

## BAB II

### TINJAUAN PUSTAKA

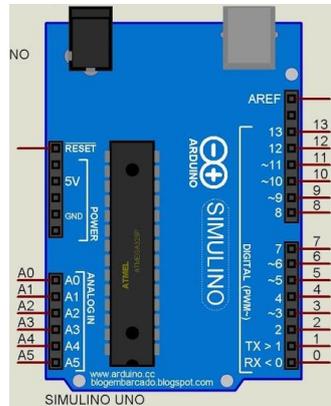
#### 2.1 Modul Mikrokontroler ATmega328P Arduino Uno

Modul Mikrokontroler ATmega328P Arduino Uno adalah sebuah *board* mikrokontroler yang menyatukan berbagai macam komponen seperti resistor, kapasitor, transistor, *Crystal*, IC (*Integrated Circuit*), dan di desain sedemikian rupa sehingga lebih mudah digunakan. Modul ini memiliki total 20 pin *input/output* yaitu 14 pin *input/output* digital dimana 6 pin *input* tersebut dapat digunakan sebagai *output* PWM dan 6 pin *input* analog, 16 MHz osilator kristal, koneksi USB, *jack power*, *ICSP (In Circuit Serial Programming) header*, dan tombol reset. Arduino Uno menggunakan fitur ATmega16U2 yang diprogram sebagai konverter *USB-to-serial* sehingga dapat langsung dihubungkan dengan komputer. Tabel 2.1 menjelaskan spesifikasi yang ada pada Modul Mikrokontroler Atmega328P Arduino Uno.

Tabel 2.1 Spesifikasi Modul Mikrokontroler ATmega328P Arduino Uno

Mikrokontroler	ATmega328P
Tegangan Kerja	5V
Tegangan <i>Input</i> (disarankan)	7-12V
Tegangan <i>Input</i> (batas)	6-20V
Pin I/O Digital	14 (6 pin diantaranya adalah <i>output</i> PWM)
Pin <i>Input</i> Analog	6
Arus DC tiap pin I/O	20 mA
Arus DC untuk pin 3.3V	50 mA
Flash Memory	32 KB (ATmega328P) yang mana 0.5 KB digunakan oleh <i>bootloader</i>
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
<i>Clock Speed</i>	16 MHz

Gambar 2.1 merupakan gambar dari modul mikrokontroler ATmega328P Arduino Uno.



Gambar 2.1 Modul ATmega328P Arduino Uno

(Sumber : <http://www.leselektronika.com/2014/10/library-arduino-dan-sensor-untuk.html>)

### 2.1.1 ATmega328P

ATmega328P adalah mikrokontroler keluaran dari atmel yang mempunyai arsitektur RISC (*Reduce Instruction Set Computer*) yang dimana setiap proses eksekusi data lebih cepat dari pada arsitektur CISC (*Completed Instruction Set Computer*).

Mikrokontroler ini memiliki beberapa fitur antara lain :

- 130 macam instruksi yang hampir semuanya dieksekusi dalam satu siklus *clock*.
- 32 x 8-bit register serba guna.
- Kecepatan mencapai 16 MIPS dengan clock 16 MHz.
- 32 KB *Flash memory* dan pada arduino memiliki *bootloader* yang menggunakan 2 KB dari flash memori sebagai *bootloader*.
- Memiliki *EEPROM (Electrically Erasable Programmable Read Only Memory)* sebesar 1KB sebagai tempat penyimpanan data semi permanent karena *EEPROM* tetap dapat menyimpan data meskipun catu daya dimatikan.
- Memiliki *SRAM (Static Random Access Memory)* sebesar 2KB.

- Memiliki pin I/O digital sebanyak 14 pin 6 diantaranya *PWM* (*Pulse Width Modulation*) output.
- *Master / Slave SPI Serial interface*.

Mikrokontroler ATmega328P memiliki arsitektur Harvard, yaitu memisahkan memori untuk kode program dan memori untuk data sehingga dapat memaksimalkan kerja dan *parallelism*. Instruksi – instruksi dalam memori program dieksekusi dalam satu alur tunggal, dimana pada saat satu instruksi dikerjakan instruksi berikutnya sudah diambil dari memori program. Konsep inilah yang memungkinkan instruksi – instruksi dapat dieksekusi dalam setiap satu siklus clock. 32 x 8-bit register serba guna digunakan untuk mendukung operasi pada ALU (*Arithmetic Logic unit*) yang dapat dilakukan dalam satu siklus. 6 dari register serbaguna ini dapat digunakan sebagai 3 buah register pointer 16-bit pada mode pengalamatan tidak langsung untuk mengambil data pada ruang memori data.

