

BAB II

LANDASAN TEORI

2.1 Prinsip Sistem Pendingin

Sistem pendinginan adalah suatu rangkaian untuk mengatasi terjadinya *over heating* (panas yang berlebihan) pada mesin agar mesin bisa bekerja secara stabil. Fungsi dari sistem pendinginan pada kendaraan dapat dibagi menjadi empat yaitu:

1. Mencegah terjadinya *over heating*.

Panas yang dihasilkan oleh pembakaran campuran bahan bakar dengan udara dapat mencapai temperatur pada ruang bakar. Panas yang cukup tinggi ini dapat merusak logam atau bagian lain yang digunakan pada motor, hal ini disebabkan karena logam dan minyak pelumas pada suhu yang tinggi akan merusak komponen-komponen pada mesin dan apabila motor tidak dilengkapi dengan sistem pendinginan dapat merusak bagian-bagian dari motor tersebut.

2. Mempertahankan temperatur mesin

Temperatur mesin harus dipertahankan, agar selalu pada temperatur kerja yang efisien. Hal ini dapat dilakukan dengan menyerap panas yang dihasilkan oleh proses pembakaran yang berlebihan, berputarnya kipas pendingin ketika mesin dalam kondisi panas, dan katup *thermostat* yang membuka dalam kondisi mesin pada suhu kerja.

3. Mempercepat motor mencapai temperatur kerja.

Mempercepat pencapaian suhu kerja pada awal pengoperasian mesin.

4. Memanaskan ruangan di dalam ruang penumpang.

Pemanaskan ruangan di dalam ruang penumpang berlaku pada negara- negara yang mengalami musim dingin.

2.2 Prinsip kerja mesin pendingin AC

Prinsip kerja mesin pendingin atau lebih dikenal dengan AC ,sebenarnya prinsip kerja AC hanya sirkulasi saja ,mulai freon bekerja dari kompresor dalam keadaan gas tekanan tinggi , setelah itu didinginkan oleh kondensor yang nya di depan radiator ,lalu disaring oleh filter sebelum masuk ke ekspansi valve.zat pendingin yang telah diturunkan tekanan nya oleh katup ekspansi ,kemudian berubah bentuk menjadi uap dan sampai ke evaporator dalam keadaan suhu bertekanan

rendah, setelah dari evaporator lalu freon ditarik lagi oleh kompresor dan seterusnya, seperti itu lah bentuk umum kerja sistem AC.

Manfaat AC tidak hanya mampu mendinginkan udara ruangan, tetapi AC juga menjaga kelembaban udara /mengurangi kadar air. Sehingga dengan demikian ruangan yang ber-AC selain dingin juga mengandung sedikit kadar air, sebagai contoh ketika mobil melaju dalam kondisi hujan lebat dan kondisi AC mati, maka kaca mobil anda akan berembun /kondensasi dan mengganggu pandangan anda dalam berkedara, namun ketika ada menghidupkan AC maka embun pun berangsur hilang, karena udara yang keluar pada sistem AC cukup kering dan udara lembab cepat akan dihilangkan.

Udara kotor dari luar juga dibersihkan oleh evaporator ,karena sebelum udara kotor masuk kedalam ruangan terlebih dulu disaring oleh evaporator agar pendingin lebih merata maka saluran-saluran udara dingin akan turun dengan sendirinya.

Hal ini akan terbalik kalau menggunakan sistem pemanas pada bagian samping dekat kaca belakang di buat ventilasi ke luar udara dari dalam ruangan, hal ini juga dimaksudkan agar sirkulasi udara terjadi dengan baik pada bagian ruangan dekat kaca belakang.

2.3 Sensor 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. sensor suhu LM35 tidak membutuhkan kalibrasi eksternal yang menyediakan akurasi $\pm 1/4^{\circ}\text{C}$ pada temperatur ruangan dan $\pm 3/4^{\circ}\text{C}$ pada kisaran -55°C hingga $+150^{\circ}\text{C}$. Sensor suhu LM35 dimaksudkan untuk beroperasi pada -55° hingga $+150^{\circ}\text{C}$, sedangkan sensor suhu LM35C pada -40°C hingga $+110^{\circ}\text{C}$, dan sensor suhu LM35D pada kisaran

0°C-100°C. Sensor suhu LM35D juga tersedia pada paket 8 kaki dan paket TO-220. Sensor suhu LM35 umumnya akan naik sebesar 10mV setiap kenaikan 1°C (300mV pada 30 °C).

