

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Tikus**

Tikus adalah binatang yang termasuk dalam ordo *Rodentia*, Sub ordo *Myormorpha*, famili *Muridae*. Famili *Muridae* ini merupakan famili yang dominan dari ordo *Rodentia* karena mempunyai daya reproduksi yang tinggi, pemakan segala macam makanan (Omnivorous) dan mudah beradaptasi dengan lingkungan yang diciptakan manusia. Jenis tikus yang sering ditemukan di habitat rumah dan ladang adalah jenis *Rattus* dan *Mus*. (Solicha, 2007:18)

Berdasarkan jenis dan ciri-cirinya tikus dibedakan antara lain adalah sebagai berikut :

1. Tikus rumah
2. Tikus Riul (*Rattus norvegicus*)
3. Tikus Ladang (*Rattus exulans*)
4. Tikus Belukar (*Rattus tiomanicus sabae*)
5. Tikus Besar Gunung (*Sundamys infraluteus*)
6. Tikus sawah (*Rattus argentiventer*)
7. Mencit rumah (*Mus musculus*)
8. Mencit ladang (*Mus caroli*)

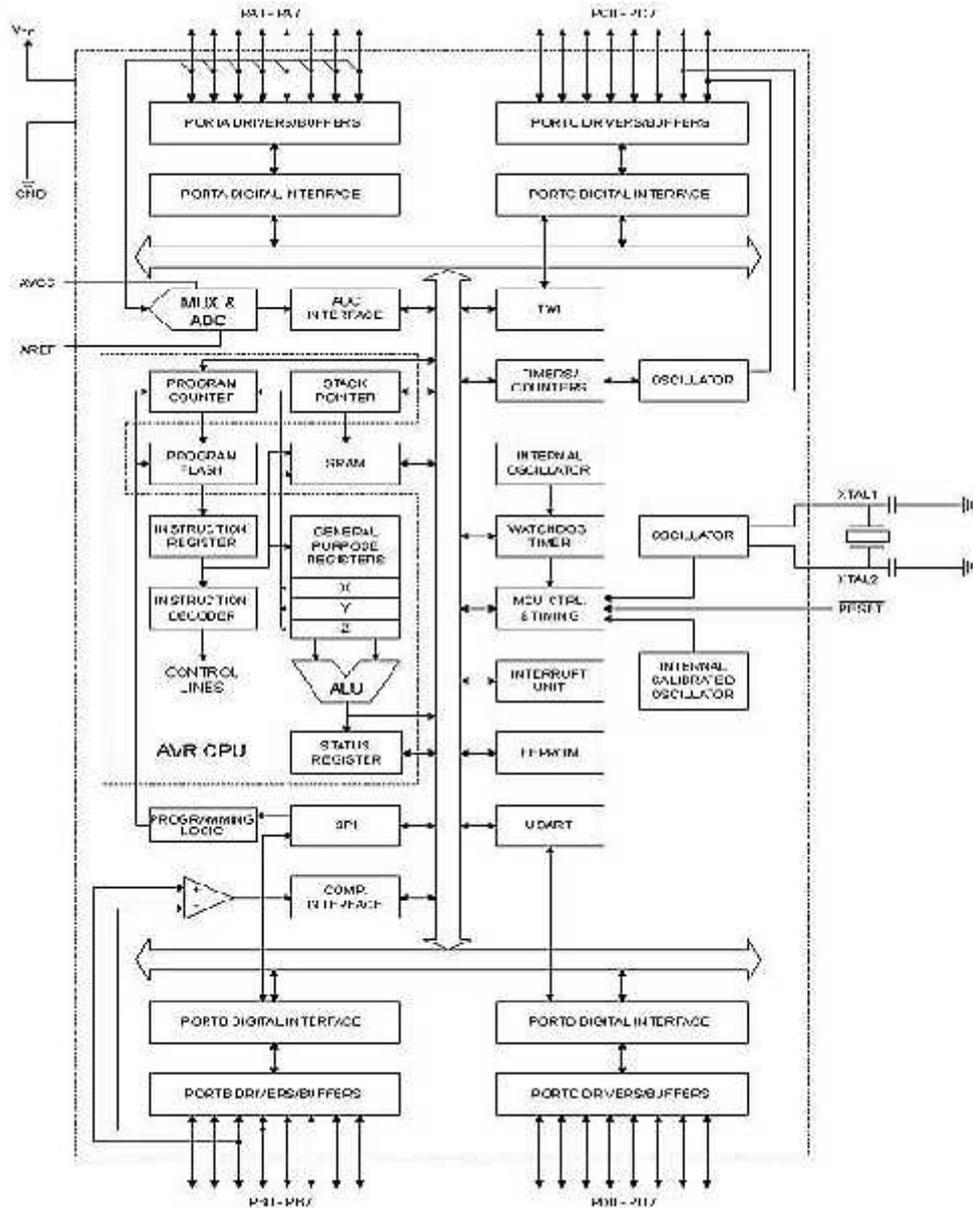
#### **2.2 Mikrokontroler ATMega 8535**

Mikrokontroler adalah sebuah sistem komputer fungsional dalam sebuah chip. Di dalamnya terkandung sebuah inti prosesor, memori (sejumlah kecil RAM, memori program, atau keduanya), dan perlengkapan *input output* (Ilham bukhori, 2012:5).

Mikrokontroler berbeda dari mikroprosesor serba guna yang digunakan dalam sebuah PC, karena sebuah mikrokontroler umumnya telah berisi komponen pendukung sistem minimal mikroprosesor, yakni memori dan antarmuka *I/O*. Sistem minimum mikrokontroler adalah rangkaian elektronik minimum yang diperlukan untuk beroperasinya IC mikrokontroler. Sistem

minimum ini kemudian bisa dihubungkan dengan rangkaian lain untuk menjalankan fungsi tertentu.

ATmega8535 banyak digunakan untuk sistem yang kompleks, memiliki input sinyal analog, dan membutuhkan memori yang relatif lebih besar. Berikut adalah blok diagram fungsional mikrokontroler seri ATmega8535.



