

BAB II

TINJAUAN PUSTAKA

2.1 Sensor

Sensor adalah transduser yang berfungsi untuk mengolah variasi gerak, panas, cahaya atau sinar, magnetis dan kimia menjadi tegangan arus listrik. Transduser sendiri memiliki arti mengubah, reasapan dari bahasa latin *traducere*. Bentuk perubahan yang dimaksud ialah kemampuan merubah suatu energi kedalam bentuk energi lain. Energi yang diolah bertujuan untuk menunjang daripada kinerja piranti yang menggunakan sensor itu sendiri. Sensor sering digunakan dalam proses pendeteksi untuk proses pengukuran. Sensor yang sering digunakan dalam berbagai rangkaian elektronik antara lain sensor cahaya, atau sinar, sensor suhu, sensor tekanan, sensor ultrasonik dan sensor asap.

Dari pengertian sensor tersebut wajar jika sensor menjadi alat yang banyak diminati oleh berbagai pabrikan elektronik. Bicara tentang sensor sangatlah luas, namun kita ambil salah satu contoh pabrikan yang tengah gencar menggunakan sensor pada produk mereka adalah pabrikan handphone dengan model touch screen. Sensor tekanan pada berbagai handphone sekarang ini membutuhkan adanya dukungan dari sensor tekanan. Selain pada gadget dengan teknologii canggih tersebut, sensor tekanan juga biasa diaplikasikan kepada berbagai alat elektronik lain seperti kalkulator serta remote bekerja dengan mengubah daya tekan tersebut menjadi daya atau sinyal listrik.

Dengan pengerian sensor beserta kinjera dari sensor tekanan diatas dapat kita ambil kesimpulan bahwa sensor memilikii banyak andil pada berbagai tekbologi. Pada sensor sushu sendiri terdapat empat jenis sensor yang sering digunakan, yaitu thermocouple, resistance temperature detectore, IC sensor dan termistor. Pada komponen thermocouple terdapat dua komponen transduser panas dan juga dingin. Kedua transedur tersebut berfungsi untuk membandingkan objek serta untuk mendapatkan hasil akan suhu dari objek. Termistor merupakan resistor yang tahan terhadap panas serta IC sensor suhu dengan rangkaian yang menggunakan chipsilikon guna mendeteksi tingkat suhu yang terdapat pada objek

(Sumber : Afrie Setiawan, 2011)

2.2 Mikrokontroler

Mikrokontroler merupakan sebuah sistem komputer yang seluruh atau sebagian besar elemennya dikemas dalam satu chip IC, sehingga sering disebut *single chip microcomputer*. Mikrokontroler merupakan sebuah sistem komputer yang mempunyai satu atau beberapa tugas yang sangat spesifik, berbeda dengan PC yang memiliki beragam fungsi. Perbedaan yang lainnya adalah perbandingan RAM dan ROM yang sangat berbeda antara komputer dengan mikrokontroler. Dalam mikrokontroler ROM jauh lebih besar dibanding RAM, sedangkan dalam komputer atau PC RAM jauh lebih besar dibanding ROM.

Mikrokontroler adalah suatu chip berupa IC (Integrated Circuit) yang dapat menerima sinyal input, mengolahnya dan memberikan sinyal output sesuai dengan program yang diisikan ke dalamnya. Sinyal input mikrokontroler berasal dari sensor yang merupakan informasi dari lingkungan sedangkan sinyal output ditujukan kepada aktuator yang dapat memberikan efek ke lingkungan.

Jadi secara sederhana mikrokontroler dapat diibaratkan sebagai otak dari suatu perangkat/produk yang mampu berinteraksi dengan lingkungan sekitarnya, mikrokontroler dijalankan dengan menggunakan bahasa pemrograman, salah satu contoh mikrokontroler yang saya gunakan untuk pembuatan alat pada proyek akhir saya adalah Mikrokontroler Atmega 8535

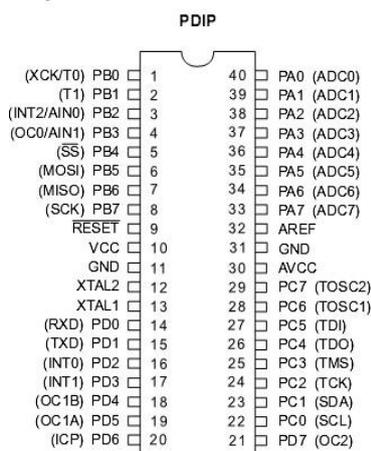
2.2.1 Mikrokontroler ATmega8535

Atmega 8535 adalah mikrokontroler CMOS 8-bit daya-rendah berbasis arsitektur RISC. Kebanyakan instruksi dikerjakan pada satu siklus clock, Atmega 8535 mempunyai throughput mendekati 1 MIPS per MHz, hal ini membuat Atmega 8535 dapat bekerja dengan kecepatan tinggi walaupun dengan penggunaan daya rendah.