Ketiga register pointer 16-bit ini disebut dengan register X (gabungan R26 dan R27), register Y (gabungan R28 dan R29), dan register Z (gabungan R30 dan R31). Hampir semua instruksi AVR memiliki format 16-bit. Setiap alamat memori program terdiri dari instruksi 16-bit atau 32-bit.

Selain register serba guna di atas, terdapat register lain yang terpetakan dengan teknik *memory mapped* I/O selebar 64 byte. Beberapa register ini digunakan untuk fungsi khusus antara lain sebagai register control Timer/ Counter, Interupsi, ADC, USART, SPI, EEPROM, dan fungsi I/O lainnya. Register – register ini menempati memori pada alamat 0x20h – 0x5Fh. (Nurjannah, 2014)

### **2.1.2 *Input/Output Digital dan Input Analog Pada Modul Mikrokontroler***

Setiap 14 pin digital pada modul mikrokontroler ini dapat digunakan sebagai *input* ataupun *output*, dan pin analog hanya sebagai *input*. *Input/output* dioperasikan pada 5 volt. Setiap pin dapat menghasilkan atau menerima maximum 40 mA dan memiliki internal pull-up resistor 20-50K Ohm. Beberapa pin memiliki fungsi sebagai berikut :

### 1. **RX dan TX**

RX dan TX menggunakan protokol yang diimplementasikan dalam sebuah perangkat bernama UART (*Universal Asynchronous Receiver / Transmitter*). RX adalah jalur penerimaan data (perpindahan data) dari satu komputer ke komputer lain. RX biasa disebut *receiver*, yang berguna menangkap data yang dikirim oleh *transmitter* (TX). TX disebut *transmitter* yang berfungsi untuk mengirim data/mengeluarkan data, atau merupakan jalan yang dilalui dalam mengirim data antar perangkat. data akan dikirim melalui TX (*transmitter*) dan di ujung lainnya data akan diterima melalui RX (*receiver*). Dalam arduino uno terdapat pin 0 sebagai pin RX dan 1 sebagai TX yang digunakan untuk menerima (RX) dan mengirim (TX) TTL data serial. Pin ini terhubung pada pin yang koresponding dari USB ke TTL *chip serial*.

### 2. **Interupsi Eksternal**

Interupsi adalah suatu permintaan khusus untuk melakukan sesuatu, lalu akan menghentikan dahulu apa yang sedang dikerjakan, dan baru di lanjutkan setelah selesai. Banyak yang bisa menjadi penyebab interupsi seperti timer, *hardware error*, I/O, program dan lainnya. Tujuan interupsi yaitu agar pengekseskusion program pada mikrokontroler berjalan lancar serta efisien. Dalam arduino uno pin interupsi eksternal terdapat pada pin 2 dan 3. Pin ini dapat dikonfigurasi untuk trigger sebuah interupsi pada *low value*, *rising* atau *falling edge*, dan perubahan nilai.

### 3. **PWM**

PWM adalah singkatan dari Pulse Width Modulation yang merupakan suatu metode untuk mendapatkan bentuk sinyal analog dari sinyal digital. Dengan adanya fungsi ini dapat membuat transisi antar *state* hidup dan mati menjadi lebih halus. PWM pada arduino bekerja pada frekuensi 500Hz, artinya 500 siklus/ketukan dalam satu detik. Untuk setiap siklus, dapat diberikan nilai dari 0 hingga 255. Ketika diberikan angka 0, berarti pada pin tersebut akan pernah bernilai 0 volt atau

setara dengan GND. Sedangkan jika diberikan nilai 255, maka sepanjang siklus akan bernilai 5 volt. Pin yang bisa dimanfaatkan untuk PWM pada Arduino Uno adalah pin yang bertanda tilde (~), yaitu pin 3, 5, 6, 9, 10, dan pin 11.

