 4.20. Email Notifikasi Kondisi Tidak Aman.....	80
Gambar 4.21. Pintu Dibuka Paksa.....	80
Gambar 4.22. Buzzer Berdering.....	80
Gambar 4.23. Tombol Darurat.....	81
Gambar 4.24. Kunci Manual Elektronik di Rumah.....	82
Gambar 4.25. Kunci Manual Biasa.....	82

Gambar 4.26. Pengecekan Status Internet.....	93
Gambar 4.27. Koneksi ke Server.....	94
Gambar 4.28. Target Server.....	94
Gambar 4.29. URL Tanpa Domain.....	94
Gambar 4.30. Arduino Menggantikan Perintah Koneksi Server.....	95
Gambar 4.31. Buka Kunci Pintu Dari Web.....	95
Gambar 4.32. Buka Kunci Pintu Dari Tombol Keluar.....	96
Gambar 4.33. Cek Buka Pintu Paksa.....	96
Gambar 4.34. Buka Kunci Pintu Dari Tombol Darurat.....	97

DAFTAR LAMPIRAN

Lampiran 1. Source Code Arduino IDE

Lampiran 2. Source Code SQLite

Lampiran 3. Source Code Go-Language

Lampiran 4. Skematik Arduino Mega 2560 dan Data Sheet At Mega 2560

Lampiran 5. Skematik Ethernet Shield

Lampiran 6. Skematik Driver Solenoid (Relay)

Lampiran 7. Skematik Rangkaian Buzzer

Lampiran 8. Skematik Rangkaian Regulator dan Data Sheet IC 7805

BAB I

PENDAHULUAN

1.1 Latar Belakang

Di era yang semakin maju dan teknologi yang semakin canggih, menuntut pekerjaan manusia yang lebih dinamis karena kebutuhan akan efektifitas dan efisiensi sangat diutamakan dalam berbagai bidang. Hal tersebut mendorong manusia untuk berkreasi dan berinovasi dalam bidang teknologi untuk menciptakan suatu alat dengan tingkat efektifitas dan efisiensi yang tinggi sebagai sebuah solusi dari permasalahan yang ada.

Seperti halnya pekerjaan terutama di kota-kota besar yang banyak menuntut jam kerja di luar rumah dengan frekuensi tinggi ditambah dengan kemacetan yang membuat waktu di perjalanan semakin lama dan dampaknya adalah lebih sedikit waktu yang dihabiskan di rumah dengan keluarga kecuali pada hari libur. Untuk kalangan menengah ke atas mungkin akan lebih memilih mempekerjakan pembantu dan satpam untuk menjaga rumah dan harta benda mereka ketika si pemilik rumah berada di luar rumah, tetapi untuk kalangan menengah ke bawah dengan tuntunan pekerjaan yang dilakukan diluar rumah juga tentunya akan sangat merepotkan si pemiliknya untuk mengatur efisiensi dan keamanan rumah mereka.

Untuk menjawab permasalahan tersebut, solusinya adalah dengan pemanfaatan teknologi canggih pengendali pintar dengan sistem keamanan pintu yang bisa dikontrol melalui *smartphone* android dengan internet (wifi) sebagai

serial *interface*-nya. Untuk keamanan rumah agar terhindar dari sesuatu yang tidak kita inginkan, pemilik rumah dapat memasang alarm anti maling yang dipasang pada pintu yang dapat dipantau dari jarak jauh melalui sistem yang dapat bekerja sendiri dan berinteraksi dengan pemilik rumah menggunakan *smartphone* android yang terkoneksi dengan internet.

Untuk masalah keamanan dapat ditambahkan beberapa sensor seperti *limit switch* yang dipasang di kusen pintu. *Limit switch* ini berfungsi sebagai sensor yang dapat memberikan sinyal ke mikrokontroler saat kondisi terbuka ataupun tertutup, sehingga saat si pemilik rumah tidak ada di tempat dan ada orang yang tidak diinginkan masuk ke dalam rumah, maka sistem mikrokontroler yang dipasang di rumah akan langsung mengirim pesan pemberitahuan bahwa ada orang yang masuk ke rumah, dan pemilik rumah bisa membalasnya untuk mengaktifkan alarm rumah atau hanya sekedar memberitahu melalui sms/menelpon tetangga untuk sekedar menyuruh cek rumah tersebut. Untuk pintu yang terbuka atau tertutup juga dapat kita pantau dari jarak jauh dan dapat kita kontrol sendiri menggunakan aplikasi pada *smartphone* android yang terkoneksi internet.

1.2 Identifikasi Masalah

Berdasarkan latar belakang masalah yang telah dipaparkan, maka dapat diidentifikasi masalah yang terjadi yaitu sebagai berikut :

1. Apakah yang dimaksud dengan pengendali pintu dengan sistem keamanan berbasis web dan android?

2. Apakah Arduino Mega 2560 dapat digunakan sebagai kontroler sistem pada rancang bangun alat tersebut ?
3. Bagaimana jika listrik pada rumah tersebut mati atau mengalami gangguan ?
4. Bagaimana jika pulsa internet pada *Smartphone* (Android) habis ?
5. Bagaimana jika *smartphone* tertinggal dirumah atau ditempat lain ?
6. Bagaimana jika *smartphone* sedang *dicharge* atau batre habis ?
7. Bagaimana cara menjaga *security account* pada sistemnya ?
8. Bagaimana jika ada seseorang yang bukan penghuni rumah masuk rumah dengan mendobrak pintu ?

1.3 Pembatasan Masalah

Dalam pembuatan rancang bangun alat ini permasalahan yang dibahas oleh penulis hanya membahas pembuatan pengendali pintu dengan sistem keamanan berbasis web dan android untuk 3 orang penghuni rumah dan 2 pintu yaitu pintu utama (depan) dan pintu belakang rumah.

1.4 Rumusan Masalah

Berdasarkan latar belakang, identifikasi masalah, pembatasan masalah yang telah diuraikan diatas maka dapat dirumuskan masalah dari penelitian ini adalah bagaimana membuat rancang bangun alat pengendali pintu dengan sistem keamanan berbasis web dan android?

1.5 Tujuan Penelitian

Adapun beberapa tujuan yang ingin dicapai dari pembuatan rancang bangun alat pengendali pintu dengan sistem keamanan berbasis web dan android adalah sebagai berikut :

1. Membangun pengendali pintu dengan sistem keamanan berbasis web dan android.
2. Meningkatkan kenyamanan dan keamanan bagi penghuni rumah.

1.6 Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini, yaitu mempunyai kontribusi pada beberapa aspek antara lain :

1. Aspek Keteknikan

Alat ini masih menggunakan Arduino Mega 2560, namun peneliti lain masih bisa gunakan dan memodifikasi dengan menggunakan sistem mikrokontroler lain dan ditambah dengan fitur lainnya ataupun dipadukan dengan sumber alternatif sehingga dapat berfungsi lebih optimal serta menghemat tenaga dan waktu.

2. Aspek Kependidikan

Menjadi sumber referensi untuk pembelajaran mengenai pengendali pintu dengan sistem keamanan berbasis web dan android.

3. Aspek Ekonomi

Alat ini merupakan salah satu alat yang mempunyai nilai ekonomis, dinamis, efisiensi, praktis dan daya jual yang tinggi. Alat ini dapat dimanfaatkan bagi semua lapisan masyarakat baik yang bertempat tinggal di kompleks

perumahan maupun perumahan biasa atau kavling, keamanan sebagai alat monitoring pintu gerbang maupun pintu rumah tiap blok-nya sehingga dapat mencegah tindak kriminal pencurian yang sedang marak terjadi.

BAB II

TINJAUAN PUSTAKA

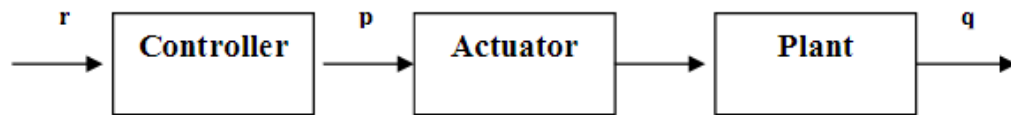
Pada perancangan sistem ini, ada beberapa perangkat yang akan digunakan sebagai perangkat utama maupun sebagai perangkat pendukung. Untuk perangkat utama didasarkan kepada besarnya ketergantungan sistem terhadap alat yang bersangkutan begitupun sebaliknya dengan perangkat pendukung.

2.1. Kerangka Teoritik

2.1.1. Pengertian Sistem Pengendali

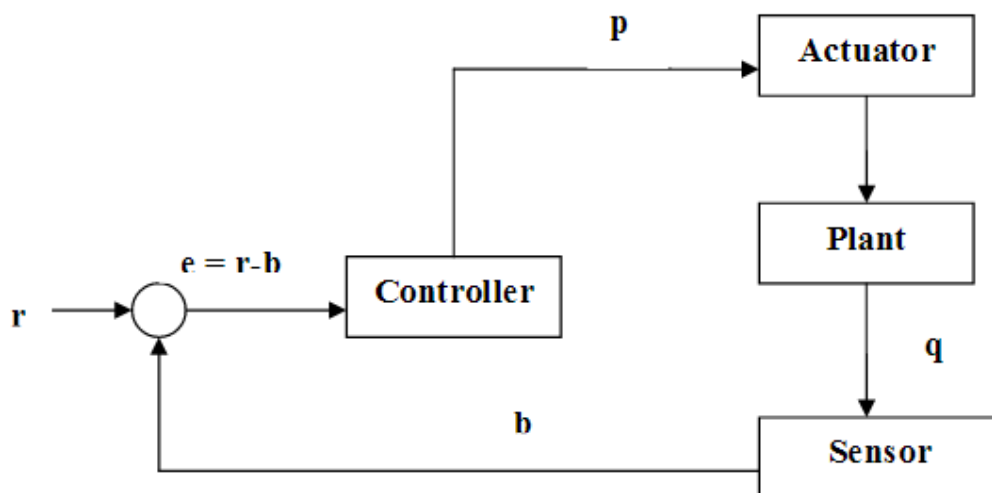
Definisi dari sistem kontrol adalah, jalinan berbagai komponen yang menyusun sebuah sistem untuk menghasilkan respon yang diinginkan terhadap perubahan waktu. Dengan demikian sebuah sistem kontrol dapat dicirikan dengan adanya mekanisme pengkondisian sistem untuk mencapai respon yang diinginkan. Berdasarkan tujuan perancangan sistem kontrol, mekanisme kerja sistem kontrol dapat dibagi menjadi dua golongan, yaitu: regulator dan servomekanis. Mekanisme kerja regulator, bertujuan supaya nilai respon keluaran dijaga konstan sesuai nilai yang diinginkan. Namun mekanisme kerja servomekanis, bertujuan supaya respon keluaran sistem menghasilkan nilai-nilai tertentu yang menyebabkan sistem stabil. Dalam rangka memenuhi kebutuhan perancangan sistem kontrol, struktur sistem kontrol dapat digolongkan kedalam dua bagian besar, yaitu : sistem kontrol loop tertutup (closed loop) dan sistem kontrol loop terbuka (open loop). Sistem kontrol loop terbuka adalah konfigurasi dari sistem kontrol yang tidak memerlukan informasi mengenai keadaan keluaran sistem.

Ilustrasi diagram blok sistem kontrol loop terbuka ditunjukkan melalui gambar berikut:



Gambar 2.1. Konfigurasi Sistem Kontrol Loop Terbuka
(Sumber : <http://www.digilib.itb.ac.id/> diakses pada 16 Agustus 2017)

Sistem kontrol loop tertutup merupakan konfigurasi rancangan sistem kontrol, dimana kondisi keluaran selalu diukur dan diumpanbalikan pada terminal input. Ilustrasi diagram blok sistem kontrol loop tertutup ditunjukkan melalui gambar berikut:



Gambar 2.2. Konfigurasi Sistem Kendali Loop Tertutup.
(Sumber : <http://www.digilib.itb.ac.id/> diakses pada 16 Agustus 2017)

Merujuk pada kedua gambar diatas, setiap jenis sinyal yang bekerja pada konfigurasi diagram blok terdiri dari : sinyal input (r), sinyal kontrol (p), sinyal output (q) dan sinyal umpan balik (b). Pada prakteknya konfigurasi rancangan

sistem kontrol yang sering digunakan adalah, konfigurasi sistem kontrol loop tertutup. Konfigurasi rancangan sistem kontrol loop tertutup lebih tahan terhadap gangguan dibandingkan rancangan sistem kontrol loop terbuka. Hal ini terjadi karena pada rancangan sistem kontrol loop tertutup respon keluaran selalu diawasi sehingga dapat mengkompensasi sinyal kontrol (u) yang akan diberikan pada plant. Dengan demikian peluang ketercapaian untuk mencapai nilai set point relatif lebih besar dibandingkan konfigurasi rancangan sistem kontrol loop terbuka (Aswan, 2008).

2.1.2. Arduino Mega 2560

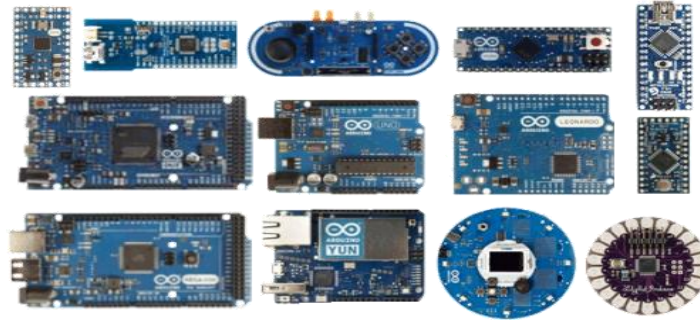
Arduino adalah kit elektronik atau papan rangkaian elektronik *open source* yang di dalamnya terdapat komponen utama yaitu sebuah chip mikrokontroler dengan jenis AVR dari perusahaan Atmel (Kadir, 2015)

Arduino memiliki kelebihan tersendiri dibanding *board* mikrokontroler yang lain.

Beberapa kelebihan yang dimiliki oleh arduino antara lain :

- a. Tidak perlu perangkat chip *programmer* karena di dalamnya sudah ada *bootloader* yang akan menangani *upload* program dari komputer.
- b. Memiliki sarana komunikasi USB, sehingga pengguna laptop yang tidak memiliki port serial/ RS323 bisa menggunakannya.
- c. Bahasa pemrograman relatif mudah karena *software* arduino dilengkapi dengan kumpulan *library* yang cukup lengkap.
- d. Memiliki modul siap pakai (*shield*) yang bisa ditancapkan pada *board* arduino, misalnya *shield* GPS, Ethernet, SD Card, dan lain-lain.

Karena sifat *open source*-nya ini, maka ada banyak varian arduino. Ada arduino uno, arduino mega, arduino yun, dan sebagainya. Masing-masing punya kelebihan dan keunikan tersendiri. Adapun, biasanya pemula menyukai arduino uno karena harganya yang relatif murah dan fitur-fitur yang relatif standar.



Gambar 2.3. Berbagai Macam Arduino
(Sumber : Kadir, 2015)

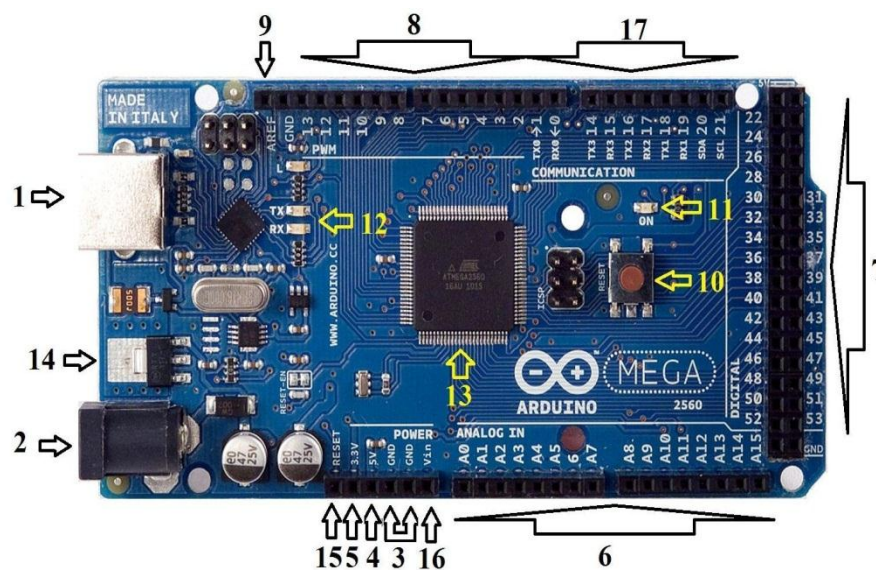
Arduino mega 2560 adalah sebuah *board* mikrokontroler yang berbasis ATmega2560. Arduino mega 2560 memiliki 54 pin digital *input/output* (15 di antaranya dapat digunakan sebagai *output* PWM), 16 pin *input* analog, 4 pin sebagai UART (*port serial hardware*), sebuah osilator kristal 16 MHz, sebuah koneksi USB, sebuah *power jack*, sebuah ICSP *header*, dan sebuah tombol *reset*. Gambar 2.4. merupakan contoh arduino mega 2560.



Gambar 2.4. Arduino Mega 2560
(Sumber : Kadir, 2015)

Arduino mega 2560 memiliki semua yang dibutuhkan untuk menunjang mikrokontroler. Sangat mudah menghubungkannya ke sebuah komputer, hanya dengan sebuah kabel USB atau menyuplainya dengan sebuah adaptor AC ke DC atau menggunakan baterai untuk memulainya. Arduino mega 2560 adalah versi terbaru yang menggantikan versi Arduino mega.

2.1.2.1. Bagian – bagian Arduino Mega 2560



Gambar 2.5. Bagian – bagian dari Arduino Mega 2560

(Sumber : Kadir, 2015)

Arduino mega 2560 memiliki beberapa bagian yang dapat dilihat pada gambar 2.5. Berikut adalah penjelasan dari bagian-bagian tersebut.

1. Port USB

Setiap *board* arduino membutuhkan jalur untuk terhubung ke sumber listrik. Arduino mega 2560 dapat diaktifkan melalui kabel USB yang berasal dari komputer. *Port* USB selain digunakan sebagai jalur listrik untuk mengaktifkan *board*, juga digunakan untuk meng-*upload* kode dari komputer ke *board* arduino.

2. DC Adapter Jack

Selain melalui port USB, arduino mega 2560 juga dapat diaktifkan melalui DC *adapter jack*. Saat program sudah ter-*upload* ke arduino, *board* arduino bisa berfungsi tanpa harus terhubung ke komputer via USB dengan menggunakan DC *adapter jack*. DC *adapter jack* ini bisa menerima tegangan 6 – 20 volt, namun yang disarankan adalah 7 – 12 volt.

3. Pin Ground

GND merupakan singkatan dari GROUND. Ada beberapa pin GND pada arduino, dan semuanya dapat digunakan untuk hubungan ke *ground* (kutub negatif). Seandainya terdapat komponen yang diberikan arus listrik melalui arduino, maka komponen tersebut juga harus dihubungkan ke GND.

4. Pin 5 Volt

Pin 5 volt digunakan untuk menyediakan tegangan sebesar 5 volt. Biasanya ada komponen-komponen yang membutuhkan arus listrik agar bisa berfungsi. Jika kebetulan tegangan yang dibutuhkan adalah 5 volt, maka pin ini bisa digunakan. Dan tentunya dihubungkan juga ke pin GND.

5. Pin 3,3 Volt

Selain pin 5 volt, terdapat pula pin 3 volt yang digunakan untuk menyediakan tegangan 3,3 volt. Sebagian besar komponen sederhana yang digunakan bersama arduino berjalan pada kisaran tegangan 3,3 volt hingga 5 volt. Oleh karena itu disediakan pula tegangan 3,3 volt pada arduino.

6. Pin Input Analog

Pin yang berada pada area berlabel “ANALOG IN” (A0 sampai A15 pada arduino mega 2560) digunakan sebagai pin analog. Yaitu pin yang digunakan

untuk membaca sinyal-sinyal analog dari sensor-sensor analog (misalnya sensor suhu) dan mengubahnya menjadi nilai digital yang dapat kita baca.

7. Pin Input / Output Digital

Pin digital berada di sebelah kanan pin analog (22 sampai 53 pada arduino mega 2560). Pin ini dapat digunakan dalam 2 arah digital yaitu *input* digital (misalnya untuk melihat kondisi bahwa tombol sedang ditekan) dan *output* digital (memberikan tegangan sebuah LED)

8. Pin Output PWM

Pin output PWM terletak di seberang pin analog (2 sampai 13 pada arduino mega 2560). Pin ini dapat digunakan sebagai pin digital biasa, tetapi juga dapat digunakan sebagai pin PWM (*Pulse Width Modulation*). PWM biasanya digunakan sebagai pin yang mampu mensimulasikan *output* analog (seperti mengatur pemudaran cahaya dan warna LED saat datang dan pergi). PWM bukan analog, hanya mampu mensimulasikan analog saja. Jadi tidaklah sama antara analog dengan PWM.

9. Pin AREF

Kepanjangan dari AREF adalah *Analog Reference* atau Referensi Analog. Pin ini terkadang digunakan sebagai referensi dalam mengatur tegangan eksternal (antara 0 sampai 5 volt) untuk memberikan *limit* (batasan akhir) pada *input* pin analog.

10. Tombol Reset

Menekan tombol *reset* beberapa saat pada board arduino akan menghubungkan pin *reset* ke *ground*. Ini digunakan untuk me-*restart* kode yang

telah dimuat oleh arduino. Ini sangat bermanfaat untuk menguji ulang kode, jika kode tersebut ditulis tanpa pengulangan (*repeat*) atau sekali jalan.

11. LED Indikator Daya

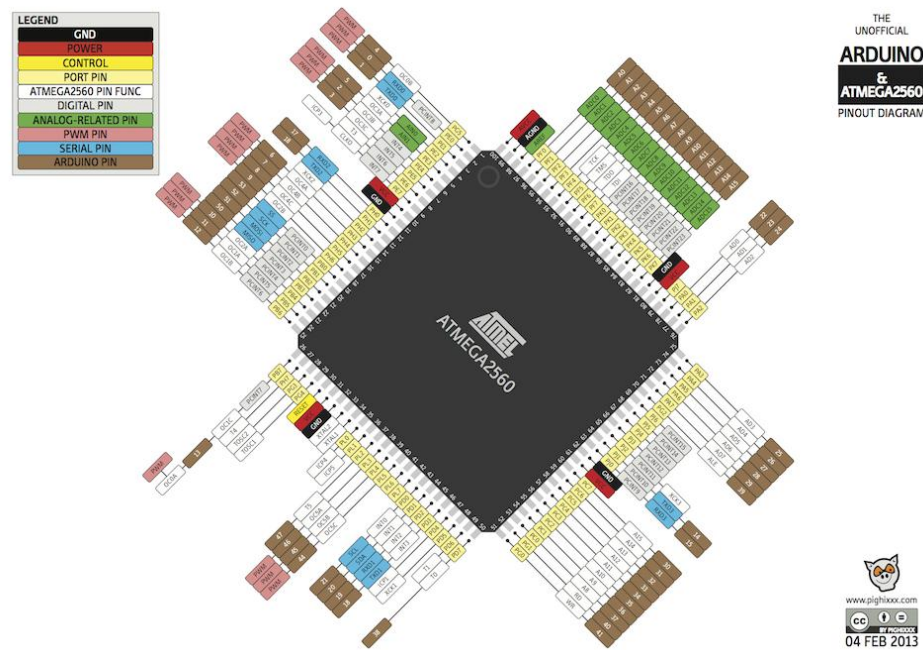
Tepat di atas kata “MEGA” pada *board* arduino mega 2560, terdapat sebuah LED kecil berlabel “ON”. LED ini harus menyala setiap kali *board* arduino dihubungkan ke sumber listrik. Jika lampu LED ini tidak menyala, kemungkinan *board* mengalami kerusakan atau terjadi kesalahan.

12. LED TX RX

TX adalah singkatan dari ‘*transmit*’ (kirim), dan RX adalah singkatan dari ‘*receive*’ (terima). LED indikator TX dan RX ini akan berkedip redup atau terang dengan jeda tak tentu untuk memberitahukan bahwa telah terjadi komunikasi serial. Kedipan LED ini sebagai indikasi visual yang merupakan pertanda baik bahwa telah terjadi pengiriman dan penerimaan data pada *board* arduino (misalnya ketika meng-*upload* kode baru ke dalam *board*).

13. IC Utama (ATmega 2560)

Sesuatu yang berwarna hitam dengan semua kakinya terbuat dari logam inilah yang disebut IC atau *Integrated Circuit*. Ini adalah otak dari *board* arduino. Pada dasarnya, pin pada *board* arduino adalah perpanjangan dari kaki-kaki yang ada di ATmega seperti yang diilustrasikan pada gambar 2.6 tentang hubungan pin ATmega2560 dengan pin arduino mega 2560.



Gambar 2.6. Hubungan Pin ATmega2560 dengan Pin Arduino Mega 2560 (Sumber : Kadir, 2015)

IC utama arduino berbeda-beda sesuai dengan jenis *board*, tetapi biasanya memiliki ciri khas bertuliskan ATmega yang merupakan IC buatan perusahaan ATMEL. Ini merupakan hal penting, karena kita perlu memahami jenis IC (sesuai juga dengan jenis *board* arduino) sebelum memuat atau meng-*upload* sekumpulan kode program yang telah dibuat dari perangkat lunak arduino IDE ke dalam *board* arduino. Informasi tentang tipe IC dapat ditemukan pada permukaan IC tersebut. Jika ingin mengetahui lebih jauh tentang perbedaan jenis IC yang digunakan pada keluarga arduino maka sebaiknya selalu unduh datasheet dari masing-masing tipe IC.

14. Regulator Tegangan

Regulator tegangan bukanlah sesuatu yang mampu berinteraksi dengan *board* arduino, tetapi harus diketahui fungsinya. Regulator tegangan akan

mengalirkan sejumlah tegangan teregulasi ke dalam *board* arduino. Regulator tegangan berperan sebagai penjaga pintu gerbang, dia akan membalikkan atau membuang tegangan berlebihan yang bisa membahayakan rangkaian. Tapi tentu saja ada batasnya, pastikan bahwa tegangan yang masuk ke dalam *board* tidak lebih besar dari 20 volt.

15. Pin Reset

Apabila tombol *reset* terhalang oleh sesuatu, maka kita dapat membuat tombol *reset* tambahan menggunakan pin ini.

16. Pin Vin

Selain menggunakan port USB atau DC *adapter jack*, arduino juga bisa diaktifkan menggunakan baterai. Yaitu dengan menghubungkan kutub positif baterai ke pin Vin, dan menghubungkan kutub negatifnya ke pin GND.

17. Pin Serial

Arduino Mega 2560 memiliki 4 port serial, sehingga dapat menggunakan lebih dari 1 modul serial, seperti modul GSM atau GPS misalnya, secara bersamaan. Pin RX digunakan untuk menerima data serial TTL, sedangkan pin TX digunakan untuk mengirimkan data serial TTL.

2.1.3. Solenoid

Solenoid adalah aktuator yang mampu melakukan gerakan linier. *Solenoid* dapat berupa elektromekanis (AC/DC), hidrolik atau pneumatik. Semua operasi berdasar pada prinsip-prinsip dasar yang sama. Dengan memberikan sumber tegangan maka *solenoid* dapat menghasilkan gaya yang linier (Widodo, 2006). Contohnya untuk menekan tombol, memukul tombol pada piano, operator katup,

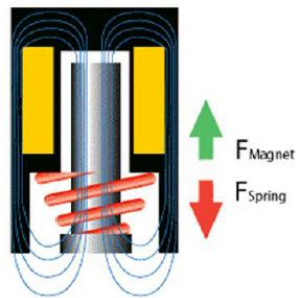
dan bahkan untuk robot melompat. *Solenoid* DC beroperasi pada prinsip-prinsip seperti motor DC. Perbedaan antara *solenoid* dan motor adalah bahwa *solenoid* adalah motor yang tidak dapat berputar. Berikut merupakan bentuk fisik *solenoid* yang digunakan, terdapat pada gambar 2.7. berikut.



