BAB II

LANDASAN TEORI

2.1 Pompa Air

Mesin pompa air pada dasarnya terdiri dari dua jenis dilihat dari cara kerja dan rancangannya. Jenis pompa tersebut adalah pompa sistem rotari dan pompa sistem sentrifugal.

a. Pompa air sistem rotari

   Pompa jenis ini memiliki *impeller* yang berputar untuk menimbulkan kekuatan tarikan, sehingga air yang dipindahkan akan mampu terus menerus menarik air dari dasar sumur untuk dialirkan menuju ke pipa outlet. Jenis pompa ini banyak dipergunakan pada pompa air untuk kebutuhan rumah tangga. Hampir semua jenis pompa kecil menggunakan sistem kerja rotari.

b. Pompa air sistem sentrifugal

   Pompa jenis ini banyak dipergunakan pada peralatan *marine* atau kapal laut untuk membuang air dari dok kapal secara cepat. Jenis pompa ini bekerja dengan kecepatan tinggi, sehingga *volume* air yang bergerak secara memutar dapat terlempar keluar dari *outlet* air.

2.1.1 Prinsip Kerja Pompa Air

Sebuah pompa air bekerja dengan cara memindahkan sejumlah air melalui ruang *suction* menuju keruang *outlet* dengan menggunakan *impeller*, sehinggga seluruh ruangan udara terisi oleh air dan menimbulkan tekanan *fluida* untuk ditarik melalui dasar sumur menuju penampungan. Air yang terdapat dalam ruang *impeller* akan digerakkan sebuah motor. Selama motor berputar air akan terus didorong keluar menuju ke pipa penyaluran atau *outlet* pompa.

2.1.2 Bagian-Bagian Pompa Air

a. Motor

   Bagian ini merupakan bagian utama dari pompa air dengan menggunakan motor tersebut sebuah pompa baik dari jenis sentrifugal maupun rotari dapat berfungsi.

b. Valve

   Bagian ini berfungsi untuk memisahkan bagian hisap dan bagian pompa, sehingga terjadi perbedaan tekanan dan pemisahan air.selain terdapat dalam ruang kompresi mesin jenis tertentu, *valve* ini juga terdapat pada ujung pipa untuk menjaga agar ruangan pompa air terus terisi air dan tidak diisi oleh udara. *Valve* juga dipergunakan untuk melakukan pengendalian terhadap tekanan pompa

air agar terhindar dari kerusakan secara otomatis. *Valve* tersebut umumnya dihubungkan dengan saklar pemutus arus atau *relay*. Jika ouput pompa air mengalami peningkatan tekanan hingga tekanan tertentu, *valve* atau membrane tersebut akan terdorong ke atas dan memutus arus listrik secara otomatis.

c. Saklar otomatis pompa air

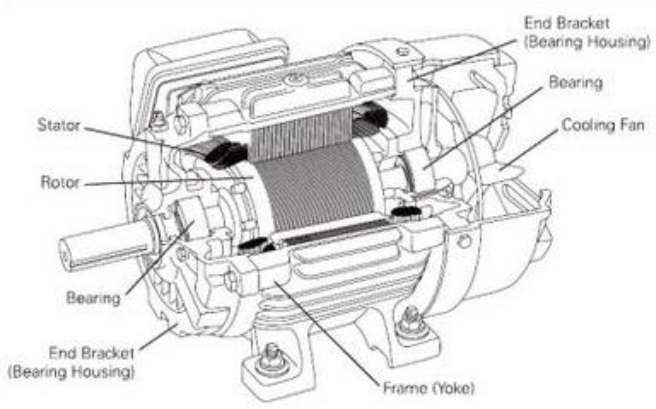
Saklar otomatis ini bertugas melindungi pompa dari kelebihan beban pada keadaan tertentu. Air pada bagian *output* akan terputus atau memiliki beban yang sangat besar. Kondisi tersebut dapat menimbulkan kerusakan motor jika dibiarkan. Untuk melindungi pompa dari kerusakan terdapat sebuah saklar otomatis pada jenis semua pompa air. Berikut merupakan contoh bentuk saklar atau *relay* yang dipergunakan pada pompa air.

d. Kapasitor

Kapasitor terdapat pada pompa air untuk membantu *start* motor penggerak agar lebih cepat berputar.

e. Tangki penampung

Tangki penampung berfungsi untuk menampung air sementara waktu agar pompa tidak hidup danmati setiap saat menghidupkan air di kran. Dengan menggunakan tangki penampung air penggunaan listrik menjadi lebih hemat dan pompa menjadi lebih awet karena tidak terus menerus mengalami tekanan besardalam waktu singkat.



