BAB II
TINJAUAN PUSTAKA

2.1 Solder Listrik

Solder listrik digunakan untuk menempelkan timah pada papan PCB dan rangkaian elektronik lainnya. (Santoso, Martinus & Sugiyanto, 2013)

Hasil soldering yang baik merupakan salah satu aspek terpenting dalam realisasi suatu rangkaian elektronika. Soldering digunakan untuk menghubungkan antara kaki-kaki komponen – komponen elektronika dengan suatu sirkuit pada PCB (Printed Circuit Board). Sehingga dapat dikatakan bahwa soldering adalah proses penyambungan antara komponen elektronika dengan cirkuit. Baik – buruknya koneksi antar komponen dalam cirkuit (sistem) sangat dipengaruhi dari baik-buruknya soldering yang dilakukan. (Rugianto, 2013)

2.2 Suhu

Suhu adalah besaran yang menyatakan derajat panas dingin suatu benda dan alat yang digunakan untuk mengukur suhu adalah termometer. Dalam kehidupan sehari-hari masyarakat untuk mengukur suhu cenderung menggunakan indera peraba. Sensor suhu pada rangkaian elektronik, merupakan suatu alat yang berfungsi mengubah besaran suhu menjadi besaran elektrik. Ada beberapa macam sensor yang dapat digunakan sebagai sensor suhu pada rangkaian elektronik. (Syafitri : 2015)

2.3 Sensor Suhu LM35

Sensor suhu LM35 adalah komponen elektronika yang memiliki fungsi untuk mengubah besaran suhu menjadi besaran listrik dalam bentuk tegangan. Sensor Suhu LM35 yang dipakai dalam penelitian ini berupa komponen elektronika elektronika yang diproduksi oleh *National Semiconductor*. LM35 memiliki keakuratan tinggi dan kemudahan perancangan jika dibandingkan dengan sensor suhu yang lain, LM35 juga mempunyai keluaran impedansi yang rendah dan linieritas yang tinggi sehingga dapat dengan mudah dihubungkan dengan rangkaian kendali khusus serta tidak memerlukan

penyetelan lanjutan. Meskipun tegangan sensor ini dapat mencapai 30 volt akan tetapi yang diberikan ke sensor adalah sebesar 5 volt, sehingga dapat digunakan dengan catu daya tunggal dengan ketentuan bahwa LM35 hanya membutuhkan arus sebesar 60 µA hal ini berarti LM35 mempunyai kemampuan menghasilkan panas (*self-heating*) dari sensor yang dapat menyebabkan kesalahan pembacaan yang rendah yaitu kurang dari 0,5 ºC pada suhu 25 ºC.

Gambar 2.1 Tampak Bawah Sensor Suhu LM35

Gambar diatas menunjukan bentuk dari LM35 tampak bawah. 3 pin LM35 menujukan fungsi masing-masing pin diantaranya, pin 1 berfungsi sebagai sumber tegangan kerja dari LM35, pin 2 atau tengah digunakan sebagai tegangan keluaran atau Vout dengan jangkauan kerja dari 0 Volt sampai dengan 1,5 Volt dengan tegangan operasi sensor LM35 yang dapat digunakan antar 4 Volt sampai 30 Volt. Keluaran sensor ini akan naik sebesar 10 mV setiap derajat *celcius* sehingga diperoleh persamaan sebagai berikut.

VLM35 = Suhu * 10 mV

Secara prinsip sensor akan melakukan penginderaan pada saat perubahan suhu setiap suhu 1 ºC akan menunjukan tegangan sebesar 10 mV. Pada penempatannya LM35 dapat ditempelkan dengan perekat atau dapat pula disemen pada permukaan akan tetapi suhunya akan sedikit berkurang sekitar 0,01 ºC karena terserap pada suhu permukaan tersebut. Dengan cara seperti ini diharapkan selisih antara suhu udara dan suhu permukaan dapat dideteksi oleh sensor LM35 sama dengan suhu disekitarnya, jika suhu udara disekitarnya jauh lebih tinggi atau jauh lebih rendah dari suhu permukaan, maka LM35 berada pada suhu permukaan dan suhu udara disekitarnya.