Gambar 2.7. Bentuk Fisik Solenoid

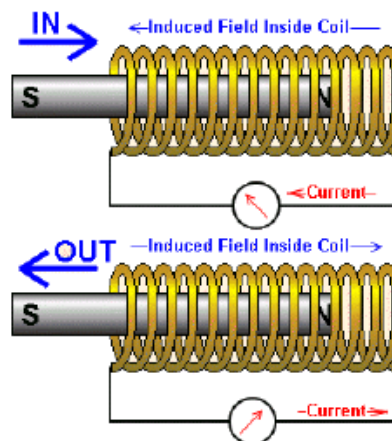
(Sumber : Ramakumbo, 2012)

Di dalam solenoida terdapat kawat melingkar pada inti besi. Ketika arus listrik melalui kawat ini, maka terjadi medan magnet untuk menghasilkan energi yang bisa mendorong inti besi. Poros dalam dari *solenoid* adalah piston seperti silinder terbuat dari besi atau baja, yang disebut *plunger* (setara dengan sebuah dinamo). Medan magnet kemudian menerapkan kekuatan untuk *plunger* ini, baik menarik atau *repeling* (kembali posisi). Ketika medan magnet dimatikan, pegas *plunger* kemudian kembali ke keadaan semula. Prinsip dari kerja *solenoid* tersebut seperti pada dijelaskan pada gambar berikut ini :



Gambar 2.8. Cara Kerja Solenoid
(Sumber : Ramakumbo, 2012)

Pergerakan *solenoid* juga ditampilkan seperti Gambar 2.9, yakni saat lilitan arus teraliri maka inti besi akan bergerak. Gerakan pada inti besi, mengikuti dari arah arus pada lilitan (Widodo, 2006)



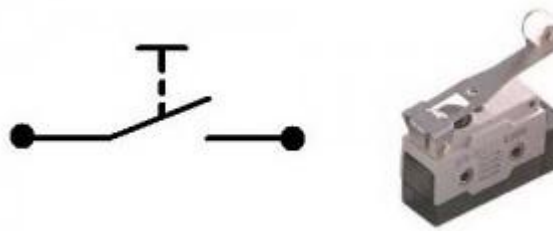
Gambar 2.9. Pergerakan Solenoid
(Sumber : Ramakumbo, 2012)

2.1.4. *Limit Switch*

Limit switch merupakan jenis saklar yang dilengkapi dengan katup yang berfungsi menggantikan tombol. Prinsip kerja *limit switch* sama seperti saklar *push on* yaitu hanya akan menghubungkan pada saat katupnya ditekan pada batas penekanan tertentu yang telah ditentukan dan akan memutus saat katup tidak

ditekan. *Limit switch* termasuk dalam kategori sensor mekanis yaitu sensor yang akan memberikan perubahan elektrik saat terjadi perubahan mekanik pada sensor tersebut. Penerapan dari *limit switch* adalah sebagai sensor posisi suatu benda (objek) yang bergerak (Firmansyah, 2009).

Prinsip kerja *limit switch* diaktifkan dengan penekanan pada tombolnya pada batas/ daerah yang telah ditentukan sebelumnya sehingga terjadi pemutusan atau penghubungan rangkaian dari rangkaian tersebut. Limit switch memiliki dua kontak yaitu NO (*Normally Open*) dan kontak NC (*Normally Close*) dimana salah satu kontak akan aktif jika tombolnya tertekan.



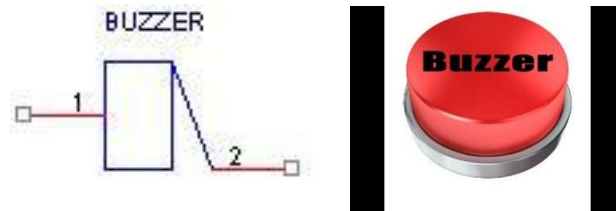
Gambar 2.10. Limit Switch

(Sumber : Firmansyah, 2009)

2.1.5. Buzzer

Buzzer adalah sebuah komponen yang memiliki fungsi mengubah arus listrik menjadi suara. Dan pada dasarnya prinsip kerja *buzzer* hampir sama dengan speaker. *Buzzer* terdiri dari sebuah diafragma yang memiliki kumparan. Ketika kumparan tersebut dialiri arus listrik sehingga menjadi elektromagnet, kumparan akan tertarik ke dalam atau ke luar tergantung dari polaritas magnetnya. Karena kumparan dipasang pada diafragma, maka setiap getaran diafragma secara bolak – balik membuat udara bergetar sehingga menghasilkan suara. *Buzzer* ini akan

digunakan sebagai indikator bunyi atau suara (alarm) dalam sebuah sistem. (St & Fahruzi, 2016)

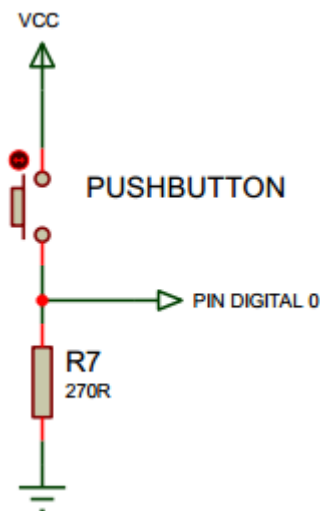


Gambar 2.11. Buzzer
(Sumber : St & Fahruzi, 2006)

2.1.6. Push Button

Push button switch (saklar tombol tekan) adalah perangkat / saklar sederhana yang berfungsi untuk menghubungkan atau memutuskan aliran arus listrik dengan sistem kerja tekan unlock (tidak mengunci). Sistem kerja unlock disini berarti saklar akan bekerja sebagai device penghubung atau pemutus aliran arus listrik saat tombol ditekan, dan saat tombol tidak ditekan (dilepas), maka saklar akan kembali pada kondisi normal.

Sebagai device penghubung atau pemutus, push button switch hanya memiliki 2 kondisi, yaitu On dan Off (1 dan 0). Istilah On dan Off ini menjadi sangat penting karena semua perangkat listrik yang memerlukan sumber energi listrik pasti membutuhkan kondisi On dan Off.



Gambar 2.12. Rangkaian Push Button

(Sumber : Ramakumbo, 2012)

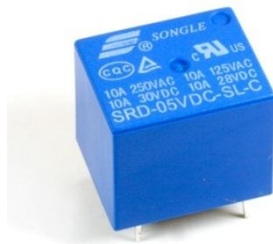
2.1.7. Relay

Relay adalah komponen elektronika berupa saklar elektronik yang digerakkan oleh arus listrik. Relay bekerja secara otomatis mengatur dan memasukan suatu rangkaian listrik akibat adanya perubahan rangkaian lain. Alat ini dapat berkeja dengan memanfaatkan gaya magnet dari coil yang terdapat dalam relay karena diberikan tegangan listrik. Adapun susunan kontak pada relay adalah sebagai berikut:

- *Normally Open*: Relay akan menutup bila dialiri arus listrik.
- *Normally Close* : Relay akan membuka bila dialiri arus listrik.

Relay ini memiliki kontak tengah yang akan melepaskan diri dan membuat kontak lainnya berhubungan. Relay adalah suatu peranti yang bekerja berdasarkan elektromagnetik untuk menggerakkan sejumlah kontaktor yang tersusun atau sebuah saklar elektronis yang dapat dikendalikan dari rangkaian elektronik lainnya dengan memanfaatkan tenaga listrik sebagai sumber energinya. Kontaktor akan tertutup (menyala) atau terbuka (mati) karena efek induksi magnet yang dihasilkan kumparan (induktor) ketika dialiri arus listrik. Berbeda dengan saklar, pergerakan kontaktor (on atau off) dilakukan manual tanpa perlu arus listrik. Relay memiliki prinsip kerja magnet dengan induksi listrik. Relay terdiri dari bagian bagian utama yaitu :

1. *Coil* atau Kumparan, merupakan gulungan kawat yang mendapa arus listrik adalah sejenis saklar yang pergerakannya tergantung dari ada tidaknya arus listrik di *coil*.
2. *Contact* atau Penghubung, adalah sejenis saklar yang pergerakannya tergantung dari ada tidaknya arus listrik di coil. *Contact* ada 2 jenis: *Normally Open* (kondisi awal sebelum diaktifkan open), dan *Normally Closed* (kondisi awal sebelum diaktifkan close). Adapun cara kerja relay adalah saat coil mendapatkan energy listrik akan menimbulkan gaya elektromagnetik gaya magnet yang ditimbulkan akan menarik plat/ *armature* berpegas (bersifat berlawanan), sehingga menghubungkan dua titik kontak (Muttaqin, Hafidudin, & Kurniawan, 2010).



Gambar 2.13. Relay

(Sumber : Dokumen Pribadi)

2.1.8. EMS Ethernet Module

Embedded Module System (EMS) Ethernet Module merupakan modul untuk mempermudah antarmuka antara komunikasi *ethernet* dengan mikrokontroler/ mikroprosesor. Modul ini bekerja dengan level tegangan TTL. Aplikasi dari modul ini adalah *embedded web server*, *embedded DHCP server*, serta aplikasi lainnya yang berhubungan dengan komunikasi *Ethernet* (Siregar, 2012). Gambar modul ini ditunjukkan pada gambar 2.14. di bawah ini.



Gambar 2.14. EMS Ethernet Module

(Sumber : Siregar, 2012)

2.1.9. Router

Router adalah piranti elektronik yang fungsinya mem-*forward* data antara jaringan komputer. *Router* adalah piranti dimana *software* dan *hardware* disetting untuk melakukan *routing* dan mem-*forward* informasi. *Router* akan

menghubungkan dua atau lebih subnet. *Routing* bekerja di level 3 dan berfungsi sebagai penghubung antar dua atau lebih jaringan untuk meneruskan dari satu jaringan ke jaringan lainnya (Rafiudin, 2003). Gambar berikut ini menunjukkan bentuk fisik dari *router*.



Gambar 2.15. Router

(Sumber : Rafiudin, 2003)

2.1.10. Modem

Modem yaitu singkatan dari modulator dan demodulator. Modulator berfungsi untuk melakukan proses menumpangkan data pada sinyal informasi ke sinyal pembawa agar dapat dikirim ke pengguna melalui media tertentu, proses ini biasa disebut dengan proses modulasi. Pada proses ini data dari komputer yang berbentuk sinyal digital akan diubah menjadi sinyal analog. Sedangkan demodulator berfungsi sebagai proses mendapatkan kembali data yang dikirim oleh pengirim. Pada proses ini data akan dipisahkan dari frekuensi tinggi dan data yang berupa sinyal analog akan diubah kembali menjadi sinyal digital agar bisa dibaca oleh komputer. Fungsi modem adalah untuk berkomunikasi dua arah yang merubah sinyal digital menjadi sinyal analog dan sebaliknya untuk mengirim

pesan atau data ke alamat yang dituju. Modem bisa juga diartikan sebagai perantara untuk menghubungkan komputer kita ke jaringan internet (Antok, 2016).



Gambar 2.16. Modem

(Sumber : AliExpress.com diakses 17 Agustus 2017)

2.1.11. Website

Website adalah sekumpulan halaman yang menampilkan konten atau sesuatu yang bisa diakses atau dibuka apabila kita mengakses internet. Sebutan *website* mempunyai halaman awal yaitu halaman yang pertama kali tampil apabila kita membuka alamat pada internet. Halaman pertama ini kerap di sebut *homepage*.

Website digunakan sekumpulan halaman yang menampilkan informasi dan menampilkan apa saja yang diinginkan oleh pemiliknya dengan harapan bisa diakses oleh siapa saja yang terkoneksi di internet dan menghubungi alamat *website* tersebut. Jenis *website* itu sendiri ada bermacam-macam mulai dari portal, *online shopping*, blog, forum, dan banyak jenis *website* adalah informasi dan komunikasi (Winoto & T, 2012)

Website merupakan kumpulan dari berbagai macam halaman situs, yang terangkum di dalam sebuah domain atau juga subdomain, yang lebih tepatnya berada di dalam WWW (*World Wide Web*) yang tentunya terdapat di dalam

Internet. Halaman *website* biasanya berupa dokumen yang ditulis dalam format *Hyper Text Markup Language* (HTML) yang bisa diakses melalui *Hyper Text Transfer Protocol* (HTTP). HTTP adalah suatu protokol yang menyampaikan berbagai informasi dari *server website* untuk ditampilkan kepada para *user* atau pemakai melalui *web browser* (Winoto & T, 2012).

Domain *website*, contohnya bisa di baca di bawah ini:

- .co.id : biasanya digunakan untuk badan usaha yang memiliki badan hukum sah.
- .go.id : khusus digunakan untuk lembaga pemerintahan RI.
- .ac.id : dipakai untuk lembaga pendidikan.
- .or.id : dipakai untuk segala macam organisasi yang tidak termasuk kedalam kategori ”co.id”, ”go.id”, ”mil.id”, ”ac.id” dan sebagainya.
- .war.net.id : dipakai untuk industri warung internet (warnet) yang ada di Indonesia
- .sch.id : dipakai khusus untuk lembaga pendidikan SD, SMP dan SMU atau SMK
- .web.id : biasanya digunakan untuk organisasi, badan usaha, ataupun perseorangan yang melakukan kegiatannya di WWW.



Gambar 2.17. Website

(Sumber : <http://www.ruangfreelance.com> diakses 17 Agustus 2017)

2.1.12. Sistem Operasi Android

Android adalah sistem operasi berbasis linux yang dirancang untuk perangkat seluler layar sentuh seperti telpon pintar dan komputer tablet (Utomo, 2012)



Gambar 2.18. Logo Android

(Sumber : Utomo, 2012)

Dimana pada awalnya Google Inc. membeli Android Inc., pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance*, Konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.

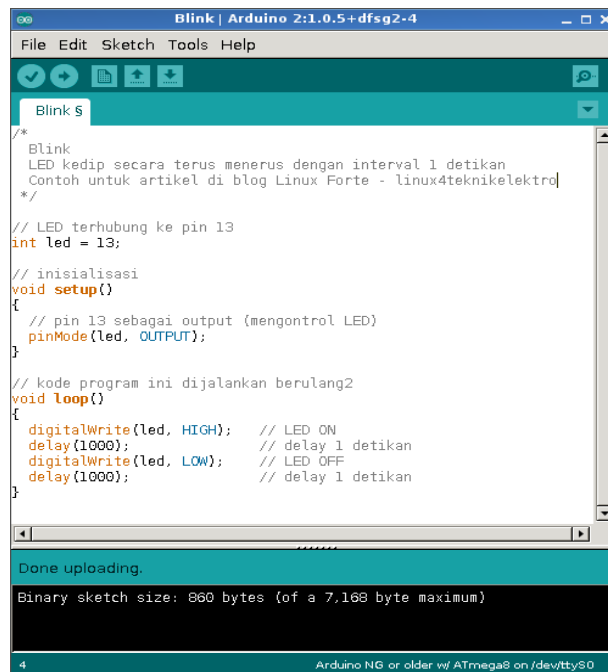
Jenis-jenis sistem operasi android :

- Android Versi 1.1 (Februari 2009)
- Android Versi 1.5 *Cupcake* (April 2009)
- Android *Donut* (Versi 1.6)
- *Android Eclair* (Versi 2.0 / 2.1)
- Android 2.2.x (*Froyo* : *Frozen Yoghurt*)
- *Android 2.3.x* (*Gingerbread*)
- Android 3.x (*Honeycomb*)
- Android 4.0.x (*Ice Cream Sandwich*)

- Android 4.4.x (*Kitkat*)

2.1.13. Arduino IDE

Arduino IDE adalah sebuah editor yang digunakan untuk menulis program, meng-*compile*, dan mengunggah ke papan arduino (Kadir, 2015) Arduino IDE adalah singkatan dari *Arduino Integrated Development Environment* berisi pesan, *console teks*, *toolbar* dengan tombol-tombol untuk fungsi umum dan sederetan menu. *Software* yang ditulis menggunakan arduino dinamakan *sketches*. *Sketches* ini ditulis di *editor teks* dan disimpan dengan *file* yang berekstensi *.ino*. *editor teks* ini mempunyai fasilitas untuk *cut/ paste* dan *search/ replace*. Area pesan berisi umpan balik ketika menyimpan dan mengunggah *file*, dan juga menunjukkan jika terjadi *error*.



Gambar 2.19. Editor Arduino IDE

(Sumber : Dokumen Pribadi)

2.1.14. *Go-Language (Golang)*

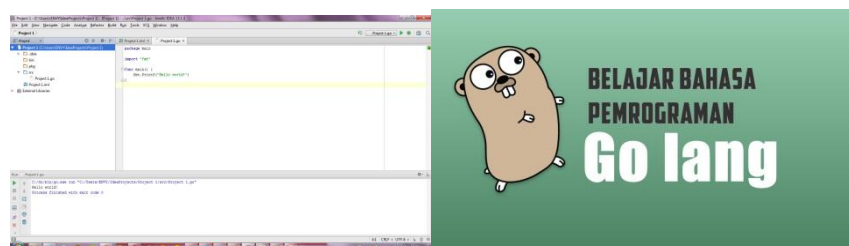
Golang (atau biasa disebut dengan Go) adalah bahasa pemrograman baru yang dikembangkan di Google oleh Robert Griesemer, Rob Pike, dan Ken Thompson pada tahun 2007 dan mulai diperkenalkan di publik tahun 2009. Penciptaan bahasa golang didasari bahasa C dan C++, oleh karena itu gaya sintaks-nya mirip.

Golang memiliki kelebihan dibanding bahasa lainnya, beberapa diantaranya :

- Mendukung konkurensi di level bahasa dengan pengaplikasian cukup mudah
- Mendukung pemrosesan data dengan banyak prosesor dalam waktu yang bersamaan (*pararel processing*)
- Memiliki *garbage collector*
- Proses kompilasi sangat cepat
- Bukan bahasa pemrograman yang hirarkial, menjadikan developer tidak perlu repot memikirkan segmen OOP-nya
- *Package/* modul yang disediakan terbilang lengkap. Karena bahasa ini *open source*, banyak sekali *developer* yang juga mengembangkan modul-modul lain yang bisa dimanfaatkan (Prayogo, 2016)

Go adalah bahasa pemrograman yang ekspresif, ringkas, bersih dan efisien. Saat ini perkembangan perangkat komputer sangat cepat namun tidak diimbangi dengan bahasa pemrograman, banyak *programmer* yang menemukan kerumitan pada Java atau C++ sehingga harus menuliskan beberapa kode lagi menggunakan Phyton atau JavaScript, disisi lain isu terhadap konsep *garbage-collection* dan

parallel-computation yang belum didukung baik oleh beberapa bahasa pemrograman, termasuk munculnya teknologi *multicore* yang membuat kekhawatiran dan kebingungan sehingga para pengembang masih mengandalkan dukungan teknologi dari sistem operasi.



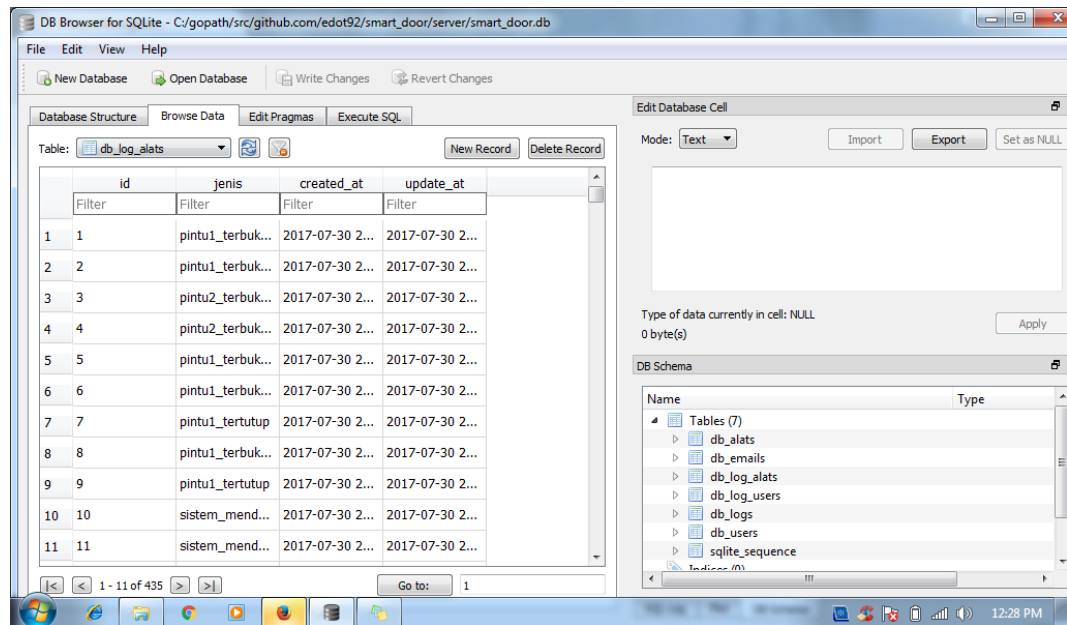
Gambar 2.20. Tampilan Golang

(Sumber : Prayogo, 2016)

2.1.15. SQLite

SQLite merupakan sebuah sistem manajemen basisdata relasional yang bersifat ACID-compliant dan memiliki ukuran pustaka kode yang relatif kecil, ditulis dalam bahasa C. SQLite merupakan proyek yang bersifat public domain yang dikerjakan oleh D. Richard Hipp. – Wikipedia

Jadi gampangnya, SQLite ini adalah mesin database SQL yang tertanam pada sistem yang kita gunakan. Tidak seperti pada paradigma client-server umumnya, inti SQLite bukanlah sebuah sistem yang mandiri yang berkomunikasi dengan sebuah program, melainkan sebagai bagian integral dari sebuah program secara keseluruhan.



Gambar 2.21. SQLite

(Sumber : Dokumen Pribadi)

2.2. Simbol Komponen Aktif dan Pasif

2.2.1. Komponen Aktif

Komponen aktif yaitu komponen yang tidak dapat bekerja tanpa adanya sumber tegangan serta komponen yang dapat menguatkan dan menyearahkan sinyal listrik, serta mengubah energi dari satu bentuk ke bentuk yang lain.

a. Transistor

Merupakan komponen elektronika yang terdiri dari tiga lapisan semikonduktor sebagai contoh NPN dan PNP. Transistor mempunyai tiga kaki yang disebut dengan Emitor (E), Basis (B), dan Kolektor (C) (tipe BJT).



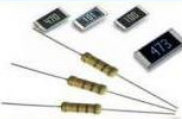
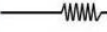










Gambar 2.24. IC (Integrated Circuit)
(Sumber : Prihono, 2009)

2.2.2. Komponen Pasif

Komponen pasif adalah komponen yang dapat bekerja tanpa sumber tegangan serta komponennya tidak dapat menguatkan dan menyearahkan sinyal listrik serta tidak dapat mengubah suatu energi ke bentuk lainnya.

a. Resistor

Resistor atau disebut juga dengan Hambatan adalah Komponen Elektronika Pasif yang berfungsi untuk menghambat dan mengatur arus listrik dalam suatu rangkaian Elektronika. Satuan Nilai Resistor atau Hambatan adalah Ohm (Ω). Nilai Resistor biasanya diwakili dengan Kode angka ataupun Gelang Warna yang terdapat di badan Resistor. Hambatan Resistor sering disebut juga dengan Resistansi atau Resistance.

Nama Komponen	Gambar	Simbol
Resistor (Nilai Tetap)		 atau 
Variable Resistor		 atau 
LDR (Light Depending Resistor)		 atau 
Thermistor (NTC / PTC)		 atau 

Gambar 2.25. Resistor
(Sumber : Prihono, 2009)

b. Kapasitor

Kapasitor atau disebut juga dengan Kondensator adalah Komponen Elektronika Pasif yang dapat menyimpan energi atau muatan listrik dalam sementara waktu. Fungsi-fungsi Kapasitor (Kondensator) diantaranya adalah dapat memilih gelombang radio pada rangkaian Tuner, sebagai perata arus pada rectifier dan juga sebagai Filter di dalam Rangkaian Power Supply (Catu Daya). Satuan nilai untuk Kapasitor (Kondensator) adalah Farad (F)

Nama Komponen	Gambar	Simbol
Kapasitor Biasa (Non-Polaritas)		
Kapasitor Elektrolit (memiliki Polaritas)		 atau 
Kapasitor Variabel (Variable Capacitor)		

Gambar 2.26. Kapasitor
(Sumber : Prihono, 2009)

c. Induktor

Induktor atau disebut juga dengan Coil (Kumparan) adalah Komponen Elektronika Pasif yang berfungsi sebagai Pengatur Frekuensi, Filter dan juga sebagai alat kopel (Penyambung). Induktor atau Coil banyak ditemukan pada Peralatan atau Rangkaian Elektronika yang berkaitan dengan Frekuensi seperti Tuner untuk pesawat Radio. Satuan Induktansi untuk Induktor adalah Henry (H).

Nama Komponen	Gambar	Simbol
Induktor (Nilai Tetap)		
Induktor Variabel (Variabel Coil)		

Gambar 2.27. Induktor
(Sumber : Prihono, 2009)

2.3 Kerangka Berpikir

Rancangan alat pengendali pintu dengan sistem keamanan berbasis web dan android ini bekerja menggunakan arduino mega 2560 sebagai kontrolernya dan modul ethernet sebagai serial antarmuka yang menghubungkan *smartphone* android dan web ke arduino mega 2560. Ketika *smartphone* android memberikan perintah agar pintu dari solenoid tersebut terbuka, maka arduino mega 2560 akan memprosesnya dan mengirimkan perintah ke solenoid untuk aktif dan membuka pintunya.

Jika ada orang yang mau masuk ke rumah dan terdeteksi oleh *limit switch* kemudian arduino mega 2560 mengirimkan sinyal ke *smartphone* untuk memasukkan *username* dan *password* ke *smartphone*. Jika *username* dan *password* itu benar maka arduino akan memerintahkan solenoid untuk membuka kuncinya, tetapi jika itu salah maka tidak akan bisa membuka kunci pintu rumah. Jika *username* dan *password* yang dimasukkan benar, maka buzzer akan berbunyi beep singkat 2x menandakan pintu siap dibuka, kunci terbuka, beep sekali menandakan *limit switch* aktif karena tertekan saat pintu terbuka, kemudian beep sekali lagi menandakan pintu siap ditutup kembali, kemudian pintu tertutup dan mengunci secara otomatis kembali.

