BAB II

LANDASAN TEORI

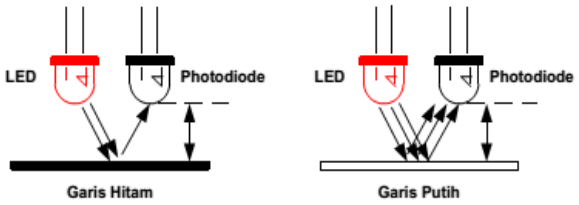
2.1 Sensor *Photodiode*

Photodiode adalah salah satu komponen semikonduktor yang peka terhadap cahaya (*photodetector*). *Photodiode* merupakan sebuah *dioda* dengan sambungan p-n *junction* yang dipengaruhi cahaya dan kerjanya. Cahaya yang dapat dideteksi oleh *photodiode* ini mulai dari cahaya infra merah, cahaya tampak, cahaya ultra sampai dengan sinar-X.



Gambar 2.1 Bentuk Fisik Sensor *Photodiode*

Photodiode akan mengalirkan arus yang sesuai dengan intensitas cahaya yang diterima. Arus ini umumnya teratur terhadap *power density*. Perbandingan antara arus keluaran dengan *power density* disebut sebagai *current responsitivity*. Arus yang dimaksud adalah arus yang mengalir ketika *photodioda* ini disinari langsung oleh cahaya dari *Light Emitting Diode* (LED). LED merupakan salah satu komponen elektronika yang dapat mengeluarkan cahaya bila diberikan bias maju. Di dalam perancangan, jenis LED yang digunakan dapat memancarkan cahaya merah terang sehingga sangat cocok digunakan pada rangkaian sensor *photodiode*. Di dalam rangkaian, LED difungsikan sebagai alat untuk memancarkan cahaya.



Gambar 2.2 Cara Kerja dari Sensor *Photodiode*

Photodiode digunakan sebagai penangkap gelombang cahaya infra merah yang dipancarkan oleh sebuah LED. Besarnya tegangan atau arus listrik yang dihasilkan oleh *photodiode* tergantung pada besar kecilnya radiasi cahaya yang dipancarkan oleh LED tersebut.

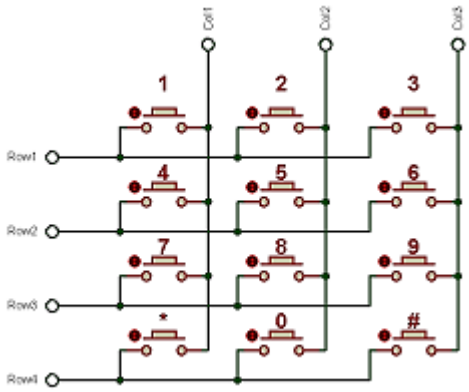
2.2 *Keypad*

Keypad adalah rangkaian tombol yang berfungsi untuk memberi sinyal pada suatu rangkaian dengan menghubungkan *pin-pin* tertentu dari *keypad*. *Keypad* terdiri dari beberapa macam berdasarkan jumlah tombol dan fungsinya. Dalam perancangan ini digunakan *keypad* matriks 3x4 yang bila dilihat dari bentuk fisiknya memiliki 12 sakelar dengan penghubung rangkaian sebanyak 7 *pin*.



Gambar 2.3 Bentuk Fisik (*Keypad* 3x4)

Ketujuh *pin* penghubung tersebut terbagi menjadi dua kelompok yaitu 4 *pin* sebagai *input* dan 3 *pin* sebagai *output*. Maksud dari ketujuh *pin input/output* (I/O) tersebut adalah untuk digunakan sebagai kombinasi penghubung pada rangkaian yang akan disambungkan dengan *keypad* ini. Dimana dalam setiap penekanan satu tombol pada *keypad*, maka akan terjadi kombinasi antara dua buah *pin* dalam pembacaan sinyalnya. Sebenarnya terdapat beberapa teknik untuk pembacaan data dari matriks *keypad* ini, salah satunya adalah dengan teknik *scanning*. Dimana baris atau kolom selalu dipindai (di-*scan*) untuk mendeteksi tombol yang di tekan. Berikut ini adalah rangkaian matriks dari *keypad* ukuran 3x4.



