

BAB II

TINJAUAN PUSTAKA

Menurut Ogata (1996) dalam bukunya yang berjudul “Teknik Kontrol Automatik” sistem itu merupakan kombinasi dari beberapa komponen yang bekerja bersama-sama dan melakukan suatu sasaran tertentu. Sistem tidak dibatasi hanya untuk sistem fisik saja. Konsep sistem dapat digunakan pada gejala yang abstrak dan dinamis.

2.1 Definisi Umum

2.1.1 Suhu

Suhu adalah besaran derajat panas dingin suatu benda dan alat yang digunakan untuk mengukur suhu adalah termometer. Dalam kehidupan sehari-hari masyarakat mengukur suhu cenderung menggunakan indra.

2.1.2 Antisipasi

Menurut Brain Quote dalam *Definition of Anticipation* mengemukakan bahwa antisipasi adalah tindakan mengambil, menempatkan, atau mempertimbangkan sesuatu terlebih dahulu, atau sebelum waktu yang tepat untuk alam. (diakses 13-06-2014)

Atau dalam pengertian umum bahwa pengertian antisipasi adalah pemecahan masalah dalam suatu hal yang akan terjadi.

2.1.3 Hujan

Hujan adalah jatuhnya hydrometeor yang berupa partikel-partikel air dengan diameter 0.5 mm atau lebih. Jika jatuhnya sampai ke tanah maka disebut hujan, akan tetapi apabila jatuhnya tidak dapat mencapai tanah karena menguap lagi maka jatuhnya tersebut disebut Virga. Hujan juga dapat didefinisikan dengan uap yang mengondensasi dan jatuh ke tanah dalam rangkaian proses hidrologi. (Sulthony, hal:4)

Hujan merupakan salah satu bentuk presipitasi uap air yang berasal dari awan yang terdapat di atmosfer.

2.2 Hardware

2.2.1 Mikrokontroler ATMEGA16

Mikrokontroler adalah suatu alat elektronika digital yang mempunyai masukan dan keluaran serta kendali program yang bisa ditulis dan dihapus dengan cara khusus. Mikrokontroler adalah prosesor yang dikhususkan untuk instrumentasi dan kendali. (Sumardi, hal:1)

Mikrokontroler adalah IC yang dapat diprogram berulang kali, baik ditulis atau dihapus (Agus bejo, 2007). Biasanya digunakan untuk pengontrolan otomatis dan manual pada perangkat elektronika.

Beberapa tahun terakhir, mikrokontroler sangat banyak digunakan terutama dalam pengontrolan robot. Seiring perkembangan elektronika mikrokontroler dibuat semakin kompak dengan bahasa pemrograman yang juga ikut berubah. Salah satunya adalah mikrokontroler AVR (*Alf and vegard's risc processor*) Armega16 yang menggunakan teknologi RISC (Reduce instruction Set Computing) dimana program berjalan lebih cepat karena hanya membutuhkan satu siklus clock untuk mengeksekusi satu instruksi Program.

AVR merupakan seri mikrokontroler CMOS 8-bit buatan Atmel, berbasis arsitektur RISC (Reduced Instruction Set Computer). Hampir semua instruksi dieksekusi dalam satu siklus clock. AVR mempunyai 32 register general-purpose, timer/counter fleksibel dengan mode compare, interrupt internal dan eksternal, serial UART, programmable Watchdog Timer, dan mode power saving, ADC dan PWM internal. AVR juga mempunyai In-System Programmable Flash on-chip yang memungkinkan memori program untuk diprogram ulang dalam sistem menggunakan hubungan serial SPI. ATMega16. ATMega16 mempunyai throughput mendekati 1 MIPS per MHz membuat disainer sistem untuk mengoptimasi konsumsi daya versus kecepatan proses.

1. Beberapa keistimewaan dari AVR ATMega16 antara lain:

1. Advanced RISC Architecture
 - a. 130 Powerful Instructions – Most Single Clock Cycle Execution
 - b. 32 x 8 General Purpose Fully Static Operation
 - c. Up to 16 MIPS Throughput at 16 MHz

- d. On-chip 2-cycle Multiplier
2. Nonvolatile Program and Data Memories
 - a. 8K Bytes of In-System Self-Programmable Flash
 - b. Optional Boot Code Section with Independent Lock Bits
 - c. 512 Bytes EEPROM *f* 512 Bytes Internal SRAM
 - d. Programming Lock for Software Security
 2. Peripheral Features
 - a. Two 8-bit Timer/Counters with Separate Prescalers and Compare Mode
 - b. Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
 - c. One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
 - d. Real Time Counter with Separate Oscillator
 - e. Four PWM Channels
 - f. 8-channel, 10-bit ADC
 - g. Byte-oriented Two-wire Serial Interface
 - h. Programmable Serial USART
 3. Special Microcontroller Features
 - a. Power-on Reset and Programmable Brown-out Detection
 - b. Internal Calibrated RC Oscillator
 - c. External and Internal Interrupt Sources
 - d. Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby
 4. I/O and Package
 - a. 32 Programmable I/O Lines
 - b. 40-pin PDIP, 44-lead TQFP, 44-lead PLCC, and 44-pad MLF 6. Operating Voltages
 - c. 2.7 - 5.5V for Atmega16L
 - d. 4.5 - 5.5V for Atmega16

2.2.2 Konstruksi Atmega16

Mikrokontroler Atmega16 memiliki 3 jenis memori, yaitu memori program, memori data dan memori EEPROM. Ketiganya memiliki ruang sendiri dan terpisah.

