



**TUGAS AKHIR - TE 145561**

## **Rancang Bangun Sistem Pengaman Loker Susun Berbasis Mikrokontroler**

Hendra Wahyu Budianto  
NRP 10311500000058

Dosen Pembimbing  
Ir. Hany Boedinugroho, MT.

PROGRAM STUDI KOMPUTER KONTROL  
Departemen Teknik Elektro Otomasi  
Fakultas Vokasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2018





FINAL PROJECT - TE 145561

***CONSTRUCTION SECURITY SYSTEM OF STACKING LOKERS  
BASED ON MICROCONTROLLER***

Hendra Wahyu Budianto  
1031150000058

Supervisor  
Ir. Hany Boedinugroho, MT.

COMPUTER CONTROL STUDY PROGRAM  
Electrical and Automation Engineering Department  
Vocational Faculty  
Sepuluh Nopember Institute of Technology  
Surabaya 2018



## **PERNYATAAN KEASLIAN TUGAS AKHIR**

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul **“Rancang Bangun Sistem Pengaman Loker Susun Berbasis Mikrokontroler”** adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 24 Juli 2018



Hendra Wahyu Budianto  
NRP 1031150000058

-----Halaman ini sengaja dikosongkan-----

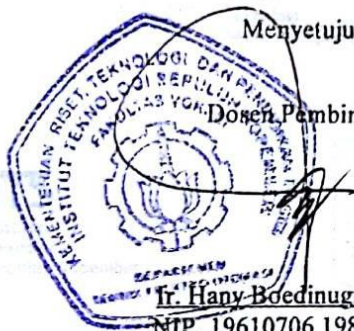
**RANCANG BANGUN SISTEM PENGAMAN LOKER SUSUN  
BERBASIS MIKROKONTROLER**

**TUGAS AKHIR**

Diajukan Guna Memenuhi Sebagian Persyaratan  
Memperoleh Gelar Ahli Madya Teknik  
Pada  
Departemen Teknik Elektro Otomasi  
Fakultas Vokasi  
Institut Teknologi Sepuluh Nopember

Menyetujui:

Dosen Pembimbing



Ir. Hany Boedinugroho, MT.  
NIP. 19610706 198701 1 001

**SURABAYA  
JULI, 2018**

-----Halaman ini sengaja dikosongkan-----



## RANCANG BANGUN SISTEM PENGAMAN LOKER SUSUN BERBASIS MIKROKONTROLER

**Nama** : Hendra Wahyu Budianto  
**Pembimbing** : Ir. Hany Boedinugroho, MT.

### ABSTRAK

Kemajuan teknologi semakin lama semakin berkembang pesat dalam berbagai aspek kehidupan, salah satunya adalah dalam bidang elektronika. Kemajuan teknologi elektronika turut membantu dalam pengembangan sistem keamanan yang lebih baik. Peminjaman loker pada suatu tempat seperti pada perpustakaan atau tempat perbelanjaan masih dilakukan secara konvensional yaitu dengan cara registrasi dan mengambil kunci loker terlebih dahulu pada penjaga. Sayangnya cara tersebut masih terbilang kurang efektif dikarenakan masih harus melakukan penjagaan dari pihak penyedia loker belum lagi apabila terjadi kehilangan kunci.

Berdasarkan uraian yang telah dijelaskan sebelumnya maka dibuatlah sistem mikrokontroler yang berfungsi sebagai pengaman setiap pintu loker. Diawali dengan cara pengguna memilih loker yang tersedia pada alat dan melakukan input *password* pada keypad yang telah disediakan sebagai pengaman pada loker. Kemudian pengguna akan mengambil barang dalam loker dengan cara memasukkan *password* yang telah dimasukkan sebelumnya dan pintu loker akan terbuka. *Password* akan ter-reset setelah pengguna mengambil barang bawaannya

Hasil dari Tugas Akhir ini adalah *keypad* dapat digunakan sebagai *input password*, kemudian *limit switch* dapat digunakan sebagai penanda apakah loker dalam keadaan tertutup atau terbuka. *Solenoid door lock* dapat mengunci ketika loker penuh dan *solenoid door lock* membuka pada saat loker dalam kondisi kosong. Pada alat ini memiliki presentase keberhasilan  $(73/80) \times 100\% = 91.25\%$  dengan presentase *error* sebesar  $100\% - 91.25\% = 8.75\%$ .

**Kata Kunci** : Loker, *Keypad*, Mikrokontroler, *password*

-----Halaman ini sengaja dikosongkan-----

## **CONSTRUCTION SECURITY SYSTEM OF STACKING LOKERS BASED ON MICROCONTROLLER**

**Name : Hendra Wahyu Budianto**  
**Supervisor: Ir. Hany Boedinugroho, MT.**

### **ABSTRACT**

*Technological progress is increasingly rapidly developing in various aspects of life, one of which is in the field of electronics. Advances in electronics technology helped to develop better security systems. Loan lockers at a place like the library or shopping place is still done in a conventional way by registration and take the locker lock first on the guard. Unfortunately the way is still somewhat less effective due to still have to take care of the provider of lockers not to mention in case of lost keys.*

*Based on the description that has been described previously, a microcontroller system that serves as a safety every locker door. Begins with the user select the available lockers on the tool and input the password on the keypad that has been provided as a safety on the locker. The user will then retrieve the items in the locker by entering the previously entered password and the locker door will open. The password will reset after the user retrieves the luggage.*

*The result of this final task is keypad can be used as input password, and limit switch can be used as a marker if locker closed or open condition. Solenoid door lock can lock when locker full and door lock solenoid open when locker in empty condition. In this tool has the percentage of success  $(73/80) \times 100\% = 91.25\%$  with percentage error of  $100\% - 91.25\% = 8.75\%$ .*

**Keywords : lockers, keypad, microcontroller, password**

-----Halaman ini sengaja dikosongkan-----

## **KATA PENGANTAR**

Puji syukur penulis panjatkan kehadiran Allah SWT yang selalu memberikan rahmat dan hidayah-Nya sehingga Tugas Akhir ini dapat terselesaikan dengan baik. Shalawat serta salam semoga selalu dilimpahkan kepada Rasulullah Muhammad SAW, keluarga, sahabat, dan umat muslim yang senantiasa meneladani beliau.

Tugas Akhir ini disusun untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan Diploma-3 pada Program Studi Komputer Kontrol, Departemen Teknik Elektro Otomasi, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember Surabaya dengan judul:

### **RANCANG BANGUN SISTEM PENGAMAN LOKER SUSUN BERBASIS MIKROKONTROLER**

Penulis mengucapkan terima kasih kepada Ibu dan Bapak penulis yang memberikan berbagai bentuk doa serta dukungan tulus tiada henti, Bapak Ir. Hany Boedinugroho, MT. atas segala bimbingan ilmu, moral, dan spiritual dari awal hingga terselesaikannya Tugas Akhir ini. Penulis juga mengucapkan banyak terima kasih kepada semua pihak yang telah membantu baik secara langsung maupun tidak langsung dalam proses penyelesaian Tugas Akhir ini.

Penulis menyadari dan memohon maaf atas segala kekurangan pada Tugas Akhir ini. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat dalam pengembangan keilmuan di kemudian hari.

Surabaya, 24 Juli 2018

Penulis

-----Halaman ini sengaja dikosongkan-----

# DAFTAR ISI

## HALAMAN

HALAMAN JUDUL .....	i
PERNYATAAN KEASLIAN TUGAS AKHIR .....	v
HALAMAN PENGESAHAN .....	v
ABSTRAK .....	ix
<i>ABSTRACT</i> .....	xi
KATA PENGANTAR .....	xiii
DAFTAR ISI .....	xv
DAFTAR GAMBAR .....	xvii
DAFTAR TABEL .....	xix
<b>1. BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Permasalahan .....	2
1.3 Tujuan .....	2
1.4 Batasan Masalah .....	3
1.5 Metodologi Penelitian .....	3
1.6 Sistematika Laporan .....	4
1.7 Relevansi .....	5
<b>2. BAB II TEORI DASAR .....</b>	<b>7</b>
2.1 Keamanan .....	7
2.2 <i>Keypad</i> .....	8
2.3 <i>Solenoid Door Lock</i> .....	9
2.4 <i>Relay</i> .....	10
2.5 Mikrokontroler ATmega 16 .....	13
2.5 ATmega 8 .....	17
2.6 <i>Limit Switch</i> .....	20
2.7 XL4005 .....	21
2.8 <i>Power Supply</i> .....	22
2.9 CodeVisionAVR .....	24
2.10 Bahasa C .....	26
2.11 Khazama .....	27
<b>3 BAB III PERENCANAAN DAN PEMBUATAN ALAT .....</b>	<b>31</b>
3.1 Blok Fungsional Sistem .....	31

3.2	Cara Kerja Alat.....	32
3.3	Perancangan <i>Hardware</i> .....	33
3.3.1	Rangkaian <i>Minimum System</i> ATmega16.....	33
3.3.2	Rangkaian Modul <i>Relay</i> .....	35
3.3.3	Rangkaian <i>Keypad</i> .....	36
3.4	Perancangan Software .....	36
3.4.1	<i>Flowchart Set Password</i> .....	39
3.5	Desain Kubus Kontroler .....	40
3.6	Desain Loker.....	42
<b>4.</b>	<b>BAB IV PENGUKURAN DAN PENGUJIAN .....</b>	<b>45</b>
4.1	Pengukuran Rangkaian <i>Power Supply</i> .....	45
4.2	Pengukuran <i>Limit Switch</i> .....	46
4.3	Pengukuran <i>Solenoid Door Lock</i> .....	48
4.4	Pengujian <i>Solenoid Door Lock</i> Terhadap Catu Daya .....	49
4.5	Pengujian <i>Password Keypad</i> .....	50
4.5.1	Pengujian <i>Keypad</i> .....	50
4.5.2	Pengujian <i>Password</i> .....	51
4.6	Pengujian Sensor <i>Coin Acceptor</i> CH-923.....	51
4.7	Petunjuk Penggunaan Alat.....	53
4.8	Pengujian Keseluruhan Sistem Alat .....	62
4.8.1	Fitur <i>Limit Switch</i> .....	62
4.8.2	Fitur <i>Password</i> .....	62
4.8.3	Fitur <i>Solenoid Door lock</i> .....	64
<b>5.</b>	<b>BAB V PENUTUP .....</b>	<b>71</b>
5.1	Kesimpulan.....	71
5.2	Saran .....	71
	<b>DAFTAR PUSTAKA.....</b>	<b>73</b>
	<b>LAMPIRAN .....</b>	<b>75</b>
	<b>RIWAYAT HIDUP PENULIS .....</b>	<b>101</b>



## DAFTAR GAMBAR

### HALAMAN

Gambar 2. 1 <i>Keypad</i> .....	8
Gambar 2. 2 Rangkaian Susunan Matriks ( <i>Keypad 4x4</i> ) .....	9
Gambar 2. 3 Bentuk Fisik <i>Solenoid door lock</i> .....	10
Gambar 2. 4 <i>Relay</i> .....	10
Gambar 2. 5 Bagian dan Prinsip Kerja Dari <i>Relay</i> .....	11
Gambar 2. 6 Jenis Relay Berdasarkan <i>Pole</i> dan <i>Throw</i> .....	12
Gambar 2. 7 Konfigurasi Mikrokontroler ATmega16 .....	14
Gambar 2. 8 Blok diagram fungsional Atmega16.....	16
Gambar 2. 9 Susunan Pin ATmega8 .....	17
Gambar 2. 10 <i>Limit Switch</i> .....	21
Gambar 2. 11 XL4005 .....	21
Gambar 2. 12 Diagram Blok <i>Power Supply</i> .....	22
Gambar 2. 13 <i>Trafo Step Down</i> .....	22
Gambar 2. 14 <i>Rectifier</i> pada <i>Power Supply</i> .....	23
Gambar 2. 15 <i>Filter</i> atau Penyaring pada <i>Power Supply</i> .....	24
Gambar 2. 16 Read chip signature .....	28
Gambar 2. 17 Pemberitahuan Chip Signature .....	28
Gambar 2. 18 Memasukkan File.hex .....	28
Gambar 2. 19 Pilih <i>File</i> .....	29
Gambar 2. 20 <i>File.hex</i> Berhasil di <i>Download</i> .....	29
Gambar 2. 21 Memilih <i>Manual</i> .....	29
Gambar 2. 22 Program Berhasil di <i>Download</i> .....	30
Gambar 3. 1 Blok Fungsional Sistem.....	31
Gambar 3. 2 Rangkaian Mikrokontroler ATmega 16 .....	33
Gambar 3. 3 Foto Rangkaian Mikrokontroler ATmega 16 .....	34
Gambar 3. 4 Rangkaian Modul Relay 4 <i>Channel</i> dengan <i>ATmega16</i> ...35	
Gambar 3. 5 Rangkaian <i>Keypad</i> pada pin ATmega 16 .....	36
Gambar 3. 6 <i>Flowchart</i> Proses Penitipan Barang .....	37
Gambar 3. 7 <i>Flowchart</i> Proses Pengambilan Barang.....	38
Gambar 3. 8 <i>Flowchart Set Password</i> .....	39
Gambar 3. 9 Desain Kubus Kontroler .....	40
Gambar 3. 10 Kubus Kontroler Tampak Depan.....	40

Gambar 3. 11 Kubus Kontroler Tampak Samping .....	41
Gambar 3. 12 Kubus Kontroler Tampak Atas .....	41
Gambar 3. 13 Desain Loker .....	42
Gambar 3. 14 Loker Tampak Depan.....	43
Gambar 3. 15 Loker Tampak Atas.....	43
Gambar 3. 16 Loker Tampak Samping.....	44
Gambar 3. 17 Rancangan Keseluruhan Alat.....	44
Gambar 4. 1 Petunjuk Penggunaan .....	54
Gambar 4. 2 Kondisi Loker .....	54
Gambar 4. 3 Buka Loker .....	55
Gambar 4. 4 Taruh Barang .....	55
Gambar 4. 5 <i>Set Password</i> .....	56
Gambar 4. 6 Masukkan <i>Password</i> .....	56
Gambar 4. 7 Hapus <i>Password</i> .....	57
Gambar 4. 8 Simpan <i>Password</i> .....	57
Gambar 4. 9 Petunjuk Penggunaan .....	58
Gambar 4. 10 Pilih Loker .....	58
Gambar 4. 11 <i>Enter Password</i> .....	59
Gambar 4. 12 Hapus <i>Password</i> .....	59
Gambar 4. 13 Tekan # .....	60
Gambar 4. 14 Notif Tarif Pembayaran .....	60
Gambar 4. 15 Masukkan <i>Coin</i> .....	61
Gambar 4. 16 Jenis <i>Coin</i> .....	61
Gambar 4. 17 Tampilan <i>Set Password</i> .....	62
Gambar 4. 18 Tampilan Perintah <i>Enter Password</i> .....	63
Gambar 4. 19 Tampilan <i>Password</i> Salah.....	63
Gambar 4. 20 Tampilan <i>Password</i> Benar .....	64
Gambar 4. 21 Tampilan Loker Penuh.....	64
Gambar 4. 22 <i>Solenoid Door Lock</i> Mengunci .....	65
Gambar 4. 23 Tampilan Loker Kosong .....	65
Gambar 4. 24 <i>Solenoid Door Lock</i> Terbuka .....	65

## DAFTAR TABEL

### HALAMAN

Tabel 4. 1 Pengukuran Tegangan pada <i>Power Supply</i> .....	45
Tabel 4. 2 Data Pengujian <i>Power Supply</i> .....	46
Tabel 4. 3 Pengukuran Tegangan pada <i>Limit Switch</i> Loker A .....	46
Tabel 4. 4 Pengukuran Tegangan pada <i>Limit Switch</i> Loker B .....	47
Tabel 4. 5 Pengukuran Tegangan pada <i>Limit Switch</i> Loker C .....	47
Tabel 4. 6 Pengukuran Tegangan pada <i>Limit Switch</i> Loker D .....	47
Tabel 4. 7 Arus <i>Solenoid Door Lock</i> 5 V .....	48
Tabel 4. 8 Arus <i>Solenoid Door Lock</i> 12 V .....	48
Tabel 4. 9 Pengujian <i>Solenoid Door Lock</i> 5 V Terhadap Catu Daya .....	49
Tabel 4. 10 Pengujian <i>Solenoid Door Lock</i> 12 V Terhadap Catu Daya .....	49
Tabel 4. 11 Pengujian <i>Keypad</i> .....	50
Tabel 4. 12 Pengujian <i>Password</i> .....	51
Tabel 4. 13 Pengujian <i>Coin Acceptor</i> Dengan <i>Coin</i> Rp 500 kuning .....	52
Tabel 4. 14 Pengujian <i>Coin Acceptor</i> Dengan <i>Coin</i> Rp 500 perak .....	52
Tabel 4. 15 Pengujian <i>Coin Acceptor</i> Dengan <i>Coin</i> Rp 200 perak .....	52
Tabel 4. 16 Pengujian <i>Coin Acceptor</i> Dengan <i>Coin</i> Rp 1000 perak .....	53
Tabel 4. 17 Hasil Pengujian Kinerja Alat Keseluruhan pada Loker A...66	
Tabel 4. 18 Hasil Pengujian Kinerja Alat Keseluruhan pada Loker B...67	
Tabel 4. 19 Hasil Pengujian Kinerja Alat Keseluruhan pada Loker C...67	
Tabel 4. 20 Hasil Pengujian Kinerja Alat Keseluruhan pada Loker D...68	

-----Halaman ini sengaja dikosongkan-----

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Perkembangan teknologi dari tahun ke tahun selalu mengalami perkembangan yang sangat pesat. Banyaknya fasilitas kemudahan-kemudahan yang ditimbulkan oleh perkembangan teknologi secara langsung berdampak kepada kehidupan manusia. Berkembangnya ilmu pengetahuan dan teknologi merupakan ciri dari perkembangan zaman. Salah satu contoh perkembangan teknologi tersebut adalah dalam bidang mikrokontroler. Perkembangan sistem berbasis mikrokontroler pada saat ini berkembang dengan sangat pesat, mulai dari sistem yang sederhana hingga sistem yang sangat fatal bagi sebuah perusahaan. Pada era kemajuan teknologi, kini banyak mahasiswa berlomba – lomba dalam menciptakan inovasi teknologi demi menunjang dalam membantu kebutuhan manusia. Perkembangan teknologi pada saat ini mencakup banyak sekali aspek kehidupan. Salah satunya adalah aspek keamanan. Maraknya pencurian barang hingga pembobolan tempat penyimpanan barang mendorong masyarakat berlomba -lomba untuk berinovasi membuat suatu sistem pengaman untuk mengurangi peluang bagi para pencuri tersebut mengambil barang milik orang lain. Salah satu contoh dari perkembangan sistem berbasis mikrokontroler saat ini yaitu sistem keamanan pada loker yang biasa terdapat pada tempat penyewaan loker yang tersedia di tempat – tempat umum. Loker merupakan tempat penyimpanan barang dimana biasa dipakai pada tempat-tempat wisata, perpustakaan, tempat olahraga ataupun tempat umum lainnya. Fungsi loker sebagai tempat penyimpanan seharusnya memiliki tingkat keamanan tinggi karena yang disimpan di dalamnya adalah barang-barang yang berharga. Keamanan sebuah loker sangat bergantung pada kunci pintunya. Pada dasarnya penitipan barang pada loker masih dilakukan secara konvensional yaitu ketika pengguna akan menitipkan barang pada loker maka pengguna harus mengisikan identitas terlebih dahulu ataupun pengguna akan diberi kunci loker untuk membuka dan menutup lokernya sendiri. Dalam hal ini sering kali terjadi kelalaian pada pihak penyewa loker dalam melakukan penjagaan loker. Salah satu kenyataannya adalah sering terjadinya pencurian dan kehilangan barang pada tempat penyewaan loker. Para pencuri dengan mudahnya dapat membuka pengunci

loker menggunakan seutas kawat atau dengan kunci tiruan lainnya. Selain itu kunci konvensional dapat dengan mudah untuk digandakan, rusak bahkan ada kemungkinan hilang atau lupa mengunci pintu loker. Selain itu sistem pembayaran pada penyewaan loker tersebut dinilai masih kurang efektif dikarenakan harus melibatkan orang dalam melakukan transaksi pembayaran sewa loker tersebut.