### 2.1.3 SPI (*Serial Peripheral Interface*)

*Serial Peripheral Interface* ( SPI ) merupakan salah satu mode komunikasi *serial synchronous* kecepatan tinggi yang dimiliki oleh ATmega328P. Komunikasi SPI membutuhkan 3 jalur yaitu MOSI, MISO, dan SCK. Melalui komunikasi ini data dapat saling dikirimkan baik antara mikrokontroler maupun antara mikrokontroler dengan peripheral lain di luar mikrokontroler. Jalur utama dari SPI arduino uno adalah sebagai berikut :

1. MOSI : *Master Output Slave Input* Artinya jika dikonfigurasi sebagai master maka pin MOSI sebagai output tetapi jika dikonfigurasi sebagai slave maka pin MOSI sebagai input.
2. MISO : *Master Input Slave Output* Artinya jika dikonfigurasi sebagai master maka pin MISO sebagai input tetapi jika dikonfigurasi sebagai slave maka pin MISO sebagai output.
3. SCK : *Serial Clock*, mensinkronisasi transmisi data yang dihasilkan oleh master
4. SS : *Select Slave*, pin pada setiap perangkat yang dapat digunakan master untuk mengaktifkan dan menonaktifkan perangkat tertentu

Dalam arduino uno SPI terdapat pada pin: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Pin ini mendukung komunikasi SPI, yang mana masih mendukung hardware, yang tidak termasuk pada bahasa arduino. MISO (*Master In Slave Out*) & MOSI (*Master Out Slave In*) adalah jalur data untuk komunikasi antara *master* (*programmer/downloader, USBAsp*) dan *slave* (IC mikrokontoler). Sesuai dengan namanya, MISO merupakan jalur yang digunakan *downloader* untuk menerima data, sedangkan MOSI adalah jalur yang digunakan *downloader* untuk mengirim data ke IC mikrokontroller. Kedua jalur ini adalah jalur utama yang digunakan *downloader* dan mikrokontroler berkomunikasi. Untuk menghindari kesalahan

dalam berkomunikasi, maka dibutuhkan sinkronisasi. Sinkronisasi tersebut dilakukan dengan memanfaatkan jalur SCK (atau ada yang disebut SCLK, *Serial Clock*). Data (MISO dan atau MOSI) akan dianggap valid hanya saat SCK dalam keadaan tinggi. (DP Pertiwi, 2015)

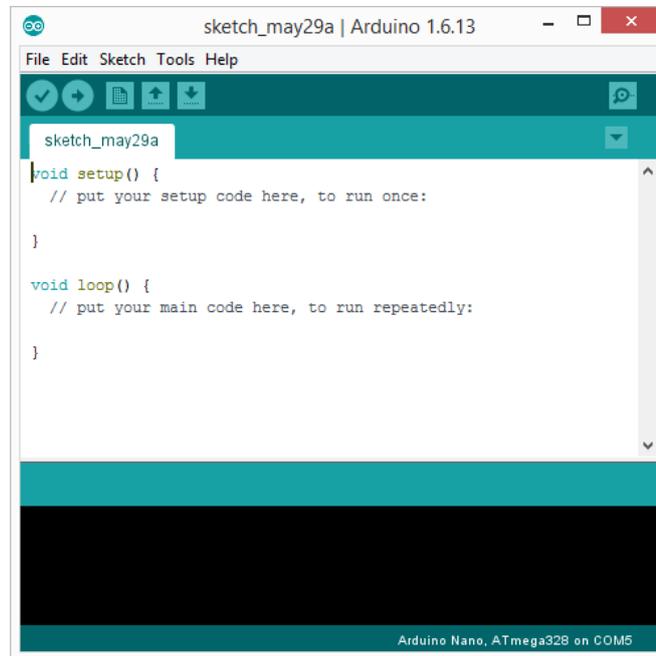
#### **2.1.4 *Integrated Development Environment (IDE)***

Arduino Uno dapat diprogram dengan perangkat lunak Arduino. Pada ATmega328P di Arduino terdapat *bootloader* yang memungkinkan untuk meng-*upload* kode baru untuk itu tanpa menggunakan *programmer hardware* eksternal. *Integrated Development Environment (IDE)* Arduino terdiri dari *text editor* untuk menulis kode, sebuah area pesan, sebuah konsol, sebuah *toolbar* dengan tombol-tombol untuk fungsi yang umum dan beberapa menu. *Integrated Development Environment (IDE)* Arduino terhubung ke arduino board untuk meng-*upload* program dan juga untuk berkomunikasi dengan *board* arduino.

