BAB II
LANDASAN TEORI

Pada bab ini berisi tentang teori mengenai permasalahan yang dibahas dalam tugas akhir ini, dimulai dari definisi logika *fuzzy*, mikrokontroler ATMega32, sensor ultrasonik SR-04, *Pulse With Modulation* (PWM), motor DC, motor servo, *driver* motor, *Liquid Crystal Display* (LCD), *rotary encoder*, *remote control* (RX dan TX), komparator, sampai dengan perangkat lunak (*software*) bahasa C sebagai pemrograman.

2.1 Logika Fuzzy

Dalam kondisi yang nyata, beberapa aspek dalam dunia nyata selalu atau biasanya berada di luar model matematis dan bersifat samar. Konsep ketidak pastian inilah yang menjadi konsep dasar munculnya konsep logika *fuzzy*. Pencetus gagasan logika *fuzzy* adalah L.A. Zadeh (1965) dari California University.

Logika *fuzzy* adalah salah satu cara yang tepat untuk memetakan suatu ruang *input* ke dalam suatu ruang *output*. Logika *fuzzy* berbeda dengan logika digital biasa, dimana logika digital biasa hanya mengenal 2 keadaan yaitu: “ya” dan “tidak” atau ”*on”* dan “*off”* atau “*high”* dan “*low”* atau "1" dan "0". Sedangkan logika *fuzzy* meniru cara berpikir manusia dengan menggunakan konsep sifat kesamaran suatu nilai. Dengan himpunan *fuzzy*, suatu objek dapat menjadi anggota dari banyak himpunan dengan derajat keanggotaan yang berbeda dalam masing-masing himpunan.

2.1.1 Himpunan *Fuzzy*

Himpunan *fuzzy* merupakan suatu grup yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel *fuzzy*. Himpunan *fuzzy* memiliki 2 atribut, yaitu:

a. linguistik, yaitu penamaan suatu grup yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alami, seperti muda, parobaya, tua.
b. numerik, suatu nilai (angka) yang menunjukkan ukuran dari suaru variabel seperti 10,20,30.

Beberapa hal yang perlu diketahui dalam memahami *fuzzy*, yaitu:

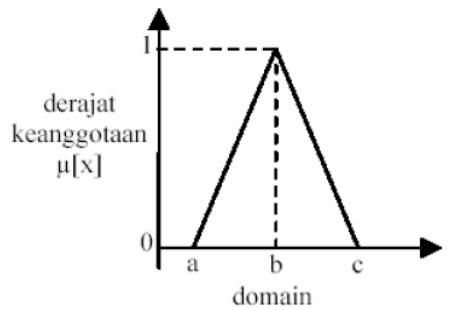
a. variable *fuzzy*, merupakan variabel yang hendak dibahas dalam suatu sistem *fuzzy*. Contoh: umur, suhu, permintaan, dan lain sebagainya.
b. semesta pembicaraan, adalah keseluruhan nilai yang diperbolehkan untuk dioperasikan dalam suatu variabel *fuzzy*. Semesta pembicara merupakan himpunan bilangan real yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan. Nilai semesta pembicara dapat berupa bilangan positif maupun negatif. Adakalanya nilai semesta pembicara ini tidak dibatasi nilai atasnya
c. domain himpunan *fuzzy*, merupakan keseluruhan nilai yang diizinkan dalam semesta pembicaraan dan boleh dioperasikan dalam suatu himpunan *fuzzy*. Seperti halnya semesta pembicaraan, domain merupakan himpunan bilangan *real* yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan. Nilai domain dapat berupa bilangan positif maupun bilangan negatif.

2.1.2 Fungsi Keanggotaan *(Membership Function)*

Fungsi keanggotaan *(Membership Function)* adalah suatu kurva yang menunjukkan pemetaan titik-titik *input* data ke dalam nilai keanggotaannya (sering juga disebut dengan derajat keanggotaan) yang memiliki interval antara 0 sampai 1. Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi. Ada 2 fungsi yang digunakan dalam tugas akhir ini, yaitu:

a. representasi kurva segitiga

Kurva segitiga pada dasarnya merupakan gabungan antara 2 garis linear seperti terlihat pada Gambar 2.1.



Gambar 2.1 Representasi Kurva Segitiga

Representasi kurva segitiga adalah sebagai berikut:

$$\mu[x] = \begin{cases} 0 & ; x \leq a \; atau \; x \geq c \\ \frac{(x-a)}{(b-a)} & ; a \leq x \leq b \\ \frac{(c-x)}{(c-b)} & ; b \leq x \leq c \end{cases} \quad \text{.....(2.1)}$$

Untuk menentukan nilai minimum dan maximum fungsi keanggotaan pada representasi kurva segitiga, dapat menggunakan persamaan dibawah ini.

$$\mu[x] = max\left(min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right) \text{.....(2.2)}$$

dimana:

a = nilai domain terkecil yang mempunyai derajat keanggotaan nol

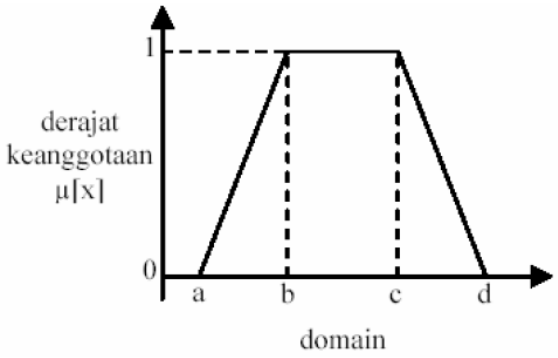
b = nilai domain yang mempunyai derajat keanggotaan satu

c = nilai domain terbesar yang mempunyai derajat keanggotaan nol

x = nilai *input* yang akan diubah ke dalam bilangan *fuzzy*

b. representasi kurva trapesium

Kurva trapesium pada dasarnya seperti bentuk segitiga, hanya saja ada beberapa titik yang memiliki nilai keanggotaan 1. Representasi kurva travesium dapat dilihat pada Gambar 2.2.