a. Memori program

Atmega16 memiliki kapasitas memori program sebesar 16 Kbyte yang terpetakan dari alamat 0000h – 0FFFh dimana masing-masing alamat memiliki lebar data 16 bit. Memori program ini terbagi menjadi 2 bagian yaitu bagian program *boot* dan bagian program aplikasi.

b. Memori data

Atmega16 memiliki kapasitas memori data sebesar 2 Kbyte yang terbagi menjadi 3 bagian yaitu register serba guna, register I/O dan SRAM. Atmega16 memiliki 16 byte register serba guna, 64 byte register I/O yang dapat diakses sebagai bagian dari memori RAM (menggunakan instruksi LD atau ST) atau dapat juga diakses sebagai I/O (menggunakan instruksi IN atau OUT), dan 2 Kbyte digunakan untuk memori data SRAM.

c. Memori EEPROM

Atmega16 memiliki memori EEPROM sebesar 1024 byte yang terpisah dari memori program maupun memori data. Memori EEPROM ini hanya dapat diakses dengan menggunakan register-register I/O yaitu register EEPROM *Address*, register EEPROM *Data*, dan register EEPROM *Control*. Untuk mengakses memori EEPROM ini diperlakukan seperti mengakses data eksternal, sehingga waktu eksekusinya relatif lebih lama bila dibandingkan dengan mengakses data dari SRAM.

Atmega16 merupakan tipe AVR yang telah dilengkapi dengan 8 saluran ADC internal dengan fidelitas 10 bit. Dalam mode operasinya, ADC Atmega16 dapat dikonfigurasi, baik secara *single ended input* maupun *differential input*. Selain itu, ADC Atmega16 memiliki konfigurasi pewaktuan, tegangan referensi, mode operasi, dan kemampuan filter derau yang amat fleksibel, sehingga dengan mudah disesuaikan dengan kebutuhan ADC itu sendiri.

Atmega16 memiliki 3 modul timer yang terdiri dari 2 buah timer/counter 8 bit dan 1 buah timer/counter 16 bit. Ketiga modul timer/counter ini dapat diatur dalam mode yang berbeda secara individu dan tidak saling mempengaruhi satu sama lain. Selain itu, semua timer/counter juga dapat difungsikan sebagai sumber interupsi. Masing-masing timer/counter ini memiliki register tertentu yang digunakan untuk mengatur mode dan cara kerjanya.

Serial Peripheral Interface (SPI) merupakan salah satu mode komunikasi serial *synchronous* kecepatan tinggi yang dimiliki oleh Atmega16. *Universal Synchronous and Asynchronous Serial Receiver and Transmitter* (USART) juga merupakan salah satu mode komunikasi serial yang dimiliki oleh Atmega16. USART merupakan komunikasi yang memiliki fleksibilitas tinggi, yang dapat digunakan untuk melakukan transfer data baik antar mikrokontroler maupun dengan modul-modul eksternal termasuk PC yang memiliki fitur USART.

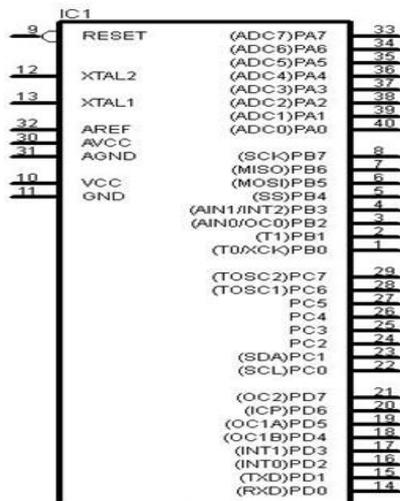
USART memungkinkan transmisi data baik secara *synchronous* maupun *asynchronous*, sehingga dengan memiliki USART pasti kompatibel dengan UART. Pada ATmega16, secara umum pengaturan mode *synchronous* maupun *asynchronous* adalah sama. Perbedaannya hanyalah terletak pada sumber clock.

Jika pada mode *asynchronous* masing-masing peripheral memiliki sumber clock sendiri, maka pada mode *synchronous* hanya ada satu sumber clock yang digunakan secara bersama-sama. Dengan demikian, secara hardware untuk mode *asynchronous* hanya membutuhkan 2 pin yaitu TXD dan RXD, sedangkan untuk mode *synchronous* harus 3 pin yaitu TXD, RXD dan XCK.

2.2.2 Pin-Pin Mikrokontroler Atmega16

Konfigurasi *pin* Atmega16 dengan kemasan 40 pin DIP (*Dual Inline Package*) dapat dilihat pada gambar 2.11. Dari gambar di atas dapat dijelaskan fungsi dari masing-masing *pin* Atmega16 sebagai berikut:

1. VCC merupakan *pin* yang berfungsi sebagai masukan catu daya.
2. GND merupakan *pin Ground*.



Gambar 2.1 Konfigurasi pin Atmega16.

3. Port A (PortA0...PortA7) merupakan *pin input/output* dua arah dan *pin* masukan ADC.

Table 2.1 Fungsi Khusus Port B

PIN	Fungsi Khusus
PB7	SCK (<i>SPI Bus Serial Clock</i>)
PB6	MISO (<i>SPI Bus Master Input/Slave Output</i>)
PB5	MOSI (<i>SPI Bus Master Out/Slave Input</i>)
PB4	SS (<i>SPI Slave Select Input</i>)
PB3	AIN1 (<i>Analog Comparator Negative Input</i>) OC0 (<i>Timer/Counter0 Out Compare Match Output</i>)
PB2	AIN2 (<i>Analog Comparator Positive Input</i>) INT2 (<i>External Interrupt 2 Input</i>)
PB1	TI (<i>Timer/Counter External Counter Input</i>)
PB0	T0 T1 (<i>Timer/Counter External Counter Input</i>) XCK (<i>USART External Clock Input/Output</i>)