Gambar 2.4 Rangkaian Susunan Matriks (*Keypad* 3x4)

2.3 Sakelar *Push Button Switch*

Sakelar adalah suatu perangkat elektronika yang digunakan untuk menghubungkan atau memutuskan aliran listrik. Sakelar tekan atau *push button switch* merupakan suatu jenis sakelar yang dapat menghubungkan aliran listrik sesaat saja saat ditekan dan setelah dilepas maka akan kembali lagi ke posisi *off*.



Gambar 2.5 Bentuk Fisik Sakelar Tekan (*Push Button Switch*)

2.4 Mikrokontroler

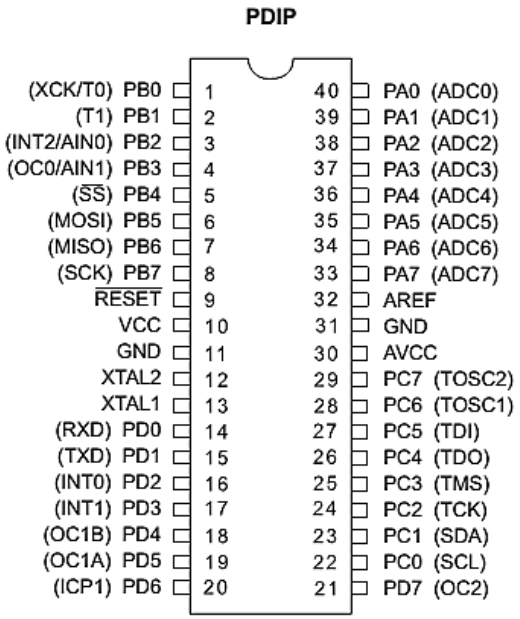
Mikrokontroler adalah prosesor mikro yang yang terdiri CPU ditambah dengan RAM, ROM, I/O *ports*, dan *timer* yang jumlahnya tetap dan dikemas dalam satu chip. Mikrokontroler merupakan salah satu bagian dasar dari suatu sistem komputer. Meskipun mempunyai bentuk yang jauh lebih kecil dari suatu komputer pribadi dan komputer mainframe, mikrokontroler dibangun dari elemen-elemen dasar yang sama. Mikrokontroler AVR ATmega 16 merupakan mikrokontroler 8-*bit* dengan konsumsi daya rendah yang mempunyai arsitektur *Reduced Instruction Set Computer* (RISC). Berbeda dengan mikrokontroler keluarga 8051 yang mempunyai arsitektur *Complex Instruction Set Computer* (CISC).



Gambar 2.6 ATmega 16

ATmega 16 menjalankan sebuah instruksi tunggal dalam satu siklus/*clock* dan memiliki struktur I/O yang cukup lengkap sehingga penggunaan komponen eksternal dapat dikurangi. Mikrokontroler AVR ATmega 16 di desain menggunakan arsitektur *Harvard*, dimana ruang dan jalur *bus* bagi memori program

dipisahkan dengan memori data. Memori program diakses dengan *single-level pipelining*, dimana ketika sebuah instruksi dijalankan, instruksi lain berikutnya akan di *prefetch* dari memori program.



(Sumber : Datasheet ATMEL AVR ATmega 16)
Gambar 2.7 Konfigurasi *Pin* Mikrokontroler AVR ATmega 16

2.4.1 Konfigurasi *Pin* ATmega 16

Mikrokontroler AVR ATmega memiliki 40 *pin* dengan 32 *pin* diantaranya digunakan sebagai *port paralel*. Satu *port paralel* terdiri dari 8 *pin*, sehingga jumlah *port* pada mikrokontroler adalah 4 *port*, yaitu *port* A, *port* B, *port* C dan *port* D. Sebagai contoh adalah *port* A memiliki *pin* antara *port* A.0 sampai *port* A.7, demikian selanjutnya untuk *port* B, *port* C, *port* D. Berikut ini adalah penjelasan mengenai *pin* yang terdapat pada mikrokontroler ATmega 16 :