Dikarenakan penitipan barang pada loker masih dilakukan secara konvensional, maka dibuatlah inovasi terbaru yang dapat membantu mengamankan setiap loker tersebut dari tindak pencurian sehingga memberikan rasa aman bagi pengguna dan juga para penyewa loker tidak harus mengamati loker tersebut. Pembaruan inovasi tersebut dilakukan dengan cara memberikan sistem keamanan menggunakan *password* yang dimasukkan menggunakan keypad yang tersedia untuk dapat membuka setiap loker maupun untuk menutup setiap loker penitipan barang tersebut. *Password* pada setiap loker tersebut hanya diketahui oleh penggunanya saja.

## **1.2 Permasalahan**

Adapun permasalahan yang mendasari pembuatan dari Tugas Akhir ini adalah loker yang digunakan pada tempat-tempat umum seperti pada perpustakaan atau tempat perbelanjaan saat ini masih menggunakan sistem penguncian yang dilakukan secara konvensional dengan kunci berbentuk fisik. Sistem penguncian yang masih dilakukan secara konvensional dapat menimbulkan masalah, salah satunya adalah ketika pengunjung yang telah menyewa loker kemudian kehilangan kunci lokernya, akibatnya loker tersebut tidak dapat dibuka oleh penyewa loker tersebut. Serta penggunaan kunci yang berbentuk fisik dapat dengan mudah digandakan oleh setiap orang sehingga hal tersebut membahayakan barang yang ada di dalam loker.

## **1.3 Tujuan**

Tujuan yang akan dicapai dari Program Tugas Akhir ini adalah membuat setiap loker dapat diamankan dengan *password* yang dimasukkan menggunakan *keypad* sehingga meningkatkan pengamanan pada setiap loker karena tidak menggunakan kunci secara konvensional.

#### 1.4 Batasan Masalah

Pada tugas akhir ini, memiliki batasan-batasan masalah yang diambil, diantaranya :

1. *Password* yang dimasukkan hanya berupa kombinasi angka.
2. Banyaknya angka dari *password* yang dapat digunakan.
3. Loker dapat dibuka dan dikunci dengan *password*.

#### 1.5 Metodologi Penelitian

Dalam pelaksanaan tugas akhir berupa Rancang Bangun Sistem Pengaman Loker Susun Berbasis Mikrokontroler. Ada beberapa tahap yang perlu dipersiapkan yaitu sebagai berikut:

- **Pengamatan Permasalahan**

Pengamatan dilakukan untuk mengamati kinerja dari mikrokontroler yang telah dibuat yaitu dengan mengetahui dari *password* yang telah dimasukkan pada mikrokontroler dapat membuka pengaman pada setiap loker untuk menghindari nilai kesalahan dalam pengaplikasian program.

- **Studi Literatur**

Pada tahap ini akan dilakukan dengan cara mencari teori penunjang yang akan digunakan untuk mendukung perancangan dan peralatan alat pada Tugas Akhir ini. Dasar teori dan teori penunjang didapatkan dari buku, jurnal, artikel pada media cetak maupun pada media elektronik. Hal tersebut juga dilakukan untuk memperoleh referensi dari alat – alat sebelumnya yang membahas permasalahan dan tema yang sama. Sehingga dapat acuan untuk inovasi pada alat yang akan dibuat. Sumber-sumber dari studi literatur ini merupakan sumber yang relevan dengan pengerjaan alat dan sumber tersebut dapat dipertanggung jawabkan. Studi literatur yang akan digunakan yaitu mengenai pemrograman *password*.

- **Perancangan Alat**

Pada tahapan ini dilakukan pembuatan rancangan alat setelah melewati tahapan sebelumnya. Rancang bangun didesain sedemikian rupa sesuai dengan rencana yang diinginkan. Pada peletakan komponen seperti *solenoid door lock* dibuat secara presisi pada setiap pintu loker agar hasil sesuai dengan apa yang diinginkan. Untuk perancangan pada sistem ini digunakanlah *keypad* untuk memasukkan *password* sebelum dan sesudah menggunakan loker. Pertama

pengguna akan memilih dan membuka loker yang akan digunakan . Setelah memilih loker pengguna kemudian menggunakan *keypad* untuk memasukkan *password* maka *solenoid door lock* akan mengunci sehingga pengguna dapat menaruh barangnya pada loker yang telah dipilih. Kemudian pada saat akan mengambil barang, pengguna diwajibkan untuk memasukkan *password* yang telah dimasukkan sebelumnya. Setelah *password* yang dimasukkan benar maka loker akan terbuka.

- **Pengujian Alat dan Analisis Data**

Setelah pembuatan rancangan selesai, selanjutnya akan dilakukan pengujian alat secara langsung. Hal tersebut bertujuan untuk mengetahui apakah rancangan yang dibuat telah sesuai dengan apa yang direncanakan. Pada tahap ini juga akan dilakukan pengambilan data. Pengambilan data bertujuan untuk mengetahui apakah data yang dihasilkan sudah sesuai dengan data yang diinginkan sehingga apabila terjadi kesalahan pada alat maka kesalahan tersebut dapat diselesaikan dengan baik.

- **Kesimpulan**

Pada tahap ini akan menyimpulkan mengenai alat yang sudah dibuat sudah sesuai dengan yang diharapkan. Bagian kesimpulan berpacu pada data pengujian alat sehingga dapat menganalisa hasil data yang telah didapat untuk diberikan tindakan pada tahap berikutnya.

- **Penyusunan Laporan Akhir**

Tahap terakhir yang perlu dilakukan yaitu menyusun laporan akhir yang bertujuan sebagai bukti tertulis bahwa pernah melakukan penelitian mengenai alat tersebut dengan menggunakan metode yang ditetapkan. Laporan juga bertujuan memberikan referensi pada penelitian berikutnya.

## 1.6 Sistematika Laporan

Pembahasan Tugas Akhir ini akan dibagi menjadi lima Bab dengan sistematika sebagai berikut:

### **Bab I      Pendahuluan**

Bab ini meliputi latar belakang, permasalahan, tujuan penelitian, metodologi penelitian, sistematika laporan dan relevansi.



## **Bab II Teori Dasar**

Bab ini menjelaskan tentang tinjauan pustaka dari *hardware* yang digunakan, yaitu *keypad*, *solenoid door lock*, Relay, mikrokontroler ATmega16, ATmega8, *limit switch*, XL4005, *Power Supply*, CodeVisionAVR, Bahasa C dan Khazama. Sedangkan *software* yang digunakan adalah CV AVR.

## **Bab III Perancangan dan Pembuatan Alat**

Bab ini membahas perencanaan dan pembuatan perangkat keras (*Hardware*) yang meliputi desain alat dan aktuator yang digunakan dan pembuatan perangkat lunak (*Software*) yang meliputi program pada CodeVision AVR untuk menjalankan alat tersebut

## **Bab IV Pengukuran dan Pengujian**

Bab ini memuat tentang pemaparan dan analisis hasil pengujian alat pada keadaan sebenarnya. Seperti pengujian *keypad* yang digunakan dan juga *password* yang telah dibuat untuk mengamankan loker penitipan barang.

## **Bab V Penutup**

Bab ini berisi kesimpulan dan saran dari hasil pembahasan yang telah diperoleh.

### **1.7 Relevansi**

- a) Membuat pengunci loker dengan menggunakan *password*.
- b) Menghasilkan setiap loker yang dapat dikendalikan oleh mikrokontroler .

-----Halaman ini sengaja dikosongkan-----

## **BAB II**

### **TEORI DASAR**

Bab ini membahas mengenai teori dasar dari peralatan yang digunakan dalam Tugas Akhir yang berjudul Rancang Bangun Sistem Pengaman Loker Susun Berbasis Mikrokontroler. Uraian teori terdiri dari *hardware* dan *software* yang digunakan, antara lain: *Keypad*, *Solenoid Door Lock*, *Relay*, Mikrokontroler ATmega 16, ATmega 8, *Limit Switch*, *XL4005*, *Power Supply*, *CodeVisonAVR*, Bahasa C dan Khazama.

#### **2.1 Keamanan**

Keamanan berasal dari kata pokok "aman" yang berarti : bebas, terlindung dari bahaya, selamat, tidak membahayakan, yakin, dapat dipercaya, dapat diandalkan. Sedangkan "keamanan memiliki arti "suasana aman" ketenteraman, ketenangan. Keamanan tidak hanya mencegah rasa sakit atau cedera tapi keamanan juga dapat membuat individu aman dalam aktifitasnya, mengurangi stres dan meningkatkan kesehatan umum (Raharjo, 2005).

Keamanan fisik (*biologic safety*) merupakan keadaan fisik yang aman terbebas dari ancaman kecelakaan dan cedera (*injury*) baik secara mekanis, *thermis*, elektrik maupun bakteriologis. Kebutuhan keamanan fisik merupakan kebutuhan untuk melindungi diri dari bahaya yang mengancam kesehatan fisik, yang pada pembahasan ini akan difokuskan pada *providing for safety* atau memberikan lingkungan yang aman. Konsep dasar keamanan terkait dengan kemampuan seseorang dalam menghindari bahaya, yang ditentukan oleh pengetahuan dan kesadaran serta motivasi orang tersebut untuk melakukan tindakan pencegahan. Ada tiga faktor penting yang terkait dengan keamanan yaitu: tingkat pengetahuan dan kesadaran individu, kemampuan fisik dan mental untuk melakukan upaya pencegahan, serta lingkungan fisik yang membahayakan atau berpotensi menimbulkan bahaya. Karakteristik keamanan ada beberapa yaitu *Pervasiveness* (insidensi), *Perception* (persepsi), *Management* (pengaturan).

Keamanan bersifat *pervasive* artinya luas mempengaruhi semua hal. Artinya klien membutuhkan keamanan pada seluruh aktifitasnya seperti makan, bernafas, tidur, kerja, dan bermain. Persepsi seseorang tentang keamanan dan bahaya mempengaruhi aplikasi

keamanan dalam aktifitas sehari-harinya. Tindakan penjagaan keamanan dapat efektif jika individu mengerti dan menerima bahaya secara akurat. Ketika individu mengenali bahaya pada lingkungan klien akan melakukan tindakan pencegahan agar bahaya tidak terjadi dan itulah praktek keamanan. Pencegahan adalah karakteristik mayor dari keamanan. (Raharjo, 2005)

## 2.2 Keypad

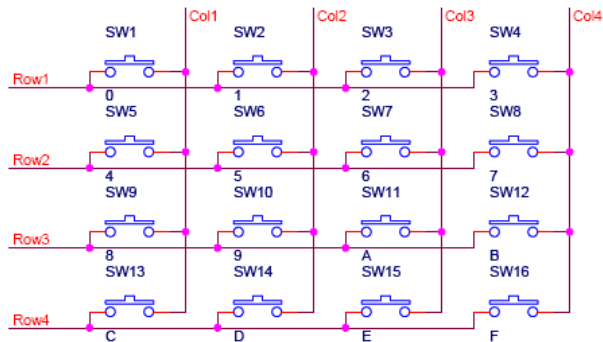
*Keypad* adalah rangkaian tombol yang berfungsi untuk memberi sinyal pada suatu rangkaian dengan menghubungkan pin-pin tertentu dari *Keypad*. *Keypad* terdiri dari beberapa macam berdasarkan jumlah tombol dan fungsinya. Dalam perancangan ini digunakan *Keypad* matriks 4x4 yang bila dilihat dari bentuk fisiknya memiliki 16 saklar dengan penghubung rangkaian sebanyak 8 pin. Gambar *Keypad* dapat dilihat pada Gambar 2.1 (Heri, 2015).



**Gambar 2. 1 Keypad** (Heri, 2015)

Proses pengecekan dari tombol yang dirangkai secara matriks adalah dengan teknik *scanning*, yaitu proses pengecekan yang dilakukan dengan cara memberikan umpan-data pada satu ba-

gian dan mengecek *feedback* (umpan-balik) pada bagian yang lain. Dalam hal ini, pemberian umpan data dilakukan pada bagian baris dan pengecekan umpan balik pada bagian kolom. Pada saat pemberian umpan data pada satu baris, maka baris yang lain harus dalam kondisi inversinya. Tombol yang ditekan dapat diketahui dengan melihat asal data dan di kolom mana data tersebut terdeteksi. Rangkaian susunan matriks dapat dilihat pada Gambar 2.2



**Gambar 2. 2** Rangkaian Susunan Matriks (*Keypad 4x4*) (Heri, 2018)

### 2.3 Solenoid Door Lock

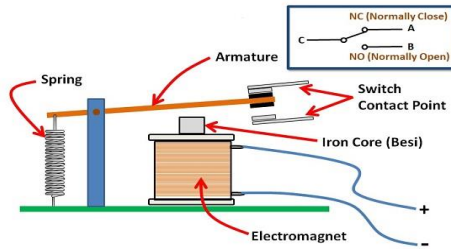
*Solenoid Door Lock* adalah alat elektronik yang dibuat untuk pengunci pintu yang dapat dikontrol oleh mikrokontroler. Alat ini sering digunakan pada Kunci Pintu Otomatis.

*Solenoid* ini akan bekerja apabila diberi tegangan. Arsitektur *Solenoid* terdiri dari kawat yang melingkar pada inti besi. Pada prinsip kerjanya ketika arus listrik mengalir melalui kawat ini, maka terjadi medan magnet untuk menghasikan energi yang akan menarik inti besi kedalam. (Gadre, 2001).

Pada kondisi normal *Solenoid* dalam posisi tuas memanjang atau terkunci. Jika *Solenoid* diberi tegangan tuas akan memendek atau terbuka. *Solenoid* ini dapat digabungkan dengan sistem pengunci elektrik yang berbasis *password*. Bentuk *solenoid door lock* dapat dilihat pada Gambar 2.3



Pada dasarnya, relay terdiri dari 4 komponen dasar yaitu : Electromagnet (Coil), Armature, Switch Contact Point (Saklar) dan Spring. Gambar 2.5 merupakan gambar bagian-bagian dari relay:



**Gambar 2. 5** Bagian dan Prinsip Kerja Dari *Relay* (John, 2007)

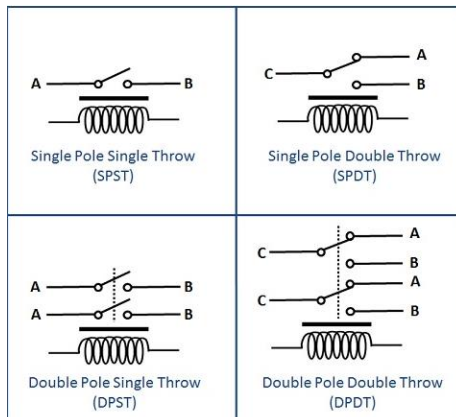
Kontak Poin (Contact Point) Relay terdiri dari 2 jenis yaitu :

1. Normally Close (NC) yaitu kondisi awal sebelum diaktifkan akan selalu berada di posisi CLOSE (tertutup)
2. Normally Open (NO) yaitu kondisi awal sebelum diaktifkan akan selalu berada di posisi OPEN (terbuka)

Berdasarkan gambar diatas, sebuah Besi (Iron Core) yang dililit oleh sebuah kumparan Coil yang berfungsi untuk mengendalikan Besi tersebut. Apabila Kumparan Coil diberikan arus listrik, maka akan timbul gaya Elektromagnet yang kemudian menarik Armature untuk berpindah dari Posisi sebelumnya (NC) ke posisi baru (NO) sehingga menjadi Saklar yang dapat menghantarkan arus listrik di posisi barunya (NO). Posisi dimana Armature tersebut berada sebelumnya (NC) akan menjadi OPEN atau tidak terhubung. Pada saat tidak dialiri arus listrik, Armature akan kembali lagi ke posisi Awal (NC). Coil yang digunakan oleh Relay untuk menarik Contact Poin ke Posisi Close pada umumnya hanya membutuhkan arus listrik yang relatif kecil. Karena Relay merupakan salah satu jenis dari Saklar, maka istilah *pole* dan *throw* yang dipakai dalam Saklar juga berlaku pada Relay. *Pole* adalah banyaknya kontak (*contact*) yang dimiliki oleh sebuah relay. Sedangkan *throw* adalah banyaknya kondisi yang dimiliki oleh sebuah Kontak (*Contact*). Berdasarkan penggolongan jumlah *Pole* dan *Throw*-nya sebuah relay, maka relay dapat digolongkan menjadi :

1. *Single Pole Single Throw (SPST)* : Relay golongan ini memiliki 4 Terminal, 2 Terminal untuk Saklar dan 2 Terminalnya lagi untuk Coil.
2. *Single Pole Double Throw (SPDT)* : Relay golongan ini memiliki 5 Terminal, 3 Terminal untuk Saklar dan 2 Terminalnya lagi untuk Coil.
3. *Double Pole Single Throw (DPST)* : Relay golongan ini memiliki 6 Terminal, diantaranya 4 Terminal yang terdiri dari 2 Pasang Terminal Saklar sedangkan 2 Terminal lainnya untuk *Coil*. Relay DPST dapat dijadikan 2 Saklar yang dikendalikan oleh 1 Coil.
4. *Double Pole Double Throw (DPDT)* : Relay golongan ini memiliki Terminal sebanyak 8 Terminal, diantaranya 6 Terminal yang merupakan 2 pasang Relay SPDT yang dikendalikan oleh 1 (*single*) *Coil*. Sedangkan 2 Terminal lainnya untuk *Coil*.

Selain Golongan Relay diatas, terdapat juga Relay-relay yang *Pole* dan *Throw*-nya melebihi dari 2 (dua). Misalnya 3PDT (*Triple Pole Double Throw*) ataupun 4PDT (*Four Pole Double Throw*) dan lain sebagainya. Untuk lebih jelas mengenai Penggolongan Relay berdasarkan Jumlah Pole dan Throw. Dapat dilihat pada gambar 2.6 :



**Gambar 2. 6** Jenis Relay Berdasarkan *Pole* dan *Throw* (John, 2007)



## 2.5 Mikrokontroler ATmega 16

Mikrokontroler merupakan suatu terobosan teknologi mikroprosesor dan mikrokomputer yang merupakan teknologi semikonduktor dengan kandungan transistor yang lebih banyak namun hanya membutuhkan ruang yang sangat kecil. Mikrokontroler merupakan sistem komputer yang mempunyai satu atau beberapa tugas yang sangat spesifik, berbeda dengan PC (Personal Computer) yang memiliki beragam fungsi (Heri, 2015).

