

## BAB II TINJAUAN PUSTAKA

### 2.1 Mikrokontroler ATmega8535

Mikrokontroler adalah IC yang dapat diprogram berulang kali, baik ditulis atau dihapus (Agus Bejo, 2007). Biasanya digunakan untuk pengontrolan otomatis dan manual pada perangkat elektronika. Beberapa tahun terakhir, mikrokontroler sangat banyak digunakan terutama dalam pengontrolan robot. Seiring perkembangan elektronika, mikrokontroler dibuat semakin kompak dengan bahasa pemrograman yang juga ikut berubah. Salah satunya adalah mikrokontroler AVR (*Alf and Vegard's Risc processor*) ATmega8535 yang menggunakan teknologi RISC (*Reduce Instruction Set Computing*) dimana program berjalan lebih cepat karena hanya membutuhkan satu *siklus clock* untuk mengeksekusi satu instruksi program. Secara umum, AVR dapat dikelompokkan menjadi 4 kelas, yaitu kelas ATtiny, keluarga AT90Sxx, keluarga ATmega, dan AT86RFxx. Pada dasarnya yang membedakan masing-masing kelas adalah memori, peripheral, dan fungsinya. Dari segi arsitektur dan instruksi yang digunakan, mereka bisa dikatakan hampir sama.

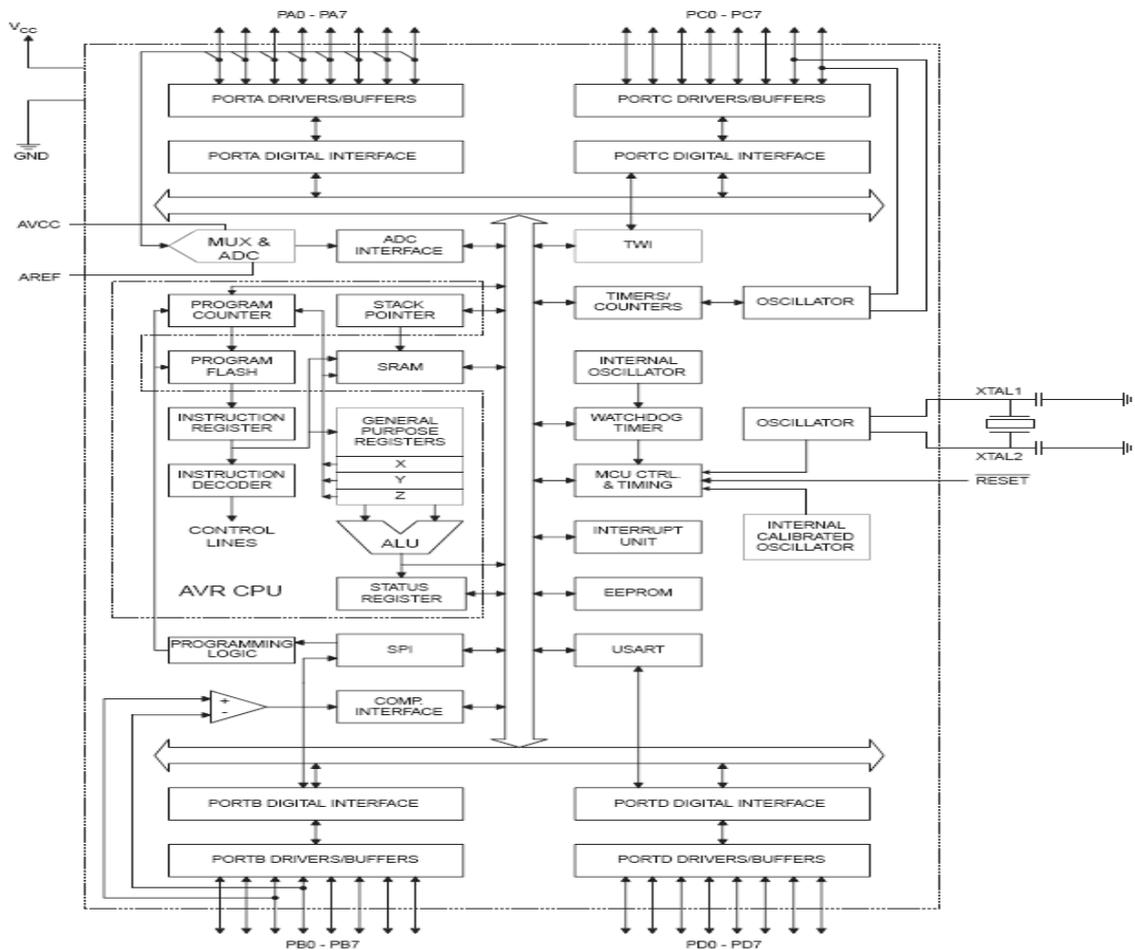
Mikrokontroler AVR ATmega8535 memiliki fitur yang cukup lengkap. Mikrokontroler AVR ATmega8535 telah dilengkapi dengan ADC *internal*, *EEPROM internal*, *Timer/Counter*, *PWM*, *analog comparator*, dll (M.Ary Heryanto, 2008). Sehingga dengan fasilitas yang lengkap ini memungkinkan kita belajar mikrokontroler keluarga AVR dengan lebih mudah dan efisien, serta dapat mengembangkan kreativitas penggunaan mikrokontroler ATmega8535.

Mikrokontroler ATmega8535 secara garis besar terdiri dari CPU yang terdiri dari 32 buah register, saluran I/O, ADC, *Port* antarmuka, *Port serial*. Mikrokontroler ATmega8535 merupakan anggota keluarga mikrokontroler AVR (*Alfand Vegard's Risc Processor*).

AVR ATmega8535 memiliki ruang pengalamatan memori data dan memori program yang terpisah. Memori data terbagi menjadi 3 bagian, yaitu 32 buah register umum, 64 buah register I/O dan 512 byte SRAM Internal.

### 2.1.1 Blok Diagram Mikrokontroler ATmega8535

Berikut blok diagram yang terdapat pada piranti mikrokontroler :



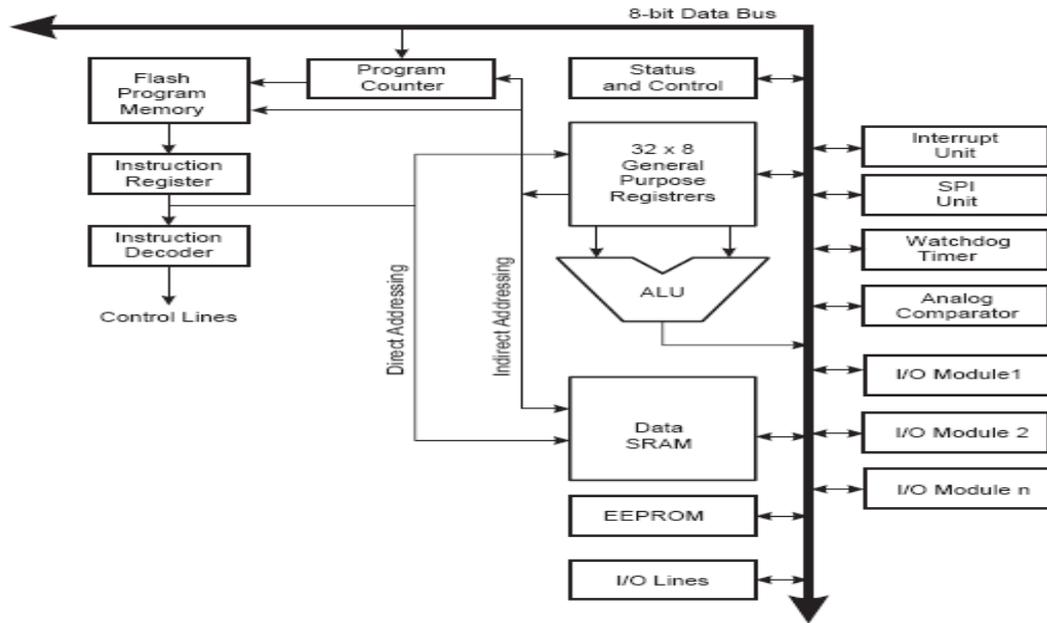
**Gambar 2.1** Blok Diagram Mikrokontroler ATmega8535

