

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Mikrokontroler**

Mikrokontroler (pengendali mikro) pada suatu rangkaian elektronik berfungsi sebagai pengendali yang mengatur jalannya proses kerja dari rangkaian elektronik. Didalam IC mikrokontroler terdapat CPU, memori, timer, saluran komunikasi serial dan paralel, port input/output, ADC, dan lain-lain (Andrianto, 2016).

##### **2.1.1 Mikrokontroler AVR**

AVR mempunyai kepanjangan Advanced Versatile RISC atau Alf And Vegards Risc processor yang berasal dari dua nama mahasiswa Norwegian Institute Of Technology (NTH), yaitu Alf-Egil Boken dan Vegard Wollan. Secara umum mikrokontroler AVR dapat dikelompokkan menjadi lima kelompok yaitu keluarga Attiny, AT90Sxx, ATmega, AVR Xmega, dan AVR32 UC3 (Andrianto, 2016).

##### **2.1.2 Fitur AVR ATmega328**

ATmega328 adalah mikrokontroller 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*).

Mikrokontroller 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.

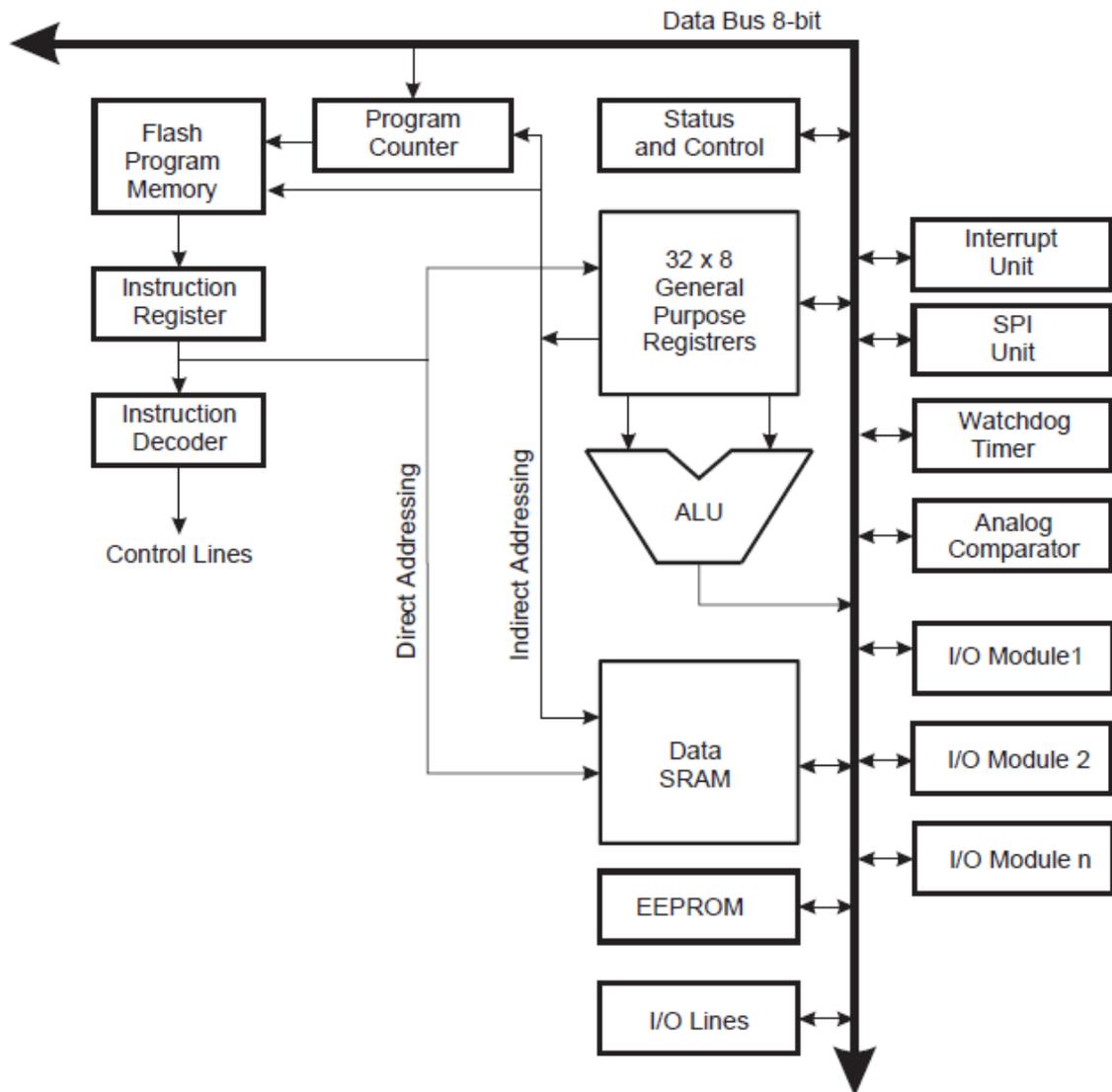
- Memiliki *SRAM (Static Random Access Memory)* sebesar 2KB.
- Memiliki pin I/O digital sebanyak 14 pin 6 diantaranya *PWM (Pulse Width Modulation)* output.
- *Master / Slave SPI Serial interface.*

Mikrokontroler ATmega 328 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.