Sensor suhu LM35 mempunyai 3 pin, masing-masing pin mempunyai fungsi yang berbeda, pin 1 berfungsi sebagai sumber tegangan kerja dari sensor suhu 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 suhu LM35 yang dapat digunakan antar 4Volt sampai 30Volt. Keluaran sensor ini akan naik sebesar 10mV setiap derajat celcius sehingga diperoleh persamaan sebagai berikut :

$$V_{LM35} = \text{Suhu} * 10 \text{ mV } ^\circ\text{C}^{-1}$$

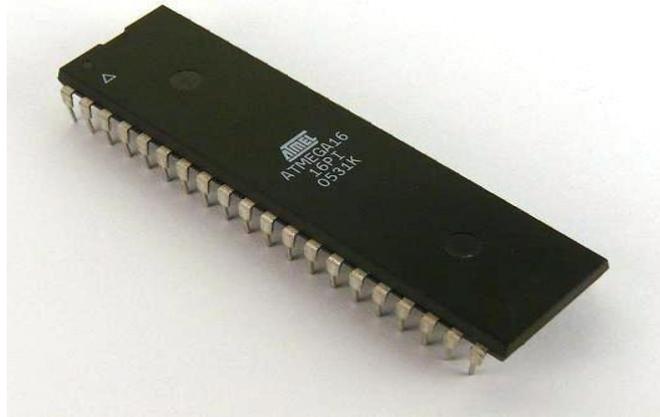
Secara prinsip sensor suhu akan melakukan penginderaan pada saat perubahan suhu setiap suhu 1°C akan menunjukkan tegangan sebesar 10mV. Pada penempatannya sensor suhu LM35 dapat ditempelkan dengan perekat atau dapat pula disemen pada permukaan akan tetapi suhunya akan sedikit berkurang sekitar 0,01°C karena terserap pada suhu permukaan tersebut. Dengan cara seperti ini diharapkan selisih antara suhu udara dan suhu permukaan dapat dideteksi oleh sensor suhu LM35 sama dengan suhu disekitarnya, jika suhu udara disekitarnya jauh lebih tinggi atau jauh lebih rendah dari suhu permukaan, maka sensor suhu 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 di dalamnya, juga dapat bertindak sebagai perata arus yang mengkoreksi pada kasus yang sedemikian, dengan menggunakan metode bypass kapasitor dari Vin untuk ditanahkan.

Berikut ini adalah karakteristik dari sensor suhu LM35 :

1. Memiliki sensitivitas suhu, dengan faktor skala linier antara tegangan dan suhu 10 mVolt/°C, sehingga dapat dikalibrasi langsung dalam celcius.
2. Memiliki ketepatan atau akurasi kalibrasi yaitu 0,5°C pada suhu 25 °C.
3. Memiliki jangkauan maksimal operasi suhu antara -55 °C sampai +150 °C.

mikroprosesor, mikrokontroler menyediakan memori dalam serpih yang sama dengan prosesornya (*in chip*). Bentuk fisik IC mikrokontroler ATmega16 di tunjukkan pada gambar 2.2.



Gambar 2.2. ATMEGA16²

2.4.1 Arsitektur ATMEGA16

Mikrokontroler ini menggunakan arsitektur Harvard yang memisahkan memori program dari memori data, baik bus alamat maupun bus data, sehingga pengaksesan program dan data dapat dilakukan secara bersamaan (*concurrent*).

Secara garis besar mikrokontroler ATMEGA16 terdiri dari :

Mikrokontroler AVR 8 bit yang memiliki kemampuan tinggi, dengan daya rendah

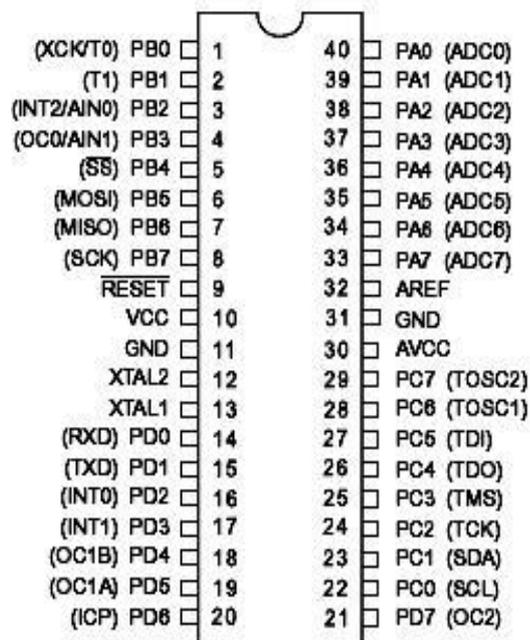
1. Arsitektur RISC dengan throughput mencapai 16 MIPS pada frekuensi 16Mhz.
2. Memiliki kapasitas Flash memori 16Kbyte, EEPROM 512 Byte, dan SRAM 1Kbyte
3. Saluran I/O 32 buah, yaitu Port A, Port B, Port C, dan Port D.
4. CPU yang terdiri dari 32 buah register.
5. User interupsi internal dan eksternal
6. Port antarmuka SPI dan Port USART sebagai komunikasi serial
7. Fitur Peripheral