Tidak seperti sistem komputer yang mampu menangani berbagai macam program aplikasi, mikrokontroler hanya bisa digunakan untuk suatu aplikasi tertentu saja, perbedaannya terletak pada perbandingan RAM (Random Access Memory) dan ROM (Read Only Memory) dimana program-program pengguna disimpan dalam ruang RAM (Random Access Memory) yang relatif besar, sedangkan prosedur antarmuka perangkat keras disimpan dalam ruang ROM (Read Only Memory) yang kecil. Sedangkan pada perbandingan mikrokontroler ROM (Read Only Memory) dan RAM (Random Access Memory) yang besar, artinya program kontrol disimpan dalam ROM (Read Only Memory) pada bias Masked ROM (Read Only Memory) atau Flash PEROM yang ukurannya relatif lebih besar, sedangkan RAM (Random Access Memory) digunakan sebagai tempat penyimpanan sementara, termasuk register-register yang digunakan pada mikrokontroler yang bersangkutan. (Wididio, 2008)

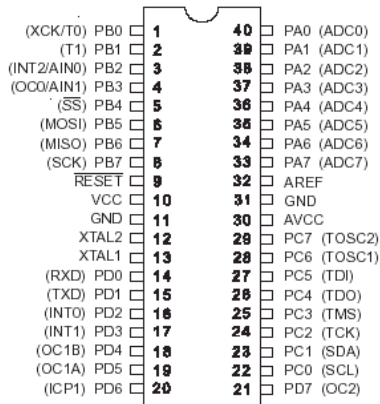
Mikrokontroler ATmega16 merupakan suatu sistem komputer yang seluruh atau sebagian besar elemennya dikemas dalam satu *chip* IC, sehingga sering juga disebut dengan *single chip* mikrokomputer. Mikrokontroler biasa dikelompokkan dalam satu keluarga, masing-masing mikrokontroler mempunyai spesifikasi tersendiri namun masih kompatibel dalam pemrogramannya.

Didalam pembuatan Tugas Akhir ini penulis memilih mikrokontroler AVR ATmega 16 sebagai prosesor dari alat yang akan dibuat. AVR merupakan seri mikrokontroler CMOS 8 bit buatan Atmel, berbasis arsitektur RISC (*Reduced Instruction Set Computer*). Atmel merupakan salah satu vendor yang bergerak dibidang mikroelektronika, telah mengembangkan AVR (*Alf and Vegard's Risc processor*) sekitar tahun 1997. Berbeda dengan mikrokontroler MCS51, AVR menggunakan arsitektur RISC (*Reduce Instruction Set Computer*) yang mempunyai lebar bus data 8 bit, perbedaan ini bisa

dilihat dari frekuensi kerjanya. MCS51 memiliki frekuensi kerja 1/12 kali frekuensi osilator sedangkan frekuensi kerja AVR sama dengan frekuensi osilator. Jadi dengan frekuensi osilator yang sama, kecepatan AVR 12 kali lebih cepat dibanding kecepatan MCS51.

Secara umum AVR dibagi menjadi 4 kelas, yaitu Attiny, AT90Sxx, ATmega dan AT86RFxx. Perbedaan antar tipe AVR terletak pada fitur-fitur yang ditawarkan, sementara dari segi arsitektur dan set instruksi yang digunakan hampir sama.

Konfigurasi pin mikrokontroler Atmega16 dengan kemasan 40 pin dapat dilihat pada Gambar 2.7. Dari gambar tersebut dapat terlihat ATmega16 memiliki 8 pin untuk masing-masing *port A* , *port B*, *port C*, dan *port D*. Atmega16 dapat dilihat pada Gambar 2.7



**Gambar 2. 7** Konfigurasi Mikrokontroler ATmega16(Wididio, 2008)

Adapun fitur Mikrokontroler ATmega16 kapabilitas detail dari adalah sebagai berikut :

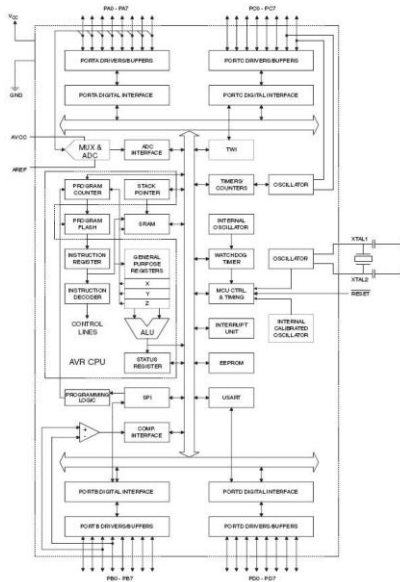
1. Mikrokontroler AVR 8 bit yang memiliki kemampuan tinggi, dengan daya rendah.
2. Arsitektur RISC dengan throughput mencapai 16 MIPS pada frekuensi 16MHz.
3. Memiliki kapasitas Flash memori 16 KByte, EEPROM 512 Byte dan SRAM 1 KByte.
4. Saluran I/O sebanyak 32 buah, yaitu port A, port B, port C, dan port D.
5. CPU yang terdiri atas 32 buah register.

6. Unit interupsi internal dan eksternal.
7. Port USART untuk komunikasi serial.
8. Fitur Peripheral.

Arsitektur ATmega16 mempunyai dua memori utama, yaitu memori data dan memori program. Selain itu, ATmega16 memiliki memori EEPROM untuk menyimpan data. ATmega16 memiliki 16 *Kbyte On-chip* di dalam sistem *Reprogrammable Flash Memory* untuk menyimpan program. Instruksi ATmega16 semuanya memiliki format 16 atau 32 bit, maka memori *flash* diatur dalam 8K x 16 bit.

Memori data (SRAM) AVR ATmega16 terbagi menjadi 3 bagian, yaitu 32 register umum, 64 buah register I/O dan 1 *Kbyte SRAM internal*. *General purpose* register menempati alamat data terbawah, yaitu \$00 sampai \$1F. Sedangkan memori I/O menempati 64 alamat berikutnya mulai dari \$20 hingga \$5F. Memori I/O merupakan register yang khusus digunakan untuk mengatur fungsi terhadap berbagai fitur mikrokontroler seperti kontrol register, *timer* atau *counter*, fungsi-fungsi I/O, dan sebagainya. 1024 alamat berikutnya mulai dari \$60 hingga \$45F digunakan untuk SRAM *internal*.

ATmega16 terdiri dari 512 *byte* memori data EEPROM 8 bit, data dapat ditulis atau dibaca dari memori ini, ketika catu daya dimatikan, data terakhir yang ditulis pada memori EEPROM masih tersimpan pada memori ini, atau dengan kata lain memori EEPROM bersifat *nonvolatile*. Alamat EEPROM mulai dari \$000 sampai \$1FF. Blok diagram fungsional ATmega16 dapat dilihat pada Gambar 2.8



**Gambar 2. 8** Blok diagram fungsional Atmega16(Wididio, 2008)

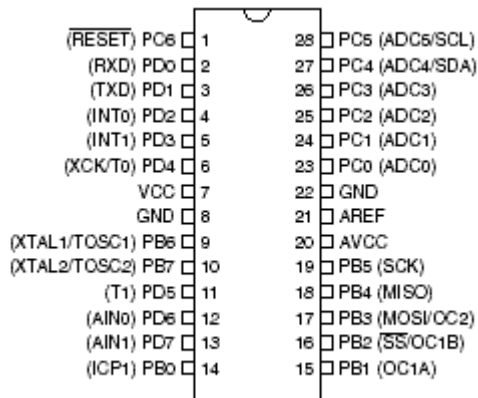
Dari gambar blok diagram tersebut dapat dilihat bahwa AT-Mega16 memiliki bagian-bagian sebagai berikut :

1. Saluran I/O sebanyak 32 buah, yaitu *port A*, *port B*, *port C* dan *port D*.
2. ADC 8 channel 10 bit.
3. Tiga buah *Timer* atau *Counter* dengan kemampuan pembanding.
4. CPU yang terdiri atas 32 buah register.
5. *Watchdog timer* dengan osilator *internal*.
6. SRAM sebesar 512 *byte*.
7. Memori *Flash* sebesar 8 KB dengan kemampuan *Read While Write*.
8. Interrupt internal dan eksternal.
9. *Port* antarmuka SPI (*Serial Peripheral Interface*).
10. EEPROM sebesar 512 *byte* yang dapat diprogram saat operasi.
11. Antarmuka komparator analog.
12. Port USART untuk komunikasi serial.

## 2.5 ATmega 8

ATMEGA 8 adalah mikrokontroler CMOS 8-bit daya rendah berbasis arsitektur RISC yang ditingkatkan. Kebanyakan instruksi dikerjakan pada satu siklus clock, ATMEGA 8 mempunyai throughput mendekati 1 MPS per MHz membuat disain dari sistem untuk mengoptimasi konsumsi daya versus kecepatan proses (John, 2007).

Susunan pin – pin dari IC mikrokontroler ATMEGA 8 diperlihatkan pada gambar dibawah ini. IC ini tersusun dari 28 pin yang memiliki beberapa fungsi tertentu. Susunan pin ATmega16 dapat dilihat pada Gambar 2.9



**Gambar 2. 9** Susunan Pin ATmega8(Wididio, 2008)

ATMega8 memiliki 28 pin yang masing – masing pin – nya memiliki fungsi yang berbeda – beda baik sebagai port ataupun sebagai fungsi yang lain. Berikut akan dijelaskan tentang kegunaan dari masing – masing kaki pada ATMega8.

1. VCC  
Merupakan supply tegangan untuk digital.
2. GND  
Merupakan ground untuk semua komponen yang membutuhkan grounding.
3. Port B  
Adalah 8 buah pin mulai dari pin B.0 sampai dengan pin B.7. Tiap pin dapat digunakan sebagai input dan juga output. Port B merupakan sebuah. 8-bit bit-directional I/O port dengan internal

pull-up resistor. Sebagai input, pin – pin yang terdapat pada port B yang secara eksternal diturunkan, maka akan mengeluarkan arus jika pull-up resistor diaktifkan. Jika ingin menggunakan tambahan kristal, maka cukup untuk menghubungkan kaki dari kristal ke kaki pada pin port B. Namun jika tidak digunakan, maka cukup untuk dibiarkan saja. Pengguna kegunaan dari masing – masing kaki ditentukan dari clock fuse setting-nya.

#### 4. Port C

Port C merupakan sebuah 7-bit bi-directional I/O yang di dalam masing – masing pin terdapat pull-up resistor. Jumlah pin-nya hanya 7 buah mulai dari C.0 sampai dengan pin C.6. Sebagai keluaran / output, port C memiliki karakteristik yang sama dalam hal kemampuan menyerap arus ( sink ) ataupun mengeluarkan arus ( source).

#### 5. Reset / PC6

Jika RSTDISBL Fuse diprogram, maka PC6 akan berfungsi sebagai pin I/O. Untuk diperhatikan juga bahwa pin ini memiliki karakteristik yang berbeda dengan pin – pin yang terdapat pada port C. Namun jika RSTDISBL Fuse tidak diprogram, maka pin ini akan berfungsi sebagai input reset. Dan jika level tegangan yang masuk ke pin ini rendah dan pulsa yang ada lebih pendek dari pulsa minimum, maka akan menghasilkan suatu kondisi reset meskipun clock-nya tidak berkerja.

#### 6. Port D

Port D merupakan 8-bit bi-directional I/O dengan internal pull-up resistor. Fungsi dari port ini sama dengan port – port yang lain. Hanya saja pada port ini tidak terdapat kegunaan-kegunaan yang lain. Pada port ini hanya berfungsi sebagai masukan dan keluaran saja atau biasa disebut dengan I/O.

#### 7. AVCC

Pada pin ini memiliki fungsi sebagai power supply tegangan untuk ADC. Untuk pin ini harus dihubungkan secara terpisah dengan VCC karena pin ini digunakan untuk analog saja. Bahkan jika ACD pada AVR tidak digunakan, tetap saja disarankan untuk menghubungkan secara terpisah dengan VCC. Cara menghubungkan AVCC adalah melewati low-pass filter setelah itu dihubungkan dengan VCC.

#### 8. AREF

Merupakan pin referensi analog jika menggunakan ADC. Pada AVR status Register mengandung beberapa informasi mengenai hasil dari kebanyakan hasil eksekusi intruksi aritmatik. Informasi ini dapat digunakan untuk altering arus program sebagai kegunaan untuk meningkatkan performa pengoperasian. Perlu diketahui bahwa register ini di-update setelah semua operasi ALU ( Arithmetic Logic Unit ). Hal tersebut seperti yang telah tertulis dalam datasheet khususnya pada bagian Intruction Set Reference. Dalam hal ini untuk beberapa kasus dapat membuang kebutuhan penggunaan instruksi perbandingan yang telah didedikasikan serta dapat menghasilkan peningkatan dalam hal kecepatan dan kode yang lebih sederhana dan singkat. Register ini tidak secara otomatis tersimpan ketika memasuki sebuah rutin interupsi dan juga ketika menjalankan sebuah perintah setelah kembali dari interupsi. Namun hal iini harus dilakukan melalui software.

9. Bit 7

Merupakan bit Global Interrupt Enable. Bit ini harus di-set supaya semua perintah interupsi dapat dijalankan. Jika bit ini di-reset, maka semua perintah interupsi baik yang secara individual maupun yang secara umum akan diabaikan. Bit ini akan dibersihkan atau cleared oleh hardware setelah sebuah interupsi dijalankan dan akan di-set kembali oleh perintah RETI. Bit ini juga dapat di-set dan di-reset melalui aplikasi dengan instruksi SEI dan CLI.

10. Bit 6

Merupakan bit Copy Storage. Instruksi bit Copy Instruction BLD ( Bit Load ) dan BST ( Bit Store ) menggunakan bit ini sebagai asal atau tujuan untuk bit yang telah dioperasikan. Sebuah bit dari sebuah register dan Register File dapat disalin ke dalam bit ini dengan menggunakan intruksi BST, dan sebuah bit di dalam bit ini dapat disalin ke dalam sebuah bit di register pada Register File dengan menggunakan perintah BLD.

11. Bit 5

Merupakan bit Half Carry Flag. Bit ini menandakan sebuah Half Carry dalam beberapa operasi aritmatika. Bit ini berfungsi dalam aritmatik BCD.

12. Bit 4

Merupakan Sign bit. Bit ini selalu merupakan sebuah eksklusif di antara Negative Flag (N) dan Two's Complement Overflow Flag (V).

13. Bit 3

Merupakan bit Two's Complement Overflow Flag. Bit ini menyediakan fungsi aritmatika dua komplemen.

14. Bit 2

Merupakan bit Negative Flag. Bit ini menyediakan sebuah hasil negative di dalam sebuah fungsi logika atau aritmatika.

15. Bit 1

Merupakan bit Zero Flag. Bit ini mengindikasikan sebuah hasil nol "0" dalam sebuah fungsi aritmatika atau logika.

16. Bit 0

Merupakan bit Carry Flag. Bit ini mengindikasikan sebuah Carry atau sisa dalam sebuah fungsi aritmatika atau logika.

## 2.6 *Limit Switch*

Limit switch (saklar pembatas) merupakan alat elektronik yang berupa saklar atau perangkat elektromekanis yang mempunyai tuas aktuator sebagai pengubah posisi kontak terminal (dari Normally Open/ NO ke Close atau sebaliknya dari Normally Close/NC ke Open). Posisi kontak akan berubah ketika tuas aktuator tersebut terdorong atau tertekan oleh suatu objek. Sama halnya dengan saklar pada umumnya, limit switch juga hanya mempunyai 2 kondisi, yaitu menghubungkan atau memutuskan aliran arus listrik. Dengan kata lain hanya mempunyai kondisi *on* atau *off* (Latif, 2012).

Namun sistem kerja limit switch berbeda dengan saklar pada umumnya, jika pada saklar umumnya sistem kerjanya akan diatur/dikontrol secara manual oleh manusia (baik diputar atau ditekan). Sedangkan limit switch dibuat dengan sistem kerja yang berbeda, limit switch dibuat dengan sistem kerja yang dikontrol oleh dorongan atau tekanan (kontak fisik) dari gerakan suatu objek pada aktuator, sistem kerja ini bertujuan untuk membatasi gerakan ataupun mengendalikan suatu objek/mesin tersebut, dengan cara memutuskan atau menghubungkan aliran listrik yang melalui terminal kontakannya sesuai dengan keadaan dari *Limit Switch*. *Limit Switch* dapat dilihat pada Gambar 2.10 (Latif, 2012)





**Gambar 2. 10** *Limit Switch* (Latif, 2012)

## 2.7 XL4005

XL4005 adalah konverter DC / DC frekuensi tetap 300MHz (step-down) DC, yang mampu menggerakkan beban 5A dengan efisiensi tinggi, riak rendah dan garis yang sangat baik serta pengaturan beban. Dengan mewajibkan jumlah minimum komponen eksternal, pengatur mudah digunakan dan termasuk kompensasi frekuensi internal dan osilator frekuensi tetap. Rangkaian kontrol PWM dapat menyesuaikan rasio tugas secara linear dari 0 hingga 100%. Fungsi pengaktifan, fungsi perlindungan lebih dari saat ini dibangun di dalam. Ketika fungsi perlindungan pendek terjadi, frekuensi operasi akan berkurang dari 300KHz menjadi 60KHz. Blok kompensasi internal dibangun untuk meminimalkan jumlah komponen eksternal. XL 4005 mempunyai Rentang tegangan *input* 5V - 32V, *output* tegangan dapat diatur 0.8V -30V, *duty cycle* maksimal 100%, *drop out* minimal 0.6V, frekuensi *switching* 300 KHz tetap, Kemampuan keluaran arus konstan 5A, optimalkan daya MOSFET internal, pengaturan jalur dan beban yang sangat baik, kemampuan shutdown TTL, EN pin dengan fungsi hysteresis, Dibangun dalam fungsi *thermal shutdown*, Dibangun pada output fungsi perlindungan singkat, tersedia dalam paket TO-263 (Iwan, 2006). XL4005 dapat dilihat pada Gambar 2.11

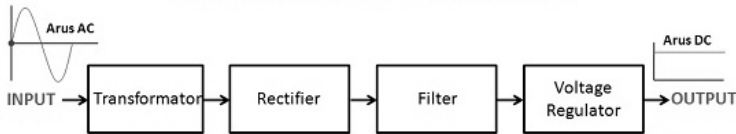


**Gambar 2. 11** XL4005 (Agfianto, 2002)

## 2.8 Power Supply

*Power Supply* atau pencatu daya merupakan rangkaian elektronika yang dapat menghasilkan energi listrik atau sebagai sumber energi untuk rangkaian elektronika lainnya. Sumber arus dari *power supply* adalah arus bolak – balik (AC) dari pembangkit listrik yang kemudian diubah menjadi arus searah (DC) (Wididio, 2005).