2.2.2 Konfigurasi Pin ATmega16

Konfigurasi pin (pin) mikrokontroler Atmega16 terdiri dari 40 pin, dimana ATmega16 memiliki 8 pin untuk masing-masing Gerbang A (Port A), Gerbang B (Port B), Gerbang C (Port C), dan Gerbang D (Port D).

(Sumber : Afrie Setiawan, 2011)



Gambar 2.1. Konfigurasi pin ATMega16
(Sumber : Afrie Setiawan, 2011)

Penjelasan tiap pin :

Tabel 2.1. Penjelasan Pin ATMega16

No Pin	Nama	Fungsi
1	PB0(XCK/TO)	Port B.0/Counter/Clock eksternal untuk USART (xck)
2	PB1 (T1)	Port B.1/ Counter 1
3	PB2 (INT2/AIN0)	Port B.2/ Input (+) Analog Komparator (AIN0) dan interupsi eksternal 2 (INT2)
4	PB3 (OC0/AIN1)	Port B.3/ Input (-) Analog Komparator (AIN1) dan output PWM 0
5	PB4 (SS)	Port B.4/ SPI Slave Select Input (SS)
6	PB5 (MOSI)	Port B.5/ SPI Bus Master Out Slave In
7	PB6 (MISO)	Port B.6/ SPI bus Master In Slave Out
8	PB7 (SCK)	Port B.7/ Sinyal Clock Serial SPI
9	RESET	Me-Reset Mikrokontroler
10	VCC	Catu Daya (+)
11	GND	Sinyal Ground Catu Daya
12-13	XTAL2-XTAL1	Sinyal Input Clock eksternal (kristal)
14	PD0 (RXD)	Port D.0/ Penerima data serial

15	PD1 (TXD)	Port D.1/ Pengirim data serial
16	PD2 (INT0)	Port D.2/ Interupsi eksternal 0
17	PD3 (INT1)	Port D.3/ Interupsi eksternal 1
18	PD4 (OC1)	Port D.4/ Pembanding <i>Timer-Counter</i> 1
19	PD5 (OC1A)	Port D.5/ Output PWM 1A
20	PD6 (ICP1)	Port D.6/ <i>Timer-Counter</i> 1 <i>Input</i>
21	PD7 (OC2)	Port D.7/ Output PWM 2A
22	PC0 (SCL)	Port C.0/ Serial <i>bus clock line</i>
23	PC1 (SDA)	Port C.1/ Serial <i>bus data input-output</i>
24-27	PC2 – PC5	Port C.2- Port C.5
28	PC6 (TOSC1)	Port C.6/ <i>Timer</i> Osilator 1
29	PC7 (TOSC2)	Port C.7/ <i>Timer</i> Osilator 2
30	AVCC	Tegangan ADC
31	GND	Sinyal <i>Ground</i> ADC
32	AREF	Tegangan Referensi ADC
33-40	PA0 (ADC0) – PA7 (ADC7)	Port A.0- Port A.7 dan <i>input</i> untuk ADC (8 <i>channel</i> : ADC0-ADC7)

(Afrie Setiawan, 2010 : 5)

2.3 Bahasa Pemrograman

Software adalah suatu komponen di dalam suatu sistem data yang berupa program-program atau atau instruksi untuk mengontrol suatu sistem. Pada umumnya istilah *software* menyatakan cara-cara yang menghasilkan hubungan yang lebih efisien antara manusia dan komputer. (Rusmawan, 2004:65)