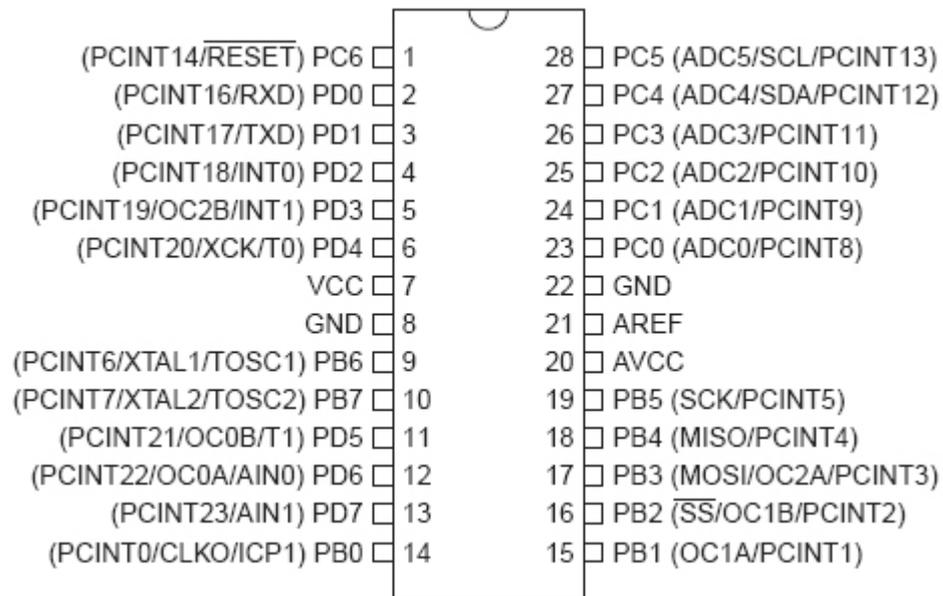
Tampilan architecture ATmega 328, seperti gambar 2.1.



**Gambar 2.1** Architecture ATmega328

### 2.1.3 Konfigurasi PIN ATmega328

Tapilan *datasheet* Atmega328 seperti gambar 2.2.



**Gambar 2.2** Konfigurasi Pin ATmega328

Mikrokontroler Atmega328 memiliki pin berjumlah 28. Masing-masing pin mikrokontroler Atmega328 mempunyai kegunaan seperti yang ditunjukkan pada tabel 2.1 , tabel 2.2 dan table 2.3.

Table 2.1 Konfigurasi Port B

Port Pin	Alternate Functions
PB7	XTAL2 (Chip Clock Oscillator pin 2) TOSC2 (Timer Oscillator pin 2) PCINT7 (Pin Change Interrupt 7)
PB6	XTAL1 (Chip Clock Oscillator pin 1 or External clock input) TOSC1 (Timer Oscillator pin 1) PCINT6 (Pin Change Interrupt 6)
PB5	SCK (SPI Bus Master clock Input) PCINT5 (Pin Change Interrupt 5)
PB4	MISO (SPI Bus Master Input/Slave Output) PCINT4 (Pin Change Interrupt 4)
PB3	MOSI (SPI Bus Master Output/Slave Input) OC2A (Timer/Counter2 Output Compare Match A Output) PCINT3 (Pin Change Interrupt 3)
PB2	$\overline{SS}$ (SPI Bus Master Slave select) OC1B (Timer/Counter1 Output Compare Match B Output) PCINT2 (Pin Change Interrupt 2)
PB1	OC1A (Timer/Counter1 Output Compare Match A Output) PCINT1 (Pin Change Interrupt 1)
PB0	ICP1 (Timer/Counter1 Input Capture Input) CLKO (Divided System Clock Output) PCINT0 (Pin Change Interrupt 0)

Table 2.2 Konfigurasi Port C

Port Pin	Alternate Function
PC6	$\overline{RESET}$ (Reset pin) PCINT14 (Pin Change Interrupt 14)
PC5	ADC5 (ADC Input Channel 5) SCL (2-wire Serial Bus Clock Line) PCINT13 (Pin Change Interrupt 13)
PC4	ADC4 (ADC Input Channel 4) SDA (2-wire Serial Bus Data Input/Output Line) PCINT12 (Pin Change Interrupt 12)
PC3	ADC3 (ADC Input Channel 3) PCINT11 (Pin Change Interrupt 11)
PC2	ADC2 (ADC Input Channel 2) PCINT10 (Pin Change Interrupt 10)
PC1	ADC1 (ADC Input Channel 1) PCINT9 (Pin Change Interrupt 9)
PC0	ADC0 (ADC Input Channel 0) PCINT8 (Pin Change Interrupt 8)

Tabel 2.3 Konfigurasi Port D

Port Pin	Alternate Function
PD7	AIN1 (Analog Comparator Negative Input) PCINT23 (Pin Change Interrupt 23)
PD6	AIN0 (Analog Comparator Positive Input) OC0A (Timer/Counter0 Output Compare Match A Output) PCINT22 (Pin Change Interrupt 22)
PD5	T1 (Timer/Counter 1 External Counter Input) OC0B (Timer/Counter0 Output Compare Match B Output) PCINT21 (Pin Change Interrupt 21)
PD4	XCK (USART External Clock Input/Output) T0 (Timer/Counter 0 External Counter Input) PCINT20 (Pin Change Interrupt 20)
PD3	INT1 (External Interrupt 1 Input) OC2B (Timer/Counter2 Output Compare Match B Output) PCINT19 (Pin Change Interrupt 19)
PD2	INT0 (External Interrupt 0 Input) PCINT18 (Pin Change Interrupt 18)
PD1	TXD (USART Output Pin) PCINT17 (Pin Change Interrupt 17)
PD0	RXD (USART Input Pin) PCINT16 (Pin Change Interrupt 16)

## 2.2 Arduino

### 2.2.1 Sejarah Arduino

Arduino dikembangkan dari thesis hernando barragan pada tahun 2004, seorang mahasiswa asal Kolombia. Judul tesisnya yaitu “Arduino-Revolusi Open Hardware”. Arduino diawali di runag kelas *Interactive Design Institute* di Ivrea (IDII), pada tahun 2005 di Ivrea, Italia. Arduino ditemukan oleh Massimo Banzi dan David Cuartielles dengan tujuan awal yaitu untuk membantu para siswa membuat perangkat desain daninteraksi dengan harga yang murah dibandingkan dengan perangkat lain yang tersedia pada saat itu, seperti BASIC Stamp yang harganya cukup mahal bagi pelajar pada saat itu. Arduino berasal dari bahasa italia yang berarti teman yang berani. Pada bulan Mei 2011, Arduino sudah terjual lebih dari 300.000 unit. Arduino saat ini sudah menjadi salah satu platform OSHW (Open Source Hadrware). ( Andrianto, 2016)