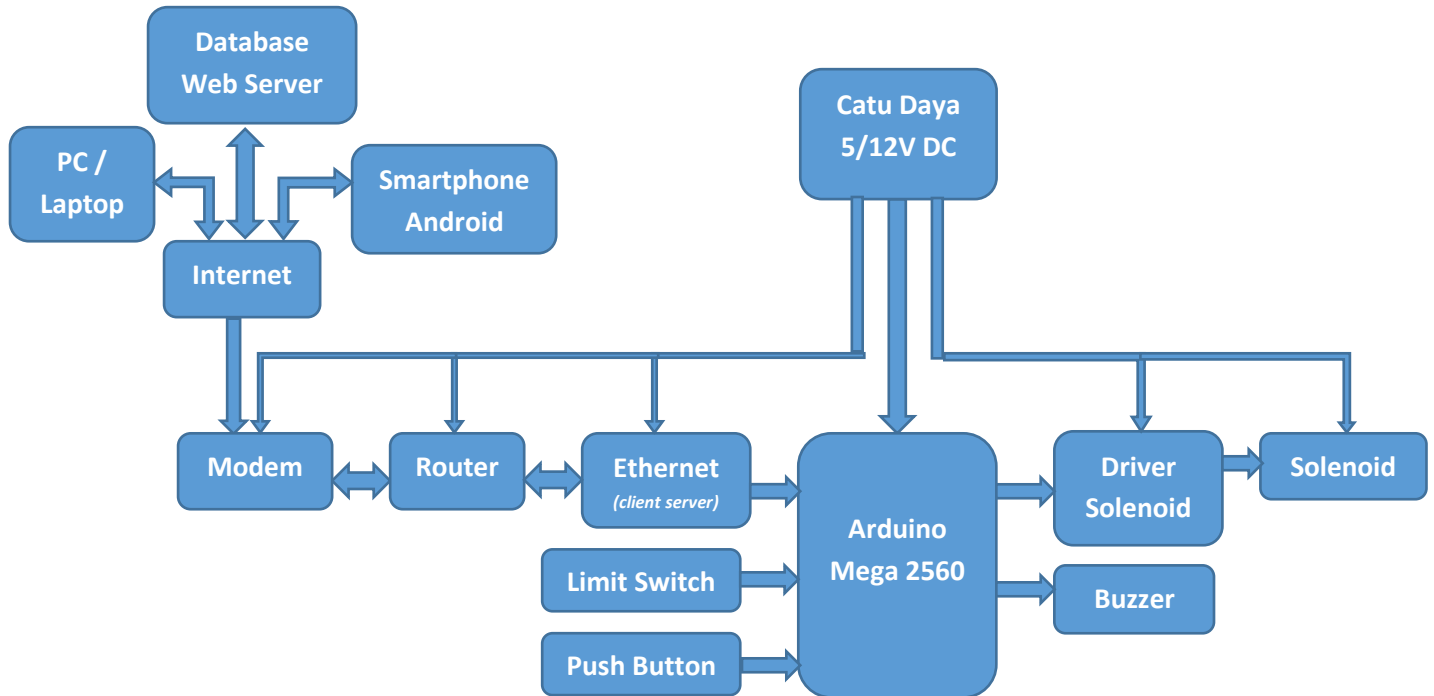
Untuk pemilik rumah yang sudah berada di dalam rumah dan ingin keluar rumah, pemilik cukup membuka pintu dengan cara menekan tombol keluar yang ada pada pintu bagian dalam, jadi tidak perlu repot lagi membuka kunci melalui aplikasi pada *smartphone* android atau web. Siklus membuka pintunya sama dengan membuka lewat aplikasi pada *smartphone* android atau web yang sudah dijelaskan di atas.

Namun jika ada seseorang mendobrak pintu rumah atau memaksa masuk dengan cara lain tanpa memasukkan *username* dan *password*, maka arduino akan memberikan perintah ke *solenoid* untuk tetap mengunci sehingga seseorang tersebut (dianggap maling) akan terperangkap di dalam rumah dan *solenoid* akan mengaktifkan *buzzer* sebagai alarm agar tetangga ataupun satpam kompleks yang ada di sekitar rumah tersebut mendengar bunyi alarmnya dan langsung mendatangi pelaku di rumah tersebut.

Jika tetangga rumah atau pemilik rumah ataupun satpam dari luar ingin menangkap maling yang ada di dalam rumah tersebut, caranya adalah dengan menekan tombol darurat yang ada di atas pojok pintu, sehingga buzzer bisa off dan kunci terbuka, setelah itu pintu akan menutup kembali dan mengunci secara otomatis.

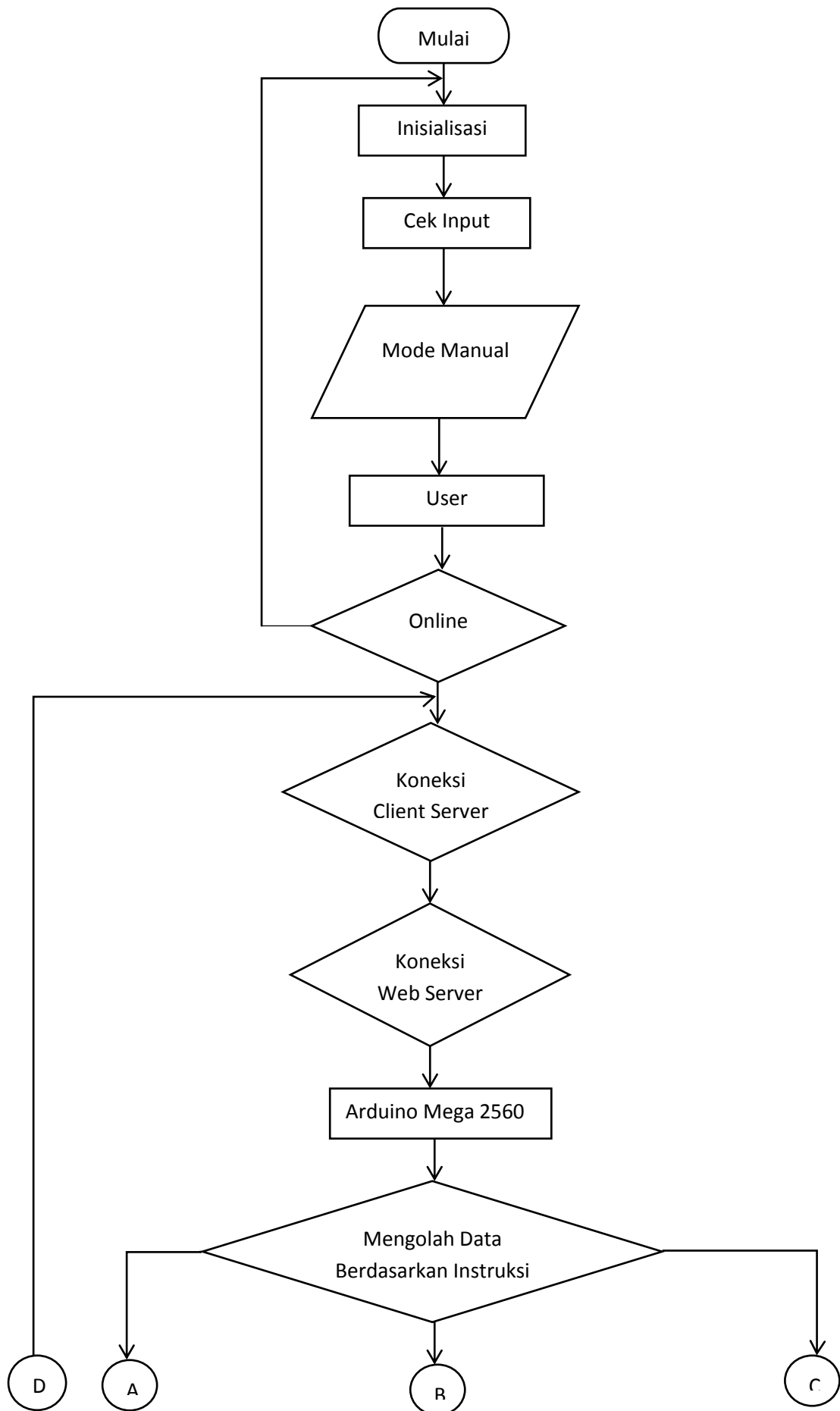
Hal lain yang perlu diperhatikan bahwa jika *smartphone* android kita tertinggal di dalam rumah atau di luar rumah (hilang) atau batrenya *off* atau mungkin rusak, maka alternatifnya adalah dengan menggunakan komputer atau laptop dan membuka *website* sebagai sarana lain pengganti aplikasi di *smartphone* tersebut. Sedangkan jika listrik di rumah sedang padam namun pintu masih mengunci karena sistem masih aktif ketika penghuni rumah sedang di luar dan listrik masih menyala, alternatifnya adalah dengan membuka pintu secara manual menggunakan kunci manual elektronik.

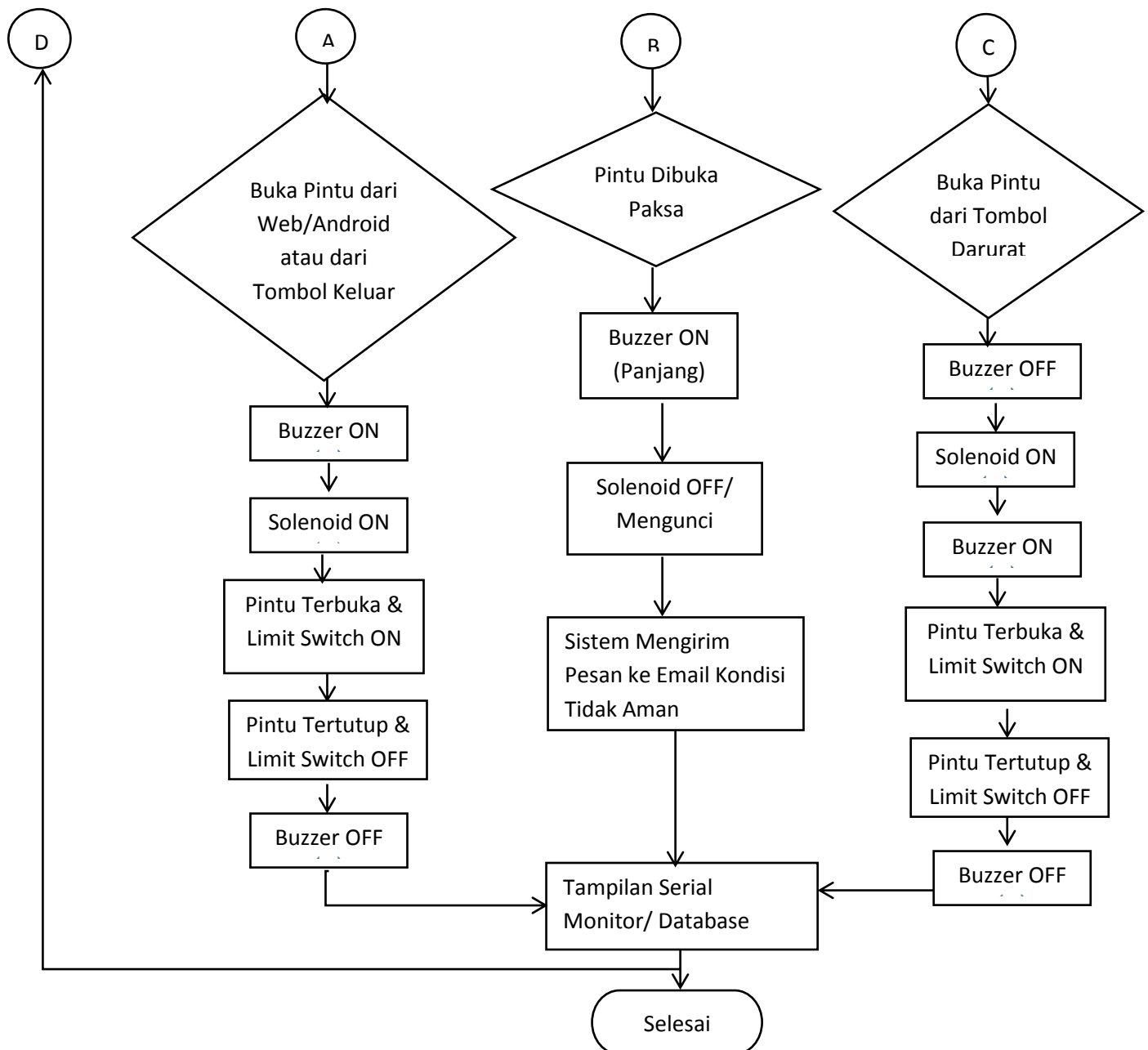
2.3.1. Blok Diagram



Gambar 2.28. Blok Diagram

2.3.2. Flowchart Sistem





Gambar 2.29. Flowchart

2.4. Penelitian Terkait

Penelitian pengendali pintu ini sebelumnya pernah dilakukan oleh Andrian Muttaqin dkk. dari Universitas Telkom Bandung yaitu dengan judul pengecekan pintu dan pengunci otomatis jarak jauh berbasis SMS gateway dan mikrokontroler. Alat yang dibuat berfungsi sebagai pengecekan pintu serta

pengunci otomatis jarak jauh berbasis mikrokontroler AVR AT-Mega 16. Sistem kerja dari alat ini yaitu mikrokontroler mengatur Tx dan Rx dengan modem Icomsat v1.1 sehingga dapat berkomunikasi. Lalu lcd akan menampilkan indikator modem siap atau tidaknya dan untuk memunculkan SMS terkirim atau tidaknya.

Hasil yang diperoleh dari pembuatan alat ini yaitu dapat mengendalikan sebuah kunci secara otomatis dengan mengirim *password* menggunakan SMS, dan jika kode *password* yang dimasukkan benar, maka mikrokontroler akan memberikan *input high* pada relay untuk mengaktifkan solenoid. Dari hasil pengujian yang telah dilakukan menunjukkan bahwa alat ini mampu membuka, mengunci serta mengecek keadaan kunci pintu dengan SMS.

Penelitian yang lainnya juga pernah dilakukan oleh Zulfauzi dari STMIK MURA Lubuklinggau dengan judul penelitian rancang bangun pintu geser otomatis berbasis android menggunakan jaringan wifi. Rancang bangun alat yang dibuat oleh beliau masih berupa *prototipe* dengan pintu gesernya menggunakan slot DVD dan pembatasan masalahnya hanya pada membuka dan menutup pintu secara otomatis yang digerakkan oleh motor DC sebagai penggerakannya dan dihubungkan ke arduino sebagai kontrolernya serta wifi sebagai serial antarmukanya yang menghubungkan ke HP android sebagai tampilan sistemnya.

BAB III

METODOLOGI PENELITIAN

3.1. Tempat dan Waktu Penelitian

Penelitian ini dilakukan di Laboratorium Instrumentasi Kendali Gedung L Teknik Elektro, Universitas Negeri Jakarta. Waktu penelitian dilaksanakan pada bulan Januari – Juli 2017.

3.2. Alat dan Bahan Penelitian

Alat penelitian yang diperlukan dalam pembuatan pengendali pintu dengan sistem keamanan berbasis web dan android adalah sebagai berikut.

- 1) Pintu ukuran 41 cm x 75 cm 2 buah
- 2) Peralatan bengkel
- 3) Alat ukur : multimeter

Bahan penelitian yang diperlukan dalam pembuatan pengendali pintu dengan sistem keamanan berbasis web dan android adalah sebagai berikut.

3.2.1. *Hardware* :

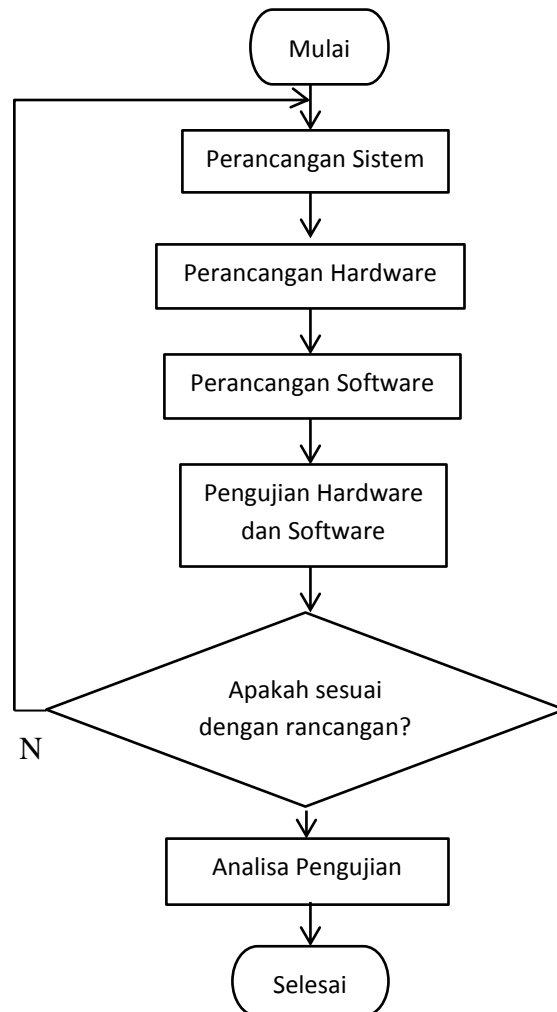
- | | |
|-----------------------|--------------------|
| 1) Arduino Mega 2560 | 7) PC / laptop |
| 2) Solenoid | 8) Router |
| 3) Driver Solenoid | 9) Modem |
| 4) Limit Switch | 10) Modul Ethernet |
| 5) Buzzer | 11) Push Button |
| 6) Smartphone Android | 12) Regulator |

3.2.2. Software :

- 1) Arduino IDE
- 2) Go-Language
- 3) SQLite
- 4) Google Sketch-Up
- 5) PCB Wizard
- 6) Web2apk

3.3. Diagram Alir Penelitian

Metode penelitian yang digunakan pada penelitian alat pengendali pintu dengan sistem keamanan berbasis web dan android ini adalah rekayasa teknik. Metode penelitian rekayasa teknik terdiri atas perancangan sistem, perancangan *hardware*, perancangan *software*, pengujian *hardware* dan *software*, analisa pengujian.



Gambar 3.1. Diagram Alir Metode Penelitian Alat

a. Perancangan Sistem

Dalam tahap ini peneliti membuat rancangan sistem berupa rangkaian elektronik dalam pembuatan alat pengendali pintu dengan sistem keamanan berbasis web dan android. Pada tahap ini peneliti menuangkan pikiran dan perancangan sistem terhadap solusi dari permasalahan yang ada dengan menggunakan perangkat pemodelan sistem seperti desain sistem dan prinsip kerja alat.

b. Perancangan *Hardware*

Dalam perancangan *hardware* ini menentukan keberhasilan suatu sistem. Dalam perancangan hardware atau perangkat keras dirancang agar sesuai dengan kebutuhan Pengendali Pintu Dengan Sistem Keamanan Berbasis Web dan Android berjalan dengan baik dan sesuai dengan yang diharapkan.

c. Perancangan *Software*

Perancangan *software* adalah merancang sekumpulan data elektronik berupa program atau instruksi yang akan menjalankan suatu perintah. Melalui perangkat lunak arduino IDE, *SQLite* dan Golang, maka alat ini dapat menjalankan suatu perintah.

d. Pengujian *Hardware* dan *Software*

Pengujian *hardware* dan *software* dilakukan untuk mengetahui tingkat keberhasilan alat yang dibuat. Jika *hardware* mengalami masalah atau terdapat error, maka alat tidak akan bekerja sempurna dan begitupula sebaliknya jika terdapat masalah pada *software*, maka alat juga tidak bisa dikerjakan sesuai tujuan. Maka dari itu pengujian *hardware* dan *software* sangat diperlukan dalam penelitian ini.

e. Analisa Pengujian

Analisa pengujian merupakan teknik yang digunakan dalam memecahkan masalah pada saat perancangan alat setelah dilakukan pengujian *hardware* dan *software* serta untuk mendapatkan data yang diperlukan pada sistem, kriteria pengujian diperlukan untuk menyatakan berhasil atau tidaknya penelitian alat ini.

3.3.1. Perancangan Hardware

3.3.1.1. Menentukan Jenis Pintu

Dalam melakukan perancangan dan pembuatan alat pengendali pintu dengan sistem keamanan berbasis web dan android ini terlebih dahulu harus ditentukan jenis pintu yang akan digunakan. Pada pembuatan alat ini akan digunakan pintu dengan jenis *swing*. Jenis pintu *swing* ini adalah jenis pintu yang umum digunakan oleh masyarakat untuk digunakan sebagai pintu rumah maupun bangunan fasilitas umum. Pintu jenis *swing* ini atau yang biasa disebut dengan pintu kupu-kupu adalah pintu yang bisa ditarik ke depan maupun ke belakang dan dapat berputar searah ataupun dua arah. Jenis pintu *swing* ini seperti terlihat pada gambar 3.2. di bawah ini.



Gambar 3.2. Pintu *Swing*

3.3.1.2. Menentukan Sistem Kendali

Dalam pembuatan rancangan alat pengendali pintu dengan sistem keamanan berbasis web dan android dikendalikan oleh Arduino Mega 2560. Spesifikasi dari Arduino Mega 2560 akan dijelaskan dalam tabel 3.1. berikut.

Tabel 3.1. Spesifikasi Arduino Mega 2560

Mikrokontroler	ATmega2560
Tegangan Operasi	5 Volt
Tegangan <i>Input</i> (Disarankan)	7 - 12 Volt
Tegangan <i>Input</i> (Batas bawah - atas)	6 – 20 Volt
Jumlah Pin I/O Digital	54 (15 pin sebagai <i>output</i> PWM)
Jumlah Pin <i>Input</i> Analog	16
Arus DC per pin I/O	40 Ma
Arus DC untuk pin 3,3 V	50 Ma
<i>Flash Memory</i>	256 KB (8 KB untuk <i>bootloader</i>)
SRAM	8 KB (ATmega328)
EEPROM	4 KB (ATmega328)
<i>Clock Speed</i>	16 MHz
Panjang	101,98 mm
Lebar	53,63 mm
Tinggi	15,29 mm
Berat	34,9 g
Konektor USB	Tipe B
<i>Windows Compatibility</i>	8 (32 & 64 bit), 7 (32 & 64 bit), Vista (32 & 64 bit), XP (32 & 64 bit)
<i>Mac Compatibility</i>	OS X
<i>Linux Compatibility</i>	Ya



Gambar 3.3. Arduino Mega 2560

3.3.1.3. Menentukan *Solenoid*

Pada penelitian alat pengendali pintu rumah dengan sistem keamanan berbasis web dan android, peneliti menggunakan *solenoid* yang mempunyai sistem kerja *normally close* atau NC yaitu dengan cara kerja *solenoid* akan menarik jika ada tegangan dan memanjang jika tidak ada tegangan yang mengalir pada *solenoid*. Tegangan yang diberikan pada solenoid ini yaitu 12 Volt agar solenoid dapat bekerja.

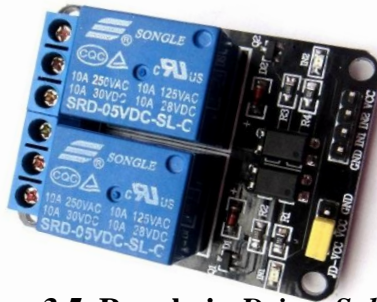


Gambar 3.4. Solenoid NC

3.3.1.4. Perancangan Rangkain *Driver Solenoid*

Rangkaian *driver solenoid* berfungsi untuk memungkinkan penggunaan arus kecil yang didapat dari arduino untuk mengontrol arus dan tegangan yang lebih besar guna mengurangi beban kerja pada *solenoid*. Spesifikasinya yaitu 5V *voltage driver* sehingga bisa langsung terhubung dengan mikrokontroler, 1-

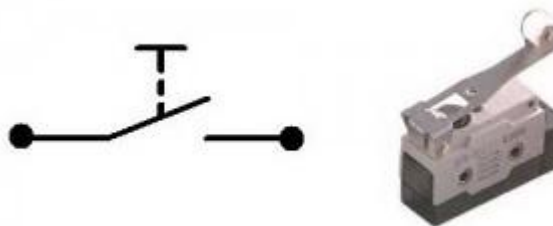
Channel Relay, Max relay load : AC250V 10A ; DC30V 10A, *Interface standard TTL logic, optocoupler isolation, LED indicator.* Rangkaian *driver solenoid* ini dapat dilihat pada gambar 3.5. di bawah ini.



Gambar 3.5. Rangkain Driver Solenoid

3.3.1.5. Menentukan *Limit Switch*

Pada penelitian alat pengendali pintu dengan sistem keamanan berbasis web dan android, peneliti menggunakan *limit switch* berjenis NC sebagai sensor jarak yang mendeteksi adanya obyek di depan pintu dan terlebih ketika pintu didobrak. *Limit switch* NC (*Normally Close*) yaitu terjadi pada saat saklar tidak tertekan, saklar dalam keadaan terhubung (ON). Sebaliknya, pada saat saklar tertekan, kondisi tidak terhubung (OFF).



Gambar 3.6. Limit Switch

3.3.1.6. Menentukan *Buzzer*

Pada penelitian alat pengendali pintu dengan sistem keamanan berbasis web dan android, peneliti menggunakan *passive piezoelektrik buzzer* sebagai alarm penanda terjadinya suatu keadaan pada rumah khususnya dalam penelitian

kali ini yaitu pintu jika didobrak oleh obyek yang dideteksi oleh sensor. Spesifikasinya sebagai berikut.

- Frekuensi : 2K-5K *to drive*
- Transistor *board* : 9012 *drive*
- Tegangan Kerja : 3.3V - 5V
- Ukuran PCB : 3.3cm x 1.3cm
- Vcc : 3.3V – 5V
- GND : Ground
- I/O : I/O *Interface (Active Low)*



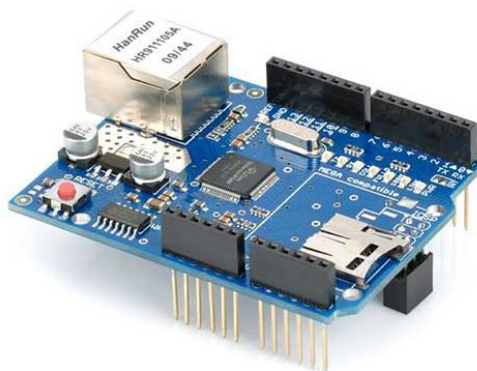
Gambar 3.7. Passive Piezoelektrik Buzzer

3.3.1.7. Menentukan Ethernet

Pada penelitian alat pengendali pintu dengan sistem keamanan berbasis web dan android, peneliti memilih menggunakan modul Ethernet dengan spesifikasi sebagai berikut.

1. Berbasis IC ENC28J60, *ethernet controller* yang kompatibel dengan IEEE 802.3™ dan jaringan 10/100/1000 Base-T.
2. Memiliki MAC dan 10 Base-T PHY yang terintegrasi.
3. Memiliki 1 *port* 10 Base-T (konektor MAC *jack*) dengan kemampuan deteksi dan koreksi polaritas secara otomatis.
4. Mendukung mode *Full* dan *Half-Duplex*.
5. Pengiriman ulang secara otomatis (dapat diprogram).
6. Perhitungan CRC (dapat diprogram).

7. Penolakan paket yang salah secara otomatis (dapat diprogram).
8. Antarmuka ke host controller melalui SPI dengan kecepatan sampai dengan 20 MHz.
9. Mempunyai 8 KByte SRAM sebagai *buffer* bagi paket data yang akan dikirimkan sekaligus bagi paket data yang akan diterima (*shared buffer*).
10. Mendukung paket data *unicast*, *multicast*, dan *broadcast*.
11. Memiliki alamat MAC yang dapat diprogram.
12. Memiliki *filter* untuk paket data yang diterima.
13. Kecepatan transfer data *ethernet* sampai dengan 10Mb/s. Memiliki 2 indikator LED untuk menunjukkan status koneksi.
14. Memiliki pin interupsi untuk menginformasikan adanya paket data yang telah diterima.
15. Memiliki pin *clock* yang dapat menghasilkan frekuensi hingga 25 Mhz pada level tegangan TTL.
16. Memiliki fasilitas *power-down*.
17. Membutuhkan catu daya 5 Volt DC.
18. Kompatibel dengan DT-AVR Low Cost Series dan mendukung sistem kontroler lainnya.