Gambar 2.1 Konstruksi pompa air[1]

[1]http://jakartapiranti.com/blog/kenali-kerusakan-mesin-pompa-air-di-rumah-anda: 3 Maret 2017

2.2 LCD Karakter 2x16

Liquid Crystal Display (LCD) adalah suatu perangkat elektronika yang dirancang sedemikan 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. *Controll*er pada modul *LCD* menerima instruksi dan data dari suatu prosesor atau mikrokontroller 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 2.2

Gambar 2.2 LCD 2x16[2]

Modul *LCD* pada umumnya terdiri dari 14 pin, tetapi *LCD* yang memiliki backlight mempunyai 16 pin, yaitu 2 pin tambahan untuk menyalakan *LED backlight*.

Cara mengirimkan instruksi untuk dieksekusi oleh *controller LCD*:

a. Set supaya pin RS = 0, R/W = 0, E = 1.
b. Kemudian kirim data berupa instruksi untuk dieksekusi *controller* pada *LCD* melalui DB0 - DB7 (pin 7 – pin 14).
c. Set supaya pin E = 0, kemudian berikan *delay* sesaat, dan set kembali pin E = 1.

[2]https://bekoy.wordpress.com/2012/02/15/pemrograman-lcd-karakter-2x16-menggunakan-cv-avr: 10 Maret 2017

Cara mengirimkan karakter atau data untuk dicetak pada layar *LCD*:

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

Tabel 2.1 Fungsi Pin LCD Karakter 2x16[3]

PIN	**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 (-)

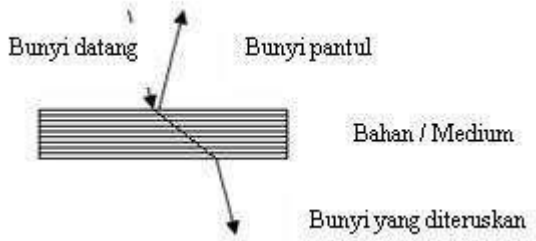
[3]http://myactivities-mazen.blogspot.co.id/2013/09/controlling-216-lcd-with-stm32f4.html: 10 Maret 2017

2.3 Sensor Ultrasonik

Sensor Ultrasonik merupakan gelombang akustik yang memiliki frekuensi mulai 20 KHz hingga sekitar 20 MHz. Frekuensi kerja yang digunakan dalam gelombang ultrasonik bervariasi tergantung pada medium yang dilalui, mulai dari kerapatan rendah pada fase gas, cair hingga padat. Jika gelombang ultrasonik berjalan melalui sebuah medium, secara matematis besarnya jarak dapat dihitung sebagai berikut:

$$S = v.t/2$$

Dimana s adalah jarak satuan meter, v adalah kecepatan suara yaitu 344m/detik dan t adalah waktu tempuh dalam satuan detik. Ketika gelombang ultrasonik menumbuk suatu penghalang maka sebagian gelombang tersebut akan dipantulkan sebagian diserap dan sebagian yang lain akan diteruskan. Proses ini ditunjukkan pada gambar 2.3



Gambar 2.3 Fenomena gelombang ultrasonik saat ada penghalang

Sensor ultrasonik adalah sebuah sensor yang mengubah besaran *fisis* (bunyi) menjadi besaran listrik. Pada sensor ini gelombang ultrasonik dibangkitkan melalui sebuah benda yang disebut *piezoelektrik*. *Piezoeletrik* ini akan menghasilkan gelombang ultrasonik dengan frekuensi 40 KHz ketika sebuah *osilator* diterapkan pada benda tersebut. sensor ultrasonik secara umum digunakan untuk pengukapan tak sentuh yang beragam seperti aplikasi pengukuran jarak. Alat ini secara umum memencarkan gelombang suara ultrasonik menuju suatu target yang memantulkan balik gelombang kearah sensor. Kemudian sistem mengukur waktu yang diperlukan untuk pemancaran gelombang sampai kembali kesensor dan menghitung jarak target dengan menggunakan kecepatan suara dalam medium.