### 2.2.2 Pengertian Arduino

Arduino adalah sebuah board mikrokontroler yang bersifat *open source*, dimana desain skematik dan PCB bersifat *open source*, sehingga kita dapat menggunakan maupun modifikasi, Board Arduino menggunakan *Chip/IC* mikrokontroler Atmel AVR, misalnya: Arduino NG or Older w/Atmega8 (Severino), Arduino Duemilanove or Nano w/Atmega328, Arduino Uno, Arduino Mega2560, dll. (Andrianto, 2016)

*Software* untuk membuat, mengkompilasi dan meng-*upload* program yaitu Arduino IDE atau disebut juga Arduino *Software* yang juga bersifat *open source*. *Software* ini dapat diunduh pada situs <http://www.arduino.cc>. Arduino IDE (Arduino *Software*) menghasilkan *file hex* dari baris kode instruksi program yang menggunakan bahasa C yang dinamakan *sketch* setelah dilakukan compile dengan perintah Verify/Compile.

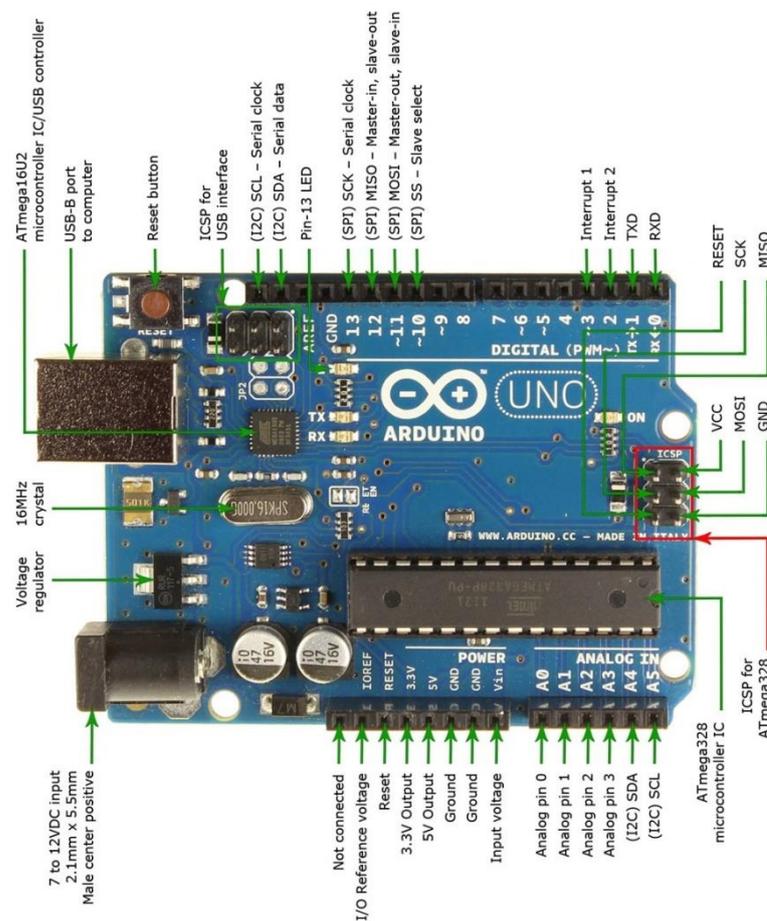
*Bootloader* Chip/IC pada Arduino *Board* telah diisi oleh program yang dinamakan Arduino **bootloader**, yang memungkinkan kita meng-*upload code* program tanpa menggunakan hardware tambahan (tanpa menggunakan programmer dari luar, seperti: AVR-ISP, STK500, *parallel programmer*, *usb programmer*). *Bootloader* akan aktif selama beberapa detik ketika board mengalami reset.

Hasil kompilasi dari Arduino *Software* dapat dipergunakan dan dijalankan tidak hanya pada arduino *board* tetapi juga dapat dijalankan di sistem mikrokontroler avr yang sesuai bahkan tanpa *bootloader*. Jika kita tidak menggunakan *bootloader*, berarti semakin besar program yang dapat dimasukkan ke *flash* memori mikrokontroler, Karen *flash* memori hanya digunakan untuk program aplikasi kita, selain itu kita dapat menghindari *delay* ketika *board* mengalami reset yang diakibatkan oleh karena menjalankan program yang ada pada *bootloader*. Namun, untuk memasukan program atau melakukan *burn sketches*, kita harus menggunakan *programmer* external seperti: AVR-ISP, STK500, *parallel programmer*, *usb programmer* (USBasp). (Andrianto, 2016)

### 2.2.3 Board Arduino UNO

*Board* Arduino Uno menggunakan mikrokontroler ATmega328. Secara umum posisi/letak pin-pin terminal I/O pada berbagai *Board* Arduino posisinya sama dengan posisi/letak pin-pin terminal I/O dari Arduino Uno yang mempunyai 14 pin Digital yang dapat di set sebagai *Input/Output* (beberapa diantaranya mempunyai fungsi ganda), 6 pin *Input* analog. (Ardianto, 2016)

Pada bagian ini akan dijelaskan fungsi dari pin dan terminal pada *Board* Arduino Uno. *Board* Arduino Uno dapat dilihat pada gambar 2.3.



Gambar 2.3 Board Arduino UNO

#### USB to Computer

Digunakan untuk koneksi ke komputer atau alat lain menggunakan komunikasi serial RS-232 standard. Bekerja ketika JP0 dalam posisi 2-3.

**DC1, 2.1mm power jack**

Digunakan sebagai sumber teggangan (catu daya) dari luar, sudah terdapat regulator tegangan yang dapat meregulasi masukan tegangan antara +7V sampai +12V (masukan tegangan ayng disarankan antara +9V sampai dengan +12V). Pin Vin dan 5V dapat digunakan sebagai sumber ketika diberi sumber tagangan dari luar.

**ICSP, 2x3 pin header**

Untuk memprogram *bootloder* ATmega atau memprogram Arduino dengan *software* lain, berikut ini keterangan fungsi tiap pin pada tabel 2.4.