Gambar 2.2 Representasi Kurva Trapesium

Representasi kurva trapesium adalah sebagai berikut:

$$\mu[x] = \begin{cases} 0 & ; x \leq a \; atau \; x \geq d \\ \frac{(x-a)}{(b-a)} & ; a \leq x \leq b \\ 1 & ; \; b \leq x \leq c \\ \frac{(d-x)}{(d-c)} & ; c \leq x \leq d \end{cases} \quad \text{.....(2.3)}$$

Untuk menentukan nilai minimum dan maximum fungsi keanggotaan pada representasi kurva trapesium, dapat menggunakan persamaan dibawah ini.

$$\mu[x] = \; max\left(min\left(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c}\right), 0\right) \text{.....(2.4)}$$

dimana:

a = nilai domain terkecil yang mempunyai derajat keanggotaan nol

b = nilai domain terkecil yang mempunyai derajat keanggotaan satu

c = nilai domain terbesar yang mempunyai derajat keanggotaan satu

d = nilai domain terbesar yang mempunyai derajat keanggotaan nol

x = nilai *input* yang akan diubah ke dalam bilangan *fuzzy*

2.1.3 Operator Dasar *Fuzzy*

Pada logika *fuzzy*, Ada beberapa operasi yang didefinisikan secara khusus untuk mengkombinasi dan memodifikasi himpunan *fuzzy*. Nilai keanggotaan sebagai hasil dari 2 operasi disebut α-predikat. Ada tiga operator dasar yaitu :

a. operator AND

Operator ini berhubungan dengan operasi interseksi pada himpunan. Hasil opeasi dengan menggunakan operator AND diperoleh dengan mengambil nilai keanggotaan terkecil antara 2 himpunan yang bersangkutan.

$$\mu A \cap B = min\ (\mu A[x], \mu B[x]) \quad \text{(2.5)}$$

b. operator OR

Operator ini berhubungan dengan operasi union pada himpunan. Hasil opeasi dengan menggunakan operator OR diperoleh dengan mengambil nilai keanggotaan terbesar antara 2 himpunan yang bersangkutan.

$$\mu A \cup B = max\ (\mu A[x], \mu B[y]) \quad \text{(2.6)}$$

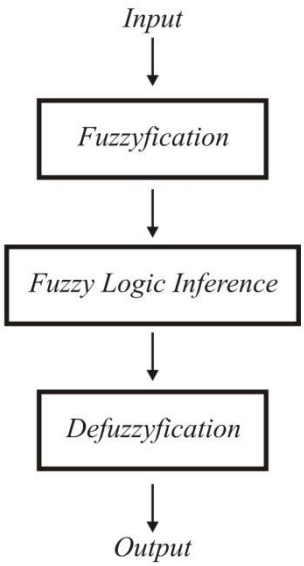
c. operator NOT

Operator ini berhubungan dengan operasi komplemen himpunan. Hasil operasi dengan menggunakan operator NOT diperoleh dengan mengurangkan nilai keanggotaan elemen pada himpunan dengan 1.

$$\mu A' = 1 - \mu A[x] \text{.....} \text{.....} (2.7)$$

2.1.4 Tahapan Membangun Sistem *Fuzzy*

Pada tahap membangun sistem *fuzzy* tergantung pada metoda yang akan digunakan, karena banyak metoda untuk membangun sistem *fuzzy*. Secara garis besar dapat tahap membangun sistem *fuzzy* dilihat pada Gambar 2.3 sebagai berikut.



Gambar 2.3 Proses Sistem Logika *Fuzzy*

a. *Fuzzyfication*

Proses *fuzzyfication* yaitu mengubah nilai suatu masukan menjadi suatu fungsi keanggotaan *fuzzy*. Pada proses ini *membership function* ditentukan. Melalui fungsi keanggotaan yang telah disusun maka nilai-nilai masukan tersebut menjadi informasi *fuzzy* yang berguna nantinya untuk proses pengolahan secara *fuzzy* pula.

b. *Fuzzy Logic Inference*

Ada beberapa proses pada *Fuzzy Logic Inference*, yaitu :

- mengaplikasikan aturan *(Fuzzy Rule)* pada masukan *fuzzy* yang dihasilkan dalam proses *fuzzyfication.*
- mengevaluasi tiap aturan dengan masukan yang dihasilkan dari proses *fuzzyfication* dengan mengevaluasi hubungan atau derajat keanggotaannya.