4. Port C (Merupakan pin Input/Output dua arah dan pin fungsi khusus, seperti dapat dilihat pada table dibawah ini.

Table 2.2 Fungsi Khusus Port C

PIN	Fungsi Khusus
PC7	TOSC2 (<i>Timer Oscillator Pin2</i>)
PC6	TOSC1 (<i>Timer Oscillator pin1</i>)
PC5	<i>Input/Output</i>
PC4	<i>Input/Output</i>
PC3	<i>Input/Output</i>
PC2	<i>Input/Output</i>
PC1	SDA (<i>Two-wire Serial Bus Data Input/Output line</i>)
PC0	SCL (<i>Two-wire Serial Bus Clock Line</i>)

5. Port D (PortD0-PortD7) Merupakan pin *input/Output* dua arah dan pin fungsi khusus yang terlihat pada table dibawah ini.

Table 2.3 Fungsi Port D

PIN	Fungsi Pin
PB7	OC2 (<i>Timer/Counter Compare Match Output</i>)
PB6	ICP (<i>Timer/Counter1 Input Compare</i>)
PB5	OC1A (<i>Timer/Counter1 Output Compare A Match Output</i>)
PB4	OC1B (<i>Timer/Counter1 Output Compare B Match Output</i>)
PB3	INT1 (<i>External Interupt 1 Input</i>)
PB2	INT0 (<i>External Interupt 0 Input</i>)
PB1	TXD (<i>USART Output Pin</i>)
PB0	RXD (<i>USART Input pin</i>)

6. *RESET* merupakan pin yang digunakan untuk *me-reset* mikrokontroler.
7. XTAL1 dan XTAL2 merupakan pin masukkan *Clock* eksternal.
8. AVCC merupakan *pin* masukkan tegangan referensi ADC.
9. AREFF merupakan *pin* masukkan tegangan referensi ADC.

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 : $V_{LM35} = \text{Suhu} * 10 \text{ mV} \dots\dots\dots (1)$ Secara prinsip sensor akan melakukan penginderaan pada saat perubahan suhu setiap suhu 1 °C akan menunjukkan tegangan sebesar 10 mV. (Tri Utomo,dkk :2011)

Table 2.4 Pin LM35

PIN	Fungsi
1	Supply tegangan 5V (+35 V ke-2V)
2	Tegangan Output (6V-1V)
3	Ground

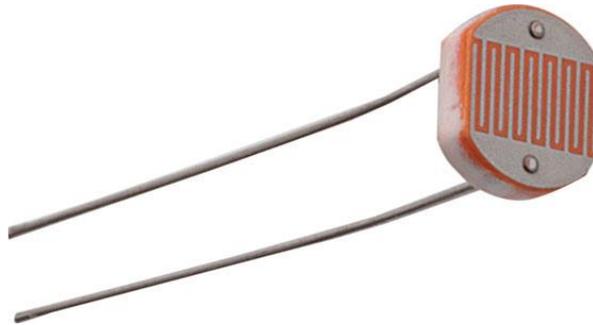
2.2.4 Sensor Cahaya LDR (Light Dependent Resistor)

Resistor peka cahaya atau fotoresistor adalah komponen elektronik yang resistansinya akan menurun jika ada penambahan intensitas cahaya yang mengenainya. Fotoresistor dapat merujuk pula pada light-dependent resistor (LDR), atau fotokonduktor.

Fotoreistor dibuat dari semikonduktor beresistansi tinggi yang tidak dilindungi dari cahaya. Jika cahaya yang mengenainya memiliki frekuensi yang cukup tinggi, foton yang diserap oleh semikonduktor akan menyebabkan elektron memiliki energi yang cukup untuk meloncat ke pita konduksi. Elektron bebas yang dihasilkan (dan pasangan lubangnya) akan mengalirkan listrik, sehingga menurunkan resistansinya.(Supatmi, Sri :2010)

a. Aplikasi yang sering menggunakan sensor

Ini adalah aplikasi pada lampu taman dan lampu di jalan yang bisa menyala di malam hari dan padam di siang hari secara otomatis. Atau bisa juga kita gunakan di kamar kita sendiri.



Gambar 2.3 Sensor LDR

b. KARAKTERISTIK LDR

LDR adalah suatu bentuk komponen yang mempunyai perubahan resistansi yang besarnya tergantung pada cahaya. Karakteristik LDR terdiri dari dua macam yaitu Laju Recovery dan Respon Spektral.

c. Respon Spektral Sensor Cahaya LDR (Light Dependent Resistor)

Sensor Cahaya LDR (Light Dependent Resistor) tidak mempunyai sensitivitas yang sama untuk setiap panjang gelombang cahaya yang jatuh padanya (yaitu warna). Bahan yang biasa digunakan sebagai penghantar arus listrik yaitu Tembaga, aluminium, baja, emas dan perak. Dari kelima bahan tersebut tembaga merupakan penghantar yang paling banyak, digunakan karena mempunyai daya hantaryang baik.(TEDC,1998)

d. Prinsip Kerja Sensor Cahaya LDR (Light Dependent Resistor)

Resistansi Sensor Cahaya LDR (Light Dependent Resistor) akan berubah seiring den-gan perubahan intensitas cahaya yang