**Gambar 2.1** Blok Diagram Fungsional ATMEGA 8535

Dari diagram blok diatas dapat dilihat mikrokontroler Atmega 8535 memiliki bagian sebagai berikut :

1. Saluran I/O sebanyak 32 buah, yaitu Port A, Port B, Port C, dan Port D.
2. ADC 10 bit sebanyak 8 saluran.
3. Tiga buah Timer/Counter dengan kemampuan perbandingan.
4. CPU yang terdiri atas 32 buah register.
5. Watchdog Timer dengan osilator internal.
6. SRAM sebesar 512 byte.
7. Memori Flash sebesar 8 kb dengan kemampuan Read While Write.
8. Unit interupsi internal dan eksternal.
9. Port antarmuka SPI.
0. EEPROM sebesar 512 byte yang dapat diprogram saat operasi.
11. Antarmuka komparator analog.
12. Port USART untuk komunikasi serial

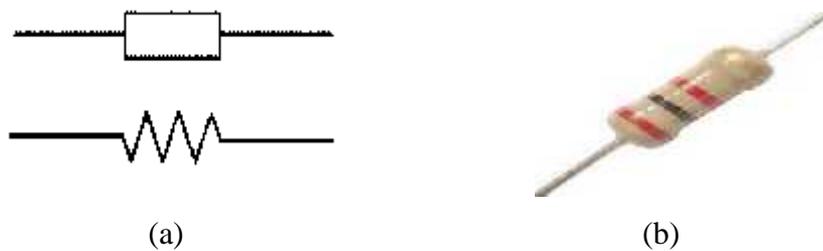
### **2.2.1 Fitur ATMEGA 8535**

Atmega 8535 memiliki fitur dan kapabilitas sebagai berikut :

1. Memori Flash 8 Kbytes untuk program
2. Memori EEOM 512 bytes untuk data
3. Memori SRM 512 bytes untuk data
4. Maksimal 32 pin I/O
5. 20 interrupt
6. Satu 16-bit timer dan dua 8-bit timer
7. 8 channel ADC 10 bit
8. Komunikasi serial melalui SPI dan USART
9. Analog komparator
10. I/O PWM
11. Fasilitas In System Programming (ISP)

### 2.3 Resistor

Resistor merupakan suatu benda yang dibuat sebagai penghambat atau penahanan arus listrik yang mengalir pada suatu rangkaian dengan tujuan untuk mengatur arus yang mengalir yang dinyatakan dengan satuan *ohm*.(Budiman, 1992:207)



**Gambar 2.2** (a) Simbol Resistor dan (b) Resistor

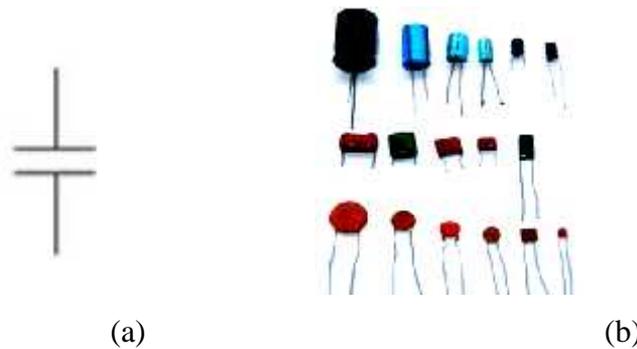
Gelang yang terdapat dalam resistor sebenarnya merupakan kode dari nilai resistansi yang terkandung di dalamnya, untuk dapat membacanya kita harus mengetahui kode tersebut. Berikut adalah kode warna resistor:

**Tabel 2.3** Kode warna resistor

Warna	Gelang 1	Gelang 2	Gelang 3	Multiplier	Toleransi
				Gelang 4	Gelang 5
Hitam	0	0	0	1	
Coklat	1	1	1	10	± 1 %
Merah	2	2	2	100	± 2 %
Orange	3	3	3	1 K	
Kuning	4	4	4	10 K	
Hijau	5	5	5	100 K	± 0,5 %
Biru	6	6	6	1 M	± 0,25 %
Ungu	7	7	7	10 M	± 0,10 %
Abu-abu	8	8	8		± 0,05 %
Putih	9	9	9		
Emas				0,1	±5 %
Perak				0,01	± 10 %

## 2.4 Kapasitor

Kapasitor merupakan suatu alat elektronis yang terdiri dari konduktor dan insulator yang mempunyai sifat sebagai penyimpan muatan listrik.(Budiman, 1992, hal:45)



Gambar 2.3 (a) Simbol Kapasitor dan (b) Kapasitor

## 2.5 Dioda

Dioda merupakan suatu komponen elektronik yang terdiri dari dua buah elektroda (yaitu *anoda* dan *katoda*) yang digunakan untuk meratakan atau mengarahkan aliran ke satu jurusan, yaitu dari anoda menuju katoda. Bahan untuk dioda yang umum digunakan yaitu *silikon* (Si) dan *Germanium*(Ge) dimana kedua bahan tersebut merupakan bahan semi konduktor.(Budiman, 1992, hal:72)

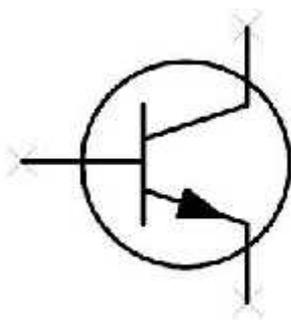


Gambar 2.4 (a) Simbol Dioda dan (b) Dioda

## 2.6 Transistor

Menurut Budiman (1992) Transistor berasal dari dua buah perkataan yaitu dari kata *transfer* dan resistor. *Transfer* berarti pemindahan dan resistor berarti penahan. Jadi transistor adalah pemindahan penahan. Elektroda-elektroda pada

transistor terdiri dari basis, kolektor dan emitor. Transistor biasanya digunakan sebagai saklar dalam rangkaian driver relay. Prinsip kerja dari transistor yang difungsikan sebagai saklar ialah dengan mendapatkan manfaat dari cut-off dan kondisi jenuh dari transistor itu sendiri, yang mana kedua keadaan tersebut bisa didapat dengan mengatur besarnya arus yg melewati basis dari transistor. Saturasi atau disebut juga kondisi / keadaan jenuh akan didapat bila basis transistor diberi arus yang cukup besar hingga transistor menjadi jenuh dan fungsinya menjadi saklar yang menutup. Sedangkan keadaan cut-off didapatkan apabila arus basisnya dilewati dengan arus yg amat kecil bahkan hampir nol ampere, yang menjadikan transistor berfungsi sebagai saklar yg membuka.



