

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Pada penelitian sebelumnya yang dilakukan (Desriyeni dan Miftahur, 2018) dalam laporan akhir yang berjudul “**Alat Pembuka Pintu Kamar Menggunakan Sensor Sidik Jari**”. Permasalahannya adalah bagaimana merancang sebuah alat menggunakan sensor sidik jari atau *fingerprint* yang di kombinasikan dengan arduino. Arduino merupakan suatu modul yang cukup mudah digunakan karena sudah terdapat kombinasi CPU dengan memori dan I/O didalam suatu *chip IC (Integrated Circuit)*. Alat ini menggunakan sensor sidik jari *fingerprint* R305 sebagai sensor pendeteksi sidik jari dan arduino sebagai media pengontrol dan pemroses *input* data kontrol. Pada laporan akhir ini terdapat kekurangan dan kelebihan dalam pembuatan alat. Kelebihannya yaitu, Pengisian data sidik jari dilakukan diluar perangkat alat menggunakan aplikasi SFG Demo yang telah disediakan untuk modul sensor sidik jari jenis ini. Kekurangan dalam pembuatan alat ini yaitu penulis menggunakan sensor *fingerprint* tipe R305 dimana tipe ini hanya dapat digunakan pada modular duino karena sensor ini dirancang untuk modular duino.

Pada penelitian sebelumnya yang dilakukan (Yuliza dan Toibah, 2015) dalam laporan akhir yang berjudul “**Alat Keamanan Pintu Brankas Berbasis Sensor Sidik Jari Dan Password Digital Dengan Menggunakan Mikrokontroler Atmega 16**”. Permasalahan dalam laporan akhir ini adalah bagaimana membuat sistem keamanan pintu brankas menggunakan sensor sidik jari seri R305. Peralatan utama lainnya meliputi mikrokontroler ATmega 16 dan Visual Basic 6.0. Hasil penelitian menunjukkan bahwa Alat Keamanan Pintu Brankas dapat dimonitor dengan menggunakan aplikasi yang dibuat menggunakan *Visual Basic 6.0*. Sesuai dengan perintah yang diberikan. Hasil pengujian alat menunjukkan bahwa respon alat terhadap perintah-perintah dari aplikasi berjalan dengan normal. Keberhasilan dari hasil ujicoba alat adalah 100%. Pada laporan akhir ini terdapat kekurangan dan kelebihan dalam pembuatan alat. Kelebihannya

yaitu, alat ini dapat dimonitor untuk menampilkan data dengan membuat *interface* menggunakan *Visual Basic 6.0*. Kekurangannya yaitu, tidak menambahkan waktu berapa lama akses pintu brankas terbuka.

Pada penelitian sebelumnya yang dilakukan (Lingga, dkk. 2017) dalam laporan akhir yang berjudul **“Sistem Keamanan Buka Tutup Kunci Brankas Menggunakan Sidik Jari Berbasis Arduino Mega”**. Permasalahan dalam Laporan akhir ini adalah bagaimana membuat sistem buka tutup brankas menggunakan sidik jari agar tidak mudah dibobol. Sistem ini dibuat dengan menggunakan sumber tegangan +5V DC untuk modul sidik jari, *Micro Switch*, LCD, LED, *Buzzer*, dan +12V DC untuk Solenoid. Sidik jari dan penekanan pada *Switch* sebagai masukan, Arduino Mega sebagai Mikrokontroler, serta Solenoid, LCD, LED, dan *Buzzer* sebagai media keluarannya. Berdasarkan hasil analisis dan pengujian yang telah dilakukan, diperoleh kesimpulan bahwa sistem keamanan brankas dalam penelitian ini dapat bekerja secara optimal. Alat ini dapat membuka menggunakan sidik jari dan mengunci brankas menggunakan sidik jari atau *Switch*, sehingga dapat meminimalkan tindak kejahatan pencurian terhadap barang berharga. Pada laporan akhir ini terdapat kekurangan dan kelebihan. Kelebihannya yaitu, Alat ini memiliki indikator LED hijau menyala jika sidik jari yang ditempelkan terdaftar dan solenoid bergerak membuka, dan LED merah menyala dan *buzzer* berbunyi jika sidik jari atau terjadi penekanan pada *switch* terjadi untuk mengunci brankas. Kekurangannya yaitu, belum menambahkan waktu berapa lama akses pintu brankas terbuka.