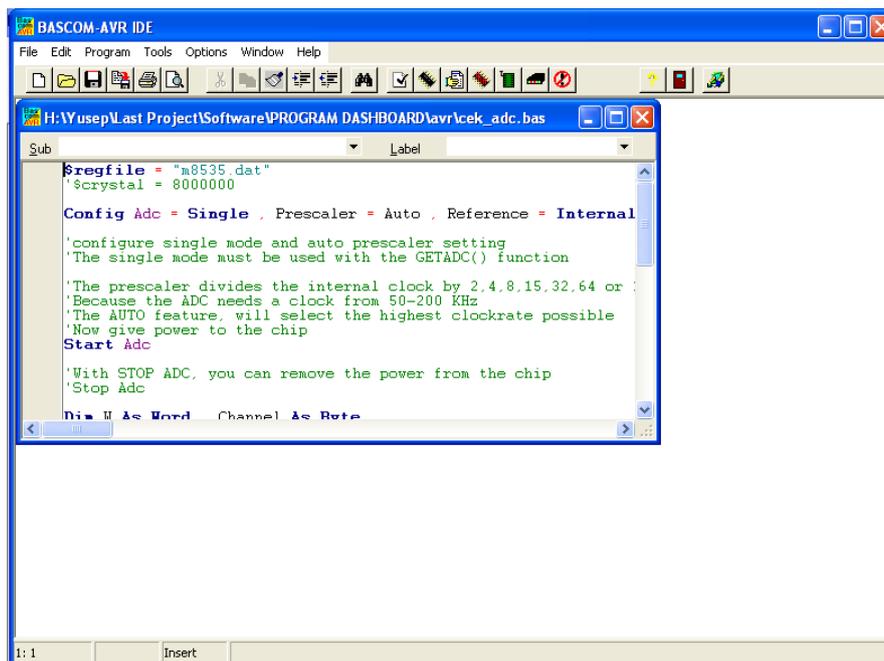
Dapat kita simpulkan bahwasanya *software* adalah rangkaian intruksi dari suatu program yang diperlukan untuk menghasilkan informasi yang ditulis dengan bahasa yang dapat dimengerti oleh komputer dan manusia. Dalam melakukan pemrograman banyak *software* yang dapat digunakan seperti BASCOM-8051, AVR Studio-4, BASCOM-AVR dan Pinneacle. Dalam perancangan sistem keamanan ruang multi sensor ini menggunakan bahasa pemrograman Basic dengan menggunakan *software* BASCOM-AVR

2.3.1. Software (BASCOS AVR)

BASCOS adalah program *Basic Compiler* berbasis windows yang digunakan untuk berbagai jenis mikrokontroller seperti MCS-51 dan AVR. BASCOS merupakan pemrograman dengan bahasa tingkat tinggi. Penguasaan BASCOS ini sangat didukung oleh pemahaman *hardware* mikrokontroller yang baik, karena setiap langkah dari program ini pasti berhubungan dengan *hardware* itu sendiri. Hal itulah yang membedakan BASCOS dengan bahasa pemrograman lainnya. BASCOS merupakan bahasa pemrograman basic yang dikembangkan dan dikeluarkan oleh MCS Electronic. Bahasa pemrograman ini memiliki beberapa keunggulan diantaranya :

- Menggunakan bahasa pemrograman basic sebagai control program
- Memiliki jendela simulasi berupa LCD (*Liquid Crystal Display*), simulasi port mikro dan sebagainya. (Didin Wahyudin;hal 43;2007)

Interface dari BASCOS AVR dapat dilihat pada gambar berikut :



Gambar 2.2 Interface BASCOS-AVR

(sumber:<http://avrhelp.mcselec.com/>)

Keterangan lengkap ikon-ikon dari program BASCOS-8051 dapat dilihat ditabel 2.2

Tabel 2.2 Keterangan ikon-ikon pada program BASCOM

<i>Icon</i>	<i>Nama</i>	<i>Fungsi</i>	<i>Shortcut</i>
	<i>File New</i>	Membuat file baru	Ctrl+N
	<i>Open File</i>	Untuk Membuka File	Ctrl+N
	<i>File Close</i>	Untuk Menutup proram yang dibuka	Ctrl+O
	<i>File Save</i>	Untuk menyimpan file	Ctrl+S
	<i>Save as</i>	Menyimpan dengan nama yang lain	-
	<i>Print preview</i>	Untuk melihat tampilan sebelum dicetak	-
	<i>Print</i>	Untuk mencetak dokumen	Ctrl+P
	<i>Exit</i>	Untuk Keluar dari program	-
	<i>Program compile</i>	Untuk mengkompile program yang dibuat, Outputnya bisa berupa *.hex, *.bin dll	F7
	<i>Syntax check</i>	Untuk memeriksa kesalahan bahasa	Ctrl+F7
	<i>Show result</i>	Untuk menampilkan hasil kompilasi program	Ctrl+W