(a)



(b)

**Gambar 2.5** (a) Simbol Transistor dan (b) Transistor

## 2.7 ISD 1760 Recording Sound Module

ISD 1760 adalah sebuah modul yang dapat digunakan untuk merekam dan memainkan suara dengan penggunaan yang sangat mudah dan dengan kemampuan penyimpanan suara dari 60 – 75 detik. Modul ini sudah berisi komponen-komponen yang diperlukan untuk dapat mengoperasikan IC ISD 1760 dan tombol, sehingga siap untuk digunakan sebagai piranti perekam atau pemain suara dengan hanya penambahan loudspeaker dan supply tegangan saja. Modul ini dapat dioperasikan langsung melalui tombol-tombol yang sudah tersedia pada modul dan juga dapat menggunakan interface SPI. Tegangan supply dari modul ini adalah 5volt DC. ISD1700 ini dirancang untuk mudol mikrokontroler (SPI). Perangkat ini menggabungkan sistem manajemen pesan proprietary yang

memungkinkan chip untuk mengelola sendiri lokasi alamat untuk beberapa pesan/rekaman. Fitur unik ini memberikan fleksibilitas pesan canggih dalam lingkungan push-tombol sederhana. Termasuk perangkat osilator on-chip (dengan kontrol resistor eksternal), mikrofon preamplifier dengan Automatic Gain Control (AGC), input analog tambahan, antialiasing filter, Multi-Level Storage (MLS) array, smoothing filter, kontrol volume, Pulse Width Modulation (PWM) Kelas D speaker driver, dan output arus / tegangan.



**Gambar 2.6** Modul ISD1760

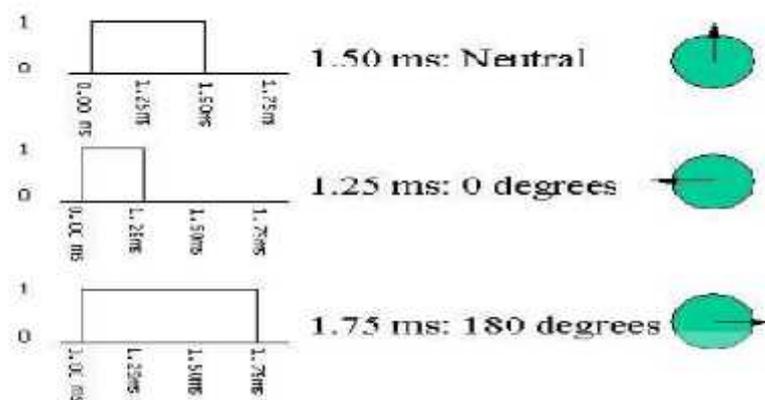
## **2.8 Motor Servo**

Motor servo adalah sebuah motor dengan menggunakan sebuah sistem umpan balik tertutup, posisi motor akan diinformasikan kembali kerangkaian kontrol yang ada didalam motor servo. Motor ini terdiri dari sebuah motor DC, serangkaian gear ,potensiometer dan rangkaian kontrol. Potensiometer berfungsi untuk menentukan batas sudut dari putaran servo. Sedangkan sudut dari sumbu motor servo diatur berdasarkan lebar pulsa yang dikirm melalui kaki sinyal dari kabel motor (Mirza,2009:132).

Magnet permanent motor DC servo akan mengubah energi listrik ke dalam energi mekanik melalui interaksi dari dua medan magnit . salah satu medan dihasilkan oleh magnit permanent dan yang satunya dihasilakn oleh arus yang mengalir dalam kumparan motor. Resultan dari dua medan magnit tersebut menghasilkan torsi yang membangkitkan putaran motor tersebut . Saat motor berputar, arus pada kumparan motor menghasilkan torsi yang nilainya konstan.

Secara umum terdapat dua jenis motor servo yaitu motor servo standart dan motor servo Continuous dimana motor servo standart lebih sering dipakai pada sistem robotika misalnya pada pembuatan robot arm atau robot lengan, motor servo jenis ini hanya mampu bergerak dua arah (CW dan CCW) dengan defleksi masing-masing sudut mencapai  $90^\circ$  sehingga total defleksi sudut dari kanan-tengan – kiri adalah  $180^\circ$ . Sedangkan motor servo continuous sering dipakai untuk mobile robot, motor servo jenis ini mampu bergerak dua arah (CW dan CCW) tanpa batasan defleksi sudut putar (dapat berputar secara kontinyu).