² http://store.nerokas.co.ke/index.php?route=product/category&path=135_136, diakses : oktober 2016

- Dua buah 8-bit *timer/counter* dengan prescaler terpisah dan mode *compare*
- Satu buah 16-bit timer/counter dengan prescaler terpisah, mode *compare*, dan mode *capture*
- *Real time counter* dengan osilator tersendiri
- Empat kanal PWM dan Antarmuka komparator analog
- 8 kanal, 10 bit ADC
- *Byte-oriented Two-wire Serial Interface*
- *Watchdog timer* dengan osilator internal

2.4.2 Konfigurasi Pin ATMEGA16

Konfigurasi *pin* mikrokontroler Atmega16 dengan kemasan 40 pin dapat dilihat pada Gambar 2.1 Dari gambar tersebut dapat terlihat ATmega16 memiliki 8 pin untuk masing-masing *Port A*, *Port B*, *Port C*, dan *Port D*. gambar pin ATmega16 ditunjukkan pada gambar 2.3.



Gambar 2.3. Port-port Atmega16³

³ <http://www.atmel.com/images/doc2466.pdf>, diakses : oktober2016

2.4.3 Deskripsi Mikrokontroler ATMEGA16

- VCC (*Power Supply*) dan GND (*Ground*)

VCC merupakan pin yang berfungsi sebagai masukan catu daya

- GND (*Ground*)

GND merupakan pin *ground*

- Port A (PA7..PA0)

Port A berfungsi sebagai *input* analog pada konverter A/D. Port A juga sebagai suatu port I/O 8-bit dua arah, jika A/D konverter tidak digunakan. Pin-pin port dapat menyediakan resistor *internal pull-up* (yang dipilih untuk masing-masing bit). Port A *output* buffer mempunyai karakteristik gerakan simetris dengan keduanya sink tinggi dan kemampuan sumber. Ketika pin PA0 ke PA7 digunakan sebagai input dan secara eksternal ditarik rendah, pin-pin akan memungkinkan arus sumber jika resistor *internal pull-up* diaktifkan. Pin port A adalah tri-stated manakala suatu kondisi reset menjadi aktif, sekalipun waktu habis.

- Port B (PB7..PB0)

Port B adalah suatu port I/O 8-bit dua arah dengan resistor *internal pull-up* (yang dipilih untuk beberapa bit). Port B output buffer mempunyai karakteristik gerakan simetris dengan keduanya sink tinggi dan kemampuan sumber. Sebagai input, pin Port B yang secara eksternal ditarik rendah akan arus sumber jika resistor *pull-up* diaktifkan. Pin port B adalah tri-stated manakala suatu kondisi reset menjadi aktif, sekalipun waktu habis.

- Port C (PC7..PC0)

Port C adalah suatu port I/O 8-bit dua arah dengan resistor *internal pull-up* (yang dipilih untuk beberapa bit). Port C output buffer mempunyai karakteristik gerakan simetris dengan keduanya sink tinggi dan kemampuan sumber. Sebagai input, pin port C yang secara eksternal ditarik rendah akan arus sumber jika resistor *pull-up* diaktifkan. Pin port C adalah tri-stated manakala suatu kondisi reset menjadi aktif, sekalipun waktu habis.

- Port D (PD7..PD0)

Port D adalah suatu port I/O 8-bit dua arah dengan resistor *internal pull-up* (yang dipilih untuk beberapa bit). Port D output buffer mempunyai karakteristik

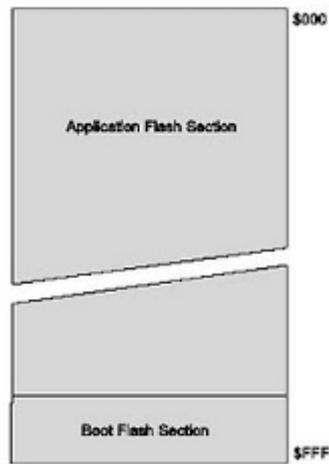
gerakan simetris dengan keduanya sink tinggi dan kemampuan sumber. Sebagai input, pin port D yang secara eksternal ditarik rendah akan arus sumber jika resistor *pull-up* diaktifkan. Pin port D adalah tri-stated manakala suatu kondisi reset menjadi aktif, sekalipun waktu habis.

- RESET (*Reset input*)
- XTAL1 (*Input Oscillator*)
- XTAL2 (*Output Oscillator*)
- AVCC adalah pena penyedia tegangan untuk bandar A dan Konverter A/D.
- AREF adalah pena referensi analog untuk konverter A/D.