(sumber:<http://avrhelp.mcselec.com/>)

Untuk menu *show result* informasi yang akan ditampilkan berupa :

Tabel 2.3 Info tampilan menu pada program BASCOM

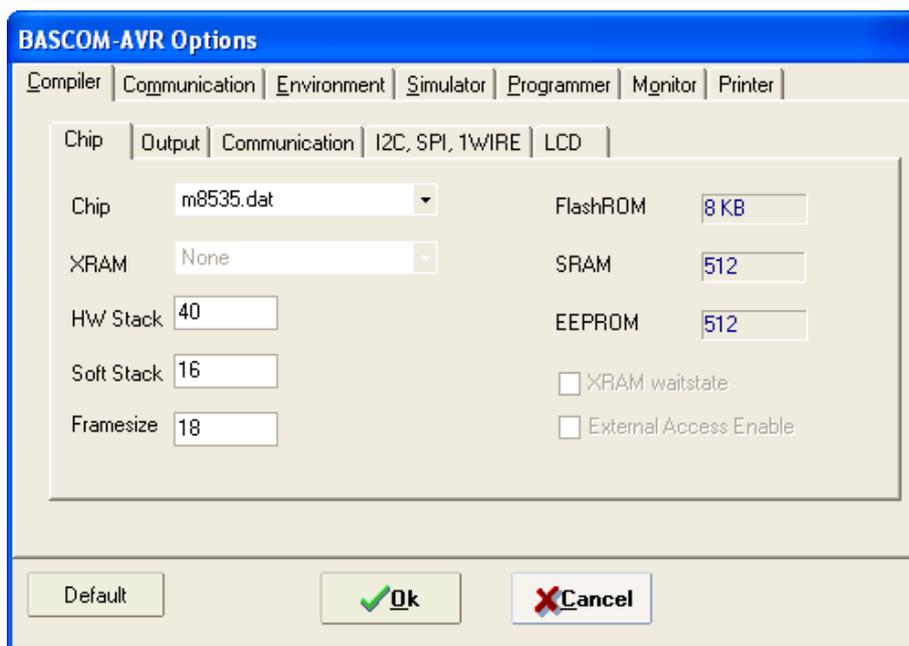
<i>Info</i>	<i>Keterangan</i>
Compiler	Versi dari compiler yang digunakan
Processor	Menampilkan target prosesor yang dipilih
Date and time	Tanggal dan waktu kompilasi
Baud rate dan xtal	Baudrate yang dipilih dan kristal yang digunakan uP.

Error	Error nilai Baud yang di set dengan nilai baud sebenarnya
Flash Used	Persentase flash ROM yang terisi program
Stack Start	Lokasi awal stack pointer memori
RAM Start	Lokasi awal eksternal RAM.
LCD Mode	Mode LCD yang digunakan, 4 bit atau 8 bit

(sumber:<http://avrhelp.mcselec.com/>)

2.3.2 Compiler

BASCOM-AVR menyediakan pilihan untuk memodifikasi pilihan-pilihan pada kompilasi. Dengan memilih menu **Compiler** maka jendela berikut akan ditampilkan:



Gambar 2.3 Jendela *Option*

(sumber:<http://avrhelp.mcselec.com/>)

Keterangan dari pilihan tersebut adalah sebagai berikut:

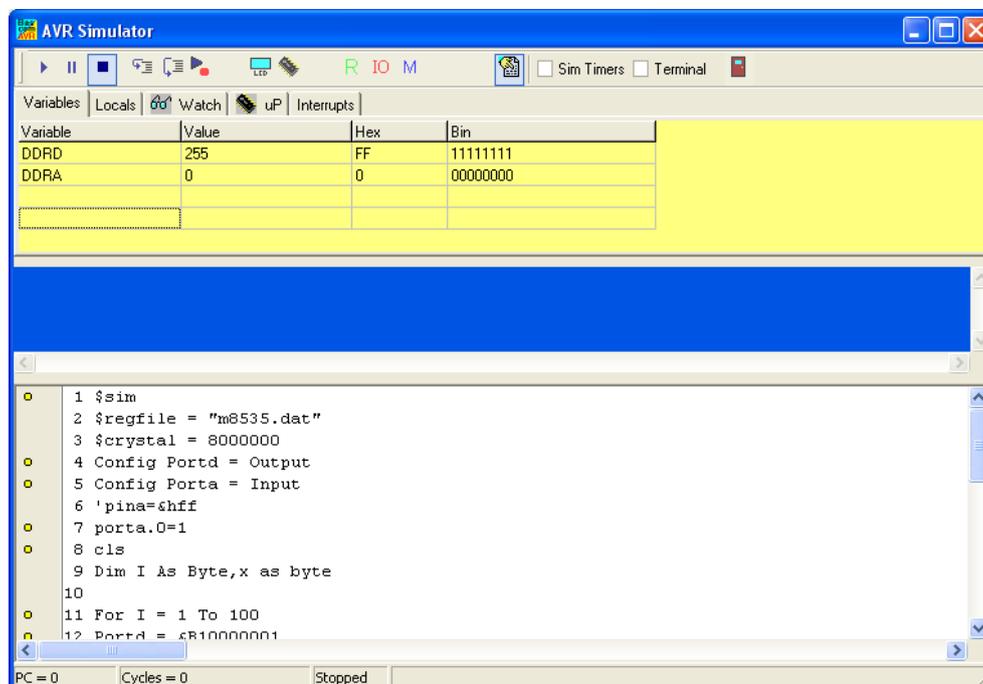
Tabel 2.4 Keterangan menu pilihan

TAB Menu	OPTION	Keterangan
Chip	Chip	Mikrokontroler yang digunakan, sebagai contoh m8535.dat untuk ATMEGA8535
	XRAM	Jika menggunakan eksternal RAM nilai ini bisa ditampilkan
	HW Stack	Stack memory hardware, setiap Gosub membutuhkan 2 byte. Jika menggunakan interupsi, naikan nilainya
	Soft stack	Stack software, nilai defaultnya 8
	FlashROM	Nilai flashROM Chip yang dipilih
	SRAM	Nilai RAM internal Chip yang dipilih
	EEPROM	Nilai EEPROM chip yang terpilih
	Output	
Communication	Baudrate 0	Nilai Baudrate yang digunakan dalam komunikasi serial
	Frekuensi	Nilai osilator yang digunakan
	Error	Error antara baudrate yang dipilih dengan nilai sebenarnya, hal ini tergantung pada osilator yang dipilih
I2C, SPI, 1 wire	SDA	Pin yang berfungsi untuk data serial dalam komunikasi I2C

	SCL	Pin yang berfungsi untuk data clock dalam komunikasi I2C
	1Wire	Pin yang digunakan untuk komunikasi 1 wire
	SPI	Pin yang digunakan untuk komunikasi serial sinkron
LCD		Pemilihan Port yang digunakan untuk tampilan LCD, jenis LCD

(sumber:<http://avrhelp.mcselec.com/>)

BASOM-AVR menyediakan pilihan yang dapat mensimulasikan program. Agar dapat menjalankan simulator ini, file DBG dan OBJ harus dipilih pada menu Options Compiler Output. Tampilan program simulasi adalah sebagai berikut:

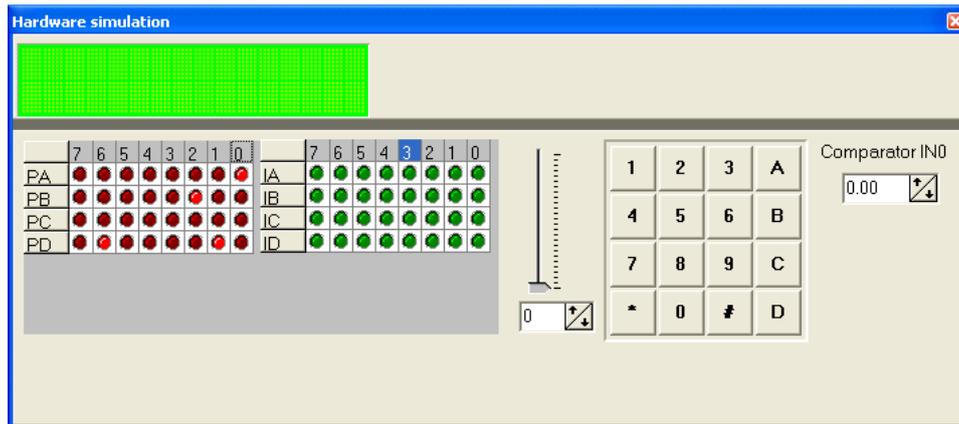


Gambar 2.4 Interface Simulator BASCOM-AVR

(sumber:<http://avrhelp.mcselec.com/>)

Tekan tombol  untuk memulai simulasi. Dan untuk memberhentikan simulasi atau menahan proses simulasi gunakan tombol disebelahnya. Layar biru ditengah merupakan simulasi layar monitor ketika menggunakan komunikasi serial.