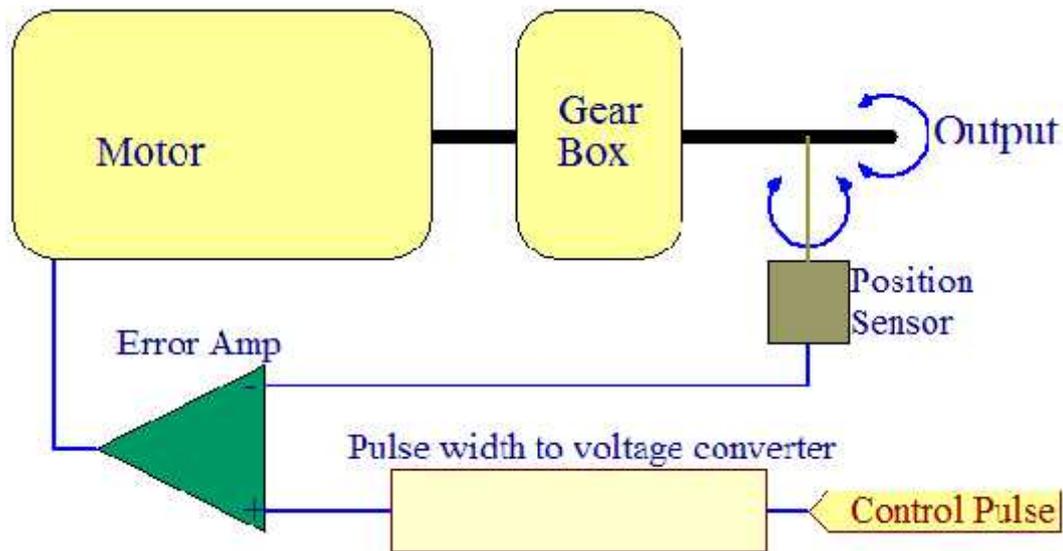
Motor servo memiliki putaran yang lambat dan torsi yang kuat (berkat adanya sistem gear). Poros pada motor servo dapat dihubungkan langsung dengan roda. Lalu, sistem kontrol jalur untuk motor servo relatif lebih sedikit (hanya diperlukan 1 jalur data saja).



**Gambar 2.7** Pemberian Pulsa Untuk Perputaran Motor Servo

Karena ada tiga posisi utama seperti yang dijelaskan diatas maka dibuatlah secara khusus mengatur motor srvo tersebut, dengan cara memberikan pulsa digital dengan lebar yang berbeda – beda. Jika diberikan pulsa dengan lebar 1.5ms maka motor servo akan berputar 90 derajat, pulsa dengan 1.75ms akan membuat motor servo menuju 180 derajat, sedangkan pulsa dengan lebar 1.25ms akan membuat motor servo bergerak menuju 0 derajat, motor servo tersebut disebut Motor servo standart yang memiliki batas, hal ini menyebabkan poros servo tidak

berputar 360 derajat, sedangkan motor servo continuous jika diberi puls 1.25ms akan berputar CW dan sedangkan jika diberi 1.75ms maka akan berputar CCW dan juga bila diberi 1,5ms motor servo kan diam tidak bergerak. Pada dasarnya motor servo continuous akan berputar 360 derajat. Berikut adalah diagram blok dari motor servo :



**Gambar 2.8** Diagram Blok Motor Sevo

## 2.9 Sensor

### 2.9.1 Pengertian Umum Sensor

Sensor adalah alat untuk mendeteksi / mengukur sesuatu yang digunakan untuk mengubah variasi mekanis, magnetis, panas, sinar dan kimia menjadi tegangan arus listrik. Sensor itu sendiri terdiri dari transduser dengan atau tanpa penguat/pengolah sinyal yang terbentuk dalam satu sistem pengindera. Dalam lingkungan sistem pengendali dan robotika, sensor memberikan kesamaan yang menyerupai mata, pendengaran, hidung, lidah yang kemudian akan diolah oleh controller sebagai otaknya. (Fauzi, 2011:7)

### 2.9.2 Sensor PIR (Passive Infra Red)

PIR (Passive Infrared) adalah sebuah sensor yang menangkap pancaran sinyal infra merah yang dikeluarkan oleh tubuh manusia. Sensor PIR (Passive Infrared) dapat merespon perubahan perubahan pancaran sinyal infra merah yang

dipancarkan oleh tubuh manusia, pancaran sinyal infra merah ditubuh manusia dengan panjang gelombang  $9,4\mu\text{m}$ . Sensor PIR (Passive Infrared) terbuat dari bahan kristal yang akan menimbulkan beban listrik ketika terkena panas dan pancaran sinyal infra merah. Perubahan intensitas pancaran dari sinyal infra merah juga menyebabkan perubahan beban listrik pada sensor. Elemen-elemen pada sensor juga sensitif terhadap penyinaran yang melebihi lebar jangkauan, sehingga ditambahkan filter pada kemasan TO5 untuk membatasi pancaran tubuh manusia. (Fauzi, 2011:8)



**Gambar 2.9** Sensor PIR

## 2.10 Relay

Relay adalah saklar elektronik yang dapat membuka atau menutup rangkaian dengan menggunakan kontrol dari rangkaian elektronik lain. Sebuah relay tersusun atas kumparan, pegas, saklar (terhubung pada pegas) dan 2 kontak elektronik (*normally close dan normally open*).

- a. *Normally close* (NC) : saklar terhubung dengan kontak ini saat *relay* tidak aktif atau dapat dikatakan saklar dalam kondisi terbuka.
- b. *Normally open* (NO) : saklar terhubung dengan kontak ini saat *relay* aktif atau dapat dikatakan saklar dalam kondisi tertutup.

(Dwi Nugroho, Ichsan. 2012. *Alat Pengatur Lampu Dan Pembalik Telur Otomatis Pada Bok Penetasan Telur Berbasis Mikrokontroler Atmega 16 Dilengkapi Uninterruptible Power Supply*. Yogyakarta : Universitas Negeri Yogyakarta)

Berdasarkan pada prinsip dasar cara kerjanya, *relay* dapat bekerja karena adanya medan magnet yang digunakan untuk menggerakkan saklar. Saat kumparan diberikan tegangan sebesar tegangan kerja *relay* maka akan timbul medan magnet pada kumparan karena adanya arus yang mengalir pada lilitan kawat. Kumparan yang bersifat sebagai elektromagnet ini kemudian akan menarik saklar dari kontak *NC* ke kontak *NO*. Jika tegangan pada kumparan dimatikan maka medan magnet pada kumparan akan hilang sehingga pegas akan menarik saklar ke kontak *NC*. *Relay* yang digunakan pada rangkaian ini memiliki spesifikasi SRU 12 VDC-SL-C. Jumlah pin pada *relay* ada 5 dan bertegangan kerja 12 VDC. Kemampuan arus yang dapat dilewatkan kontaktor adalah 10A pada tegangan 250VAC, 15A pada tegangan 120VAC, dan 10A pada tegangan 30VDC.