Gambar 3.8. EMS Ethernet Module

3.3.1.8. Menentukan Router

Pada penelitian alat pengendali pintu dengan sistem keamanan berbasis web dan android menggunakan router, peneliti menggunakan router tipe TP-Link TL-MR3020. Berikut spesifikasi router TP-Link TL-MR3020 :

- 1) Berbagi koneksi mobile 3G / 4G *, kompatibel dengan modem USB 120+ LTE / HSPA + / HSPA / UMTS / EVDO 3G / 4G, yang diuji di lapangan
- 2) Desain travel-size, kecil dan ringan untuk dibawa ke jalan, memungkinkan pengguna berbagi koneksi seluler 3G / 4G * di manapun cakupan 3G / 4G * tersedia.
- 3) Kecepatan wireless hingga 150Mbps
- 4) Tiga mode tersedia untuk situasi yang berbeda: Router 3G / 4G *, WISP Client Router dan Travel Router Mode (AP Mode)
- 5) Dengan strategi back-up 3G / 4G * dan WAN, TL-MR3020 dapat menyediakan koneksi internet yang berkelanjutan.



Gambar 3.9. Router TP-Link TL-MR3020

3.3.1.9. Menentukan Modem

Pada penelitian alat pengendali pintu dengan sistem keamanan berbasis web dan android menggunakan router, peneliti menggunakan Modem Huawei model E160 dengan spesifikasi sebagai berikut.

- HSDPA/UMTS?900/2100MHz
- GSM/GPRS/EDGE?850/900/1800/1900MHz
- *Support 7.2Mbps HSDPA services*
- *MicroSD Card Slot*
- *Support PC voice (optional) and SMS services*
- *Support data statistics*
- *Plug & Play*
- *Support Windows 2000/XP/Vista and Mac Operating Systems*



Gambar 3.10. Modem

3.3.1.10. Menentukan *Smartphone*

Pada penelitian alat pengendali pintu dengan sistem keamanan berbasis web dan android menggunakan router, peneliti menggunakan *smartphone*

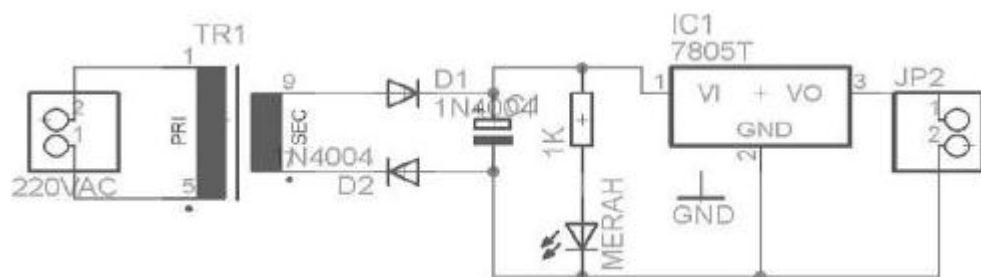
android yang dapat digunakan memiliki persyaratan minimum yaitu memiliki sistem operasi *android* 4.0.x (*Ice Cream Sandwich*)



Gambar 3.11. Smartphone Android Ice Cream Sandwich

3.3.1.11. Menentukan Regulator

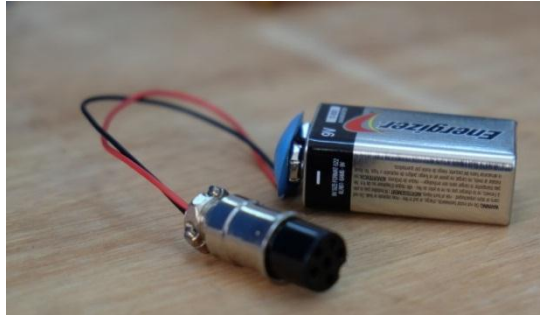
Untuk mengaktifkan rangkaian-rangkaian elektronika pada sistem pengendali pintu berbasis keamanan menggunakan *web* dan *android* ini, peneliti menggunakan regulator Untuk mengaktifkan rangkaian-rangkaian elektronika pada sistem pengendali pintu berbasis keamanan menggunakan *web* dan *android* ini dibutuhkan *supply* tegangan dari catu daya sebesar +5V. rangkaian catu daya ini terdiri dari sebuah transformator dengan keluaran 12Volt, penyearah arus *diode bridge*, *filter (elcho)* dan IC regulator LM7805.



Gambar 3.12. Regulator

3.3.1.12. Menentukan Kunci Manual Elektronik

Kunci manual elektronik ini berfungsi untuk membuka kunci pintu rumah ketika listrik mati. Dengan menggunakan baterai 9V kunci manual ini bisa membuka solenoid. Kunci manual elektronik dapat dilihat pada gambar 3.13. di bawah ini.



Gambar 3.13. Kunci Manual Elektronik

3.3.1.13. Menentukan *Door Closer*

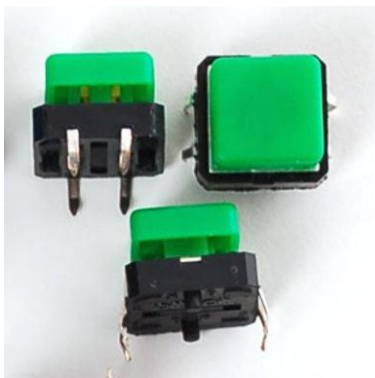
Dalam melakukan pembuatan alat pengendali pintu dengan sistem keamanan berbasis web dan android ini, peneliti membuat door closer atau penutup pintu menggunakan sistem katrol yaitu tali yang dikaitkan ke balok kemudian diberi beban disamping pintu yang dibungkus dengan box hitam agar dapat menutup secara otomatis ketika pintu dibuka.



Gambar 3.14. Door Closer Manual

3.3.1.14 Menentukan Push Button

Dalam melakukan pembuatan alat pengendali pintu dengan sistem keamanan berbasis web dan android ini, peneliti membuat tombol keluar dan tombol darurat menggunakan push button yang bisa dilihat pada gambar 3.15. dan gambar 3.16. berikut.



Gambar 3.15. Tombol Keluar

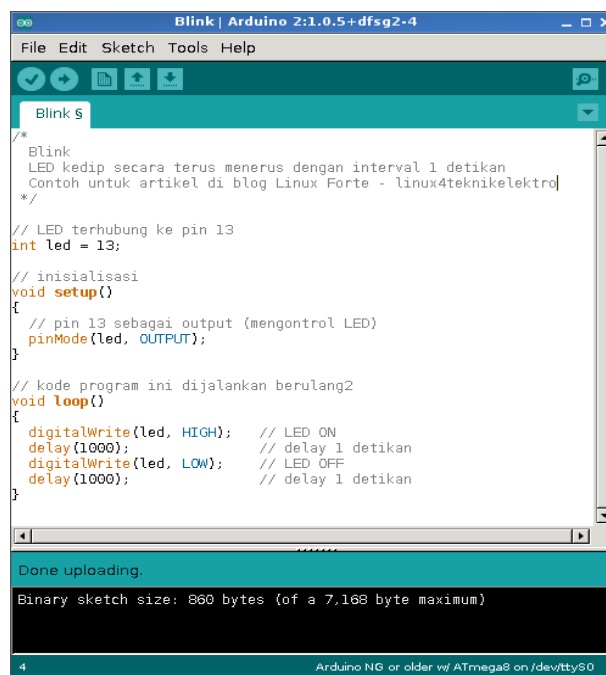


Gambar 3.16. Tombol Darurat

3.3.2. Perancangan Software

3.3.2.1. Arduino IDE

Dalam melakukan pembuatan alat pengendali pintu dengan sistem keamanan berbasis web dan android ini, peneliti terlebih dahulu harus menentukan sistem kendalinya. Sistem kendali yang digunakan untuk pembuatan alat ini adalah Arduino Mega 2560 dan software pemrograman yang digunakan adalah Arduino IDE (*Integrated Development Environment*)

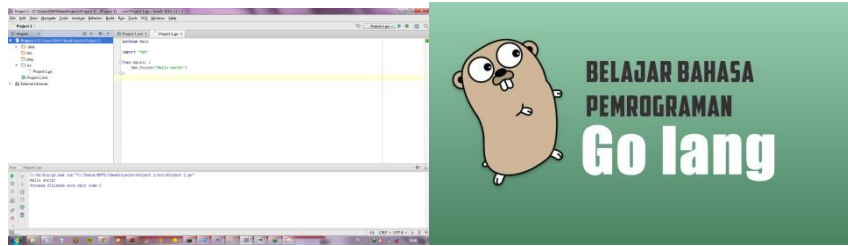


```
Blink | Arduino 2:1.0.5+dfsg2-4
File Edit Sketch Tools Help
Blink $
/*
 * Blink
 * LED kedip secara terus menerus dengan interval 1 detikan
 * Contoh untuk artikel di blog Linux Forte - linux4teknikelektro
 */
// LED terhubung ke pin 13
int led = 13;
// inisialisasi
void setup()
{
  // pin 13 sebagai output (mengontrol LED)
  pinMode(led, OUTPUT);
}
// kode program ini dijalankan berulang2
void loop()
{
  digitalWrite(led, HIGH); // LED ON
  delay(1000); // delay 1 detikan
  digitalWrite(led, LOW); // LED OFF
  delay(1000); // delay 1 detikan
}
Done uploading.
Binary sketch size: 860 bytes (of a 7,168 byte maximum)
4 Arduino NG or older w/ ATmega8 on /dev/ttyS0
```

Gambar 3.17. Tampilan Arduino IDE

3.3.2.2. Go-Language (Golang)

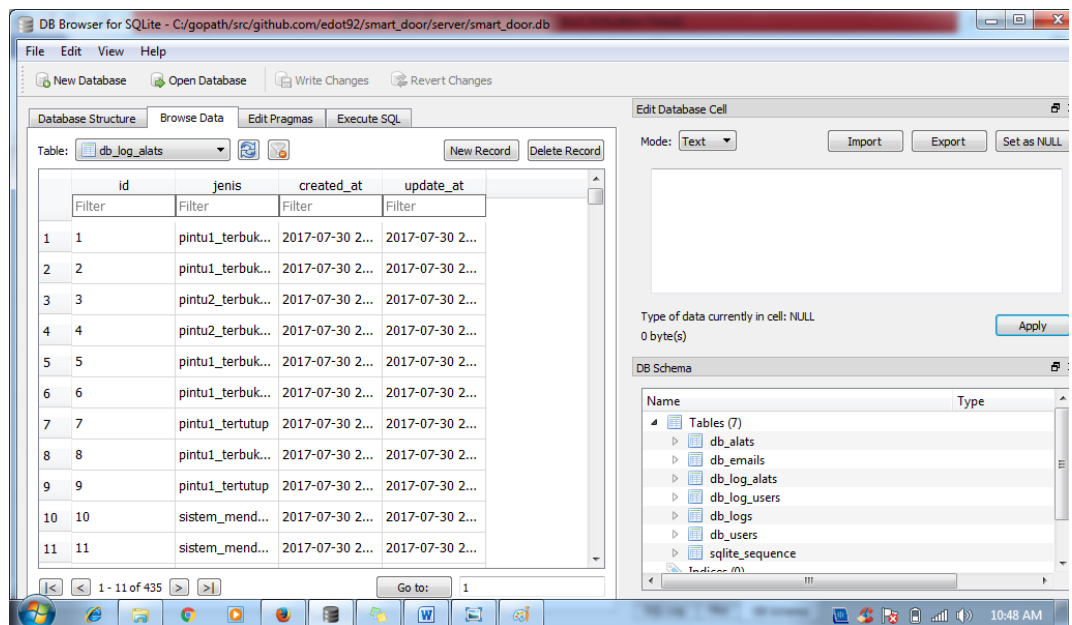
Dalam melakukan pembuatan alat pengendali pintu dengan sistem keamanan berbasis web dan android ini, peneliti menggunakan *software Go-language* (Golang) untuk *view website* dan *database* sistem.



Gambar 3.18. Go-Language (Golang)

3.3.2.3. SQLite

Dalam melakukan pembuatan alat pengendali pintu dengan sistem keamanan berbasis web dan android ini, peneliti menggunakan aplikasi *software SQLite* untuk pemrograman *database* sistem yang dibuat yang akan bekerjasama dengan web *server* yang ada. Penelitian rancang bangun alat ini menggunakan database yang harus dibuat sistem keamanannya untuk setiap individu atau pengguna (*user*) dalam pengaksesannya, untuk itu peneliti membuat sistem *login* agar lebih aman dan tidak digunakan oleh sembarang pihak.



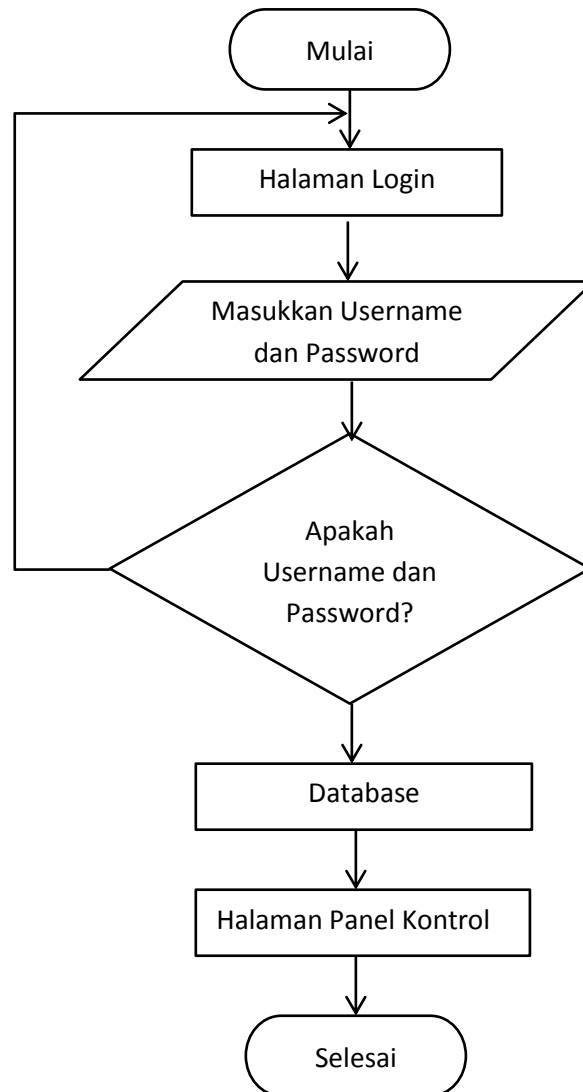
Gambar 3.19. SQLite Database



Gambar 3.20. Tampilan Halaman *Login*

Untuk membuat tampilan halaman-halaman web seperti gambar di atas dan berfungsi sebagai sistem pengendali, tahapan yang perlu dilakukan yaitu membuat *database* tampilan pada kontrol panel yang telah disediakan, fungsi dari *database* tampilan adalah untuk mendesain tampilan yang ada pada halaman *login* dan panel kontrol.

Berikut adalah gambar diagram alir untuk halaman *login*.

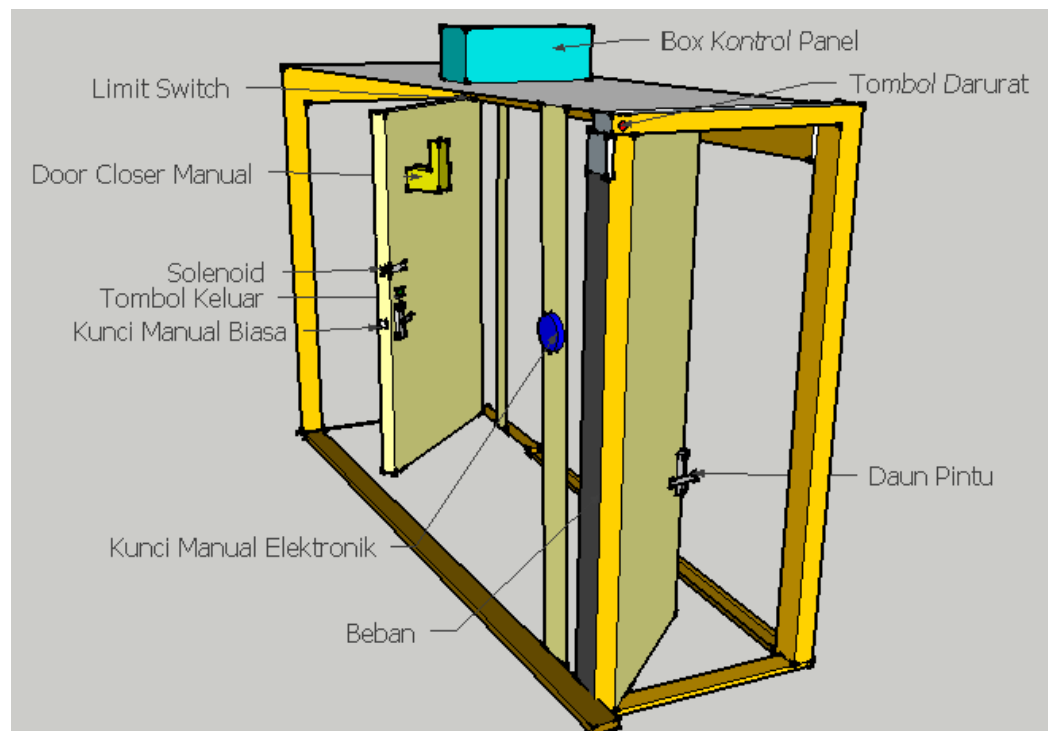


Gambar 3.21. Diagram Alir Halaman Login

Berdasarkan diagram di atas, untuk mengakses halaman web untuk menuju kontrol panel harus login terlebih dahulu menggunakan *username* dan *password* yang tersedia.

3.3.3. Menentukan Desain Maket Alat

Peneliti merancang desain maket alat yang terbuat dari 2 buah pintu kayu jenis *swing* berwarna coklat-putih dengan ukuran 41 x 75 cm yang dilengkapi dengan *door closer* agar pintu dapat menutup secara otomatis. Berikut rancangan desain maket alat serta keterangannya pada gambar 3.22.



Gambar 3.22. Maket Alat

3.4. Teknik dan Prosedur Pengumpulan Data

Teknik dan prosedur pengumpulan data yang akan dilakukan dalam kegiatan penelitian ini adalah studi pustaka atau studi literatur dan pembuatan serta pengukuran alat. Di mana studi pustaka ini dimaksudkan untuk mendapatkan teori-teori, konsep-konsep, sebagai bahan pertimbangan penguat atau penolakan terhadap temuan hasil penelitian dan untuk mengambil beberapa kesimpulan, literatur buku-buku yang dikaji dalam studi kepustakaan yang berkaitan langsung

dengan permasalahan penelitian. Selain itu juga untuk memperoleh data yang bersifat teoritis dan konsep sebagai pendukung penelitian ini dengan cara membaca buku-buku dokumen yang relevan dengan penelitian sehingga didapatkan informasi agar dalam penulisan tidak menyimpang dari tema. Literature-literatur yang didapatkan dari perpustakaan dan internet, selanjutnya dijadikan pedoman tambahan dalam penulisan.

Untuk pembuatan dan pengukuran alat, prosedur penelitian yang ditempuh dalam merancang dan membuat pengendali pintu dengan sistem keamanan berbasis web dan android terdiri dari beberapa tahapan yaitu :

1. Perakitan dimulai dari pemilihan modul, komponen, pemasangan, penyolderan, dan *wiring* kabel.
2. Mengecek dengan manual setiap komponen yang diperlukan apakah dapat digunakan atau tidak.
3. Komponennya yaitu *push button*, *buzzer*, *relay*, *limit switch*, *regulator*, *solenoid*, *modem*, *router*.
4. Meletakkan rangkaian kunci pintu yaitu *driver (relay)* dan *solenoid* untuk membuka dan menutup pintu rumah.
5. Meletakkan *limit switch* sebagai indikator ada atau tidaknya maling yang masuk ke dalam rumah jika *limit switch* tertekan.
6. Meletakkan *push button* untuk membuka pintu dari dalam rumah sebagai tombol pintu keluar dan dari pintu bagian luar sebagai tombol darurat.
7. Meletakkan *buzzer* sebagai indikator saat pintu terbuka dan tertutup serta saat ada maling atau bukan pengguna rumah masuk ke dalam rumah.
8. Meletakkan *door closer* pada pintu rumah sebagai alat penutup pintu otomatis.

9. Pemrograman arduino mega 2560 menggunakan software arduino IDE. Setelah program selesai dibuat, program siap diupload ke arduino mega 2560.
10. Membuat *database* dengan menggunakan *software SQLite*. Setelah *database* selesai dibuat, *database* siap untuk bekerjasama mengirimkan data dengan arduino.
11. Membuat *interface* dengan menggunakan *software Go-language (Golang)*. Setelah *interface* selesai dibuat, *interface* siap untuk bekerjasama mengirimkan data dengan arduino.
12. Setelah arduino, *database* dan *interface* dibuat, terjadi sinkronisasi komunikasi ketiganya. Jika sudah terhubung, lakukan pengujian.
13. Selanjutnya uji coba dengan pengambilan video.
14. Melakukan pengukuran tegangan, arus serta kriteria pengujian lainnya untuk mendapatkan data penelitian.
15. Menyimpulkan hasil penelitian dari hasil pengujian yang telah dilakukan.

3.5. Teknik Analisis Data

Teknik analisis data merupakan penjelasan tentang teknik yang akan digunakan dalam memecahkan masalah dalam bagian pembahasan serta untuk mendapatkan data yang diperlukan pada sistem, kriteria pengujian diperlukan untuk menyatakan berhasil atau tidaknya sistem ini.

Berikut merupakan kriteria pengujian yang dilakukan pada rancangan alat :

3.5.1 Pengujian Pada *Driver Solenoid (Relay)*

Hasil pengujian *relay solenoid* didapat setelah dilakukan pengukuran tegangan yang dihasilkan saat *solenoid* aktif dan *solenoid* tidak aktif. Hasil pengujian *relay solenoid* terlihat pada tabel 3.2.

Tabel 3.2. Hasil Pengujian *Relay Solenoid*

Kondisi	Kriteria	Hasil (V)		Gambar	
		Pintu 1	Pintu 2	Pintu 1	Pintu 2
<i>Solenoid Aktif</i>	<i>High</i>				
<i>Solenoid Tidak Aktif</i>	<i>Low</i>				

3.5.2. Pengujian *Regulator*

Hasil pengujian *regulator* didapat setelah dilakukan pengukuran tegangan input dan output dari *regulator* ke arduino dan *driver*. Hasil pengujian *regulator* terlihat pada tabel 3.3.

Tabel 3.3. Hasil Pengujian *Regulator*

Kondisi	Hasil (V)	Gambar
Vin Regulator ke Arduino		
Vout Regulator ke Arduino		

Vin Regulator ke Driver		
Vout Regulator ke Driver		

3.5.3. Pengujian *Limit Switch*

Pengujian pada *limit switch* dilakukan dengan mengukur tegangan yang dihasilkan saat *limit switch* aktif dan tidak aktif. Pengujian *limit switch* ini tertuang dalam tabel 3.4. di bawah ini.

Tabel 3.4. Hasil Pengujian *Limit Switch*

Kondisi	Kriteria	Hasil Vout (V)		Gambar	
		Pintu 1	Pintu 2	Pintu 1	Pintu 2
Sensor Aktif (ON)	<i>High</i>				
Sensor Tidak Aktif (OFF)	<i>Low</i>				

3.5.4. Pengujian *Push Button*

Pengujian pada *push button* dilakukan dengan mengukur tegangan yang dihasilkan saat *push button* aktif dan tidak aktif pada tombol keluar dan tombol darurat. Pengujian *push button* ini tertuang dalam tabel 3.5. di bawah ini.

Tabel 3.5. Hasil Pengujian *Push Button*

Kondisi	Kriteria	Hasil (V)	Gambar
---------	----------	-----------	--------

		Pintu 1	Pintu 2	Pintu 1	Pintu 2
Tombol Keluar					
ON	<i>High</i>				
OFF	<i>Low</i>				
Tombol Darurat					
ON	<i>Low</i>				
OFF	<i>High</i>				

3.5.5. Pengujian Kecepatan Buka Kunci Pintu Melalui Web dan Aplikasi Android

Hasil Pengujian kecepatan membuka kunci pintu ini didapat setelah menghitung rata – rata waktu yang dibutuhkan dari saat menekan tombol buka pintu pada web/ aplikasi android hingga kunci pintu terbuka. Hasil pengujian kecepatan membuka kunci terlihat pada tabel 3.6.

Tabel 3.6. Hasil Pengujian Kecepatan Buka Kunci Pintu Melalui Web dan Aplikasi Android

Pengujian Ke-	Waktu Yang Dibutuhkan Untuk Membuka Kunci (Detik)		Gambar	
	Pintu 1	Pintu 2	Pintu 1	Pintu 2
1				
2				
3				

4				
5				
Rata-rata				
Selisih				

3.5.6. Pengujian Kecepatan Buka Kunci Pintu Melalui Tombol Keluar

Hasil Pengujian kecepatan membuka kunci pintu ini didapat setelah menghitung rata – rata waktu yang dibutuhkan dari saat menekan tombol buka pintu melalui tombol keluar hingga kunci pintu terbuka. Hasil pengujian kecepatan membuka kunci terlihat pada tabel 3.7.

Tabel 3.7. Hasil Pengujian Kecepatan Buka Kunci Pintu Melalui Tombol Keluar

Pengujian Ke-	Waktu Yang Dibutuhkan Untuk Membuka Kunci (Detik)		Gambar	
	Pintu 1	Pintu 2	Pintu 1	Pintu 2
1				
2				
3				
4				
5				
Rata-rata				
Selisih				

3.5.7. Pengujian Kecepatan Buka Kunci Pintu Melalui Tombol Darurat

Hasil Pengujian kecepatan membuka kunci pintu ini didapat setelah menghitung rata – rata waktu yang dibutuhkan dari saat menekan tombol buka pintu melalui tombol darurat hingga kunci pintu terbuka. Hasil pengujian kecepatan membuka kunci terlihat pada tabel 3.8.

Tabel 3.8. Hasil Pengujian Kecepatan Buka Kunci Pintu Melalui Tombol Darurat

Pengujian Ke-	Waktu Yang Dibutuhkan Untuk Membuka Kunci (Detik)		Gambar	
	Pintu 1	Pintu 2	Pintu 1	Pintu 2
1				
2				
3				
4				
5				
Rata-rata				
Selisih				

3.5.8. Komunikasi Web dengan Arduino Mega 2560

Web server adalah sebuah hardware atau software yang menyediakan layanan akses ke orang atau si pemakai melalui komunikasi HTTP seperti halaman web yang disimpan di web server. Jadi jika seseorang sebagai pengguna akses suatu web, orang tersebut masuk ke web server.

Tentunya fungsi web server sangat beragam tergantung dari siapa yang memakai dan untuk apa memakainya. Misalkan punya domain karyateknologi.com, saya simpan file saya di web server. jika saya buka karyateknologi.com, untuk mengambil filenya tidak di komputer saya langsung, tapi ke web server yang saya titipkan.

Sebelum masuk ke hal diatas, tentunya kita harus tahu dan bisa membuat web server yang sederhana. Membuat web server disini, saya akan menggunakan sebuah board yaitu Arduino Mega 2560 yang terhubung ke Ethernet shield yaitu board yang dapat terhubung dengan jaringan internet serta dapat juga digabungkan dengan beberapa sensor sebagai alat monitoring maupun kontrol. Karena ethernet *controller* ini berkomunikasi melalui jalur SPI (Serial Peripheral Interface), terlebih dahulu kita masukkan dulu file header dari SPI dan ethernet nya. Setelah itu tentunya kita harus menentukan sendiri Mac addr yang akan dipakai oleh Ethernet ini. selain itu, karena dia sebagai server, fungsi DHCP tidak akan diaktifkan disini melainkan kita tentukan IP nya manual. Tentunya class IP juga harus sama dengan jaringan yang ada. Agar ethernet kontroler mau bekerja dengan baik, terlebih dahulu harus memulai koneksi ethernet dengan mengetik coding pada void setup() pada Arduino IDE.

