

BAB II

LANDASAN TEORI

Landasan teori ini meliputi tentang pengertian Truck, pengertian sensor Ultrasound, Pengertian Mikrokontroler ATmega 16, Pengertian pemrogram Bahasa C, Pengertian Alarm/Buser, Pengertian LCD. Untuk memudahkan dalam perancangan perangkat keras dan perangkat lunak.

2.1. Kendaraan Truck

Truk adalah sebuah kendaraan bermotor untuk mengangkut barang, disebut juga sebagai *mobil barang*. Dalam bentuk yang kecil mobil barang disebut sebagai *pick-up*, sedangkan bentuk lebih besar dengan 3 sumbu, 1 di depan, dan tandem di belakang disebut sebagai truk tronton, sedang yang digunakan untuk angkutan peti kemas dalam bentuk tempelan disebut sebagai truk trailer. Juga ada jenis truk tangki yang berguna untuk mengangkut cairan seperti BBM dan lainnya.

Daya angkut truk tergantung kepada beberapa variabel, diantaranya jumlah ban, jumlah sumbu/ konfigurasi sumbu, muatan sumbu, kekuatan ban, daya dukung jalan. Pada daftar berikut ditunjukkan hubungan antara daya angkut dengan konfigurasi sumbu truk.

Tabel 2.1. Tipe data Truck¹

Konfigurasi Sumbu	Jumlah Sumbu	Jenis	JBI Kelas II	JBI Kelas III	Jumlah Ban
1 – 1	2	Truk Engkel Tunggal	12 ton	12 ton	4
1 – 2	2	Truk Engkel Ganda	16 ton	14 ton	6
1.1 – 2	3	Truk Trintin	18 ton	16 ton	8
1 - 2.2	3	Truk Tronton	22 ton	20 ton	10
1.1 - 2.2	4	Truk Trinton	30 ton	26 ton	12
1 - 2 - 2.2	4	Truk Trailer Engkel	34 ton	28 ton	14
1 - 2 - 2.2.2 1 - 2.2 - 2.2	5	Truk Trailer Engkel Truk Trailer Tronton	40 ton	32 ton	18
1 - 2.2 - 2.2.2	6	Truk Trailer Tronton	43 ton	40 ton	22

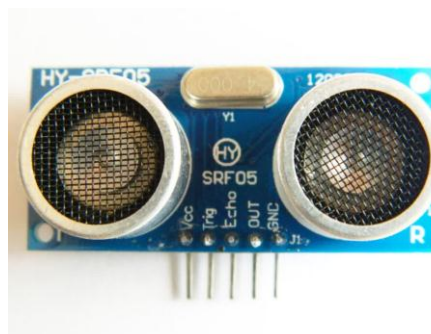
¹ Sumber = Wikipedia bahasa Indonesia, ensiklopedia bebas : diakses april 2017
Sember Gambar tribunnews



Gambar 2.1 Kendaraan Truck¹

2.2. HC-SR05 (Sensor Ultrasonik)

HC-SR05 adalah Sensor Ultrasonik yang memiliki dua elemen, yaitu elemen Pendeteksi gelombang ultrasonik, dan juga sekaligus elemen Pembangkit gelombang ultrasonik. Sensor Ultrasonik adalah sensor yang dapat mendeteksi gelombang ultrasonik, yaitu gelombang suara yang memiliki frekuensi ultrasonik atau frekuensi di atas kisaran frekuensi pendengaran manusia. Sensor ultrasonic tipe SR05 ini secara prinsip sama saja dengan sensor ultrasonic (PING)). Memancarkan gelombang ultrasonic dengan frekuensi 40 KHz kemudian menunggu sampai pantulan gelombangnya diterima kembali. Tanda kalau gelombang sudah diterima atau belum berupa sinyal HIGH – LOW.



Gambar 2.2 Tampilan Sensor HC-SR05².

Fungsi Pin-pin HC-SR05

1. **VCC** = *5V Power Supply*. Pin sumber tegangan positif sensor.
2. **Trig** = *Trigger/Penyulut*. Pin ini yang digunakan untuk membangkitkan sinyal ultrasonik.
3. **Echo** = *Receive/Indikator*. Pin ini yang digunakan untuk mendeteksi sinyal pantulan ultrasonik.
4. **GND** = *Ground/0V Power Supply*. Pin sumber tegangan negatif sensor.