2.4.4 Struktur Memori ATmega16

2.4.4.1 Memori Program

Arsitektur ATmega16 mempunyai dua memori utama, yaitu memori data dan memori program. Selain itu, ATmega16 memiliki memori EEPROM untuk menyimpan data. ATmega16 memiliki 16K byte *On-chip In-System Reprogrammable Flash Memory* untuk menyimpan program. Instruksi ATmega16 semuanya memiliki format 16 atau 32 bit, maka memori *flash* diatur dalam 8K x 16 bit. Memori *flash* dibagi kedalam dua bagian, yaitu bagian program *boot* dan aplikasi seperti terlihat pada Gambar 2.4. *Bootloader* adalah program kecil yang bekerja pada saat sistem dimulai yang dapat memasukkan seluruh program aplikasi ke dalam memori prosesor. Gambar peta memori program ATmega16 AVR ditunjukkan pada gambar 2.4.

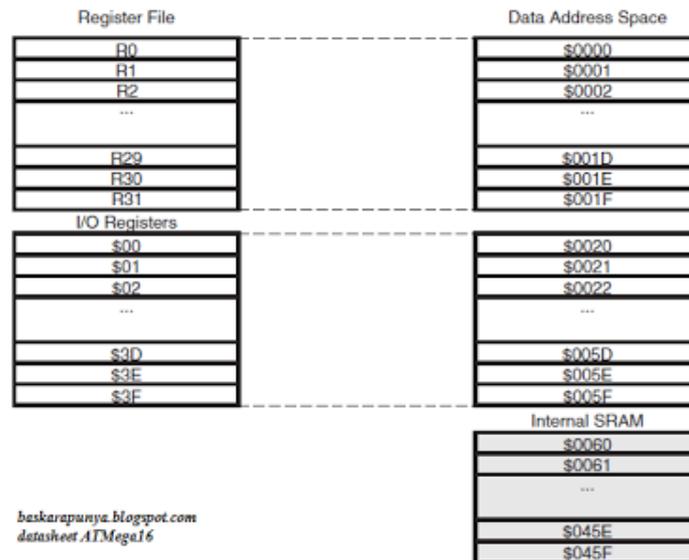


Gambar 2.4. Peta Memori Program AVR ATMEGA16⁴

2.4.4.2 Memori Data (SRAM)

Memori data AVR ATMega16 terbagi menjadi 3 bagian, yaitu 32 register umum, 64 buah register I/O dan 1 Kbyte SRAM internal. *General purpose Register* menempati alamat data terbawah, yaitu \$00 sampai \$1F. Sedangkan memori I/O menempati 64 alamat berikutnya mulai dari \$20 hingga \$5F. Memori I/O merupakan register yang khusus digunakan untuk mengatur fungsi terhadap berbagai fitur mikrokontroler seperti kontrol register, *timer/counter*, fungsi-fungsi I/O, dan sebagainya. 1024 alamat berikutnya mulai dari \$60 hingga \$45F digunakan untuk SRAM internal. Gambar peta memori data ATmega16 ditunjukkan pada gambar 2.5.

⁴ <http://www.atmel.com/images/doc2466.pdf>, diakses : oktober 2016



Gambar 2.5. Peta Memori Data AVR ATMEGA16⁵

2.4.4.3 Memori Data EEPROM

ATmega16 terdiri dari 512 byte memori data EEPROM 8 bit, data dapat tulis/baca dari memori ini, ketika catu daya dimatikan, data terakhir yang ditulis pada memori EEPROM masih tersimpan pada memori ini, atau dengan kata lain memori EEPROM bersifat *nonvolatile*. Alamat EEPROM mulai dari \$000 sampai \$1FF=.

2.5 Bahasa Pemrograman Mikrokontroler AVR ATmega16

Pengembangan sebuah sistem menggunakan mikrokontroler AVR buatan ATMEL menggunakan *software* AVR STUDIO dan CodeVision AVR. AVR STUDIO merupakan *software* khusus untuk bahasa *assembly* yang mempunyai fungsi sangat lengkap, yaitu digunakan untuk menulis program, kompilasi, simulasi dan *download* program ke IC mikrokontroler AVR. Sedangkan CodeVision AVR merupakan *software C-cross compiler*, dimana program dapat ditulis dalam bahasa C, CodeVision memiliki IDE (Integrated Development Environment) yang lengkap, dimana penulisan program, *compile*, *link*, pembuatan kode mesin (*assembler*) dan *download* program ke *chip* AVR dapat dilakukan pada CodeVision, selain itu ada fasilitas terminal, yaitu untuk melakukan komunikasi

⁵ <http://www.atmel.com/images/doc2466.pdf>, diakses : oktober2016

serial dengan mikrokontroler yang sudah diprogram. Proses *download* program ke IC mikrokontroler AVR dapat menggunakan *downloader* secara ISP (*In-System Programming*). *In-System Programming flash on-chip* mengizinkan memori program untuk diprogram ulang dalam sistem menggunakan hubungan serial SPI.