Pada penelitian sebelumnya yang dilakukan (Tobing, 2014) dalam laporan akhir yang berjudul **“Rancang Bangun Pengaman Pintu Menggunakan Sidik Jari (*Fingerprint*) dan Smartphone Android Berbasis Mikrokontroler Atmega8”**. Permasalahannya adalah merancang sistem keamanan pintu dengan menggunakan *fingerprint* dan aplikasi yang dipasang pada smartphone android. *Fingerprint* yang telah diakses oleh jari-jari dari anggota keluarga akan memberikan data kepada mikrokontroler untuk diolah yang kemudian akan memberikan perintah kepada solenoid untuk membuka kunci pintu. Selain itu sistem keamanan ini juga dapat dikendalikan lewat smartphone android yang telah

diinstal aplikasi yang dirancang sendiri oleh penulis. Metode penelitian dalam skripsi ini meliputi studi pustaka, perancangan sistem, pembuatan mekanik, perancangan perangkat keras (*hardware*) dan perangkat lunak (*software*). Pada laporan akhir ini terdapat kekurangan dan kelebihan dalam pembuatan alat. Kelebihannya yaitu, sistem keamanan dikendalikan lewat smartphone android yang telah diinstal aplikasi yang dirancang sendiri oleh penulis. Dengan jarak kontrol antara smartphone android dengan perangkat sistem maksimum sebesar 24 meter. Kekurangannya yaitu, ketika sistem tidak dapat bekerja, menyikapi hal tersebut solusi yang diberikan oleh penulis adalah melakukan *flashing* ulang *firmware* mikrokontroler dengan menghubungkan perangkat bluetooth dengan komputer yang sebelumnya telah terinstal aplikasi Bascom AVR, akan tetapi *flashing firmware* dengan komunikasi Bluetooth sering *error* jadi harus di ulang-ulang sampai bisa terhubung.

Jika dibandingkan dengan penelitian terdahulu dengan penelitian yang akan penulis lakukan, hanya berbeda pada tempat implementasi alat, dimana penulis akan mengimplementasikan kunci pintu pada laboratorium 6 jurusan Teknik Komputer Politeknik Negeri Sriwijaya menggunakan *fingerprint* untuk input data sidik jari dan modul mikrokontroler Arduino Uno.

2.2 Mikrokontroler

Mikrokontroler adalah mikroprosesor yang dikhususkan untuk instrumentasi dan kendali. Contoh aplikasi pada kendali motor, berperan seperti PLC (*Programmable Logic Controller*), pengaturan pengapian dan injeksi bahan bakar pada kendaraan bermotor atau alat pengukur suatu besaran, seperti suhu, tekanan, kelembaban dan lain-lain. Mikrokontroler merupakan suatu alat elektronika digital yang mempunyai masukan dan keluaran serta kendali dengan program yang bisa ditulis dan dihapus dengan cara khusus, cara kerja mikrokontroler sebenarnya membaca dan menulis data. Mikrokontroler merupakan komputer didalam chip yang digunakan untuk mengontrol peralatan elektronik, yang menekankan efisiensi dan efektifitas biaya. (Sumardi, 2013).

Beberapa fitur yang umumnya ada di dalam mikrokontroler adalah sebagai berikut:

1. ROM (*Read Only Memory*)

ROM berfungsi untuk tempat penyimpanan variable. Memori ini bersifat *volatile* yang berarti akan kehilangan semua datanya jika tidak mendapat catu daya.

2. RAM (*Random Access Memory*)

RAM digunakan oleh mikrokontroler untuk tempat penyimpanan program yang akan diberikan oleh *user*.

3. Register

Merupakan tempat penyimpanan nilai – nilai yang akan digunakan dalam proses yang telah disediakan oleh mikrokontroler.

4. Special Function Register

Merupakan register khusus yang berfungsi untuk mengatur jalannya mikrokontroler. Register ini terletak pada RAM.

5. Input dan Output Pin

Pin input adalah bagian yang berfungsi sebagai penerima signal dari luar, pin ini dapat dihubungkan ke berbagai media inputan seperti keypad, sensor, dan sebagainya. Pin output adalah bagian yang berfungsi untuk mengeluarkan signal dari hasil proses algoritma mikrokontroler.

6. Interupt

Interupt adalah bagian mikrokontroler yang berfungsi sebagai bagian yang dapat melakukan interupsi, sehingga ketika program utama sedang berjalan, program utama tersebut dapat di interupsi dan menjalankan program instruksi terlebih dahulu. (Agustina, 2016).

2.3 Modul Mikrokontroler Arduino Uno

Arduino uno adalah suatu papan elektronik yang mengandung mikrokontroler Atmega 328 (sebuah keping yang secara langsung fungsional bertindak seperti sebuah komputer). Peranti ini dimanfaatkan untuk mewujudkan rangkaian elektronik dari yang sederhana hingga kompleks. (Setiawan, 2015).

Arduino UNO mempunyai 14 pin digital *input/output* (6 di antaranya dapat digunakan sebagai output PWM), 6 input analog, sebuah osilator Kristal 16 MHz, sebuah koneksi USB, sebuah power jack, sebuah ICSP header, dan sebuah tombol reset. Arduino UNO memuat semua yang dibutuhkan untuk menunjang mikrokontroler, mudah menghubungkannya ke sebuah komputer dengan sebuah kabel USB atau mensuplainya dengan sebuah adaptor AC ke DC atau menggunakan baterai untuk memulainya.