Rangkaian penyusun sensor ultrasinok ini terdiri dari ***transmitter, reiceiver, dan komparator,*** selain itu, gelombang ultrasonik dibangkitkan oleh sebuah Kristal tipis bersifat ***piezoeletrik.*** Bagian-bagian dari sensor ultrasonik adalah sebagai berikut:

a. Piezoeletrik

Peralatan *piezoeletrik* secara langsung mengubah energi listrik menjadi energi mekanik. Tegangan *input* yang digunakan menyebabkan bagian keramik meregang dan memancarkan gelombang ultrasonik. Tipe operasi transmisi elemen *piezoeletrik* sekitar frekuensi 32 KHz. Efisiensi lebih baik, jika frekuensi *osilator* diatur pada frekuensi resonansi *piezoelektrik* dengan sensitifitas dan efisiensi paling baik. Jika rangkaian pengukur beroperasi pada mode pulsa elemen *piezoelektrik* yang sama dapat digunakan sebagai *transmitter* dan *receiver*. Frekuensi yang disesuaikan frekuensi kerja dari masing-masing *transduser*. Karena kelebihannya inilah maka *tranduser piezoelektrik* lebih sesuai digunakan untuk sensor ultrasonik.

b. Transmitter

Transmitter adalah sebuah alat yang berfungsi sebagai pemancar gelombang ultrasonik dengan frekuensi sebesar 40 KHz yang dibangkitkan dari sebuah *osilator*. Untuk mengasilkan frekuensi 40 KHz, harus dibuat sebuah rangkaian *osilator* dan keluaran dari *osilator* dilanjutkan menuju penguat sinyal. Besaranya frekuensi ditentukan oleh komponen kalang *RLC* / Kristal tergantung dari desain *osilator* yang digunakan. Penguat sinyal akan memberikan sebuah sinyal listrik akan memberikan sebuah sinyal listrik yang diumpankan ke *piezoelektrik* dan terjadi reaksi mekanik sehingga bergetar dan memancarakan gelombang yang sesuai dengan besar frekuensi pada *osilator*.

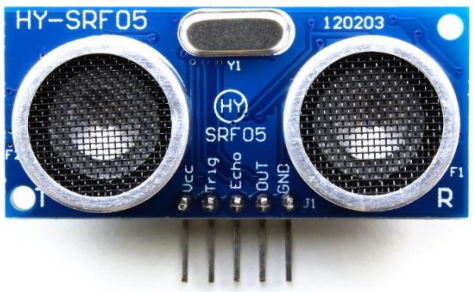
c. Receiver

Receiver terdiri dari *transduser* ultrasonik menggunakan bahan *piezoeletrik*, yang berfungsi sebagai penerima gelombang pantulan yang berasal dari *transmitter* yang dikenakkan pada permukaan suatu benda atau gelombang langsung ***LOS (Line of Sight)*** dari *transmitter*. Oleh karena bahan *piezoelektrik* reaksi yang *reversible*, elemen keramik akan membangkitkan tegangan listrik pada saat gelombang datang dengan frekuensi yang resonan dan akan mengetarkan bahan *piezoeletrik* tersebut.

2.3.2 Sensor Ultrasonik PING

Modul sensor Ultrasonik ini dapat mengukur jarak antara 3 cm sampai 300 cm. Keluaran dari modul sensor ultrasonik ping ini berupa pulsa yang lebarnya mempresentasikan jarak. Lebar pulsanya yang dihasilkan modul sensor ultrasonik ini bervariasi dari 115 Us sampai 18,5 Ms. Secara prinsip modul sensor ultrasonik ini terdiri dari sebuah *chip* pembangkit sinyal 40 KHz, sebuah *speaker* ultrasonik dan sebuah *mikrofon* ultrasonik. Speker ultrasonik mengubah sinyal 40 KHz

menjadi suara sementara mikropon ultrasonik berfungsi untuk mendeteksi pantulan suaranya. Bentuk sensor ultrasonik diperlihatkan pada gambar 2.4.