c. *Defuzzyfication*

Proses *defuzzyfication* merupakan pengubahan kembali data-data *fuzzy* kembali ke dalam bentuk numerik yang dapat dikirimkan ke peralatan pengendalian. Proses *defuzzyfication* dapat dilakukan dengan beberapa cara, diantaranya:

- *centroid of area*

$$Z_{COA} \frac{\int z\mu A(z) z\, dz}{\int z\mu A(z) dz} \text{.....(2.8)}$$

- *bisector of area*

$$\int_{\alpha}^{z\ BOA} \mu A(z) dz = \int_{z\ BOA}^{\beta} \mu A(z) dz \text{.....(2.9)}$$

- *mean of maximum*

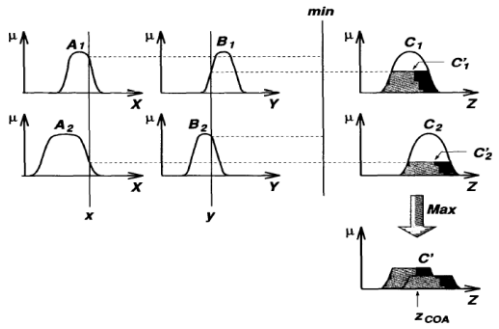
$$Z_{MOM=\frac{\int z'\, z\, dz}{\int z'\, dz}} \text{.....(2.10)}$$

2.1.5 *Fuzzy Inference System*

Sistem inferensi *fuzzy* (*Fuzzy Inference System* atau FIS) merupakan suatu kerangka komputasi yang didasarkan pada teori himpunan *fuzzy*, aturan *fuzzy* berbentuk IF – THEN, dan penalaran *fuzzy*. Terdapat beberapa jenis FIS, salah satunya adalah mamdani dan sugeno.

a. FIS mamdani

FIS yang paling mudah dimengerti, karena paling sesuai dengan naluri manusia adalah FIS mamdani. FIS tersebut bekerja berdasarkan kaidah-kaidah linguistik dan memiliki algoritma *fuzzy* yang menyediakan sebuah aproksimasi untuk dimasuki analisa matematik. FIS mamdani dapat dilihat pada Gambar 2.4.

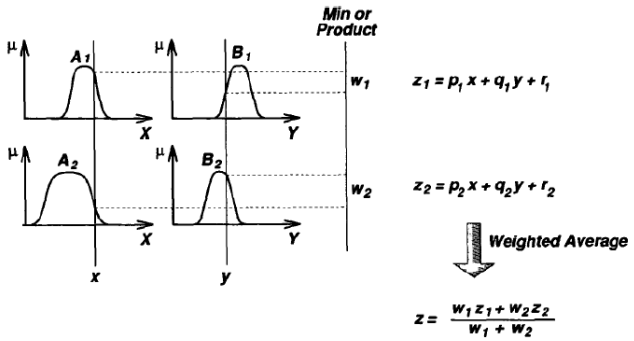


Gambar 2.4 FIS Mamdani

b. FIS sugeno

Penalaran dengan metode sugeno hampir sama dengan penalaran mamdani, hanya saja keluaran sistem tidak berupa himpunan *fuzzy* melainkan berupa konstanta atau persamaan linier. Metoda ini diperkenalkan oleh Takagi-Sugeno Kang pada tahun 1985. Sistem *fuzzy* sugeno memperbaiki kelemahan yang dimiliki oleh sistem *fuzzy* murni untuk menambah suatu perhitungan matematika sederhana sebagai bagian THEN. Pada perubahan ini, sistem *fuzzy* memiliki suatu nilai rata-rata tertimbang *(Weighted Average Values)* di dalam bagian aturan *fuzzy* IF-THEN. Sistem *fuzzy* sugeno juga memiliki kelemahan terutama pada bagian THEN, yaitu dengan adanya perhitungan matematika sehingga tidak dapat menyediakan kerangka alami untuk merepresentasikan

pengetahuan manusia dengan sebenarnya. FIS sugeno dapat dilihat pada Gambar 2.5.



Gambar 2.5 FIS Sugeno

2.2 Mikrokontroler ATmega-32

Mikrokontroler merupakan suatu *device* yang didalamnya sudah terintegrasi dengan I/O *Port*, RAM, ROM, sehingga dapat digunakan untuk berbagai keperluan kontrol. Mikrokontroler AVR Atmega-32 merupakan *low power* CMOS mikrokontroler 8-bit yang dikembangkan oleh Atmel dengan arsitektur RISC (*Reduced Instruction Set Computer*) sehingga dapat mencapai *throughtput* eksekusi instruksi 1 MIPS (*Million Instruction Per Second*). Bentuk fisik Atmega-32 dapat dilihat pada Gambar 2.6.