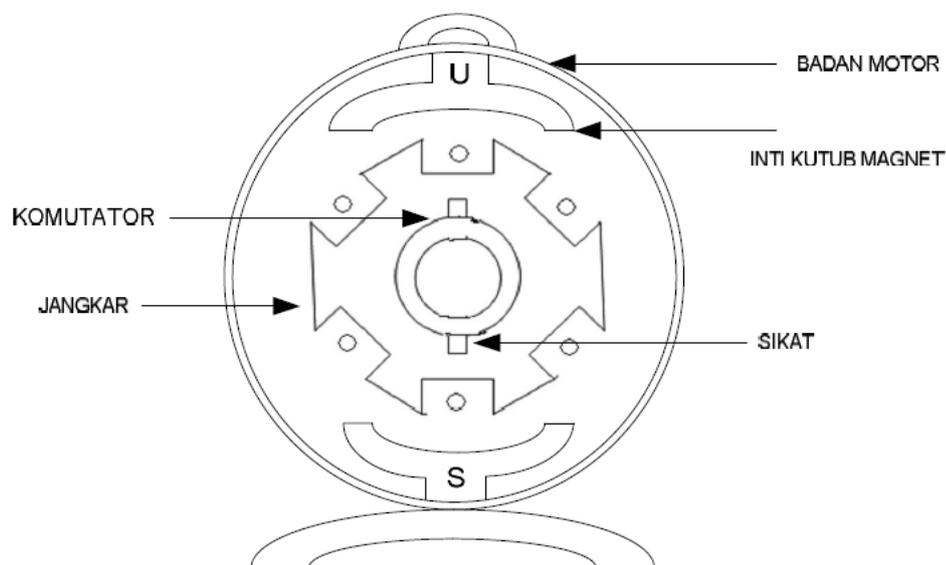
mengenaunya atau yang ada disekitarnya. Dalam keadaan gelap resistansi LDR seki-tar $10M\Omega$ dan dalam keadaan terang sebe-sar $1K\Omega$ atau kurang. LDR terbuat dari ba-han semikonduktor seperti kadmium sul-fida. Dengan bahan ini energi dari cahaya yang jatuh menyebabkan lebih banyak mua-tan yang dilepas atau arus listrik meningkat. Artinya resistansi bahan telah men-galami penurunan.

2.2.5 Motor DC

Motor DC adalah suatu perangkat elektromagnetik yang berfungsi untuk mengubah energi listrik menjadi mekanik. Selain dari pada itu motor DC adalah motor yang bekerja apabila diberi arus searah pada terminal masukannya. (Suryawan, 10)

1. Konstruksi Motor DC

Konstruksi motor DC terdiri atas beberapa bagian seperti yang terlihat pada gambar 2.17 dibawah ini



Gambar 2.4 Konstruksi Motor DC (Suryawan, 2008)

a. Badan Motor

Fungsi utama dari badan motor yaitu sebagai bagian dari tempat mengalirnya *fluks* magnet yang dihasilkan kutub-kutub magnet. Disamping itu badan motor juga berfungsi untuk melindungi bagian-bagian mesin lainnya. Badan motor terbuat dari bahan *ferromagnetic* yaitu dari besi tuang untuk motor kecil dan dari plat-plat campuran baja untuk motor yang berukuran besar.

b. Inti Kutub Magnet

Inti kutub magnet berfungsi untuk menghasilkan fluksi magnet, dimana magnet yang digunakan adalah magnet permanen.

c. Sikat-sikat

Fungsi dari sikat-sikat adalah sebagai jembatan bagi aliran arus listrik dari sumber tegangan ke lilitan jangkar. Sikat-sikat juga memegang peranan penting dalam terjadinya komutasi.

d. Komutator

Komutator berfungsi sebagai pendistribusi arus ke kumparan jangkar yang terkena medan magnet terbanyak. Agar menghasilkan putaran yang lebih baik (lebih cepat) maka komutator yang digunakan berjumlah cukup banyak.

e. Jangkar

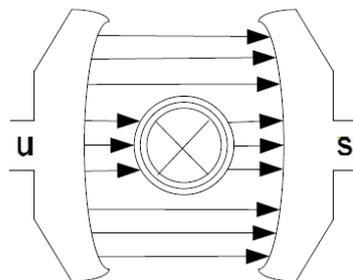
Jangkar pada motor DC pada umumnya berbentuk silinder yang terbuat dari bahan *ferromagenetic* dengan diberi alur-alur pada permukaannya sebagai tempat melilitkan kumparan-kumparan (lilitan jangkar) terletak dalam daerah yang induksi magnetnya besar sehingga gaya gerak listrik (ggl) yang terbentuk dapat bertambah besar. Untuk menghindari / mengurangi panas yang terbentuk karena adanya arus liar, jangkar dibuat dari bahan berlapis-lapis tipis sejenis baja silikon.

Suatu motor DC dapat berfungsi apabila memiliki kumparan medan, kumparan jangkar, dan celah udara. Kumparan medan pada motor DC digunakan untuk menghasilkan medan magnet, sedangkan kumparan jangkar digunakan untuk mengibaskan gaya gerak listrik (ggl) pada konduktor-konduktor yang terletak pada alur-alur jangkar, dan celah udara digunakan untuk memberi ruang berputarnya jangkar dalam medan magnet.

Pada motor DC, kumparan medan yang berbentuk kutub sepatu merupakan *stator* (bagian yang tidak berputar), kumparan jangkar merupakan rotor (bagian yang berputar). Motor DC akan berputar karena adanya :

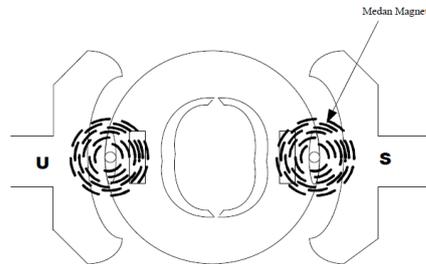
- a. Garis-garis gaya medan magnet (*fluks*), antara kutub yang berada di *stator*.
- b. Penghantar yang dilalui arus listrik dan ditempatkan pada jangkar yang berada diantara kutub-kutub magnet.