Gambar 2.4 Sensor Ultrasonik[4]

Sensor *output* modul sensor ultrasonik dapat langsung dihubungkan dengan mikrokontroler tanpa tambahan komponen apapun. Modul sensor ultrasonik hanya akan mengirimkan suara ultrasonik ketika ada pulsa *trigger* dari mikrokontroler (Pulsa *high* selama 5µS). Suara ultrasonik dengan frekuensi sebesar 40 KHz akan dipancarkan selama 200µS oleh modul sensor ultrasonik ini. Suara ini akan merambat diudara dengan kecepatan 344,424 m/detik (atau 1cm setiap 29.034µS) yang kemudian mengenai objek dan dipantulkan kembali ke modul sensor ultrasonik tersebut, Selama menunggu pantulan sinyal ultrasonik dari bagian *transmitter*, modul sensor ultrasonik ini akan menghasilkan sebuah pulsa. Pulsa ini akan berhenti (*low*) ketika suara pantulan terdeteksi oleh modul sensor ultrasonik. Oleh karena itulah lebar pulsa tersebut dapat mereperentasikan jarak antara modul sensor ultrasonik dengan objek.

2.4 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 (*Integreted 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 serial dengan mikrokontroler

[4]https://components101.com/ultrasonik-sensor-working-pinout-datasheet: 7 Mei 2017

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.

2.4.1 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….

2.4.1.1 *Identifiers*

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

2.4.1.2 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 lingkungi oleh tanda kutip (‘a’)

2.4.1.3 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. Sebaiknya pemanfaatan perinta *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 daipada 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)
```

2.4.2 *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 menujuk 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;
Nilai[2]=3;
```

2.4.3 Fungsi

Sebuah program yang besar dapat dipecah menjadi bebepara 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***.

2.4.3.1 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);
}
```

2.4.3.2 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 menbaca ulang sebuah program yang besar.

2.4.3.3 Parameter dalam sebuah fungsi

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

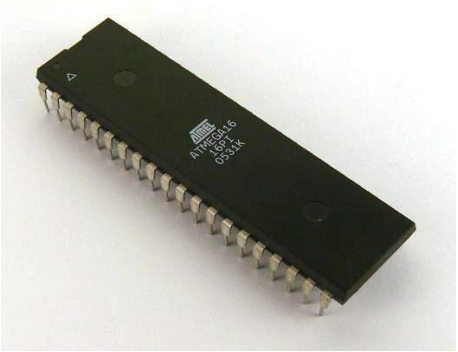
Contoh:

```
//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
```

2.5 ATMEGA 16

Mikrokontroler adalah sebuah sistem komputer lengkap dalam satu serpih (*chip*). Mikrokontroler lebih dari sekedar sebuah mikroprosesor karena sudah terdapat atau berisikan ROM (*Read-Only Memory*), RAM (*Read-Write Memory*), beberapa port masukan maupun keluaran, dan beberapa *peripheral* seperti pencacah/pewaktu, ADC (*Analog to Digital converter*), DAC (*Digital to Analog converter*) dan serial komunikasi.

Salah satu mikrokontroller yang banyak digunakan saat ini yaitu mikrokontroler AVR. AVR adalah mikrokontroller RISC (*Reduce Instuction Set Compute*) 8 bit berdasarkan arsitektur Harvard. Secara umum mikrokontroler AVR dapat dapat dikelompokkan menjadi 3 kelompok, yaitu keluarga AT90Sxx, ATMega dan ATtiny. Pada dasarnya yang membedakan masing-masing kelas adalah memori, *peripheral*, dan fiturnya. Seperti mikroprosessor pada umumnya, secara internal mikrokontroller ATMega16 terdiri atas unit-unit fungsionalnya *Arithmetic and Logical Unit* (ALU), himpunan register kerja, register dan dekoder instruksi, dan pewaktu beserta komponen kendali lainnya. Berbeda dengan mikroprosessor, mikrokontroler menyediakan memori dalam serpih yang sama dengen prosesornya (*in chip*).Bentuk fisik IC mikrokontroller ATmega16 di tunjukkan pada gambar 2.4.



Gambar 2.5 ATMEGA16[5]

2.5.1 Arsitektur ATMEGA16

Mikrokontroller 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*). Mikrokontroller AVR 8 bit yang memiliki kemampuan tinggi, dengan daya rendah. Secara garis besar mikrokontroller ATMega16 terdiri dari :

a. Arsitektur RISC dengan throughput mencapai 16 MIPS pada frekuensi 16Mhz.
b. Memiliki kapasitas Flash memori 16Kbyte, EEPROM 512 Byte, dan SRAM 1Kbyte
c. Saluran I/O 32 buah, yaitu Port A, Port B, Port C, dan Port D.
d. CPU yang terdiri dari 32 buah register.

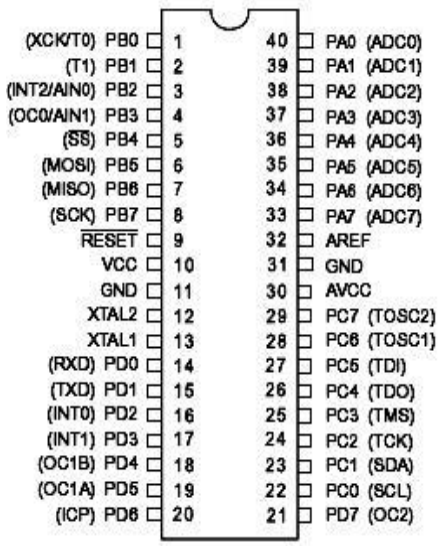
[5]http://www.atmel.com/images/doc2466.pdf: 22 Maret 2017

e. *User* interupsi internal dan eksternal
f. *Port* antarmuka SPI dan Port USART sebagai komunikasi serial
g. Fitur *Peripheral*

- 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

2.5.2 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 berikut,



Gambar 2.6 Port-port Atmega16[6]

[6]http://www.atmel.com/images/doc2466.pdf: 22 Maret 2017

2.5.3 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 porr 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)

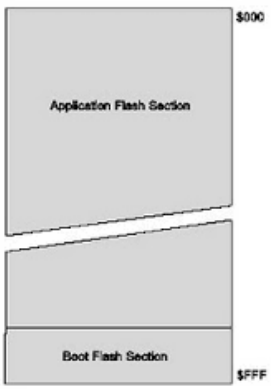
  Port D adalah suatu port I/O 8-bit dua arah dengan resistor *internal pull-up* (yang dipilih untuk beberapa bit). *Port D output buffer* mempunyai karakteristik gerakan simetris dengan keduanya sink tinggi dan kemampuan sumber. Sebagai *input*, pin *port* D yang secara eksternal ditarik rendah akan arus 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.

2.5.4 Struktur Memori ATMega16

2.5.4.1 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 2.6.

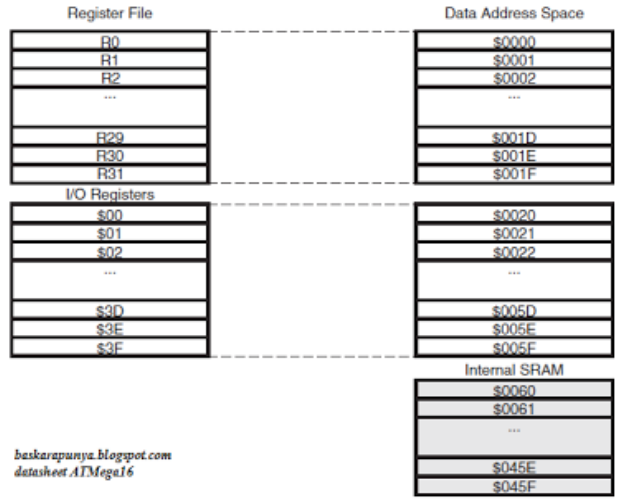


Gambar 2.7 Peta Memori Program AVR ATMEGA16[7]

[7]http://www.atmel.com/images/doc2466.pdf: 22 Maret 2017

2.5.4.2 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 2.7.



Gambar 2.8 Peta Memori Data AVR ATMEGA16[8]

2.5.4.3 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=.

2.6 TRAFO STEP DOWN

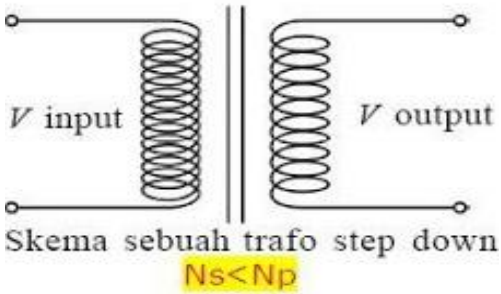