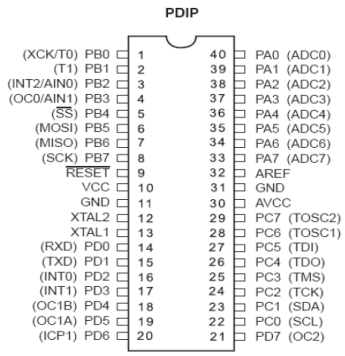
Gambar 2.6 ATmega-32

Karakteristik dari mikrokontroler ini adalah:

- Menggunakan arsitektur AVR RISC
  - 131 perintah dengan satu *clock cycle*
  - 32 x 8 register umum

- Data dan program memori
    - 32 Kb *In-System Programmable Flash*
    - 2 Kb SRAM
    - 1 Kb *In- System EEPROM*
- 8 *Channel* 10-bit ADC
- *Two Wire Interface*
- *USART Serial Communication*
- *Master/Slave SPI Serial Interface*
- *On-Chip Oscillator*
- *Watch-dog Timer*
- 32 *Bi-directional* I/O
- Tegangan operasi 2,7 – 5,5 V

Arsitektur AVR ini menggabungkan perintah secara efektif dengan 32 register umum. Semua register tersebut langsung terhubung dengan *Arithmetic Logic Unit* (ALU) yang memungkinkan 2 register terpisah diproses dengan satu perintah tunggal dalam satu *clock cycle*. Hal ini menghasilkan kode yang efektif dan kecepatan prosesnya 10 kali lebih cepat dari pada mikrokontroler CISC biasa. Konfigurasi pin Atmega-32 tersebut dapat dilihat pada Gambar 2.7.



Gambar 2.7 Pin-Pin ATMega32

Secara fungsional konfigurasi pin ATMega32 adalah sebagai berikut:

a. VCC, tegangan sumber

b. GND (*Ground)*

c. Port A (PA7 – PA0)

Port A adalah 8-bit port I/O yang bersifat *bi-directional* dan setiap pin memilki *internal pull-up* resistor. *Output buffer* port A dapat mengalirkan arus sebesar 20 mA. Ketika port A digunakan sebagai *input* dan di *pull-up* secara langsung, maka port A akan mengeluarkan arus jika *internal pull-up* resistor diaktifkan. Pin-pin dari port A memiliki fungsi khusus yaitu dapat berfungsi sebagai *channel* ADC *(Analog to Digital Converter)* sebesar 10 bit. Fungsi-fungsi khusus pin-pin port A dapat ditabelkan seperti yang tertera pada Tabel 2.1.

Tabel 2.1 Fungsi Khusus Port A

Port	Fungsi
PA7	ADC *input channel* 7 (ADC7)
PA6	ADC *input channel* 6 (ADC6)
PA5	ADC *input channel* 5 (ADC5)
PA4	ADC *input channel* 4 (ADC4)
PA3	ADC *input channel* 3 (ADC3)
PA2	ADC *input channel* 2 (ADC2)
PA1	ADC *input channel* 1 (ADC1)
PA0	ADC *input channel* 0 (ADC0)

d. Port B (PB7 – PB0)

Port B adalah 8-bit port I/O yang bersifat *bi-directional* dan setiap pin mengandung *internal pull-up* resistor. *Output buffer* port B dapat mengalirkan arus sebesar 20 mA. Ketika port B digunakan sebagai *input*

dan di *pull-down* secara *external*, port B akan mengalirkan arus jika *internal pull-up* resistor diaktifkan. Pin-pin port B memiliki fungsi-fungsi khusus, diantaranya:

- SCK port B,bit 7 (input pin clock untuk *up/downloading memory*)
- MISO port B, bit 6 (pin *output* data untuk *uploading memory*)
- MOSI prt B, bit 5 (pin *input* untuk *downloading memory*)

Fungsi-fungsi khusus pin-pin port B dapat ditabelkan seperti pada Tabel 2.2

Tabel 2.2 Fungsi Khusus Port B

Port	Fungsi
PB7	*SPI Bus Serial Clock* (SCK)
PB6	*SPI Bus Master Input/Slave Output* (MISO)
PB5	*SPI Bus Master Output/Slave Input* (MOSI)
PB4	*SPI Slave Select Input* (SS)
PB3	*Analog Comparator Negative Input* (AIN1) *Timer/Counter0 Output Compare Match Output* (OCO)
PB2	*Analog Comparator Positive Input* (AIN0) *External Interrupt 2 Input* (INT2)
PB1	*Timer/Counter1 External Counter Input* (T1)
PB0	*Timer/Counter External Counter Input* (T0) *USART External Clock Input/Output* (XCK)

e. Port C (PC7 – PC0)

Port C adalah 8-bit port I/O yang berfungsi *bi-directional* dan setiap pin memiliki *internal pull-up* resistor. *Output buffer* port C dapat mengalirkan arus sebesar 20 mA. Ketika port C digunakan sebagai *input* dan di *pull-down* secara langsung, maka port C akan mengeluarkan arus jika *internal*

pull-up resistor diaktifkan. Fungsi-fungsi khusus pin-pin port C dapat ditabelkan seperti yang tertera pada Tabel 2.3.

Tabel 2.3 Fungsi Khusus Port C

Port	Fungsi
PC7	*Timer Oscillator* Pin 2 (TOSC2)
PC6	*Timer Oscillator* Pin 1 (TOSC1)
PC5	JTAG *Test Data In* (TD1)
PC4	JTAG *Test Data Out* (TD0)
PC3	JTAG *Test Mode Select* (TMS)
PC2	JTAG *Test Clock* (TCK)
PC1	*Two-wire Serial Bus Data Input/Output Line* (SDA)
PC0	*Two-wire Serial Bus Clock Line* (SCL)

f. Port D (PD7 – PD0)

Port D adalah 8-bit port I/O yang berfungsi *bi-directional* dan setiap pin memiliki *internal pull-up* resistor. *Output buffer* port D dapat mengalirkan arus sebesar 20 mA. Ketika port D digunakan sebagai *input* dan di *pulldown* secara langsung, maka port D akan mengeluarkan arus jika *internal pull-up* resistor diaktifkan. Fungsi-fungsi khusus pin-pin port D dapat ditabelkan seperti yang tertera pada Tabel 2.4.