Tabel 2.1 Deskripsi Pin Mikrokontroler AVR ATmega 16

VCC	Tegangan suplai antara 2,7 – 5,5 *Volt*
GND	*Ground*
RESET	*Input reset* level rendah, pada *pin* ini selama lebih dari panjang pulsa minimum akan menghasilkan *reset* walaupun *clock* sedang berjalan. *Reset* pada *pin* 9 merupakan *reset* dari AVR. Jika pada *pin* ini diberi masukan *low* selama minimal 2 *machine cycle* maka sistem akan di-*reset*
XTAL 1	*Input* penguat *oscillator inverting* dan *input* pada rangkaian operasi *clock internal*
XTAL 2	*Output* dari penguat *oscillator inverting*
AVCC	*Pin* tegangan suplai untuk *port* A dan ADC
AREF	*Pin* referensi tegangan *analog* untuk ADC
AGND	*Pin* untuk *analog ground*

Penjelasan konfigurasi *pin* pada mikrokontroler AVR ATmega 16 yang mempunyai fungsi khusus sebagai berikut :

a. Fungsi *pin* 33 sampai 40 (*port* A), dijelaskan pada Tabel 2.2

Tabel 2.2 Fungsi Khusus *port* A

Port	**Fungsi Khusus**
PA.0	*Input Analog to Digital Converter* (ADC) *port* A.0
PA.1	*Input Analog to Digital Converter* (ADC) *port* A.1
PA.2	*Input Analog to Digital Converter* (ADC) *port* A.2
PA.3	*Input Analog to Digital Converter* (ADC) *port* A.3
PA.4	*Input Analog to Digital Converter* (ADC) *port* A.4
PA.5	*Input Analog to Digital Converter* (ADC) *port* A.5
PA.6	*Input Analog to Digital Converter* (ADC) *port* A.6
PA.7	*Input Analog to Digital Converter* (ADC) *port* A.7

b. Fungsi *pin* 1 sampai 8 (*port* B), dijelaskan pada Tabel 2.3

Tabel 2.3 Fungsi Khusus *port* B

Port	Fungsi Khusus
PB.0	T0 (*Timer/Counter* 0 *External Counter Input*), XCK (USART *External Clock Input/Output*)
PB.1	T1 (*Timer/Counter* 1 *External Counter Input*),
PB.2	AIN0 (*Analog Comparator Positive Input*), INT2 (*External Interrupt* 2 *Input*)
PB.3	AIN1 (*Analog Comparator Negative Input*), OC0 (*Timer/Counter* 0 *Output Compare Match Output*)
PB.4	SS (SPI *Slave Select Input*)
PB.5	MOSI (SPI *Bus Master Output/Slave Input*)
PB.6	MISO (SPI *Bus Master Input/Slave Output*)
PB.7	SCK (SPI *Bus Serial Clock*)

c. Fungsi *pin* 22 sampai 29 (*port* C), dijelaskan pada Tabel 2.4

Tabel 2.4 Fungsi Khusus *port* C

Port	Fungsi Khusus
PC.0	SCL (*Two-wire Serial Bus Clock Line*)
PC.1	SDA (*Two-wire Serial Bus Data Input/Output Line*)
PC.2	TCK (JTAG *Test Clock*)
PC.3	TMS (JTAG *Test Mode Select*)
PC.4	TDI (JTAG *Test Data Output*)
PC.5	TDI (JTAG *Test Data Input*)
PC.6	TOSC1 (*Timer Oscillator Pin* 1)
PC.7	TOSC2 (*Timer Oscillator Pin* 2)

d. Fungsi *pin* 14 sampai 21 (*port* D), dijelaskan pada Tabel 2.5

Tabel 2.5 Fungsi Khusus *port* D

Port	Fungsi Khusus
PD.0	RXD (USART *Input Pin*)
PD.1	TXD (USART *Output Pin*)
PD.2	INT0 (*External Interrupt* 0 *Input*)
PD.3	INT1 (*External Interrupt* 1 *Input*)
PD.4	OC1B (*Timer/Counter* 1 *Output Compare* B *Match Output*)
PD.5	OC1A (*Timer/Counter* 1 *Output* Compare A *Match Output*)
PD.6	ICP1 (*Timer/Counter* 1 *Input Capture Pin*)
PD.7	OC2 (*Timer/Counter* 2 *Output Compare Match Output*)