Trafo *step down* merupakan trafo yang berfungsi untuk menurunkan tegangan listrik pada rangkaian elektronika, trafo *step down* sering dipakai pada catu daya baik itu catu daya yang sudah terlegulasi ataupun yang belum terlegulasi.

[8]http://www.atmel.com/images/doc2466.pdf: 22 Maret 2017

Kegunaan dari trafo *stepo down* sebagai penganti baterai. Tanpa *power supply* yang menggunakan trafo *step down* makan sistem catu daya baterai pada rangkaian elektronika sangat tidak efisien terutama dari segi biaya.

2.6.1 Fungsi Trafo Step Down

Fungsi dasar dari sebuah trafo *step down* tentu saja untuk menurunkan tegangan listrik untuk mengasilkan tegangan yang lebih kecil sesuai dengan kebutuhan proyek elektronika. Meskipun fungsi dasar dari trafo *step down* hanya satu, namun kegunaannya bisa dipakai hampir semua perangkat eletronika seperti *amplifier*, radio, *gadget*, *booster* antena televisi dan lain-lain.



Gambar 2.9 Kumparan trafo step down[9]

Trafo *step down* memiliki jumlah kumparan sekunder yang lebih sedikit dibandingkan dengan jumlah kumparan primernya. Sesuai dengan kontruksi dasar *transformator*. Trafo *step down* terdiri dari lilitan kumparan kawat email dengan diameter tertentu yang dilapisi oleh kawat email agar tumpukan lilitan kumparan tidak terhubung langsung satu sama lain yang dapat mengakibatkan terjadinya hubungan singkat, baik itu pada kumparan primernya maupun pada kumparan sekundernya.

Untuk ukuran trafo *step down* bervariasi tergantung arus *output* yang dapat dikeluarkan oleh trafo tersebut. Semakin besar arus yang dikeluarkan oleh sebuah trafo *step down*, maka dimensi trafo akan semakin besar pula, karena selain diameter lilitan sekunder yang lebih besar yang berbanding lurus dengan kemampuan arus yang dikeluarkan, ukuran inti besi juga otomatis akan semakin besar untuk dapat menompang gaya gerak listrik induksi *electromagnet* dari kumparan primer ke kumpara sekunder. Ketika sebuah trafo *step down* dialiri oleh tegangan listrik AC

[9]http://www.berpendidikan.com/2015/10/macam-macam-dan-ciri-ciri-transformator-trafo-step-up-step-down.html: 6 Mei 2017

220V (jaringan PLN) pada bagian primer, maka kumparan primer yang telah dikelilingi oleh inti besi akan timbul *electromagnet*.

Gaya *electromagnet* ini akan muncul seiring dengan perubahan garis gaya magnet yang ditimbulkan oleh arus AC. Karena garis gaya magnet ini timbul pada kumparan sekunder trafo. Jumlah gaya gerak listrik yang akan timbul pada kumparan sekunder tergantung jumlah lilitan dan diameternya. Pada perancangan rangkaian elektronika sebuah trafo *step down* harus di desain sesuai dengan kebutuhan beban, ketika arus yang dibutuhkan oleh beban lebih besar dari arus *output* yang dikeluarkan oleh trafo *step down*, maka hal ini akan berbahaya dari komponen trafo *step down* itu sendiri, selain dapat menimbulkan panas yang berlebihan pada kumparan dan inti besinya, hal tersebut dapat menyebabkan kerusakan trafo.

Oleh karena itulah, kebayangkan sebuah adaptor AC ke DC yang memiliki kualitas baik biasanya dilengkapi dengan rangkaian regulator tegangan dan proteksi arus terhadap beban lebih, selain *output*nya bebas dari tegangan *ripple*, juga terlindungi dari arus beban yang lebih, karena ketika hal itu terjadi, secara otomatis rangkaian proteksi akan bekerja dan sumber catu terhadap beban akan terputus.