Tabel 2.4 Fungsi Khusus Port D

Port	Fungsi
PD7	*Timer / Counter2 Output Compare Match Output* (OC2)
PD6	*Timer/Counter1 Input Capture Pin* (ICP1)
PD5	*Timer/Counter1 Output Compare B Match* (OCIB)
PD4	JTAG *Test Data Out* (TD0)
PD3	*External Interrupt 1 Input* (INT1)
PD2	*External Interrupt 0 Input* (INT0)
PD1	USART Output Pin (TXD)
PD0	USART Input Pin (RXD)

2.3 Sensor Jarak Ultrasonik (SR-04)

Sensor ultrasonik SR-04 adalah sensor yang bekerja berdasarkan prinsip pantulan gelombang suara dan digunakan untuk mendeteksi keberadaan suatu objek tertentu di depannya, frekuensi kerjanya pada daerah di atas gelombang suara dari 40 KHz hingga 400 KHz. Sensor ultrasonik terdiri dari dari dua unit, yaitu unit pemancar dan unit penerima. Bentuk sensor ultrasonik SR-04 dapat dilihat pada Gambar 2.8.

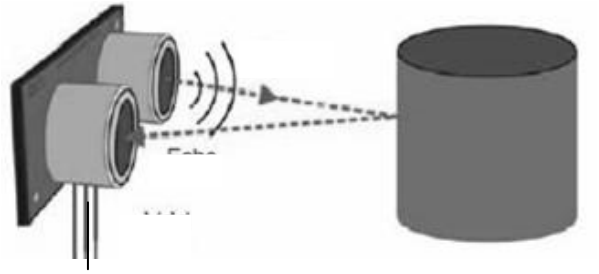


Gambar 2.8 Sensor Jarak Ultrasonik (SR-04)

Struktur unit pemancar dan penerima sangatlah sederhana, sebuah kristal *piezoelectric* dihubungkan dengan mekanik jangkar dan hanya dihubungkan dengan diafragma penggetar. Tegangan bolak-balik yang memiliki frekuensi kerja 40 KHz – 400 KHz diberikan pada plat logam. Struktur atom dari kristal *piezoelectric* akan berkontraksi (mengikat), mengembang atau menyusut terhadap polaritas tegangan yang diberikan dan ini disebut dengan efek *piezoelectric*.

Kontraksi yang terjadi diteruskan ke diafragma penggetar sehingga terjadi gelombang ultrasonik yang dipancarkan ke udara (tempat sekitarnya). Pantulan gelombang ultrasonik akan terjadi bila ada objek tertentu dan pantulan gelombang ultrasonik akan diterima kembali oleh unit sensor penerima. Selanjutnya unit sensor penerima akan menyebabkan diafragma penggetar akan bergetar dan efek *piezoelectric* menghasilkan sebuah tegangan bolak-balik dengan frekuensi yang

sama. Untuk lebih jelas tentang prinsip kerja dari sensor ultrasonik dapat dilihat prinsip dari sensor ultrasonik pada Gambar 2.9.

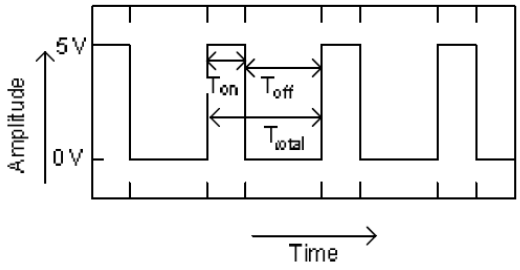


Gambar 2.9 Cara Kerja Sensor Ultrasonik

Besar amplitudo sinyal elekrik yang dihasilkan unit sensor penerima tergantung dari jauh dekatnya objek yang dideteksi serta kualitas dari sensor pemancar dan sensor penerima. Proses sensoring yang dilakukan pada sensor ini menggunakan metode pantulan untuk menghitung jarak antara sensor dengan obyek sasaran. Jarak antara sensor tersebut dihitung dengan cara mengalikan setengah waktu yang digunakan oleh sinyal ultrasonik dalam perjalanannya dari rangkaian pengirim sampai diterima oleh rangkaian penerima, dengan kecepatan rambat dari sinyal ultrasonik tersebut pada media rambat yang digunakannya, yaitu udara.

2.4 *Pulse Width Modulation (PWM)*

PWM merupakan sebuah metoda untuk membangkitkan sinyal keluaran yang periodenya berulang antara *high* atau *low* dimana kita dapat mengontrol durasi sinyal *high* atau *low* sesua dengan yang kita inginkan. Pengaturan PWM merupakan salah satu teknik yang banyak digunakan dalam sistem kendali (*control system*) saat ini. Gambaran sinyal PWM dapat dilihat pada Gambar 2.10.