Tabel 2.4 Keterangan pin ICSP pada Arduino Uno

1	MISO	+5V	2
3	SCK	MOSI	4
5	RST	GND	6

**JP0, 3 pin jumper**

Ketika posisi 2-3, *board* pada keadaan serial *enable* (*X1 connector* dapat digunakan). Ketika posisi 1-2 board pada keadaan serial *disable* (*X1 connector* tidak berfungsi) dan eksternal pull-down resistor pada pin0 (Rx) dan pin1 (Tx) dalam keadaan aktif, resistor pull-down untuk mencegah noise dari Rx.

**JP4**

Ketika pada posisi 1-2, *board* dapat mengaktifkan fungsi *auto-reset*, yang befungsi ketika meng-*upload* program pada *board* tanpa perlu menekan tombol reset S1.

**S1**

Adalah *push button* yang berfungsi sebagai tombol *reset*.

**LED**

POWER led :menyala ketika arduino dinyalakan dengan diberi tengan dari DC1

Rx led :berkedip ketika menerima data melalui komputer lewat komunikasi serial

Tx led :berkedip ketika mengirim data melalui komunikasi serial

L led :terhubung dengan digital pin13. Berkedip ketika *bootloading*

## **DIGITAL PINOUT IN/OUT**

13 digital pin inputs/outputs : pin 0-7 (terhubung pada PORT D dari ATmega). Pin0 (Rx) dan Pin1 (Tx) dapat digunakan sebagai pin komunikasi. Untuk Atmega 168/328 pin 3,5 dan 6 dapat digunakan sebagai output PWM. Enam (6) pin inputs/outputs digital: pin 8-13 (terhubung pada PORT B). Pin10 (SS), pin11 (MOSI), pin12 (MISO), pin13 (SCK) yang bisa digunakan sebagai SPI (Serial Peripheral Interface). Pin 9,10,da 11 dapat digunakan sebagai output PWM untuk ATmega 8 dan ATmega 168/328.

## **ANALOG PINOUT INPUT**

Enam (6) analog input analog: pin 0-5 (A0-A5) (terhubung pada PORT C). pin4 (SDA) dan pin5 (SCL) yang dapat digunakan sebagai I2C (two-wire serial bus). Pin analog ini dapat digunakan sebagai pin digital14 (A0) sampai pin pin19 (A5).

## **2.3 Radio Frequency Identification (RFID)**

### **2.3.1 Pengertian Radio Frequency Identification (RFID)**

RFID adalah proses identifikasi seseorang atau objek dengan menggunakan frekuensi transmisi radio. RFID menggunakan frekuensi radio untuk membaca informasi dari sebuah device kecil yang disebut tag atau transponder (Transmitter + Responder). Tag RFID akan mengenali diri sendiri ketika mendeteksi sinyal dari device yang kompatibel, yaitu pembaca RFID (RFID Reader).

RFID adalah teknologi identifikasi yang fleksibel, mudah digunakan, dan sangat cocok untuk operasi otomatis. RFID mengkombinasikan keunggulan yang tidak tersedia pada teknologi identifikasi yang lain. RFID dapat disediakan dalam device yang hanya dapat dibaca saja (Read Only) atau dapat dibaca dan

ditulis (Read/Write), tidak memerlukan kontak langsung maupun jalur cahaya untuk dapat beroperasi, dapat berfungsi pada berbagai variasi kondisi lingkungan, dan menyediakan tingkat integritas data yang tinggi. Sebagai tambahan, karena teknologi ini sulit untuk dipalsukan, maka RFID dapat menyediakan tingkat keamanan yang tinggi.

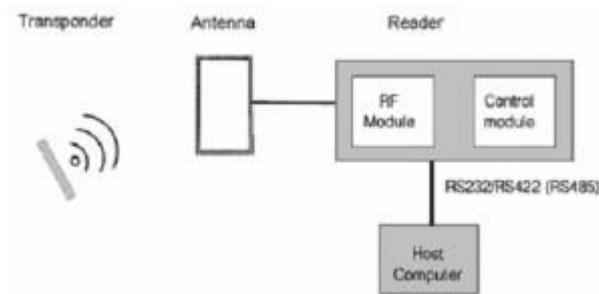
Pada sistem RFID umumnya, tag atau transponder ditempelkan pada suatu objek. Setiap tag membawa dapat membawa informasi yang unik, di antaranya: serial number, model, warna, tempat perakitan, dan data lain dari objek tersebut. Ketika tag ini melalui medan yang dihasilkan oleh pembaca RFID yang kompatibel, tag akan mentransmisikan informasi yang ada pada tag kepada pembaca RFID, sehingga proses identifikasi objek dapat dilakukan.

Sistem RFID terdiri dari empat komponen, di antaranya seperti dapat dilihat pada gambar berikut :

- Tag: Ini adalah device yang menyimpan informasi untuk identifikasi objek. Tag RFID sering juga disebut sebagai transponder.
- Antena: untuk mentransmisikan sinyal frekuensi radio antara pembaca RFID dengan tag RFID. Pembaca RFID: adalah device yang kompatibel dengan tag RFID yang akan berkomunikasi secara wireless dengan tag.
- Software Aplikasi: adalah aplikasi pada sebuah workstation atau PC yang dapat membaca data dari tag melalui pembaca RFID. Baik tag dan pembaca RFID dilengkapi dengan antena sehingga dapat menerima dan memancarkan gelombang elektromagnetik.

### **2.3.2 Sistem RFID**

Berikut ini ada sistem RFID seperti gambar 2.4.



**Gambar 2.4** Sistem RFID

### 2.3.3 Pembaca RFID

Sebuah pembaca RFID harus menyelesaikan dua buah tugas, yaitu:

- Menerima perintah dari software aplikasi
- Berkomunikasi dengan tag RFID

Pembaca RFID adalah merupakan penghubung antara software aplikasi dengan antenna yang akan meradiasikan gelombang radio ke tag RFID. Gelombang radio yang diemisikan oleh antenna berpropagasi pada ruangan di sekitarnya. Akibatnya data dapat berpindah secara wireless ke tag RFID yang berada berdekatan dengan antenna.