Saat koneksi sudah berlangsung tentunya Ethernet tidak langsung menuliskan file webnya ke jaringan tapi harus dibuatkan logika jika ada client (kita sebagai pengguna) yang request ke IP Arduino maka Arduino harus menghubungkan terus ke jaringan dan mengecek apa perintah dari sana, setelah itu baru Ethernet mengirimkan webnya.

Saat ini bahasa HTML langsung kita tulis pada listing codingnya langsung, Tetunya HTML tersebut tidak dapat kita tulis secara sembarang, tapi harus kita awali dengan client, karena Ethernet terhubung dengan client. Jadi masih belum file HTML, melainkan kita tulis format html nya di coding Arduinonya. Jadi inti dari komunikasi ini adalah Arduino menggantikan perintah manual dari web yang terhubung ke jaringan internet melalui Ethernet shield.

BAB IV

HASIL PENELITIAN

4.1 Deskripsi Hasil Penelitian

4.1.1 Maket Alat

Berdasarkan blok diagram serta *flowchart* yang telah dirancang pada penjelasan sebelumnya, maka pengendali pintu dengan sistem keamanan berbasis web dan android diimplementasikan oleh peneliti pada Gambar 4.1.



Gambar 4.1. Penampakan Keseluruhan Maket Alat

Pada maket alat terdiri dari dua buah pintu yaitu pintu 1 (pintu depan rumah) dan pintu 2 (pintu belakang rumah) dengan *box* panel kontrol berada di sisi atas maket. Untuk halaman depan pintu dipasang daun pintu yang di dalamnya terdapat kunci manual biasa, tombol darurat berwarna merah yang dipasang pada sisi kiri atas pintu yang dapat dilihat seperti gambar 4.2. berikut.



Gambar 4.2. Penampakan Pintu dari Sisi Depan

Untuk bagian belakang pintu (dalam rumah) terdapat tombol keluar yaitu *push button* yang berwarna hijau dipasang diatas daun pintu serta *door closer* dengan sistem katrol yang dibuat manual menggunakan tali yang diikat pada balok kayu dan digantungi beban pada *box* hitam yang ada pada sisi pintu sehingga ketika pintu dibuka akan langsung terdorong oleh beban dan tertutup otomatis serta *solenoid* yang dipasang di dalam pintu di atas kunci pintu manual yang dapat dilihat pada gambar 4.3. berikut.



Gambar 4.3. Penampakan Pintu dari Sisi Belakang Pintu (Dalam Rumah)

sedangkan untuk *limit switch* ditaruh masing-masing di atas pintu, sehingga ketika pintu dibuka dan ditutup, *limit switch* akan tertekan dan memberikan sinyal kepada arduino untuk selanjutnya diproses sesuai dengan program. Berikut gambar *limit switch* yang terdapat pada gambar 4.4.



Gambar 4.4. *Limit Switch*

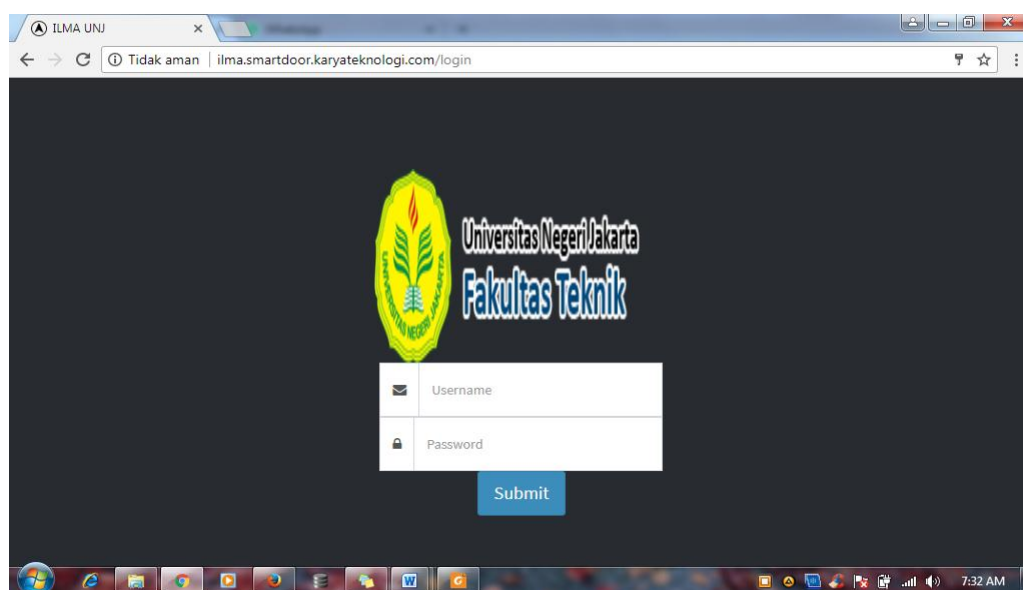
Selanjutnya *box* kontrol panel yang diletakkan di bagian atas pintu, terdiri dari *router, modem, regulator, arduino mega2560, Ethernet shield, buzzer, modul relay (driver solenoid)* dan *power supply*. Dapat dilihat pada gambar 4.5. berikut.



Gambar 4.5. Box Panel Kontrol

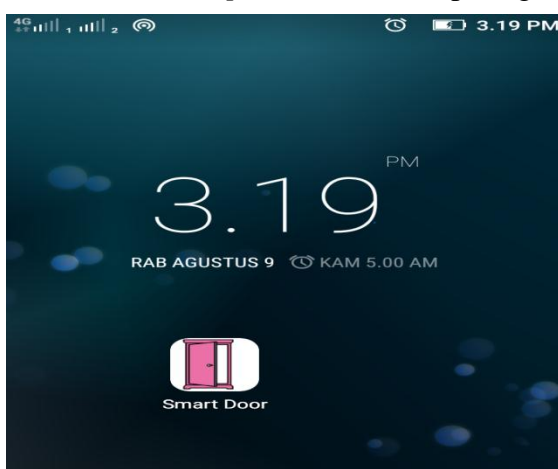
4.1.2 Tampilan Web dan Android

Berdasarkan maket alat yang sudah dibuat, alat dapat berjalan secara *online* atau dapat dikendalikan lewat jarak jauh jika laptop atau *smartphone android* sudah terkoneksi dengan internet dan dapat dibuka melalui *website* yang sudah dibuat dengan alamat <http://ilma.smartdoor.karyateknologi.com> sesuai gambar 4.6. di bawah ini.



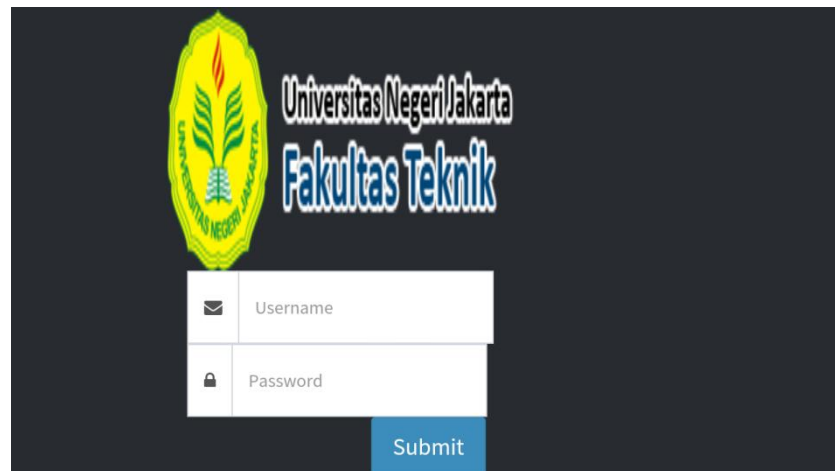
Gambar 4.6. Halaman Login dari Website

Halaman *login* ini dapat dibuka dengan cara lain yang lebih mudah yaitu melalui aplikasi yang sudah terinstall di *smartphone android* seperti gambar 4.7. berikut.



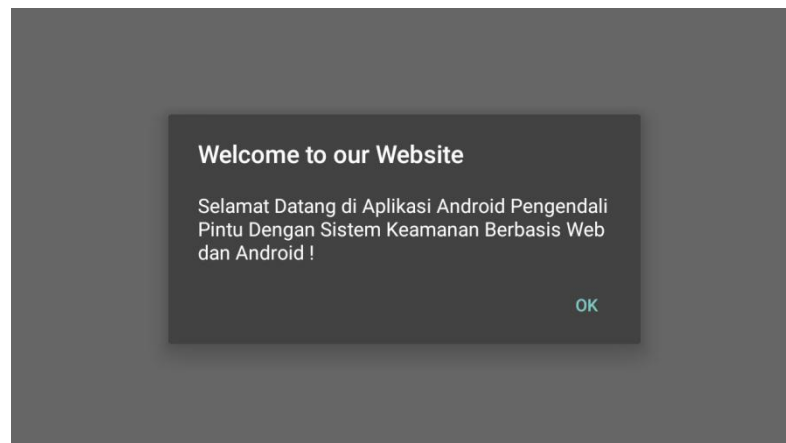
Gambar 4.7. Aplikasi Pengendali Pintu (*Smart Door*)

Untuk tampilan halaman *login*-nya sama, yaitu dapat dilihat pada gambar 4.8. berikut.



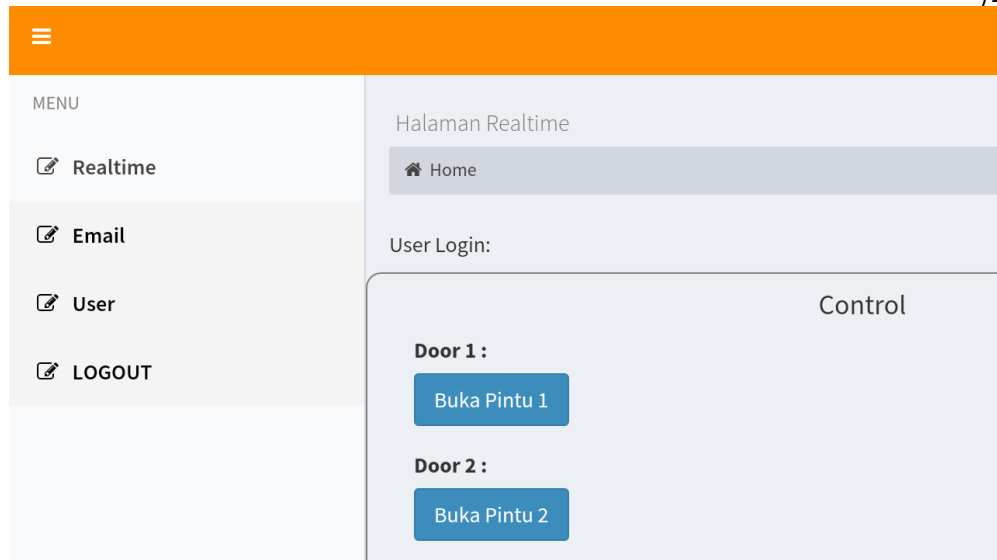
Gambar 4.8. Halaman *Login* Pada Aplikasi Android

Untuk tampilan menu-menu baik pada *website* maupun aplikasi android sama, hanya saja pada web tidak ada halaman ucapan selamat datang dan *application exit*.



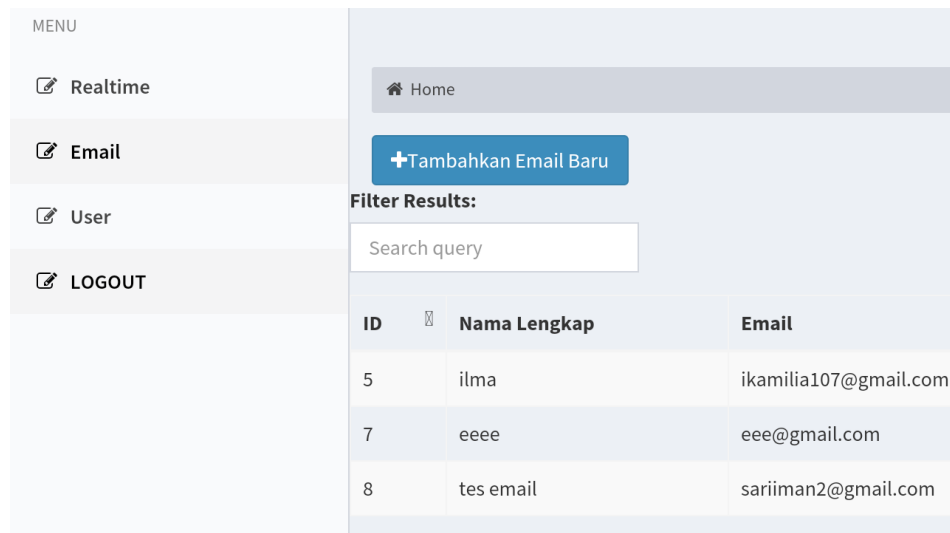
Gambar 4.9. Halaman Ucapan Selamat Datang Pada Aplikasi Android

Selanjutnya masuk ke menu, ada menu *realtime* yaitu halaman pokok untuk membuka pintu 1 dan pintu 2 seperti pada gambar 4.10. berikut.



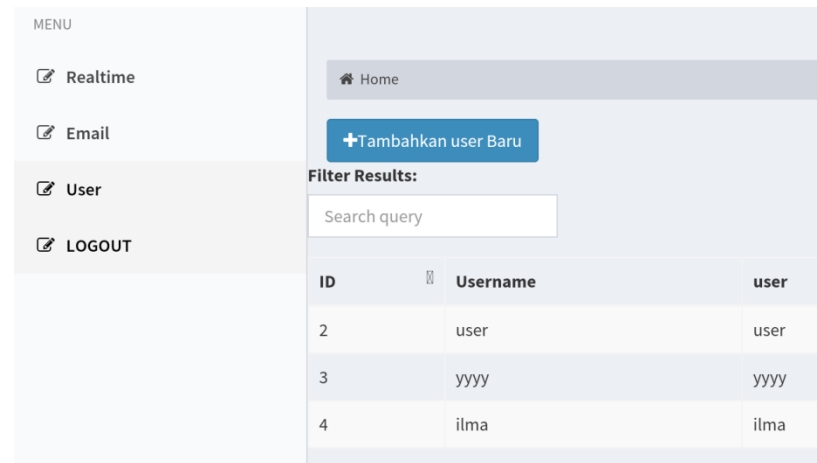
Gambar 4.10. Menu Realtime

Selanjutnya ada menu Email yaitu menu untuk menambahkan email bagi *user*/ pengguna.



Gambar 4.11. Menu Email

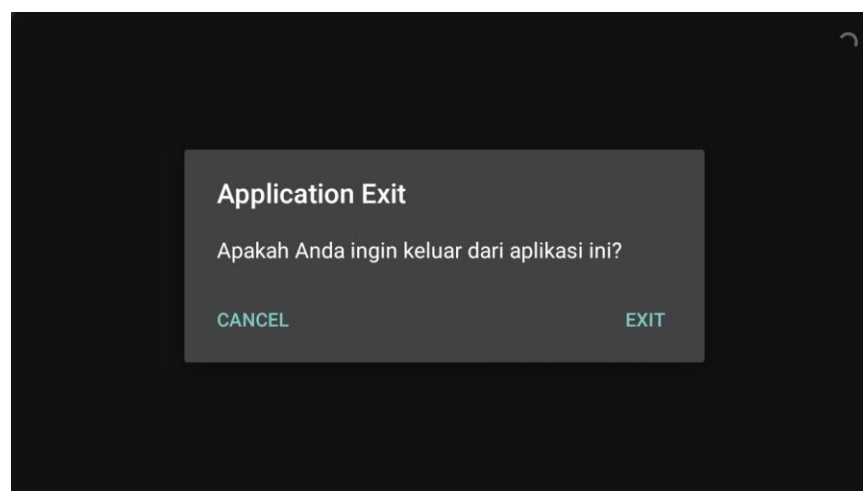
Di bawah menu Email ada menu *User* yaitu kita dapat melihat ataupun menambahkan *user/* pengguna. Dapat dilihat pada gambar 4.12 berikut.



Gambar 4.12. Menu *User*

Pada menu terakhir ada menu *logout* yaitu menu untuk keluar dari sistem atau jika ingin berganti pengguna, maka akan muncul halaman *login* seperti semula.

Ketika pengguna ingin menyudahi atau menutup aplikasi pada android, caranya dengan menekan tombol *back/* kembali sebanyak 2x yang ada pada *smartphone*, maka akan muncul notifikasi *application exit* yang bertujuan untuk memastikan lagi apakah pengguna benar-benar akan *exit* atau meng-*cancel*-nya.



Gambar 4.13. *Application Exit*

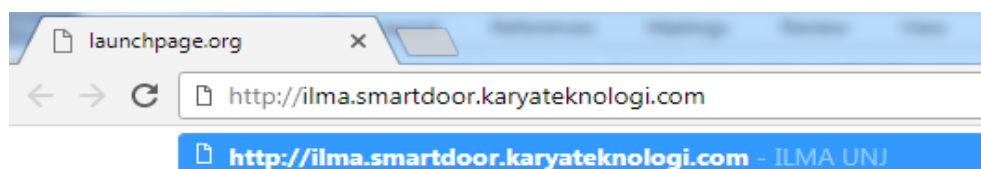
4.1.3 Langkah Kerja

Langkah kerja dari alat pengendali pintu dengan sistem keamanan berbasis web dan android adalah sebagai berikut :

1. Buka kunci pintu melalui web atau aplikasi android yang sudah terinstall pada smartphone pengguna.

a. Melalui Web

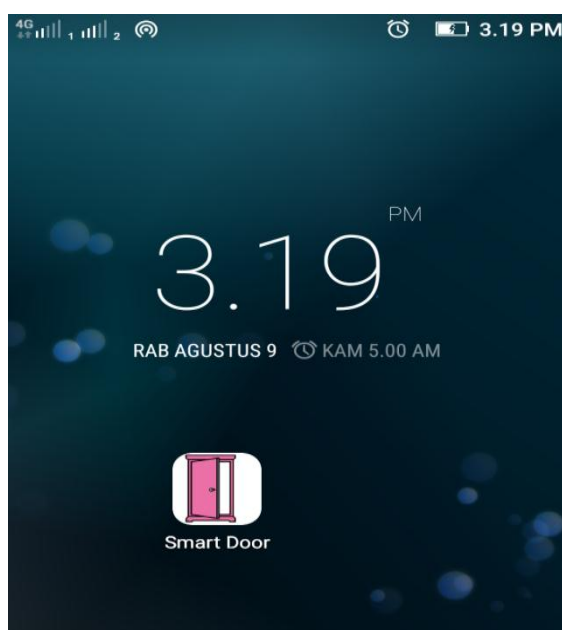
Buka alamat website <http://ilma.smartdoor.karyateknologi.com> sesuai gambar 4.14. di bawah ini.



Gambar 4.14. Alamat URL Website

b. Melalui aplikasi android

Buka aplikasi smart door yang sudah terinstall di *smartphone android* seperti gambar 4.15. berikut, klik sekali.



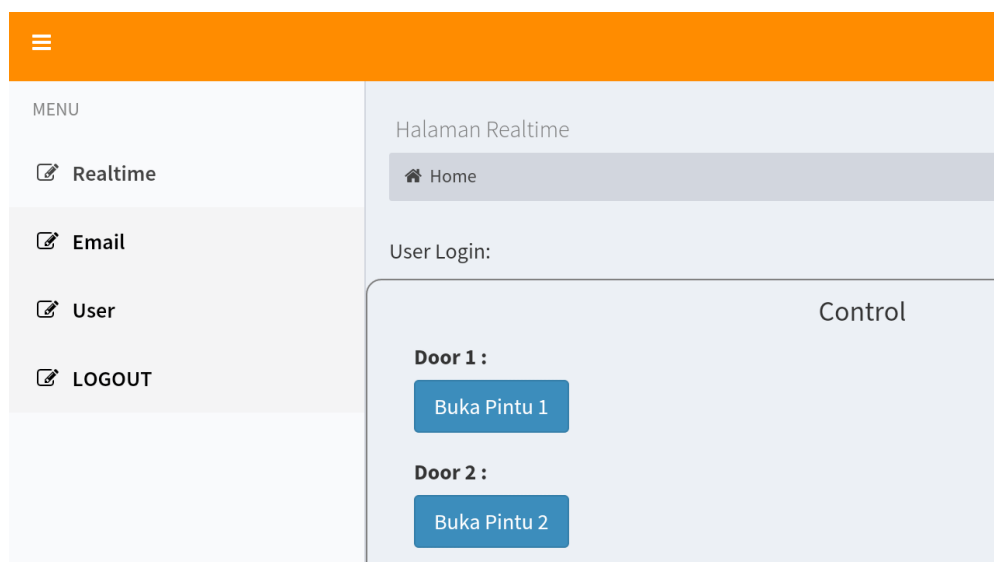
Gambar 4.15. Aplikasi Pengendali Pintu (*Smart Door*)

2. Menu selanjutnya yang akan tampil pada web atau aplikasi android yaitu halaman login. Pemilik rumah / *user* dimohon untuk memasukkan *username* dan *password* apabila ingin mengakses sistem buka kunci pintu rumah. Tampilan halaman login dapat dilihat pada gambar 4.16. berikut.



Gambar 4.16. Tampilan Halaman *Login*

3. Selanjutnya masuk ke menu *realtime* yaitu halaman pokok untuk membuka pintu 1 dan pintu 2 seperti pada gambar 4.17. berikut.



Gambar 4.17. Menu *Realtime*

Sebelumnya, alat sudah harus dihubungkan ke power supply untuk mendapat arus dan tegangan.

Klik menu buka pintu 1 untuk membuka pintu 1 (depan) atau klik menu buka pintu 2 untuk membuka pintu 2 (belakang). Selanjutnya, sistem akan memberikan sinyal ke arduino untuk membuka pintu, arduino akan memberi inputan sinyal ke driver untuk segera mengaktifkan solenoid yang semula dalam keadaan terkunci dan menjadi terbuka. Buzzer akan berbunyi 2x sebagai penanda bahwa sistem sudah berjalan sesuai perintah. Jika solenoid sudah aktif, maka pintu dapat terbuka dan buzzer berbunyi sekali lagi sebagai penanda bahwa pintu sudah siap untuk dibuka.



Gambar 4.18. Pintu Terbuka

Setelah pintu terbuka, lalu pintu akan siap ditutup kembali dengan penanda buzzer yang berbunyi sekali. Pintu akan terdorong otomatis oleh door closer manual, kemudian kunci akan menutup lagi secara otomatis.

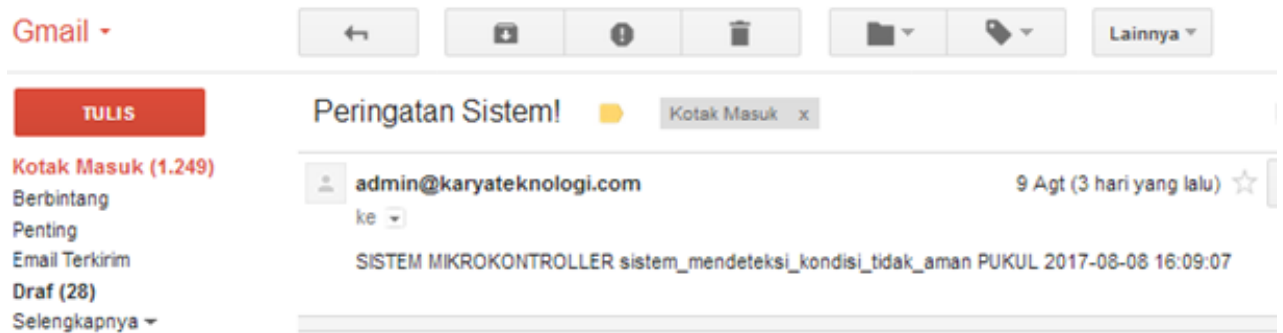
4. Ketika pemilik rumah/ pengguna sistem ingin keluar atau membuka pintu dari dalam, pemilik rumah bisa menekan tombol keluar warna hijau untuk membuka pintu 1 ataupun pintu 2 seperti terlihat pada gambar 4.19. di bawah ini.



Gambar 4.19. Buka Pintu dari Tombol Keluar

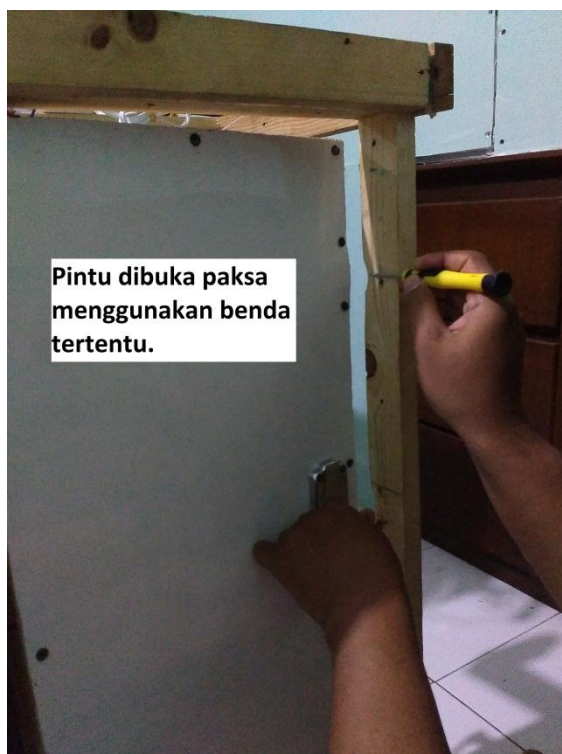
Buzzer sebagai indikator pintu dibuka akan berbunyi sama dengan ketika pintu dibuka dari luar oleh pemilik rumah.

5. Selanjutnya jika ingin keluar rumah dan mengakhiri sistem, pemilik bisa *logout* dari menu pada aplikasi di web atau android dengan mengklik menu *logout* sekali seperti yang ada pada gambar 4.17.
6. Jika ada seseorang memaksa masuk tanpa melalui sistem yang ada di web atau pada aplikasi android, maka sistem akan mendeteksi kondisi tidak aman atau ada maling dan secara otomatis akan mengirimkan notifikasi ke alamat *email user* yang sudah terdaftar pada sistem seperti gambar berikut.

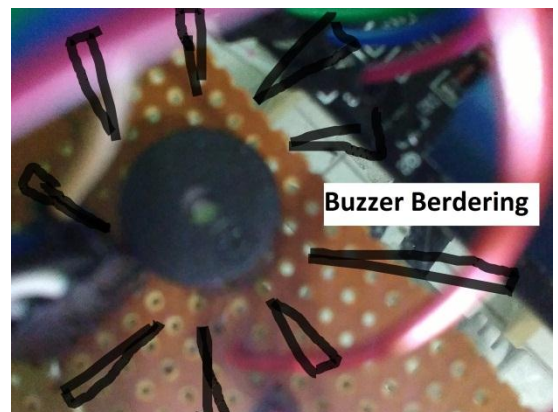


Gambar 4.20. Email Notifikasi Kondisi Tidak Aman

7. Selanjutnya keadaan pintu yang terbuka karena dibuka paksa, maka ketika pintu ditutup akan terkunci otomatis sehingga maling dapat terperangkap di dalam rumah dan *buzzer* berdering panjang menandakan rumah dalam kondisi tidak aman atau ada maling.