Gambar 2.1 Arduino uno

Sumber : <http://www.arduino.cc>

(PCINT14/RESET) PC6	1	28	PC5 (ADC5/SCL/PCINT13)
(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12)
(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11)
(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10)
(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9)
(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8)
VCC	7	22	GND
GND	8	21	AREF
(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC
(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5)
(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4)
(PCINT22/OC0A/AIN0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3)
(PCINT23/AIN1) PD7	13	16	PB2 (SS/OC1B/PCINT2)
(PCINT0/CLKO/ICP1) PB0	14	15	PB1 (OC1A/PCINT1)

Gambar 2.2 Atmega 328

Sumber : (<http://www.arduino.cc>)

Tabel 2.1 Spesifikasi Arduino Uno

Mikrokontroler	Arduino Uno
Tegangan pengoperasian	5V
Tegangan input	7-12V
Batas tegangan input	6-20V
Jumlah pin I/O digital	14 (6PWM)
Jumlah pin input analog	6
Arus DC tiap pin I/O	40 Ma
Arus DC untuk pin 3.3V	50Ma
Memori Flash	32 KB (Atmega328)
SRAM	2 KB (Atmega328)
EEPROM	1 KB (Atmega28)
Clock Speed	16 MHz

2.4 Program Arduino IDE

Kode Program *Arduino* biasa disebut *sketch* dan dibuat menggunakan bahasa pemrograman C. Program atau *sketch* yang sudah selesai ditulis di *Arduino IDE* bisa langsung *dicompile* dan *diupload* ke *Arduino Board*. Secara sederhana, *sketch* dalam *Arduino* dikelompokkan menjadi 3 blok : *Header*, *Setup* dan *Loop*.

A. *Header*

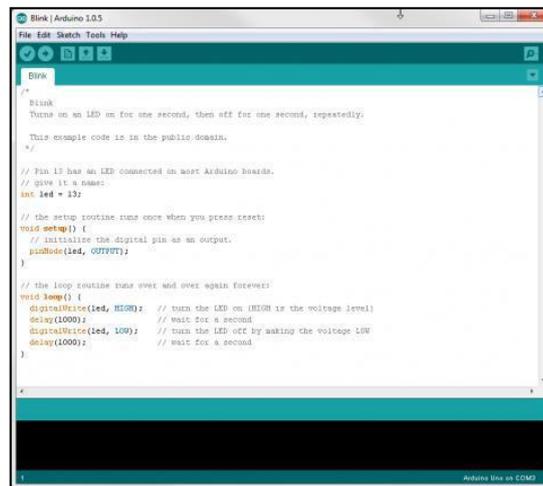
Pada bagian ini biasanya ditulis definisi-definisi penting yang akan digunakan selanjutnya dalam program, misalnya penggunaan *library* dan pendefinisian *variable*. Code dalam blok ini dijalankan hanya sekali pada waktu *compile*.

B. *Setup*

Di sinilah awal program *Arduino* berjalan, yaitu di saat awal atau ketika *power on Arduino board*. Biasanya di blok ini diisi penentuan apakah suatu pin digunakan sebagai *input* atau *output*, menggunakan perintah pin Mode.

C. *Loop*

Blok ini akan dieksekusi secara terus menerus. Apabila program sudah sampai akhir blok, maka akan dilanjutkan dengan mengulang eksekusi dari awal blok. Program akan berhenti apabila tombol power Arduino di matikan. Di sinilah fungsi utama program Arduino kita berada. (Rafly dan M.Alhafiz,2018).



```

Blink
Turns on an LED on for one second, then off for one second, repeatedly.

This example code is in the public domain.
*/

// Pin 13 has an LED connected on most Arduino boards.
// Give it a name:
const int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}

```

Gambar 2.3 Tampilan Program Arduino IDE

(Sumber : Rafly dan M Alhafiz, 2018)

2.5 *Fingerprint*

Sidik jari (bahasa Inggris: *fingerprint*) adalah hasil reproduksi tapak jari baik yang sengaja diambil, dicapkan dengan tinta, maupun bekas yang ditinggalkan pada benda karena pernah tersentuh kulit telapak tangan atau kaki.

Sensor sidik jari (*Fingerprint*) telah banyak yang beredar di pasaran, untuk itu salah satu sensor sidik jari yang murah meriah akan tetapi sangat baik kerjanya adalah *fingerprint* dari adafruit.com yang mana sensor ini akan mengirim data ID sidik jari melalui komunikasi serial. (Tobing, 2014).