2.4.2 Arsitektur Mikrokontroler AVR ATmega 16

Mikrokontroler AVR ATmega 16 memiliki beberapa fitur utama atau spesifikasi yang menjadikannya sebuah solusi yang efektif untuk berbagai keperluan diantaranya adalah :

1. Memiliki 131 instruksi yang sebagian besar dieksekusi dalam satu siklus/*cycle*.
2. 32 *register* umum yang terhubung dengan *Arithmetic Logic Unit* (ALU).
3. Memiliki kemampuan untuk mencapai *throughput* sekitar 16 *Million Instruction Per Second* (MIPS) pada 16 MHz.
4. Memiliki memori dengan kapasitas 16 *Kbytes In-System Self-Programmable Flash* dengan kemampuan (10.000 siklus hapus/tulis).
5. Memiliki memori dengan kapasitas 512 *bytes In-System Programmable* EEPROM dengan kemampuan (100.000 siklus hapus/tulis).

6. Memiliki dua *Timer/Counter* 8-*bit* untuk *Separate Prescaler* dan Compare Modes.
7. Memiliki satu *Timer/Counter* 16-*bit* untuk *Separate Prescaler*, *Compare Mode* dan *Capture Mode*.
8. *Real Time Counter* (RTC) dengan osilator terpisah.
9. Memiliki *Pulse Width Modulation* (PWM) sebanyak 4-*Channel*.
10. Memiliki *Analog to Digital Converter* (ADC) internal dengan fidelitas 10-*bit* sebanyak 8-*Channel*.
11. *Full Duplex Universal Synchronous Asynchronous Receiver Transmitter* (USART) untuk komunikasi *serial* terprogram.
12. *Watchdog Timer* yang dapat diprogram dengan osilator internal.
13. Memiliki 1 *Kbytes internal Static Random Access Memory* (SRAM) yang digunakan untuk menyimpan data sementara dari program *flash*.
14. 32 jalur I/O (*Input-Output*) yang terpisah dan dikelompokkan dalam empat *port* yaitu : *port* A, *port* B, *port* C dan *port* D.
15. Memiliki *External/Internal Interrupt Sources* yang digunakan untuk melakukan proses interupsi secara internal maupun eksternal.
16. Memiliki enam pilihan mode *Sleep* yaitu : *Idle*, ADC *Noise Reduction*, *Power-save*, *Power Down*, *Standby* dan *Extended Standby*. Keenam mode *Sleep* ini dapat menghemat penggunaan daya listrik.
17. Memiliki *Analog Comparator*.
18. Frekuensi *Clock* maksimum 16 MHz.
19. Dapat beroperasi pada tegangan 2,7 – 5,5 *Volt*.

2.5 *Liquid Crystal Display* (LCD)

Liquid Crystal Display (LCD) merupakan komponen yang dapat menampilkan suatu nilai hasil sensor, menampilkan teks atau menampilkan menu pada aplikasi mikrokontroler. Jenis LCD M1632 yang merupakan modul LCD dengan tampilan 16 baris x 2 kolom (16x2) dengan konsumsi daya rendah. Modul tersebut dilengkapi dengan mikrokontroler yang di desain khusus untuk mengendalikan LCD. Mikrokontroler yang digunakan untuk mengendalikan LCD adalah mikrokontroler HD44780 yang memiliki memori *Character Generator Read Only Memory* (CGROM), *Character Generator Random Access Memory* (CGRAM) dan *Display Data Random Access Memory* (DDRAM). Berikut ini merupakan bagian-bagian dari LCD :

a. *Display Data Random Access Memory* (DDRAM)

   DDRAM merupakan memori tempat karakter yang ditampilkan berada. Contoh, untuk karakter ‘L’ atau 4CH yang ditulis pada alamat 00, karakter tersebut akan tampil pada baris pertama dan kolom pertama pada LCD. Apabila karakter tersebut ditulis pada alamat 40, maka karakter tersebut akan tampil pada baris kedua kolom pertama dari LCD.