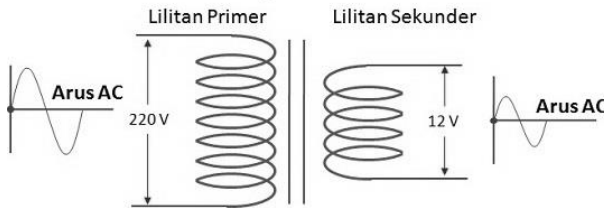
Untuk dapat melakukan hal tersebut *power supply* memerlukan perangkat yang bisa mengubah arus AC menjadi DC. Sebuah DC *power supply* pada dasarnya memiliki 4 bagian utama agar dapat menghasilkan arus DC yang stabil. Keempat bagian utama tersebut diantaranya adalah Transformator, *rectifier*, *filter* dan *voltage regulator*. Diagram blok *power supply* dapat dilihat pada Gambar 2.12



**Gambar 2. 12** Diagram Blok *Power Supply* (Wididio, 2005)

Berikut ini adalah penjelasan singkat tentang prinsip kerja DC *power supply* pada masing-masing blok berdasarkan diagram blok diatas. Trafo *step down* dapat dilihat pada Gambar 2.13

a. Transformator (Trafo)



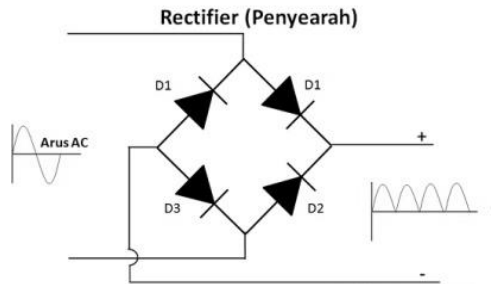
**Gambar 2. 13** Trafo *Step Down* (Wididio, 2005)

Transformator yang digunakan untuk DC *power supply* adalah Transformer jenis *Step-down* yang berfungsi untuk menurunkan tegangan listrik sesuai dengan kebutuhan komponen Elektronika yang terdapat pada rangkaian adaptor (DC *power supply*). Transformator bekerja berdasarkan prinsip Induksi elektromagnetik yang

terdiri dari 2 bagian utama yang berbentuk lilitan yaitu lilitan primer dan lilitan sekunder. Lilitan primer merupakan *input* dari transformator sedangkan *output*-nya pada lilitan sekunder. Meskipun tegangan telah diturunkan, *output* dari transformator masih berbentuk arus bolak-balik (arus AC) yang harus diproses selanjutnya (Wididio, 2005).

b. *Rectifier* (Penyearah Gelombang)

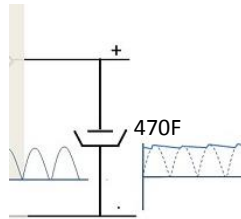
*Rectifier* atau penyearah gelombang adalah rangkaian elektronika dalam *power supply* yang berfungsi untuk mengubah gelombang AC menjadi gelombang DC setelah tegangannya diturunkan oleh transformator *step down*. Rangkaian *Rectifier* biasanya terdiri dari komponen dioda. Terdapat 2 jenis rangkaian *Rectifier* dalam *power supply* yaitu "*Half Wave Rectifier*" yang hanya terdiri dari 1 komponen dioda dan "*Full Wave Rectifier*" yang terdiri dari 2 atau 4 komponen dioda. *Rectifier* dapat dilihat pada Gambar 2.14



**Gambar 2. 14** *Rectifier* pada *Power Supply* (Agfianto, 2002).

c. *Filter* (Penyaring)

Dalam rangkaian *power supply* (Adaptor), *filter* digunakan untuk meratakan sinyal arus yang keluar dari *Rectifier*. *Filter* ini biasanya terdiri dari komponen kapasitor (kondensator) yang berjenis elektrolit atau ELCO (*Electrolyte Capacitor*). *Filter* dapat dilihat pada Gambar 2.15



**Gambar 2. 15** Filter atau Penyaring pada Power Supply (Agfianto, 2002)

d. Pengatur Tegangan (*Voltage Regulator*)

Untuk menghasilkan tegangan dan arus DC (arus searah) yang tetap dan stabil, diperlukan *voltage regulator* yang berfungsi untuk mengatur tegangan sehingga tegangan *output* tidak dipengaruhi oleh suhu, arus beban dan juga tegangan *input* yang berasal *output filter*. *Voltage regulator* pada umumnya terdiri dari Dioda Zener, Transistor atau IC (*Integrated Circuit*). Pada DC *power supply* yang canggih, biasanya *voltage regulator* juga dilengkapi dengan *short circuit protection* (perlindungan atas hubung singkat), *current limiting* (pembatas Arus) ataupun *over voltage protection* (perlindungan atas kelebihan tegangan) (Latif, 2012).

## 2.9 CodeVisionAVR

Ada banyak jenis software yang dapat digunakan sebagai editor yang sekaligus menyediakan compiler untuk mikrokontroler Atmel AVR dengan menggunakan bahasa C, diantaranya MikroC for AVR, WinAVR, Image Craft ICC AVR, IAR Embedded Workbench for AVR, dan CodeVision AVR. CodeVisionAVR menyediakan sebuah editor yang didesain untuk menghasilkan program C secara otomatis untuk mikrokontroler AVR. Program C yang akan diimplementasikan menggunakan standar ANSI C yang sesuai dengan arsitektur AVR. CodeVisionAVR adalah sebuah compiler C yang telah dilengkapi dengan fasilitas Integrated Development Environment (IDE) dan didesain agar dapat menghasilkan kode program secara otomatis untuk mikrokontroler Atmel AVR. Program ini dapat berjalan dengan menggunakan sistem operasi Windows® XP, Vista, Windows 7, dan Windows 8, 32-bit dan 64-bit. Integrated Development Environment (IDE) telah dilengkapi dengan fasilitas pemrograman chip melalui metode In-System Programming sehingga

ga dapat secara otomatis mentransfer file program ke dalam chip mikrokontroler AVR setelah sukses dikompilasi. Software In-System Programmer didesain untuk bekerja ketika dihubungkan dengan development board STK500, STK600, AVRISP mkII, AVR Dragon, AVRProg (AVR910 application note), Atmel JTAGICE mkII, Kanda System STK200+STK300, Dontronics DT006, Vogel Elektronik VTEC-SIP, Futurlec JRAVR and MicroTronics ATCPU, dan Mega2000. (Heri, 2013)

Untuk meningkatkan kehandalan program, maka pada CodeVisionAVR juga terdapat kumpulan pustaka (library) untuk:

1. Modul LCD Alphanumeric.
2. Philips I2C bus.
3. National Semiconductor Sensor Temperatur LM75.
4. Philips PCF8563, PCF8583, dan Maxim/Dallas Semiconductor Real Time Clock DS1302 dan DS1307.
5. Maxim/Dallas Semiconductor 1 wire protocol.
6. Maxim/Dallas Semiconductor Sensor Temperatur DS1820, DS18S20, dan DS18B20.
7. Maxim/Dallas Semiconductor Termometer/Thermostat DS1621.
8. Maxim/Dallas Semiconductor EEPROMs DS2430 dan DS2433.
9. SPI.
10. Power Management.
11. Delays.
12. Gray Code Conversion.
13. MMC/SD/SD HC Flash memory cards low level access.
14. Akses FAT pada MMC/SD/SD HC Flash memory card.

CodeVisionAVR dapat menghasilkan kode program secara otomatis melalui fasilitas CodeWizardAVR Automatic Program Generator. Dengan adanya fasilitas ini maka penulisan program dapat dilakukan dengan cepat dan lebih efisien. Seluruh kode dapat diimplementasikan dengan fungsi sebagai berikut:

1. Identifikasi sumber reset
2. Mengatur akses memori eksternal
3. Inisialisasi port input/output
4. Inisialisasi interupsi eksternal
5. Inisialisasi timer/counter dan watchdog timer

6. Inisialisasi USART dan interupsi buffer untuk komunikasi serial
7. Inisialisasi komparator analog dan ADC
8. Inisialisasi interface SPI dan two wire interface (TWI)
9. Inisialisasi interface CAN
10. Inisialisasi I2C Bus, sensor suhu LM75, thermometer/thermostat DS1621, dan real time clock PCF8563, PCF8583, DS1302, DS1307
11. Inisialisasi 1 wire bus dan sensor suhu DS1820/DS18S20
12. Inisialisasi modul LCD

## 2.10 Bahasa C

Bahasa Pemrograman C diciptakan dan dikembangkan oleh Brian Kernighan dan Denis Ritchie di Bell Research Labs. Bahasa Pemrograman C secara khusus diciptakan dengan tujuan agar para programmer (orang yang membuat program komputer) dapat mengakses seluruh internal register. I/O slots dan absolute address dari sebuah komputer. Pada awal tahun 1960-an, sistem operasi komputer mulai menjadi jauh lebih kompleks dari sebelumnya karena adanya pengenalan multi terminal dan kemampuan multi prosesor. Pada saat itu, sistem operasi diciptakan dengan menggunakan bahasa assembly (bahasa pemrograman tingkat rendah). Dimana banyak pengembang yang menyadari bahwa suatu sistem operasi dapat dikembangkan lebih lagi, tidak cukup hanya dengan menggunakan bahasa assembly. Inilah asal mula Bahasa C yang diimplementasikan pada Digital Equipment Corporation PDP-7. Pada perkembangan selanjutnya Bahasa Pemrograman C digunakan untuk mengimplementasikan Sistem Operasi Unix (Heri, 2015).

Bahasa C atau C++ adalah suatu bahasa pemrograman. Bahasa C termasuk sebagai bahasa pemrograman tingkat menengah, maksudnya bahasa C bisadipelajari dengan lebih mudah karena mudah dimengerti tetapi mempunyai kemampuan yang tinggi. Bahasa C bisa digunakan untuk merekayasa program untuk segala kebutuhan, baik untuk aplikasi bisnis, matematis atau bahkan game.

Bahasa Pemrograman C telah mengalami banyak evolusi sejak awal diciptakannya hingga saat ini. Saat ini, banyak Bahasa Pemrograman yang merupakan turunan/varian/keluarga dari Bahasa C seperti: C ++, Java Script, PHP, Java, perl dan lain sebagainya. Bisa

juga dibilang bahasa C adalah induk dari bahasa pemrograman saat ini. Beberapa kelebihan dari bahasa C adalah sebagai berikut :

1. Banyak memiliki operator untuk mengolah / memanipulasi data.
2. Bahasa C termasuk sebagai bahasa yang terstruktur sehingga program dapat lebih mudah dipahami atau dikembangkan.
3. Bahasa C lebih mudah dimengerti karena lebih mirip kepada bahasa manusia.
4. Mengenal data pointer.
5. Bahasa C tersedia hampir di semua jenis computer.
6. Kode bahasa C sifatnya adalah portable dan fleksibel untuk semua jenis computer.
7. Bahasa C hanya menyediakan sedikit kata-kata kunci, hanya terdapat 32 kata kunci.
8. Proses executable program bahasa C lebih cepat.
9. Dukungan pustaka yang banyak.
10. C adalah bahasa yang terstruktur.
11. Bahasa C termasuk bahasa tingkat menengah

### **2.11 Khazama**

Khazama AVR programmer merupakan salah satu software untuk menulis (*mendownload*) file. hex ke board mikrokontroler. Tujuan dari program ini adalah ukuran yang ringan, program cepat, handal dan mudah digunakan. Langkah – langkah penggunaan Khazama :

1. Buka program khazama
2. pilih *read chip signature* ( berfungsi membaca *chip/* dapat difungsikan)



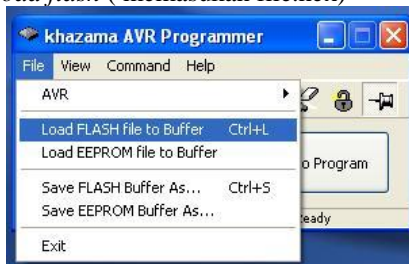
**Gambar 2. 16** Read chip signature

3. Muncul chip signature (menandakan mikrokontroler bisa difungsikan)



**Gambar 2. 17** Pemberitahuan Chip Signature

4. Pilih file> *load flash* ( memasukan file.hex)



**Gambar 2. 18** Memasukkan File.hex





Gambar dibawah, menunjukan program berhasil di download



**Gambar 2. 22** Program Berhasil di *Download*

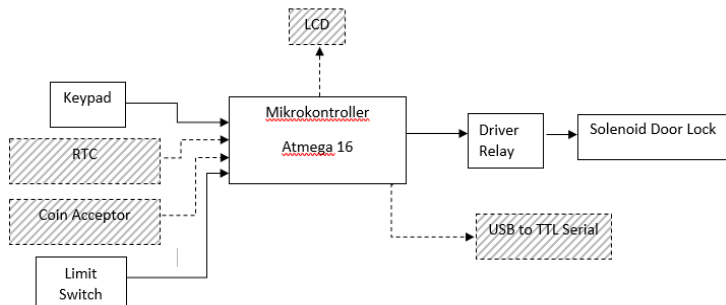
## BAB III PERENCANAAN DAN PEMBUATAN ALAT

Pada bab ini membahas tentang tahapan yang dilakukan terhadap perancangan dan pembuatan Tugas Akhir yang berjudul Rancang Bangun Sistem Pengaman Loker Susun Berbasis Mikrokontroler. Blok diagram alat Tugas Akhir pada gambar 3.1 merupakan sistem kerja Tugas Akhir Rancang Bangun Sistem Pengaman Loker Susun Berbasis Mikrokontroler. Sistem kerja keseluruhan di kontrol oleh mikrokontroler ATmega 16 yang terintegrasi dengan rangkaian dan komponen – komponen pendukung. Adapun fungsi masing – masing komponen yang terdapat pada diagram perancangan alat Tugas Akhir, antara lain:

### 3.1 Blok Fungsional Sistem

Sebelum Melakukan perancangan perangkat keras dan perangkat lunak, diperlukan sebuah perancangan blok fungsional sistem berupa blok diagram yang menjelaskan sistem kerja secara keseluruhan Tugas Akhir ini, untuk bagian yang diarsir dikerjakan oleh rekan kelompok. Secara keseluruhan blok fungsional sistem dapat dilihat pada Gambar 3.1.

Pada perancangan *hardware* dibagi menjadi beberapa sub bab yang akan dijelaskan per sub bab nya, antara lain :



**Gambar 3. 1** Blok Fungsional Sistem

Sesuai dengan gambar blok fungsional sistem diatas Mikontroler ATmega 16 digunakan sebagai kontroler untuk menga- tur masukan dari Keypad, dan juga Mikrokontroler ATmega 16

mengontrol kerja dari *Limit Switch* agar dapat mendeteksi keadaan pintu loker apakah masih dalam keadaan terbuka atau telah tertutup dan juga menghubungkannya dengan *Solenoid Door Lock*. Untuk mengatur tegangan pada Solenoid Door Lock juga dibutuhkan *Driver Relay*. Fungsi dari masing – masing komponen juga dapat dilihat dibawah ini:

1. Mikrokontroler ATmega 16  
Merupakan bagian yang terpenting pada sistem kerja yang berfungsi untuk mengolah *input*, kemudian diproses untuk dapat memberi perintah pada *output*.
2. *Keypad*  
*Keypad* berfungsi sebagai *input password* bagi pengguna loker.
3. Modul *Relay*  
Digunakan untuk menggerakkan arus/tegangan *solenoid door lock*.
4. *Solenoid Door Lock*  
Berfungsi sebagai pengunci pada loker.
5. *Limit Switch*  
Mendeteksi keadaan pintu loker dalam keadaan terbuka atau tertutup.

### 3.2 Cara Kerja Alat

Pada Tugas Akhir ini dilakukan cara kerja alat. Cara kerja alat tersebut adalah *Keypad* berfungsi sebagai *input password* melalui ATmega 16. *Keypad* yang digunakan adalah *keypad 4x4*. *Keypad 4x4* mempunyai sebanyak 8 buah pin yang akan terhubung pada pin digital dari mikrokontroler ATmega 16 Untuk menitipkan barang, penyewa loker harus terlebih dahulu melihat tampilan LCD dan pilih loker yang masih dalam keadaan kosong. Setelah itu buka pintu loker. Akan muncul tampilan *set password* untuk memasukkan *password* setelah *password* disimpan maka pintu pada loker akan mengunci. Untuk mengambil barang yang telah ditiptkan pada loker sebelumnya. Penyewa loker harus membuka loker terlebih dahulu. Membuka loker tersebut mempunyai beberapa tahapan yaitu membuka loker terlebih dahulu dengan menggunakan *password* yang telah di *set* sebelumnya. Setelah *password* benar maka penyewa harus melakukan proses pembayaran terlebih dahulu. Setelah membayar maka loker akan terbuka dan barang dapat diambil kembali.

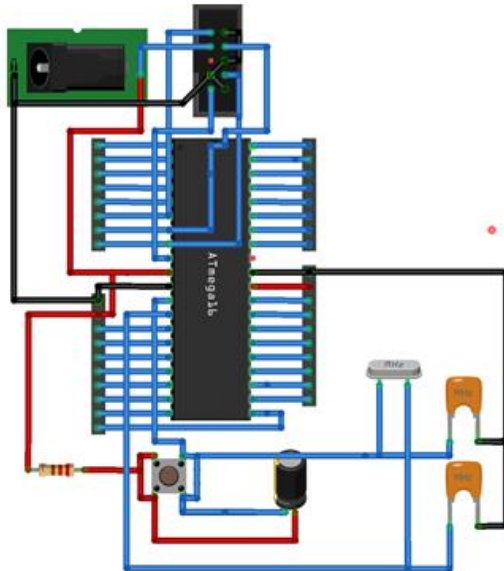
### 3.3 Perancangan *Hardware*

Pada perancangan *Hardware* dilakukan dengan merancang rangkaian elektronika dan rancangan mekanik. Perancangan *Hardware* meliputi:

1. Rangkaian *Minimum System* ATmega16
2. Rangkaian Modul *Relay*.
3. Rangkaian *Keypad*.

#### 3.3.1 Rangkaian *Minimum System* ATmega16

Rangkaian *Minimum System* ATmega16 ini digunakan sebagai pengendali dari seluruh kegiatan yang dilakukan oleh alat. Pada Mikrokontroler ATmega 16 memiliki 40 jumlah pin dengan 32 pin diantaranya terdiri dari 4 PORT yaitu PORT A yang memiliki 8 pin yaitu PORT PA0 sampai PA7, PORT B memiliki 8 pin yaitu PORT PB0 sampai PB7, PORT C memiliki 8 pin yaitu PORT PC0 sampai PC7, PORT D mempunyai 8 pin yaitu PORT PD0 sampai PD7. Rangkaian mikrokontroler ATmega16 dapat dilihat pada Gambar 3.2

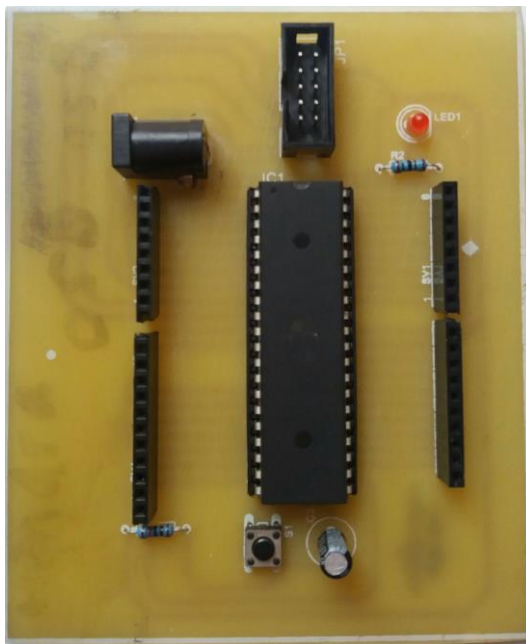


**Gambar 3. 2** Rangkaian Mikrokontroler ATmega 16

Fungsi utama dari rangkaian mikrokontroler ATmega 16 pada Gambar 3.2 adalah sebagai pengendali utama yang digunakan untuk mengendalikan *solenoid door lock* yang digerakkan oleh *driver relay*, mengoperasikan *keypad* dan menerima masukan dari *Limit Switch*.