Gambar 2.10 Sinyal PWM

Duty cycle merupakan perbandingan periode lamanya suatu system bernilai logika *high* dan *low*. Variasi *duty cycle* ini memberikan harga tegangan rata–rata yang berbeda–beda. Sinyal PWM dengan *duty cycle* yang besar memiliki nilai rata-rata tegangan yang lebih besar dibandingkan dengan *duty cycle* kecil. Nilai tegangan yang diberikan sebanding dengan nilai duty cycle yang diberikan. Adapun rumus *duty cycle*, yaitu:

$$Duty\ Cycle = \frac{t_{high}}{(t_{high}+t_{low})} \times 100\% \text{.....(2.11)}$$

Sedangkan untuk menghitung nilai tegangan rata-rata *output* dapat dihitung dengan rumus sebagai berikut:

$$V_{out} = Duty\ Cycle \times V_{total} \text{.....(2.12)}$$

Dimana Vtotal merupakan tegangan yang diberikan untuk mengaktifkan motor.

2.5 Motor DC

Motor DC adalah mesin yang mengubah energi listrik arus searah menjadi energi mekanis. Bentuk fisik motor DC dapat dilihat pada Gambar 2.11.



Gambar 2.11 Motor DC

Cepat lambatnya motor DC dapat diatur berdasarkan *duty cycle* yang diberikan, jika *duty cycle* pada sinyal PWM besar, maka akan menghasilakan rata-rata tegangan yang besar pula sehingga motor DC akan berputar cepat, begitu juga sebaliknya jika *duty cycle* yang diberikan kecil akan mengahasilkan tegangan rata-rata yang kecil dan motor DC akan berputar lebih lambat. Pengaturan nilai tegangan rata-rata tersebut dapat diatur dengan menggunakan persamaan (2.12).

2.6 Motor Servo

Motor servo adalah sebuah motor DC dengan sistem umpan balik tertutup dimana posisi rotornya akan diinformasikan kembali ke rangkaian kontrol yang ada di dalam motor servo. Motor ini terdiri dari sebuah motor DC, serangkaian *gear*, potensiometer, dan rangkaian kontrol. Potensiometer berfungsi untuk menentukan batas sudut dari putaran servo. Sedangkan sudut dari sumbu motor servo diatur berdasarkan lebar pulsa yang dikirim melalui kaki sinyal dari kabel motor. Pada motor servo biasanya terdapat tiga buah kabel, yang pertama untuk tegangan masukkan, yang kedua untuk menerima sinyal PWM, dan yang ketiga untuk ground. Bentuk fisik motor servo dapat dilihat pada Gambar 2.12.



Gambar 2.12 Motor Servo

Motor servo ini terbagi menjadi 2 jenis, yaitu :

a. Motor servo standard

   Motor servo jenis ini hanya mampu bergerak dua arah yaitu searah jarum jam dan kebalikan dari arah jarum jam dengan defleksi masing-masing sudut mencapai 90° sehingga total defleksi sudut dari kanan – tengah – kiri adalah 180°.

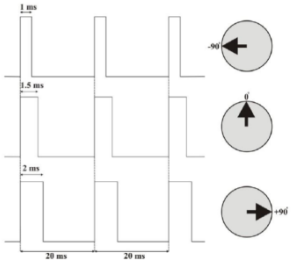
b. Motor servo *continous*

   Motor servo jenis ini mampu bergerak dua arah, yaitu searah jarum jam dan kebalikan dari arah jarum jam tanpa batasan defleksi sudut putar (dapat berputar secara kontinyu).

Dari kedua jenis motor servo tersebut, metoda PWM dapat digunakan untuk menentukan posisi sudut motor, hanya saja terdapat perbedaan pada besarnya arah puataran motor.

Pada motor servo standard hanya dapat bergerak sebesar 180° yaitu dari 0°-180° atau (-90°)-90°. Secara umum untuk mengakses motor servo tipe standard adalah dengan cara memberikan pulsa *high* selama 1,5 ms dan mengulangnya setiap 20 ms, maka posisi servo akan berada ditengah atau netral (0°). Untuk pulsa 1 ms maka akan bergerak berkebalikan arah jarum jam dengan sudut -90°. Dan

pulsa *high* selama 2 ms akan bergerak searah jarum jam sebesar 90° seperti terlihat pada gambar 2.13.

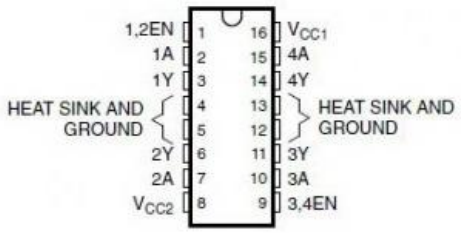


Gambar 2.13 Sinyal Kontrol Motor Servo

Pengaturan pergerakan motor servo *continous* tidak jauh berbeda dengan pengaturan pengendalian motor servo standard, secara umum untuk berputar (rotasi) searah jarum jam harus diberi pulsa *high* selama 1,3 ms. Sedangkan untuk berputar berlawanan arah jarum jam harus diberi logika *high* selama 1,7 ms. Jika motor servo *continous* diberi pulsa *high* selama 1,5 ms maka akan berhenti. Pin signal pada motor servo dapat dikoneksi ke pin mikrokontroler sebagai pengendali dari pergerakan motor servo ini.