b. *Character Generator Random Access Memory* (CGRAM)

   CGRAM merupakan memori untuk membuat bentuk karakter yang dapat diubah-ubah sesuai keinginan. Karakter yang disimpan di CGRAM akan hilang apabila tidak ada sumber tegangan suplai, karena memori RAM bersifat tidak permanen.

c. *Character Generator Read Only Memory* (CGROM)

CGROM merupakan memori yang menyimpan karakter-karakter yang sudah permanen di dalam LCD, sehingga tidak dapat diubah-ubah lagi bentuknya oleh pengguna. Namun karena ROM bersifat permanen, pola karakter tersebut tidak akan hilang saat tidak ada sumber tegangan suplai.



Gambar 2.8 Bentuk Fisik LCD (*Character* 16x2)

Bila dilihat dari bentuk fisiknya, LCD karakter 16x2 ini memiliki 16 *pin* yang memiliki fungsi masing-masing. Berikut ini adalah konfigurasi *pin* dari LCD karakter 16x2 yaitu :

Tabel 2.6 Fungsi dan Konfigurasi *Pin* LCD 16x2

Pin	Simbol	Level	Deskripsi
1	VSS	0V	*Ground*
2	VDD	+5V	*Supply Voltage for Logic*
3	VO	(Variable)	*Operating Voltage for LCD*
4	RS	H/L	H : *Data,* L : *Instruction Code*
5	R/W	H/L	H : *Read,* L : *Write*
6	E	H, H->L	*Chip Enable Signal*
7	DB0	H/L	*Data bit 0*
8	DB1	H/L	*Data bit 1*
9	DB2	H/L	*Data bit 2*
10	DB3	H/L	*Data bit 3*
11	DB4	H/L	*Data bit 4*
12	DB5	H/L	*Data bit 5*
13	DB6	H/L	*Data bit 6*
14	DB7	H/L	*Data bit 7*
15	A	4,2 – 4,6 V	LED +
16	K	0V	LED -

2.6 Motor DC

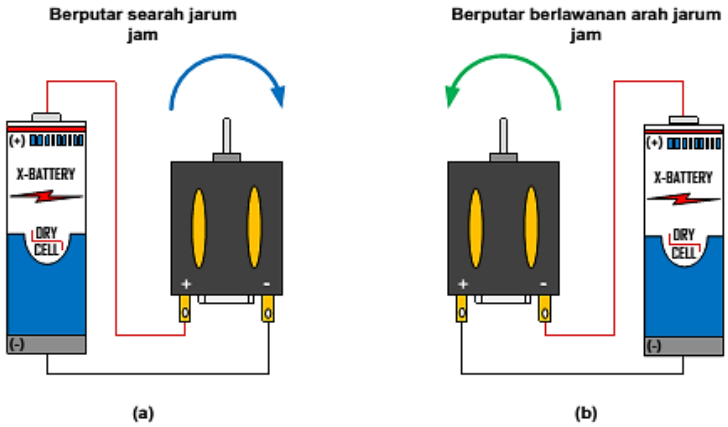
Motor DC merupakan perangkat elektromagnetis yang mengubah energi listrik menjadi energi mekanik. Energi mekanik ini digunakan dalam berbagai peralatan, misalnya memutar *impeller* pompa, *fan* atau *blower*, menggerakkan kompresor, mengangkat bahan dan lain sebagainya. Motor DC ini memerlukan suplai tegangan yang searah pada kumparan medan untuk diubah menjadi energi mekanik.



Gambar 2.9 Bentuk Fisik Motor DC

Prinsip dasar kerja dari motor DC adalah jenis motor ini menggunakan tegangan searah sebagai sumber tenaganya, dengan memberikan beda tegangan pada kedua terminal tersebut, motor akan berputar pada satu arah dan bila polaritas dari tegangan tersebut dibalik maka arah putaran motor akan terbalik pula. Polaritas dari tegangan yang diberikan pada kedua terminal ini menentukan kecepatan motor.