(Sumber : Syahrul,2012: 12)

### 2.1.2 Arsitektur Mikrokontroler ATmega8535

Mikrokontroler ATmega8535 memiliki arsitektur Harvard, yaitu memisahkan memori untuk kode program dan memori untuk data sehingga dapat memaksimalkan unjuk kerja dan paralelisme. Instruksi-instruksi dalam memori program dieksekusi dalam satu alur tunggal, dimana pada saat satu instruksi dikerjakan instruksi berikutnya sudah diambil (*pre-fetched*) dari memori program. Konsep inilah yang memungkinkan instruksi-instruksi dapat dieksekusi dalam setiap satu siklus *clock*.

Untuk mengetahui alur hubungan dari *architecture* ATmega8535 dapat di lihat pada gambar berikut:

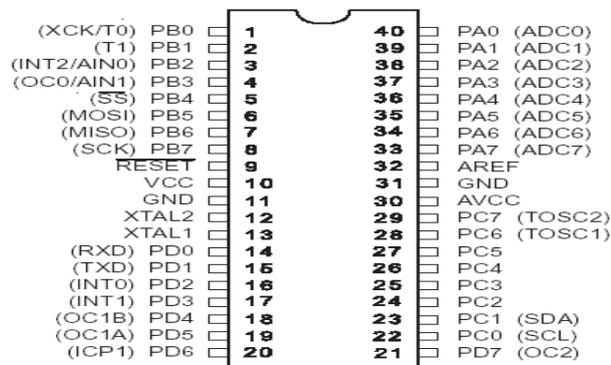


**Gambar 2.2** Arsitektur Mikrokontroler ATmega8535

(Sumber : Anonim, 2013)

### 2.1.3 Konfigurasi Pin Mikrokontroler ATmega8535

Berikut susunan pin-pin mikrokontroler ATmega 8535:



**Gambar 2.3** Konfigurasi Pin Mikrokontroler Atmega8535

(Sumber : Syahrul, 2012:13)

Keterangan gambar yaitu:

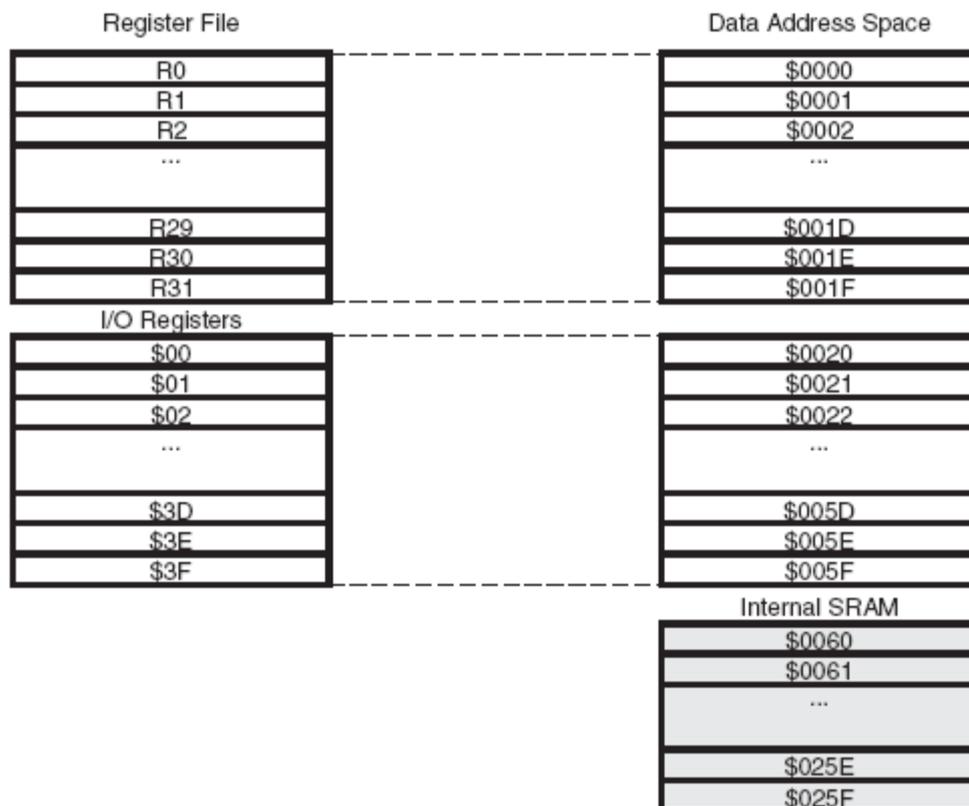
- a. Pin 1 – 8 adalah Port B (PB0 – PB7) merupakan pin I/O dua arah dan pin fungsi khusus, yaitu *timer/ counter*, komparator analog, dan SPI. Port B merupakan I/O 8 bit *bidirectional* dengan *resistor pull-up internal* (dipilih untuk setiap bit).

- b. Pin 9 (*reset*) adalah pin yang digunakan untuk mereset mikrokontroler, dan bekerja bila diberi pulsa rendah (aktif *low*) selama minimal 1.5 us.
- c. Pin 10 (*Vcc*) merupakan pin masukan positif catu daya. Setiap peralatan elektronika digital tentunya butuh sumber catu daya yang umumnya sebesar 5V itulah sebabnya di PCB kit mikrokontroler selalu ada IC regulator 7805.
- d. Pin 11 (*Ground*) sebagai pin *ground*.
- e. Pin 12 dan Pin 13 (*XTAL 2* dan *XTAL 1*) sebagai pin masukan *clock external*. Suatu mikrokontroler membutuhkan sumber detak atau *clock* agar dapat mengeksekusi instruksi yang ada di memori. Semakin tinggi nilai kristalnya maka semakin cepat mikrokontroler tersebut.
- f. Pin 14 – 21 adalah *Port D* (*PD0 - D7* ) merupakan pin I/O dua arah dan pin fungsi khusus yaitu komparator analog, interupsi internal dan komunikasi serial. *Port D* merupakan I/O 8 bit *bidirectional* dengan resistor *pull-up internal* (dipilih untuk setiap bit).
- g. Pin 22 – 29 adalah *Port C* (*PC0 – PC7*) merupakan pin I/O dua arah dan pin fungsi khusus yaitu *TWI*, komparator analog, dan *timer osilator*. *Port C* merupakan I/O 8 bit *bidirectional* dengan resistor *pull-up internal* (dipilih untuk setiap bit).
- h. Pin 30 (*AVCC*) sebagai pin masukan tegangan untuk ADC.
- i. Pin 31 (*GND*) sebagai pin *ground*.
- j. Pin 32 (*AREF*) sebagai pin masukan tegangan referensi analog untuk ADC.
- k. Pin 33 - 40 adalah *Port A* (*PA0 – PA7*) merupakan pin I/O dua arah dan dapat diprogram sebagai pin masukan 8 chanel ADC. *Port A* berfungsi sebagai input analog ke ADC. *Port A* juga dapat berfungsi sebagai *port* I/O 8 bit *bidirectional*, jika ADC tidak digunakan maka *port* dapat menyediakan resistor *pull-up internal* (dipilih untuk setiap bit).