Gambar 2.4 *Fingerprint Sensor*

Sumber : <http://adafruit.com/fingerprint>

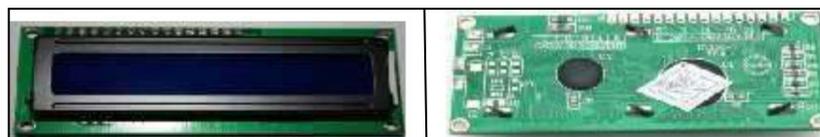
<i>Fingerprint Spesifikasi :</i>	
•	Supply voltage: 3.6 – 6.0VDC
•	Operating current: 120mA max
•	Peak current: 150mA max
•	Fingerprint imaging time: <1.0 seconds
•	Window area: 14mm x 18mm
•	Signature file: 256 bytes
•	Template file: 512 bytes
•	Storage capacity: 162 templates
•	Safety ratings (1-5 low to high safety)
•	False Acceptance Rate: <0.001% (Security level 3)
•	False Reject Rate: <1.0% (Security level 3)
•	Interface: TTL Serial
•	Baud rate: 9600, 19200, 28800, 38400, 57600 (default is 57600)
•	Working temperature rating: -20°C to +50°C
•	Working humidity: 40%-85% RH
•	Full Dimensions: 56 x 20 x 21.5mm
•	Exposed Dimensions (when placed in box): 21mm x 21mm x 21mm triangular
•	Weight: 20 grams

Gambar 2.5 Spesifikasi *Fingerprint*

Sumber : <http://adafruit.com/fingerprint>

2.6 *Liquid Crystal Display (LCD)*

Liquid Crystal Display (LCD) adalah komponen yang dapat menampilkan tulisan. Salah satu jenisnya memiliki dua baris dengan setiap baris terdiri dari 16 karakter. LCD seperti itu bisa disebut LCD 16x2 Cm. (Setiawan, 2015).



(a)

(b)

Gambar 2.6 LCD 16x2 cm

A. LCD Tampak Atas

B. LCD Tampak Bawah

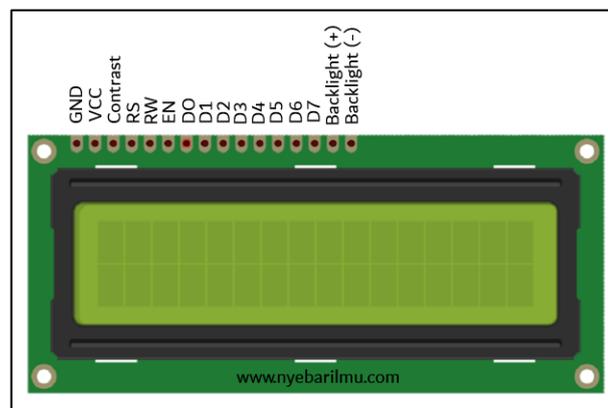
Sumber : <http://www.arduino.cc>

2.6.1 Fitur dan Spesifikasi LCD 16x2

A. Fitur – fitur LCD 16x2

- Terdiri dari 16 kolom dan 2 baris.
- Dilengkapi dengan *back light*.
- Mempunyai 192 karakter tersimpan.
- Dapat dialamatasi dengan mode 4-bit dan 8-bit.
- Terdapat karakter generator terprogram.

B. Spesifikasi LCD 16x2



Gambar 2.7 Pin-pin LCD 16x2

Sumber : <http://www.arduino.cc>

Keterangan :

1. **GND** : catu daya 0Vdc
2. **VCC** : catu daya positif
3. **Constrate** : untuk kontras tulisan pada LCD
4. **RS** atau **Register Select** :
 - High : untuk mengirim data
 - Low : untuk mengirim instruksi
5. **R/W** atau **Read/Write**
 - High : mengirim data

- Low : mengirim instruksi
 - Disambungkan dengan LOW untuk pengiriman data ke layar
6. **E (enable)** : untuk mengontrol ke LCD ketika bernilai LOW, LCD tidak dapat diakses
 7. **D0 – D7** = Data Bus 0 – 7
 8. **Backlight +** : disambungkan ke VCC untuk menyalakan lampu latar
 9. **Backlight –** : disambungkan ke GND untuk menyalakan lampu latar.
- (Agus, 2017).

2.6.2 Cara Kerja LCD Secara Umum

Pada aplikasi umumnya RW diberi logika rendah “0”. Bus data terdiri dari 4-bit atau 8-bit. Jika jalur data 4-bit maka yang digunakan ialah DB4 sampai dengan DB7. Sebagaimana terlihat pada table diskripsi, *interface* LCD merupakan sebuah paralel bus, dimana hal ini sangat memudahkan dan sangat cepat dalam pembacaan dan penulisan data dari atau ke LCD. Kode ASCII yang ditampilkan sepanjang 8-bit dikirim ke LCD secara 4-bit atau 8 bit pada satu waktu. Jika mode 4-bit yang digunakan, maka 2 nibble data dikirim untuk membuat sepenuhnya 8-bit (pertama dikirim 4-bit MSB lalu 4-bit LSB dengan pulsa clock EN setiap nibblenya). Jalur kontrol EN digunakan untuk memberitahu LCD bahwa mikrokontroller mengirimkan data ke LCD. Untuk mengirim data ke LCD program harus menset EN ke kondisi high “1” dan kemudian menset dua jalur kontrol lainnya (RS dan R/W) atau juga mengirimkan data ke jalur data bus.