### 2.3.4 Tag RFID

Tag RFID adalah device yang dibuat dari rangkaian elektronika dan antenna yang terintegrasi di dalam rangkaian tersebut. Rangkaian elektronik dari tag RFID umumnya memiliki memori sehingga tag ini mempunyai kemampuan untuk menyimpan data. Memori pada tag secara dibagi menjadi sel-sel. Beberapa sel menyimpan data Read Only, misalnya serial number yang unik yang disimpan pada saat tag tersebut diproduksi. Sel lain pada RFID mungkin juga dapat ditulis dan dibaca secara berulang.

Berdasarkan catu daya tag, tag RFID dapat digolongkan menjadi:

- **Tag Aktif:** yaitu tag yang catu dayanya diperoleh dari batere, sehingga akan mengurangi daya yang diperlukan oleh pembaca RFID dan tag dapat

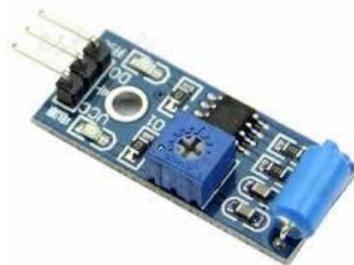
mengirimkan informasi dalam jarak yang lebih jauh. Kelemahan dari tipe tag ini adalah harganya yang mahal dan ukurannya yang lebih besar karena lebih kompleks. Semakin banyak fungsi yang dapat dilakukan oleh tag RFID maka rangkaianannya akan semakin kompleks dan ukurannya akan semakin besar.

- **Tag Pasif:** yaitu tag yang catu dayanya diperoleh dari medan yang dihasilkan oleh pembaca RFID. Rangkaianannya lebih sederhana, harganya jauh lebih murah, ukurannya kecil, dan lebih ringan. Kelemahannya adalah tag hanya dapat mengirimkan informasi dalam jarak yang dekat dan pembaca RFID harus menyediakan daya tambahan untuk tag RFID.

Tag RFID telah sering dipertimbangkan untuk digunakan sebagai barcode pada masa yang akan datang. Pembacaan informasi pada tag RFID tidak memerlukan kontak sama sekali. Karena kemampuan rangkaian terintegrasi yang modern, maka tag RFID dapat menyimpan jauh lebih banyak informasi dibandingkan dengan barcode. Fitur pembacaan jamak pada teknologi RFID sering disebut sebagai anti collision

#### 2.4 Sensor Getaran

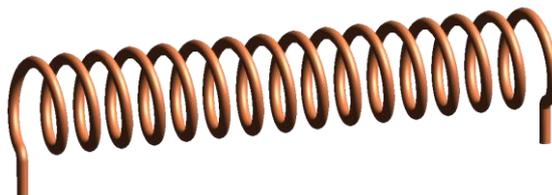
Merupakan salah satu sensor yang dapat mengukur getaran suatu benda yang nantinya dimana data tersebut akan diproses untuk kepentingan percobaan ataupun digunakan untuk mengantisipasi sebuah kemungkinan adanya mara bahaya. Salah satu jenis sensor getaran adalah sensor SW-420. Tampilan bentuk sensor getaran SW-420 seperti gambar 2.5.



**Gambar 2.5** Sensor Getaran SW-420

## 2.5 Solenoid Door Lock

Solenoid adalah sebuah lilitan kawat tembaga yang kemudian dililitkan dengan rapat pada sebuah inti besi untuk menghasilkan medan electromagnet. Lilitan tersebut seperti gambar 2.6 yang sering 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 tembaganya. Secara ideal, solenoid memiliki panjang lilitan yang tak terhingga dengan lilitan dari kabelnya yang rapat saling berhimpit satu sama lainnya. Maka akan menghasilkan medan electromagnet yang sama dengan konstan yang bersifat parallel terhadap inti besi menjadi sumbunya.



**Gambar 2.6** Kumbaran Solenoid

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. Prinsip kerja dari sebuah solenoid DC cukup mirip dengan sebuah solenoid Ac, keduanya dirancang khusus dan menghasilkan medan elektromagnet.

Inti besi yang berbentuk bulat dan kerucut itu, salah satu ujungnya memiliki kutub positif. Ketika inti besi tersebut dimasukkan ke tengah kumbaran 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).

## 2.6 Transformator ( Trafo )

Transformator atau sering disingkat dengan istilah Trafo adalah suatu alat listrik yang dapat mengubah taraf suatu tegangan AC ke taraf yang lain. Maksud dari perubahan taraf tersebut diantaranya seperti menurunkan Tegangan AC dari 220VAC ke 12 VAC ataupun menaikkan Tegangan dari 110VAC ke 220 VAC.

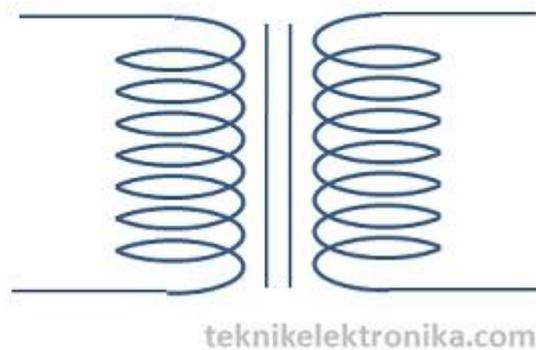
Transformator atau Trafo ini bekerja berdasarkan prinsip Induksi Elektromagnet dan hanya dapat bekerja pada tegangan yang berarus bolak balik (AC). Transformator (Trafo) memegang peranan yang sangat penting dalam pendistribusian tenaga listrik. Transformator menaikkan listrik yang berasal dari pembangkit listrik PLN hingga ratusan kilo Volt untuk di distribusikan, dan kemudian Transformator lainnya menurunkan tegangan listrik tersebut ke tegangan yang diperlukan oleh setiap rumah tangga maupun perkantoran yang pada umumnya menggunakan Tegangan AC 220Volt.