### 2.1.4 Peta Memory ATmega8535

ATmega8535 memiliki ruang pengalamatan memori data dan memori program yang terpisah. Memori data terbagi menjadi 3 bagian yaitu : 32 buah register umum, 64 buah register I/O, dan 512 byte SRAM internal.

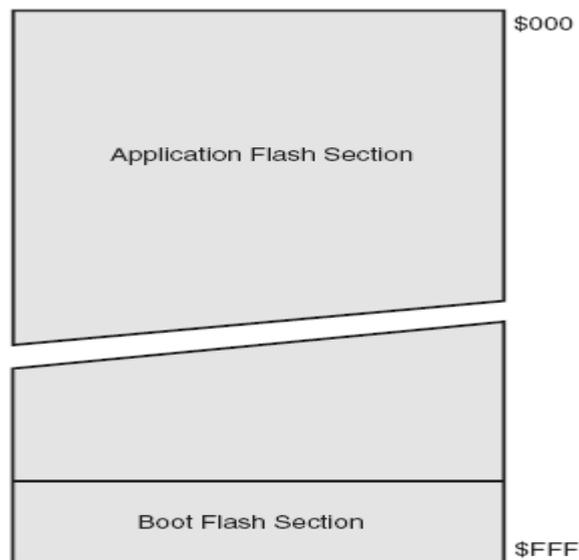
Register untuk keperluan umum menempati space data pada alamat terbawah yaitu \$00 sampai \$1F. Sementara itu register khusus untuk menangani I/O dan kontrol terhadap mikrokontroler menempati 64 alamat berikutnya, yaitu mulai dari \$20 sampai \$5F. Register tersebut merupakan register yang khusus digunakan untuk mengatur fungsi terhadap berbagai peripheral mikrokontroler, seperti kontrol register, timer/counter, fungsi fungsi I/O, dan sebagainya. Register khusus alamat memori secara lengkap dapat dilihat pada tabel dibawah . Alamat memori berikutnya digunakan untuk SRAM 512 byte, yaitu pada lokasi \$60 sampai dengan \$25F.



**Gambar 2.4 Memori data AVR ATmega8535**

(Sumber : Afrie Setiawan, 2011 : 7)

Memori program yang terletak pada Flash Perom tersusun dalam word atau 2 byte karena setiap instruksi memiliki lebar 16-bit atau 32bit. AVR ATmega8535 memiliki 4KByte x 16 Bit Flash Perom dengan alamat mulai dari \$000 sampai \$FFF. AVR tersebut memiliki 12 bit Program Counter (PC) sehingga mampu mengamati isi Flash.



**Gambar 2.5 Memori Program AVR Mikrokontroller ATmega8535**

(Sumber : Syahrul, 2012 : 16)

Selain itu AVR ATmega8535 juga memiliki memori data berupa EEPROM 8-bit sebanyak 512 byte. Alamat EEPROM dimulai dari \$000 sampai \$1FF.

### **2.1.5 Status Register**

Status register adalah register berisi status yang dihasilkan pada setiap operasi yang dilakukan ketika suatu instruksi dieksekusi. SREG merupakan bagian dari inti CPU mikrokontroler.

Status Register ATmega8535 , yaitu:

1. Bit7 --> I (Global Interrupt Enable), Bit harus di Set untuk mengenable semua jenis interupsi.
2. Bit6 --> T (Bit Copy Storage), Instruksi BLD dan BST menggunakan bit T sebagai sumber atau tujuan dalam operasi bit. Suatu bit dalam sebuah register GPR dapat disalin ke bit T menggunakan instruksi BST, dan sebaliknya bit T dapat disalin kembali kesuatu bit dalam register GPR dengan menggunakan instruksi BLD.
3. Bit5 --> H (Half Carry Flag)
4. Bit4 --> S (Sign Bit) merupakan hasil operasi EOR antara flag -N (negatif) dan flag V (komplemen dua overflow).
5. Bit3 --> V (Two's Component Overflow Flag) Bit ini berfungsi untuk mendukung operasi matematis.
6. Bit2 --> N (Negative Flag) Flag N akan menjadi Set, jika suatu operasi matematis menghasilkan bilangan negatif.
7. Bit1 --> Z (Zero Flag) Bit ini akan menjadi Set apabila hasil operasi matematis menghasilkan bilangan 0.
8. Bit0 --> C (Carry Flag) Bit ini akan menjadi set apabila suatu operasi menghasilkan carry.

### 2.1.6 Pengarah Assembler

Pengarah assembler berguna untuk mengubah penunjuk kode assembly. Berikut adalah daftar beberapa sintaks pengarah assembler yang terdapat pada ATmega8535.

1. **.cseg (code segment)** pengarah ini berguna sebagai penunjuk bahwa kode atau ekspresi dibawahnya diletakkan pada memori program pengarah ini biasanya diletakkan setelah pengarah **.deseg**

2. **.db (data byte)** pengarah ini memungkinkan kita dapat meletakkan konstanta seperti serial number, dan lookup table di memory program pada alamat tertentu.
3. **.dw (data word)** pengarah ini seperti data byte, tetapi dalam ukuran word.
4. **.org** digunakan untuk mengeset program counter pada alamat tertentu
5. **.byted** digunakan untuk inisialisasi besar byte yang digunakan pada SRAM untuk label tertentu
6. **.dseg (data segment)** pengarah ini berguna sebagai penunjuk bahwa kode dibawahnya berfungsi untuk melakukan seting SRAM
7. **.def (define)** pengarah ini memungkinkan suatu register dapat didefinisikan.
8. **.equ** berguna untuk memberi nama suatu konstanta yang tidak dapat berubah.
9. **.set** sama seperti **.equ** tetapi konstantanya dapat diubah.
10. **.endm (end macro)** untuk mengakhiri macro.
11. **.include** untuk mengincludekan sebuah file kedalam program agar program lebih cepat dimengerti atau memisahkan kedo dalam dua file terpisah.
12. **.device** sebagai penunjuk jenis AVR yang digunakan.
13. **.exit** sebagai penunjuk agar berhenti melakukan assembly pada file saat ini.
14. **.list** berguna membangkitkan file list.
15. **.listmac** berguna agar penambahan macro ditampilkan pada file list yang dibangkitkan.
16. **.nolist** berguna agar suatu runtun instruksi tidak dimasukkan dalam file list yang dibangkitkan.