Gambar 4.21. Pintu Dibuka Paksa



Gambar 4.22. Buzzer Berdering

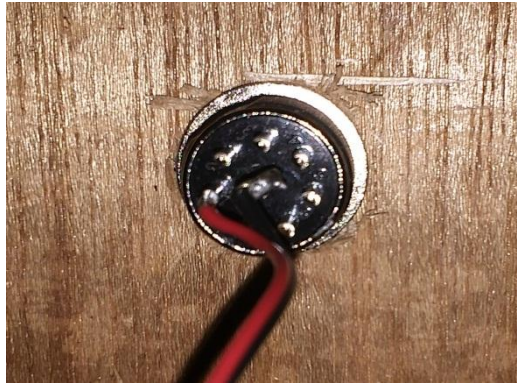
8. Jika ingin mengeluarkan maling atau seseorang yang terperangkap di dalam rumah, maka seseorang dari luar yang mendengar bunyi *buzzer* bisa menekan

tombol darurat warna merah yang ada di ujung kiri atas depan pintu untuk mematikan bunyi *buzzer* dan membuka kunci secara otomatis.



Gambar 4.23. Tombol Darurat

9. Setelah pintu dapat terbuka, maling yang berada di dalam rumah ditangkap dan pintu mengunci secara otomatis, untuk membukanya kembali dengan menekan tombol keluar yang berwarna hijau, begitu seterusnya dan kembali ke keadaan normal.
10. Jika listrik di rumah padam, sedangkan sistem masih mengunci pintu, maka pemilik rumah dapat membukanya dengan kunci manual elektronik yang dipasang di tempat rahasia dan hanya pemiliknya pula yang dapat mengetahui pin rahasia untuk membuka kuncinya.



Gambar 4.24. Kunci Manual Elektronik di Rumah

11. Untuk keamanan ganda, pemilik dapat tetap menggunakan kunci manual biasa yang dapat di buka melalui daun pintu.



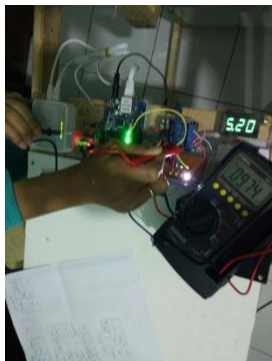
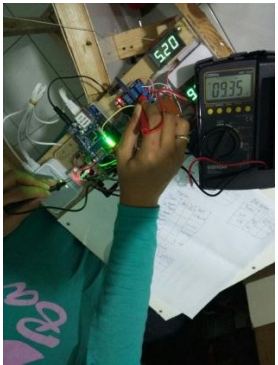
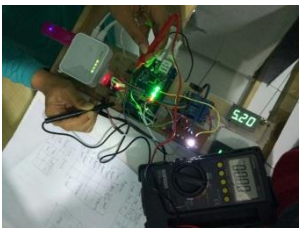
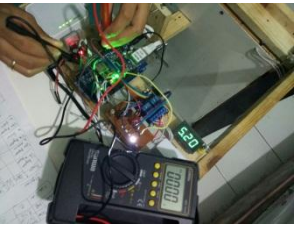
Gambar 4.25. Kunci Manual Biasa

4.2 Analisis Data Penelitian

4.2.1 Pengujian Pada *Driver Solenoid (Relay)*

Hasil pengujian *relay solenoid* didapat setelah dilakukan pengukuran tegangan yang dihasilkan saat *solenoid* aktif dan *solenoid* tidak aktif. Hasil pengujian *relay solenoid* terlihat pada tabel 4.1.

Tabel 4.1. Hasil Pengujian *Relay Solenoid*

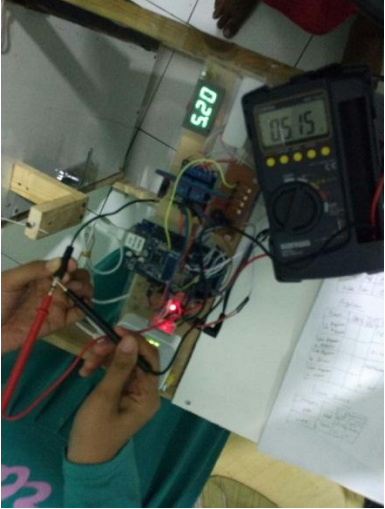


Kondisi	Kriteria	Hasil (V)		Gambar	
		Pintu 1	Pintu 2	Pintu 1	Pintu 2
<i>Solenoid Aktif</i>	<i>High</i>	9,74	9,35		
<i>Solenoid Tidak Aktif</i>	<i>Low</i>	0	0		

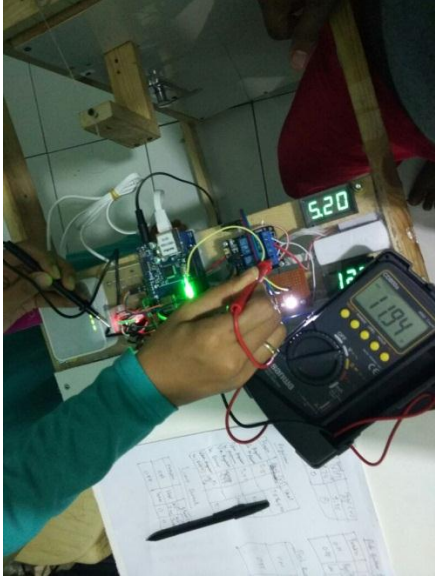
4.2.2. Pengujian Regulator

Hasil pengujian regulator didapat setelah dilakukan pengukuran tegangan *input* dan *output* dari *regulator* ke *arduino* dan *driver*. Hasil pengujian *regulator* terlihat pada tabel 4.2.

Tabel 4.2. Hasil Pengujian *Regulator*

Kondisi	Hasil (V)	Gambar
Vin Regulator ke Arduino	5,15	

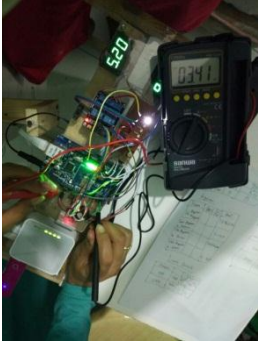
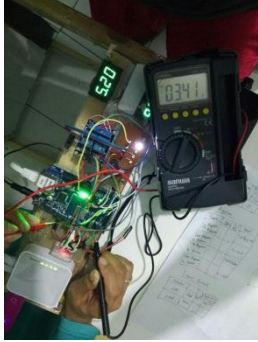
		
Vout Regulator ke Arduino	3,52	
Vin Regulator ke Driver	3,52	

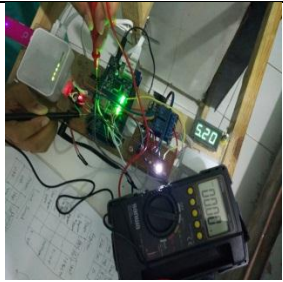
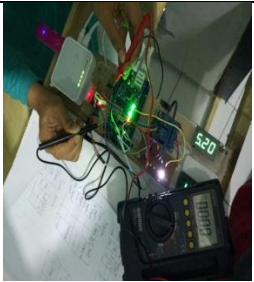
Vout Regulator ke Driver	11,94	
--------------------------	-------	--

4.2.3. Pengujian *Limit Switch*

Pengujian pada *limit switch* dilakukan dengan mengukur tegangan yang dihasilkan saat *limit switch* aktif dan tidak aktif. Pengujian *limit switch* ini tertuang dalam tabel 4.3. di bawah ini.

Tabel 4.3. Hasil Pengujian *Limit Switch*



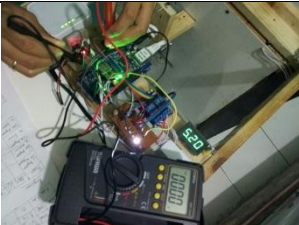
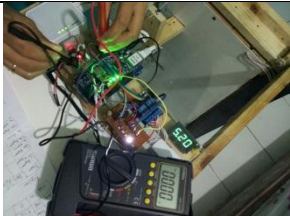
Kondisi	Kriteria	Hasil Vout (V)		Gambar	
		Pintu 1	Pintu 2	Pintu 1	Pintu 2
Sensor Aktif (ON)	<i>High</i>	3,41	3,41		

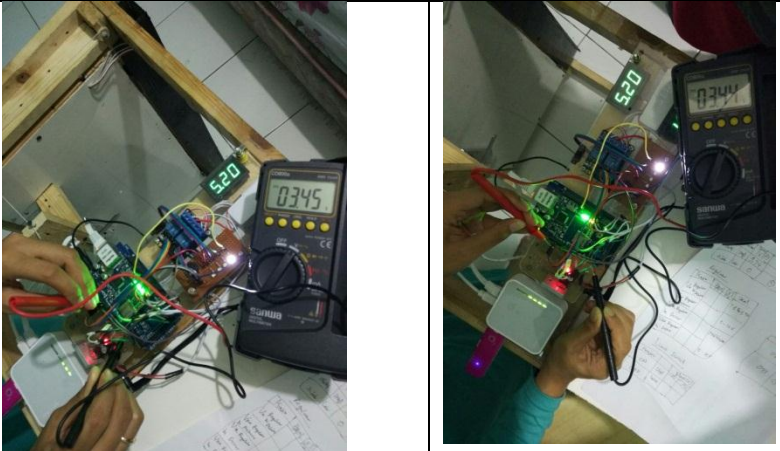
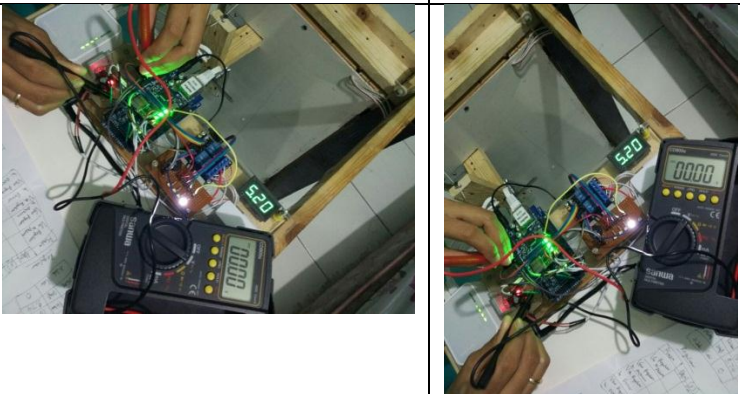
Sensor Tidak Aktif (OFF)	<i>Low</i>	0	0		
-----------------------------------	------------	---	---	--	---

4.2.4 Pengujian *Push Button*

Pengujian pada *push button* dilakukan dengan mengukur tegangan yang dihasilkan saat *push button* aktif dan tidak aktif pada tombol keluar dan tombol darurat. Pengujian *push button* ini tertuang dalam tabel 4.4. di bawah ini.

Tabel 4.4. Hasil Pengujian *Push Button*

Kondisi	Kriteria	Hasil (V)		Gambar	
		Pintu 1	Pintu 2	Pintu 1	Pintu 2
Tombol Keluar					
ON	<i>High</i>	3,41	3,41		
OFF	<i>Low</i>	0	0		

Tombol Darurat				
ON	High	3,45	3,44	
OFF	Low	0	0	

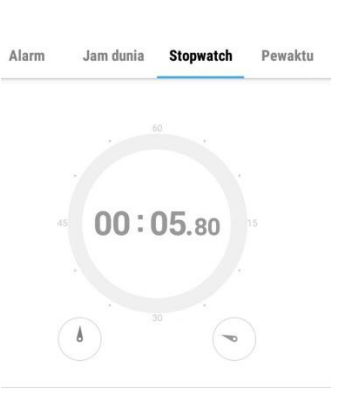
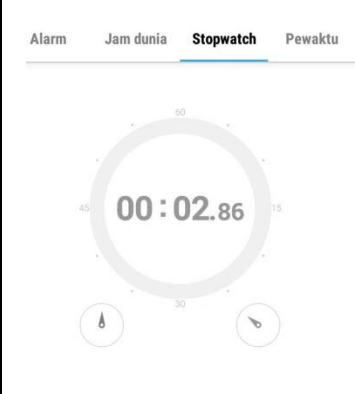
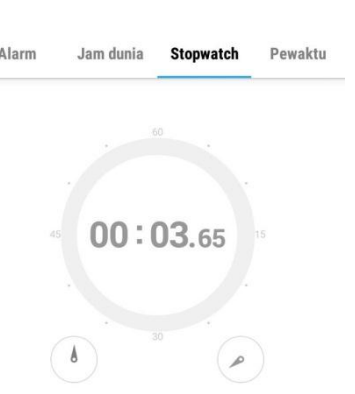
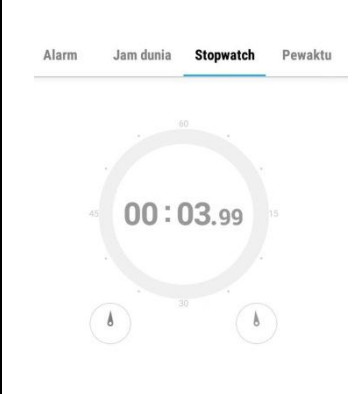
4.2.5 Pengujian Kecepatan Buka Kunci Pintu Melalui Web dan Aplikasi Android

Hasil Pengujian kecepatan membuka kunci pintu ini didapat setelah menghitung rata – rata waktu yang dibutuhkan dari saat menekan tombol buka pintu pada web/ aplikasi android hingga kunci pintu terbuka. Hasil pengujian kecepatan membuka kunci terlihat pada tabel 4.5.

Tabel 4.5. Hasil Pengujian Kecepatan Buka Kunci Pintu Melalui Web dan Aplikasi Android

Pengujian Ke-	Waktu Yang Dibutuhkan	Gambar
---------------	-----------------------	--------

	Untuk Membuka Kunci (Detik)			
	Pintu 1	Pintu 2	Pintu 1	Pintu 2
1	2,98	4,07	<p>Alarm Jam dunia Stopwatch Pewaktu</p> <p>00 : 02.98</p>	<p>Alarm Jam dunia Stopwatch Pewaktu</p> <p>00 : 04.07</p>
2	2,07	2,83	<p>Alarm Jam dunia Stopwatch Pewaktu</p> <p>00 : 02.07</p>	<p>Alarm Jam dunia Stopwatch Pewaktu</p> <p>00 : 02.83</p>
3	5,13	2,17	<p>Alarm Jam dunia Stopwatch Pewaktu</p> <p>00 : 05.13</p>	<p>Alarm Jam dunia Stopwatch Pewaktu</p> <p>00 : 02.17</p>

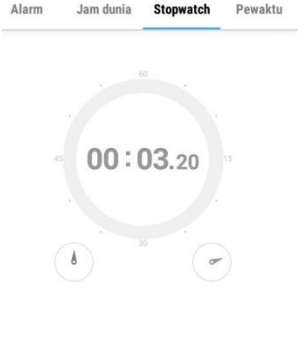
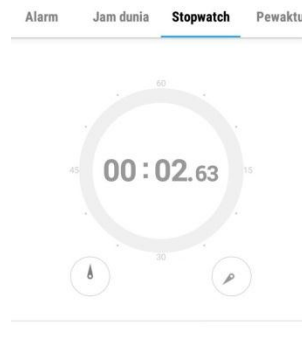
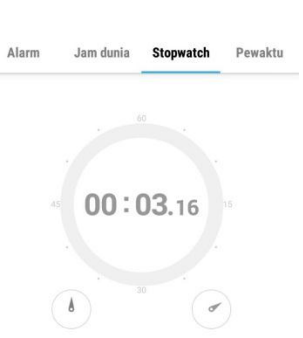
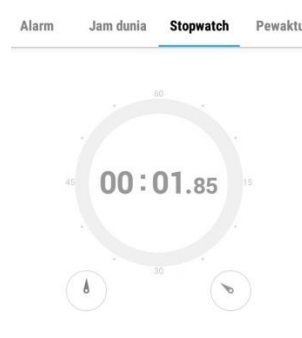
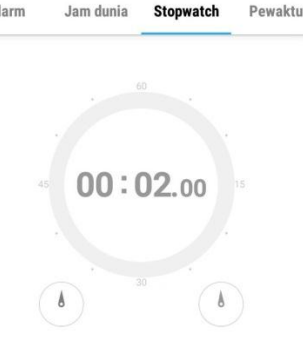
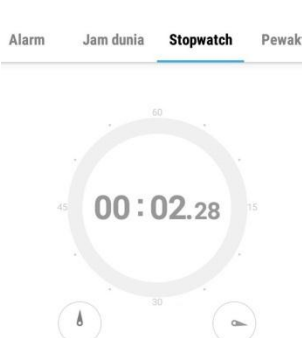
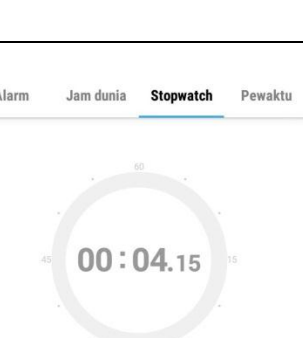
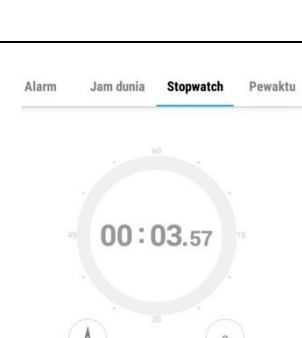
4	5,80	2,86		
5	3,65	3,99		
Rata-rata	3,926	3,184		
Selisih	0,742			

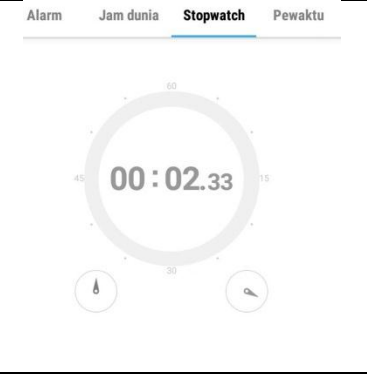
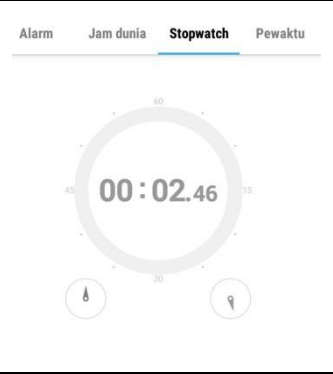
4.2.6 Pengujian Kecepatan Buka Kunci Pintu Melalui Tombol Keluar

Hasil Pengujian kecepatan membuka kunci pintu ini didapat setelah menghitung rata – rata waktu yang dibutuhkan dari saat menekan tombol buka pintu melalui tombol keluar hingga kunci pintu terbuka. Hasil pengujian kecepatan membuka kunci terlihat pada tabel 4.6.

Tabel 4.6. Hasil Pengujian Kecepatan Buka Kunci Pintu Melalui Tombol Keluar

Pengujian Ke-	Waktu Yang Dibutuhkan Untuk Membuka Kunci	Gambar

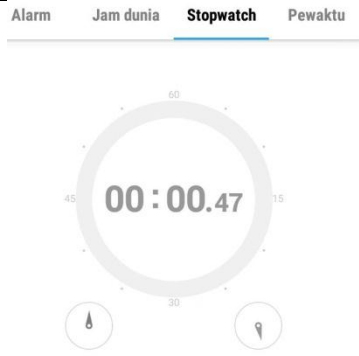
	(Detik)			
	Pintu 1	Pintu 2	Pintu 1	Pintu 2
1	3,20	2,63		
2	3,16	1,85		
3	2,00	2,28		
4	4,15	3,57		

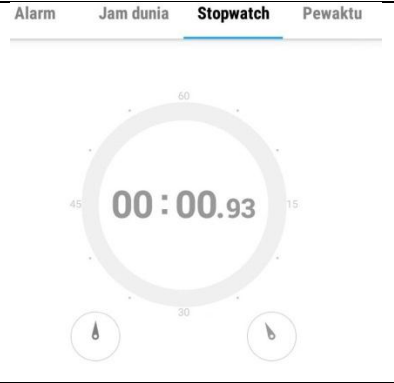
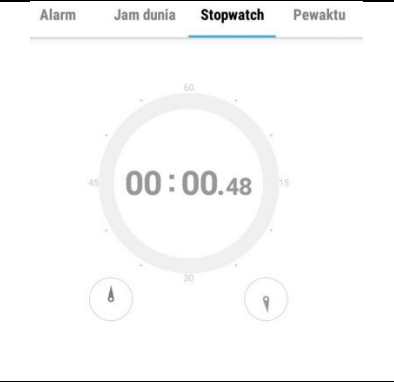
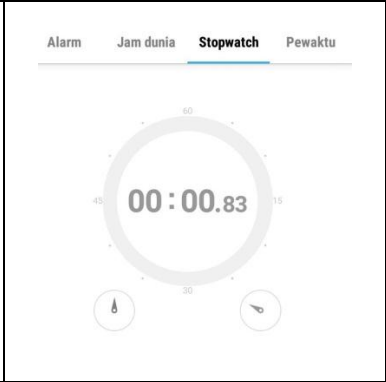
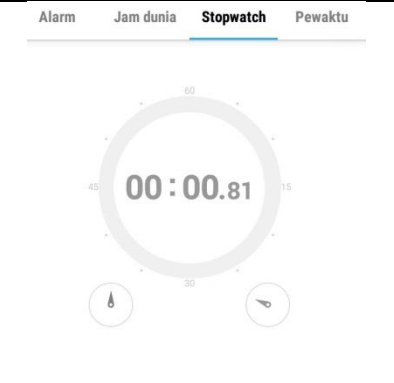
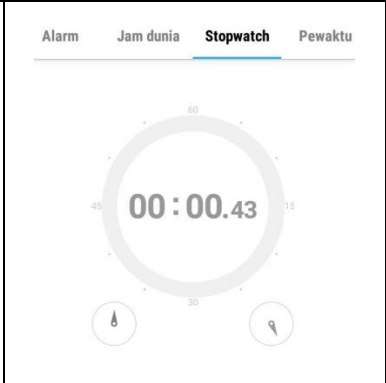
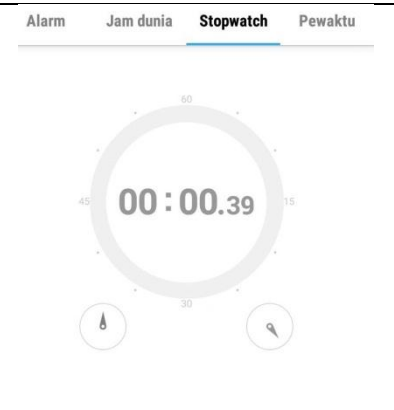
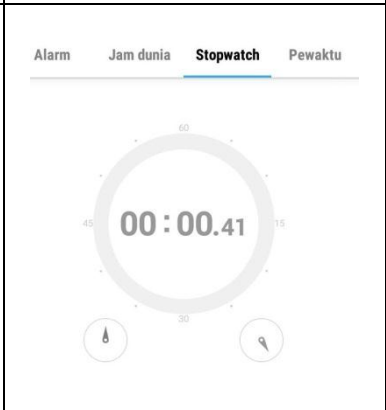
5	2,33	2,46		
Rata-rata	2,968	2,558		
Selisih	0,41			

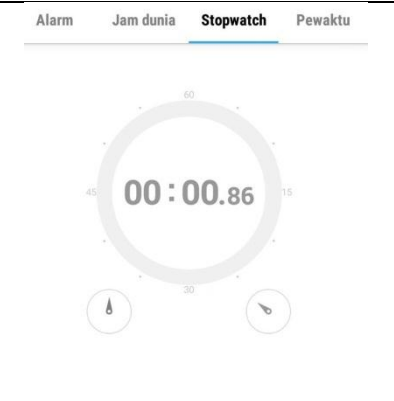
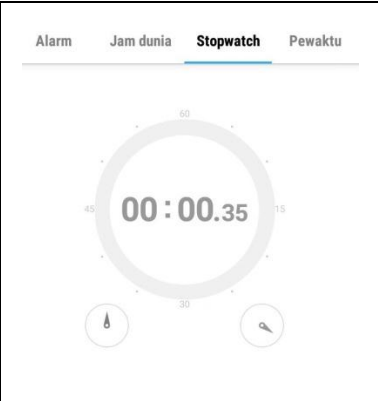
4.2.7 Pengujian Kecepatan Buka Kunci Pintu Melalui Tombol Darurat

Hasil Pengujian kecepatan membuka kunci pintu ini didapat setelah menghitung rata – rata waktu yang dibutuhkan dari saat menekan tombol buka pintu melalui tombol darurat hingga kunci pintu terbuka. Hasil pengujian kecepatan membuka kunci terlihat pada tabel 4.7.