Jarak yang jauh diperlukan penghubung yang tidak terpengaruh oleh interferensi dari luar, dengan demikian digunakan kabel selubung yang

ditanahkan sehingga dapat bertindak sebagai suatu antena penerima dan simpangan didalamnya, juga dapat bertindak sebagai perata arus yang mengkoreksi pada kasus yang sedemikian, dengan mengunakan metode *bypass* kapasitor dari Vin untuk ditanahkan. (Romain, 2014).

2.3.1 Karakteristik Sensor Suhu LM35

Berikut ini adalah karakteristik dari sensor suhu LM35 diantaranya:

1. Memiliki sensitivitas suhu, dengan faktor skala linier antara tegangan dan suhu 10 mVolt/ºC, sehingga dapat dikalibrasi langsung dalam *celcius*.
2. Memiliki ketepatan atau akurasi kalibrasi yaitu 0,5ºC pada suhu 25 ºC.
3. Memiliki jangkauan maksimal operasi suhu -55 ºC sampai +150 ºC.
4. Bekerja pada tegangan 4 sampai 30 volt.
5. Memiliki arus rendah yaitu kurang dari 60 µA.
6. Memiliki pemanasan sendiri yang rendah (*low-heating*) yaitu kurang dari 0,1 ºC pada udara diam.
7. Memiliki impedansi keluaran yang rendah yaitu 0,1 W untuk beban 1 mA.
8. Memiliki ketidaklinieran hanya sekitar ± ¼ ºC.

Kelebihan dari sensor suhu LM 35 antara lain:

1. Rentang suhu yang jauh, antara -55 sampai +150 ºC.
2. *Low self-heating*, sebesar 0.08 ºC.
3. Beroperasi pada tegangan 4 sampai 30 V
4. Rangkaian tidak rumit
5. Tidak memerlukan pengkondisian sinyal

2.4 Relay

Menurut Owen Bishop, (2004 : 55). Relay adalah sebuah saklar yang di kendalikan oleh arus. Relay memiliki sebuah kumparan tegangan rendah yang dililitkan pada sebuah inti dan arus nominal yang harus dipenuhi output rangkaian pendriver atau pengemudinya.

2.5 Pengenalan Mirokontroler

Mikrokontroler adalah *chip* yang berfungsi sebagai pengontrol atau

pengendali rangkaian elektronik dan umumnya dapat menyimpan program didalamnya (Fujianto, 2012 : 4).

Mikrokontroler tidak dapat bekerja bila tanpa program. Program tersebut memberikan instruksi kepada mikrokontroler apa yang harus dikerjakan. Mikrokontroler yang sudah bekerja dengan satu program, tidak dapat bekerja lagi jika program diganti. Dengan mikrokontroler ini memudahkan desainer untuk merancang suatu fungsi tertentu, karena kerja mikrokontroler ini dapat diprogram sesuai dengan kemauan. Dan yang lebih mudah lagi mikrokontroler ini merupakan suatu *device* yang merupakan penggabungan beberapa jenis *device* yaitu (RAM), *internal electrical erasable programmable read only memory* (EEPROM) sebagai program memori dan I/O *port*, sehingga tidak memerlukan I/O untuk penyimpanan data, karena semua media tersebut telah ada di dalam *chip* mikorkontroler tersebut. Hanya bila diperlukan fasilitas tersebut dapat ditambah diluar *chip*.

ATMEL misalnya seperti AT89C51, AT89C52 dan AT89X051. Keluarga mikrokontroler dibagi menjadi:

1. Keluarga MCS-51
2. Keluarga MCS68HC05
3. Keluarga MC68HC11
4. Keluarga AVR
5. Keluarga PIC 8

Pada pembuatan alat kali ini penulis menggunakan Mikrokontroler Jenis AVR yaitu AVR ATMega 8535.