Penghantar yang akan dilalui oleh arus listrik juga menghasilkan garis-garis gaya medan magnet (*fluks*), sehingga akan terjadi interaksi antara garis-garis medan magnet (*fluks*) ini dengan garis-garis gaya medan magnet yang dihasilkan oleh kutub-kutub magnet. Interaksi antara dua garis-garis gaya medan magnet tersebut akan menghasilkan gaya (F) yang selanjutnya akan menghasilkan torsi (τ) yang akan memutar jangkar.



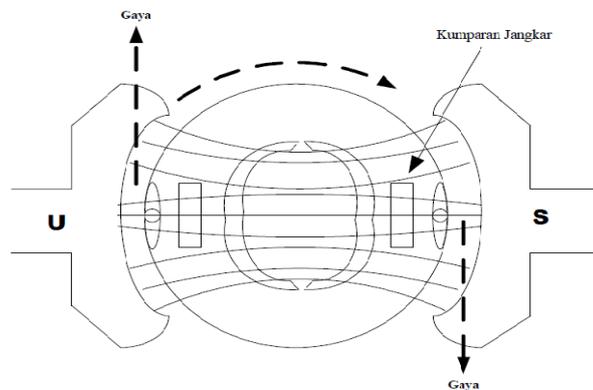
Gambar 2.5 Medan Magnet Yang Dihasilkan Oleh Kutub (Suryawan, 2008)

Dalam gambar diatas dapat diketahui bahwa kutub-kutub magnet akan menghasilkan garis-garis gaya medan magnet (fluks).



Gambar 2.6 Medan Magnet Yang Timbul Pada Penghantar (Suryawan, 2008)

Gambar diatas menjelaskan bahwa pada saat penghantar dialiri arus, maka akan menimbulkan medan magnet (garis – garis gaya *fluks*).

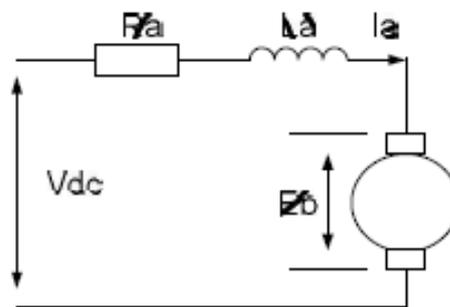


Gambar 2.7 Torsi Yang Dihasilkan Dari Perbedaan Gaya (Suryawan, 2008)

Gambar di atas menunjukkan adanya interaksi yang terjadi pada kedua medan magnet yang menimbulkan medan magnet yang tidak seragam sehingga timbul gaya (F) yang akan menghasilkan torsi ($\tilde{\alpha}$) yang akan memutar jangkar.

2.2.6 Pengaturan Kecepatan Motor DC

Motor arus searah dapat digambarkan dengan model sebagai berikut



Gambar 2.8 Pemodelan Motor Searah (Suryawan, 2008)

Saat motor diberi tegangan, arus listrik mengalir dari terminal negatif ke terminal positif melalui kumparan medan. Arus ini juga akan mengalir melalui kumparan jangkar sehingga membangkitkan medan magnet pada jangkar tersebut. Medan magnet pada jangkar dan kumparan medan ini akan mengakibatkan jangkar berputar seperti yang dijelaskan di atas. Kecepatan putaran jangkar dapat ditingkatkan dengan meningkatkan arus yang melewati masing - masing kumparan. Arus yang melewati jangkar dan kumparan medan ini dapat diubah dengan mengubah nilai tegangan yang diberikan. Semakin tinggi tegangan yang diberikan maka arus yang melewati kumparan juga semakin besar.

Hubungan antara tegangan dan kecepatan dapat dirumuskan pada persamaan kecepatan:

$$\omega = \frac{Eb}{Kb \Phi}$$

$$Eb = Vdc - IaRa$$

Keterangan:

$\dot{\omega}$ = kecepatan motor (radian / detik)

E_b = GGL lawan yang dibangkitkan dalam jangkar (Volt)

K_b = tetapan kecepatan motor (V/A.rad/detik)

\hat{O} = fluks / kutub (Wb)

I_a = arus jangkar (Ampere)

V_{dc} = tegangan yang masuk motor (Volt)

R_a = hambatan pada jangkar (Ohm)

L_a = induktansi dalam jangkar (*Henry*)

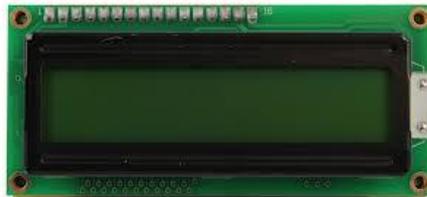
Dari persamaan di atas terlihat bahwa selain dengan mengubah tegangan yang masuk ke motor, pengaturan kecepatan juga dapat dilakukan dengan perubahan *fluks* yaitu dengan menurunkannya.

2.2.7 LCD (Liquid Crystal Display)

LCD (*Liquid Crystal Display*) adalah suatu jenis media tampilan yang menggunakan kristal cair sebagai penampil utama, selain itu LCD juga dapat digunakan untuk menampilkan karakter ataupun gambar.