Saat jalur lainnya sudah siap, EN harus diset ke “0” dan tunggu beberapa saat (tergantung pada datasheet LCD), dan set EN kembali ke high “1”. Ketika jalur RS berada dalam kondisi low “0”, data yang dikirimkan ke LCD dianggap sebagai sebuah perintah atau instruksi khusus (seperti bersihkan layar, posisi kursor dll). Ketika RS dalam kondisi high atau “1”, data yang dikirimkan adalah data ASCII yang akan ditampilkan dilayar. Misal, untuk menampilkan huruf “A” pada layar maka RS harus diset ke “1”. Jalur kontrol R/W harus berada dalam kondisi low (0) saat informasi pada data bus akan dituliskan ke LCD. Apabila R/W berada dalam kondisi high “1”, maka program akan melakukan query

(pembacaan) data dari LCD. Instruksi pembacaan hanya satu, yaitu Get LCD status (membaca status LCD), lainnya merupakan instruksi penulisan. Jadi hampir setiap aplikasi yang menggunakan LCD, R/W selalu diset ke “0”. Jalur data dapat terdiri 4 atau 8 jalur (tergantung mode yang dipilih pengguna), DB0, DB1, DB2, DB3, DB4, DB5, DB6 dan DB7. Mengirim data secara parallel baik 4-bit atau 8-bit merupakan 2 mode operasi primer. Untuk membuat sebuah aplikasi interface LCD, menentukan mode operasi merupakan hal yang paling penting.

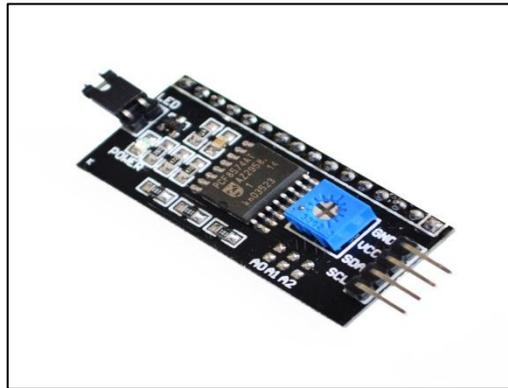
Mode 8-bit sangat baik digunakan ketika kecepatan menjadi keutamaan dalam sebuah aplikasi dan setidaknya minimal tersedia 11 pin I/O (3 pin untuk kontrol, 8 pin untuk data). Sedangkan mode 4 bit minimal hanya membutuhkan 7-bit (3 pin untuk kontrol, 4 pin untuk data). Bit RS digunakan untuk memilih apakah data atau instruksi yang akan ditransfer antara mikrocontroller dan LCD. Jika bit ini di set ($RS = 1$), maka byte pada posisi kursor LCD saat itu dapat dibaca atau ditulis. Jika bit ini di reset ($RS = 0$), merupakan instruksi yang dikirim ke LCD atau status eksekusi dari instruksi terakhir yang dibaca. (Aris, 2012).

2.7 I2C

I2C (*Inter Integrated Circuit*) merupakan standar komunikasi serial dua arah menggunakan dua saluran yang didesain khusus untuk mengirim maupun menerima data. I2C didesain oleh Philips pada awal tahun 1980-an untuk memudahkan komunikasi antar komponen pada suatu rangkaian. Semula kecepatan komunikasi maksimum 100kbps karena kecepatan tinggi belum dibutuhkan. Pada tahun 1998 dibutuhkan kecepatan tinggi, sehingga terdapat mode kecepatan 3,4Mbps. I2C dapat digunakan untuk menghubungkan komponen yang terhubung melalui kabel. Ciri utama dari I2C terletak pada bentuknya yang simpel. Fitur dari bus I2C adalah (Endaryono dkk, 2014):

1. Memiliki 2 jalur dan kabel.
2. Pada bus I2C tidak ada aturan baud rate yang signifikan.
3. Setiap komponen yang terhubung dengan bus memiliki alamat yang berbeda.

4. I2C bertugas sebagai master pengontrol jalur komunikasi dengan mengatur clock sekaligus menentukan pengguna jalur komunikasi.
5. I2C bisa digunakan sebagai multi-master yang memiliki kemampuan pendeteksi tabrakan datadan arbitrase.



Gambar 2.8 I2C

Sumber : <http://adafruit.com/fingerprint>

Sistem I2C terdiri dari saluran SCL (*Serial Clock*) dan SDA (*Serial Data*) yang membawa informasi data antara I2C dengan pengontrol. Untuk menyambungkan LCD dengan *board* arduino uno memerlukan 6 pin digital untuk mengendalikan sebuah modul LCD. Modul I2C yang digunakan pada tugas akhir ini adalah I2C LCD 1602 2004 LCD 16x2. Dengan menggunakan modul I2C ini dapat mengurangi penggunaan pin pada *board* arduino yang hanya menggunakan 2 pin analog A4 dan A5 yang dihubungkan dengan SDA dan SCL untuk menghubungkan LCD dengan *board* arduino uno.