Selain itu untuk dapat melihat perubahan data pada setiap port atau ketika kita ingin memberikan input pada pin-pin tertentu dari mikrokontroller, maka gunakan tombol  untuk menampilkan jendela sebagai berikut:



Gambar 2.5 Jendela simulasi Hardware

(sumber:<http://avrhel.mcselec.com/>)

2.4 Dasar Pemrograman Basic Compiler (BASCOM)

2.4.1 Tipe data

Setiap variabel dalam BASCOM memiliki tipe data yang menunjukkan daya tampung variabel tersebut, hal ini berhubungan dengan penggunaan memori dari mikrokontroler. Berikut ini adalah tipe data pada BASCOM berikut keterangannya.

Tabel 2.5 Tipe Data BASCOM

Tipe Data	Ukuran (byte)	Range
Bit	1/8	-
Byte	1	0-255
Integer	2	-32,768 - +32,767

Word	2	0 - 65535
Long	4	-2147483648 - +2147483647
Single	4	-
String	S/d 254 byte	-

(sumber:<http://avrhelp.mcselec.com/>)

2.4.2 Variabel

Variabel dalam sebuah pemrograman berfungsi sebagai tempat penyimpanan data atau penampung data sementara, misalnya menampung hasil perhitungan, menampung data hasil pembacaan register dll. Variabel merupakan pointer yang menunjuk pada alamat memori fisik di mikrokontroler.

Dalam BASCOM ada beberapa aturan dalam penamaan sebuah variabel:

1. Nama variabel maksimum terdiri atas 32 karakter
2. Karakter bias berupa angka atau huruf
3. Nama variabel harus dimulai dengan huruf
4. Variabel tidak boleh menggunakan kata-kata yang digunakan oleh BASCOM sebagai perintah, pernyataan, internal register dan nama operator (AND,OR, DIM dll)

Sebelum variabel digunakan maka variabel tersebut harus dideklarasikan terlebih dahulu, dalam BASCOM ada beberapa cara untuk mendeklarasikan sebuah variabel. Yang pertama dengan menggunakan pernyataan "**DIM**" diikuti nama dan tipe datanya, contoh pendeklarasian menggunakan DIM sebagai berikut:

Dim nama **as** byte

Dim tombol1 **as** integer

Dim tombol2 **as** word

Dim tombol3 **as** word

Dim tombol4 **as** word

Dim Kas **as** string*10

2.4.3 Alias

Dengan menggunakan **ALIAS** sebuah variabel yang sama dapat diberikan nama yang lain, tujuannya untuk mempermudah proses pemrograman. Biasanya **ALIAS** digunakan untuk mengganti nama variabel yang telah baku seperti port mikrokontroler.

```
LEDBAR alias portA
```

```
Tombol1 alias portA.1
```

```
Tombol2 alias PortA.2
```

Dengan deklarasi seperti di atas maka perubahan pada tombol1 akan merubah kondisi dari P0.1. Selain mengganti nama port **ALIAS** juga dapat digunakan untuk mengakses bit tertentu dari sebuah variabel yang telah dideklarasikan

```
Dim LedBar as byte
```

```
Led1 as LedBar.0
```

```
Led2 as LedBar.1
```

```
Led3 as LedBar.2
```

2.4.4 Konstanta

Dalam BASCOM selain variabel dikenal juga konstanta, konstanta ini juga merupakan variabel. Perbedaannya dengan variabel biasa adalah nilai yang dikandungnya tetap. Dengan konstanta, kode program yang kita buat akan lebih mudah dibaca dan dapat mencegah kesalahan penulisan pada program kita. Misal kita akan lebih mudah menulis *phi* daripada menulis 3,14159867. Sama seperti variabel agar konstanta ini bias dikenali oleh program maka harus dideklarasikan terlebih dahulu. Berikut ini adalah cara pendeklarasian sebuah konstanta

```
Dim A As Const 5
```

```
Dim B1 As Const &B1001
```