Tegangan *supply* VCC pada rangkaian mikrokontroler ATmega 16 adalah 5V DC. Untuk mengisi program dengan menggunakan ICSP dengan menggunakan 4 sinyal yaitu RESET, SCK, MISO dan MOSI. Kristal resonator yang digunakan pada rangkaian di atas menggunakan 16 MHz.

Pada sistem alat Tugas Akhir yaitu rancang bangun sistem pengaman loker susun berbasis mikrokontroler, pin – pin yang dapat digunakan sebagai pin *input* dan *output* sejumlah 32 pin. Foto Rangkaian mikrokontroler ATmega16 dapat dilihat pada Gambar 3.3



**Gambar 3. 3** Foto Rangkaian Mikrokontroler ATmega 16

Untuk pengkabelan setiap rangkaian ke pin mikrokontroler ATmega 16 adalah sebagai berikut.

#### 1. Pengendali Modul Relay

*Modul Relay* membutuhkan pemicu berupa sinyal digital dari output pin mikrokontroler ATmega 16, pada alat yang dibuat pin pemicu pada *driver relay* dihubungkan ke pin C0, C1, C2, C3.

#### 2. Input Keypad

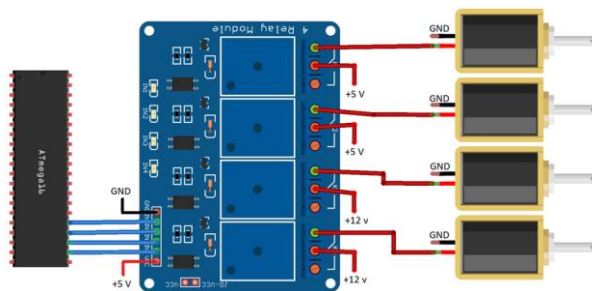
*Keypad* yang digunakan pada alat ini adalah *keypad 4x4*. *Keypad 4x4* membutuhkan sebanyak 8 pin digital dimana pin yang digunakan adalah pin PA0, PA1, PA2, PA3, PA4, PA5, PA6 dan PA7.

#### 3. Penerima Data Digital Limit Switch

*Limit Switch* menghasilkan *output* yang berupa sinyal digital, dimana data yang didapat tersebut dapat digunakan sebagai input pada mikrokontroler ATmega 16. Pada Alat ini pin *output* dari limit switch terhubung pada pin PC4, PC5, PD6, PD7.

### 3.3.2 Rangkaian Modul Relay

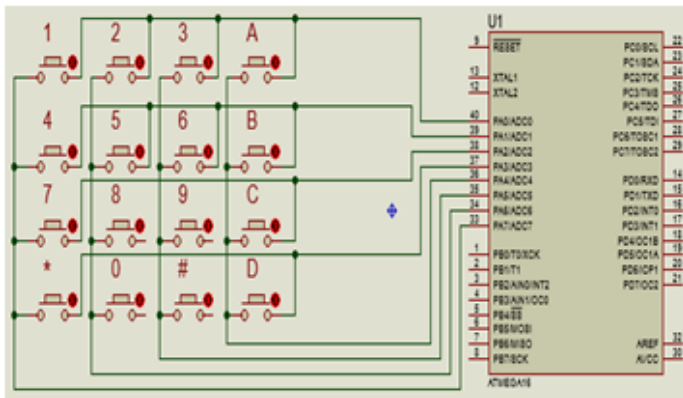
Rangkaian Modul *relay* pada alat ini berfungsi menggerakkan kontak saklar sehingga dengan arus listrik yang kecil (*low power*) dapat menghantarkan listrik yang bertegangan lebih tinggi untuk mengaktifkan *solenoid door lock* yang digunakan. *Driver Relay 4 channel* mempunyai 6 buah pin yang terdiri dari 1 buah pin VCC, 1 buah pin GND dan 4 buah pin digital. 4 pin digital tersebut terhubung dengan mikrokontroler ATmega 16 pada port PC0,PC1,PC2 dan PC3. Rangkaian modul relay 4 channel dengan ATmega16 dapat dilihat pada Gambar 3.4



**Gambar 3.4** Rangkaian Modul Relay 4 Channel dengan ATmega16

### 3.3.3 Rangkaian Keypad

Pada perancangan alat Tugas Akhir ini, *Keypad* berfungsi untuk memasukkan *password* yang berjumlah 6 buah kombinasi angka. *Keypad* yang digunakan pada perancangan alat Tugas Akhir ini adalah *Keypad* 4x4. *Keypad* 4x4 mempunyai sebanyak 8 buah pin yang akan terhubung pada pin digital dari mikrokontroler ATmega 16. Pin - pin tersebut akan terhubung pada pin digital PA0, PA1, PA2, PA3, PA4, PA5, PA6 dan PA7 mikrokontroler ATmega 16. *Keypad* yang terhubung pada mikrokontroler ATmega 16 dapat dilihat pada Gambar 3.5



Gambar 3. 5 Rangkaian Keypad pada pin ATmega 16

### 3.4 Perancangan Software

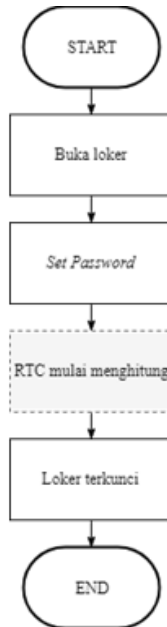
Perancangan *Software* dilakukan setelah seluruh perancangan elektronik telah dilakukan. Perancangan *Software* dilakukan agar alat pada Tugas Akhir ini dapat mencapai tujuan.

Pada Tugas Akhir ini ATmega16 memiliki peranan penting. *Minimum system* yang telah dirancang dengan ATmega16 sebagai mikrokontrolernya berfungsi sebagai pengendali utama alat. Sehingga seluruh perintah akan diolah oleh *minimum System* dengan ATmega16 sebagai mikrokontrolernya. Ditekan dan tidak ditekannya *Limit Switch* berfungsi untuk melakukan *set password* pada loker yang akan diolah oleh *minimum system*. Begitu pula dengan *Keypad* yang berfungsi sebagai memasukkan kombinasi angka untuk *password* pada loker dan juga *Solenoid Door Lock* yang berfungsi sebagai mengunci loker, semua perintah tersebut dilakukan dengan



memasukkan kedalam sebuah program sehingga dapat digunakan pada *minimum system*.

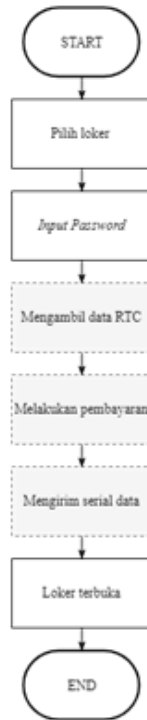
Pada alat Tugas Akhir ini *software* yang digunakan untuk pemrograman mikrokontroler ATmega16 adalah CodeVision AVR. Program yang telah dibuat tersebut akan di *upload* kedalam mikrokontroler ATmega16 dengan menggunakan *downloader yang telah dibuat*. Untuk mempermudah pembuatan program maka dibuatlah flowchart. Bagian *flowchart* yang dikerjakan adalah yang berwarna putih sedangkan untuk yang berwarna abu – abu dikerjakan oleh rekan kelompok. Beberapa tahapan yang dilakukan untuk menjalankan Rancang Bangun Sistem Pengaman Loker Berbasis Mikrokontroler dapat dilihat pada Gambar 3.6 dan 3.7



**Gambar 3. 6** Flowchart Proses Penitipan Barang

*Flowchart* yang telah dibuat diatas menunjukkan bagaimana proses yang telah terjadi pada saat penyewa loker akan melakukan penyimpanan atau penitipan barang bawaannya pada loker yang telah tersedia. Penyewa loker melakukan buka pintu loker terlebih

dahulu. Setelah itu akan keluar perintah *Set Password* pada LCD. *Password* yang dapat dimasukkan berupa kombinasi angka. Kombinasi angka - angka tersebut maksimal sebanyak 4 digit. Setelah melakukan *Set Password* pada loker maka loker akan terkunci. Tuas *Solenoid Door Lock* akan memanjang sehingga loker akan terkunci dan tidak dapat dibuka.



**Gambar 3. 7** *Flowchart* Proses Pengambilan Barang

*Flowchart* yang telah dibuat diatas menunjukkan bagaimana proses yang telah terjadi pada saat penyewa loker akan melakukan pengambilan barang bawaannya pada loker yang telah digunakan. Penyewa loker akan memilih loker terlebih dahulu sesuai dengan loker yang digunakan. Kemudian penyewa akan *input password* sesuai dengan *password* yang telah di *set* terlebih dahulu. Setelah *password* benar maka penyewa akan melakukan pembayaran. Pada

tahap pembayaran dikerjakan oleh rekan kelompok. Setelah melakukan pembayaran maka loker akan terbuka.

### 3.4.1 *Flowchart Set Password*

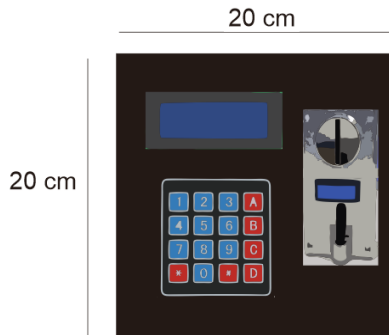
Saat penyewa loker akan menitipkan barang bawaannya. Maka penyewa loker tersebut akan melakukan set password. Set password bertujuan untuk membuat kombinasi angka rahasia dari penyewa loker agar tidak diketahui oleh orang lain. Sehingga barang yang ditaruh pada loker tersebut menjadi aman. *Flowchart* dari *set password* dapat dilihat pada Gambar 3.8



**Gambar 3. 8** *Flowchart Set Password*

### 3.5 Desain Kubus Kontroler

Pada Tugas Akhir ini desain alat berupa tempat kontroler berbentuk kubus dengan ukuran 20cm x 20cm x 20cm. Tempat kontroler ini terbuat dari akrilik. Ketebalan dari tempat ini berukuran 2mm. Pada bagian depan telah disiapkan lubang untuk memasang *keypad* yang berfungsi sebagai *input password*, *coin acceptor* yang berfungsi sebagai pembayaran sewa loker dan LCD 20 x 4 yang berfungsi sebagai tampilan kerja loker agar dapat digunakan oleh pengguna. Pada bagian belakang terdapat lubang sebagai saluran untuk kabel *Power Supply*, kabel untuk *Limit Switch* dan juga kabel untuk *Solenoid Door Lock*. Rancangan desain dari kubus kontroler dapat dilihat pada Gambar 3.9 dan kubus kontroler pada Gambar 3.10, Gambar 3.11 dan Gambar 3.12.



Gambar 3. 9 Desain Kubus Kontroler



Gambar 3. 10 Kubus Kontroler Tampak Depan



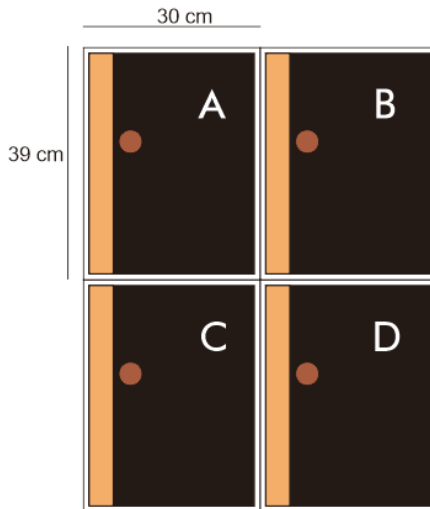
**Gambar 3. 11** Kubus Kontroler Tampak Samping



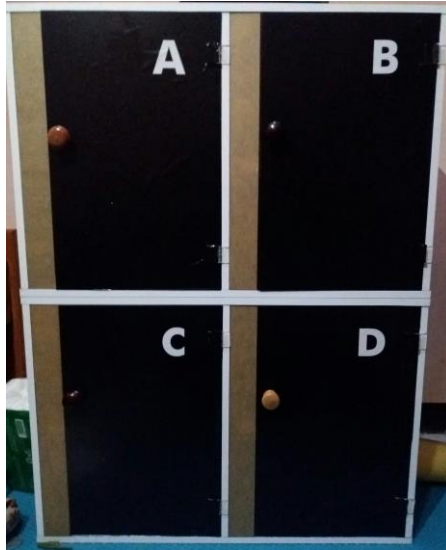
**Gambar 3. 12** Kubus Kontroler Tampak Atas

### 3.6 Desain Loker

Pada Tugas Akhir ini desain alat berupa loker sebanyak 4 buah loker berbentuk persegi yang memiliki ukuran 30cm x 30cm x 39cm pada setiap lokernya. Ketebalan dari desain loker ini sebesar 1cm. Pada loker ini berfungsi sebagai penyimpanan barang yang didalamnya terdapat komponen pendukung seperti *Limit Switch* dan juga *Solenoid Door Lock*. Pintu dari loker ini terbuat dari akrilik dengan ketebalan 3mm dengan ukuran 27cm x 37cm. Loker didesain sesedemikian rupa agar dapat digunakan dengan baik. Bentuk Loker dapat dilihat pada Gambar 3.13 – 3.17



**Gambar 3. 13** Desain Loker



**Gambar 3. 14** Loker Tampak Depan



**Gambar 3. 15** Loker Tampak Atas



**Gambar 3. 16** Loker Tampak Samping



**Gambar 3. 17** Rancangan Keseluruhan Alat



## BAB IV

### PENGUKURAN DAN PENGUJIAN

Untuk mengetahui apakah tujuan-tujuan dari pembuatan alat ini telah terlaksana atau tidak, perlu dilakukan pengujian alat yang meliputi pengujian perangkat keras dan pengujian perangkat lunak. Pengujian ini dilakukan pada peralatan untuk mengetahui kesesuaian antara teori dengan hasil perancangan, yaitu dengan mengetahui hasil pengukuran pada setiap perangkat yang telah dibuat.

#### 4.1 Pengukuran Rangkaian *Power Supply*

Pengujian *power supply* 5 dan 12 V DC bertujuan untuk mengetahui apakah tegangan yang keluar telah sesuai. Tujuan dalam pengujian tersebut untuk dapat mengetahui tegangan yang keluar. Secara lengkap dapat dilihat pada Tabel 4.1 dibawah ini.

**Tabel 4. 1** Pengukuran Tegangan pada *Power Supply*

No	Power supply 12V DC (Tanpa beban)	Modul Regu- lator 5V DC (Tanpa beban)	Power supply 12V DC	Modul Regulator 5V DC
1	12.05	5.35	12.04	5.35
2	12.05	5.36	12.04	5.36
3	12.06	5.35	12.04	5.36
4	12.05	5.35	12.04	5.35
5	12.05	5.35	12.04	5.35

Dari tabel diatas dapat dilihat bahwa tegangan *Output* dari *Power Supply* 12V DC pada saat dengan dan tanpa beban memiliki tegangan sebesar 12.04 – 12.06 V, sehingga tegangan *Output* dari *Power Supply* 12V DC dapat digunakan sebagai sumber tegangan untuk *Solenoid Door Lock dan coin acceptor* yang membutuhkan sumber tegangan sebesar 12 V. Sedangkan tegangan *Output* dari regulator 5V DC pada saat dengan dan tanpa beban adalah sebesar  $\pm 5.35 - 5.36$  sehingga tegangan *Output* dari Regulator 5V DC dapat

digunakan sebagai sumber tegangan untuk *Driver Relay* yang membutuhkan sumber tegangan sebesar  $\pm 5$  V. Selanjutnya, untuk mengetahui apakah power supply dapat digunakan atau tidak, dengan cara menghitung presentase error dengan rumus pada Persamaan 4.1 dan menghasilkan data pada Tabel 4.2 :

$$\% \text{ Error} = \left| \frac{(\text{NilaiAcuan} - \text{NilaiPengukuran})}{\text{NilaiAcuan}} \right| \times 100\% \dots\dots\dots(4.1)$$

**Tabel 4. 2** Data Pengujian *Power Supply*

<i>Power Supply</i>	<i>Output (V)</i>	<i>Error(%)</i>
+12 V	12,05	0,41

Pengujian dilakukan dengan mengukur keluaran dengan volt-meter, kemudian dihitung presentasi *error*.

$$\% \text{ Error} = \left| \frac{(12 - 12,05)}{12} \right| \times 100\% = 0,41 \%$$

#### 4.2 Pengukuran *Limit Switch*

Pengukuran tegangan pada *Limit Switch* dilakukan agar dapat mengetahui apakah tegangan sesuai. Tujuan dari pengujian tersebut bertujuan untuk dapat mengetahui tegangan yang keluar. Hasil pengujian dari *Limit Switch* dapat dilihat pada Tabel 4.3 – 4.6 dibawah ini.