2.5.1 Mikrokontroler AVR ATMega 8535

AVR ATMega merupakan IC CMOS 8-bit yang memilki daya rendah dalam pengoperasiannya. ATMega 8535 dapat mengeksekusi satu instruksi dalam sebuah siklus *clock*, dan dapat mencapai 1 MIPS per MHz sehingga dapat mengoptimalkan penggunaan daya rendah dengan kecepatan tinggi dan juga sudah terdapat ADC didalamnya.

Mikrokontroler dapat disebut sebagai *"one chip solution"* karena terdiri:

1. CPU (*Central Processing Unit*)
2. RAM (*Random Access Memory*)
3. EPROM/PROM/ROM (*Erasable programmable Read Only Memory*)
4. I/O (*Input/Output*) – serial dan paralel
5. *Timer Interupt Controller*

Mikrokontroler AVR (*Alf and Vegard's Risc processor*) memiliki arsitektur 8 bit, dimana semua instruksi dikemas dalam kode 16-bit (26-bit word) dan sebagian besar instruksi dieksekusi dalam 1 (satu) siklus clock atau dikenal dengan teknologi RIS (*Reduced Instruction Set Computing*), berbeda dengan instruksi MCS52 yang membutuhkan 22 siklus *clock* atau dikenal dengan teknologi CISC (*Complex Instruction Set Computing*).

Secara umum, AVR dapat dikelompokkan ke dalam 4 kelas, yaitu keluarga AT90Sxx, keluarga ATMega dan AT86RFxx. Pada dasarnya yang membedakan masing-masing adalah kelas memori, *peripheral* dan fungsinya.

2.5.2 Sistem Mikrokontroller ATmega8535

ATmega8535 merupakan IC CMOS 8-bit yang memiliki daya rendah dalam pengoperasiannya dan berbasis pada arsitektur RISC AVR. ATmega8535 dapat mengeksekusi satu instruksi dalam sebuah siklus *clock* dan dapat mencapai 1 MIPS per MHZ, sehingga para perancang dapat mengoptimalkan penggunaan daya rendah dengan kecepatan yang tinggi. Fitur-fitur yang terdapat pada ATmega8535:

a 8 *Kbyte in-system programmable flash* dengan kemampuan membaca-ketika menulis

b 512 *byte* EEPROM

EEPROM yaitu *electrically erasable programmable ROM* jenis eprom yang dapat diprogram dan dihapus dengan tegangan listrik, volume *chip* memori eeprom terdapat program bios dengan spesifikasi khusus eeprom menggunakan teknologi MNOS *metal nitride* MOS dengan kecepatan akses 1 mikro detik per instruksinya.