Tabel 4.7. Hasil Pengujian Kecepatan Buka Kunci Pintu Melalui Tombol Darurat

Pengujian Ke-	Waktu Yang Dibutuhkan Untuk Membuka Kunci (Detik)		Gambar	
	Pintu 1	Pintu 2	Pintu 1	Pintu 2
1	0,93	0,47		

				
2	0,48	0,83		
3	0,81	0,43		
4	0,39	0,41		

5	0,86	0,35		
Rata-rata	0,694	0,498		
Selisih	0,196			

4.2.8. Program Komunikasi Web dengan Arduino Mega 2560

a. Pengecekan status internet, jika terhubung ke server maka Led menyala dan sebaliknya.

```
bool skipResponseHeaders()
{
  char endOfHeaders[] = "\r\n\r\n";
  client.setTimeout(HTTP_TIMEOUT);
  bool ok = client.find(endOfHeaders);

  if (!ok)
  {
    Serial.println("gagal terhubung ke server, cek jaringan!");
  }
  if (ok)
  {
    //terhubung ke server
    digitalWrite(ledInternet, HIGH);
  }
  else
  {
    // tidak terhubung ke server
    digitalWrite(ledInternet, LOW);
  }
  return ok;
}
```

Gambar 4.26. Pengecekan Status Internet

b. Perintah koneksi ke server

```
void loop()
{
  Serial.flush();
  if (connect(server)) // mencoba koneksi dengan server
  {
    //jika berhasil jalankan program di sini
    Serial.println("membaca perintah buka pintu dari server dan membaca kondisi tidak aman");

    if (sendRequest(server, urlGetLastValue) && skipResponseHeaders()) // mencoba mendapatkan nilai status terakhir pintu dari database
    {
      UserData userData;
      if (readReponseContent(&userData)) // jika ada respon maka jalankan perintah d bawahnya
      {
        printUserData(&userData);
        (&userData);
        olahResponseBukaPintu(&userData); // fungsi untuk membuka pintu, pintu 1 atau pintu 2 , fungsi ada di file oleah json
      }
    }
  }
  disconnect();// diskonek server setelah dapat nilainya
}
```

Gambar 4.27. Koneksi ke Server

c. Target Server

```
char *serverOffline = "192.168.137.1";
//char *serverOnline = "ilma.smartdoor.karyateknologi.com";
char *serverOnline = "104.168.87.182";
const unsigned long HTTP_TIMEOUT = 10000; // max response time from server
int portOffline = 10007;
int portOnline = 10007;
bool isOnlineWWW=true;
```

Gambar 4.28. Target Server

d. URL tanpa domain



The image shows a browser address bar with the URL `104.168.87.182:10007/v1/mikro/getlastvalue`. Below the address bar, a JSON response is displayed:

```
{
  "status_door1": "1",
  "status_door2": "0",
  "control_door1": "0",
  "control_door2": "0"
}
```

Gambar 4.29. URL Tanpa Domain

e. Arduino menggantikan perintah koneksi ke server dengan perintah ini

```

{
  "status_door1": "1",
  "status_door2": "0",
  "control_door1": "0",
  "control_door2": "0"
}

```

url dari web

jika nilainya 1 untuk perintah buka pintu d mikro

```

int duration = 4;
int aman = 1;
int standby = 0;
int amanPintu1 = standby;
int amanPintu2 = standby;
EthernetClient client;
int relayOn = 0;
int relayOff = 1;
const char *urlGetLastValue = "/v1/mikro/getlastvalue";
const char *urlUpdateDoorOpened = "/v1/mikro/updateclosealldoor?door="; // http urlGetLastValue
const char *urlUpdateLog = "/v1/mikro/updatelogalat?log="; // http log alat update
const unsigned long BAUD RATE = 9600; // serial connection speed

```

Gambar 4.30. Arduino Menggantikan Perintah Koneksi Server

f. Buka Kunci Pintu dari Web

```

webclient $ kirim_status loginput olahjson $ web

```

```

void olahResponseBukaPintu(const struct UserData *userData)
{
  bool updated = false;
  char noPintu[2];
  int pintu = 0;
  const char *temp1 = userData->control_door1;
  const char *temp2 = userData->control_door2;
  if (strcmp(temp1, "1") >= 0)
  {
    BeepBuzzer();
    delay(100);
    BeepBuzzer();

    updated = true;
    digitalWrite(ledSwPintu1, HIGH);
    strcpy(noPintu, "1");
    digitalWrite(pintu1, relayOn);
    amanPintu1 = aman;
    Serial.println("pintu1_terbuka_dari_web");
    pintu = 1;
  }
  else if (strcmp(temp2, "1") >= 0)
  {
    BeepBuzzer();
    delay(100);
    BeepBuzzer();

    updated = true;
    digitalWrite(ledSwPintu2, HIGH);
    strcpy(noPintu, "2");
  }
}

```

Gambar 4.31. Buka Kunci Pintu Dari Web

g. Buka kunci pintu dari tombol keluar

```

void scanKeluar()
{
  int pintu = 0;
  if (digitalRead(btnKeluar1) == 0)
  {
    BeepBuzzer();
    pintu = 1;
    digitalWrite(ledSwPintu1Out, HIGH);
    digitalWrite(pintu1, relayOn);

    Serial.println("ada pintu_1_terbuka_dari_tombol_keluar");
    Serial.println("ada menunggu limit switch aktif");
    tungguSWTerkena(pintu);
  }
  else if (digitalRead(btnKeluar2) == 0)
  {
    pintu = 2;
    BeepBuzzer();

    digitalWrite(ledSwPintu2Out, HIGH);
    digitalWrite(pintu2, relayOn);
    Serial.println("ada pintu_2_terbuka_dari_tombol_keluar");
    Serial.println("ada menunggu limit switch aktif");
    tungguSWTerkena(pintu);
  }
}

```

Gambar 4.32. Buka Kunci Pintu Dari Tombol Keluar

h. Cek buka pintu paksa/ ada maling

```

void scanMaling()
{
  int adaMaling = 0;
  if (digitalRead(swPintu1) == 0) // ada maling di pintu 1
  {
    delay(2000);
    if (digitalRead(swPintu1) == 0)
    {
      kirimLogKweb("sistem_mendeteksi_kondisi_tidak_aman");
      adaMaling = 1;
    }
  }
  else if (digitalRead(swPintu2) == 0) // ada maling d pintu 2
  {
    delay(2000);
    if (digitalRead(swPintu2) == 0)
    {
      kirimLogKweb("sistem_mendeteksi_kondisi_tidak_aman");
      adaMaling = 2;
    }
  }
  if (adaMaling != 0)
  // jika ada maling di pintu 1 atau pintu 2 , maka menunggu penekanan tombol darurat
  {
    int x = 0;
    while (adaMaling != 0)
    {
      BeepBuzzer();
      delay(200);
      if (adaMaling == 1)

```

Gambar 4.33. Cek Buka Pintu Paksa

i. Buka kunci pintu dari tombol darurat

```

webclient $  kirim_status  logicinput $  olahjson $  web
}
if (adaMaling != 0)
{
  int x = 0;
  while (adaMaling != 0)
  {
    BeepBuzzer();
    delay(200);
    if (adaMaling == 1)
    {
      if (digitalRead(btnDarurat1) == 0)
      {
        adaMaling = 0;
        digitalWrite(ledSwPintu1Out, HIGH);
        digitalWrite(pintu1, relayOn);
        delay(3000);
        tungguSWTerkena(1);
        delay(1000);
      }
    }
    else if (adaMaling == 2)
    {
      if (digitalRead(btnDarurat2) == 0)
      {
        adaMaling = 0;
        digitalWrite(ledSwPintu1Out, HIGH);
        digitalWrite(pintu2, relayOn);
        tungguSWTerkena(2);
        delay(1000);
      }
    }
  }
}
}

```

Gambar 4.34. Buka Kunci Pintu Dari Tombol Darurat

4.3 Pembahasan

Berdasarkan hasil pengujian secara keseluruhan yang telah dilakukan, diketahui sistem pengendali pintu dengan sistem keamanan berbasis web dan android ini berfungsi sesuai dengan perencanaan. Hasil pengujian diketahui melalui pengujian alat yang terdiri dari pengujian *driver solenoid*, *regulator*, *limit switch*, *push button* dan kecepatan mengunci pintu yang terbagi menjadi tiga pengujian yaitu membuka pintu melalui tombol pada web dan aplikasi android,

membuka pintu melalui tombol keluar dan membuka pintu melalui tombol darurat. Berikut ini merupakan pembahasan dari hasil pengujian tersebut, diantaranya :

Pada **Tabel 4.1.** hasil pengujian *driver solenoid (relay)* dilakukan dengan cara mengukur hasil tegangan keluaran pada saat *solenoid* aktif (*on*) dan tidak aktif (*off*). Hasil pengujian *Vout relay solenoid* saat aktif adalah 9,74 V untuk pintu 1 dan 9,35 V untuk pintu 2. Sedangkan *Vout relay solenoid* saat tidak aktif adalah 0 V untuk pintu 1 dan pintu 2.

Pada **Tabel 4.2.** hasil pengujian *regulator* dilakukan dengan cara mengukur hasil tegangan *input* dan *output* dari *regulator* ke arduino dan dari *regulator* ke *driver solenoid*. Hasil pengukuran tegangan *input* dari *regulator* ke arduino adalah 5,15 V, tegangan *output* dari *regulator* ke arduino sebesar 3,52 V, tegangan *input regulator* ke *driver* sebesar 3,52 V, dan tegangan *output regulator* ke *driver* sebesar 11,94 V.

Pada **Tabel 4.3.** hasil pengujian *limit switch* dilakukan dengan cara mengukur hasil tegangan *output* saat kondisi *limit switch* aktif (*on*) dan tidak aktif (*off*). Saat kondisi *limit switch* aktif, didapatkan tegangan 3,41 V saat pintu 1 terbuka dan 3,41 V saat pintu 2 terbuka. Saat kondisi *limit switch* tidak aktif (*off*), didapatkan tegangan 0 V saat pintu tertutup untuk pintu 1 dan pintu 2.

Pada **Tabel 4.4.** hasil pengujian *push button* terbagi ke dalam dua sistem yaitu *push button* untuk tombol keluar dan *push button* untuk tombol darurat. saat kondisi aktif, tombol keluar didapatkan hasil 3,41 V untuk pintu 1 dan 3,41 V untuk pintu 2. Sedangkan saat kondisi tidak aktif, tegangan pada pintu 1 dan pintu 2 adalah 0 V.

Pada **Tabel 4.5.** hasil pengujian buka pintu melalui web dan aplikasi android dilakukan sebanyak 5x percobaan masing-masing untuk pintu 1 dan pintu 2 dan didapatkan hasil dengan cara menghitung kecepatan buka pintu dengan hasil rata-rata 3,926 detik untuk pintu 1 dan 3,184 detik untuk pintu 2 dan selisih waktu untuk keduanya yaitu 0,742 detik.

Pada **Tabel 4.6.** hasil pengujian buka pintu melalui tombol keluar dilakukan sebanyak 5x percobaan masing-masing untuk pintu 1 dan pintu 2 dan didapatkan hasil dengan cara menghitung kecepatan buka pintu dengan hasil rata-rata 2,968 detik untuk pintu 1 dan 2,558 detik untuk pintu 2 dan selisih waktu untuk keduanya yaitu 0,41 detik.

Pada **Tabel 4.7.** hasil pengujian buka pintu melalui tombol darurat dilakukan sebanyak 5x percobaan masing-masing untuk pintu 1 dan pintu 2 dan didapatkan hasil dengan cara menghitung kecepatan buka pintu dengan hasil rata-rata 0,694 detik untuk pintu 1 dan 0,498 detik untuk pintu 2 dan selisih waktu untuk keduanya yaitu 0,196 detik.

Berdasarkan hasil pengujian secara keseluruhan yang telah dilakukan, diketahui bahwa alat pengendali pintu dengan sistem keamanan berbasis web dan android bekerja dengan baik, dan pengendali yang digunakan oleh penelitian alat ini yaitu *open loop* untuk buka kunci pintu 1 dan 2, *open loop* untuk buka kunci pintu dari tombol keluar, *open loop* untuk buka kunci pintu dari tombol darurat, dan *close loop* ketika pintu dibuka paksa karena sistem akan memberi peringatan melalui email pengguna yang terdaftar.

4.4 Aplikasi Hasil Penelitian

Aplikasi produk ini dapat diterapkan sebagai sebuah sistem dan perangkat elektronik yang dapat mempermudah dalam sistem keamanan pintu bagi penghuni rumah dengan kunci otomatis yang dapat dipantau keamanannya lewat jarak jauh sehingga kecil kemungkinannya bagi pencuri yang ingin masuk ke dalam rumah ketika pemilik rumah berada di luar rumah.

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Setelah melakukan percobaan, pengambilan data, dan penganalisaan terhadap data yang telah didapat pada penelitian ini, maka didapatkan kesimpulan yaitu sebagai berikut:

1. Solenoid sebagai pengunci pintu dapat dimanfaatkan sebagai pengunci pintu elektronik dengan berbagai pengoperasian yang dapat disesuaikan dengan kebutuhan.
2. Dapat mengendalikan kunci pintu rumah dari jarak jauh dengan menggunakan jaringan internet.
3. Dengan adanya halaman login sebagai *security sistem*, keamanan lebih terjaga karena kunci pintu hanya dapat dibuka oleh pemilik rumah atau pengguna yang sudah terdaftar oleh sistem.
4. Adanya dua perangkat yaitu *personal computer/ laptop* untuk mengakses melalui web dan *smartphone android* yang sudah terinstall aplikasi sistem alat ini, memberikan pilihan akses yang dapat mempermudah pemilik rumah dalam penggunaan alat ini.

5.2. Saran

Dalam penelitian ini tentu terdapat kekurangan yang harus diperbaiki. Berdasarkan penelitian yang telah dilakukan serta kesimpulan yang telah didapat, maka saran yang dapat diberikan demi pengembangan penelitian ini adalah:

1. Alat pengendali pintu dengan sistem keamanan berbasis web dan android dapat dikembangkan untuk diaplikasikan ke jenis pintu lainnya.
2. Alat pengendali pintu dengan sistem keamanan berbasis web dan android dapat dikembangkan menggunakan mikrokontroler jenis lain.
3. Alat pengendali pintu dengan sistem keamanan berbasis web dan android dapat dikembangkan menggunakan aplikasi *interface* lainnya yang lebih canggih dan modern.
4. Alat pengendali pintu dengan sistem keamanan berbasis web dan android dapat dikembangkan dengan menambah *webcam* untuk mendapatkan informasi kondisi pintu yang lebih aktual dan valid.
5. Alat pengendali pintu dengan sistem keamanan berbasis web dan android dapat dikembangkan dengan menambah sensor lainnya yang sesuai dengan kebutuhan.

DAFTAR PUSTAKA

- [FT] Fakultas Teknik. (2015). *Buku Panduan Penyusunan Skripsi dan Non Skripsi*. Jakarta: Fakultas Teknik, Universitas Negeri Jakarta.
- Antok. (2016). Alat Pengendali Kunci Pintu Rumah Jarak Jauh Menggunakan Internet Berbasis Arduino. *Universitas Negeri Jakarta*.
- Aswan, A. (2008). Konsep Perancangan Sistem Kontrol. In *Digilab ITB*. Bandung. Retrieved from <http://www.digilib.itb.ac.id/files/disk1/611/jbptitbpp-gdl-agungaswam-30508-3-2008ta-2.pdf>
- Firmansyah. (2009). Rancang Bangun Sistem Kontrol Penggerak Panel Sel Surya Berbasis Programmable Logic Controller. *Swateknologi*, 2(2), 15–21.
- <https://www.aliexpress.com/w/wholesale-3g-modem.html>, diakses pada tanggal 06 Juni 2017 pukul 08.00 WIB
- <http://www.ruangfreelance.com/cara-bisnis-online/website/>, diakses pada tanggal 06 Juni 2017 pukul 08.22 WIB
- Kadir, A. (2015). *Buku Pintar Pemrograman Arduino*. Yogyakarta: Mediacom.
- Muttaqin, A., Hafidudin, & Kurniawan, E. (2010). Pengecekan Pintu dan Pengunci Otomatis Jarak Jauh Berbasis SMS Gateway Dan Mikrokontroller. *Universitas Telkom*.
- Prayogo, N. A. (2016). Dasar Pemrograman Golang. Jakarta. Retrieved from <https://www.gitbook.com/book/novalagung/dasarpemrogramangolang/details>
- Prihono, dkk. (2009). *Jago Elektronika Secara Otodidak* (3rd ed). Jakarta: Penerbit PT Kawan Pustaka.

- Rafiudin, R. (2003). *Panduan Membangun Jaringan Komputer Untuk Pemula* (3rd ed.). Jakarta: PT. Elex Media Komputindo.
- Ramakumbo, A. G. (2012). *Magnetic Door Lock Menggunakan Kode Pengaman Berbasis At Mega 328*.
- Siregar, S. (2012). Rancang Bangun Sistem Monitoring Ruangan Terintegrasi berbasis Ethernet. *Politeknik Telkom*, 71–76.
- St, S., & Fahruzi, I. (2016). Sistem Pengaman Motor Menggunakan Smartcard Politeknik Negeri Batam Electrical Engineering study Program, 8(1), 1–5.
- Teknik, J., Politeknik, I., Lubuklinggau, Z. S., Seiring, A., Atmega, M., Uno, A., ... Tips, C. J. (2016). Rancang Bangun Pintu Geser Otomatis Berbasis Android, *V(2)*, 34–40.
- Utomo, E. P. (2012). *From Newbie to Advanced, Mudahnya Membuat Aplikasi Android*. Yogyakarta: Penerbit Andi.
- Widodo, B. (2006). *Membuat Robot Cerdas*. Jakarta: PT. Elex Media Komputindo.
- Winoto, P., & T, T. I. (2012). Pembuatan Website Profil Sekolah Dasar Negeri 03 Kalisoro, *4(1)*, 50–55.

LAMPIRAN-LAMPIRAN

Lampiran 1. Source Code Arduino IDE

```
//weclient

#include <ArduinoJson.h>

#include <Ethernet.h>

#include <SPI.h>

char *serverOffline = "192.168.137.1";

//char *serveronline = "ilma.smartdoor.karyateknologi.com";

char *serveronline = "104.168.87.182";

const unsigned long HTTP_TIMEOUT = 10000; // max response time from server

int portOffline = 10007;

int portOnline = 10007;

bool isOnlineWWW=true;
```

```
char server[50] = "";

int portWeb = 0;

// const char *server = "192.168.137.1"; // server's address

// int portWeb = 10007;

// const char *server = "ilma.smartdoor.karyateknologi.com";

// int portWeb = 80;

int jedaPowerOnPintu = 4000;

int sensorPin = A0;           // select the input pin for the potentiometer

int ledPin = 17;             // select the pin for the LED

int sensorValue = 0;        // variable to store the value coming from the
sensor

int pinBuzzer1 = 24;

int pinBuzzer2 = 25;

int pintu1 = 26;

int pintu2 = 27;

int btnDarurat1 = A8;

int swPintu1 = A13; // A9

int btnKeluar1 = A9; //A14

int btnDarurat2 = A11;

int swPintu2 = A12;

int btnKeluar2 = A14;

int lednya = 13;
```

```

int ledInternet = 7;

int ledSwPintu1 = 44;

int ledSwPintu2 = 45;

int ledSwPintu1Out = 46;

int ledSwPintu2Out = 47;

int selectServerLokal = A0;

int selectServerOnline = A1;

int ledSelectServer = 13;

// update value

int adaMaling = 2;

int aman = 1;

int standby = 0;

int amanPintu1 = standby;

int amanPintu2 = standby;

EthernetClient client;

int relayOn = 0;

int relayOff = 1;

const char *urlGetLastValue = "/v1/mikro/getlastvalue";

const char *urlUpdateDoorOpened = "/v1/mikro/updateclosealldoor?door="; //
http urlGetLastValue

const char *urlUpdateLog = "/v1/mikro/updatelogat?log="; // http log
alat update

const unsigned long BAUD_RATE = 9600; // serial
connection speed

const size_t MAX_CONTENT_SIZE = 512; // max size of
the HTTP response

// assign a MAC address for the ethernet controller.

```

```
// fill in your address here:

byte mac[] = {

    0xDE,

    0xAD,

    0xBE,

    0xEF,

    0xFE,

    0xED};

// fill in an available IP address on your network here,

// for manual configuration:

IPAddress ip(192, 168, 137, 177);

// fill in your Domain Name Server address here:

//IPAddress myDns(192, 168, 137, 1);

struct UserData

{

    char status_door1[5];

    char status_door2[5];

    char control_door1[5];

    char control_door2[5];

};

// ARDUINO entry point #1: runs once when you press reset or power the board

void setup()

{
```

```
initSerial();

initPinMikro();

initEthernet();
}

void loop()
{
    Serial.flush();

    if (connect(server))
    {
        Serial.println("membaca perintah buka pintu dari server dan membaca kondisi
tidak aman");

        if (sendRequest(server, urlGetLastValue) && skipResponseHeaders())
        {
            UserData userData;

            if (readResponseContent(&userData))
            {
                printUserData(&userData);

                (&userData);

                olahResponseBukaPintu(&userData);
            }
        }
    }

    disconnect();

    // scanTombolDarurat();
```

```
scanKeluar();  
scanMaling();  
// client.flush();  
wait();  
}  
  
// Initialize Serial port  
void initSerial()  
{  
    Serial.begin(BAUD_RATE);  
    Serial.println("Serial ready");  
}  
  
// Initialize Ethernet library  
void initEthernet()  
{  
    // byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED};  
    if(isOnlineWWW==true){  
        if (!Ethernet.begin(mac))  
        {  
            // Serial.println("Failed to configure DHCP, SET TO STATIC");  
            Ethernet.begin(mac, ip);  
        }  
    }else{  
        Ethernet.begin(mac, ip);  
    }  
}
```

```
Serial.println("Ethernet ready");

// delay(1000);

// start the Ethernet connection using a fixed IP address and DNS server:

// print the Ethernet board/shield's IP address:
Serial.print("My IP address: ");
Serial.println(Ethernet.localIP());
delay(2000);
}

void initPinMikro()
{
    delay(1000);

    pinMode(pinBuzzer1, OUTPUT);
    pinMode(pinBuzzer2, OUTPUT);
    pinMode(pintu1, OUTPUT);
    pinMode(pintu2, OUTPUT);
    pinMode(ledInternet, OUTPUT);
    pinMode(ledSwPintu1, OUTPUT);
    pinMode(ledSwPintu2, OUTPUT);
    pinMode(ledSwPintu1Out, OUTPUT);
    pinMode(ledSwPintu2Out, OUTPUT);
    for (int x = 0; x < 1; x++)
    {
        digitalWrite(pintu1, relayOn);
        digitalWrite(pintu2, relayOn);
        digitalWrite(pinBuzzer2, HIGH);
    }
}
```



```
digitalWrite(lednya, HIGH);  
digitalWrite(ledInternet, HIGH);  
digitalWrite(ledSwPintu1, HIGH);  
digitalWrite(ledSwPintu2, HIGH);  
digitalWrite(ledSwPintu1Out, HIGH);  
digitalWrite(ledSwPintu2Out, HIGH);  
delay(1000);  
digitalWrite(pintu1, relayOff);  
digitalWrite(pintu2, relayOff);  
digitalWrite(pinBuzzer2, LOW);  
digitalWrite(lednya, LOW);  
digitalWrite(ledInternet, LOW);  
digitalWrite(ledSwPintu1, LOW);  
digitalWrite(ledSwPintu2, LOW);  
digitalWrite(ledSwPintu1Out, LOW);  
digitalWrite(ledSwPintu2Out, LOW);  
delay(1000);  
}
```

```
pinMode(btnDarurat1, INPUT_PULLUP);  
pinMode(swPintu1, INPUT_PULLUP);  
pinMode(btnKeluar1, INPUT_PULLUP);  
pinMode(btnDarurat2, INPUT_PULLUP);  
pinMode(swPintu2, INPUT_PULLUP);  
pinMode(btnKeluar2, INPUT_PULLUP);  
pinMode(selectServerLokal, INPUT_PULLUP);
```

```
pinMode(selectServerOnline, INPUT_PULLUP);
memset(server,0,50);
Serial.print("select mode:");
if (digitalRead(selectServerLokal) == 0)
{
    digitalWrite(ledSelectServer, 1);
    strcpy(server, serverOffline);
    portWeb = portOffline;
    Serial.println("offline mode");
    isOnlineWWW=false;
}
else if (digitalRead(selectServerOnline) == 0)
{
    isOnlineWWW=true;
    digitalWrite(ledSelectServer, 0);
    strcpy(server, serveronline);
    portWeb = portOnline;
    Serial.println("online mode");
}

else
{
    digitalWrite(ledSelectServer, 1);
    strcpy(server, serverOffline);
    portWeb = portOffline;
    isOnlineWWW=false;
```

```
}  
  
Serial.print("server url:");  
  
Serial.println(server);  
  
}  
  
//kirim_status  
  
void kirimLogKeweb(char *paramLog)  
{  
    disconnect();  
    if (connect(server))  
    {  
        char x[100];  
        strcpy(x, urlUpdateLog);  
        strcat(x, paramLog);  
        if (sendRequest(server, x) && skipResponseHeaders())  
        {  
        }  
        delay(2000);  
        while (client.available() > 0)  
        {  
            char a = (client.read());  
            a = 0;  
        }  
    }  
}
```

```
}  
else  
{  
  
    Serial.println("ga konek");  
}  
}  
//logicinput  
  
void scanKeluar()  
{  
    int pintu = 0;  
    // Serial.println(digitalRead(btnKeluar1));  
    // Serial.println(digitalRead(btnKeluar2));  
    if (digitalRead(btnKeluar1) == 0)  
    {  
        BeepBuzzer();  
        pintu = 1;  
        digitalWrite(ledSwPintu1Out, HIGH);  
        digitalWrite(pintu1, relayOn);  
  
        Serial.println("ada pintu_1_terbuka_dari_tombol_keluar");  
        Serial.println("ada menunggu limit switch aktif");  
        // kirimLogKeweb("pintu_1_terbuka_dari_tombol_keluar");  
        tungguSWTerkena(pintu);  
    }  
}
```

```

}
else if (digitalRead(btnKeluar2) == 0)
{
    pintu = 2;
    BeepBuzzer();

    digitalWrite(ledSwPintu2Out, HIGH);
    digitalWrite(pintu2, relayOn);
    // delay(jedaPowerOnPintu);
    // digitalWrite(pintu2, relayOff);
    Serial.println("ada pintu_2_terbuka_dari_tombol_keluar");
    Serial.println("ada menunggu limit switch aktif");
    // kirimLogKeweb("pintu_2_terbuka_dari_tombol_keluar");
    tungguSWTerkena(pintu);
}
}

void scanMaling()
{
    int adaMaling = 0;
    if (digitalRead(swPintu1) == 0)
    {
        delay(2000);
        if (digitalRead(swPintu1) == 0)
        {
            kirimLogKeweb("sistem_mendeteksi_kondisi_tidak_aman");
        }
    }
}

```

```
    adaMaling = 1;
}
}
else if (digitalRead(swPintu2) == 0)
{
    delay(2000);
    if (digitalRead(swPintu2) == 0)
    {
        kirimLogKeweb("sistem_mendeteksi_kondisi_tidak_aman");
        adaMaling = 2;
    }
}
if (adaMaling != 0)
{
    int x = 0;
    while (adaMaling != 0)
    {
        BeepBuzzer();
        delay(200);
        if (adaMaling == 1)
        {
            if (digitalRead(btnDarurat1) == 0)
            {
                adaMaling = 0;
                digitalWrite(ledSwPintu1Out, HIGH);
                digitalWrite(pintu1, relayOn);
            }
        }
    }
}
```

```

        delay(3000);
        tungguSWTerkena(1);
        delay(1000);
    }
}
else if (adaMaling == 2)
{
    if (digitalRead(btnDarurat2) == 0)
    {
        adaMaling = 0;
        digitalWrite(ledSwPintu1Out, HIGH);
        digitalWrite(pintu2, relayOn);
        tungguSWTerkena(2);
        delay(1000);
    }
}
}
}
}

//olahjson

void olahResponseBukaPintu(const struct UserData *userData)
{
    bool updated = false;
    char noPintu[2];
    int pintu = 0;

```

```
const char *temp1 = userData->control_door1;
const char *temp2 = userData->control_door2;
if (strcmp(temp1, "1") >= 0)
{
    BeepBuzzer();
    delay(100);
    BeepBuzzer();

    updated = true;
    digitalWrite(ledSwPintu1, HIGH);
    strcpy(noPintu, "1");
    digitalWrite(pintu1, relayOn);
    // delay(jedaPowerOnPintu);
    amanPintu1 = aman;
    Serial.println("pintu1_terbuka_dari_web");
    // kirimLogKeweb("pintu1_terbuka_dari_web");
    pintu = 1;
}
else if (strcmp(temp2, "1") >= 0)
{
    BeepBuzzer();
    delay(100);
    BeepBuzzer();

    updated = true;
    digitalWrite(ledSwPintu2, HIGH);
```



```
strcpy(noPintu, "2");
digitalWrite(pintu2, relayOn);
// delay(jedaPowerOnPintu);
amanPintu2 = aman;
Serial.println("pintu2_terbuka_dari_web");
// kirimLogKeweb("pintu2_terbuka_dari_web");
pintu = 2;
}
if (updated == true)
{
disconnect();
if (connect(server))
{
char x[100];
strcpy(x, urlUpdateDoorOpened);
strcat(x, noPintu);
if (sendRequest(server, x) && skipResponseHeaders())
{
}
}
}

Serial.println("ada menunggu limit switch aktif");
tungguSWTerkena(pintu);
}
}
```

```
void tungguSWTerkena(int pintu)
{
  if (pintu == 1)
  {
    BeepBuzzer();
    while (digitalRead(swPintu1) != 0)
      ;
    {
      } //tunggu di buka
    BeepBuzzer();
    while (digitalRead(swPintu1) != 1)
      ;
    {
      } //tunggu di buka
    BeepBuzzer();
    delay(1000);
    digitalWrite(pintu1, relayOff);
  }
  else if (pintu == 2)
  {
    BeepBuzzer();
    while (digitalRead(swPintu2) != 0)
      {
        } //tunggu di buka
    BeepBuzzer();
    while (digitalRead(swPintu2) != 1)
```

```

    {
    } //tunggu di buka

    BeepBuzzer();

    delay(1000);

    digitalWrite(pintu2, relayOff);

    }

}

void BeepBuzzer()

{
    digitalWrite(pinBuzzer1, 1);

    delay(80);

    digitalWrite(pinBuzzer1, 0);

}

//web

// Open connection to the HTTP server

bool connect(const char *hostName)

{

    // Serial.print("Connect to ");

    // Serial.println(hostName);

    bool ok = client.connect(hostName, portWeb);

    Serial.println(ok ? "Connected" : "Connection Failed!");

    // if (ok)

    // {