Cara lain yang paling Mudah:

```
Const Cbyte = &HF
```

```
Const Cint = -1000
```

```
Const Csingle = 1.1
```

```
Const Cstring = "test"
```

2.4.5 Array

Dengan Array kita bias menggunakan sekumpulan variabel dengan nama dan tipe yang sama, untuk mengakses variabel tertentu dalam array tersebut kita harus menggunakan indeks. Indeks ini harus berupa angka dengan tipe data byte, ineteger atau word, hal ini berarti nilai maksimum sebuah indeks adalah sebesar 65535

Proses pendeklarasian sebuah array hampir sama dengan varaiabel namun perbedaanya kita juga mengikutkan jumlah elemennya. Berikut ini contoh pemakaian array:

```
Dim kelas(10) as byte
```

```
Dim c as Integer
```

```
For C = 1 To 10
```

```
    a(c) = c
```

```
    p1 = a(c)
```

```
Next
```

(sumber:<http://avrhelp.mcselec.com/>)

2.5 Operasi-operasi dalam Basic Compiler (BASCOS)

Pada bagian ini kita akan membahas tentang bagaimana cara menggabungkan, memodifikasi, membandingkan atau mendapatkan informasi tentang sebuah pernyataan dengan menggunakan operator-operato yang tersedia di BASCOM. Bagian ini juga akan menjelaskan bagaimana sebuah pernyataan tebetuk dan dihasilkan dari operator-operator berikut:

- **Operator Aritmatika**

Digunakan dalam perhitungan, yang termasuk opertor aritmatika ialah

+ (tambah), - (Kurang), /(Bagi) dan * (kali)

- **Operator Relasi**

Digunakan untuk membandingkan nilai sebuah angka, hasilnya dapat digunakan untuk membuat keputusan sesuai dengan program yang kita buat. Yang termasuk operator relasi adalah:

Tabel 2.6 Tabel operator Relasi

Operator	Relasi	Pernyataan
=	Sama Dengan	$X = Y$
\neq	Tidak sama dengan	$X \neq Y$
<	Lebih kecil dari	$X < Y$
>	Lebih Besar dari	$X > Y$
\leq	Lebih kecil atau sama dengan	$X \leq Y$
\geq	Lebi Besar atau sama dengan	$X \geq Y$

- **Operator Logika**

Digunakan untuk menguji sebuah kondisi atau untuk memanipulasi bit dan operasi boolean. Dalam BASCOM ada empat buah operator logika Yaitu **AND, OR, NOT** dan **XOR**.

(sumber:<http://avrhelp.mcselec.com/>)

2.6 Kontrol Program

Berikut adalah beberapa kontrol program

2.6.1 **IF...THEN**

Dengan pernyataan ini kita dapat mengetes sebuah kondisi tertentu dan kemudian menentukan tindakan yang sesuai dengan kondisi yang diinginkan. Sintak penulisannya adalah sebagai berikut:

IF <syarat Kondisi> **THEN** <Pernyataaan>

Sintak diatas digunakan jika hanya ada satu kondisi yang diuji dan hanya melakukan satu tindakan. Jika melakukan lebih dari satu tindakan maka sintaknya adalah:

IF <syarat Kondisi> **THEN**

```

<Pernyataan ke-1>
<Pernyataan ke-2>
.
.
<Pernyataan ke-n>
END IF

```

Jika ada lebih dari satu kondisi atau lebih yang akan diuji maka sintaknya menjadi :

```

IF <syarat kondisi 1> THEN
    <blok pernyataan ke-1>
ELSEIF <syarat kondisi 2> THEN
    <blok pernyataan ke-2>
END IF

```

2.6.2 SELECT...CASE

Perintah ini akan mengeksekusi beberapa blok pernyataan tergantung dari nilai variabelnya. Perintah ini mirip dengan perintah IF..Then namun perintah ini memiliki kelebihan yaitu kemudahan pada penulisannya. Sintaknya sebagai berikut:

```

SELECT CASE variabel

CASE test1 : statements

CASE test2 : statements

CASE ELSE : statements

END SELECT