c 512 *byte* SRAM

d 32 *general purpose I/O*

e 32 *general purpose register*

f 3 buah *timer/counter* dengan *mode compare*

g *Intterupt external dan internal*

h USART yang dapat diprogram

i Antar muka serial *Two – Wire* dengan orientasi *byte*

j 8-channel ADC 10 bit

k *Wacthdog timer yang dapat deprogram dengan osilator internal*

l Sebuah serial non SPI

m 6 buah mode *power saving* yang dapat dipilih dengan *software*

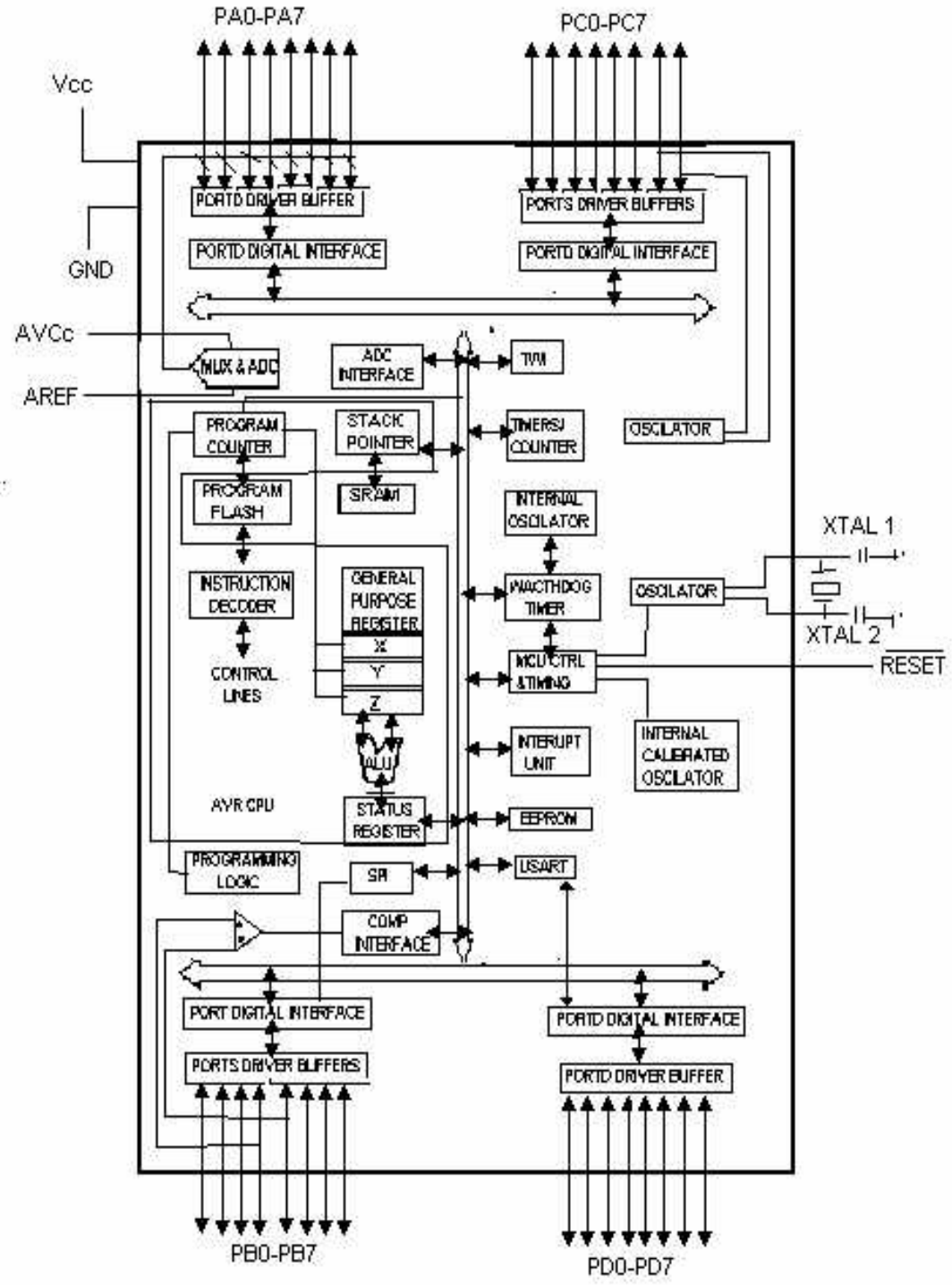
2.5.3 Diagram Blok ATmega8535

Pada diagram blok ATmega8535 digambarkan 3 *general purpose working register* yang dihubungkan secara langsung dengan *Arithmetic Logical Unit (ALU).* Sehingga dimungkinkan dua register yang berbeda dapat di *access* dalam satu siklus *clock.*

Dari gambar 2.4 dapat dilihat bahwa ATMega8535 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 3 *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. *Memory flash* sebesar 8 kb dengan kemampuan *Read While Write*
8. Unit interupsi internal dan eksternal
9. Port antarmuka SPI
10. EEPROM sebesar 512 *byte* yang dapat diprogram saat operasi.
11. Antarmuka komparator analog
12. *Port* USART untuk komunikasi serial