2.2.1. Karakteristik HC-SR05 (Sensor Ultrasonik)

Karakteristik HC-SR05 (Sensor Ultrasonik) sebagai berikut:

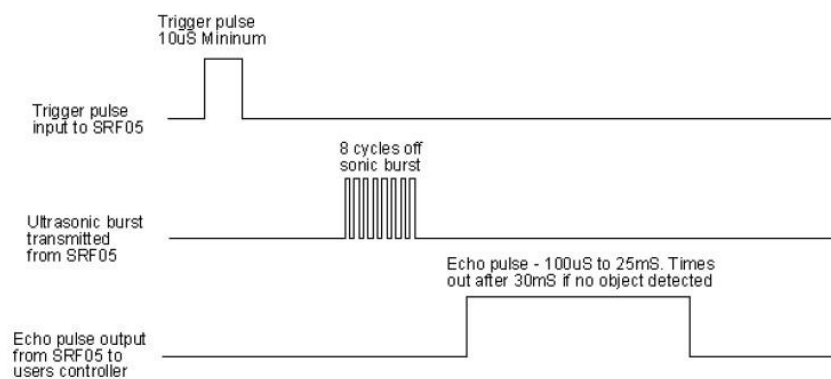
1. Tegangan sumber operasi tunggal 5.0 V
2. Konsumsi arus 15 mA
3. Frekuensi operasi 40 KHz
4. Minimum pendeteksi jarak 0.02 m (2 cm)
5. Maksimum pendeteksian jarak 4 m
6. Sudut pantul gelombang pengukuran 15 derajat
7. Minimum waktu penyulutan 10 mikrodetik dengan pulsa berlevel TTL
8. Pulsa deteksi berlevel TTL dengan durasi yang bersesuaian dengan jarak deteksi
9. Dimensi 45 x 20 x 15 mm

2.2.2. Diagram Waktu HC-SR05 (Sensor Ultrasonik)

HC-SR05 memerlukan sinyal logika '1' pada pin **Trig** dengan durasi waktu 10 mikrodetik (us) untuk mengaktifkan rentetan (*burst*) 8x40KHz gelombang ultrasonik² pada elemen Pembangkitnya. Selanjutnya pin **Echo** akan berlogika '1' setelah rentetan 8x40 KHz tadi, dan otomatis akan berlogika '0' saat gelombang pantulan diterima oleh elemen Pendeteksi gelombang ultrasonik.

² Sumber : <https://depokinstruments.com/tag/belajar-sensor-ultrasonic/> : diakses april 2017
 Sumber gambar : depokinstruments.com/tag/belajar-sensor-ultrasonic : diakses april 2017

Kemudian ketika bunyi yang dipantulkan kembali ke sensor SRF05, bunyi tadi akan diterima dan membuat keluaran sinyal high pada pin echo yang kemudian menjadi inputan pada mikrokontroler. SRF05 akan memberikan pulsa 100 μ s - 25ms pada outputnya tergantung pada informasi jarak pantulan objek yang diterima. Lamanya sinyal high dari echo inilah yang digunakan untuk menghitung jarak antara sensor SRF05 dengan benda yang memantulkan bunyi yang berada di depan sensor ini. Untuk menghitung lamanya sinyal high yang diterima mikrokontroler dari pin echo, maka digunakan fasilitas timer yang ada pada masing-masing mikrokontroler. Ketika ada perubahan dari low ke high dari pin echo maka akan mengaktifkan timer, dan ketika ada perubahan dari high ke low dari pin echo maka akan mematikan timer.