```

```
// digitalWrite(ledInternet, HIGH);  
  
// }  
  
// else  
  
// {  
  
// digitalWrite(ledInternet, LOW);  
  
// }  
  
    return ok;  
  
}  
  
  
// Send the HTTP GET request to the server  
bool sendRequest(const char *host,  
                 const char *dataUrl)  
{  
    // Serial.print("GET ");  
    // Serial.println(dataUrl);  
  
    client.print("GET ");  
    client.print(dataUrl);  
    client.println(" HTTP/1.1");  
    client.print("Host: ");  
    client.println(host);  
    client.println("Connection: close");  
    client.println();  
    delay(1000);  
    return true;  
}
```

```
// Skip HTTP headers so that we are at the beginning of the response's body
bool skipResponseHeaders()
{
    // HTTP headers end with an empty line
    char endOfHeaders[] = "\r\n\r\n";
    // char endOfHeaders[] = "#null";

    client.setTimeout(HTTP_TIMEOUT);

    bool ok = client.find(endOfHeaders);

    if (!ok)
    {

        // ok = client.find(endOfHeaders);
        // if (!ok)
        // {
            Serial.println("gagal terhubung ke server, cek jaringan !");
        // }
    }

    if (ok)
    {
        digitalWrite(ledInternet, HIGH);
    }
}
```

```

else
{
    digitalWrite(ledInternet, LOW);
}
return ok;
}

// Parse the JSON from the input string and extract the interesting values
// Here is the JSON we need to parse

// "status_door1": "1",
// "status_door2": "0",
// "control_door1": "1",
// "control_door2": "1",

bool readReponseContent(struct UserData *userData)
{
    // Compute optimal size of the JSON buffer according to what we need to parse.
    // This is only required if you use StaticJsonBuffer.

    const size_t BUFFER_SIZE =
        JSON_OBJECT_SIZE(4) // the root object has 4 elements
        // + JSON_OBJECT_SIZE(7) // the "data" object has 7 elements
        + MAX_CONTENT_SIZE; // additional space for strings

    // Allocate a temporary memory pool
    DynamicJsonBuffer jsonBuffer(BUFFER_SIZE);

```

```

JsonObject &root = jsonBuffer.parseObject(client);

if (!root.success())
{
    Serial.println("JSON parsing failed!");
    return false;
}

strcpy(userData->status_door1, root["status_door1"]);
strcpy(userData->status_door2, root["status_door2"]);
strcpy(userData->control_door1, root["control_door1"]);
strcpy(userData->control_door2, root["control_door2"]);

return true;
}

// Print the data extracted from the JSON
void printUserData(const struct UserData *userData)
{
    Serial.print("status_door1 = ");
    Serial.println(userData->status_door1);
    Serial.print("status_door2 = ");
    Serial.println(userData->status_door2);
    Serial.print("control_door1 = ");
    Serial.println(userData->control_door1);
}

```

```
Serial.print("control_door2 = ");  
Serial.println(userData->control_door2);  
}  
  
// Close the connection with the HTTP server  
void disconnect()  
{  
// Serial.println("Disconnect");  
client.stop();  
delay(2000);  
// client.flush();  
}  
  
// Pause for a 1 minute  
void wait()  
{  
// Serial.println("Wait 60 seconds");  
// delay(60000);  
// Serial.println("jeda 2 detik");  
// delay(2000);  
}
```


Lampiran 2. Source Code SQLite

```
CREATE TABLE "db_alats" ("id" integer primary key autoincrement, "status_door1" varchar(255), "status_door2" varchar(255),
`id` integer PRIMARY KEY AUTOINCREMENT
`status_door1` varchar(255)
`status_door2` varchar(255)
`control_door1` varchar(255)
`control_door2` varchar(255)
`created_at` varchar(255)
`update_at` varchar(255)
CREATE TABLE "db_emails" ("id" integer primary key autoincrement, "nama_lengkap" varchar(255), "email" varchar(255),
`id` integer PRIMARY KEY AUTOINCREMENT
`nama_lengkap` varchar(255)
`email` varchar(255)
`created_at` varchar(255)
`update_at` varchar(255)
CREATE TABLE "db_log_alats" ("id" integer primary key autoincrement, "jenis" varchar(255), "created_at" varchar(255),
`id` integer PRIMARY KEY AUTOINCREMENT
`jenis` varchar(255)
`created_at` varchar(255)
`update_at` varchar(255)
CREATE TABLE "db_log_alats" ("id" integer primary key autoincrement, "jenis" varchar(255), "created_at" varchar(255),
`id` integer PRIMARY KEY AUTOINCREMENT
`jenis` varchar(255)
`created_at` varchar(255)
`update_at` varchar(255)
CREATE TABLE "db_log_users" ("id" integer primary key autoincrement, "jenis" varchar(255), "created_at" varchar(255),
`id` integer PRIMARY KEY AUTOINCREMENT
`jenis` varchar(255)
`created_at` varchar(255)
`update_at` varchar(255)
CREATE TABLE "db_logs" ("id" integer primary key autoincrement, "jenis" varchar(255), "created_at" varchar(255), "update_at" varchar(255),
`id` integer PRIMARY KEY AUTOINCREMENT
`jenis` varchar(255)
`created_at` varchar(255)
`update_at` varchar(255)
CREATE TABLE "db_users" ("id" integer primary key autoincrement, "username" varchar(255), "password" varchar(255))
CREATE TABLE sqlite_sequence(name,seq)
```

Lampiran 3. Source Code Go-Language

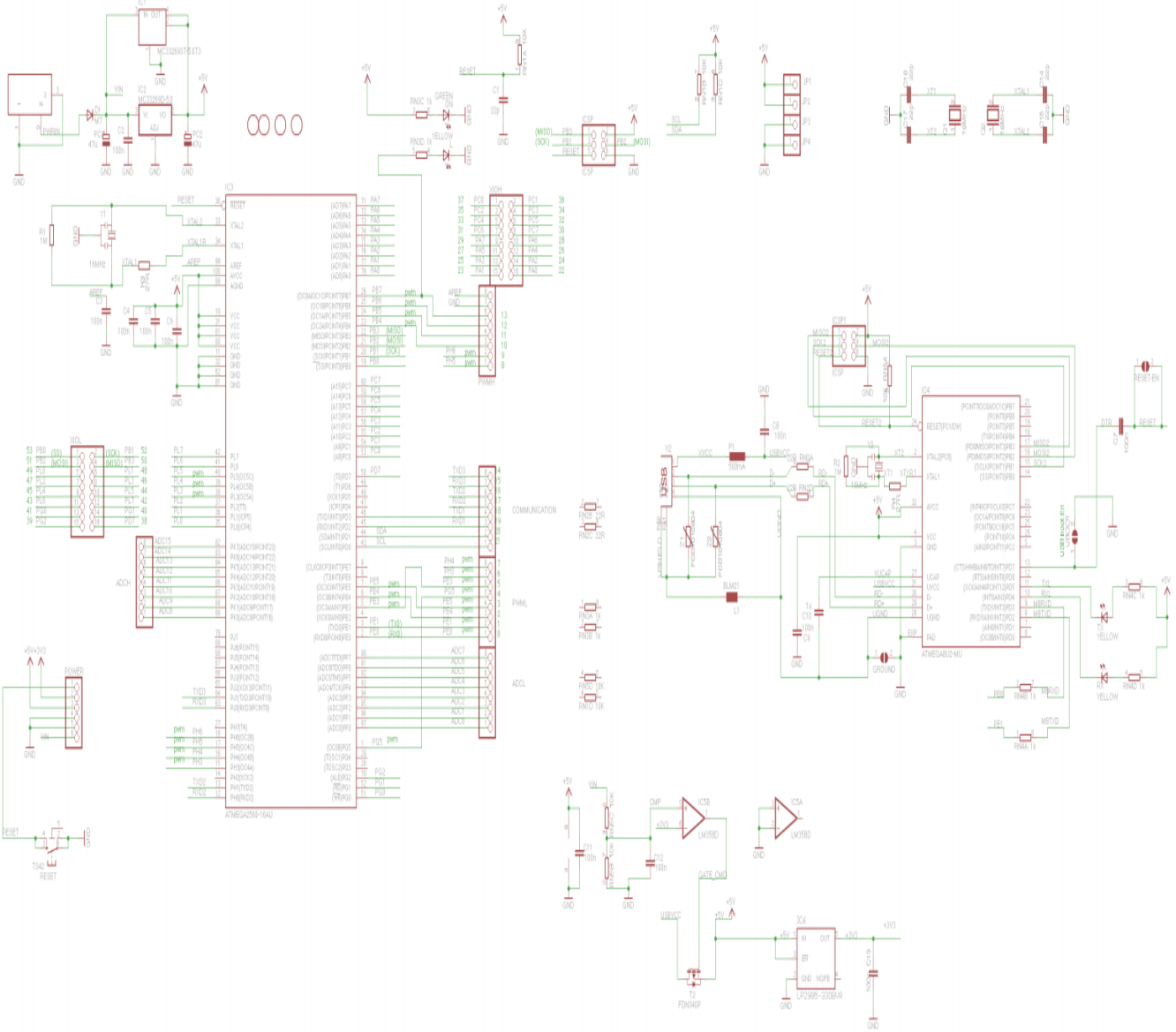
```
2017/08/11 11:31:51 [I] [router.go:244] C:\gopath\src\github.com\edot92\smart_door\server\controllers no changed
2017/08/11 11:31:52 [I] [router.go:244] C:\gopath\src\github.com\edot92\smart_door\server\controllers no changed
2017/08/11 11:31:52 [I] [router.go:244] C:\gopath\src\github.com\edot92\smart_door\server\controllers no changed
2017/08/11 11:31:52 [I] [router.go:244] C:\gopath\src\github.com\edot92\smart_door\server\controllers no changed
2017/08/11 11:31:52 [I] [asm_386.s:1629] http server Running on http://:10007
```

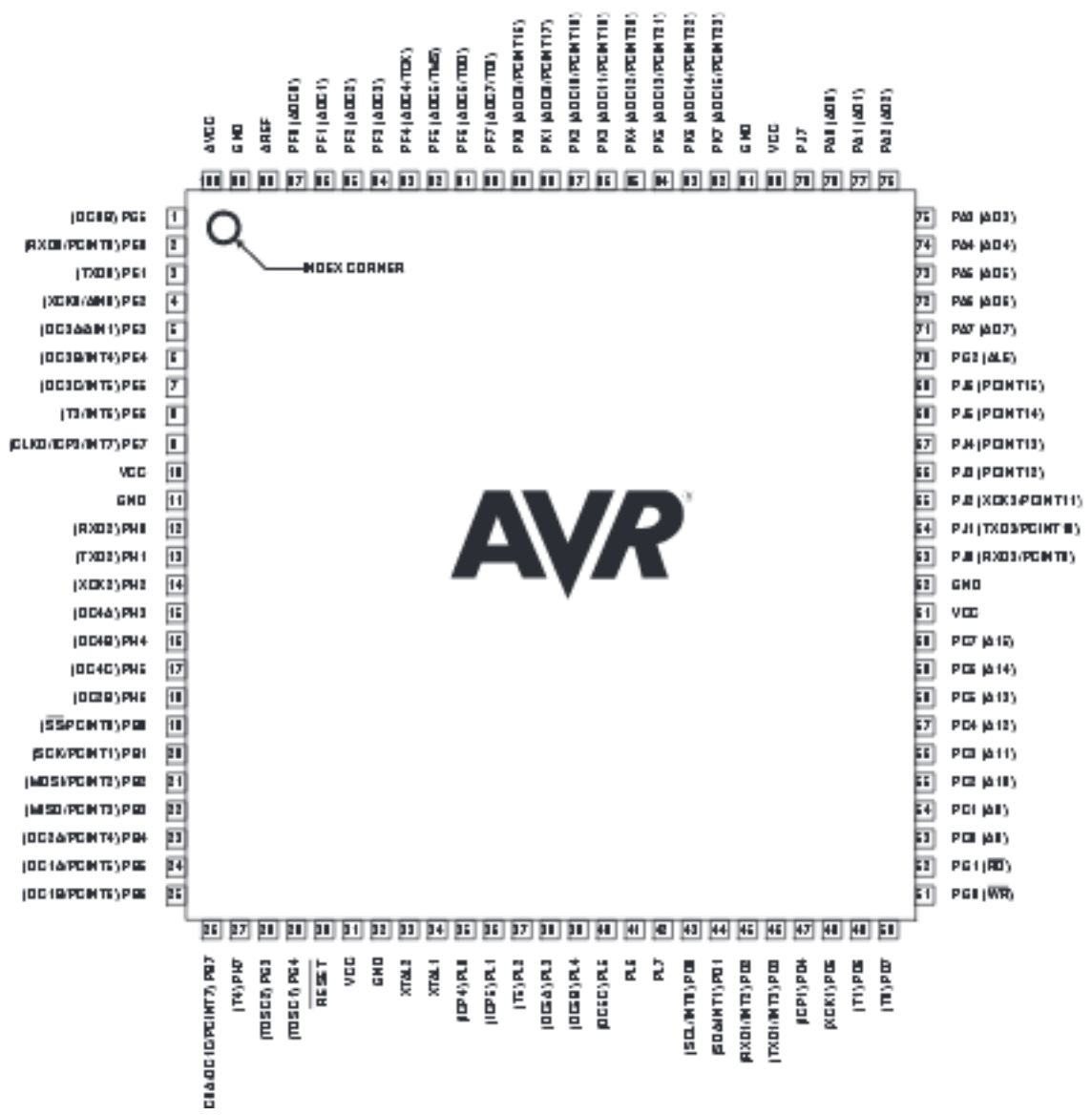
Lampiran 4. Skematik Arduino Mega 2560 dan Data Sheet At Mega 2560

Arduino Mega 2560 Reference Design

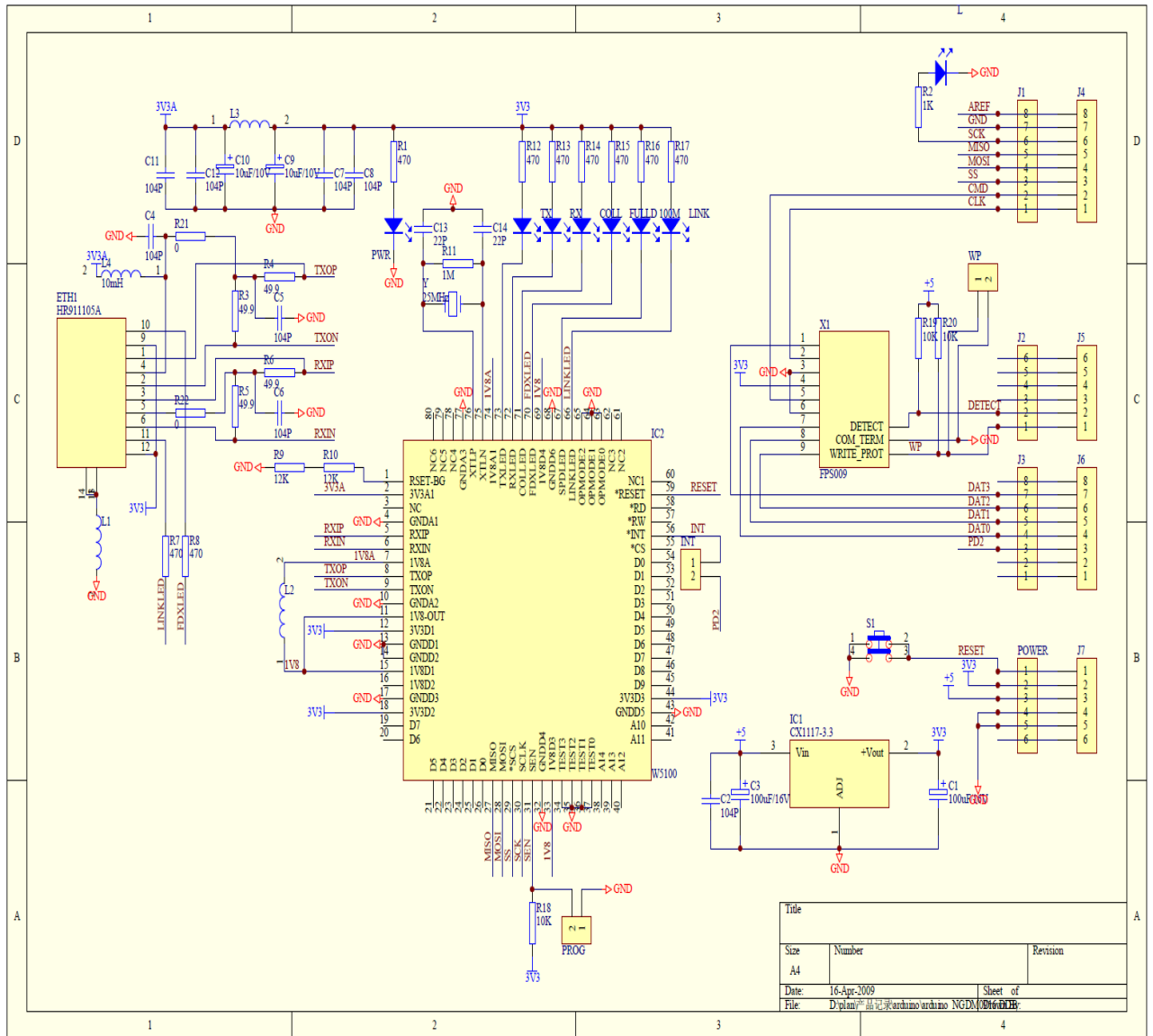
Reference Design ARE PROVIDED "AS IS" WITHOUT ANY WARRANTY, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Arduino may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any failures or instructions marked "reserved" or "obsolete". Arduino reserves the right to change specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any failures or instructions marked "reserved" or "obsolete".

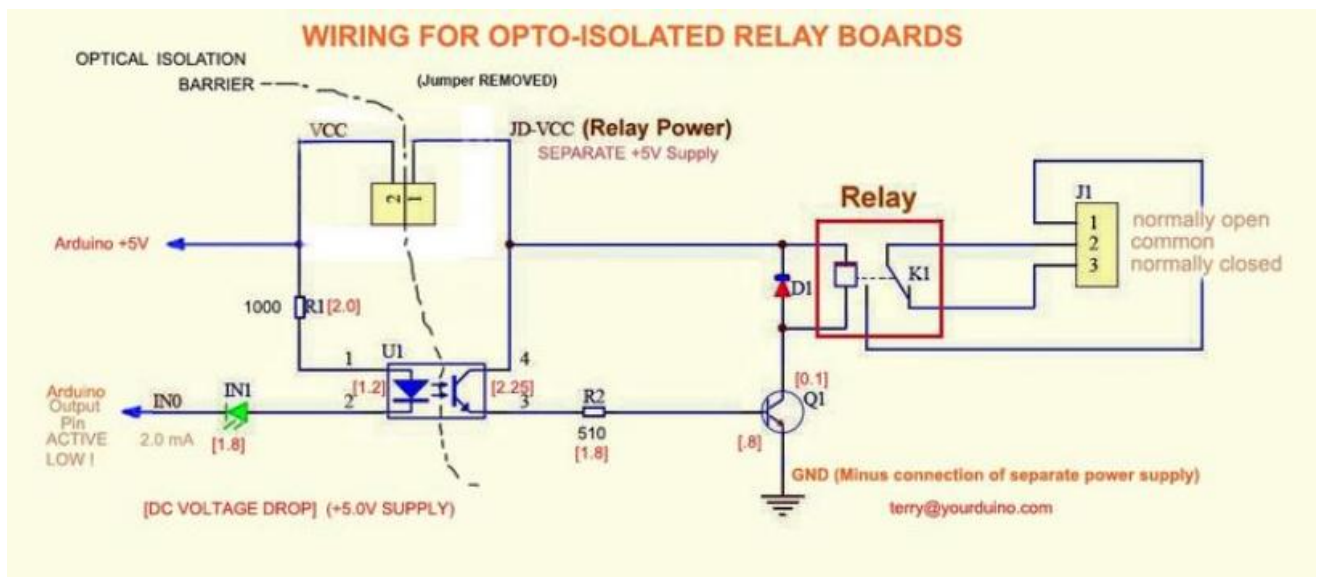




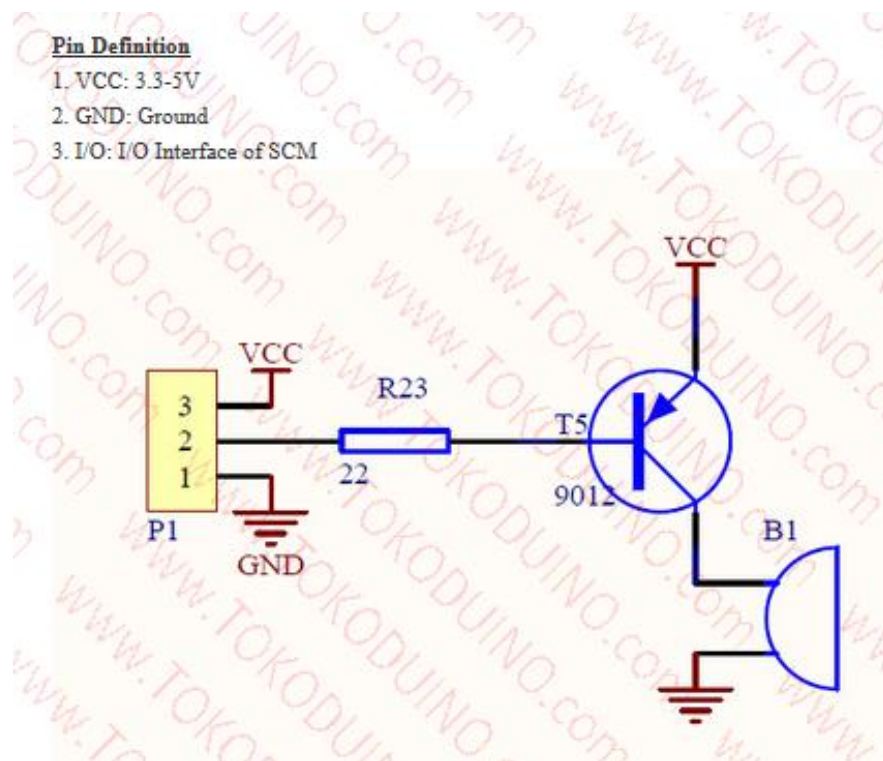
Lampiran 5. Skematik Ethernet Shield



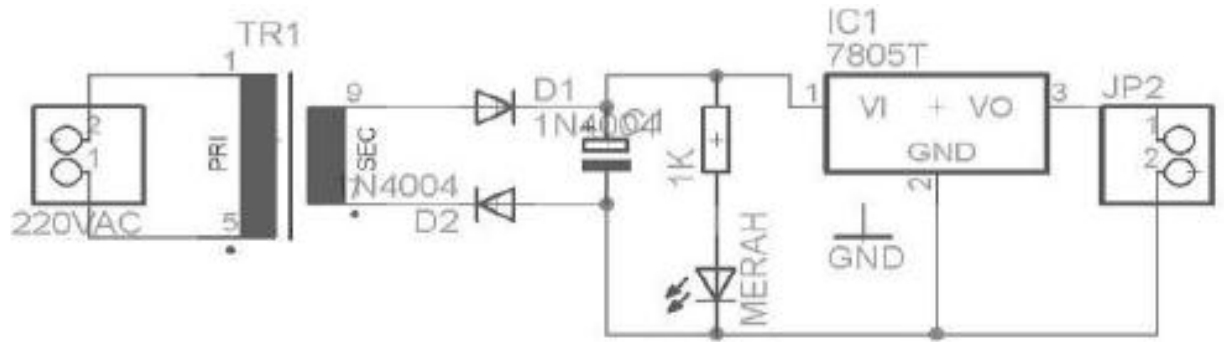
Lampiran 6. Skematik Driver Solenoid (Relay)



Lampiran 7. Skematik Rangkaian Buzzer



Lampiran 8. Skematik Rangkaian Regulator dan Data Sheet IC 7805



7805 • THREE-TERMINAL POSITIVE VOLTAGE REGULATOR IC

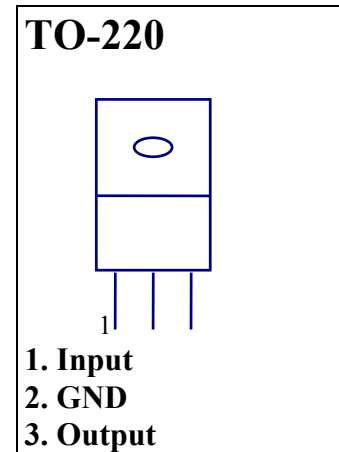
FEATURES:

- OUTPUT CURRENT IN EXCESS OF 1A;
- NO EXTERNAL COMPONENTS REQUIRED;
- INTERNAL SHORT CIRCUIT CURRENT LIMITING;
- INTERNAL THERMAL OVERLOAD PROTECTION;
- OUTPUT TRANSISTOR SAFE-AREA COMPENSATION;
- OUTPUT VOLTAGE OFFERED IN 4% TOLERANCE.

ABSOLUTE MAXIMUM RATINGS (Ta= 25° C)

Characteristic	Symbol	Norm	Unit
Input Voltage	V _{in}	V	35
Maximum Dissipated Power(with heat sink)	P _{tot(max)}	W	15
Maximum Dissipated Power(without heat sink)	P _{tot(max)}	W	1.5
Thermal Resistance Junction to Case	O _{jC}	°C/W	5.0
Thermal Resistance, Junction to Air	O _{jA}	°C/W	65
Junction Temperature	T _j	150	°C

T_c = -45 ÷ +70°C



ELECTRICAL CHARACTERISTICS

(V_{in}=10V, I_o=0.5A, C_i=0.33mkF, C_o=0.1mkF, T_j=0+125°C, unless otherwise noted.)

Characteristic	Symbol	Norm			Unit
		Min	TYP	Max	
Output Voltage(T _j =25°C)	V _o	4.8		5.2	V
Output Voltage (5.0mA ≤ I _o ≤ 1.0A, P _o ≤ 15W) 7.0V ≤ V _{in} ≤ 20V	V _o	4.75		5.25	V
Line Regulation(T _j =+25°C) 7.0V ≤ V _{in} ≤ 25 V 8.0 V ≤ V _{in} ≤ 12 V	ΔV _v			100 50	mV
Load Regulation(T _j =+25°C) 5.0mA ≤ I _o ≤ 1.5A 0.25A ≤ I _o ≤ 0.75A	ΔV _i			100 50	mV
Quiescent Current(T _j =+25°C)	I _b			8.0	mA
Quiescent Current Change 7.0 V ≤ V _{in} ≤ 25 V 5.0mA ≤ I _o ≤ 1.0 A	ΔI _b			1.3 0.5	mA
Dropout Voltage(I _o =1.0A, T _j =+25°C)	V _i - V _o		2.0		V
Short Circuit Current Limit(T _a =+25°C), V _{in} =35V	I _{sc}		0.4		A
Peak Output Current(T _j =+25°C)	I _{max}		2.2		A
Average Temperature Coefficient of Output Voltage	TCV _o		0.3		mV/°C