Berikut ini gambar diagram Blok ATMega8535 :



Gambar 2.2 Diagram Blok ATMega8535

2.5.4 Fitur ATMega 8535

Fitur-fitur dari ATMega 8535 adalah sebagai berikut :

1. Sistem mikroprosesor 8 bit berbasis RISC dengan kecepatan maksimal 26 MHz.
2. Kapabilitas memori *flash* 8 KB.
3. SRAM sebesar 522 *byte*.
4. EEPROM (*Electrically* EPROM) sebesar 522 *byte*.
5. ADC *internal* 20 bit sebanyak 8 *channel*.
6. Portal komunitas serial (USART) dengan kecepatan maksimal 2,5 Mbps.
7. 6 buah mode *sleep/power saving* yang dapat dipilih dengan *software*.

2.5.5 Konfigurasi Pin ATMega 8535

Secara fungsional, konfigurasi pin-pin ATMega 8535 dapat dijelaskan sebagai berikut :

1. VCC

   Merupakan pin yang berfungsi sebagai pin masukan catu daya 5V.

2. GND

   Merupakan pin *ground* yang berfungsi menetralkan arus.

3. Port A (PA.0…PA.7)

   Merupakan pin I/O 8 bit *bidirectional* dan pin analog ke ADC. Pin pada *Port* A dapat menyediakan resistor *full-up internal* (dipilih untuk setiap bit).

4. Port B (PB.0…PB.7)

   Merupakan pin I/O 8 bit *bidirectional* dengan resistor *pull-up internal* (dipilih untuk setiap bit) dan pin fungsi khusus, yaitu *Timer/Counter*, komparator analog dan SPI (*Serial Peripheral interfae*).

5. Port C (PC.0…PC.7)

   Merupakan pin I/O 8 bit *bidirectional* dengan resistor *pull-up internal* (dipilih untuk setiap bit) dan pin fungsi khusus, yaitu komparator analog, *interrupt eksternal* dan *Timer Osilator*.

6. Port D (PD.0...PC.7)

Merupakan pin I/O 8 bit *bidirectional* dengan resistor *pull-up internal* dipilih untuk setiap bit) dan pin fungsi khusus, yaitu komparator analog, *interrupt eksternal* dan komunikasi serial.

7. RESET

Merupakan pin yang digunakan untuk meng-*clear*/mengembalikan semua registrasi I/O ke nilai awalnya.

8. XTAL1

Merupakan pin yang digunakan untuk meng-*clear*/mengembalikan semua registrasi I/O ke nilai awalnya.

9. XTAL 2

Merupakan pin *output* dari penguat osilator *inverting.*

10. AVCC

Merupakan pin masukan untuk tegangan ADC.

11. AREF

Merupakan pin masukan tegangan referensi ADC.

2.5.6 Fungsi Alternative Port-Port ATMega 8535

Selain berfungsi sebagai *port* I/O *bidirectional* 8 bit, masing-masing port ATMega 8535 memiliki fungsi lain, yaitu sebagai berikut :

1. Fungsi Alternatif Port A

Merupakan 8-bit *directional port* I/O. Setiap pinnya dapat menyediakan *internal pull-up* resistor (dapat diatur per bit). *Output buffer Port* A dapat memberi arus 20 mA dan dapat mengendalikan *display* LED secara langsung. *Data Direction Register port* A (DDRA) harus di *setting* terlebih dahulu sebelum *Port* A digunakan. Bit-bit DDRA diisi 0 jika ingin memfungsikan pin-pin *port* A yang bersesuaian sebagai *input*, atau diisi 1 jika sebagai *output*. Selain itu, kedelapan pin port A juga digunakan untuk masukan sinyal analog bagi A/D *converter.*

2. Fungsi Alternatif Port B

Merupakan 8-bit *directional port* I/O. Setiap pinnya dapat menyediakan

internal pull-up resistor (dapat diatur per bit). *Output buffer Port* B dapat memberi arus 20 mA dan dapat mengendalikan *display* LED secara langsung. *Data Direction Register port* B (DDRB) harus di *setting* terlebih dahulu sebelum *Port* B digunakan. Bit-bit DDRB diisi 0 jika ingin memfungsikan pin-pin *port* B yang bersesuaian sebagai *input*, atau diisi 1 jika sebagai *output.* Pin-pin port B juga memiliki untuk fungsi-fungsi alternatif khusus seperti yang dapat dilihat dalam tabel berikut.