**Tabel 4. 3** Pengukuran Tegangan pada *Limit Switch* Loker A

No	<i>Limit Switch (V)</i>	
	<i>Open</i>	<i>Close</i>
1	4.77	0
2	4.77	0
3	4.78	0

No	<i>Open</i>	<i>Close</i>
4	4.77	0
5	4.77	0

**Tabel 4. 4** Pengukuran Tegangan pada *Limit Switch* Loker B

No	<i>Limit Switch (V)</i>	
	<i>Open</i>	<i>Close</i>
1	4.77	0
2	4.77	0
3	4.77	0
4	4.77	0
5	4.78	0

**Tabel 4. 5** Pengukuran Tegangan pada *Limit Switch* Loker C

No	<i>Limit Switch (V)</i>	
	<i>Open</i>	<i>Close</i>
1	4.78	0
2	4.77	0
3	4.77	0
4	4.77	0
5	4.77	0

**Tabel 4. 6** Pengukuran Tegangan pada *Limit Switch* Loker D

No	<i>Limit Switch (V)</i>	
	<i>Open</i>	<i>Close</i>
1	4.77	0
2	4.77	0
3	4.78	0
4	4.77	0
5	4.77	0

Dari Tabel diatas dapat dilihat tegangan yang keluar pada *Limit Switch* pada saat *Open* dan *Close*

### 4.3 Pengukuran *Solenoid Door Lock*

Pengukuran pada *Solenoid Door Lock* dilakukan untuk mengetahui berapa arus yang dibutuhkan saat bekerja. Pengukuran dilakukan pada *Solenoid Door Lock* 5 V dan juga pada *Solenoid Door Lock* 12 V. Pengukuran dilakukan dengan menggunakan multimeter yang telah tersedia. Hasil pengukuran dari *Solenoid Door Lock* 5 V dan juga pada *Solenoid Door Lock* 12 V dapat dilihat pada Tabel 4.7 dan 4.8 dibawah ini

**Tabel 4. 7** Arus *Solenoid Door Lock* 5 V

No	<i>Solenoid Door Lock</i> (A)	
	<i>High</i>	<i>Low</i>
1	0.55	0.00
2	0.54	0.00
3	0.55	0.00
4	0.55	0.00
5	0.55	0.00
6	0.54	0.00
7	0.55	0.00
8	0.55	0.00
9	0.55	0.00

**Tabel 4. 8** Arus *Solenoid Door Lock* 12 V

No	<i>Solenoid Door Lock</i> (A)	
	<i>High</i>	<i>Low</i>
1	0.40	0.00
2	0.40	0.00
3	0.39	0.01
4	0.40	0.00
5	0.39	0.00
6	0.40	0.00
7	0.40	0.00
8	0.40	0.00

No	High	Low
9	0.40	0.00

#### 4.4 Pengujian *Solenoid Door Lock* Terhadap Catu Daya

Pengujian yang dilakukan pada *solenoid door lock* terhadap catu daya bertujuan untuk mengetahui berapa besar tegangan yang dibutuhkan oleh *Solenoid Door Lock* ketika bekerja. Pengujian dilakukan dengan menggunakan alat ukur *multimeter*. Pengujian dilakukan dengan cara memberikan sumber tegangan pada *Solenoid Door Lock*. *Solenoid Door Lock* yang digunakan pada alat ini sebanyak 4. Dimana 2 *Solenoid Door Lock* 5 V dan 2 *Solenoid Door Lock* 12 V . Pengujian dari *Solenoid Door Lock* dapat dilihat pada Tabel 4.9 dan 4.10 berikut:

**Tabel 4. 9** Pengujian *Solenoid Door Lock* 5 V Terhadap Catu Daya

No	Tegangan (V)	Kondisi <i>Solenoid Door Lock</i>
1	1.00	Tidak Aktif
2	2.00	Tidak Aktif
3	2.50	Tidak Aktif
4	3.00	Tidak Aktif
5	3.50	Tidak Aktif
6	4.00	Tidak Aktif
7	4.50	Tidak Aktif
8	5.35	Aktif
9	5.50	Aktif

**Tabel 4. 10** Pengujian *Solenoid Door Lock* 12 V Terhadap Catu Daya

No	Tegangan (V)	Kondisi <i>Solenoid Door Lock</i>
1	5	Tidak Aktif
2	6	Tidak Aktif

No	Tegangan (V)	Kondisi <i>Solenoid Door Lock</i>
3	7	Tidak Aktif
4	7.5	Aktif
5	8	Aktif
6	9	Aktif
7	10	Aktif
8	11	Aktif
9	12	Aktif

#### 4.5 Pengujian *Password Keypad*

Fungsi dari loker adalah menyimpan suatu benda yang ditiptkan sementara ataupun dalam jangka waktu yang panjang. Tujuan dari pengujian ini adalah untuk mengetahui apakah *password* yang nantinya dimasukkan oleh pengguna melalui *keypad* sesuai dengan yang ditampilkan pada LCD dan tersimpan pada EEPROM. Pengujian ini dilakukan dengan cara melihat keluaran *keypad* yang dimasukkan oleh pengguna sehingga muncul pada LCD.

##### 4.5.1 Pengujian *Keypad*

Pengujian *keypad* dilakukan dengan cara menghubungkan *keypad* dengan LCD dan memasukkan program *keypad* sehingga pada LCD muncul keluaran dari *keypad* sesuai dengan yang dimasukkan oleh pengguna. Tujuan dari pengujian ini yaitu untuk mengetahui ketika tombol yang ditekan pada *keypad* sudah sesuai dengan yang ditampilkan pada LCD. Data dari pengujian *keypad* dapat dilihat pada Tabel 4.11 dibawah ini.

**Tabel 4. 11** Pengujian *Keypad*

No	<i>Keypad</i>	LCD	Keterangan
1	1234	1234	Sama
2	2345	2345	Sama
3	3456	3456	Sama
4	4567	4567	Sama

Berdasarkan Tabel 4.11 pengujian *Keypad* dilakukan sebanyak 4 kali untuk mengetahui keluaran *keypad* dengan keluaran yang ada pada LCD adalah sama dan tidak terdapat *error*.

#### 4.5.2 Pengujian *Password*

Pengujian *Password* dilakukan dengan cara menghubungkan Program *password* dengan *keypad* dan juga LCD (*liquid crystal display*). Pengujian *Password* bertujuan untuk melihat bagaimana kerja dari program *password* yang akan berfungsi untuk menyimpan masukan *password* yang di *input* oleh pengguna melalui *keypad*. Data pengujian *password* dapat dilihat pada Tabel 4.12 dibawah ini.

**Tabel 4. 12** Pengujian *Password*

No	<i>Password</i> masuk	<i>Pasword</i> keluar	Keterangan
1	1234	1234	Sama
2	8267	8267	Sama
3	3139	3139	Sama
4	5327	5327	Sama
5	1049	1049	Sama

#### 4.6 Pengujian Sensor *Coin Acceptor CH-923*

Pengujian sensor *Coin Acceptor CH-923* bertujuan untuk mengetahui apakah sensor ini dapat mendeteksi *Coin* yang digunakan sesuai dengan yang telah diatur atau tidak, dimana pada alat ini yang digunakan adalah uang logam Rp 500 perak dan juga Rp 500 kuning. Oleh karena itu maka dilakukan pengujian terhadap jenis – jenis dari uang logam yang ada. Pengujian dilakukan dengan cara memasukkan *Coin* yang berupa uang logam ke *Coin Acceptor CH-923*. Apabila jenis *Coin* yang dimasukkan sesuai, maka kebutuhan *Coin* akan ditampilkan pada *Display* yang terletak pada sisi kiri *Coin Acceptor*. Hasil dari pengujian terhadap *Coin* dapat dilihat pada Tabel 4.13 – 4.16 :

**Tabel 4. 13** Pengujian *Coin Acceptor* Dengan *Coin* Rp 500 kuning

Percobaan	Tampilan pada <i>Coin Acceptor</i>	Keterangan
1	1	<i>Coin</i> Terbaca
2	1	<i>Coin</i> Terbaca
3	1	<i>Coin</i> Terbaca
4	1	<i>Coin</i> Terbaca
5	1	<i>Coin</i> Terbaca
6	1	<i>Coin</i> Terbaca
7	1	<i>Coin</i> Terbaca
8	1	<i>Coin</i> Terbaca
9	1	<i>Coin</i> Terbaca
10	1	<i>Coin</i> Terbaca

**Tabel 4. 14** Pengujian *Coin Acceptor* Dengan *Coin* Rp 500 perak

Percobaan	Tampilan pada <i>Coin Acceptor</i>	Keterangan
1	2	<i>Coin</i> Terbaca
2	2	<i>Coin</i> Terbaca
3	2	<i>Coin</i> Terbaca
4	2	<i>Coin</i> Terbaca
5	2	<i>Coin</i> Terbaca
6	2	<i>Coin</i> Terbaca
7	2	<i>Coin</i> Terbaca
8	2	<i>Coin</i> Terbaca
9	2	<i>Coin</i> Terbaca
10	2	<i>Coin</i> Terbaca

**Tabel 4. 15** Pengujian *Coin Acceptor* Dengan *Coin* Rp 200 perak

Percobaan	Tampilan pada <i>Coin Acceptor</i>	Keterangan
1	-	<i>Coin</i> Tidak Terbaca
2	-	<i>Coin</i> Tidak Terbaca
3	-	<i>Coin</i> Tidak Terbaca
4	-	<i>Coin</i> Tidak Terbaca



Percobaan	Tampilan pada Coin Acceptor	Keterangan
5	-	Coin Tidak Terbaca
6	-	Coin Tidak Terbaca
7	-	Coin Tidak Terbaca
8	-	Coin Tidak Terbaca
9	-	Coin Tidak Terbaca
10	-	Coin Tidak Terbaca

**Tabel 4. 16** Pengujian *Coin Acceptor* Dengan *Coin* Rp 1000 perak

Percobaan	Tampilan pada <i>Coin Acceptor</i>	Keterangan
1	-	<i>Coin</i> Tidak Terbaca
2	-	<i>Coin</i> Tidak Terbaca
3	-	<i>Coin</i> Tidak Terbaca
4	-	<i>Coin</i> Tidak Terbaca
5	-	<i>Coin</i> Tidak Terbaca
6	-	<i>Coin</i> Tidak Terbaca
7	-	<i>Coin</i> Tidak Terbaca
8	-	<i>Coin</i> Tidak Terbaca
9	-	<i>Coin</i> Tidak Terbaca
10	-	<i>Coin</i> Tidak Terbaca

Hasil pengujian terhadap dua jenis *Coin* berupa uang logam Rp 500 kuning dan perak dapat dilihat pada tabel – tabel diatas bahwa uang logam Rp 500 kuning menghasilkan tampilan 1 pada yang menandakan mengirim pulsa sebesar 1 gelombang dan uang logam Rp 500 perak menghasilkan tampilan dapat diterima oleh *Coin Acceptor* hal tersebut dikarenakan hanya uang logam Rp 500 kuning dan perak yang sudah dilakukan proses *setting* dan *sampling* pada *Coin Acceptor* CH-923 yang digunakan pada alat.

#### **4.7 Petunjuk Penggunaan Alat**

Petunjuk penggunaan ditunjukkan untuk pengguna agar dapat mudah memahami cara penggunaan alat pada Tugas Akhir ini. Berikut adalah langkah langkah dalam penggunaan alat pada Tugas Akhir ini pada Gambar 4.1 – 4.16

a) Penggunaan pada awal penggunaan loker

- Baca Petunjuk Penggunaan



**Gambar 4. 1** Petunjuk Penggunaan

- Baca kondisi loker pada LCD



**Gambar 4. 2** Kondisi Loker

- Buka pintu loker



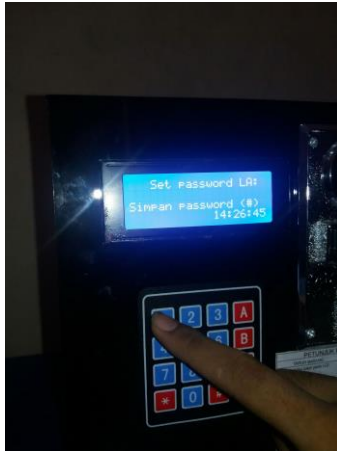
**Gambar 4. 3** Buka Loker

- Taruh barang pada loker



**Gambar 4. 4** Taruh Barang

- Set password dengan menggunakan keypad



**Gambar 4. 5** *Set Password*

- Password yang dimasukkan maks 4 digit  
Password terdiri dari kombinasi angka



**Gambar 4. 6** *Masukkan Password*

- Tekan (\*) untuk hapus angka



**Gambar 4. 7** Hapus *Password*

- Tekan (#) untuk simpan password



**Gambar 4. 8** Simpan *Password*

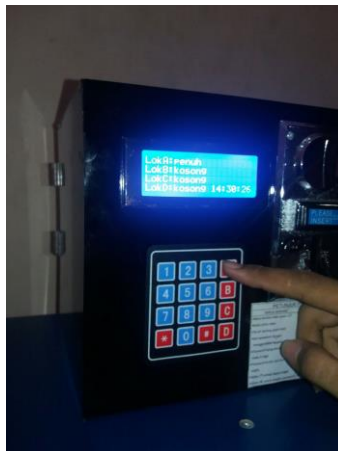
- b) Penggunaan pada saat mengambil barang

- Baca Petunjuk Penggunaan



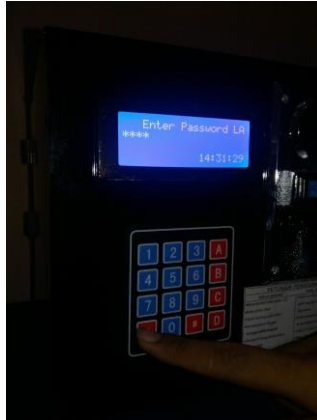
**Gambar 4. 9** Petunjuk Penggunaan

- Tekan tombol A/B/C/D pada pada keypad sesuai loker yang digunakan



**Gambar 4. 10** Pilih Loker

- Enter password yang telah di *set* sebelumnya



**Gambar 4. 11** *Enter Password*

- Tekan (\*) untuk hapus angka



**Gambar 4. 12** *Hapus Password*

- Tekan (#) untuk OK



**Gambar 4. 13** Tekan #

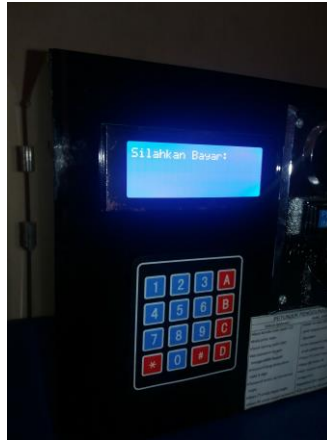
- Akan muncul notif tarif penggunaan



**Gambar 4. 14** Notif Tarif Pembayaran



- Masukkan Coin sesuai nominal



**Gambar 4. 15** Masukkan *Coin*

- Coin senilai Rp 500 perak/kuning



**Gambar 4. 16** Jenis *Coin*

#### 4.8 Pengujian Keseluruhan Sistem Alat

Pengujian keseluruhan kinerja alat bertujuan untuk menguji tingkat keberhasilan alat ketika dioperasikan secara berulang – ulang. Bagian fitur yang diuji antara lain adalah sebagai berikut:

##### 4.8.1 Fitur *Limit Switch*

Pada pengujian fitur *Limit Switch* dapat dikatakan berhasil apabila ketika membuka loker saat akan memasukan barang, maka pada LCD akan muncul perintah untuk memasukan *password*. Untuk mengetahui hal tersebut, maka dilakukan suatu pengujian. Pengujian dilakukan dengan cara membuka loker saat hendak memasukan barang. Apabila saat pintu loker dibuka pada LCD tampil tulisan “Set password LA”, “Set password LB”, “Set password LC” ataupun “Set password LD” maka pengujian dapat dikatakan telah sesuai dengan kerja dari fitur *Limit Switch* tersebut. Sebagai contohnya adalah pada Gambar 4.17



**Gambar 4. 17** Tampilan *Set Password*

##### 4.8.2 Fitur *Password*

Pengujian terhadap fitur *Password* dilakukan dengan cara pencocokan antara *Password* awal penggunaan dengan *Password* akhir. Jika telah sesuai maka dapat dikatakan berhasil. Pada prosenya dilakukan dengan cara *Set password* awal terlebih dahulu kemudian pada tahap akhir dilakukan *Enter password* akhir. Apabila keadaan *password* akhir sama dengan *password* awal maka pada LCD (*Liquid Crystal Display*) akan tampil tulisan “password benar”

dan jika tidak sesuai maka tampilan pada LCD (*Liquid Crystal Display*) adalah “password salah”.

Beberapa pengujian yang dilakukan pada fitur *password* dapat dilihat pada gambar – gambar dibawah ini. Pada tampilan awal terdapat tampilan untuk *enter password* seperti pada Gambar 4.18 Kemudian di masukkan *password* dengan *password* yang tidak sesuai maka pada LCD akan muncul tampilan “password salah” seperti pada Gambar 4.19 Jika *password* yang dimasukkan sesuai maka tampilan pada LCD akan muncul “password benar” seperti pada Gambar 4.20



**Gambar 4. 18** Tampilan Perintah *Enter Password*



**Gambar 4. 19** Tampilan *Password* Salah



**Gambar 4. 20** Tampilan *Password* Benar

#### **4.8.3** Fitur *Solenoid Door lock*

Pengujian terhadap fitur *Solenoid Door Lock* dilakukan dengan cara memasukkan *password* dengan menggunakan *Keypad* yang kemudian *password* tersimpan dan tampilan pada LCD menjadi “LokA : penuh”, “LokB : penuh”, “LokC : penuh” atau “LokD : penuh”. Pada pengujian fitur *Solenoid Door Lock* dapat dikatakan berhasil apabila saat tampilan LCD tampil “LokA : penuh”, “LokB : penuh”, “LokC : penuh” atau “LokD : penuh” kondisi tuas *Solenoid Door Lock* akan mengunci loker tersebut. Sedangkan ketika “LokA : kosong”, “LokB : kosong”, “LokC : kosong” atau “LokD : kosong” maka kondisi tuas *Solenoid Door Lock* membuka atau menarik. Tampilan LCD dapat dilihat pada Gambar 4.21 – 4.24



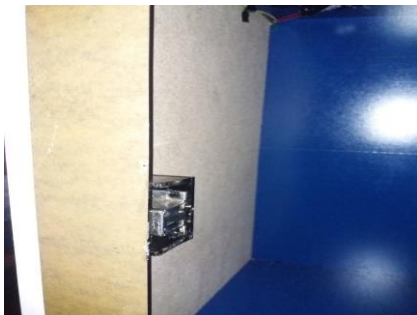
**Gambar 4. 21** Tampilan Loker Penuh



**Gambar 4. 22** *Solenoid Door Lock* Mengunci



**Gambar 4. 23** Tampilan Loker Kosong



**Gambar 4. 24** *Solenoid Door Lock* Terbuka

Untuk mendapatkan presentase keberhasilan kinerja alat maka dilakukan pengujian terhadap ketiga fitur diatas secara berulang – ulang, yang dimana apabila disetiap fitur terjadi kesalahan maka akan dianggap sebagai nilai *error* dan apabila pada setiap fitur tidak terjadi kesalahan maka dapat dianggap berhasil. Hasil pengujian dari kinerja alat keseluruhan dapat dilihat pada tabel 4.17 – 4.20

**Tabel 4. 17** Hasil Pengujian Kinerja Alat Keseluruhan pada Loker A

No	Pengujian Fitur			Keterangan
	<i>Limit Switch</i>	<i>Password</i>	<i>Solenoid Door Lock</i>	
1	Berhasil	Berhasil	Berhasil	-
2	Berhasil	Berhasil	Berhasil	-
3	Berhasil	Berhasil	Berhasil	-
4	Berhasil	Berhasil	Berhasil	-
5	Berhasil	Berhasil	Berhasil	-
6	Berhasil	Berhasil	<i>Error</i>	<i>Insert Coin Error</i>
7	Berhasil	Berhasil	Berhasil	-
8	Berhasil	Berhasil	Berhasil	-
9	Berhasil	Berhasil	Berhasil	-
10	Berhasil	Berhasil	Berhasil	-
11	Berhasil	Berhasil	Berhasil	-
12	Berhasil	Berhasil	Berhasil	-
13	Berhasil	Berhasil	Berhasil	-
14	Berhasil	Berhasil	Berhasil	-
15	<i>Error</i>	<i>Error</i>	<i>Error</i>	<i>Limit Switch Error</i>
16	Berhasil	Berhasil	Berhasil	-
17	Berhasil	Berhasil	Berhasil	-
18	Berhasil	Berhasil	Berhasil	-
19	Berhasil	Berhasil	Berhasil	-
20	Berhasil	Berhasil	Berhasil	-

**Tabel 4. 18** Hasil Pengujian Kinerja Alat Keseluruhan pada Loker B

No	Pengujian Fitur			Keterangan
	<i>Limit Switch</i>	<i>Password</i>	<i>Solenoid Door Lock</i>	
1	Berhasil	Berhasil	Berhasil	-
2	Berhasil	Berhasil	Berhasil	-
3	Berhasil	Berhasil	Berhasil	-
4	Berhasil	Berhasil	Berhasil	-
5	Berhasil	Berhasil	Berhasil	-
6	Berhasil	Berhasil	Berhasil	-
7	Berhasil	Berhasil	Berhasil	-
8	<i>Error</i>	<i>Error</i>	<i>Error</i>	<i>Limit Switch Error</i>
9	Berhasil	Berhasil	Berhasil	-
10	Berhasil	Berhasil	Berhasil	-
11	Berhasil	Berhasil	Berhasil	-
12	Berhasil	Berhasil	Berhasil	-
13	Berhasil	Berhasil	Berhasil	-
14	Berhasil	Berhasil	Berhasil	-
15	Berhasil	Berhasil	Berhasil	-
16	Berhasil	Berhasil	Berhasil	-
17	Berhasil	Berhasil	Berhasil	-
18	Berhasil	Berhasil	Berhasil	-
19	Berhasil	Berhasil	Berhasil	-
20	Berhasil	Berhasil	Berhasil	-