Penggunaan perangkat LCD sebagai peraga pada alat ini karena LCD banyak memiliki kelebihan :

1. Pemakaian arusnya kecil
2. Dapat menampilkan semua simbol ASCII maupun simbol yang dibuat sendiri
3. Pengendaliannya sangat mudah karena sudah dilengkapi dengan unit pengendali di dalam
4. Mudah dirangkaikan ke sistem mikrokomputer



Gambar 2.9 Liquid Crystal Display (Suyadhi, 2010)

Berikut ini adalah tabel yang menjelaskan mengenai konfigurasi pin dari LCD 16 x 2 :

Tabel 2.5. konfigurasi pin LCD 16 X 2

Pin	Simbol	Level	Tujuan	Fungsi
1	V _{SS}	-	Power Supply	Ground
2	V _{DD}	-	Power Supply	Tegangan Supply (+ 5 Volt)
3	V _{LS}	-	Power Supply	Power supply untuk mendrive LCD guna mengatur kontrasnya

4	RS	H/L	uC	H : Data ; L : Instruction Code
5	R/W	H/L	uC	H : Read ; L : Write
6	E	H/L	uC	Enable
7	DB0	H/L	uC	Data Bus Line
8	DB1	H/L	uC	
9	DB2	H/L	uC	
10	DB3	H/L	uC	
11	DB4	H/L	uC	
12	DB5	H/L	uC	
13	DB6	H/L	uC	
14	DB7	H/L	uC	
15	V+BL	-	Back Ligh Supply	Tegangan Supply (+ 5 Volt)
16	V-BL	-	Back Ligh Supply	Ground

Karakteristik yang ada pada LCD antara lain :

- a. Mempunyai 16 karakter dengan 2 baris tampilan yang terbentuk dari matrik titik (*dot matrix*).
- b. *Duty ratio* : 1/16
- c. ROM pembangkit karakter untuk 192 jenis karakter dengan bentuk karakter huruf : 5 x 7 matrik titik.
- d. Mempunyai 8 tipe RAM pembangkit karakter.
- e. RAM data tampilan dan RAM pembangkit karakter dapat dibaca dari unit Mikrokontroller.
- f. Dilengkapi dengan beberapa perintah yaitu penghapusan tampilan , posisi awal kursor, tampilan karakter kedip (*display clear*), posisi awal kursor (*cursor home*), tampilan karakter kedip (*display character blink*), dan penggeseran tampilan (*display shift*).
- g. Rangkaian pembangkit detak (*clock*) internal.
- h. Catu daya tunggal + 5V.

- i. Rangkaian otomatis reset saat daya dihidupkan.
- j. Pemrosesan dengan CMOS.
- k. Jangkauan suhu 0° C sampai 50° C.

2.2.8 Resistor

Menurut Budiharto (2009), Resistor adalah suatu komponen elektronika yang berfungsi untuk memberikan hambatan terhadap aliran arus listrik. Menurut Suyadhi (2010), Resistor adalah komponen dasar elektornik yang digunakan untuk membatasi jumlah arus yang mengalir dalam satu rangkaian elektronika. Bahan pembuat resistor adalah karbon. Dalam SI (Satuan Internasional), satuan resistansi (hambatan) dari suatu resistor disebut ohm atau dilambangkan dengan simbol Ω (Omega). Satuan ini di ambil oleh penemunya, yaitu Simon Georg Ohm.

Gelang yang terdapat dalam resistor sebenarnya merupakan kode dari nilai resistansi yang terkandung didalamnya, untuk dapat membacanya maka kita harus mengetahui kode tersebut. Berikut adalah kode warna pada resistor:

Tabel 2.6. Nilai Gelang pada resistor

warna	Gelang 1	Gelang 2	Gelang 3	Multiplier	Toleransi
				Gelang 4	Gelang 5
Hitam		0	0	1 Ohm	
Coklat	1	1	1	10 Ohm	± 1 %
Merah	2	2	2	100 Ohm	± 2 %
Orange	3	3	3	1 K Ohm	
Kuning	4	4	4	10 K Ohm	
Hijau	5	5	5	100 K Ohm	± 0,5 %
Biru	6	6	6	1 M Ohm	± 0,25 %
Ungu	7	7	7	10 M Ohm	± 0,10 %
Abu-abu	8	8	8		± 0,05 %
Putih	9	9	9		
Emas				0,1 Ohm	± 5 %
Perak				0,01 Ohm	± 10 %



Gambar 2.10 Resistor (Suyadhi, 200)

2.9 Kapasitor

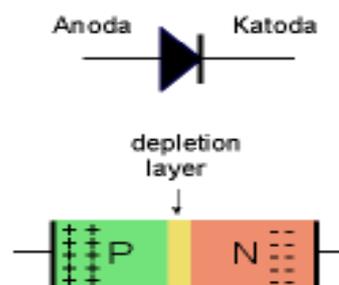
Menurut Budiharto (2009) dan Suyadhi (2010) Kapasitor adalah komponen elektronik yang berfungsi untuk menyimpan muatan listrik. Kapasitor terdiri atas dua buah keping metal sejajar yang dipisahkan oleh isolator yang disebut dielektrik.



Gambar 2.11 Kapasitor (Suyadhi, 2010)

2.10 Dioda

Menurut Budiharto (2007) Dioda adalah piranti semikonduktor yang mengalirkan arus satu arah saja. dioda terbuat dari *germanium* atau *silicon* yang lebih dikenal dengan dioda junction. struktur dari dioda ini, sesuai dengan namanya, adalah sambungan antara semikonduktor tipe P dan Semikonduktor tipe N. semikonduktor tipe P berperan sebagai anoda dan semikonduktor tipe N berperan sebagai katoda. Dengan struktur seperti ini arus hanya dapat mengalir dari sisi P ke sisi N

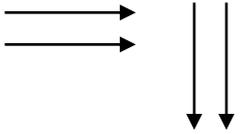
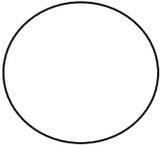
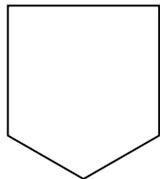