Perangkat lunak (*software*) yang ditulis menggunakan *Integrated Development Environment (IDE)* Arduino disebut *sketch*. *Sketch* ditulis pada editor teks. *Sketch* disimpan dengan file berekstensi.area pesan memberikan informasi dan pesan *error* ketika kita menyimpan atau membuka *sketch*. Konsol menampilkan *output teks* dari *Integrated Development Environment (IDE)* Arduino dan juga menampilkan pesan *error* ketika kita mengkompilasi *sketch*. Pada sudut kanan bawah jendela *Integrated Development Environment (IDE)* Arduino menunjukkan jenis *board* dan port serial yang sedang digunakan. Tombol *toolbar* digunakan untuk memeriksa dan meng-*upload sketch*, membuat, membuka, atau menyimpan *sketch*, dan menampilkan serial monitor.

1. *Editor* program, sebuah *window* yang memungkinkan pengguna menulis dan mengedit program.
2. *Compiler*, sebuah modul yang mengubah kode program menjadi kode biner. Bagaimanapun sebuah mikrokontroler tidak akan bisa memahami bahasa program. Yang bisa dipahami oleh mikrokontroler adalah kode biner. Itulah sebabnya compiler diperlukan dalam hal ini.

3. *Uploader*, sebuah modul yang memuat kode biner dari komputer ke dalam memori di dalam papan Arduino.



Gambar 2.2 *Integrated Development Environment (IDE) Arduino*

Struktur dasar dalam pemrograman arduino terdiri atas dua bagian, yaitu fungsi persiapan (*setup()*) dan fungsi utama(*loop()*). Fungsi *setup()* digunakan untuk mendefinisikan variabel-variabel yang digunakan dalam program, sedangkan *loop()* adalah program inti/utama dari arduino yang dijalankan secara terus menerus. (Heri Andrianto & Aan Darmawan, 2016: 45)

Berikut adalah fungsi-fungsi dasar dalam bahasa pemrograman C pada arduino :

1. *setup()*

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

2. *loop()*

Program yang berada dalam fungsi *loop()* akan dieksekusi secara terus menerus.

3. *Function*

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

4. {} (kurung kurawal)

Digunakan untuk mengawali dan mengakhiri sebuah fungsi, blok instruksi seperti *loop()*, *void()* dan instruksi *for* dan *if*.

5. ; (titik koma)

Digunakan sebagai tanda akhir instruksi.

6. /\*.....\*/ (blok komentar)

Digunakan pada komentar yang memiliki baris lebih dari satu. Apapun yang ditulis dalam blok komen ini tidak berpengaruh terhadap program yang dibuat dan tidak akan menghabiskan memori.

7. // (komentar baris)

Sama seperti blok komentar hanya saja digunakan untuk satu baris komentar.

8. Variabel

Adalah suatu ekspresi yang digunakan untuk mewakili suatu nilai yang digunakan dalam program. Suatu variabel akan menampung nilai sesuai definisi yang telah dibuat. Variabel hanya perlu didefinisikan satu kali saja tetapi nilainya dapat sesuai program. Terdapat dua macam variabel. Ada variabel global yang dapat digunakan oleh semua fungsi dan instruksi dalam program. Variabel ini didefinisikan pada awal program sebelum fungsi *setup()*. Dan ada variabel lokal yang mana variabel ini didefinisikan pada suatu fungsi atau dalam fungsi *loop*. Variabel ini hanya dapat dilihat dan digunakan di dalam fungsi tersebut.

Tipe-tipe data dalam variabel :

a) *Byte*

b) *Int*

c) *Long*

d) *Float*

#### 9. *Array*

*Array* adalah kumpulan nilai yang diakses dengan nomor indeks. Setiap nilai dalam *array* dapat dipanggil dengan memanggil nama *array* dan nomor tersebut.

#### 10. Aritmatika

Operator aritmatika meliputi penambahan, pengurangan, perkalian, dan pembagian.

#### 11. Operasi Gabungan

Adalah operasi matematika gabungan yang biasa digunakan dalam program.

#### 12. Operator Perbandingan

Operator untuk membandingkan 2 konstanta atau variabel yang sering digunakan untuk menguji suatu kondisi benar atau salah.

#### 13. Operator Logika

Operator logika, *AND*, *OR*, dan *NOT* sering digunakan dalam pernyataan *if*.

#### 14. Konstanta

Bahasa arduino memiliki nilai-nilai yang telah ditetapkan yang disebut konstanta. Mereka digunakan untuk membuat program lebih mudah dibaca.