**Tabel 4. 19** Hasil Pengujian Kinerja Alat Keseluruhan pada Loker C

No	Pengujian Fitur			Keterangan
	<i>Limit Switch</i>	<i>Password</i>	<i>Solenoid Door Lock</i>	
1	Berhasil	Berhasil	Berhasil	-
2	Berhasil	Berhasil	Berhasil	-
3	Berhasil	Berhasil	Berhasil	-
4	<i>Error</i>	<i>Error</i>	<i>Error</i>	<i>Limit Switch Error</i>

No	<i>Limit Switch</i>	<i>Password</i>	<i>Solenoid Door Lock</i>	Keterangan
5	Berhasil	Berhasil	Berhasil	-
6	Berhasil	Berhasil	Berhasil	-
7	Berhasil	Berhasil	Berhasil	-
8	Berhasil	Berhasil	Berhasil	-
9	Berhasil	Berhasil	Berhasil	-
10	Berhasil	Berhasil	Berhasil	-
11	Berhasil	Berhasil	<i>Error</i>	<i>Insert Coin Error</i>
12	Berhasil	Berhasil	Berhasil	-
13	Berhasil	Berhasil	Berhasil	-
14	Berhasil	Berhasil	Berhasil	-
15	Berhasil	Berhasil	Berhasil	-
16	Berhasil	Berhasil	Berhasil	-
17	Berhasil	Berhasil	Berhasil	-
18	Berhasil	Berhasil	Berhasil	-
19	Berhasil	Berhasil	Berhasil	-
20	Berhasil	Berhasil	Berhasil	-

**Tabel 4. 20** Hasil Pengujian Kinerja Alat Keseluruhan pada Loker D

No	Pengujian Fitur			Keterangan
	<i>Limit Switch</i>	<i>Password</i>	<i>Solenoid Door Lock</i>	
1	Berhasil	Berhasil	Berhasil	-
2	Berhasil	Berhasil	Berhasil	-
3	Berhasil	Berhasil	Berhasil	-
4	Berhasil	Berhasil	Berhasil	-
5	Berhasil	Berhasil	Berhasil	-
6	Berhasil	Berhasil	Berhasil	-
7	Berhasil	Berhasil	Berhasil	-
8	Berhasil	Berhasil	Berhasil	-
9	Berhasil	Berhasil	Berhasil	-
10	Berhasil	Berhasil	Berhasil	-
11	Berhasil	Berhasil	Berhasil	-
12	Berhasil	Berhasil	<i>Error</i>	<i>Inset Coin Error</i>



No	<i>Limit Switch</i>	<i>Password</i>	<i>Solenoid Door Lock</i>	Keterangan
13	Berhasil	Berhasil	Berhasil	-
14	Berhasil	Berhasil	Berhasil	-
15	Berhasil	Berhasil	Berhasil	-
16	Berhasil	Berhasil	Berhasil	-
17	<i>Error</i>	<i>Error</i>	<i>Error</i>	<i>Limit Switch Error</i>
18	Berhasil	Berhasil	Berhasil	-
19	Berhasil	Berhasil	Berhasil	-
20	Berhasil	Berhasil	Berhasil	-

Dari hasil pengujian alat secara keseluruhan pada setiap loker yaitu loker A, loker B, loker C dan juga loker D dilakukan pengujian sebanyak 20 kali pada masing – masing loker. Total *error* yang terjadi pada keseluruhan loker adalah sebanyak tujuh kali. *Error* yang terjadi disebabkan oleh *Limit Switch* dan juga *Insert Coin*. Dari hasil pengujian alat secara keseluruhan tersebut presentase keberhasilan  $(73/80) \times 100\% = 91.25\%$  dengan presentase *error* sebesar  $100\% - 91.25\% = 8.75\%$ .

-----Halaman ini sengaja dikosongkan-----

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Pada hasil uji ukur kali ini, dapat diambil kesimpulan antara lain sebagai berikut:

1. *Input* dari *Keypad* tampil dan dapat dilihat melalui LCD.
2. Tegangan pada *Limit Switch* pada saat keadaan *open* bernilai 4.77 – 4.78 V.
3. Tegangan pada *Limit Switch* pada saat keadaan *close* bernilai 0 V.
4. *Solenoid Door Lock 5 V* akan bekerja optimal ketika mendapatkan *supply* tegangan sebesar 5.35 V.
5. *Solenoid Door Lock 12 V* akan bekerja optimal ketika mendapatkan *supply* tegangan sebesar 12 V.
6. *Output* tegangan dari *power supply* 12 V tanpa beban sebesar 12.05 – 12.06 V.
7. *Output* tegangan *power supply* tanpa beban lebih besar 0.01 V dibandingkan dengan *power supply* dengan beban.
8. Arus yang dibutuhkan *Solenoid Door Lock 5 V* lebih besar 0.1 A dibandingkan dengan *Solenoid Door Lock 12 V*.
9. Pada pengujian secara keseluruhan dengan mencoba seluruh fitur yang telah tersedia didapatkan presentase nilai *error* sebesar 8.75% sedangkan untuk presentase nilai keberhasilan sebesar 91.25 %

#### **5.2 Saran**

Saran untuk penelitian selanjutnya yaitu diharapkan untuk menambahkan fitur *fingerpint*, dikarenakan ketika ditambahkan fitur *fingerpint* akan terhindar dari kemungkinan lupa *password*. Kemudian menambahkan proses validasi *password* kembali.

-----Halaman ini sengaja dikosongkan-----

## DAFTAR PUSTAKA

- Andrianto, Heri. 2013. “ Pemrograman Mikrokontroler AVR ATmega16 menggunakan Bahasa C (CodeVisionAVR) edisi revisi”. Bandung : Penerbit Informatika.
- Andrianto, Heri. 2015. “ Pemrograman Mikrokontroler AVR ATmega16 menggunakan Bahasa C (CodeVisionAVR)”. Bandung : Penerbit Informatika.
- Budiharto, Wididio. 2005. Perancangan Sistem dan Aplikasi Mikrokontroler. Jakarta : PT Elex Media Komputindo.
- Budiharto, Wididio. 2008. “Panduan Praktikum Mikrokontroler AVR ATmega16”. Jakarta : Elex Media Komputindo.
- Dhananjay, Gadre. 2001. ”*Programing and Customizing The AVR Microcontroller*”. New York: McGraw-Hill.
- Eko Putra, Agfianto . 2002 . “Belajar Mikrokontroller ATmega16”. Yogyakarta : Gaya Media.
- Lingga, Wardana. 2006. “Belajar Sendiri Mikrokontroler AVR Seri ATmega16, Simulasi *Hardware* dan Aplikasi”. Yogyakarta: Andi.
- Morton, John. 2007. “ *AVR an Introductory Course*”. England : Newnes
- Perdana, Latif . 2012. “Modul AVR *Trainer*”. Yogyakarta : Mercubuana.
- Raharjo. 2005. “ Keamanan Sistem Informasi Berbasis Internet “. Bandung : PT. Insan Indonesia dan PT. Indocisc.
- Setiawan, Iwan. 2006. “*Tutorial Microcontroller AVR Part I*”. Semarang: UNDIP.
- Team IE. 2006. “ Panduan Praktis Mikrokontroler Keluarga AVR”. Surabaya. : Innovative Electronic.

-----Halaman ini sengaja dikosongkan-----

## LAMPIRAN

```
#include <mega16.h>
#include <delay.h>
#include <stdlib.h>

#define ls1 PIND.6
#define ls2 PIND.7
#define ls3 PINC.5
#define ls4 PINC.4
#define door_lock1 PORTC.3
#define door_lock2 PORTC.2
#define door_lock3 PORTC.1
#define door_lock4 PORTC.0

// I2C Bus functions

// I2C Bus functions
#include <i2c.h>

// DS1307 Real Time Clock functions
#include <ds1307.h>

// Alphanumeric LCD functions
#include <alcd.h>

// Declare your global variables here

#define DATA_REGISTER_EMPTY (1<<UDRE)
#define RX_COMPLETE (1<<RXC)
#define FRAMING_ERROR (1<<FE)
#define PARITY_ERROR (1<<UPE)
#define DATA_OVERRUN (1<<DOR)

// USART Receiver buffer
#define RX_BUFFER_SIZE 8
char rx_buffer[RX_BUFFER_SIZE];

#if RX_BUFFER_SIZE <= 256
```

```

unsigned char rx_wr_index=0,rx_rd_index=0;
#else
unsigned int rx_wr_index=0,rx_rd_index=0;
#endif

#if RX_BUFFER_SIZE < 256
unsigned char rx_counter=0;
#else
unsigned int rx_counter=0;
#endif

// This flag is set on USART Receiver buffer overflow
bit rx_buffer_overflow;

// USART Receiver interrupt service routine
interrupt [USART_RXC] void usart_rx_isr(void)
{
char status,data;
status=UCSRA;
data=UDR;
if ((status & (FRAMING_ERROR | PARITY_ERROR | DA-
TA_OVERRUN))==0)
{
rx_buffer[rx_wr_index++]=data;
#if RX_BUFFER_SIZE == 256
// special case for receiver buffer size=256
if (++rx_counter == 0) rx_buffer_overflow=1;
#else
if (rx_wr_index == RX_BUFFER_SIZE) rx_wr_index=0;
if (++rx_counter == RX_BUFFER_SIZE)
{
rx_counter=0;
rx_buffer_overflow=1;
}
#endif
}
}
}

#endif _DEBUG_TERMINAL_IO_

```



```

// Get a character from the USART Receiver buffer
#define _ALTERNATE_GETCHAR_
#pragma used+
char getchar(void)
{
char data;
while (rx_counter==0);
data=rx_buffer[rx_rd_index++];
#if RX_BUFFER_SIZE != 256
if (rx_rd_index == RX_BUFFER_SIZE) rx_rd_index=0;
#endif
asm("cli")
--rx_counter;
asm("sei")
return data;
}
#pragma used-
#endif

// Standard Input/Output functions
#include <stdio.h>

long int nilai=0,nilai2=0,array[10];
int i,a,flag=0,flag2=0;
char temp[32];
unsigned char jam, menit, detik, tanggal, bulan, tahun;
int l1,l2,l3,l4,d=0,e=0;
eeprom int status[5];
eeprom int password[4],correct_password[e];
int timer,data_key,bayar_detik,bayar_menit,bayar_jam;
int prev_jam[4],prev_menit[4],prev_detik[4];
int prev_jam2[4],prev_menit2[4],prev_detik2[4];
float freq; // to store value of frequency value
unsigned int u=0,dur; //i=number of overflows in one second
// dur to store the value of TCNT1 register
char buffer[8]; // to store the frequency value as a string to be
displayed on lcd
int counter_duit=0,bayar_duit=0;

```

```

// Timer1 overflow interrupt service routine
interrupt [TIM1_OVF] void timer1_ovf_isr(void)
{
// Place your code here
    u++;
}

void baca_ls(){
    if(ls1==0){l1=0;}
    else{l1=1;d=1;e=1;}
    if(ls2==0){l2=0;}
    else{l2=1;d=2;e=2;}
    if(ls3==0){l3=0;}
    else{l3=1;d=3;e=3;}
    if(ls4==0){l4=0;}
    else{l4=1;d=4;e=4;}
}

void waktu(){
    //lcd_clear();
    rtc_get_time(&jam,&menit,&detik);
    lcd_gotoxy(12,3);
    sprintf(temp,"%d:%d:%d ", jam, menit, detik);
    lcd_puts(temp);
}

void lock(){
    //digitalWrite(solenoid,HIGH);
}

void unlocked(){
    //digitalWrite(solenoid,LOW);
}

void simpan_variabel(){
    if (i==1){
        nilai=array[i];
        lcd_gotoxy(i-1,1);
        sprintf(temp,"%d",array[i]);
        lcd_puts(temp);
        password[e]=nilai;
        correct_password[e]=password[e];
        //status[d]=1;
    }
}

```

```

    delay_ms(500);
    lcd_gotoxy(i-1,1);
    lcd_putsf("*");
}
if (i>=2 && i<=4){
    nilai=(nilai*10)+array[i];
    lcd_gotoxy(i-1,1);
    sprintf(temp,"%d",array[i]);
    lcd_puts(temp);
    password[e]=nilai;
    correct_password[e]=password[e];
    delay_ms(500);
    lcd_gotoxy(i-1,1);
    lcd_putsf("*");
}
}
}

```

```

void simpan_variabel2(){
    if (i==1){
        nilai2=array[i];
        lcd_gotoxy(i-1,1);
        sprintf(temp,"%d",array[i]);
        lcd_puts(temp);
        delay_ms(500);
        lcd_gotoxy(i-1,1);
        lcd_putsf("*");
    }
    if (i>=2 && i<=4){
        nilai2=(nilai2*10)+array[i];
        lcd_gotoxy(i-1,1);
        sprintf(temp,"%d",array[i]);
        lcd_puts(temp);
        delay_ms(500);
        lcd_gotoxy(i-1,1);
        lcd_putsf("*");
    }
}
}
void hapus(){
    //c++;
}

```

```

nilai=(nilai-array[i])/10;
//itoa(nilai,temp);
lcd_clear();
lcd_gotoxy(0,0);
lcd_putsf("enter password");
for(a=0;a<i;a++){
    lcd_gotoxy(a,1);
    lcd_putsf("*");
}
lcd_gotoxy(7,1);
sprintf(temp,"%d",nilai);
lcd_puts(temp);
}
void enter(){
    i=0; flag=0; d=0;
    status[e]=1;
    password[e]=nilai;
    prev_jam[e]=jam;
    prev_menit[e]=menit;
    prev_detik[e]=detik;
    delay_ms(100);
    lcd_clear();
    delay_ms(500);
    lcd_gotoxy(3,0);
    lcd_putsf("Password Saved");
    lcd_gotoxy(0,1);
    sprintf(temp,"%d",nilai);
    lcd_puts(temp);
    delay_ms(1000);
    if(status[1]==1){
        door_lock1=0;
    }
    if(status[2]==1){
        door_lock2=0;
    }
    if(status[3]==1){
        door_lock3=0;
    }
    if(status[4]==1){

```

```

        door_lock4=0;
    }
}
void enter2(){

    prev_jam2[e]=jam;
    prev_menit2[e]=menit;
    prev_detik2[e]=detik;
    if(correct_password[e]==nilai2){
        bayar_detik=prev_detik2[e]-prev_detik[e];
        bayar_menit=prev_menit2[e]-prev_menit[e];
        bayar_jam=prev_jam2[e]-prev_jam[e];
        if(bayar_detik<0){
            bayar_detik=(prev_detik2[e]+60)-prev_detik[e];
            bayar_menit=bayar_menit-1;
        }
        if(bayar_menit<0){
            bayar_menit=(prev_menit2[e]+60)-prev_menit[e];
            bayar_jam=bayar_jam-1;
        }
        if(bayar_jam<0){
            bayar_jam=(prev_jam2[e]+60)-prev_jam[e]-1;
        }
        delay_ms(100);
        lcd_clear();
        delay_ms(250);
        lcd_gotoxy(3,0);
        lcd_putsf("password benar");
        lcd_gotoxy(0,1);
        lcd_putsf("Total Waktu: ");
        sprintf(temp,"%d:%d:%d
",bayar_jam,bayar_menit,bayar_detik);
        bayar_menit=((bayar_menit/15) *500)+500;
        lcd_puts(temp);
        lcd_gotoxy(0,2);
        lcd_putsf("Total Bayar: ");
        sprintf(temp,"Rp %d ",bayar_menit);
        lcd_puts(temp);
        delay_ms(1500);

```

```

        i=0; flag2=5; d=0;
        status[e]=0;
    }
    else{
        delay_ms(100);
        lcd_clear();
        delay_ms(250);
        lcd_gotoxy(3,0);
        lcd_putsf("password salah");
        delay_ms(1500);
        flag=0;
        i=0;
        status[e]=1;
        d=0;
    }
}
void keypad(){
    //lcd_gotoxy(0,0);
    //lcd_putsf("Enter Password");

    PORTA = 0b11110111;
    delay_ms(50);
    if(PINA.7==0){data_key=44; delay_ms(50);} //D
    else if(PINA.6==0){ enter(); delay_ms(50);}
    else if(PINA.5==0){ i++; array[i]=0; sim-
pan_variabel();delay_ms(50);}
    else if(PINA.4==0){ i--; hapus(); delay_ms(50);}

    PORTA = 0b11111011;
    delay_ms(50);
    if(PINA.7==0){data_key=33; delay_ms(50);} //C
    else if(PINA.6==0){i++; array[i]=9; simpan_variabel(); de-
lay_ms(50);}
    else if(PINA.5==0){i++; array[i]=8; simpan_variabel(); de-
lay_ms(50);}
    else if(PINA.4==0){i++; array[i]=7; simpan_variabel(); de-
lay_ms(50);}

    PORTA = 0b11111101;

```

```

        delay_ms(50);
        if(PINA.7==0){data_key=22; delay_ms(50);} //B
        else if(PINA.6==0){i++; array[i]=6; simpan_variabel(); de-
delay_ms(50);}
        else if(PINA.5==0){i++; array[i]=5; simpan_variabel(); de-
delay_ms(50);}
        else if(PINA.4==0){i++; array[i]=4; simpan_variabel(); de-
delay_ms(50);}

        PORTA = 0b11111110;
        delay_ms(50);
        if(PINA.7==0){data_key=11; delay_ms(50);} //A
        else if(PINA.6==0){i++; array[i]=3; simpan_variabel(); de-
delay_ms(50);}
        else if(PINA.5==0){i++; array[i]=2; simpan_variabel(); de-
delay_ms(50);}
        else if(PINA.4==0){i++; array[i]=1; simpan_variabel(); de-
delay_ms(50);}

        if(i>=4){i=4;}
    }
    void keypad2(){
        //lcd_gotoxy(0,0);
        //lcd_putsf("Enter Password");

        PORTA = 0b11110111;
        delay_ms(50);
        if(PINA.7==0){data_key=44; delay_ms(50);} //D
        else if(PINA.6==0){ enter2(); delay_ms(50);}
        else if(PINA.5==0){ i++; array[i]=0; sim-
pan_variabel2();delay_ms(50);}
        else if(PINA.4==0){ i--; hapus(); delay_ms(50);}

        PORTA = 0b11111011;
        delay_ms(50);
        if(PINA.7==0){data_key=33; delay_ms(50);} //C
        else if(PINA.6==0){i++; array[i]=9; simpan_variabel2(); de-
delay_ms(50);}
        else if(PINA.5==0){i++; array[i]=8; simpan_variabel2(); de-