2.5.1 Bahasa C

Berikut ini penjelasan aturan penulisan program dalam bahasa C. Untuk seterusnya, pemrograman mikrokontroler AVR menggunakan bahasa C dengan penjelasan sebagai berikut:

Penulisan program dalam bahasa C

```
#include<mega16.h>
#include <delay.h>
#define      IRsensor      PINA.0
#define      pompa         PORTB.0
//variable global
unsigned int I,j;
void main(void)
{
//variable local
Chart data_rx;
DDRA=0x00;
PORTA=0xFF;
DDRB=0xFF;
PORTB=0x00;
....
....
While(1)
{
.....
.....
};
}
```

Penjelasan:

Preprocessor (#) : digunakan untuk memasukan (*include*) *text* dan *file* lain, mendefinisikan *macro* yang dapat mengurangi beban kerja pemrograman dan meningkatkan *legibility source code* (mudah dibaca).

#define : digunakan untuk mendefinisikan *macro*

Contoh	#define	ALFA	0xff
	#define	SUM(a,b)	a+b
	#define	sensor	PINA.2
	#define	pompa	PORTB.0

Komentar :

penulisan komentar untuk beberapa baris komentar sekaligus

```
/*...komentar...*/
```

Penulisan untuk satu baris saja

```
//...komentar...
```

2.5.1.1 Identifiers

Identifier adalah nama yang diberikan pada *variable*, fungsi, label, atau objek lain. *Identifier* dapat mengandung huruf (A ...Z, a ...z) dan angka (0 ... 9) dan karakter (_). *Identifiers* bersifat *Case sensitive*. *Indetifier* dapat mencapai maksimal 32 karakter.

2.5.1.2 Konstanta

Konstanta *integer* dan *long integer* ditulis dalam bentuk *decimal* (1234), dalam bentuk biner (0b101001), *hexadecimal* mempunyai awalan 0x (0xff), atau Dalam *octal* dengan awalan 0 (0777).

Unsigned integer mempunyai akhiran U(10000U)

Long integer mempunyai akhiran L(99L)

Unsigned long integer mempunyai akhiran UL(99UL)

Floating point mempunyai akhiran F(1.234F)

Konstanta karakter harus di lingkungan oleh tanda kutip ('a')

2.5.1.3 Tipe Data

Tabel 2.1. Tipe data⁶

Tipe	Ukuran (bit)	Rentang (Range)
Bit	1	0, 1 (tipe data bit hanya dapat digunakan untuk variable global)
Char	8	-128 to 127
Unsigned char	8	0 to 255
signed char	8	-128 to 127
Int	16	-32768 to 32767
Short int	16	-32768 to 32767
Unsigned int	16	0 to 65535
Signed int	16	-32768 to 32767
Long int	32	-2147463648 to 2147483647
Unsigned long int	32	0 to 4294967295
Signed long int	32	-2147483648 to 2147483647
Float	32	$\pm 1.175e-38$ to $\pm 3.402e38$
Double	32	$\pm 1.175e-38$ to $\pm 3.402e38$

2.5.1.4 Operator

Tabel 2.2. Daftar Operator Kondisi⁷

Operator Kondisi	Keterangan
<	Lebih kecil
<=	Lebih kecil atau sama dengan
>	Lebih besar
>=	Lebih besar sama dengan
==	Sama dengan
!=	Tidak sama dengan

⁶ Heri Andrianto, Op.Cit., 25

⁷ Ibid., 26

Tabel 2.3. Daftar Operator Aritmatika⁸

Operator Aritmatika	Keterangan
+	Penjumlahan
-	Pengurangan
*	Perkalian
/	Pembagian

Tabel 2.4. Daftar Operator Logika⁹

Operator Logika	Keterangan
!	<i>Boolean NOT</i>
&&	<i>Boolean AND</i>
	<i>Boolean OR</i>

Tabel 2.5. Daftar Operator *Bitwise*¹⁰

Operatot <i>Bitwise</i>	Keterangan
~	Komplemen <i>Bitwise</i>
&	<i>Bitwise AND</i>
	<i>Bitwise OR</i>
^	<i>Bitwise Exclusive OR</i>
>>	<i>Right Shift</i>
<<	<i>Left Shift</i>

⁸ Loc.Cit.

⁹ Loc.Cit.

¹⁰ Loc.Cit.

Tabel 2.6. Daftar Operator *Assignment*¹¹

Operator <i>Assignment</i>	Keterangan
=	Untuk memasukkan nilai
+=	Untuk menambah nilai dari keadaan semula
-=	Untuk mengurangi nilai dari keadaan semula
*=	Untuk mengalikan nilai dari keadaan semula
/=	Untuk melakukan pembagian terhadap bilangan semula
%=	Untuk memasukkan nilai sisa bagi dari pembagin bilangan semula
<<=	Untuk memasukkan <i>shift left</i>
>>=	Untuk memasukkan <i>shift Right</i>
&=	Untuk memasukkan nilai <i>bitwise AND</i>
^=	Untuk memasukkan nilai <i>bitwise XOR</i>
\ =	Untuk memasukkan nilai <i>bitwise OR</i>

2.5.1.5 Program Control

a. Percabangan

Perintah *if* dan *if ... else ...*

Perintah *if* dan *if ... else ...* digunakan untuk melakukan operasi percabangan bersyarat. Fungsi-fungsi untuk menetapkan kondisi dapat dilihat dalam tabel.