**Gambar 2.10.** Bentuk fisik *relay* SRU-12VDC-SL-C

### 2.11 Modul TLP-315 dan RLP-315

Modul RF yang digunakan adalah TLP-315 (Pemancar) dan RLP-315 (Penerima). Modul RF buatan LAIPAC ini bekerja pada system modulasi ASK (Amplitude Shift Keying) dan memiliki frekuensi 315 Mhz dan dapat diberi supply voltage untuk TLP sebesar 2-12 VDC dan untuk RLP 3,3-6.0 VDC output dari modul ini sudah berbentuk digital.

(Ricky, 2011:31-32)

Keunggulan dari modul TLP-315 ini adalah :

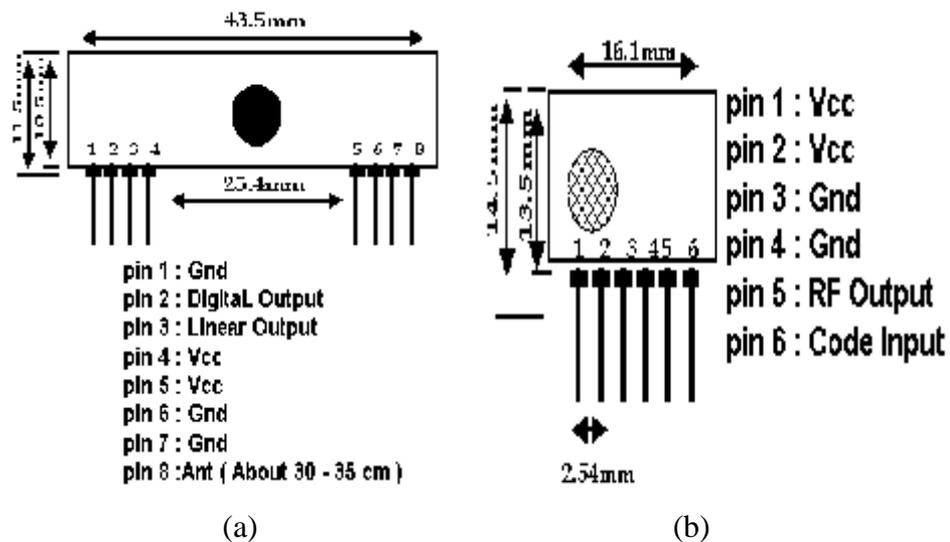
1. Menggunakan modulasi digital
2. Mempunyai frekuensi kerja yang aman digunakan
3. Bentuk fisik yang kecil

Karakteristik pin dari TLP-315 adalah sebagai berikut :

1. Pin 1 : Ground

2. Pin 2 : Data In
3. Pin 3 : Vcc
4. Pin 4 : Antena (RF output)

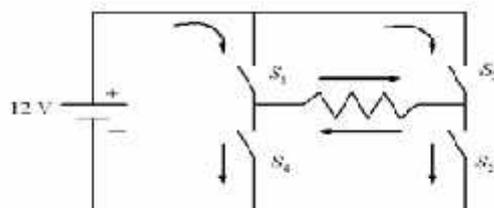
TLP-315 Transmitter masing-masing mempunyai pasangan receivernya. Masing-masing module transmitter mempunyai frekuensi kerja yang tidak bisa diubah-ubah.



Gambar 2.11 (a) Modul RLP-315 dan (b) Modul TLP-315

### 2.12 Inverter Tegangan DC-AC

Inverter adalah suatu alat yang dapat digunakan untuk mengubah tegangan masukan dc menjadi tegangan keluaran ac. Keluaran inverter dapat berupa tegangan yang dapat diatur dan tegangan yang tetap. Sumber tegangan masukan inverter dapat menggunakan baterai, tenaga surya, atau sumber tegangan dc yang lain. Tegangan keluaran yang biasa dihasilkan adalah 120 V, 220 V dan 115 V (Rahmi Dewi, dkk. 2006).



Gambar 2.12 Prinsip Kerja Inverter

Prinsip kerja inverter dapat dijelaskan dengan menggunakan 4 sakelar seperti ditunjukkan pada Gambar 2. Bila sakelar 1 S dan sakelar 2 S dalam kondisi bekerja serta sakelar 3 S dan sakelar 4 S dalam keadaan tidak bekerja, maka akan mengalir arus listrik dc ke beban R dari arah kiri ke kanan. Jika yang bekerja adalah sakelar 3 S dan 4 S serta dalam keadaan tidak bekerja, maka akan mengalir arus dc ke beban R dari arah kanan ke kiri. Apabila sakelar 1 S dan sakelar 2 S serta sakelar 3 S dan sakelar 4 S bekerja secara bergantian maka akan timbul arus bolak-balik pada beban.

## 2.13 Bahasa C

Pencipta bahasa C adalah Brian W. Kernighan dan Dennis M. Ritchie padasekitar tahun 1972. C adalah bahasa pemrograman terstruktur, yang membagi program dalam bentuk sejumlah blok. Tujuannya adalah untuk memudahkan dalam pembuatan dan pengembangan program. Program yang ditulis dengan menggunakan C mudah sekali untuk dipindahkan dari satu jenis mesin ke jenis mesin lainnya. Hal ini berkat adanya standarisasi bahasa C yaitu berupa standar ANSI (*American National Standards Institute*) yang dijadikan acuan oleh para pembuat kompilasi C. (Bejo, 2008).

### 2.13.1 Bentuk Dasar Program C

Sebuah program dalam bahasa C setidaknya memiliki sebuah fungsi. Fungsi ini disebut dengan fungsi utama (fungsi *main*) dan memiliki kerangka program sebagai berikut :

```
Void main (void)
{
    //pernyataan
}
```

Jika kita mempunyai beberapa fungsi yang lain maka fungsi utama inilah yang memiliki kedudukan tertinggi dibandingkan fungsi-fungsi yang lain sehingga setiap kali program dijalankan akan selalu dimulai dari pemanggilan

fungsi utama. Fungsi-fungsi yang lain akan dipanggil setelah fungsi utama dan dijalankan melalui pernyataan-pernyataan yang berada dalam fungsi utama.