Gambar 2.3 Diagram Waktu HC-SR05³

2.3. Mikrokontroler ATMEGA16

Merupakan seri mikrokontroler *Complementary Metal Oxide Semiconductor* (CMOS) 8-bit buatan Atmel berbasis arsitektur RISC (*Reduced Instruction Set Computer*). Hampir semua instruksi pada program dieksekusi dalam satu siklus *clock*. AVR mempunyai 32 register *general-purpose*, *timer/counter* fleksibel dengan mode *compare*, interupsi *internal* dan *eksternal*, serial UART, *programmable Watchdog Timer*, *power saving mode*, ADC dan PWM. AVR pun mempunyai *In-System Programmable (ISP) Flash on-chip* yang mengijinkan memori program untuk diprogram ulang (*read/write*)

dengan koneksi secara serial yang disebut *Serial Peripheral Interface* (SPI). AVR memiliki keunggulan dibandingkan dengan mikrokontroler lain, keunggulan mikrokontroler AVR yaitu memiliki kecepatan dalam mengeksekusi program yang lebih cepat, karena sebagian besar instruksi dieksekusi dalam 1 siklus *clock* (lebih cepat dibandingkan mikrokontroler keluarga MCS 51 yang memiliki arsitektur *Complex Instruction Set Compute*). ATMEGA16 mempunyai *throughput* mendekati 1 *Millions Instruction Per Second* (MIPS) per MHz, sehingga membuat konsumsi daya menjadi rendah terhadap kecepatan proses eksekusi perintah.

2.3.1. Beberapa keistimewaan dari AVR ATMEGA16:

1. Mikrokontroler³ AVR 8 bit yang memiliki kemampuan tinggi dengan konsumsi daya rendah
2. Arsitektur RISC dengan *throughput* mencapai 16 MIPS pada frekuensi 16MHz
3. Memiliki kapasitas *Flash* memori 16 Kbyte, EEPROM 512 Byte dan SRAM 1 Kbyte
4. Saluran I/O sebanyak 32 buah, yaitu *Port A*, *Port B*, *Port C* dan *Port D*
5. CPU yang terdiri dari 32 buah *register*
6. Unit interupsi dan eksternal
7. *Port* USART untuk komunikasi serial
8. Fitur *peripheral*
 - Tiga buah *Timer/Counter* dengan kemampuan perbandingan (*compare*)
 - Dua buah *Timer/Counter* 8 bit dengan *Prescaler* terpisah dan *Mode Compare*
 - Satu buah *Timer/Counter* 16 bit dengan *Prescaler* terpisah, *Mode*

³ Sumber : Lapantech.com : diakses april 2017
 elektronika-dasar.web.id/mikrokontroler/at-mega16 : diakses april 2017
 Andrianto, Heri. 2008. Pemrograman Mikrokontroler AVR Atmega16: Mikrokontroler . Informatika

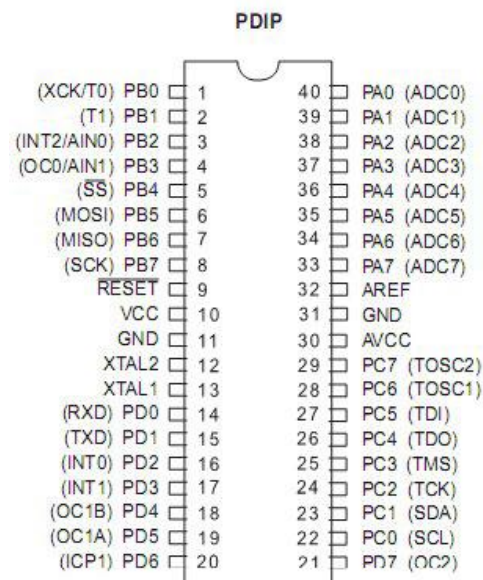
Sumber gambar : datasheet mikrokontroler : diakses april 2017

Compare dan Mode Capture

- *Real Time Counter* dengan *Oscillator* tersendiri
- Empat kanal PWM
- 8 kanal ADC
- 8 *Single-ended Channel* dengan keluaran hasil konversi 8 dan 10 resolusi (register ADCH dan ADCL)
- 7 *Diferrential Channel* hanya pada kemasan *Thin Quad Flat Pack* (TQFP)
- 2 *Differential Channel* dengan *Programmable Gain*
- Antarmuka *Serial Peripheral Interface (SPI) Bus*
- *Watchdog Timer* dengan *Oscillator Internal*
- *On-chip Analog Comparator*

9. *Non-volatile program memory*

2.3.2. Konfigurasi Pin AVR ATMEGA16



Gambar 2.4 Konfigurasi ATMEGA16⁴

Konfigurasi pin mikrokontroler Atmega16 dengan kemasan 40-pin dapat dilihat pada gambar diatas. Dari gambar tersebut dapat terlihat ATMEGA16

memiliki 8 pin untuk masing-masing Gerbang A (Port A), Gerbang B (Port B), Gerbang C (Port C), dan Gerbang D (Port D).