2.7 Driver Motor L293D

IC L293D adalah IC yang didesain khusus sebagai *driver* motor DC dan dapat dikendalikan dengan rangkaian TTL maupun mikrokontroler. Motor DC yang dikontrol dengan *driver* IC L293D dapat dihubungkan ke g*round* maupun ke sumber tegangan positif karena di dalam *driver* L293D sistem *driver* yang digunakan adalah *totem pool*. Dalam 1 unit *chip* IC L293D terdiri dari 4 buah driver motor DC yang berdiri sendiri sendiri dengan kemampuan mengalirkan arus 1 Ampere tiap drivernya. Sehingga dapat digunakan untuk membuat *driver* H-*bridge* untuk 2 buah motor DC. Konstruksi pin *driver* motor DC IC l293D seperti pada Gambar 2.14.



Gambar 2.14 *Driver* Motor L293D

Fungsi Pin:

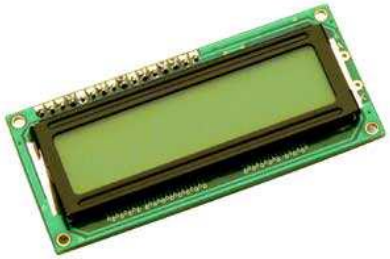
- Pin EN (Enable, EN1.2, EN3.4) berfungsi untuk mengijinkan *driver* menerima perintah untuk menggerakan motor DC.
- Pin In (Input, 1A, 2A, 3A, 4A) adalah pin *input* sinyal kendali motor DC.
- Pin Out (Output, 1Y, 2Y, 3Y, 4Y) adalah jalur *output* masing-masing *driver* yang dihubungkan ke motor DC.
- Pin VCC (VCC1, VCC2) adalah jalur *input* tegangan sumber *driver* motor DC, dimana VCC1 adalah jalur *input* sumber tegangan rangkaian *dirver* dan VCC2 adalah jalur *input* sumber tegangan untuk motor DC yang dikendalikan.
- Pin GND (Ground) adalah jalur yang harus dihubungkan ke *ground*, pin GND ini ada 4 buah yang berdekatan dan dapat dihubungkan ke sebuah pendingin kecil.

Driver motor DC IC L293D ini memiliki fungsi yang lengkap untuk sebuah *driver* motor DC sehingga dapat diaplikasikan dalam beberapa teknik *driver* motor DC dan dapat digunakan untuk mengendalikan beberapa jenis motor DC.

2.8 Liquid Crystal Display (LCD)

Liquid crystal display (LCD) adalah satu layar bagian dari modul peraga yang menampilkan karakter yang diinginkan. Layar LCD menggunakan 2 buah lembaran bahan yang dapat mempolarisasikan dan kristal cair diantara kedua lembaran tersebut. Arus listrik yang melewati cairan menyebabkan kristal merata sehingga cahaya tidak dapat melalui setiap kristal. Sehingga dapat mengubah bentuk kristal cairnya membentuk tampilan angka atau huruf pada layar.

Dalam tugas akhir ini LCD digunakan untuk menampilkan jarak antara mobil dengan hambatan di depannya, sebagai indikator kecepatan, sebagai indikator nilai PWM, dan sebagai indikator nilai persentase rem hasil perhitungan logika *fuzzy* . Bentuk LCD 16x2 ini dapat dilihat pada Gambar 2.15.

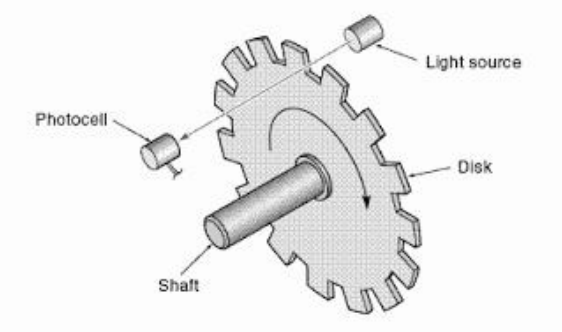


Gambar 2.15 16x2 *Character* LCD *Module*

2.9 *Rotary Encoder*

Rotary encoder adalah perangkat elektromekanik yang dapat memonitor gerakan dan posisi. *Rotary encoder* umumnya menggunakan sensor optik untuk menghasilkan serial pulsa yang dapat diartikan menjadi gerakan, posisi, dan arah. Sehingga posisi sudut suatu poros benda berputar dapat diolah menjadi informasi berupa kode digital oleh *rotary encoder* untuk diteruskan oleh rangkaian kendali.

Rotary encoder tersusun dari suatu piringan tipis yang memiliki lubang-lubang pada bagian lingkaran piringan. LED ditempatkan pada salah satu sisi piringan sehingga cahaya akan menuju ke piringan. Di sisi yang lain suatu photo-transistor diletakkan sehingga photo-transistor ini dapat mendeteksi cahaya dari LED yang berseberangan. Piringan tipis tadi dikopel dengan poros motor, atau perangkat berputar lainnya yang ingin kita ketahui posisi atau kecepatannya, sehingga ketika motor berputar piringan juga akan ikut berputar. Apabila posisi piringan mengakibatkan cahaya dari LED dapat mencapai photo-transistor melalui lubang-lubang yang ada, maka photo-transistor akan mengalami saturasi dan akan menghasilkan suatu pulsa. Semakin banyak deretan pulsa yang dihasilkan pada satu putaran menentukan akurasi *rotary encoder* tersebut, akibatnya semakin banyak jumlah lubang yang dapat dibuat pada piringan menentukan akurasi *rotary encoder* tersebut. Cara kerja dari *rotary encoder* dapat dilihat pada Gambar 2.16.