### 2.13.2 Pengenal

Pengenal (*identifier*) merupakan sebuah nama yang didefinisikan oleh pemrogram untuk menunjukkan identitas dari sebuah konstanta, variabel, fungsi, label, atau tipe data khusus. Pemberian nama sebuah pengenal ditentukan bebas sesuai keinginan pemrogram tetapi harus memenuhi aturan berikut:

1. Karakter pertama tidak boleh menggunakan angka.
2. Karakter kedua dapat berupa huruf, angka, atau garis bawah.
3. Bersifat *Case Sensitive*, yaitu huruf kapital dan huruf kecil dianggap berbeda.
4. Tidak boleh menggunakan kata-kata yang merupakan sintaks maupun operator dalam pemrograman C

### 2.13.3 Tipe Data

Tipe data merupakan bagian program yang paling penting karena tipe data mempengaruhi setiap instruksi yang akan dilaksanakan oleh komputer. Pemilihan tipe data yang tepat akan membuat proses operasi data menjadi lebih efisien dan efektif. Adapun tipe-tipe data yang ada dalam bahasa C adalah sebagai berikut:

**Tabel 2.1** Tipe data

Type Data	Ukuran	Jangkauan nilai
<i>Bit</i>	1 bit	0 atau 1
<i>Char</i>	1 byte	-128 s/d 127
<i>Unsigned Char</i>	1 byte	0 s/d 255
<i>Signed Char</i>	1 byte	-128 s/d 127
<i>Int</i>	2 byte	-32.768 s/d 32.767
<i>Short Int</i>	2 byte	-32.768 s/d 32.767
<i>Unsigned Int</i>	2 byte	0 s/d 65.535
<i>Signed Int</i>	2 byte	-32.768 s/d 32.767
<i>Long Int</i>	4 byte	-2.147.483.648 s/d

		2.147.483.647
<i>Unsigned Long Int</i>	4 byte	0 s/d 4.294.967.295
<i>Signed Int</i>	4 byte	-2.147.483.648 s/d 2.147.483.647
<i>Float</i>	4 byte	$1.2 \cdot 10^{-38}$ s/d $3.4 \cdot 10^{+38}$
<i>Double</i>	4 byte	$1.2 \cdot 10^{-38}$ s/d $3.4 \cdot 10^{+38}$

#### 2.13.4 Konstanta dan Variabel

Konstanta dan variabel merupakan sebuah tempat untuk menyimpan data yang berada di dalam memori. Konstanta berisi data yang nilainya tetap dan tidak dapat diubah selama program dijalankan, sedangkan variabel berisi data yang bisa berubah nilainya pada saat program dijalankan.

Untuk membuat sebuah konstanta atau variabel maka kita harus mendeklarasikannya terlebih dahulu yaitu dengan sintaks sebagai berikut.

##### 1. Deklarasi Konstanta :

```
Const [tipe_data] [nama_konstanta]=[nilai]
```

Contoh :

```
Const char konstan=0x10;
```

##### 2. Deklarasi Variabel :

```
[tipe_data] [nama_variabel]=[nilai_awal];
```

Contoh :

```
Char variabelku;  
Char variabelkku=0x20;  
Bit variable_bit;  
Bit variable_bit=1;
```

#### 2.13.5 Pernyataan IF

Pernyataan *if* digunakan untuk melakukan pengambilan keputusan terhadap dua buah kemungkinan yaitu mengerjakan suatu blok pernyataan atau tidak. Bentuk pernyataan *if* adalah sebagai berikut :

```
If (kondisi)
{
    // blok pernyataan yang akan dikerjakan
    // jika kondisi if terpenuhi
}
```

Contoh :

```
If (PINA>0x80)
{
Dataku = PINA; PORTC=0xFF;
```

Pernyataan *if* diatas akan mengecek apakah data yang terbaca pada *port A* (PINA) nilainya lebih dari 0x80 atau tidak, jika ya maka variabel dataku diisi dengan nilai PINA dan data 0xFF dikeluarkan ke *port C*.

Apabila dalam blok pernyataan hanya terdapat satu pernyataan saja maka tanda { dan } dapat dihilangkan seperti pada contoh berikut :

```
If (PINA>0x80)
PORTC=0xFF
```

### 2.13.6 Pernyataan Switch

Pernyataan *Switch* digunakan untuk melakukan pengambilan keputusan terhadap banyak kemungkinan. Bentuk pernyataan *Switch* adalah sebagai berikut :

```
switch (ekspresi) {
    Case nilai_1 : pernyataan_1 ; break;
    Case nilai_2 : pernyataan_1 ; break;
    Case nilai_3 : pernyataan_1 ; break;
    Default : pernyataan_default; break;
}
```

Pada pernyataan *switch*, masing-masing pernyataan (pernyataan\_1 sampai dengan pernyataan\_default) dapat berupa satu atau beberapa perintah dan tidak perlu berupa blok pernyataan. Pernyataan\_1 akan dikerjakan jika ekspresi bernilai sama dengan nilai\_1, pernyataan\_2 akan dikerjakan jika ekspresi bernilai sama dengan nilai\_2, pernyataan\_3 akan dikerjakan jika ekspresi bernilai sama dengan nilai\_3, dan seterusnya. Pernyataan\_default bersifat opsional, artinya boleh ada boleh tidak. Jika ada maka pernyataan\_default akan dikerjakan jika nilai ekspresi tidak ada yang sama satupun dengan salah satu nilai\_1, nilai\_2, nilai\_3 dan seterusnya.