```

2.6.3 DO...LOOP

Perintah **Do..loop** digunakan untuk mengulangi sebuah blok pernyataan terus menerus. Untuk membatasi perulangannya dapat ditambahkan sebuah syarat kondisi agar perulangan berhenti dan perintahnya menjadi **Do..loop until**. Sintaknya sebagai berikut:

DO

<blok pernyataan>

LOOP

Yang menggunakan perintah **Do..Loop Until**

Do

<blok pernyataan>

loop Until <syarat kondisi>

2.6.4 FOR...NEXT

Perintah ini digunakan untuk mengeksekusi sebuah blok pernyataan secara berulang. Perintah ini hampir sama dengan perintah **Do..Loop** namun pada perintah **For..Next** ini nilai awal dan akhir perulangan serta tingkat kenaikan atau turunnya bisa ditentukan. Penggunaannya sebagai berikut:

FOR var = start **TO/DOWNTO** end [**STEP value**]

<Blok pernyataan>

Next

2.6.5 EXIT

Perintah ini digunakan untuk keluar secara langsung dari blok program **For..next**, **Do..Loop**, **Sub..EndSub**, **While..Wend**. Sintak penulisannya adalah sebagai berikut:

EXIT [**Do**] [**For**] [**while**] [**Sub**]

2.6.6 GOSUB

Dengan **Gosub** program akan melompat ke sebuah label dan akan menjalankan program yang ada dalam sub rutin tersebut sampai menemui perintah **Return** . Perintah **Return** akan mengembalikan program ke titik setelah perintah **Gosub**. Sintaknya sebagai berikut:

```
GOSUB Label
    <blok pernyataan>
...
END
```

2.6.7 GOTO

Perintah **GOTO** digunakan untuk melakukan percabangan, perbedaannya dengan **GOSUB** ialah perintah **GOTO** tidak memerlukan perintah **Return** sehingga programnya tidak akan kembali lagi ke titik dimana perintah **GOTO** itu berada. Berikut ini sintak perintah **GOTO** :

```
GOTO label
Label :
```

(Didin Wahyudin :hal.49-58;2007)

2.7 Flowchart

Flowchart (Bagan alir) merupakan suatu bagan yang menggambarkan arus logika dari data yang akan diproses dari awal sampai akhir. Tujuan utama dari penggunaan *Flowchart* adalah untuk menggambarkan suatu tahapan penyelesaian masalah secara sederhana, terurut, rapi dan jelas dengan menggunakan simbol-simbol yang standar. Tahap masalah yang disajikan harus jelas, sederhana, efektif dan tepat. Dalam penulisan *flowchart* dikenal dua metode, yaitu sistem *flowchart* dan program *flowchart*. (Tosin, 1994:16-25)

2.7.1 Sistem *Flowchart*

Sistem *flowchart* menggambarkan suatu sistem peralatan komputer yang digunakan dalam proses pengolahan data serta hubungan antar peralatan tersebut. Sistem *flowchart* ini tidak digunakan untuk menggambarkan urutan langkah untuk

memecahkan masalah, tetapi hanya untuk menggambarkan prosedur dalam sistem yang dibentuk.

Dalam menggambar *flowchart* biasanya digunakan simbol-simbol yang standar, tetapi pemrograman juga dapat membuat simbol-simbol yang telah tersedia dirasa masih kurang. Dalam kasus ini pemrograman harus melengkapi gambar *flowchart* tersebut dengan kamus simbol untuk menjelaskan arti dari tiap-tiap simbol yang digunakan agar pemrogram lain dapat mengetahui maksud dari simbol-simbol tersebut.

(Tosin, 1994:16-25)

2.7.2 Program *Flowchart*

Program *Flowchart* merupakan diagram alir yang menggambarkan urutan logika dari suatu prosedur pemecahan masalah. Dalam menggambarkan program *flowchart*, telah tersedia simbol-simbol standar, tetapi seperti pada sistem *flowchart*, pemrograman dapat menambah khasanah simbol-simbol tersebut, tetapi pemrograman harus melengkapi penggambaran program *flowchart* dengan kamus simbol.

(Tosin, 1994:16-25)

Perbedaan program *flowchart* dan sistem *flowchart* ialah pada sistem *flowchart* digunakan hanya untuk menggambarkan urutan langkah untuk memecahkan masalah, tetapi hanya untuk menggambarkan prosedur dalam sistem yang dibentuk. Sedangkan sistem *flowchart* merupakan diagram alir yang menggambarkan urutan logika.