Tabel 2.1 Fungsi Alternatif Port B

Port	**Fungsi Khusus**
PB0	T0 = timer/counter 0 external counter input
PB1	T1 = timer/counter 0 external counter input
PB2	AIN0 = analog comparator positive input
PB3	AIN1 = analog comparator negative input
PB4	SS = SPI slave select input
PB5	MOSI = SPI bus master output / slave input
PB6	MISO = SPI bus master input / slave output
PB7	SCK = SPI bus serial clock

3. Fungsi Alternatif *Port* C

Merupakan 8-bit *directional port* I/O. Setiap pinnya dapat menyediakan *internal pull-up* resistor (dapat diatur per bit). *Output buffer Port* C dapat memberi arus 20 mA dan dapat mengendalikan *display* LED secara langsung. *Data Direction Register port* C (DDRC) harus di *setting* terlebih dahulu sebelum *Port* C digunakan. Bit-bit DDRC diisi 0 jika ingin memfungsikan pin-pin port C yang bersesuaian sebagai *input*, atau diisi 1jika sebagai output. Selain itu, dua pin port C (PC6 dan PC7) juga memiliki fungsi alternatif sebagai oscillator untuk timer/counter 2.

4. Fungsi Alternatif *Port* D

Merupakan 8-bit directional port I/O. Setiap pinnya dapat menyediakan internal pull-up resistor (dapat diatur per bit). Output buffer Port D dapat memberi arus 20 mA dan dapat mengendalikan display LED secara langsung. Data Direction Register port D (DDRD) harus disetting terlebih dahulu sebelum Port D digunakan. Bit-bit DDRD diisi 0 jika ingin memfungsikan pin-pin port D yang bersesuaian sebagai input, atau diisi 1 jika sebagai output. Selain itu, pin-pin port D juga memiliki untuk fungsi- fungsi alternatif khusus seperti yang dapat dilihat dalam tabel berikut.

Tabel 2.2 Fungsi Alternatif Port D

Port Pin	Fungsi Khusus
PD0	RDX (UART input line)
PD1	TDX (UART output line)
PD2	INT0 (external interrupt 0 input)
PD3	INT1 (external interrupt 1 input)
PD4	OC1B (Timer/Counter1 output compare B match output)
PD5	OC1A (Timer/Counter1 output compare A match output)
PD6	ICP (Timer/Counter1 input capture pin)
PD7	OC2 (Timer/Counter2 output compare match output)

2.6 LCD

Menurut (Wardhana, Lingga 2006). Display LCD 16x2 berfungsi sebagai penampil karakter yang di input melalui keypad. LCD yang digunakan pada alat ini mempunyai lebar display 2 baris 8 kolom atau biasa disebut sebagai LCD Character 8x2.

Gambar 2.3 LCD 8x2

2.7 Dasar Pemrograman C

Pemrograman bahasa C pada mikrokontroler ada beberapa perbedaan sintaks pemrograman dengan instruksi bahasa C pada umumnya, terutama yang berkaitan dengan akses *register* dan memori. Sebuah program dalam bahasa C setidaknya harus memiliki fungsi. Fungsi dasar ini disebut dengan fungsi utama (*fungsi main*) dan memiliki kerangka program sebagai berikut (Hakim, 2013) :

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

Jika kita memiliki beberapa fungsi lain maka fungsi utama inilah yang memiliki kedudukan paling tinggi dibandingkan dengan fungsi-fungsi yang lain, sehingga setiap kali program dijalankan akan selalu dimulai dari memanggil fungsi utama terlebih dahulu.