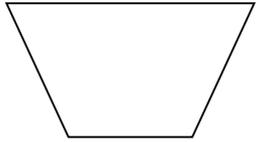
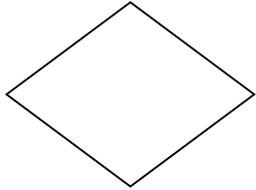
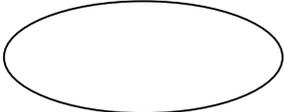
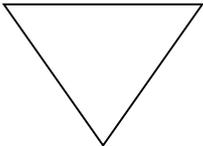
Gambar 2.12 Simbol dan Struktur Dioda (Suyadhi, 2010)

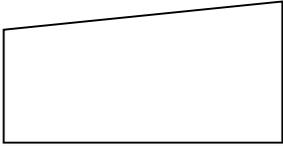
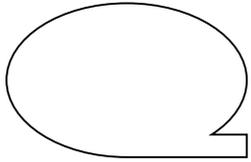
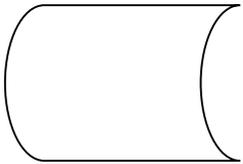
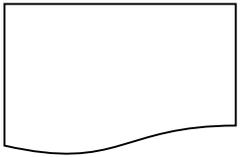
2.3 Flowchart

Menurut Kristanti (2012) dan Redita (2012) *Flowchart* adalah cara penyajian *visual* aliran data melalui sistem informasi, *Flowchart* dapat membantu menjelaskan pekerjaan yang saat ini dilakukan dan bagaimana cara meningkatkan atau mengembangkan pekerjaan tersebut. Dengan menggunakan *flowchart* dapat juga membantu untuk menemukan elemen inti dari sebuah proses, selama garis digambarkan secara jelas antara di mana suatu proses berakhir dan proses selanjutnya dimulai. Adapun simbol-simbol dari *flowchart* adalah sebagai berikut.

Tabel 2.7 Tabel simbol diagram flowchart.

No	Simbol	Keterangan
1		Simbol arus / <i>flow</i> , yaitu menyatakan jalannya arus suatu proses
2		Simbol <i>connector</i> , berfungsi menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama
3		Simbol <i>offline connector</i> , menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda
4		Simbol <i>process</i> , yaitu menyatakan suatu tindakan (proses) yang dilakukan oleh komputer

5		Simbol <i>manual</i> , yaitu menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer
6		Simbol <i>decision</i> , yaitu menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban : ya / tidak
7		Simbol <i>terminal</i> , yaitu menyatakan permulaan atau akhir suatu program
8		Simbol <i>predefined process</i> , yaitu menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal
9		Simbol <i>keying operation</i> , menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai keyboard
10		Simbol <i>offline-storage</i> , menunjukkan bahwa data dalam simbol ini akan disimpan ke suatu media tertentu

11		Simbol <i>manual input</i> , memasukkan data secara manual dengan menggunakan online keyboard
12		Simbol <i>input/output</i> , menyatakan proses input atau output tanpa tergantung jenis peralatannya
13		Simbol <i>magnetic tape</i> , menyatakan input berasal dari pita magnetis atau output disimpan ke pita magnetis
14		Simbol <i>disk storage</i> , menyatakan input berasal dari dari disk atau output disimpan ke disk
15		Simbol <i>document</i> , mencetak keluaran dalam bentuk dokumen (melalui printer)
16		Simbol <i>punched card</i> , menyatakan input berasal dari kartu atau outputmditulis ke kartu

2.4 Software

2.4.1 Pemrograman Bahasa C

Pencipta bahasa C adalah Brian W. Kernighan dan Dennis M. Ritchie pada sekitar tahun 1972. C adalah bahasa pemrograman terstruktur, yang membagi program dalam bentuk sejumlah blok. Tujuannya adalah untuk memudahkan dalam pembuatan dan pengembangan program. Program yang ditulis dengan menggunakan C mudah sekali untuk dipindahkan dari satu jenis mesin ke jenis mesin lainnya. Hal ini berkat adanya standarisasi bahasa C yaitu berupa standar ANSI (American National Standards Institute) yang dijadikan acuan oleh para pembuat kompilator C. (Agus Bejo, 2008).

2.4.1.1 Bentuk Dasar Program C

Sebuah program dalam bahasa C setidaknya harus memiliki sebuah fungsi. Fungsi dasar ini disebut dengan fungsi utama (fungsi main) dan memiliki kerangka program sebagai berikut:

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

Jika kita memiliki beberapa fungsi yang lain maka fungsi utama inilah yang memiliki kedudukan paling tinggi dibandingkan fungsi-fungsi yang lain sehingga setiap kali program dijalankan akan selalu dimulai dari memanggil fungsi utama terlebih dahulu. Fungsi-fungsi yang lain dapat dipanggil setelah fungsi utama dijalankan melalui pernyataan-pernyataan yang berada didalam fungsi utama.

Contoh:

```
// prototype fungsi inisialisasi port
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, 0x0F,
0x00) ;
```

2.4.1.2 Pengenal (identifier)

Merupakan sebuah nama yang diisikan oleh pemrogram untuk menunjukkan identitas dari sebuah konstanta, variable, fungsi, label atau tipe data khusus. Pemberian nama sebuah pengenal dapat ditentukan bebas sesuai keinginan pemrogram tetapi harus memenuhi aturan berikut:

1. Karakter pertama tidak boleh menggunakan angka.
2. Karakter kedua dapat berupa huruf, angka, atau garis bawah.
3. Tidak boleh menggunakan spasi.
4. *Case sensitive*, yaitu huruf kapital dan huruf kecil dianggap berbeda.
5. Tidak boleh menggunakan kata-kata yang merupakan sintaks maupun operator dalam pemrograman C, misalnya: *void, short, const, if, bit, long, case, do, switch, char, float, for, else, break, int, double, include, while*.