Berikut ini bentuk dan symbol trafo seperti gambar 2.7.

**Bentuk Transformator**



**Simbol Transformator**



**Gambar 2.7** Bentuk dan Simbol Trafo

## 2.7 IC Regulator

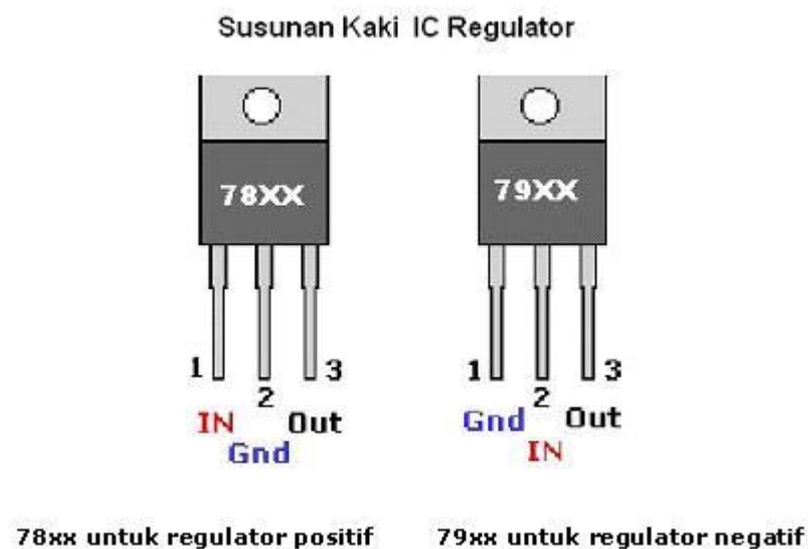
Regulator adalah suatu IC yang digunakan untuk menstabilkan tegangan agar tegangan selalu berada dalam kondisi konstan. Didalam dunia robotika caru daya sangat berperan penting karena banyak komponen contohnya IC/uC atau sebuah rangkaian sangat terganggu apabila pasokan tegangannya tidak stabil

(Tri Raharja, 2014). Ada beberapa tipe dari IC regulator ini untuk lebih jelas dapat anda lihat dalam table 2.5.

Table 2.5 Tipe IC Regulator

<b>Tipe</b>	<b>Fungsi</b>
7805	Berfungsi untuk menurunkan tegangan positif diatas 5 volt menjadi tegangan positif 5 volt
7905	Berfungsi untuk menurunkan tegangan negatif diatas 5 volt menjadi tegangan negatif 5 volt
7809	Berfungsi untuk menurunkan tegangan positif diatas 9 volt menjadi tegangan positif 9 volt
7909	Berfungsi untuk menurunkan tegangan negatif diatas 9 volt menjadi tegangan negatif 9 volt
7812	Berfungsi untuk menurunkan tegangan positif diatas 12 volt menjadi tegangan positif 12 volt
7912	Berfungsi untuk menurunkan tegangan negatif diatas 12 volt menjadi tegangan negatif 12 volt

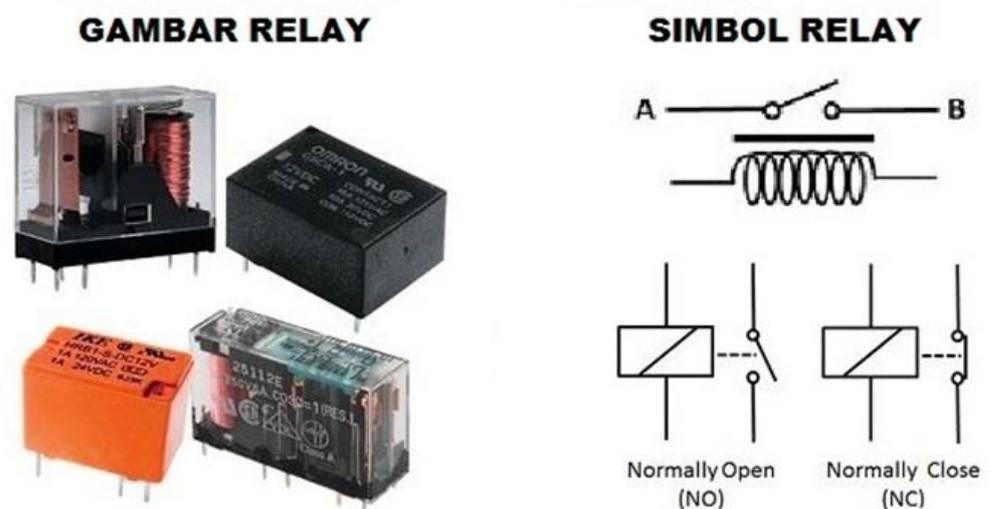
Bentuk fisik IC Regulator seperti gambar 2.8.



**Gambar 2.8** IC Regulator

## 2.8 Relay

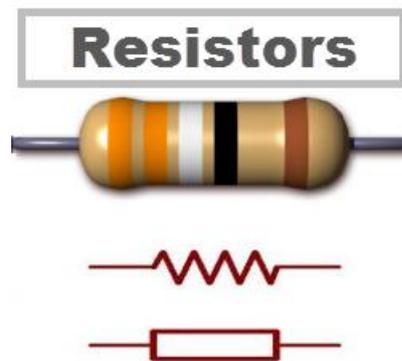
Relay merupakan komponen output yang paling sering digunakan pada beberapa peralatan elektronika dan di berbagai bidang lainnya. Relay berfungsi untuk menghubungkan atau memutuskan aliran arus listrik yang dikontrol dengan memberikan tegangan dan arus tertentu pada koilnya (Setiawan, 21:2011). Ada 2 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. Bentuk fisik relay seperti gambar 2.9.



Gambar 2.9 Relay

## 2.9 Resistor

Resistor merupakan suatu benda yang dibuat sebagai penghambat atau penahan arus listrik yang mengalir pada suatu rangkaian dengan tujuan untuk mengatur arus yang mengalir yang dinyatakan dengan satuan *ohm*. (Adjie Suseno T, 2016). Bentuk fisik resistor seperti gambar 2.10.