Contoh :

```
// prototipe fungsi inisialisasi port
void inisialisasi_port (char A,char B,char C,char D) ;
// definisi fungsi inisialisasi port
void inisialisasi_port (char A,char B,char C,char D) ;
{
DDRA = A; DDRB = B; DDRC = C; DDRD = D;
}// fungsi utama
Void main (void)
{
inisialisasi_port (0xFF, 0xF0, 0x00);
}
```

2.6.1 Tipe Data

Berikut ini adalah tipe-tipe data yang ada dalam bahasa C dan dikenali oleh Compiler CodeVisionAVR :

Tabel 2.3 Tipe Data

Tipe Data	Ukuran	Jangkauan Nilai
Bit	1 bit	0
Char	1 byte	-128 s/d 127
Unsigned Char	1 byte	0 s/d 255
Signed Char	1 byte	-128 s/d 127
Int	2 byte	-32.768 s/d 32.767
Short Int	2 byte	-32.768 s/d 32.767
Unsigned Int	2 byte	0 s/d 65.535
Signed Int	2 byte	-32.768 s/d 32.767
Long Int	4 byte	-2.147.483.683 s/d
Unsigned Long Int	4 byte	0 s/d 4.294.967.295
Signed Long Int	4 byte	-2.147.483.683 s/d
Float	4 byte	1.2×10^{-38} s/d 3.4×10^{38}
Double	4 byte	1.2×10^{-38} s/d 3.4×10^{38}

2.6.2 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.

Deklarasi konstanta :

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

Contoh :

```
Const char konstantaku=0x10;Deklarasi Variabel :
[tipe_data][nama_variabel]=[nilai_awal]
```

Contoh :

```
Char variabelku;
Char variabelku=0x20;
```

2.6.3 Variabel Bertanda (Signed) dan Tak Bertanda (Unsigned)

Untuk mendeklarasikan tipe data yang berupa bilangan bulat yaitu *char*, *int, short* dan *long* dapat ditambahkan *signed* atau *unsigned. Signed* digunakan untuk mendefinisikan bahwa data yang disimpan dalam variabel

adalah bertanda sedangkan *unsigned* untuk data yang tidak bertanda.

Contoh :

```
Unsigned char data1;
Signed char data2;
```

Pada contoh diatas variabel *data1* bertipe *char* (1 byte) dan tidak bertanda (*unsigned*) sehingga dapat menyimpan data dari 0 sampai dengan 255. Sedangkan variabel *data2* bertipe char (1 byte) dan bertanda (*signed*)sehingga dapat menyimpan data dari -128 sampai dengan 127. Nilai negatif pada bilangan bertanda disimpan dalam bentuk komplemen 2. Misalnya untuk nilai (-1) komplemen 2-nya adalah 0xFF sehingga data 0xFF inilah yang disimpan dalam variabel tersebut.

2.6.4 Variabel Static

Variabel lokal yang nilainya ingin dipertahankan pada setiap terakhir kali pemanggilan sebuah fungsi. Contoh :

```
/*deklarasi fungsi cacah*/
int cacah (void)
{
static int c = 1;
return c++;
}
/*program utama yang akan dijalankan*/
void main (void)
{
int i;
i = cacah();
i = cacah(); }
```

Jika program utama di atas dijalankan maka pada saat fungsi *cacah* pertama kali dipanggil akan menghasilkan nilai balik 1, jika dipanggil kedua kalinya nilai baliknya berubah menjadi 2, selanjutnya jika dipanggil yang ketiga kalinya akan menghasilkan nilai balik 3 dan seterusnya. Jadi nilai balik fungsi tersebut akan selalu bertambah setiap kali fungsi dipanggil. Hal ini disebabkan karena nilai variabel *c* yang berada pada fungsi *cacah* bersifat *static* sehingga nilai variabel *c* yang terakhir akan selalu dipertahankan.