Dalam aplikasinya seringkali sebuah motor digunakan untuk arah yang searah dengan jarum jam maupun sebaliknya. Untuk mengubah putaran dari sebuah motor dapat dilakukan dengan mengubah arah arus yang mengalir melalui motor tersebut. Secara sederhana seperti yang ada pada Gambar 2.18, hal ini dapat dilakukan hanya dengan mengubah polaritas tegangan motor.



Gambar 2.10 Dasar Pengaturan Arah Putaran Motor DC

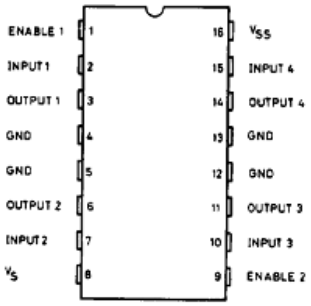
Bila kita lihat pada gambar (a) diatas, bahwa perputaran motor DC dikatakan searah jarum jam jika diberikan tegangan sumber dengan polaritas yang searah maka arah putaran motor akan mengarah ke kanan. Sedangkan pada gambar (b) diatas, bahwa perputaran motor DC dikatakan berlawanan arah jarum jam jika diberikan tegangan sumber dengan polaritas terbalik, maka arah putaran motor akan mengarah ke kiri. Kecepatan pada motor DC dapat diatur dengan berbagai cara yaitu dengan mengatur fluks medan dan tegangan sumber.

2.7 *Driver* Motor L293D

IC L293D diproduksi oleh *SGS-Thomson Microelectronics* merupakan sebuah *chip* yang di desain khusus sebagai *driver* motor DC dan dapat dikendalikan dengan rangkaian TTL maupun mikrokontroler. Motor DC yang dikontrol dengan *driver* IC L293D dapat dihubungkan ke *ground* maupun sumber tegangan positif.



Gambar 2.11 Bentuk Fisik IC L293D



(Sumber : Datasheet *STMicroelectronics* L293D)
Gambar 2.12 Konfigurasi *Pin* IC *Driver* Motor L293D

Dalam satu unit *chip* IC L293D terdiri dari 4 buah *driver* motor DC yang berdiri sendiri dengan kemampuan mengalirkan arus 1 *Ampere* pada tiap *driver*-nya. Sehingga dapat digunakan untuk membuat *driver Half-bridge* (*H-bridge*) untuk 2 buah motor DC. Selain konfigurasi *pin*, IC L293D juga memiliki beberapa fungsi dari setiap *pin*-nya yaitu :

a. *Pin Enable* (EN1 dan EN2), berfungsi untuk mengijinkan *driver* menerima perintah untuk menggerakkan motor DC.
b. *Pin Input* (1A, 2A, 3A, 4A) adalah *pin input* kendali pada motor DC.
c. *Pin Output* (1Y, 2Y, 3Y, 4Y) adalah jalur *output* masing-masing *driver* yang dihubungkan ke motor DC.
d. *Pin* VCC (Vcc1/Vss dan Vcc2/Vs) adalah jalur tegangan sumber *driver* motor DC, dimana Vcc1 adalah jalur *input* sumber tegangan rangkaian

driver kontrol dan Vcc2 adalah jalur *input* sumber tegangan untuk motor DC yang dikendalikan.

e. *Pin* GND (*Ground*) adalah jalur yang harus dihubungkan ke *ground*. Pada IC L293D memiliki 4 buah *pin ground* yang berdekatan dan dapat dihubungkan ke sebuah pendingin kecil (*heatsink*).