**Gambar 2.10** Resistor

Berikut ini merupakan tabel warna untuk menentukan besar resistor yang ditunjukkan pada tabel 2.6.

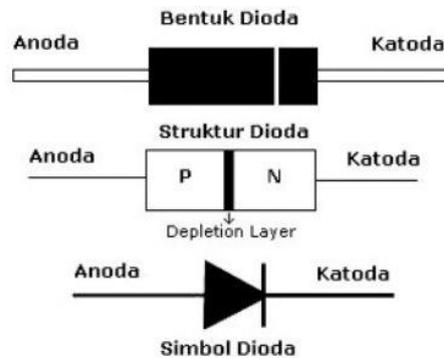
Tabel 2.6 Kode Warna Resistor

Warna Cincin	Cincin I	Cincin II	Cincin III	Cincin IV Pengali	Cincin V Toleransi
Hitam	0	0	0	$\times 1$	
Coklat	1	1	1	$\times 10^1$	$\pm 1 \%$
Merah	2	2	2	$\times 10^2$	$\pm 2 \%$
Jingga	3	3	3	$\times 10^3$	
Kuning	4	4	4	$\times 10^4$	
Hijau	5	5	5	$\times 10^5$	
Biru	6	6	6	$\times 10^6$	
Ungu	7	7	7	$\times 10^7$	
Abu-abu	8	8	8	$\times 10^8$	
Putih	9	9	9	$\times 10^9$	
Emas				$\times 0,1$	$\pm 5 \%$
Perak				$\times 0,01$	$\pm 10 \%$
Tanpa warna					$\pm 20 \%$

## 2.10 Dioda

Diode adalah piranti semikonduktor yang mengalirkan arus satu arah saja. Diode terbuat dari *germanium* atau *silicon* yang lebih dikenal dengan diode junction. Struktur dari diode ini, sesuai dengan namanya, adalah sambungan antara semikonduktor tipe P dan semikonduktor tipe N. Semikonduktor tipe P berperan sebagai anoda dan semikonduktor tipe n berperan sebagai katoda.

Dengan struktur seperti ini arus hanya dapat mengalir dari sisi P ke sisi N (Tri Raharja, 2014). Bentuk fisik diode seperti gambar 2.11.



**Gambar 2.11** Dioda

## 2.11 Kapasitor

Kapasitor adalah komponen elektronik yang berfungsi untuk menyimpan muatan listrik. Kapasitor terdiri atas dua buah keeping metal sejajar yang dipisahkan oleh isolator yang disebut dielektrik (Suyadhi, 2010). Bentuk fisik kapasitor seperti gambar 2.12..



**Gambar 2.12** Kapasitor

## 2.12 Buzzer

Buzzer adalah suatu alat yang dapat mengubah sinyal listrik menjadi sinyal suara. Pada umumnya buzzer digunakan untuk alarm karena penggunaanya cukup mudah yaitu dengan memberikan tegangan input maka buzzer akan mengeluarkan bunyi. Frekuensi suara yang dikeluarkan buzzer yaitu antara 1-5 KHz.

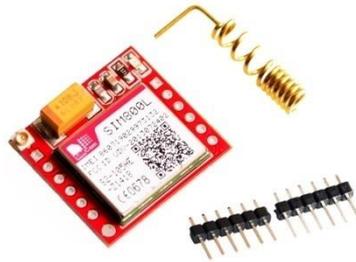
Pada dasarnya prinsip kerja buzzer hampir sama dengan loud speaker, jadi buzzer juga terdiri dari kumparan aygn terpasang pada diafragma dan kemudian kumparan tersebut dialiri arus sehingga menjadi elektromagnet, kumparan tadi akan tertarik ke dalam atau keluar, tergantung dari arah arus dan polaritas megnetnya, 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) (Suyadi, 2010). Bentuk fisik buzzer seperti gambar 2.13.



**Gambar 2.13** Buzzer

### **2.13 Module GSM (Global System Mobile)**

Modul GSM adalah peralatan yang didesain supaya dapat digunakan untuk aplikasi komunikasi dari mesin ke mesin atau dari manusia ke mesin. Modul GSM merupakan peralatan yang digunakan sebagai mesin dalam suatu aplikasi. Dalam aplikasi yang dibuat harus terdapat mikrokontroler yang akan mengirimkan perintah kepada modul GSM berupa AT command melalui RS232 sebagai komponen penghubung (communication links). Ada beberapa macam Modul GSM seperti Modem Wavecom, Sim 800L, Sim 900A, dll. Bentuk Modul GSM SIM 800L seperti pada gambar 2.14..



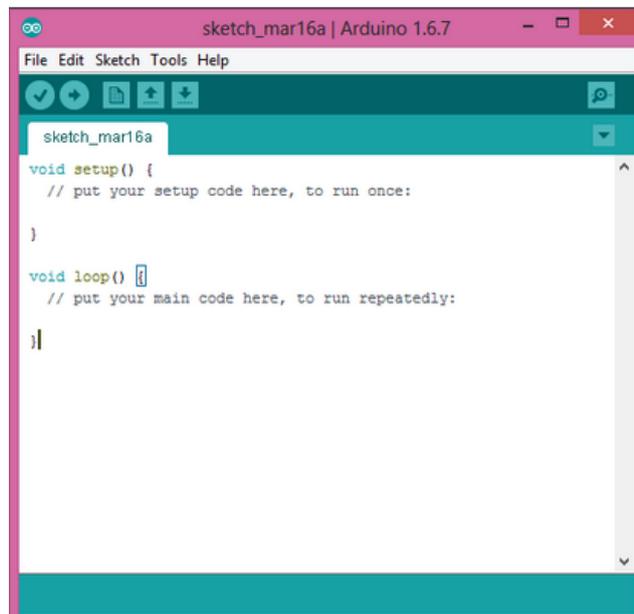
**Gambar 2.14** Modul GSM SIM 800L

Modul GSM merupakan bagian dari pusat kendali yang berfungsi sebagai transceiver. Modul GSM mempunyai fungsi yang sama dengan sebuah telepon seluler yaitu mampu melakukan fungsi pengiriman dan penerimaan SMS. Dengan adanya sebuah modul GSM maka aplikasi yang dirancang dapat dikendalikan dari jarak jauh dengan menggunakan jaringan GSM sebagai media akses.