#### 15. *TRUE / FALSE*

Adalah konstanta Boolean yang mendefinisikan nilai logika.

#### 16. *HIGH / LOW*

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

#### 17. *Input / Output*

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

18. *If*

Instruksi untuk menguji apakah suatu kondisi telah tercapai, seperti membandingkan nilai variabel berada diatas jumlah tertentu, dan menjalankan setiap instruksi di dalam kurung jika pernyataan tersebut benar, jika tidak maka akan dilewati.

19. *if.....else*

memungkinkan untuk mengeksekusi instruksi yang lain jika suatu kondisi tidak terpenuhi. *else* juga dapat digunakan lebih dari satu.

20. *for*

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

21. *while*

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

22. *do.....while*

Perintah untuk melakukan sesuatu secara terus menerus hingga mencapai suatu kondisi yang tidak memenuhi kondisi yang diinginkan.

23. *pinMode(pin,Mode)*

Instruksi yang digunakan pada fungsi *setup()* untuk menginisialisasi suatu pin sebagai *input* atau *output*.

24. *digitalRead(pin)*

Instruksi yang digunakan untuk membaca *input* dari suatu pin yang hasilnya berupa logika *HIGH* atau *LOW*. Pin dapat diartikan sebagai suatu variabel atau konstanta 0-13 yang mewakili *input* dan *output* dari *board* arduino.

25. *digitalWrite(pin,value)*

Instruksi untuk memberikan nilai output *HIGH* (1) atau *LOW* (0) pada pin digital

26. *analogRead(pin)*

Instruksi untuk membaca nilai *input* analog dengan resolusi 10 bit. Instruksi ini hanya berlaku untuk pin A0-A5 yang mampu membaca nilai analog. Karena beresolusi 10 bit maka hasil pembacaan digital adalah 0 sampai 1023.

27. *analogWrite(pin,value)*

Instruksi yang berfungsi untuk memberi nilai PWM (*pulse width modulation*) pada *output*. Pada arduino pin PWM ditandai dengan *tilde* (~), yaitu pin 3,5,6,9,10, dan 11.

28. *delay(ms)*

Instruksi untuk memberi jeda sebelum lanjut ke program selanjutnya.

29. *millis()*

Instruksi untuk mengambil nilai waktu sejak program dijalankan hingga program berhenti atau dimatikan.

30. *tone(pin, frekuensi, durasi)*

Instruksi untuk menghasilkan nada frekuensi dengan durasi tertentun dan dikirimkan ke pin yang dituju.

31. *noTone(pin)*

Instruksi untuk menghentikan frekuensi yang dihasilkan pada pin yang dituju.

32. *randomSeed(seed)*

Instruksi untuk mengambil nilai acak dengan seed nilai awal fungsi.

33. *random(max); random(min,max)*

Instruksi *random(max)* berfungsi mengambil nilai acak dengan max sebagai batas nilai maksimal fungsi *random()*. Instruksi *random(min,max)* berfungsi untuk mengambil nilai acak diantara nilai *min* dan *max*.

34. *Serial.begin(rate)*

Instruksi untuk membuka port data serial untuk komunikasi serial baik mengirim atau menerima data dari serial. Rate adalah *baud rate* yang digunakan untuk komunikasi serial (biasa digunakan 9600).

35. *Serial.print(data)*

Instruksi yang digunakan untuk mengirim data ke port serial.

36. *Serial.read()*

Instruksi untuk menerima data dari port serial.

37. *Serial.available()*

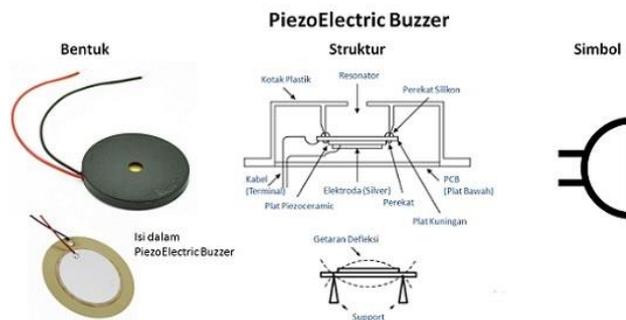
Merupakan instruksi untuk mendeteksi apakah menerima data dari port serial? Apabila menerima data, akan menghasilkan nilai >0.