2.3.3. Deskripsi Mikrokontroler ATmega 16

1) VCC

Merupakan *supply* tegangan digital. Untuk ATmega 16 besar tegangan input yang digunakan adalah 4,5v – 5,5v

2) GND

Merupakan *ground* untuk semua komponen yang membutuhkan *Grounding*

3) Port A

Yaitu (PA7..PA0) 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.

4) Port B

Adalah suatu port I/O 8-bit dua arah dengan pin fungsi khusus yaitu *Timer/Counter*, komparator analog dan SPI

5) Port C

Adalah suatu Port I/O 8-bit dua arah dengan dengan pin fungsi khusus yaitu TWI, komparator analog dan *Timer Oscillator*.

6) Port D

Adalah suatu Port I/O 8-bit dua arah dengan dengan pin fungsi khusus yaitu komparator analog, interupsi eksternal dan komunikasi serial.

7) AVCC

Adalah pin penyedia tegangan untuk Port A dan Konverter A/D. Pin ini berfungsi sebagai *supply* tegangan untuk ADC. Untuk *pin* ini harus dihubungkan secara terpisah dengan VCC karena *pin* ini digunakan untuk analog saja. Bahkan jika ADC pada AVR tidak digunakan tetap saja disarankan untuk menghubungkannya secara terpisah dengan VCC. Jika ADC digunakan, maka AVCC harus dihubungkan ke VCC melalui *low pass filter*.