2.8 Solenoid Door Lock

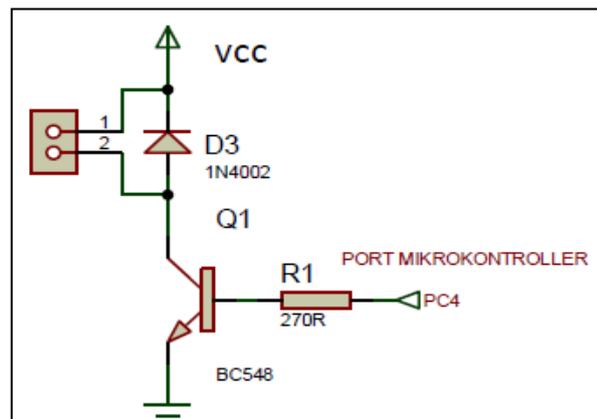
Solenoida atau *Solenoid* adalah perangkat elektromagnetik yang dapat mengubah energi listrik menjadi energi gerakan. Energi gerakan yang dihasilkan oleh Solenoid biasanya hanya gerakan mendorong (*push*) dan menarik (*pull*). Pada dasarnya, Solenoid hanya terdiri dari sebuah kumparan listrik (*electrical coil*) yang dililitkan di sekitar tabung silinder dengan aktuator ferro-magnetik atau sebuah Plunger yang bebas bergerak “Masuk” dan “Keluar” dari bodi kumparan.

Solenoid sering digunakan di aplikasi-aplikasi seperti menggerakkan dan mengoperasikan mekanisme robotik, membuka dan menutup pintu dengan listrik, membuka dan menutup katup (valve) dan sebagai sakelar listrik. Solenoida yang dapat membuka dan menutup katup biasanya disebut dengan *Solenoid Valve* (Solenoida Katup). (Kho, 2015)



Gambar 2.9 *Solenoid door lock*

Sumber: <http://www.waferstar.com/image/Lock-Solenoid-02.jpg>

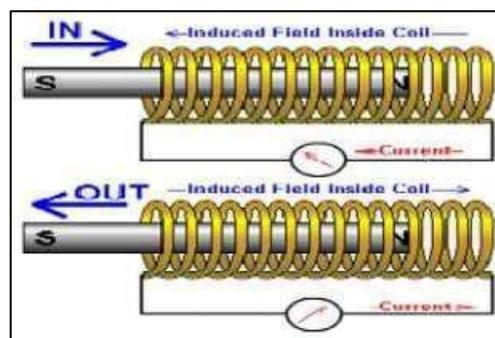


Gambar 2.10 *Skematik Solenoid Door Lock*

Sumber : www.nyebarilmu.com

2.8.1 Sistem Kerja Solenoid

Didalam solenoid terdapat kawat melingkar pada inti besi. Ketika arus listrik melalui kawat ini, maka terjadi medan magnet untuk menghasilkan energi yang bisa mendorong inti besi. Poros dalam dari solenoid adalah piston seperti silinder terbuat dari besi atau baja, yang disebut *pluger* (setara dengan sebuah dinamo) medan magnet kemudian menerapkan kekuatan untuk pluger ini, baik menarik atau *repling* (kembali posisi). Ketika medan magnet dimatikan, pegas pluger kemudian kembali ke posisi semula. (Setiawan, 2015).