2.8 Pemrograman *Basic Compiler-AVR*

Perangkat lunak (*software*) yang digunakan dalam tugas akhir ini adalah program *Basic Compiler* (*Bascom-AVR*). Pemrograman menggunakan *Bascom-AVR* adalah salah satu dari sekian banyak bahasa *Basic* untuk pemrograman mikrokontroler, misalnya bahasa *Assembly*, bahasa C dan lain sebagainya. Bahasa pemrograman *Basic* sudah dikenal di seluruh dunia sebagai bahasa pemrograman yang handal, cepat, mudah dan tergolong ke dalam bahasa pemrograman tingkat tinggi. 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-peryataan
Loop
End
```

Berikut ini adalah bagian-bagian program yang terdapat pada *Bascom-AVR*, diantaranya ialah :

1. Tipe Data

   Tipe data merupakan bagian program yang paling penting karena tipe data sangat mempengaruhi setiap instruksi dalam program yang akan dilaksanakan oleh komputer. Pemilihan tipe data yang tepat akan membuat operasi data menjadi lebih efektif dan efisien.

2. Konstanta

   Konstanta merupakan suatu nilai yang tidak dapat diubah selama proses program sedang berlangsung. Konstanta nilainya selalu tetap. Konstanta didefinisikan terlebih dahulu di awal program. Konstanta dapat bernilai integer, pecahan, karakter dan string.

3. Variabel

   Variabel adalah suatu pengenal (*identifier*) yang digunakan untuk mewakili suatu nilai tertentu di dalam proses program. Berbeda dengan konstanta yang nilainya selalu tetap, nilai dari suatu variabel bisa diubah-ubah sesuai kebutuhan. Nama dari suatu variabel dapat ditentukan sendiri oleh pemrogram dengan aturan sebagai berikut :

   a. Terdiri dari gabungan huruf dan angka dengan karakter pertama harus berupa huruf.
   b. Tidak boleh mengandung spasi.
   c. Tidak boleh mengandung simbol-simbol khusus, kecuali garis bawah (*underscore*). Disini yang termasuk simbol khusus yang tidak diperbolehkan antara lain : $, ?, %, #, !, &, *, (,), -, +, = dan lain-lain.

d. Maksimum 32 karakter.

4. Deklarasi

Deklarasi diperlukan jika akan menggunakan pengenal (*identifier*) dalam program. *Identifier* dapat berupa variabel, konstanta dan fungsi.

a. Deklarasi Variabel

Bentuk umum pendeklarasian suatu variabel adalah :

Dim Nama_variabel As tipe_data;

Contoh : Dim x As Integer ‘deklarasi x bertipe integer

b. Deklarasi Konstanta

Dalam bahasa *Basic*, konstanta dideklarasikan langsung,

Contoh : S = “Hello world” ‘*Assign String*

c. Deklarasi Fungsi

Fungsi merupakan bagian yang terpisah dari program dan dapat dipanggil dimanapun di dalam program. Fungsi dalam bahasa *Basic* ini terdapat beberapa fungsi yang disediakan sebagai fungsi pustaka seperti *Print*, *Input* dan *Data*, serta untuk menggunakannya tidak perlu dideklarasikan. Fungsi yang perlu dideklarasikan terlebih dahulu adalah fungsi yang dibuat oleh programmer, bentuk umum deklarasi sebuah fungsi adalah :

TEST[(BYREF/BYVAL] var ss type)] As type

Contoh : *Declare Function Myfunction* (byval I As Integer, S As String) As Integer.

5. Operator

Operator adalah operasi yang dikenakan pada program. Terdapat beberapa operator yang terdapat dalam program *Bascom-AVR*, diantaranya operator penugasan (*Assignment Operator*), operator aritmatika, operator hubungan (perbandingan), operator logika dan operator *bitwise*.

a. Operator Penugasan

Di dalam bahasa *Basic* terdapat satu simbol operator penugasan yaitu berupa "=".

b. Operator Aritmatika

Terdapat beberapa simbol operator aritmatika yang dapat digunakan di dalam program diantaranya :

.*	: untuk perkalian
/	: untuk pembagian
%	: untuk sisa pembagian (modulus)
+	: untuk pertambahan
-	: untuk pengurangan

c. Operator Hubungan (Perbandingan)

Operator hubungan digunakan untuk membandingkan hubungan antara dua buah *operand*/ sebuah nilai atau variabel. Dibawah ini terdapat beberapa operator hubungan (perbandingan) diantaranya :

=	: *Equality*	X = Y
< >	: *Inequality*	X < > Y
<	: *Less than*	X < Y
>	: *Greater than*	X > Y
< =	: *Less than or equal to*	X < = Y
> =	: *Greater than or equal to*	X > = Y

d. Operator Logika

Operator logika digunakan untuk membandingkan suatu logika hasil dari operator-operator hubungan. Di dalam program terdapat empat macam operator logika yang digunakan yaitu :

NOT (*Logical complement*)
AND (*Conjunction*)
OR (*Disjunction*)
XOR (*Exclusive or*)

e. Operator *Bitwise*

Operator *bitwise* digunakan untuk memanipulasi *bit* dari data yang ada di memori. Di dalam bahasa *Basic* operator *bitwise* yang digunakan yaitu :

Shift A, Left, 2 : Pergeseran *bit* ke kiri
Shift A, Right, 2 : Pergeseran *bit* ke kanan
Rotate A, Left, 2 : Putar *bit* ke kiri
Rotate A, Right, 2 : Putar *bit* ke kanan

6. Komentar Program

Komentar program hanya diperlukan untuk memudahkan pembacaan dan pemahaman suatu program (untuk keperluan dokumentasi program). Dengan kata lain, komentar program hanya merupakan sebuah keterangan atau penjelasan program. Komentar program tidak akan ikut diproses dalam program (akan diabaikan).

7. Penyeleksian Kondisi

Penyeleksian kondisi digunakan untuk mengarahkan perjalanan suatu proses. Fungsi penyeleksian kondisi ini penting karena hal ini diperlukan dalam penyusunan bahasa *Basic*, terutama untuk program yang kompleks.

Terdapat beberapa struktur kondisi, diantaranya struktur kondisi *"If ..."* struktur kondisi *" If... Else"* dan struktur kondisi *"Select Case".*

a. Struktur Kondisi *"If ..."*

Struktur *If* sering digunakan untuk menyeleksi suatu kondisi tunggal. Bila proses yang diseleksi terpenuhi, maka pernyataan yang ada di dalam blok *If* akan dikerjakan. Bentuk umum dari struktur kondisi *If* adalah :

if(kondisi) then pernyataan;

Contoh : If data1 = 1 Then Call Lestoright(1)

b. Struktur Kondisi *"If... Else"*

Di dalam struktur kondisi *If ... Else* minimal harus terdapat dua pernyataan. Jika suatu kondisi yang diperiksa bernilai benar maka pernyataan pertama yang dilaksanakan dan jika salah maka pernyataan kedua yang dilaksanakan. Bentuk umum dari struktur kondisi *If ... Else* ini adalah sebagai berikut.