## 2.2 Motor DC



**Gambar 2.6** Motor DC  
(Sumber : Repository, 2012)

**Motor DC** merupakan sebuah perangkat elektromagnetis yang mengubah energi listrik menjadi energi mekanik. Energi mekanik ini digunakan untuk, misalnya, memutar impeller pompa, fan atau blower, menggerakkan kompresor, mengangkat bahan, dll. Motor listrik digunakan juga di rumah (mixer, bor listrik, fan angin) dan di industri. Motor listrik kadangkala disebut “kuda kerja” nya industri sebab diperkirakan bahwa motor-motor menggunakan sekitar 70% beban listrik total di industri.

Keuntungan utama motor DC adalah sebagai pengendali kecepatan, yang tidak mempengaruhi kualitas pasokan daya. Motor ini dapat dikendalikan dengan mengatur:

- Tegangan dinamo – meningkatkan tegangan dinamo akan meningkatkan kecepatan
- Arus medan – menurunkan arus medan akan meningkatkan kecepatan.

Motor DC tersedia dalam banyak ukuran, namun penggunaannya pada umumnya dibatasi untuk beberapa penggunaan berkecepatan rendah, penggunaan daya rendah hingga sedang seperti peralatan mesin dan rolling mills, sebab sering terjadi masalah dengan perubahan arah arus listrik mekanis pada ukuran yang lebih besar. Juga, motor tersebut dibatasi hanya untuk penggunaan di area yang bersih dan tidak berbahaya sebab resiko percikan api pada sikatnya. Motor DC juga relatif mahal dibanding motor AC.

### 2.2.1 Mekanisme Kerja Motor DC

Mekanisme kerja untuk seluruh jenis motor secara umum sama

1. Arus listrik dalam medan magnet akan memberikan gaya
2. Jika kawat yang membawa arus dibengkokkan menjadi sebuah lingkaran/*loop*, maka kedua sisi *loop*, yaitu pada sudut kanan medan magnet, akan mendapatkan gaya pada arah yang berlawanan.
3. Pasangan gaya menghasilkan tenaga putar/ *torque* untuk memutar kumparan.
4. Motor-motor memiliki beberapa *loop* pada dinamanya untuk memberikan tenaga putaran yang lebih seragam dan medan magnetnya dihasilkan oleh susunan elektromagnetik yang disebut kumparan medan.

Dalam memahami sebuah motor, penting untuk mengerti apa yang dimaksud dengan beban motor. Beban mengacu kepada keluaran tenaga putar/ *torque* sesuai dengan kecepatan yang diperlukan. Beban umumnya dapat dikategorikan kedalam tiga kelompok :

1. ***Beban torque konstan*** adalah beban dimana permintaan keluaran energinya bervariasi dengan kecepatan operasinya namun *torque* nya tidak bervariasi. Contoh beban dengan *torque* konstan adalah *conveyors*, *rotary kilns*, dan pompa *displacement* konstan.
2. ***Beban dengan variabel torque*** adalah beban dengan *torque* yang bervariasi dengan kecepatan operasi. Contoh beban dengan variabel *torque* adalah pompa sentrifugal dan fan (*torque* bervariasi sebagai kwadrat kecepatan).
3. ***Beban dengan energi konstan*** adalah beban dengan permintaan *torque* yang berubah dan berbanding terbalik dengan kecepatan. Contoh untuk beban dengan daya konstan adalah peralatan-peralatan mesin.

### 2.2.2 Komponen Utama Motor DC

Motor arus searah, sebagaimana namanya, menggunakan arus langsung yang tidak langsung/*direct-unidirectional*. Motor DC digunakan pada penggunaan khusus dimana diperlukan penyalaan *torque* yang tinggi atau percepatan yang tetap untuk kisaran kecepatan yang luas.

Sebuah motor DC yang memiliki tiga komponen utama:

1. **Kutub medan.** Secara sederhana digambarkan bahwa interaksi dua kutub magnet akan menyebabkan perputaran pada motor DC. Motor DC memiliki kutub medan yang stasioner dan dinamo yang menggerakkan *bearing* pada ruang diantara kutub medan. Motor DC sederhana memiliki dua kutub medan: kutub utara dan kutub selatan. Garis magnetik energi membesar melintasi bukaan diantara kutub-kutub dari utara ke selatan. Untuk motor yang lebih besar atau lebih kompleks terdapat satu atau lebih elektromagnet. Elektromagnet menerima listrik dari sumber daya dari luar sebagai penyedia struktur medan.
2. **Dinamo.** Bila arus masuk menuju dinamo, maka arus ini akan menjadi elektromagnet. Dinamo yang berbentuk silinder, dihubungkan ke as penggerak untuk menggerakkan beban. Untuk kasus motor DC yang kecil, dinamo berputar dalam medan magnet yang dibentuk oleh kutub-kutub, sampai kutub utara dan selatan magnet berganti lokasi. Jika hal ini terjadi, arusnya berbalik untuk merubah kutub-kutub utara dan selatan dinamo.
3. **Commutator.** Komponen ini terutama ditemukan dalam motor DC. Kegunaannya adalah untuk membalikan arah arus listrik dalam dinamo. *Commutator* juga membantu dalam transmisi arus antara dinamo dan sumber daya.

### 2.2.3 Jenis Motor DC

#### a) Motor DC Sumber Daya Terpisah/ *Separately Excited*

Jika arus medan dipasok dari sumber terpisah maka disebut motor DC sumber daya terpisah/ *separately excited*.

#### b) Motor DC Sumber Daya Sendiri/ *Self Excited: motor shunt*

Pada motor *shunt*, gulungan medan (medan *shunt*) disambungkan secara paralel dengan gulungan dinamo. Oleh karena itu total arus dalam jalur merupakan penjumlahan arus medan dan arus dinamo.

Berikut tentang kecepatan motor *shunt* :

1. Kecepatan pada prakteknya konstan tidak tergantung pada beban (hingga *torque* tertentu setelah kecepatannya berkurang, lihat Gambar 4) dan oleh

karena itu cocok untuk penggunaan komersial dengan beban awal yang rendah, seperti peralatan mesin.

2. Kecepatan dapat dikendalikan dengan cara memasang tahanan dalam susunan seri dengan dinamo (kecepatan berkurang) atau dengan memasang tahanan pada arus medan (kecepatan bertambah).

**c) Motor DC daya sendiri: motor seri**

Dalam motor seri, gulungan medan (medan *shunt*) dihubungkan secara seri dengan gulungan dinamo (A) seperti ditunjukkan dalam gambar 5. Oleh karena itu, arus medan sama dengan arus dinamo. Berikut tentang kecepatan motor seri (Rodwell International Corporation, 1997; L.M. Photonics Ltd, 2002)

1. Kecepatan dibatasi pada 5000 RPM
2. Harus dihindarkan menjalankan motor seri tanpa ada beban sebab motor akan mempercepat tanpa terkendali.