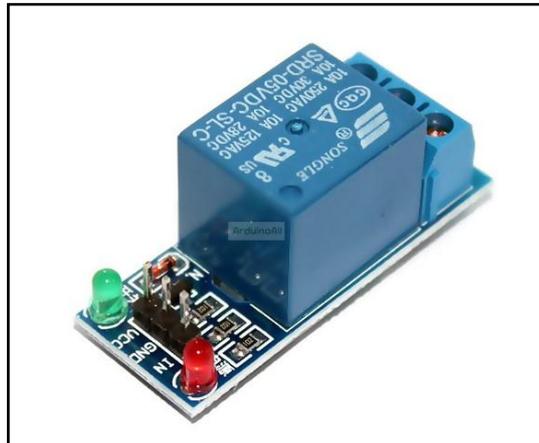


Gambar 2.11 Sistem Kerja *Solenoid Door Lock*

Sumber : <http://adafruit.com/fingerprint>

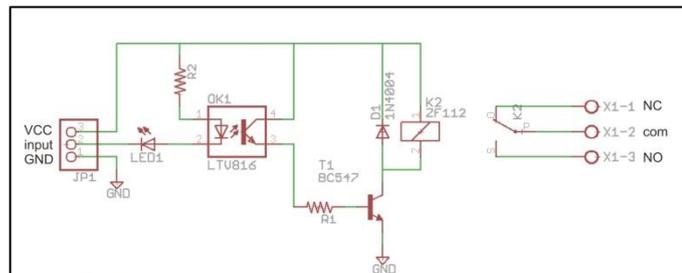
2.9 *Driver Relay*

Relay adalah saklar (*switch*) yang dioperasikan secara listrik dan merupakan komponen electromechanical (Elektromekanikal) yang terdiri dari 2 bagian utama yakni elektromagnet (*coil*) dan mekanikal (seperangkat Kontak saklar/*switch*). *Relay* menggunakan prinsip elektromagnetik untuk menggerakkan kontak saklar sehingga dengan arus listrik yang kecil (*low power*) dapat menghantarkan listrik yang bertegangan lebih tinggi. Sebagai contoh, dengan *relay* yang menggunakan elektromagnet 5V dan 50 mA mampu menggerakkan *armature Relay* (yang berfungsi sebagai saklarnya) untuk menghantarkan listrik 220V 2A. (Kho, 2015).



Gambar 2.12 *Driver Relay*

Sumber : www.teknikelektronika.com



Gambar 2.13 *Skematik Driver Relay*

Sumber : www.teknikelektronika.com

2.10 Buzzer

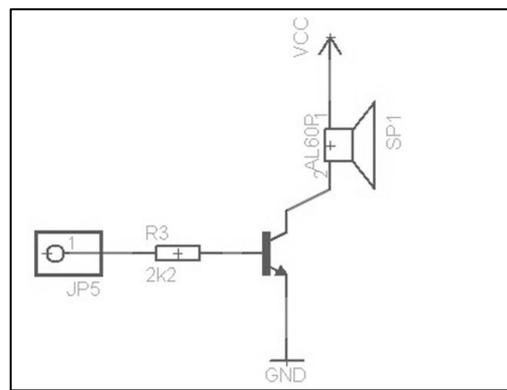
Buzzer adalah suatu alat yang dapat mengubah sinyal listrik menjadi sinyal suara. Pada umumnya buzzer digunakan untuk alarm, karena penggunaannya cukup mudah yaitu dengan memberikan tegangan input maka buzzer akan mengeluarkan bunyi. Frekuensi suara yang di keluarkan oleh buzzer yaitu antara 1-5 KHz. Pada dasarnya prinsip kerja *buzzer* hampir sama dengan loud speaker, jadi buzzer juga terdiri dari kumparan yang terpasang pada diafragma dan kemudian kumparan tersebut dialiri arus sehingga menjadi elektromagnet, kumparan tadi akan tertarik ke dalam atau keluar, tergantung dari arah arus dan polaritas magnetnya. karena kumparan dipasang pada diafragma maka setiap gerakan kumparan akan menggerakkan diafragma secara bolak-balik sehingga membuat udara bergetar yang akan menghasilkan suara. *Buzzer* biasa digunakan

sebagai indikator bahwa proses telah selesai atau terjadi suatu kesalahan pada sebuah alat (alarm). (Alfareza,2016).



Gambar 2.14 Buzzer

Sumber: <https://indraharja.wordpress.com/2012/01/07/pengertian-buzzer/>



Gambar 2.15 Skematik Buzzer

Sumber : www.nyebarilmu.com

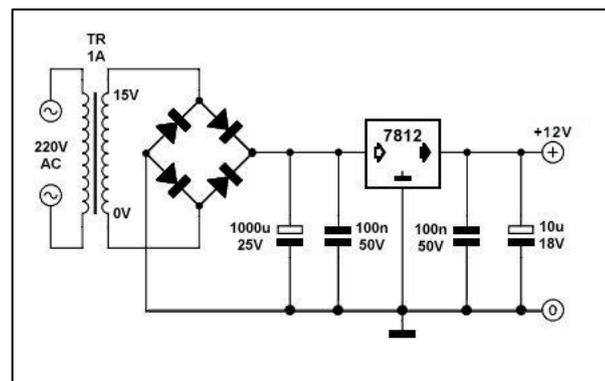
2.11 Adaptor 12V

Sebuah adaptor pada prinsipnya adalah sebuah *powersupply* yang telah disesuaikan voltasenya dengan peralatan yang akan di *supply*. Sebuah alat yang beroperasi pada voltase 12v (Volt) harus memiliki sebuah alat yang disebut dengan adaptor untuk dapat merubah voltase 220 VAC dari PLN menjadi 12 VDC. (Kho, 2015).