8) AREF

Merupakan pin masukan tegangan referensi analog untuk konverter

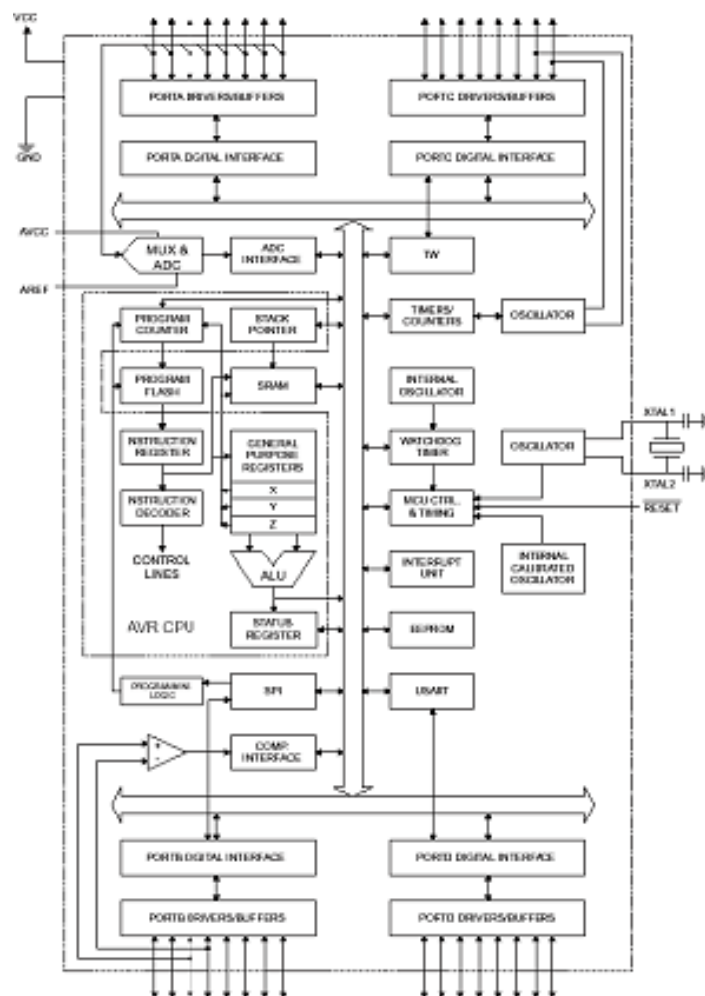
A/D

9) RESET

Pin ini berfungsi untuk me-reset mikrokontroler ke kondisi semula

10) XTAL1 dan XTAL2

Merupakan *Input Oscillator* berfungsi sebagai pin masukan *clock* eksternal. Suatu mikrokontroler membutuhkan sumber detak (*clock*) agar dapat mengeksekusi instruksi yang ada di memori. Semakin tinggi nilai kristalnya, maka semakin cepat pula mikrokontroler tersebut dalam mengeksekusi program.



Gambar 2.5 Blok Diagram ATmega 16⁵

2.3.4. Analog To Digital Converter

AVR ATmega16 merupakan tipe AVR yang telah dilengkapi dengan 8 saluran ADC internal dengan resolusi 10 bit. Dalam mode operasinya, ADC dapat dikonfigurasi, baik single ended input maupun differential input. Selain itu, ADC ATmega16 memiliki konfigurasi pewaktuan, tegangan referensi, mode operasi, dan kemampuan filter derau (noise) yang fleksibel sehingga dapat dengan mudah disesuaikan dengan kebutuhan dari ADC itu sendiri. ADC pada ATmega16 memiliki fitur-fitur antara lain :

- a. Resolusi mencapai 10-bit
- b. Akurasi mencapai ± 2 LSB
- c. Waktu konversi 13-260 μ s
- d. 8 saluran ADC dapat digunakan secara bergantian
- e. Jangkauan tegangan input ADC bernilai dari 0 hingga VCC
- f. Disediakan 2,56V tegangan referensi internal ADC
- g. Mode konversi kontinyu atau mode konversi tunggal
- h. Interupsi ADC complete
- i. Sleep Mode Noise canceler

Proses inisialisasi ADC meliputi proses penentuan *clock*, tegangan referensi, format data keluaran dan modus pembacaan. Register-register yang perlu diatur adalah sebagai berikut:

- ADC Control and Status Register A – ADCSRA

Bit	7	6	5	4	3	2	1	0	
	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	ADCSRA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Gambar 2.6 ADC Control and Status Register A – ADCSRA⁶

Keterangan :

ADEN : 1 = adc enable, 0 = adc disable

ADCS : 1 = mulai konversi, 0 = konversi belum terjadi

ADATE : 1 = auto trigger diaktifkan, trigger berasal dari sinyal yang dipilih

(set pada trigger SFIOR bit ADTS). ADC akan start konversi pada edge positif sinyal trigger.

ADIF : diset ke 1, jika konversi ADC selesai dan data register ter-update. Namun ADC Conversion Complete Interrupt dieksekusi jika bit ADIE dan bit-I dalam register SREG diset.

ADIE : diset 1, jika bit-I dalam register SREG di-set.

ADPS[0..2] : Bit pengatur clock ADC, faktor pembagi 0 ... 7 = 2, 4, 8, 16, 32, 64, 128.

Tabel 2.2 Konfigurasi Clock ADC²

ADPS2	ADPS1	ADPS0	Division Vector
0	0	0	2
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

• ADC Multiplexer-ADMUX

Bit	7	6	5	4	3	2	1	0	ADMUX
	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Gambar 2.7 ADC Multiplexer⁷

REFS 0, 1 : Pemilihan tegangan referensi ADC

00 : Vref = Aref

01 : vref = AVCC dengan eksternal capasitor pada AREF

10 : $v_{ref} = \text{internal } 2.56 \text{ volt}$ dengan eksternal kapasitor pada AREF

ADLAR : Untuk setting format data hasil konversi ADC, default = 0

- Special Function IO Register-SFIOR

SFIOR merupakan register 8 bit pengatur sumber picu konversi ADC, apakah dari picu eksternal atau dari picu internal,

ADTS[0...2] : Pemilihan trigger (pengatur picu) untuk konversi ADC, bit-bit ini akan berfungsi jika bit ADATE pada register ADCSRA bernilai 1. Konfigurasi bit ADTS[0...2] dapat dilihat pada Tabel