Motor-motor seri cocok untuk penggunaan yang memerlukan *torque* penyalan awal yang tinggi, seperti derek dan alat pengangkat *hoist*.

**d) Motor DC Kompon/Gabungan**

Motor Kompon DC merupakan gabungan motor seri dan *shunt*. Pada motor kompon, gulungan medan (medan *shunt*) dihubungkan secara paralel dan seri dengan gulungan dynamo (A) seperti yang ditunjukkan dalam gambar 6. Sehingga, motor kompon memiliki *torque* penyalan awal yang bagus dan kecepatan yang stabil. Makin tinggi persentase penggabungan (yakni persentase gulungan medan yang dihubungkan secara seri), makin tinggi pula *torque* penyalan awal yang dapat ditangani oleh motor ini. Contoh, penggabungan 40-50% menjadikan motor ini cocok untuk alat pengangkat *hoist* dan derek, sedangkan motor kompon yang standar (12%) tidak cocok (myElectrical, 2005).

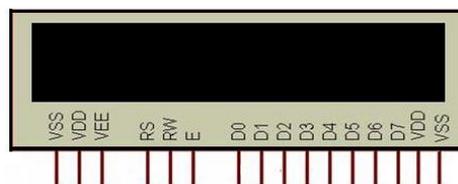
### **2.3. LCD (Liquid Crystal Display)**

Kegunaan LCD banyak sekali dalam perancangan suatu sistem dengan menggunakan mikrokontroler. LCD (Liquid Crystal Display) dapat berfungsi untuk menampilkan suatu nilai hasil sensor, menampilkan teks, atau menampilkan

menu pada aplikasi mikrokontroler. Pada praktek proyek ini, LCD yang digunakan adalah LCD 16 x 2 yang artinya lebar display 2 baris 16 kolom dengan 16 Pin konektor.

Adapun konfigurasi dan deskripsi dari pin-pin LCD antara lain:

1. **VCC ( pin 1)**  
Merupakan sumber tegangan +5V
2. **GND 0V ( pin 2)**  
Merupakan sambungan ground
3. **VEE (pin 3)**  
Merupakan input tegangan Kontras LCD
4. **RS Register Select (pin 4)**  
Merupakan register pilihan 0 = Register Perintah, 1 = Register Data
5. **R/W (pin 5)**  
Merupakan read select , 1 = Read, 0 = Write
6. **Enable Clock LCD (pin 6)**  
Merupakan masukan logika 1 setiap kali pengiriman atau pembacaan data
7. **D0 – D7 ( pin 7 – pin 14)**  
Merupakan Data Bus 1 – 7 ke port
8. **Anoda (pin 15)**  
Merupakan masukan Tegangan positif backlight
9. **Katoda (pin 16)**  
Merupakan masukan Tegangan negatif backlight



**Gambar 2.7 LCD (Liquid Cristal Display)**

(Sumber : Setiawan, 2010)

Display karakter pada LCD diatur oleh pin EN, RS dan RW. Jalur EN dinamakan Enable. Jalur ini digunakan untuk memberitahu LCD bahwa sebuah data sedang dikirimkan. Untuk mengirimkan data ke LCD, maka melalui program EN harus dibuat logika low “0” dan set pada dua jalur kontrol yang lain RS dan RW. Ketika dua jalur yang lain telah siap, set EN dengan logika “1” dan tunggu dan berikutnya di set.

#### **2.4. Modul LCD (*Liquid Crystal Display*) M1632**

M1632 merupakan modul LCD matrix dengan konfigurasi 16 karakter dan 2 baris dengan setiap karakternya dibentuk oleh 8 baris pixel dan 5 kolom pixel (1 baris pixel terakhir adalah kursor). HD44780 ini sudah tersedia dalam Modul M1632 yang dikeluarkan oleh Hitachi, Hyundai dan modul-modul M1632 lainnya.

HD44780 sebetulnya merupakan mikrokontroler yang dirancang khusus untuk mengendalikan LCD dan mempunyai kemampuan untuk mengatur proses *scanning* pada layar LCD yang terbentuk oleh 16 COM dan 40 SEG sehingga mikrokontroler/perangkat yang mengakses modul LCD ini tidak perlu lagi mengatur proses *scanning* pada layar LCD. Mikrokontroler ini hanya mengirimkan data-data yang merupakan karakter yang akan ditampilkan pada LCD atau perintah yang mengatur proses tampilan pada LCD saja.

##### **2.4.1. Struktur Memori LCD**

Modul LCD M1632 memiliki beberapa jenis memori yang digunakan untuk menyimpan atau memproses data-data yang akan ditampilkan pada layar LCD. Setiap jenis memori mempunyai fungsi-fungsi tersendiri.

###### **a. DDRAM**

DDRAM merupakan memori tempat karakter yang ditampilkan berada. Contohnya, karakter “A” atau 41h yang ditulis pada alamat 00 akan tampil pada baris pertama dan kolom pertama dari LCD. Apabila karakter tersebut ditulis di alamat 40h, karakter tersebut akan tampil pada baris kedua kolom pertama dari LCD.

### b. CGRAM

CGRAM adalah memori untuk menggambarkan pola sebuah karakter dan bentuk karakter dapat diubah-ubah sesuai keinginan. Akan tetapi isi memori akan hilang saat power supply tidak aktif sehingga pola karakter akan hilang.

### c. CGROM

CGROM adalah memori untuk menggambarkan pola sebuah karakter dan pola tersebut sudah ditentukan secara permanen dari HD44780 sehingga pengguna tidak dapat mengubah lagi. Oleh karena ROM bersifat permanen, pola karakter tersebut tidak akan hilang walaupun power supply tidak aktif.

## 2.5 Sensor Infra Red (infra merah)

Cahaya infra merah merupakan cahaya yang tidak tampak. Jika dilihat dengan spektroskop cahaya maka radiasi cahaya infra merah akan terlihat pada spektrum elektromagnet dengan panjang gelombang di atas panjang gelombang cahaya merah. Radiasi inframerah memiliki panjang gelombang antara 700 nm sampai 1 mm dan berada pada spektrum berwarna merah. Dengan panjang gelombang ini maka cahaya infra merah tidak akan terlihat oleh mata namun radiasi panas yang ditimbulkannya masih dapat dirasakan/dideteksi.

Pada dasarnya komponen yang menghasilkan panas juga menghasilkan radiasi infra merah termasuk tubuh manusia maupun tubuh binatang. Cahaya infra merah, walaupun mempunyai panjang gelombang yang sangat panjang tetap tidak dapat menembus bahan-bahan yang tidak dapat melewatkan cahaya yang nampak sehingga cahaya infra merah tetap mempunyai karakteristik seperti halnya cahaya yang nampak oleh mata.

Pada pembuatan komponen yang dikhususkan untuk penerima infra merah, lubang untuk menerima cahaya (*window*) sudah dibuat khusus sehingga dapat mengurangi interferensi dari cahaya non-infra merah. Oleh sebab itu sensor infra merah yang baik biasanya memiliki jendela (pelapis yang terbuat dari silikon) berwarna biru tua keungu-unguan. Sensor ini biasanya digunakan untuk aplikasi infra merah yang digunakan diluar rumah (*outdoor*).