```

```

lay_ms(50);}
    else if(PINA.4==0){i++; array[i]=7; simpan_variabel2(); de-
lay_ms(50);}

    PORTA = 0b11111101;
    delay_ms(50);
    if(PINA.7==0){data_key=22; delay_ms(50);} //B
    else if(PINA.6==0){i++; array[i]=6; simpan_variabel2(); de-
lay_ms(50);}
    else if(PINA.5==0){i++; array[i]=5; simpan_variabel2(); de-
lay_ms(50);}
    else if(PINA.4==0){i++; array[i]=4; simpan_variabel2(); de-
lay_ms(50);}

    PORTA = 0b11111110;
    delay_ms(50);
    if(PINA.7==0){data_key=11; delay_ms(50);} //A
    else if(PINA.6==0){i++; array[i]=3; simpan_variabel2(); de-
lay_ms(50);}
    else if(PINA.5==0){i++; array[i]=2; simpan_variabel2(); de-
lay_ms(50);}
    else if(PINA.4==0){i++; array[i]=1; simpan_variabel2(); de-
lay_ms(50);}
}
void baca_coin(){
    TIMSK=0x04;
    TCCR1B=0x07;
    delay_ms(1000);
    TCCR1B=0x00;
    TIMSK=0x00;
    dur=TCNT1;
    //freq = (((dur + i*65536)*60)/4.8)*0.0166;
    TCNT1=0x0000;
    i=0;
    //lcd_gotoxy(0,0);
    ///lcd_putsf("Coin=");
    //lcd_gotoxy(0,1);
    //ftoa(dur,2,buffer);
    //lcd_puts(buffer);

```



```

        //lcd_putsf("/M");
    }

    void main(void)
    {
        // Declare your local variables here

        // Input/Output Ports initialization
        // Port A initialization
        // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In
        Bit1=In Bit0=In
        DDRA=(0<<DDA7) | (0<<DDA6) | (0<<DDA5) | (0<<DDA4) |
        (1<<DDA3) | (1<<DDA2) | (1<<DDA1) | (1<<DDA0);
        // State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T
        Bit0=T
        PORTA=(1<<PORTA7) | (1<<PORTA6) | (1<<PORTA5) |
        (1<<PORTA4) | (1<<PORTA3) | (1<<PORTA2) | (1<<PORTA1) |
        (1<<PORTA0);

        // Port B initialization
        // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In
        Bit1=In Bit0=In
        DDRB=(0<<DDB7) | (0<<DDB6) | (0<<DDB5) | (0<<DDB4) |
        (0<<DDB3) | (0<<DDB2) | (0<<DDB1) | (0<<DDB0);
        // State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T
        Bit0=T
        PORTB=(0<<PORTB7) | (0<<PORTB6) | (0<<PORTB5) |
        (0<<PORTB4) | (0<<PORTB3) | (0<<PORTB2) | (0<<PORTB1) |
        (0<<PORTB0);
        //PORTB=0xFF;
        //DDRB=0x0F;
        // Port C initialization
        // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In
        Bit1=In Bit0=In
        DDRC=(0<<DDC7) | (0<<DDC6) | (0<<DDC5) | (0<<DDC4) |
        (1<<DDC3) | (1<<DDC2) | (1<<DDC1) | (1<<DDC0);
        // State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T
        Bit0=T
        PORTC=(1<<PORTC7) | (1<<PORTC6) | (1<<PORTC5) |

```

```
(1<<PORTC4) | (0<<PORTC3) | (0<<PORTC2) | (0<<PORTC1) |  
(0<<PORTC0);
```

```
//PORTC=0xF0;
```

```
//DDRC=0x00;
```

```
// Port D initialization
```

```
// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In  
Bit1=In Bit0=In
```

```
DDRD=(0<<DDD7) | (0<<DDD6) | (0<<DDD5) | (0<<DDD4) |  
(0<<DDD3) | (0<<DDD2) | (0<<DDD1) | (0<<DDD0);
```

```
// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T  
Bit0=T
```

```
PORTD=(1<<PORTD7) | (1<<PORTD6) | (0<<PORTD5) |  
(0<<PORTD4) | (0<<PORTD3) | (0<<PORTD2) | (0<<PORTD1) |  
(0<<PORTD0);
```

```
// Timer/Counter 0 initialization
```

```
// Clock source: System Clock
```

```
// Clock value: Timer 0 Stopped
```

```
// Mode: Normal top=0xFF
```

```
// OC0 output: Disconnected
```

```
TCCR0=(0<<WGM00) | (0<<COM01) | (0<<COM00) |  
(0<<WGM01) | (0<<CS02) | (0<<CS01) | (0<<CS00);
```

```
TCNT0=0x00;
```

```
OCR0=0x00;
```

```
// Timer/Counter 1 initialization
```

```
// Clock source: System Clock
```

```
// Clock value: Timer1 Stopped
```

```
// Mode: Normal top=0xFFFF
```

```
// OC1A output: Disconnected
```

```
// OC1B output: Disconnected
```

```
// Noise Canceler: Off
```

```
// Input Capture on Falling Edge
```

```
// Timer1 Overflow Interrupt: On
```

```
// Input Capture Interrupt: Off
```

```
// Compare A Match Interrupt: Off
```

```
// Compare B Match Interrupt: Off
```

```

    TCCR1A=(0<<COM1A1) | (0<<COM1A0) | (0<<COM1B1) |
(0<<COM1B0) | (0<<WGM11) | (0<<WGM10);
    TCCR1B=(0<<ICNC1) | (0<<ICES1) | (0<<WGM13) |
(0<<WGM12) | (0<<CS12) | (0<<CS11) | (0<<CS10);
    TCNT1H=0x00;
    TCNT1L=0x00;
    ICR1H=0x00;
    ICR1L=0x00;
    OCR1AH=0x00;
    OCR1AL=0x00;
    OCR1BH=0x00;
    OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer2 Stopped
// Mode: Normal top=0xFF
// OC2 output: Disconnected
    ASSR=0<<AS2;
    TCCR2=(0<<PWM2) | (0<<COM21) | (0<<COM20) |
(0<<CTC2) | (0<<CS22) | (0<<CS21) | (0<<CS20);
    TCNT2=0x00;
    OCR2=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
    TIMSK=(0<<OCIE2) | (0<<TOIE2) | (0<<TICIE1) |
(0<<OCIE1A) | (0<<OCIE1B) | (1<<TOIE1) | (0<<OCIE0) |
(0<<TOIE0);

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
    MCUCR=(0<<ISC11) | (0<<ISC10) | (0<<ISC01) |
(0<<ISC00);
    MCUCSR=(0<<ISC2);

// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity

```

```

// USART Receiver: On
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud Rate: 9600
UCSRA=(0<<RXC) | (0<<TXC) | (0<<UDRE) | (0<<FE) |
(0<<DOR) | (0<<UPE) | (0<<U2X) | (0<<MPCM);
UCSRB=(1<<RXCIE) | (0<<TXCIE) | (0<<UDRIE) |
(1<<RXEN) | (1<<TXEN) | (0<<UCSZ2) | (0<<RXB8) | (0<<TXB8);
UCSRC=(1<<URSEL) | (0<<UMSEL) | (0<<UPM1) |
(0<<UPM0) | (0<<USBS) | (1<<UCSZ1) | (1<<UCSZ0) |
(0<<UCPOL);
UBRRH=0x00;
UBRRL=0x67;

// Analog Comparator initialization
// Analog Comparator: Off
// The Analog Comparator's positive input is
// connected to the AIN0 pin
// The Analog Comparator's negative input is
// connected to the AIN1 pin
ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) |
(0<<ACIE) | (0<<ACIC) | (0<<ACIS1) | (0<<ACIS0);
SFIOR=(0<<ACME);

// ADC initialization
// ADC disabled
ADCSRA=(0<<ADEN) | (0<<ADSC) | (0<<ADATE) |
(0<<ADIF) | (0<<ADIE) | (0<<ADPS2) | (0<<ADPS1) |
(0<<ADPS0);

// SPI initialization
// SPI disabled
SPCR=(0<<SPIE) | (0<<SPE) | (0<<DORD) | (0<<MSTR) |
(0<<CPOL) | (0<<CPHA) | (0<<SPR1) | (0<<SPR0);

// TWI initialization
// TWI disabled
TWCR=(0<<TWEA) | (0<<TWSTA) | (0<<TWSTO) |
(0<<TWEN) | (0<<TWIE);

```

```

// Bit-Banged I2C Bus initialization
// I2C Port: PORTD
// I2C SDA bit: 5
// I2C SCL bit: 4
// Bit Rate: 100 kHz
// Note: I2C settings are specified in the
// Project|Configure|C Compiler|Libraries|I2C menu.
i2c_init();

// DS1307 Real Time Clock initialization
// Square wave output on pin SQW/OUT: Off
// SQW/OUT pin state: 0
rtc_init(0,0,0);

// Alphanumeric LCD initialization
// Connections are specified in the
// Project|Configure|C Compiler|Libraries|Alphanumeric LCD
menu:
// RS - PORTB Bit 0
// RD - PORTB Bit 2
// EN - PORTB Bit 3
// D4 - PORTB Bit 4
// D5 - PORTB Bit 5
// D6 - PORTB Bit 6
// D7 - PORTB Bit 7
// Characters/line: 16
lcd_init(20);
//rtc_set_time(14,51,30); //mengeset jam 18:18:30
printf("CLEARDATA\n"); //clears up any data left from previ-
ous projects
printf("LABEL,Time,Delay,Jam,Menit,Detik,Total,      \n");
//always write LABEL, so excel knows the next things will be the
names of the columns (instead of Acolumn you could write Time for
instance)
printf("RESETTIMER\n"); //resets timer to 0
// Global enable interrupts
door_lock1=1;
door_lock2=1;

```

```

door_lock3=1;
door_lock4=1;
#asm("sei")

while (1){
    start:
    while(flag==0){
        keypad();
        waktu();
        baca_ls();
        lcd_gotoxy(0,0);
        lcd_putsf("LokA:");
        //sprintf(temp,"%d ",status[1]);
        //lcd_puts(temp);
        lcd_gotoxy(0,1);
        lcd_putsf("LokB:");
        //sprintf(temp,"%d ",status[2]);
        //lcd_puts(temp);
        lcd_gotoxy(0,2);
        lcd_putsf("LokC:");
        //sprintf(temp,"%d ",status[3]);
        //lcd_puts(temp);
        lcd_gotoxy(0,3);
        lcd_putsf("LokD:");
        //sprintf(temp,"%d ",status[4]);
        //lcd_puts(temp);
        if(status[1]==1){
            lcd_gotoxy(5,0);
            lcd_putsf("penuh ");
            door_lock1=0;
        }
        else{
            lcd_gotoxy(5,0);
            lcd_putsf("kosong");
            door_lock1=1;
        }
        if(status[2]==1){
            lcd_gotoxy(5,1);
            lcd_putsf("penuh ");
        }
    }
}

```

```

        door_lock2=0;
    }
    else{
        lcd_gotoxy(5,1);
        lcd_putsf("kosong");
        door_lock2=1;
    }
    if(status[3]==1){
        lcd_gotoxy(5,2);
        lcd_putsf("penuh ");
        door_lock3=0;
    }
    else{
        lcd_gotoxy(5,2);
        lcd_putsf("kosong");
        door_lock3=1;
    }
    if(status[4]==1){
        lcd_gotoxy(5,3);
        lcd_putsf("penuh ");
        door_lock4=0;
    }
    else{
        lcd_gotoxy(5,3);
        lcd_putsf("kosong");
        door_lock4=1;
    }
    if(d==1){
        delay_ms(50);
        flag=1;
        lcd_clear();
        delay_ms(250);
        lcd_gotoxy(3,0);
        lcd_putsf("Set password LA:");
        lcd_gotoxy (0,2);
        lcd_putsf ("Simpan password (#)");
        while(flag==1){
            for(timer=0;timer<70;timer++){
                keypad();
            }
        }
    }

```

```

        waktu();
        delay_ms(20);
    }
    i=0;
    flag=0;
    d=0;
    lcd_clear();
}
}
else if(d==2){
    delay_ms(50);
    flag=1;
    lcd_clear();
    delay_ms(250);
    lcd_gotoxy(3,0);
    lcd_putsf("Set password LB:");
    lcd_gotoxy (0,2);
    lcd_putsf ("Simpan password (#)");
    while(flag==1){
        for(timer=0;timer<70;timer++){
            keypad();
            waktu();
            delay_ms(20);
        }
        i=0;
        flag=0;
        d=0;
        lcd_clear();
    }
}
else if(d==3){
    delay_ms(50);
    flag=1;
    lcd_clear();
    delay_ms(250);
    lcd_gotoxy(3,0);
    lcd_putsf("Set password LC:");
    lcd_gotoxy (0,2);
    lcd_putsf ("Simpan password (#)");
}
}

```



```

while(flag==1){
    for(timer=0;timer<70;timer++){
        keypad();
        waktu();
        delay_ms(20);
    }
    i=0;
    flag=0;
    d=0;
    lcd_clear();
}
}
else if(d==4){
    delay_ms(50);
    flag=1;
    lcd_clear();
    delay_ms(250);
    lcd_gotoxy(3,0);
    lcd_putsf("Set password LD:");
    lcd_gotoxy (0,2);
    lcd_putsf ("Simpan password (#)");
    while(flag==1){
        for(timer=0;timer<70;timer++){
            keypad();
            waktu();
            delay_ms(20);
        }
        i=0;
        flag=0;
        d=0;
        lcd_clear();
    }
}
if(data_key==11){
    flag=2;
    e=1;
    delay_ms(100);
    lcd_clear();
    lcd_gotoxy(3,0);
}

```

```

lcd_putsf("Enter Password LA");
while(flag==2){
    for(timer=0;timer<40;timer++){
        keypad2();
        waktu();
        delay_ms(20);
    }

    data_key=0;
    //status[e]=0;
    i=0;
    flag=0;
    d=0;
    lcd_clear();
}
while(flag2==5){
    delay_ms(50);
    lcd_gotoxy(0,0);
    lcd_putsf("Silahkan Bayar:  ");
    baca_coin();
    if(dur==2 || dur==1){
        delay_ms(50);
        lcd_clear();
        delay_ms(100);
        counter_duit++;
        bayar_duit=counter_duit*500;
        lcd_gotoxy(0,1);
        sprintf(temp, "%d ", bayar_duit);
        lcd_puts(temp);
        if(bayar_duit>=bayar_menit){
            door_lock1=1;
            delay_ms(200);
            lcd_gotoxy(0,2);
            lcd_putsf("Cukup!!!");
            printf("DATA,TIME,"); //writes the time in the
first column A and the time since the measurements started in column
B

            printf("A");
            printf(",");

```

```

        printf("%d",bayar_duit);
        printf("\n");
        delay_ms(1500);
        i=0;
        lcd_clear();
        bayar_duit=0;
        counter_duit=0;
        flag2=0;
        flag=0;
        d=0;
    }
}

}
}
else if(data_key==22){
    flag=2;
    e=2;
    delay_ms(100);
    lcd_clear();
    lcd_gotoxy(3,0);
    lcd_putsf("Enter Password LB");
    while(flag==2){
        for(timer=0;timer<40;timer++){
            keypad2();
            waktu();
            delay_ms(20);
        }
        //i=0;
        //flag=0;
        //d=0;
        data_key=0;
        //status[e]=0;
        i=0;
        flag=0;
        d=0;
        lcd_clear();
    }
    while(flag2==5){

```

```

delay_ms(50);
lcd_gotoxy(0,0);
lcd_putsf("Silahkan Bayar:  ");
baca_coin();
if(dur==2 || dur==1){
    delay_ms(50);
    lcd_clear();
    delay_ms(100);
    counter_duit++;
    bayar_duit=counter_duit*500;
    lcd_gotoxy(0,1);
    sprintf(temp, "%d ", bayar_duit);
    lcd_puts(temp);
    if(bayar_duit>=bayar_menit){
        door_lock2=1;
        delay_ms(200);
        lcd_gotoxy(0,2);
        lcd_putsf("Cukup!!!");
        printf("DATA,TIME,"); //writes the time in the
first column A and the time since the measurements started in column
B

```

```

        printf("B");
        printf(",");
        printf("%d", bayar_duit);
        printf("\n");
        delay_ms(1500);
        i=0;
        lcd_clear();
        bayar_duit=0;
        counter_duit=0;
        flag2=0;
        flag=0;
        d=0;
    }
}
}
else if(data_key==33){

```

```

flag=2;
e=3;
delay_ms(100);
lcd_clear();
lcd_gotoxy(3,0);
lcd_putsf("Enter Password LC");
while(flag==2){
    for(timer=0;timer<40;timer++){
        keypad2();
        waktu();
        delay_ms(20);
    }
    //i=0;
    //flag=0;
    //d=0;
    data_key=0;
    //status[e]=0;
    i=0;
    flag=0;
    d=0;
    lcd_clear();
}
while(flag2==5){
    delay_ms(50);
    lcd_gotoxy(0,0);
    lcd_putsf("Silahkan Bayar:  ");
    baca_coin();
    if(dur==2 || dur==1){
        delay_ms(50);
        lcd_clear();
        delay_ms(100);
        counter_duit++;
        bayar_duit=counter_duit*500;
        lcd_gotoxy(0,1);
        sprintf(temp,"%d ",bayar_duit);
        lcd_puts(temp);
        if(bayar_duit>=bayar_menit){
            door_lock3=1;
            delay_ms(200);
        }
    }
}

```



```

//status[e]=0;
i=0;
flag=0;
d=0;
lcd_clear();
}
while(flag2==5){
delay_ms(50);
lcd_gotoxy(0,0);
lcd_putsf("Silahkan Bayar:  ");
baca_coin();
if(dur==2 || dur==1){
delay_ms(50);
lcd_clear();
delay_ms(100);
counter_duit++;
bayar_duit=counter_duit*500;
lcd_gotoxy(0,1);
sprintf(temp,"%d ",bayar_duit);
lcd_puts(temp);
if(bayar_duit>=bayar_menit){
door_lock4=1;
delay_ms(200);
lcd_gotoxy(0,2);
lcd_putsf("Cukup!!!");
printf("DATA,TIME,"); //writes the time in the
first column A and the time since the measurements started in column
B

printf("D");
printf(",");
printf("%d",bayar_duit);
printf("\n");
delay_ms(1500);
i=0;
lcd_clear();
bayar_duit=0;
counter_duit=0;
flag2=0;
flag=0;

```





## RIWAYAT HIDUP PENULIS



Nama : Hendra Wahyu  
Budianto  
TTL : Surabaya, 16 April  
1997  
Jenis Kelamin : Laki-laki  
Agama : Islam  
Alamat : Pesona Alam Gunung  
Anyar I A-8 Surabaya  
Telp/HP : 087852600009  
E-mail : hendra8wahyu@gmail  
.com

### RIWAYAT PENDIDIKAN:

1. 2003-2009 : SD Negeri Rungkut Menanggal II Surabaya
2. 2009-2012 : SMP Negeri 23 Surabaya
3. 2012-2015 : SMA Negeri 17 Surabaya
4. 2015-2018 : Program Studi Komputer Kontrol, Departemen Teknik Elektro Otomasi, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember, Surabaya

### PENGALAMAN KERJA

1. Kerja Praktek di PT. Indofood Sukses Makmur Tbk. Divisi Bogasari Surabaya

### PENGALAMAN ORGANISASI

1. KESMA HIMAD3TEKTRO 2016/2017
2. KESMA HIMAD3TEKTRO 2017/2018

----Halaman ini sengaja dikosongkan----