Tabel 2.3 Pemilihan sumber picu ADC³

ADTS2	ADTS1	ADTS0	Trigger Source
0	0	0	Free Running Mode
0	0	1	Analog Comparator
0	1	0	External Interrupt Request 0
0	1	1	Timer/Counter0 Compare Match
1	0	0	Timer/Counter 0 Overflow
1	0	1	Timer/Counter Compare Match B
1	1	0	Timer/Counter1 Overflow
1	1	1	Timer/Counter1 Capture Event

ADHSM : 1. ADC *high speed mode enabled*. Untuk operasi ADC, bit ACME, PUD, PSR2 dan PSR10 tidak diaktifkan.

2.4. Pemrograman Bahasa C

Bahasa C adalah sekumpulan kode yang ditaruh dalam sebuah blok dan dibuat untuk menjalankan tugas khusus. Salah satu tujuan dari penggunaan fungsi dalam pemrograman seperti pada Bahasa C adalah untuk membuat

⁴ Andrianto, Heri. 2008. Pemrograman Mikrokontroler AVR Atmega16: Mikrokontroler. Informatika

program lebih terstruktur dan efisien sehingga program tersebut mudah dipahami atau dibaca alur programnya.

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) dan angka (0 ... 9) dan karakter (_). *Identifiers* bersifat *Case sensitive*. *Indentifier* 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 Tipe Data

Tabel 2.4. Tipe data⁴

Tipe	Ukuran (<i>bit</i>)	Rentang (<i>Range</i>)
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 $\pm 3.402e38$

2.4.1.4 Operator

Tabel 2.5. Daftar Operator Kondisi⁵

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

⁵ Ibid., 26

Tabel 2.6. Daftar Operator Aritmatika⁶

Operator Aritmatika	Keterangan
+	Penjumlahan
-	Pengurangan
*	Perkalian
/	Pembagian

Tabel 2.7. Daftar Operator Logika⁷

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

Tabel 2.8. Daftar Operator *Bitwise*⁸

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>

⁶ Loc.Cit.

⁷ Loc.Cit.

⁸ Loc.Cit.

Tabel 2.9. Daftar Operator *Assignment*⁹

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>

2.4.1.5 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:

$$if(\langle expression \rangle) \langle statement \rangle;$$

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

$$if(\langle expression \rangle) \langle statement \rangle;$$

$$else \langle statement2 \rangle;$$

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

⁹ 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-keropotan 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)10

```

2.4.1.6 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. Alarm/Buzer

Buzzer adalah suatu alat yang dapat mengubah sinyal listrik menjadi sinyal suara. Pada umumnya *buzzer* digunakan untuk alarm, karena penggunaannya cukup mudah yaitu dengan memberikan tegangan input maka *buzzer* akan mengeluarkan bunyi. Frekuensi suara yang di keluarkan oleh *buzzer* yaitu antara 1-5 KHz.

¹⁰ Kurt Goszyk, *Buzzer*,
<http://www.designnotes.com> : diakses april 2017



Gambar 2.8 Buzzer⁸

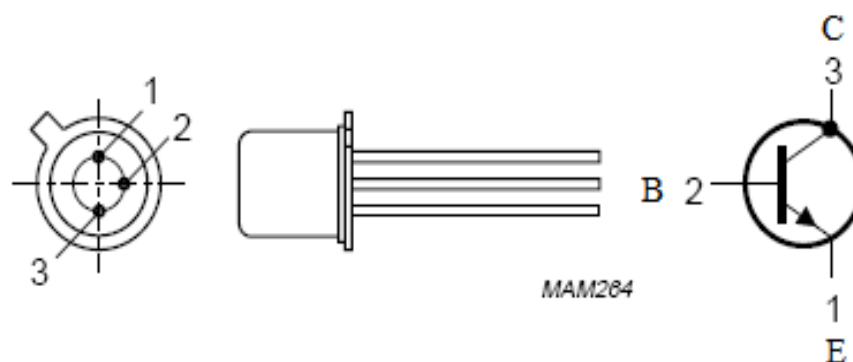
2.6. Transistor NPN 2N3904

Transistor NPN adalah transistor positif dimana transistor dapat bekerja mengalirkan arus listrik apabila basis dialiri tegangan positif. Transistor NPN 2N2222 adalah komponen elektronika yang terbuat dari bahan semikonduktor dan mempunyai tiga elektroda (triode) yaitu dasar (basis), pengumpul (kolektor) dan pemancar (emitor). Dengan ketiga elektroda (terminal) tersebut, tegangan atau arus yang dipasang di satu terminalnya mengatur arus yang lebih besar yang melalui 2 terminal lainnya. Transistor ini berfungsi sebagai sirkuit pemutus dan penyambung (*switching*). Transistor sebagai saklar elektronik, yaitu dengan mengatur bias dari sebuah transistor sampai transistor jenuh (saturasi) maka didapat hubungan singkat antar kaki kolektor dan emitor, dengan memanfaatkan kejadian ini maka transistor bisa digunakan sebagai saklar. Pada rangkaian elektronik, sinyal masukannya berlogika 1 atau 0 ini selalu dipakai pada basis transistor, yang mana kolektor dan emitor sebagai penghubung untuk pemutus (*short*) atau sebagai pembuka rangkaian. Cara kerja transistor NPN ini yaitu saat basis transistor NPN mendapat logika 1 maka kolektor dan emitor akan terhubung singkat, sehingga transistor NPN saturasi. Saat basis transistor NPN mendapat logika 0, maka transistor NPN akan *cut off*.

Fungsi transistor sangatlah besar dan mempunyai peranan penting untuk memperoleh kinerja yang baik bagi sebuah rangkaian elektronika. Dalam dunia elektronika, **fungsi transistor** ini adalah sebagai berikut:

- Sebagai sebuah penguat (amplifier).
- Sirkuit pemutus dan penyambung (*switching*).
- Stabilisasi tegangan (stabilisator).

- Sebagai perata arus.
- Menahan sebagian arus.
- Memperkuat arus¹¹.
- Membangkitkan frekuensi rendah maupun tinggi.
- Modulasi sinyal dan berbagai fungsi lainnya.



Gambar 2.9 Tansistor NPN 2N3904⁹

2.7. LED (Light Emiting Diode)

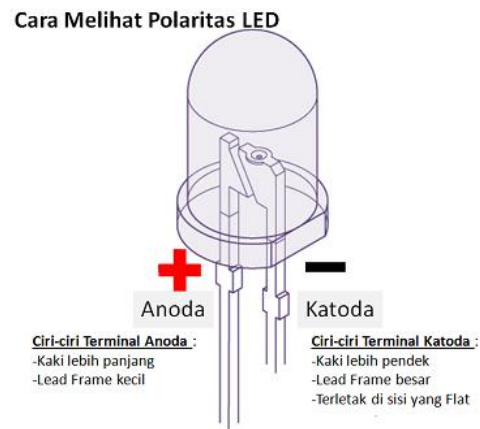