Sinar infra merah yang dipancarkan oleh pemancar infra merah tentunya mempunyai aturan tertentu agar data yang dipancarkan dapat diterima dengan

baik pada penerima. Oleh karena itu baik di pengirim infra merah maupun penerima infra merah harus mempunyai aturan yang sama dalam mentransmisikan (bagian pengirim) dan menerima sinyal tersebut kemudian mendekodekannya kembali menjadi data biner (bagian penerima). Komponen yang dapat menerima infra merah ini merupakan komponen yang peka cahaya yang dapat berupa dioda (photodioda) atau transistor (phototransistor). Komponen ini akan merubah energi cahaya, dalam hal ini energi cahaya infra merah, menjadi pulsa-pulsa sinyal listrik. Komponen ini harus mampu mengumpulkan sinyal infra merah sebanyak mungkin sehingga pulsa-pulsa sinyal listrik yang dihasilkan kualitasnya cukup baik.

## 2.6 Fotodioda

Fotodioda adalah jenis dioda yang berfungsi mendeteksi cahaya. Fotodioda merupakan sensor cahaya semikonduktor yang dapat mengubah besaran cahaya menjadi besaran listrik. Fotodioda merupakan sebuah dioda dengan sambungan pn yang dipengaruhi cahaya dalam kerjanya. Cahaya yang dapat dideteksi oleh fotodioda ini mulai dari cahaya infra merah, cahaya tampak, ultra ungu sampai dengan sinar-X. Aplikasi fotodioda mulai dari penghitung kendaraan di jalan umum secara otomatis, pengukur cahaya pada kamera serta beberapa peralatan di bidang medis.

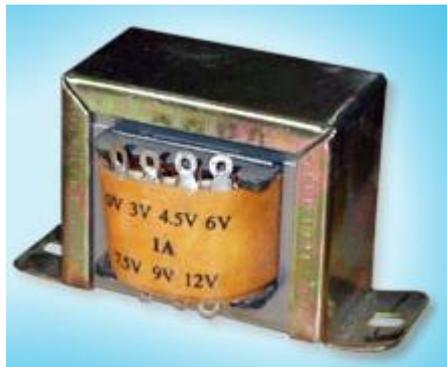
Prinsip kerja dari fotodioda jika sebuah sambungan-pn dibias maju dan diberikan cahaya padanya maka pertambahan arus sangat kecil sedangkan jika sambungan pn dibias mundur arus akan bertambah cukup besar. Cahaya yang dikenakan pada fotodioda akan mengakibatkan terjadinya pergeseran foton yang akan menghasilkan pasangan electron-hole di kedua sisi dari sambungan. Ketika elektron-elektron yang dihasilkan itu masuk ke pita konduksi maka elektron-elektron itu akan mengalir ke arah positif sumber tegangan sedangkan hole yang dihasilkan mengalir ke arah negatif sumber tegangan sehingga arus akan mengalir di dalam rangkaian. Besarnya pasangan elektron ataupun hole yang dihasilkan tergantung dari besarnya intensitas cahaya yang dikenakan pada fotodioda.

Alat yang mirip dengan fotodioda adalah fototransistor (*Phototransistor*). Fototransistor ini pada dasarnya adalah jenis transistor bipolar yang menggunakan kontak (*junction*) base-collector untuk menerima cahaya. Komponen ini

mempunyai sensitivitas yang lebih baik jika dibandingkan dengan Fotodiode. Hal ini disebabkan karena elektron yang ditimbulkan oleh foton cahaya pada *junction* ini diinjeksikan di bagian Base dan diperkuat di bagian Kolektornya. Namun demikian, waktu respons dari fototransistor secara umum akan lebih lambat dari pada fotodiode.

## . 2.7 Transformator

Transformator (trafo) adalah alat yang digunakan untuk menaikkan atau menurunkan tegangan bolak-balik (AC). Transformator terdiri dari 3 komponen pokok yaitu: kumparan pertama (primer) yang bertindak sebagai input, kumparan kedua (sekunder) yang bertindak sebagai output, dan inti besi yang berfungsi untuk memperkuat medan magnet yang dihasilkan.



**Gambar 2.8 Transformator atau Trafo**

(Sumber : Repository, 2012)

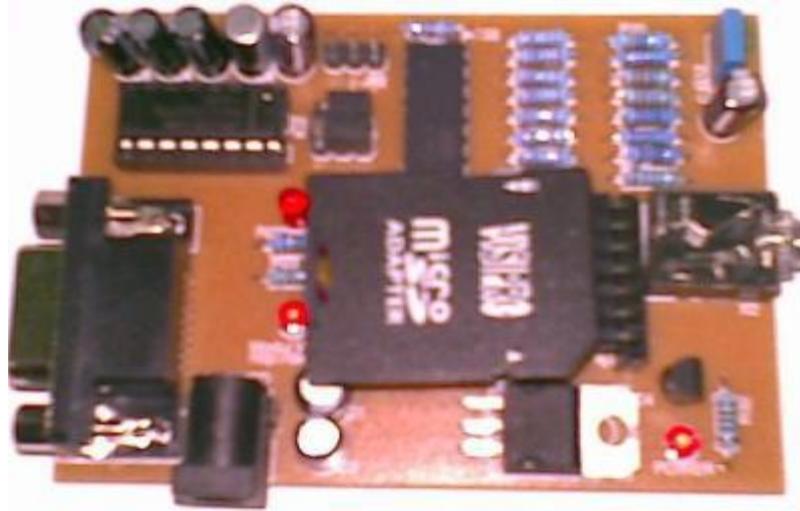
### 2.7.1 Penggunaan Transformator

Transformator (trafo) digunakan pada peralatan listrik terutama yang memerlukan perubahan atau penyesuaian besarnya tegangan bolak-balik. Misal radio memerlukan tegangan 12 volt padahal listrik dari PLN 220 volt, maka diperlukan transformator untuk mengubah tegangan listrik bolak-balik 220 volt menjadi tegangan listrik bolak-balik 12 volt.

## 2.8 Wav Player

WAV merupakan standar suara de-facto di Windows. Awalnya hasil ripping dari CD direkam dalam format ini sebelum dikonversi ke format lain.

Namun sekarang tahap ini sering dilewati karena file dalam format ini biasanya tidak dikompresi dan karenanya berukuran besar. File WAV adalah file audio. Waveform Audio Format (WAVE, atau lebih dikenal sebagai WAV karena ekstensi nama file-nya), adalah format standar file audio Microsoft dan IBM untuk menyimpan sebuah bitstream audio pada PC.



**Gambar 2.9 Wav Player**

(Sumber : Dany , 2009 :17)

## 2.9 Bahasa Pemrograman ATmega8535

Pemrograman mikrokontroler ATmega8535 dapat menggunakan *low level language (assembly)* dan *high level language* (C, Basic, Pascal, JAVA,dll) tergantung *compiler* yang digunakan (Widodo Budiharto, 2006). Bahasa *Assembler* mikrokontroler AVR memiliki kesamaan instruksi, sehingga jika pemrograman satu jenis mikrokontroler AVR sudah dikuasai, maka akan dengan mudah menguasai pemrograman keseluruhan mikrokontroler jenis mikrokontroler AVR. Namun bahasa *assembler* relatif lebih sulit dipelajari dari pada bahasa C.