Gambar 2.16 Adaptor 12V

Sumber : www.tecnoamel.com,ve



Gambar 2.17 Skematik Adaptor

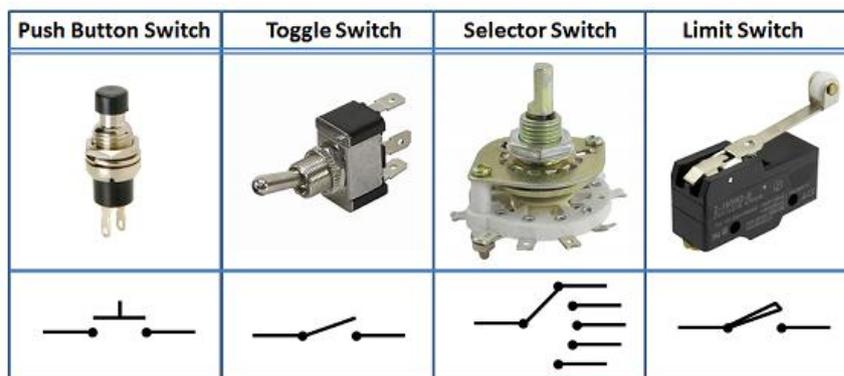
Sumber : www.tecnoamel.com,ve

2.12 Saklar (*Switch*)

Saklar atau dalam bahasa Inggris disebut *Switch* adalah salah satu komponen yang penting dalam setiap rangkaian atau perangkat elektronik. Saklar pada dasarnya merupakan perangkat mekanik yang terdiri dari dua atau lebih terminal yang terhubung secara internal ke bilah atau kontak logam yang dapat dibuka dan ditutup oleh penggunaannya. Aliran listrik akan mengalir apabila suatu kontak dihubungkan dengan kontak lainnya. Sebaliknya, aliran listrik akan terputus apabila hubungan tersebut dibuka atau dipisahkan. Selain sebagai komponen untuk menghidupkan (ON) dan mematikan (OFF) perangkat elektronik, Saklar sering juga difungsikan sebagai pengendali untuk mengaktifkan fitur-fitur tertentu pada suatu rangkaian listrik.

Berikut ini adalah jenis-jenis Saklar listrik mekanik yang digolongkan berdasarkan cara gerakan saklarnya.

1. Push Button Switch (Saklar Tombol Dorong)
2. Toggle Switch (Saklar Pengalih)
3. Selector Switch (Saklar Pemilih)
4. Limit Switch (Saklar Pembatas). (Kho, 2015)



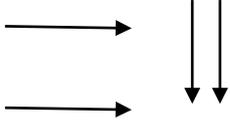
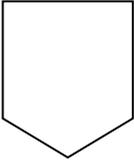
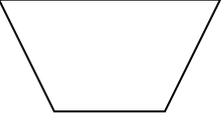
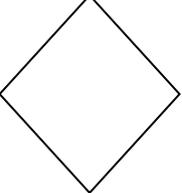
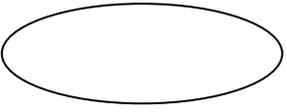
Gambar 2.18 Saklar

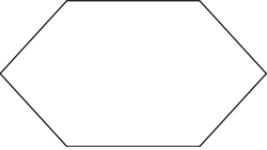
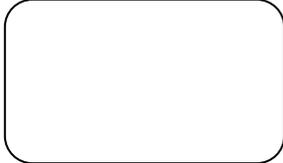
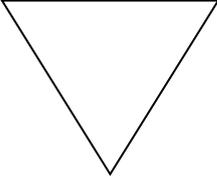
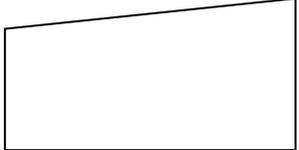
Sumber: www.teknikelektronik.com

2.13 Flowchart

Flowchart adalah langkah-langkah dan urutan prosedur program yang mempermudah penyelesaian masalah. Hal pertama dalam melakukan perancangan *software* adalah membuat alur program (*flowchart*) dari program yang akan dibuat. Dengan adanya *flowchart* maka dapat dipahami arah dari jalannya program. (Setiawan, 2013). Berikut pada Tabel 2.2 adalah simbol dan fungsi *flowchart*.

Tabel 2.2. Simbol-Simbol *Flowchart*

NO	SIMBOL	KETERANGAN
1		Simbol arus / <i>flow</i> , yaitu menyatakan jalannya arus suatu proses
2		Simbol <i>connector</i> , menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama
3		Simbol <i>offline connector</i> , menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda
4		Simbol proses, yaitu menyatakan suatu tindakan (proses) yang dilakukan oleh computer
5		Simbol <i>manual</i> , menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer
6		Simbol <i>decision</i> , yaitu menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban : ya / tidak
7		Simbol <i>terminal</i> , yaitu menyatakan permulaan atau akhir suatu program

8		<p>Simbol <i>predefined process</i>, menyatakan persediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal</p>
9		<p>Simbol <i>keying operation</i>, menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai keyboard</p>
10		<p>Simbol <i>offline-storage</i>, menunjukkan bahwa data dalam symbol ini akan disimpan ke dalam suatu media tertentu</p>
11		<p>Simbol <i>manual input</i>, menyatakan data secara manual dengan menggunakan online keyboard</p>
12		<p>Simbol <i>input / output</i>, menyatakan proses input atau output tanpa tergantung jenis peralatannya</p>