Light Emitting Diode atau sering disingkat dengan LED adalah komponen elektronika yang dapat memancarkan cahaya monokromatik ketika diberikan tegangan maju. LED merupakan keluarga Dioda yang terbuat dari bahan semikonduktor. Warna-warna Cahaya yang dipancarkan oleh LED tergantung pada jenis bahan semikonduktor yang dipergunakannya. LED juga dapat memancarkan sinar inframerah yang tidak tampak oleh mata seperti yang sering kita jumpai pada Remote Control TV ataupun Remote Control perangkat elektronik lainnya.

Cara Kerja LED (Light Emitting Diode) pun hampir sama dengan Dioda yang memiliki dua kutub yaitu kutub Positif (P) dan Kutub Negatif (N). LED hanya akan memancarkan cahaya apabila dialiri tegangan maju (bias forward) dari Anoda menuju ke Katoda. Untuk mengetahui polaritas terminal Anoda (+) dan

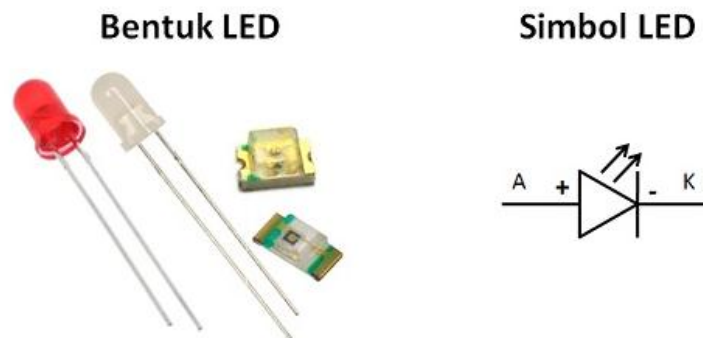
¹¹ Sumber = <http://dasarelektronika.com/pengertian-dan-fungsi-transistor> : diakses april 2017
Arifianto,Deni.2011.Kumpulan Rangkaian Elektronika Sederhana: Elektro.

Sumber gambar transistor : dasarelektronika.com/pengertian-dan-fungsi-transistor

Katoda (-) pada LED. Kita dapat melihatnya secara fisik berdasarkan gambar diatas. Ciri-ciri Terminal Anoda pada LED adalah kaki yang lebih panjang dan juga Lead Frame yang lebih kecil. Sedangkan ciri-ciri Terminal Katoda adalah Kaki yang lebih pendek dengan Lead Frame yang besar serta terletak di sisi yang Flat.¹²



Gambar 2.10 Polaritas pada LED¹⁰



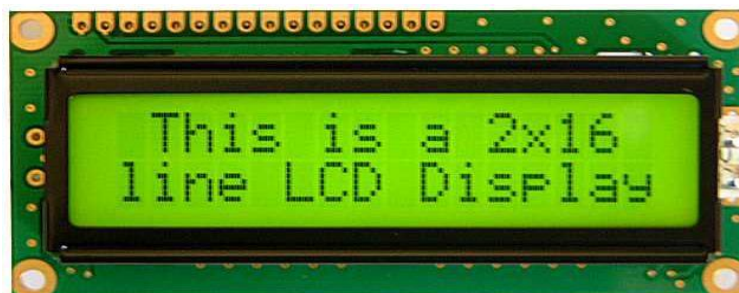
Gambar 2.11 Bentuk dan symbol LED¹¹

¹² Sumber = <http://teknikelektronika.com/pengertian-led-light-emitting-diode-cara-kerja> : diakses april 2017

Arifianto,Deni.2011.Kumpulan Rangkaian Elektronika Sederhana: Elektro.

2.8. LCD (*liquid crystal display*)

LCD (*liquid crystal display*) adalah suatu alat penampil dari bahan cairan kristal yang pengoperasiannya menggunakan sistem dot matriks. Fungsi LCD pada rancangan ini digunakan untuk menampilkan hasil dari proses perhitungan mikrokontroler. Pada perancangan ini, LCD yang digunakan adalah LCD 16x2 yang memiliki *backlamp*. LCD tersebut dihubungkan dengan Port C pada mikrokontroler ATMEGA16. Kontroler dan penggerak LCD dapat menampilkan karakter alfanumerik, karakter Jepang (katakana), dan beberapa simbol. Kontroler ini mengandung ROM pembentuk karakter (*character generator ROM*) berukuran 9920 bit yang menghasilkan 240 karakter yang terdiri atas 208 karakter dengan resolusi 5x8 titik (*dot, pixel*) dan 32 karakter dengan resolusi 5x10 titik. Kontroler ini juga mengandung RAM pembentuk karakter yang dapat menyimpan 64 karakter 8 bit.¹³



Gambar 2.12 LCD 2x16¹²

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.

¹³ Sumber = elektronika-dasar.web.id/lcd-liquid-cristal-display/ : diakses april 2017
Sumber gambar = elektronika-dasar.web.id/lcd-liquid-cristal-display

Tabel 2.10 Fungsi Pin LCD Karakter 16x2¹⁰

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 (-)

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 22 – pin 29).
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 (pin22 - pin 29).
 3. Set supaya pin E = 0, kemudian berikan delay sesaat, dan set kembali pin E = 1.
-