Untuk pembuatan suatu proyek yang besar akan memakan waktu yang lama serta penulisan programnya akan panjang. Sedangkan bahasa C memiliki keunggulan dibanding bahasa *assembler* yaitu *independent* terhadap *hardware* serta lebih mudah untuk menangani *project* yang besar. Bahasa C memiliki keuntungan-keuntungan yang dimiliki bahasa *assembler* (bahasa mesin), hampir semua operasi yang dapat dilakukan oleh bahasa mesin, dapat dilakukan dengan

bahasa C dengan penyusunan program yang lebih sederhana dan mudah. Bahasa C terletak diantara bahasa pemrograman tingkat tinggi dan *assembly* (Agus Bejo,2007).

### **2.10 Program BASCOM AVR**

BASCOM-AVR merupakan basic compiler AVR. BASCOM-AVR termasuk dalam program mikrokontroler buatan MCS *Electronics* yang mengadaptasi bahasa tingkat tinggi yang sering digunakan (Bahasa Basic). BASCOM-AVR (Basic Compiler) merupakan software compailer dengan menggunakan bahasa basic yang dibuat untuk melakukan pemrograman chip-chip mikrokontroler tertentu, salah satunya Atmega8535 BASCOM-AVR adalah program Basic Compiler berbasis windows untuk mikrokontroler keluarga AVR seperti Atmega8535, Atmega8515 dan yang lainnya.

BASCOM AVR merupakan pemrograman dengan bahasa tingkat tinggi . BASIC yang dikembangkan dan dikeluarkan oleh AVR Electronic. Program ini digunakan dalam pengisian mikrokontroller. Kompiler ini cukup lengkap karena dilengkapi simulator untuk LED, LCD dan monitor untuk komunikasi serial. Selain itu bahasa BASIC jauh lebih mudah dipahami dibandingkan bahasa pemrograman lainnya.

Dengan menggunakan bahasa pemrograman tingkat tinggi, maka pemrograman mendapatkan banyak kemudahan dalam mengatur sistem kerja dari mikrokontroler, dapat dilihat juga pada tabel 3.1 keterangan ikon-ikon dari program BASCOM-AVR:

**Tabel 2.1 Keterangan Ikon-Ikon Dari Program BASCOM-AVR:**

Ikon	Nama	Fungsi	Shortcut
	File New	Membuat file baru	Ctrl + N
	Open File	Membuka file	Ctrl + O
	File Close	Menutup program yang dibuka	-
	File Save	Menyimpan file	Ctrl + S
	Save As	Menyimpan dengan nama lain	-
	Print Preview	Melihat tampilan sebelum dicetak	-
	Print	Mencetak dokumen	Ctrl + P
	Exit	Keluar dari program	-
	Program Compile	Mengompile program yang dibuat. Outputnya bisa berupa *.hex, *.bin, dan lain-lain	F7
	Program Chip	Memasukkan program ke chip	F4
	Syntax Check	Memeriksa kesalahan bahasa	Ctrl + F7
	Show Result	Menampilkan hasil kompilasi program	Ctrl + W

### 2.10.1. Bahasa BASIC Menggunakan BASCOM-AVR

#### Pemrograman Dengan Bascom AVR

Pada umumnya bahasa yang dipergunakan untuk memprogram mikrokontroler adalah bahasa Assembly. Bahasa Assembly adalah bahasa pemrograman tingkat menengah, dimana program yang dibuat lebih mendekati bahasa mesin, sehingga pemanfaatan memori dapat dilakukan secara optimal, namun di sisi lain pemrogramannya menjadi relatif sulit. Karena bahasa yang dipergunakan Bascom, yaitu Basic, adalah bahasa tingkat tinggi, maka pemrograman menggunakan Bascom sangatlah mudah untuk dipelajari. Sintaksnya tidak jauh berbeda dari Basic pada umumnya, misalnya do-loop, for-next, while-wend, goto, gosub dan sebagainya. Selain itu Bascom dilengkapi dengan fungsifungsi khusus, misalnya LCD untuk menampilkan karakter pada

LCD, PRINT untuk mengirimkan karakter ke PC melalui kabel RS232, SHIFTIN dan SHIFTOUT untuk komunikasi serial sinkron dan lain sebagainya. Fungsi-fungsi khusus tersebut jika dituliskan dalam bahasa Assembly akan menjadi lebih panjang dan rumit, terutama karena kita harus mengetahui register-register yang ada pada mikrokontroler.

### 2.10.2 Kontruksi bahasa BASIC pada BASCOM-AVR

Setiap bahasa pemrograman mempunyai standar penulisan program. Konstruksi dari program bahasa BASIC harus mengikuti aturan sebagai berikut:

```
$regfile = "header"
'inisialisasi
'deklarasi variabel
'deklarasi konstanta
Do
'pernyataan-pernyataan
Loop
end
```

#### Pengarah preprosesor

\$regfile = "m16def.dat" merupakan pengarah preprosesor bahasa BASIC yang memerintahkan untuk menyisipkan file lain, dalam hal ini adalah *file* m16def.dat yang berisi deklarasi register dari mikrokontroler ATmega 16, pengarah preprosesor lainnya yang sering digunakan ialah sebagai berikut:

```
$crystal = 12000000 'menggunakan crystal clock 12 MHz
$baud = 9600        'komunikasi serial dengan baudrate 9600
$eeprom            'menggunakan fasilitas eeprom
```

### 2.10.3 Karakter Pada Bascom

Karakter pada Bascom dipergunakan untuk membentuk label, keyword, variabel, dan operator, yang kesemuanya akan membentuk suatu program. Pada dasarnya karakter pada Bascom terdiri dari karakter huruf (A-Z) dan karakter angka (0-9). Beberapa karakter pada Bascom yang dipergunakan secara khusus terdapat pada tabel 2-2.

Tabel 2-1 Karakter khusus pada Bascom

Karakter	Deskripsi
ENTER	Ganti baris
	Blank (spasi)
'	Single quotation mark (apostrophe)
*	Asterisks (simbol perkalian)
+	Plus sign (simbol penjumlahan)
,	Comma
-	Minus sign (simbol pengurangan)
.	Period (decimal point)
/	Slash (simbol pembagian) ditangani seperti \
:	Colon
"	Double quotation mark
;	Semicolon
<	Kurang dari
=	Sama dengan
>	Lebih dari
\	Backslash (simbol pembagian integer)

### 2.10.4 Tipe Data

Setiap variabel pada Bascom mempunyai tipe data yang menunjukkan kapasitas dan jenis data yang dapat disimpan pada variabel tersebut. Hal ini berpengaruh pada seberapa besar memori yang diperlukan untuk menyimpan variabel tersebut. Tabel 2-3 menunjukkan tipe data pada Bascom beserta ukuran dan rentangnya.