Sintaks penulisan *if* dapat ditulis sebagai berikut:

$$if(<expression>)<statement>;$$

Sintak perintah *if ... else ...* dapat ditulis sebagai berikut:

$$if(<expression>)<statement>;$$

$$else<statement2>;$$

jika hasil *testing expression* memberikan hasil tidak nol *statement1* akan dilaksanakan. Pada keadaan sebaliknya *statement2* yang akan dilaksanakan.

¹¹ Ibid., 27

Sebaiknya pemanfaatan perintah *if* untuk beberapa kondisi dilakukan dengan menggunakan blok-blok

Percabangan *switch*

Perintah percabangan *if ... else ...* dapat digantikan dengan perintah *switch*.

Dalam pernyataan *switch*, sebuah variable secara berurutan diuji oleh beberapa konstanta bilangan bulat atau konstanta karakter. Sintaks perintah *switch* dapat ditulis sebagai berikut:

```
Switch(variable)
{
    Case konstanta_1:  statement;
                    Break;
    Case konstanta_2:  statement;
                    Break;
    Case konstanta_n:  statement;
                    Break;
    Default:          statement;
}
```

Hal-hal yang perlu diperhatikan:

1. *Switch* hanya dapat memeriksa *variable* terhadap sebuah konstanta, sedangkan *if* dapat memeriksa persyaratan perbandingan (lebih besar, lebih kecil, dan seterusnya).
2. Tidak ada dua konstanta yang sama di dalam sebuah *switch*.
3. Perintah *switch* jika dimanfaatkan dengan tepat dapat memberikan hasil yang lebih baik daripada perintah *if ... else ...* yang membentuk tangga dan/atau bersarang.

b. Looping (Pengulangan)

Looping adalah pengulangan satu atau beberapa perintah sampai mencapai keadaan tertentu. Ada tiga perintah *looping*, yaitu: *for ...*, *while ...*, dan *do...while....*. Sintaks *loop for* dapat dituliskan sebagai berikut:

for

untuk pengulangan yang melakukan proses *increment*

```
for(nama_variable= nilai_awal;syarat_loop;nama_variable ++)  
{  
Statement_yang_diulang;  
}
```

//untuk pengulangan yang melakukan proses *decrement*
for(nama_variable=nilai_awal;syarat_loop;nama_variable --)
{
Statement_yang_diulang;
}

Syarat_loop adalah pernyataan rasional yang menyatakan syarat berhentinya pengulangan, biasanya berkaitan dengan *variable control*, *nama_variable++* dan *nama_variable--*, menyatakan proses *increment* dan proses *decrement* pada *variable* kontrol.

While

Perintah *while* dapat melakukan *looping* apabila persyaratannya benar.

Sintaks perintah *while* dapat dituliskan sebagai berikut:

```
Nama_variable = nilai_awal;  
while(syarat_loop)  
{  
Statement_yang_akan_diulang;  
Nama_variable++;  
}
```

do ...while

Perintah *while* terlebih dahulu melakukan pengujian persyaratan sebelum melakukan *looping*. Kadang-kadang hal ini menimbulkan keropotan-kerepotan yang tidak perlu, misalnya inisialisasi *variable control*. Salah satu solusi adalah dengan menggunakan loop *do ...while*.

```
Nama_variable = nilai_awal;  
do
```

```

{
statement_yang_akan_diulang;
nama_variable ++;
}
while(syarat_loop)

```

2.5.1.6 Array

Array adalah deretan variable yang berjenis sama dan mempunyai nama yang sama. Setiap anggota deretan (elemen) diberi nomor yang disebut indeks, dimulai dari indeks nol. *Array* diatur agar mempunyai lokasi memori yang bersebelahan dengan alamat terkecil menunjuk elemen *array* pertama dan alamat terbesar menunjukkan elemen terakhir. Elemen *array* dapat diakses dengan menggunakan indeksnya. Bentuk deklarasi array adalah:

```

Tipe nama_array[ukuran]
Int nilai[100];
Nilai[1]=10;
Niali[2]=3;

```

2.5.1.7 Fungsi

Sebuah program yang besar dapat dipecah menjadi beberapa subprogram yang terpisah yang melakukan fungsi tertentu. Subprogram yang seperti itu disebut fungsi. Sebagai contoh, sebuah program yang melakukan proses pengisian data berulang kali dapat dilengkapi dengan sebuah *fungsi* yang bertugas untuk melakukan proses pengisian data. Apabila program hendak melakukan proses pengisian data, program dapat melakukan pemanggilan *fungsi* tersebut.