#### **2.14 Program Arduino (IDE)**

Software IDE Arduino adalah pengendali mikro single-board yang bersifat open-source, diturunkan dari platform Wiring, dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang, hardware-nya menggunakan prosesor Atmel AVR dan software-nya memiliki bahasa pemrograman C++ yang sederhana dan fungsi-fungsinya yang lengkap, sehingga Arduino mudah dipelajari oleh pemula. Gambar 2.15 memperlihatkan tampilan awal software IDE Arduino (Andrianto, 2016).

Arduino IDE dibuat dari bahasa pemrograman JAVA. Arduino IDE juga dilengkapi dengan library C/C++ yang biasa disebut *Wiring* yang membuat operasi input dan output menjadi lebih mudah. Arduino IDE ini dikembangkan dari software Processing yang dirombak menjadi Arduino IDE khusus untuk pemrograman dengan Arduino.



**Gambar 2.15** Program Arduino (IDE)

## 2.15 Bahasa Pemrograman Arduino Berbasis C

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 (Andrianto, 2016).

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

1. *setup()*

Fungsi *setup()* dipanggil ketika program dijalankan, berfungsi untuk inialisasi mode *pin* sebagai *input* atau *output* dan inialisasi 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.

## 2.16 Flowchart

Bagan alir merupakan teknik analitis yang digunakan untuk menjelaskan aspek-aspek sistem informasi secara jelas, tepat dan logis. Bagan alir menggunakan serangkaian simbol standar untuk menguraikan prosedur pengolahan transaksi yang digunakan oleh sebuah perusahaan, sekaligus menguraikan aliran data dalam sebuah sistem (Krismiaji, 2010).

Terdapat beberapa jenis bagan alir yang biasa digunakan, yaitu sebagai berikut:

### A. Bagan Alir Sistem (*System Flowchart*)

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

### B. Bagan Alir Dokumen (*Document Flowchart*)

Bagan alir dokumen (*document flowchart*) atau disebut dengan bagan alir formulir (*form flowchart*) atau *paperwork flowchart* merupakan bagan alir yang menunjukkan arus dari laporan dan formulir termasuk tembusan-tembusannya. Bagan alir dokumen ini menggunakan simbol-simbol yang sama dengan yang digunakan didalam bagan alir sistem. Simbol-simbolnya seperti pada tabel 2.7.

### **C. Bagan Alir Skematik (*Schematic Flowchart*)**

Bagan alir skematik (*schematic flowchart*) merupakan bagan alir yang mirip dengan bagan alir sistem, yaitu menggambarkan prosedur dalam sistem. Perbedaannya adalah bagan alir skematik selain menggunakan simbol-simbol bagan alir sistem, juga menggunakan gambar-gambar komputer dan peralatan lainnya yang digunakan. Maksud penggunaan gambar-gambar ini adalah untuk memudahkan dalam menjelaskan simbol-simbol bagan alir kepada orang yang masih awam.

### **D. Bagan Alir Program (*Program Flowchart*)**

Bagan alir program (*program flowchart*) terdiri dari dua macam, yaitu bagan alir logika program (*program logicflowchart*) dan bagan alir program komputer terinci (*detailed computer program flowchart*). Bagan alir logika program digunakan untuk menggambarkan tiap-tiap langkah di dalam program komputer secara logika. Bagan alir logika program ini dipersiapkan oleh analis sistem

### **E. Bagan Alir Proses (*Process Flowchart*)**

Bagan alir proses (*process flowchart*) merupakan bagan alir yang banyak digunakan di teknik industri. Berguna bagi analis sistem untuk menggambarkan proses dalam suatu prosedur.

Berdasarkan penjelasan di atas penulis dapat menarik simpulan bahwa bagan alir (*flowchart*) adalah suatu gambaran umum tentang sistem yang berjalan dan berfungsi sebagai alat bantu komunikasi serta untuk mendokumentasikan dan menyajikan kegiatan mulai dari manual, semi manual maupun komputerisasi, khususnya bagan alir sistem dan dokumen yang bersangkutan dengan sistem informasi kenaikan pangkat atau golongan guru bidang PSMP pada Dinas Pendidikan Kota Bandung.

Tabel 2.7 Daftar Simbol Flowchart (Bagan Alir) Dokumen

No	Gambar	Nama	Keterangan
1		Dokumen	Menunjukkan dokumen <i>input</i> dan <i>output</i> baik untuk proses manual, mekanik atau komputer.
2		Proses Manual	Merupakan proses manual dalam <i>flowchart</i> .
3		Simbol Proses Komputerisasi	Menunjukkan kegiatan proses dari operasi program komputer.
4		<i>File Hardisk/ Database</i>	Menunjukkan kegiatan <i>input</i> atau <i>output</i> menggunakan <i>hardisk</i> .
5		<i>Offline Storage</i>	Menunjukkan tfile non-komputer yang diarsip urut tanggal (cronological)
6		Simbol <i>Keyboard</i>	Merupakan <i>input</i> data yang menggunakan <i>online keyboard</i> .
7		Arus dokumen/ pemrosesan	Menunjukkan arus dari proses.
8		Keputusan	Menunjukkan tahapan pembuatan keputusan
9		Terminal	Menunjukkan awal dan akhir dari bagan alir dokumen.
10		<i>Input/Output</i>	Mewakili data <i>input/output</i> .
11		Penjelasan	Menunjukkan penjelasan dari suatu proses
12		<i>Connector</i>	Menunjukkan penghubung ke halaman yang sama atau ke halaman lain
13		Arus dari Jaringan	Data melalui channel komunikasi