Tabel 2-2 Tipe data pada Bascom

Tipe Data	Ukuran (byte)	Rentang
Bit	1/8	0 atau 1
Byte	1	0 – 255
Integer	2	-32768 – +32767
Word	2	0 – 65535
Long	4	-2147483648 – +2147483647
Single	4	
String	maksimum 254 byte	

### 2.10.5 Variabel

Variabel adalah simbol yang digunakan untuk mewakili suatu nilai. Variabel digunakan sebagai tempat penyimpanan data atau penampung data sementara. Variabel numerik hanya dapat diisi nilai numerik (bit, byte, integer, word, long, dan single). Isi dari suatu variabel numerik dapat berupa :

1. Suatu nilai konstan

$$A = 5$$

$$C = 1.1$$

2. Nilai variabel numerik lain

$$A = B$$

3. Nilai yang didapat dengan mengkombinasikan variabel, konstan, dan operator

$$\text{Temp} = A + 5$$

Pada Bascom terdapat beberapa aturan mengenai penamaan suatu variabel, yaitu:

Nama suatu variabel maksimum terdiri atas 32 karakter dan dapat berupa huruf ataupun angka.

4. Karakter pertama variabel haruslah berupa huruf.
5. Nama variabel tidak boleh menggunakan kata-kata yang dipergunakan Bascom sebagai perintah, pernyataan, register dan operator (misal AND, OR, DIM, P1 TIMER0 dan lain sebagainya)

Sebelum digunakan suatu variabel haruslah dideklarasikan terlebih dahulu tipe data yang dipergunakan dengan menggunakan pernyataan DIM

```
DIM A As Byte
```

```
DIM Nama1 As Byte, Nama2 As Integer
```

```
DIM Kata As String*10
```

Selain menggunakan DIM, variabel dapat juga ditentukan tipe datanya menggunakan

```
DEFBYTE, DEFINT, DEFBIT, dan DEFWORD
```

```
DEFBYTE A
```

```
DEFWORD B;C;D
```

Suatu variabel dapat mempunyai nama lain atau alias. Umumnya alias

digunakan untuk mengganti variabel standar dengan nama yang lebih mudah diingat. Hal ini akan berguna pada saat membuat program yang panjang dan kompleks, jika terdapat perubahan penggunaan pin atau port, cukup diganti pada pernyataan Alias.

Saklar1 Alias PB.0

LED1 Alias PD.0

### 2.10.6 Konstanta

Konstanta adalah variabel yang mempunyai nilai konstan selama program dijalankan. Untuk mendeklarasikan suatu konstanta dapat digunakan dua cara, yaitu menggunakan Dim atau Const

Dim A As Const 5

Dim B1 As Const &B1001

Const Cbyte = &HF

Const Cint = -1000

Const Csingle = 1.1

Const Cstring = "tes"

### 2.10.7 Larik

Larik atau array adalah kumpulan variabel dengan nama dan tipe data yang sama. Untuk membedakan satu variabel dengan variabel lainnya digunakan indeks. Indeks haruslah berupa angka dengan tipe data byte, integer atau word, dengan nilai minimal 1 (bukan 0). Pendeklarasian larik mirip seperti variabel biasa, hanya ditambahkan jumlah komponen lariknya.

Dim a(10) as byte

### 2.10.8 Sistem Bilangan

Pada pemrograman mikrokontroler terdapat 3 sistem bilangan yang sering digunakan, yaitu desimal (basis 10), biner (basis 2), dan heksadesimal (basis 16). Cara penulisan bilangan pada Bascom disesuaikan dengan sistem bilangan yang digunakan,

yaitu :

- Untuk bilangan desimal tidak didahului angka ataupun huruf lain
- Untuk bilangan biner didahului dengan &B

- Untuk bilangan heksadesimal didahului dengan &H

Contoh : 240 (bilangan desimal), &B11110000 (bilangan biner), &HF0 (bilangan heksadesimal)

### 2.10.9 Operator

Operator digunakan untuk melakukan operasi terhadap bilangan. Pada Bascom operator dibedakan menjadi operator aritmetik, operator relasional, dan operator logika. Operator aritmatik adalah operator yang digunakan dalam kalkulasi, yaitu + (penjumlahan), - (pengurangan), \* (perkalian), / (pembagian), \ (pembagian integer), MOD (modulo = sisa dari pembagian). Operator relasional digunakan untuk membandingkan dua nilai, yang memberikan hasil benar (1) atau salah (0) dan dapat digunakan untuk membuat keputusan.

Tabel 2-3 Macam operator relasi pada Bascom

Operator	Relasi	Pernyataan
=	Sama dengan	X=Y
<>	Tidak sama dengan	X<>Y
<	Lebih kecil	X<Y
>	Lebih besar	X>Y
<=	Lebih kecil atau sama dengan	X<=Y
>=	Lebih besar atau sama dengan	X>=Y

Operator logika digunakan untuk menguji suatu pola bit tertentu, manipulasi bit atau operator Boolean. Misal operator AND dapat digunakan untuk mengabaikan semua bit dalam suatu byte kecuali satu bit untuk memantau status bit tersebut

Tabel 2-4 Macam operator logika pada Bascom

Operator	Makna
NOT	Komplemen/inverter
AND	Konjungsi (dan)
OR	Disjungsi (atau)
EXOR	EXclusive OR

### 2.10.10 Pernyataan Bersyarat

Pada Bascom terdapat beberapa pernyataan bersyarat yang sering digunakan yaitu If – Then, If – Then – Elseif, dan Select – Case

***Sintaksis If – Then***

```
If <syarat> Then  
  <Pernyataan 1>  
  <Pernyataan 2>  
Else  
  <Pernyataan 3>  
  <Pernyataan 4>  
End If
```

***Sintaksis If – Then – Elseif***

```
If <syarat1> Then  
  <Pernyataan 1>  
  <Pernyataan 2>  
Elseif <syarat2>Then  
  <Pernyataan 3>  
  <Pernyataan 4>  
Else  
  <Pernyataan 5>  
  <Pernyataan 6>  
End If
```

***Sintaksis Select – Case***

```
Select Case < Variabel>  
  Case < Nilai 1> : <Pernyataan 1>  
  Case < Nilai 2> : <Pernyataan 2>  
  Case Else : <Pernyataan 3>  
End Select
```

**Pernyataan Perulangan (Loop)**

Loop adalah suatu perulangan terhadap perintah atau instruksi sampai mencapai keadaan tertentu (jumlah perulangan tersebut dapat diketahui). Fungsi dari loop sendiri banyak sekali, dan dapat menghemat dalam penulisan program karena program yang sama dapat dilakukan dengan beberapa perintah dan kemudian

diulang-ulang. Terdapat 3 pernyataan perulangan yaitu Do-Loop, While-Wend, dan For-Next.

***Sintaksis Do – Loop***

```
Do  
<Pernyataan 1>  
<Pernyataan 2>  
Loop
```

***Sintaksis While – Wend***

```
While <Syarat>  
<Pernyataan 1>  
<Pernyataan 2>  
Wend
```

***Sintaksis For – Next***

```
FOR <Variabel> = <Nilai Awal> TO/DOWNTO <Nilai Akhir>  
<Pertambahan/Pengurangan>  
<Pernyataan 1>  
<Pernyataan 2>  
Exit:
```