Fungsi adalah sebuah blok yang melingkupi beberapa perintah. Deklarasi *fungsi* dapat dilakukan dengan cara:

```
Tipe nama_fungsi(argumen)
```

Parameter dalam *fungsi* dijelaskan sebagai berikut:

Tipe adalah nilai yang dihasilkan oleh *fungsi*, jika tidak dinyatakan, hasil *fungsi* dianggap bertipe integer. Deklarasi tipe *void* dapat dimanfaatkan untuk menghindari terjadinya nilai balik.

Argumen adalah deklarasi variable apa saja yang dibutuhkan *fungsi* dan bersifat *optional*.

a. Fungsi dengan nilai balik

Fungsi ini memberikan hasil yang berupa nilai

Fungsi dengan nilai *balik* (*return value*).

Contoh:

```
long luas()
{
int sisi=10;
return (sisi*sisi);
}
```

b. Fungsi tanpa nilai balik

Fungsi ini tidak memberikan hasil yang berupa nilai melainkan berupa sebuah proses. Fungsi ini bertipe *void*.

Contoh:

```
void kedip( )
{
PORTD=0;
Delay_ms(500); //delay 500 ms
PORTD=255;
delay_ms(500);
PORTD=0;
delay_ms(500);
return;
}
```

Pernyataan *return*

Pernyataan *return* dapat menyatakan dua hal:

1. *Return* mengakhiri jalanan *fungsi* dan kembali ke program utama.
2. Mengirim nilai balik *Fungsi* dapat ditulis pada akhir program dengan membuat sebuah *prototype function* dibagian awal program. Cara menulis

fungsi yang seperti itu memberikan kemudahan kepada *programmer* untuk memeriksa dan membaca ulang sebuah program yang besar.

c. Parameter dalam sebuah fungsi

Parameter dalam sebuah fungsi dibagi dua yaitu *parameter formal* dan *parameter actual*. *Parameter formal* adalah parameter pada saat fungsi itu dibuat, sedangkan *parameter actual* adalah parameter pada saat pemanggilan fungsi.

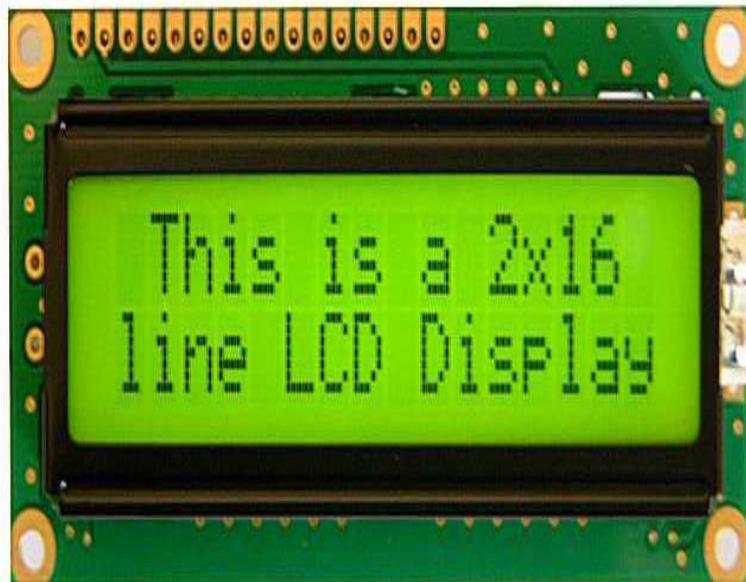
Contoh:

```
//mendeklarasikan fungsi
long func(int param_1, int param_2);
//mendefinisikan fungsi,param_1 dan param_2 disebut parameter formal
long func(int param_1, int param_2)
{
return param_1*param_2;
}
//memanggil fungsi dan mengisikan nilai yang dihasilkan ke sebuah variabel
x, nilai 20 dan 30 disebut parameter actual
x=func(20, 30); // mengisikan parameter _1=20 dan parameter_2=30
//hasil perkaliannya disimpan ke variable x
```

2.6 LCD Karakter 2x16

Liquid Crystal Display (LCD) adalah suatu perangkat elektronika yang dirancang sedemikian rupa, sehingga dapat menampilkan tulisan maupun gambar. LCD banyak digunakan sebagai layar tampilan pada berbagai jenis aplikasi elektronika, seperti monitor komputer, televisi, seluler dan lain-lain sebagainya. LCD yang khusus hanya untuk menampilkan tulisan disebut LCD karakter. Sedangkan biasanya disebutkan jumlah kolom dan barisnya misalnya 16X2, yang berarti terdapat 16 kolom dan 2 baris.