2.4.1.3 Tipe Data

Tipe data merupakan bagian program yang paling penting karena tipe data mempengaruhi setiap instruksi yang akan dilaksanakan oleh komputer. Misalnya saja 5 dibagi 2 bisa saja menghasilkan hasil yang berbeda tergantung tipe datanya. Jika 5 dan 2 bertipe *integer* maka akan menghasilkan nilai 2, namun jika keduanya bertipe *float* maka akan menghasilkan nilai 2.5000000. Pemilihan tipe data yang tepat akan membuat proses operasi data menjadi lebih efisien dan efektif.

Tabel 2.8. Tipe Data (Agus Bejo, 2009)

Tipe data	Ukuran	Jangkauan nilai
<i>Bit</i>	1 <i>bit</i>	0 atau 1
<i>Char</i>	1 <i>byte</i>	-128 s/d 127
<i>Unsigned char</i>	1 <i>byte</i>	0 s/d 255
<i>Signed char</i>	1 <i>byte</i>	-128 s/d 127
<i>Int</i>	2 <i>byte</i>	-32.768 s/d 32.767
<i>Short int</i>	2 <i>byte</i>	-32.768 s/d 32.767
<i>Unsigned int</i>	2 <i>byte</i>	0 s/d 65.535

<i>Signed int</i>	2 byte	-32.768 s/d 32.767
<i>Long int</i>	4 byte	-2.147.483.648 s/d 2.147.483.647
<i>Signed long int</i>	4 byte	-2.147.483.648 s/d 2.147.483.647
<i>Float</i>	4 byte	1.2*10 ⁻³⁸ s/d 3.4*10 ⁺³⁸

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

Untuk pendeklarasian 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 255. Sedangkan variabel data2 bertipe *char* (1 byte) dan bertanda (*signed*) sehingga dapat menyimpan data dari -128 sampai 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. (Agus Bejo 2008).

2.4.1.5 Pengarah Preprosesor

Pernyataan adalah satu buah instruksi lengkap yang berdiri sendiri. Berikut adalah contoh sebuah pernyataan: `PORTC = 0x0F`; Pernyataan `PORTC = 0x0F`; merupakan sebuah instruksi untuk mengeluarkan data 0x0F ke Port C. (Agus Bejo, 2008).

2.4.1.6 Fungsi Pustaka

Bahasa C memiliki sejumlah fungsi pustaka yang berada pada file-file tertentu dan sengaja disediakan untuk menangani berbagai hal dengan cara memanggil fungsi-fungsi yang telah dideklarasikan dalam file tersebut. Dalam banyak hal, pustaka-pustaka yang tersedia tidak berbentuk kode sumber melainkan dalam bentuk yang telah dikompilasi. Beberapa fungsi pustaka yang telah disediakan oleh CodeVisionAVR antara lain adalah:

1. Fungsi Tipe Karakter (ctype.h).
2. Fungsi Standar I/O (stdio.h).
3. Fungsi matematika (math.h).
4. Fungsi *String* (string.h).
5. Fungsi Konversi BCD (bcd.h).
6. Fungsi Konversi Akses Memori (mem.h).
7. Fungsi Tunda (delay.h).
8. Fungsi LCD (lcd.h).
9. Fungsi I2C (i2c.h). Fungsi SPI (spi.h).
10. Fungsi *Real Time Clock* (RTC) (ds 1302.h, ds1307.h).
11. Fungsi Sensor Suhu LM75, DS1621 dll (lm75.h, ds1621.h).
12. Fungsi Sensor Suhu LM75, DS1621 dll (lm75.h, ds1621.h).

2.4.1.7 Pernyataan IF

Pernyataan *if* digunakan untuk melakukan pengambilan keputusan terhadap dua buah kemungkinan yaitu mengerjakan suatu blok pernyataan atau tidak. Bentuk pernyataan *if* adalah sebagai berikut:

```
If (kondisi)
{
    // blok pernyataan yang akan dikerjakan
    // jika kondisi if terpenuhi
}
```

contoh :

```
If (PINA>0x80)
```

```
{
    Dataku = PINA; PORTC=0xFF;
}
```

Pernyataan *if* diatas akan mengecek apakah data yang terbaca pada *port A* (PINA) nilainya lebih dari 0x80 atau tidak, jika ya maka variabel *dataku* diisi dengan nilai PINA dan data 0xFF dikeluarkan ke *port C*.

Apabila dalam blok pernyataan hanya terdapat satu pernyataan saja maka tanda { dan } dapat dihilangkan seperti contoh berikut:

```
If (PINA>0x80)
PORTC = 0xFF
```

2.4.1.8 Pernyataan Switch

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

switch (ekspresi)

```
{
    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 (pernyataan_1 sampai dengan pernyataan_default) 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*. Contoh:

```
switch (PINA)
{
    Case 0xFE : PORTC = 0x00; break;
    Case 0xFD : PORTC = 0xFF; break;
}
```

Pernyataan di atas berarti membaca port A, kemudian datanya (PINA) akan dicocokkan dengan nilai *case*. Jika PINA bernilai 0xFE maka data 0x00 akan dikeluarkan ke *port C* kemudian program keluar dari pernyataan *switch* tetapi jika PINA bernilai 0xFD maka data 0xFF akan dikeluarkan ke *port C* kemudian program keluar dari pernyataan *switch*.