Setiap akhir pernyataan harus diakhiri dengan *break*, karena ini digunakan untuk keluar dari pernyataan *switch*. Contoh :

```

switch (PIN_A)
{
    Case 0xFE : PORTC = 0x00; break;
    Case 0xFD : PORTC = 0xFF; break;
}

```

Pernyataan diatas berarti membaca *port A*, kemudian datanya (PIN\_A) akan dicocokkan dengan nilai case. Jika PIN\_A bernilai 0xFE maka data 0x00 akan dikeluarkan ke *port C* kemudian program keluar dari pernyataan *switch* tetapi jika PIN\_A bernilai 0xFD maka data 0xFF akan dikeluarkan ke *port C* kemudian program keluar dari pernyataan *switch*.

### 2.13.7 Pernyataan While

Pernyataan *while* digunakan untuk perulangan sebuah pernyataan atau blok pernyataan yang secara terus menerus selama kondisi tertentu masih terpenuhi. Bentuk pernyataan *while* adalah sebagai berikut :

```

while (kondisi) {
    // sebuah pernyataan atau blok pernyataan
}

```

Jika pernyataan yang akan diulang hanya berupa sebuah pernyataan saja maka tanda { dan } bisa dihilangkan. Contoh :

```

Unsigned char a=0;
...
while (a<10) {
    PORTC=a;
    a++;
}

```

Pernyataan diatas akan mengeluarkan data a ke *port C* secara berulang-ulang. Setiap kali pengulangan nilai a akan bertambah 1 dan setelah nilai a mencapai 10 maka pengulangan selesai.

### 2.13.8 Pernyataan For

Pernyataan *for* digunakan untuk melakukan pengulangan sebuah pernyataan atau blok pernyataan, tetapi berapa kali jumlah pengulangannya dapat ditentukan secara lebih spesifik. Bentuk pernyataannya adalah sebagai berikut :

```
for (nilai_awal; kondisi; perubahan) {  
    // sebuah pernyataan atau blok pernyataan  
}
```

Nilai\_awal merupakan nilai inisial awal sebuah variabel yang didefinisikan terlebih dahulu untuk menentukan nilai variabel pertama kali sebelum pengulangan.

Kondisi merupakan pernyataan pengetesan untuk mengontrol pengulangan, jika pernyataan kondisi terpenuhi maka blok pernyataan akan diulang terus sampai pernyataan kondisi tidak terpenuhi.

Perubahan adalah pernyataan yang digunakan untuk melakukan perubahan nilai variabel baik naik maupun turun setiap kali pengulangan dilakukan.

Contoh ;

```
Unsigned int a;  
For (a =1, a<10, a++) {  
    PORTC=a;  
}
```

Pertama kali nilai a adalah 1, kemudian data a dikeluarkan ke *port C*. selanjutnya data a dinaikkan (a++) jika kondisi a <10 masih terpenuhi maka data a akan terus dikeluarkan ke *port C*.

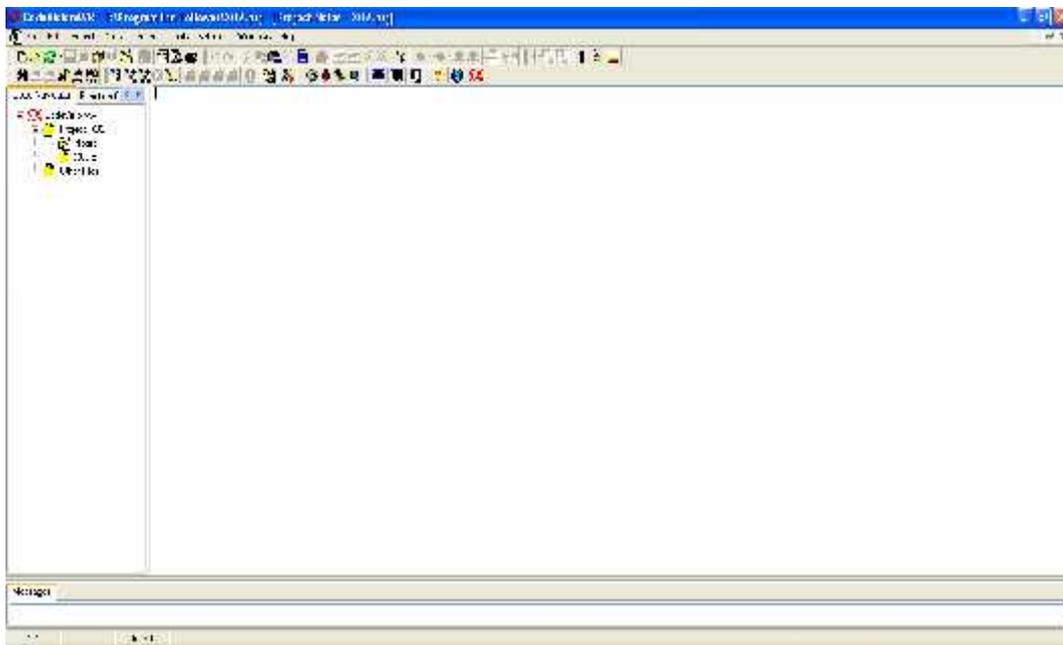
## 2.14 Code Vision AVR

Code Vision AVR *compiler* merupakan *compiler* bahasa C untuk AVR. Kompiler ini cukup memadai untuk belajar AVR, karena mudah penggunaannya juga didukung berbagai fitur yang sangat membantu dalam pembuatan *software* untuk keperluan pemrograman AVR. (Sumardi, 2013, hal:12)

CVAVR dapat berjalan dibawah sistem operasi Windows 9x, Me, NT, 2000 dan XP. CVAVR ini dapat mengimplementasikan hampir semua instruksi bahasa C yang sesuai dengan arsitektur AVR, bahkan terdapat beberapa keunggulan tambahan untuk memenuhi keunggulan spesifik dari AVR. Hasil kompilasi objek CVAVR bisa digunakan sebagai *source debug* dengan AVR Studio debugger dari ATMEL.