```
if(kondisi then pernyataan;
Pernyataan-1
else
Pernyataan-2
Contoh :    If A > 10 then
            Print "A > 10"
            Else
            Print "A not greater than 10"
            End If
```

c. Struktur Kondisi *"Select Case"*

Struktur kondisi *select case* pada dasarnya sama dengan struktur kondisi *If ... Else,* hanya saja kondisinya banyak. Struktur kondisi *switch ... case ... default* digunakan untuk penyeleksian kondisi

dengan kemungkinan yang terjadi cukup banyak. Pernyataan yang dilaksanakan terus hingga ditemukan pernyataan 'break'. Jika tidak ada kondisi yang sesuai maka yang dikerjakan adalah program dibawahnya. Bentuk umum dari struktur kondisi *"select case"* ini adalah :

```
Select Case(kondisi)
{
        case 1 : pernyataan-1;
        break;
        case 2 : pernyataan-2;
break;
.....
case n : pernyataan-n;
break;
default : pernyataan-m;
}
```

Contoh :

```
        Select Case Ultra
                Case &B01000001 : Call Maju(6)
                Case &B01001000 : Call Turnright110( )
                Case Else : Call Maju(2)
        End Select;
```

8. Perulangan

Dalam bahasa *Basic* tersedia suatu fasilitas yang digunakan untuk melakukan proses yang berulang-ulang. Struktur perulangan dalam bahasa *Basic* mempunyai bentuk yang bermacam-macam, diantaranya struktur perulangan *"While", "Do ... Loop"* dan *"For ... Next"*.