LCD memiliki suatu controller yang berfungsi untuk mengontrol tampilan layar LCD secara keseluruhan. Controller pada modul LCD menerima instruksi dan data dari suatu prosesor atau mikrokontroler untuk menentukan karakter apa yang akan ditampilkan pada layar LCD tersebut. Pada umumnya LCD 16X2 mampu mengerjakan seluruh instruksi yang didukung oleh controller jenis HD44780. Jika tidak, instruksi untuk modul LCD tersebut dapat dilihat dari datasheet yang disediakan oleh pabrik pembuatan. Gambar fisik LCD 16 x 2 ditunjukkan pada Gambar 2.7. dan modul LCD pada umumnya terdiri dari 14 pin, tetapi LCD yang memiliki backlight mempunyai 16 pin, yaitu 2 pin tambahan untuk menyalakan LED backlight. Berikut table fungsi pin LCD 16x2.



Gambar 2.6. LCD 2x16¹²

¹² Sumber:kl801.ilearning.me

Tabel 2.7. Fungsi Pin LCD Karakter 16x2¹³

PI N	Nama	Fungsi
1	VSS	Ground Voltage
2	VCC	+5V
3	VEE	Contrast Voltage
4	RS	Register Select: 0 = Send Instruction 1 = Send Data
5	R/W	Read/Write, to choose write or read mode : 0 = Write Mode 1 = Read Mode
6	EN	Enable Signal : 0 = start to lacht data to LCD character 1 = disable
7	DB0	Data bit ke-0 H/L (LSB)
8	DB1	Data bit ke-1 H/L
9	DB2	Data bit ke-2 H/L
10	DB3	Data bit ke-3 H/L
11	DB4	Data bit ke-4 H/L
12	DB5	Data bit ke-5 H/L
13	DB6	Data bit ke-6 H/L
14	DB7	Data bit ke-7 H/L (MSB)
15	ANODE	Backlight (+)
16	KATODE	Backlight (-)

Cara mengirimkan instruksi untuk dieksekusi oleh controller LCD:

1. Set supaya pin RS = 0, R/W = 0, E = 1.
2. Kemudian kirim data berupa instruksi untuk dieksekusi controller pada LCD melalui DB0 - DB7 (pin 7 – pin 14).

¹³ Sumber:kl801.ilarning.me

3. Set supaya pin E = 0, kemudian berikan delay sesaat, dan set kembali pin E = 1.

Cara mengirimkan karakter atau data untuk dicetak pada layar LCD:

1. Set supaya pin RS = 1, R/W = 0, E = 1.
2. Kemudian kirimkan data berupa ASCII dari karakter yang ingin ditampilkan pada layar LCD melalui jalur DB0 – DB7 (pin7 - pin14).
3. Set supaya pin E = 0, kemudian berikan delay sesaat, dan set kembali pin E = 1.

2.7 Kipas Pendingin (extra fan) dan motor blower

Extra fan adalah komponen pendingin berupa fan tambahan pada sistem pendingin mesin berupa motor listrik dan digerakan oleh tegangan listrik. Extra fan pada mobil berfungsi untuk membantu mendinginkan radiator dan mesin saat mesin berkerja.



Gambar 2.7. Motor Extra fan¹⁴

¹⁴ Sumber: manual book opel blazer



Gambar 2.8. Motor Blower¹⁵

Blower adalah komponen pendingin berupa motor listrik Pada sistem air conditioner atau ac mobil, blower ini terletak di belakang evaporator, yang memiliki fungsi untuk menghembuskan udara ke arah evaporator. Seperti yang kita tahu bahwa udara yang dihembuskan oleh blower ini akan melewati sirip-sirip evaporator dan turun suhunya menjadi lebih dingin karena panasnya diserap oleh refrigerant atau freon.

Baik motor extra fan dan motor blower keduanya adalah motor DC. motor listrik yang memerlukan suplai tegangan arus searah pada kumparan medan untuk diubah menjadi energi gerak mekanik. Kumparan medan pada motor dc disebut stator (bagian yang tidak berputar) dan kumparan jangkar disebut rotor (bagian yang berputar). Motor arus searah, sebagaimana namanya, menggunakan arus langsung yang tidak langsung(direct-unidirectional).

Motor DC memiliki 3 bagian atau komponen utama untuk dapat berputar sebagai berikut.

1. Kutub medan. Motor DC sederhana memiliki dua kutub medan: kutub utara dan kutub selatan. Garis magnetik energi membesar melintasi ruang terbuka diantara kutub-kutub dari utara ke selatan. Untuk motor yang lebih besar atau lebih kompleks terdapat satu atau lebih elektromagnet.

¹⁵ Sumber: manual book opel blazer

2. Current Elektromagnet atau Dinamo. 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.

3. Commutator. Komponen ini terutama ditemukan dalam motor DC. Kegunaannya adalah untuk transmisi arus antara dinamo dan sumber daya.