Selain pustaka standar bahasa C, CVAVR juga menyediakan pustaka tambahan yang sangat membantu pemrograman AVR, yaitu :

- a. Alphanumeric LCD *modules*
- b. Philips 12C *bus*
- c. National *semiconductor* LM75 *temperature sensor*
- d. Philips PCF8563, PCF8583, Maxim/Dallas *semiconductor* DS1302 and DS1307 *real time clocks*
- e. Maxim/Dallas *semiconductor* 1 *wire protocol*
- f. Maxim/Dallas *semiconductor* DS 1820, DS28S20, DS18B20 *Temperature sensor*
- g. Maxim/Dallas *semiconductor* DS 1621 *Thermometer/Thermostat*
- h. Maxim/Dallas *semiconductor* DS2430 and DS2433 *EEPROMs*
- i. SPI
- j. *Power Management*
- k. *Delays*
- l. *Gray Code Conversion*



**Gambar 2.8** Tampilan *Interface* CodeVision AVR (Sumardi, 2013)

CodeVisionAVR menyediakan suatu fasilitas yang bernama Code Wizard AVR, dimana fasilitas ini mempermudah dalam pemilihan jenis mikrokontroler,

serta pengaktifan fasilitas-fasilitas dari mikrokontroler seperti *timer*, LCD, *input/output*, *external IRQ*, dan lain-lainnya.

Keterangan *tool* pada *toolbar* CodeVisionAVR dapat dilihat pada tabel berikut :

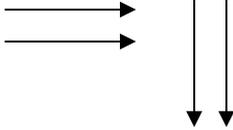
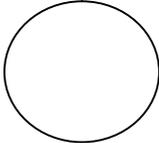
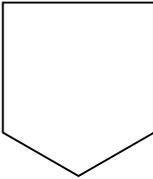
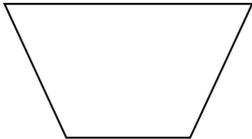
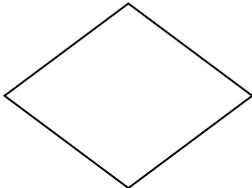
**Tabel 2.2** Keterangan *tool* pada *toolbar* CodeVisionAVR

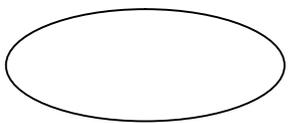
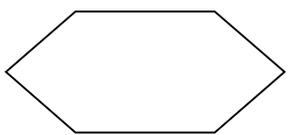
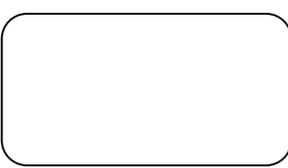
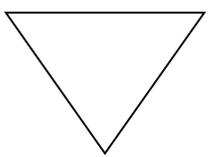
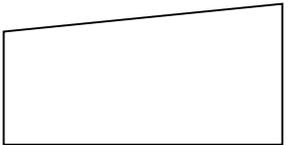
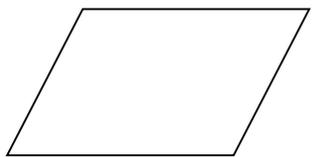
Icon	Nama	Fungsi
	<i>Create New File</i>	Membuat <i>file</i> baru
	<i>Open File</i>	Menbuka <i>file</i>
	<i>Print File</i>	Mencetak <i>file</i>
	<i>Run the CodeWizardAVR</i>	Menjalankan fasilitas CodeWizardAVR
	<i>Check Syntax</i>	Memeriksa kesalahan penulisan bahasa pemrograman
	<i>Run the Chip Programmer</i>	Menjalankan fasilitas Chip Programmer
	<i>Build the Project</i>	Men- <i>compile</i> program

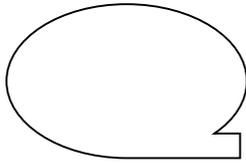
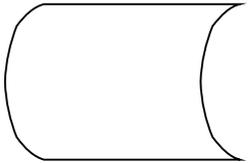
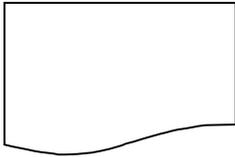
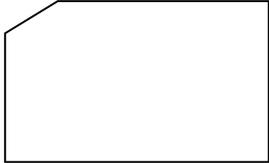
## 2.15 Flowchart

Menurut Kristanti (2012) dan Redita (2012) *Flowchart* adalah cara penyajian *visual* aliran data melalui sistem informasi, *Flowchart* dapat membantu menjelaskan pekerjaan yang saat ini dilakukan dan bagaimana cara meningkatkan atau mengembangkan pekerjaan tersebut. Dengan menggunakan *flowchart* dapat juga membantu untuk menemukan elemen inti dari sebuah proses, selama garis digambarkan secara jelas antara di mana suatu proses berakhir dan proses selanjutnya dimulai. Adapun simbol-simbol dari *flowchart* adalah sebagai berikut:

**Tabel 2.4** Tabel simbol diagram flowchart.

No	Simbol	Keterangan
1		Simbol arus / <i>flow</i> , yaitu menyatakan jalannya arus suatu proses
2		Simbol <i>connector</i> , berfungsi 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 <i>process</i> , yaitu menyatakan suatu tindakan (proses) yang dilakukan oleh komputer
5		Simbol <i>manual</i> , yaitu 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		<p>Simbol <i>terminal</i>, yaitu menyatakan permulaan atau akhir suatu program</p>
8		<p>Simbol <i>predefined process</i>, yaitu menyatakan penyediaan 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 simbol ini akan disimpan ke suatu mediatertentu</p>
11		<p>Simbol <i>manual input</i>, memasukkan 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>

13		<p>Simbol <i>magnetic tape</i>, menyatakan input berasal dari pita magnetis atau output disimpan ke pita magnetis</p>
14		<p>Simbol <i>disk storage</i>, menyatakan input berasal dari disk atau output disimpan ke disk</p>
15		<p>Simbol <i>document</i>, mencetak keluaran dalam bentuk dokumen (melalui printer)</p>
16		<p>Simbol <i>punched card</i>, menyatakan input berasal dari kartu atau output m ditulis ke kartu</p>