2.6.5 Komentar

Komentar digunakan untuk memberikan penjelasan, informasi ataupun keterangan-keterangan yang dapat membantu mempermudah dalam memahami kode program baik bagi si pembuat program maupun bagi orang lain yang

membacanya.

Contoh :

```
// Ini adalah komentar satu baris
/* Sedangkan yang ini adalah komentar yang lebih dari satu baris */
```

2.6.6 Pengarah Preprosesor

Pengarah preprosesor digunakan untuk mengidentifikasikan prosesor yang digunakan, dalam hal ini adalah untuk mendefinisikan jenis mikrokontroler yang digunakan.

```
# include <nama_preprosesor>
```

Contoh :

```
# include <mega8535.h>
```

2.6.7 Pernyataan

Pernyataan adalah satu buah instruksi lengkap yang berdiri sendiri. Berikut adalah contoh sebuah pernyataan :

```
PORTC = 0x0F;
```

Pernyataan di atas merupakan sebuah instruksi untuk mengeluarkan data 0x0F ke Port C.

2.6.8 Operator Aritmatika

Operator aritmatika adalah beberapa operator yang digunakan untuk melakukan perhitungan aritmatika.

Tabel 2.4 Operator Aritmatika

Operator	Keterangan
+	Operator untuk operasi penjumlahan
-	Operator untuk operasi pengurangan
*	Operator untuk operasi perkalian
/	Operator untuk operasi pembagian
%	Operator untuk operasi sisa pembagian

2.6.9 Operator Pembanding

Operator pembanding adalah operator yang digunakan untuk membandingkan dua buah data. Hasil operator pembanding bukan berupa

nilai data tetapi hanya bernilai benar („1“) atau salah („0“) saja. Berikut adalah tabel operator pembanding :

Tabel 2.5 Operator Pembanding

Operator	Contoh	Arti
= =	x = = y	Bernilai benar jika kedua data sama dan bernilai salah jika sebaliknya.
! =	X ! = y	Bernilai benar jika kedua data tidak sama dan bernilai salah jika sebaliknya.
>	x > y	Bernilai benar jika data x lebih besar dari y dan bernilai salah jika sebaliknya.
<	x < y	Bernilai benar jika data x lebih kecil dari y dan bernilai salah jika sebaliknya.
>=	x > = y	Bernilai benar jika data x lebih besar atau sama dengan y dan bernilai salah jika sebaliknya.
<=	x < = y	Bernilai benar jika data x lebih kecil atau sama dengan y dan bernilai salah jika sebaliknya.

2.6.10 Operator Logika

Operator logika digunakan untuk membentuk suatu logika atas dua buah kondisi atau lebih. Berikut adalah tabel operator logika :

Tabel 2.6 Operator Logika

Operator	Keterangan
&&	Operator untuk logika AND
||	Operator untuk logika OR
!	Operator untuk logika NOT

Contoh :

```
if ( (a == b) && (c != d) ) PORTC = 0xFF;
```

Pernyataan di atas terdiri dari 2 buah kondisi yaitu *a== b* dan *c!=d* yang keduanya dhubungkan dengan logika && (AND). Jika logika yang dihasilkan

benar maka perintah *PORTC = 0xFF* akan dikerjakan dan jika salah maka tidak dikerjakan.

2.6.11 Pernyataan Switch

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

```
switch (ekspresi)
45
{
case nilai_1 : pernyataan_1;
break;
case nilai_2 : pernyataan_2;
break;
case nilai_3 : pernyataan_3;
break;
…
default : pernyataan
default;break;
}
\
```

Pada pernyataan switch, masing-masing pernyataan 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 apabila nilai *ekspresi* tidak ada yang sama satupun dengan salah satu *nilai_1*, *nilai_2,nilai_3* dan seterusnya. Setiap akhir dari pernyataan harus diakhiri dengan *break*, karena ini digunakan untuk keluar dari pernyataan *switch*.