## 2.2 Piezoelectric

Efek Piezoelectric (Piezoelectric Effect) pertama kali ditemukan oleh dua orang fisikawan Perancis yang bernama Pierre Curie dan Jacques Curie pada tahun 1880. Penemuan tersebut kemudian dikembangkan oleh sebuah perusahaan Jepang menjadi Piezo Electric Buzzer dan mulai populer digunakan sejak 1970an.

### 2.2.1 Prinsip Kerja Piezoelectric

*Piezoelectric* adalah komponen yang dapat menghasilkan tegangan listrik sebagai respon dari suatu perubahan gerakan mekanik. *Piezoelectric* umumnya digunakan sebagai sensor getaran mekanik (*vibration*), tekanan (*pressure*), dan lekukan (*twist*). Gambar 2.3 adalah gambar bentuk *piezoelectric*.



Gambar 2.3 Piezoelectric Buzzer

(Sumber : <http://teknikelektronika.com/>)

Jika dibandingkan dengan Speaker, Piezo Buzzer relatif lebih mudah untuk digerakan. Sebagai contoh, Piezo Buzzer dapat digerakan hanya dengan menggunakan *output* langsung dari sebuah IC TTL, hal ini sangat berbeda dengan *speaker* yang harus menggunakan penguat khusus untuk menggerakkan *speaker* agar mendapatkan intensitas suara yang dapat didengar oleh manusia.

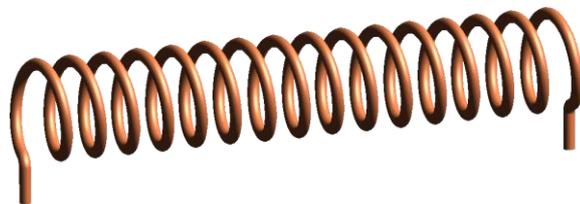
Piezo Buzzer dapat bekerja dengan baik dalam menghasilkan frekuensi di kisaran 1 – 5 kHz hingga 100 kHz untuk aplikasi *Ultrasound*. Tegangan Operasional Piezoelectric Buzzer yang umum biasanya berkisar diantara 3 Volt hingga 12 Volt.

## 2.3 Solenoid

Solenoid adalah sebuah lilitan kawat tembaga yang kemudian dililitkan dengan rapat pada sebuah inti besi untuk mengasilkan medan electromagnet.

Lilitan tersebut disebut solenoid, solenoid ini merupakan medan magnet yang sangat kuat pada inti besinya, dengan asumsi bahwa panjang lilitan tersebut lebih besar dari diameter kabel atau tembanya. Secara ideal, solenoid memiliki panjang lilitan yang tak terhingga dengan lilitan dari kabelnya yang rapat saling berhimpit satu sama lainnya. Maka akan menghasilkan medan elektromagnet yang sama dengan konstan yang bersifat parallel terhadap inti besi menjadi sumbunya.

Apabila kita alirkan listrik kepada batang besi yang kita tempatkan di tengah lilitan, maka batang besi tersebut akan mendapatkan induksi magnet dan akhirnya dapat menjadi magnet. Dengan penempatan sebagian batang besi tersebut berada di dalam solenoid dan sebagiannya lagi di sebelah luarnya. Batang besi yang terinduksi magnet tersebut akan menarik masuk benda berbahan logam ke dalam solenoid. Hal ini yang dimanfaatkan untuk menggerakkan tuas, menutup dan mengunci pintu, atau menggerakkan slot kunci pintu. Gambar 2.4 adalah ilustrasi kumparan solenoid.



Gambar 2.4 Kumparan Solenoid

(Sumber : <https://en.wikipedia.org/wiki/Solenoid>)

Inti besi yang berbentuk bulat dan kerucut itu, salah satu ujungnya memiliki kutub positif. Ketika inti besi tersebut dimasukkan ke tengah kumparan yang penuh dengan medan magnet, maka permukaan ujung yang satunya lagi memiliki kutub negative. Sementara di bagian bawahnya terdapat area yang cukup luas untuk menyalurkan aliran fluks magnet tersebut. (Adjie Suseno T, 2016).

### **2.3.1 Solenoid Door Lock**

Solenoid pengunci pintu adalah perangkat elektronik kunci pintu dengan menggunakan tegangan listrik sebagai pengendalinya. Alat ini banyak diaplikasikan pada pintu otomatis. Solenoid pengunci pintu bekerja jika diberi tegangan. Dalam keadaan normal tuas pada Solenoid pengunci pintu akan

memanjang, dan jika diberi tegangan tuas pada alat ini akan memendek. Tegangan listrik yang diberikan akan membuat medan magnet sehingga tuas pada Solenoid pengunci pintu akan tertarik oleh medan magnet. Berikut gambar 2.5 adalah bentuk solenoid *door lock*.

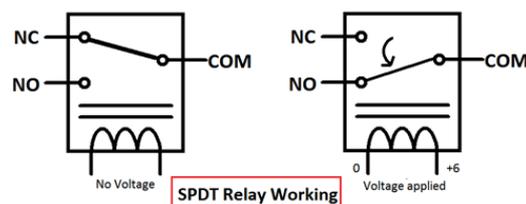


Gambar 2.5 Solenoid *Door Lock*

(Sumber : <http://www.hobbyist.co.nz/?q=solenoid-lock>)

## 2.4 Relay

Relay merupakan komponen output yang paling sering digunakan pada beberapa peralatan elektronika dan di berbagai bidang lainnya. Relay berfungsi untuk menghubungkan atau memutuskan aliran arus listrik yang dikontrol dengan memberikan tegangan dan arus tertentu pada koilnya. Ada 2 macam relay berdasarkan tegangan untuk menggerakkan koilnya, yaitu AC dan DC. Ada berbagai macam jenis relay berdasarkan *pole*-nya. Pada perancangan kali ini dipakai *Single Pole Double Throw* (SPDT) dan *Double Pole Double Throw* (DPDT) yang berfungsi untuk menghubungkan dan memutus arus untuk menggerakkan peralatan di luar rangkaian. Di bawah ini adalah gambar cara kerja relay. (R Dahlianti, 2015)



Gambar 2.6 Kerja Relay SPDT

(Sumber : <https://circuitdigest.com/sites/default/files/inlineimages/SPDT-Relay-Working.gif>)

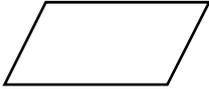
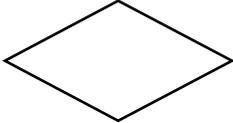
## 2.5 Flowchart

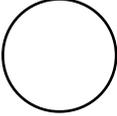
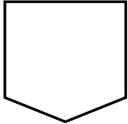
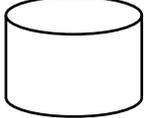
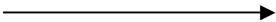
Menurut Mulyadi dalam buku Sistem Akuntansi (2015) definisi *Flowchart* yaitu : “*Flowchart* adalah bagan yang menggambarkan aliran dokumen dalam suatu sistem informasi.”.

Dengan adanya flowchart urutan poses kegiatan menjadi lebih jelas. Jika ada penambahan proses maka dapat dilakukan lebih mudah. Setelah flowchart selesai disusun, selanjutnya pemrogram (programmer) menerjemahkannya ke bentuk program dengan bahasa pemrograman.

. Flowchart mempunyai banyak simbol. Pada tabel 2.2 beberapa simbol yang sering digunakan untuk membuat diagram alir program diantaranya :

Tabel 2.2 Simbol-simbol *Flowchart*

SIMBOL	KETERANGAN
	Terminal, untuk memulai atau mengakhiri sebuah program
	Input / output, menunjukkan operasi pembacaan input atau pencetakan output.
	Keputusan ( Decision ), menunjukkan suatu seleksi yang harus dikerjakan.
	Pengolahan ( Processing ), menunjukkan suatu pemrosesan yang harus dilakukan oleh komputer.
	Predefined Process, simbol untuk pemberian harga awal dan penyediaan tempat tertentu di dalam proses.

	<p>Input / output dokumentasi, digunakan untuk pembacaan input atau pencetakan output pada printer.</p>
	<p>Penghubung halaman yang sama, digunakan untuk menghubungkan flowchart yang terpotong pada halaman yang sama.</p>
	<p>Penghubung halaman berikutnya, digunakan untuk menghubungkan pada halaman berikutnya.</p>
	<p>Simbol disk, menunjukkan I/O menggunakan harddisk atau I/O disimpan di dalam harddisk.</p>
	<p>Simbol aliran, menunjukkan arus dari data.</p>
	<p>Manual Operation, untuk operasi manual.</p>
	<p>Input yang dimasukkan secara manual dari keyboard.</p>