Gambar 2.16 Cara Kerja *Rotary Encoder*

Untuk mengetahui kecepatan putar dari pulsa yang dihasilkan oleh *rotary encoder* tersebut, maka dapat dituliskan dalam persamaan berikut:

$$V = \frac{f}{M} \times 2\pi r \quad \text{.....(2.13)}$$

dengan,

$\boldsymbol{V}$ = kecepatan benda (m/s)

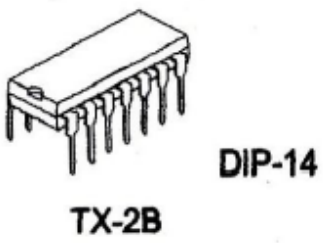
$\boldsymbol{r}$ = jari-jari (m)

$\boldsymbol{f}$ = jumlah pulsa dalam 1 detik

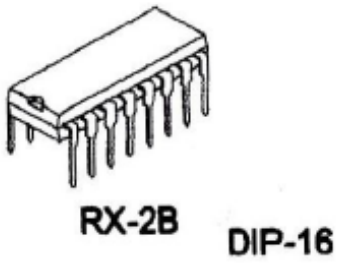
$\boldsymbol{M}$ = jumlah lubang piringan *rotary encoder*

2.10 *Remote Control* (TX – RX)

Remote Control gelombang radio terdiri dari 2 bagian, yaitu pesawat pemancar (TX) dan pesawat penerima (RX). Pada rangkaian pesawat pemancar menggunakan IC TX-2B dan pada rangkaian pesawat penerima menggunakan IC RX-2B. Sepasang IC ini memiliki 5 fungsi kerja yang biasanya digunakan untuk mengoperasikan mobil-mobilan dengan *remote control.* Pada *remote control* fungsi ini digunakan supaya mobil bergerak maju-mundur, berbelok ke kiri, dan berbelok ke kanan. *Output* IC TX-2B diletakkan sebagai *input* bagi pesawat pemancar. Sedangkan IC RX-2B diletakkan sebagai penerjemah di pesawat penerima. Bentuk fisik IC TX-2B dan IC RX-2b dapat dilihat pada Gambar 2.17 dan Gambar 2.18.

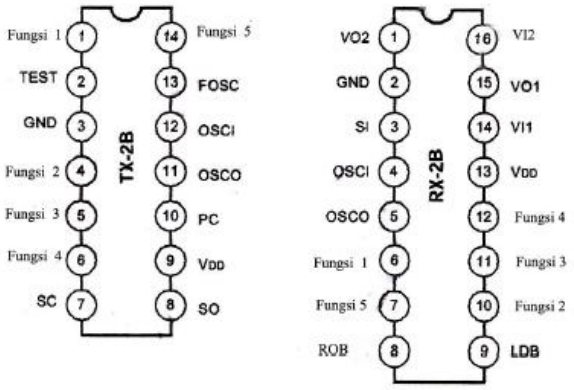


Gambar 2.17 Bentuk Fisik IC TX-2B



Gambar 2.18 Bentuk Fisik IC RX-2B

IC TX-2B memiliki 14 pin, sedangkan IC RX-2B memiliki 16 pin. Setiap pin fungsi pada IC TX-2B berpasangan dengan pin fungsi pada IC RX-2B. Untuk konfigurasi fungsi pin IC TX-2B dan RX-2B dapat dilihat pada Gambar 2.19.



Gambar 2.19 Konfigurasi Fungsi Pin-Pin IC TX-2B dan RX-2B

IC TX-2B mempunyai karakteristik *supply* tegangan sebesar 0,3 – 5 ,0 volt. Tegangan operasional antara 1,5 – 5 volt. Arus maksimal yang mengalir 2 mA. Sedanglan IC RX-2B memiliki tegangan operasional antara 1,5 – 5,0 volt dan arus operasional maksimum 3,0 mA. Fungsi dari setiap pin IC TX-2B terdapat pada Tabel 2.5.

Tabel 2.5 Pin-Pin IC TX-2B

Pin	Simbol	Keterangan
1	Fungsi 1	Jika terhubung GND, mengaktifkan F 1 di RX
2	Test B	Dipakai untuk menguji IC
3	GND	Untuk dihubungkan ke negatif
4	Fungsi 2	Jika terhubung ke GND, mengaktifkan F 2 di RX
5	Fungsi 3	Jika terhubung ke GND, mengaktifkan F 3 di RX
6	Fungsi 4	Jika terhubung ke GND, mengaktifkan F 4 di RX
7	SC	Sinyal terkode dengan frekuensi pembawa
8	SO	Sinyal terkode tanpa frekuensi pembawa
9	VDD	Untuk dihubungkan ke positif *power supply*
10	PC	Dihubungkan ke LED sebagai indikator output
11	OSC0	Kaki keluaran Osilator
12	OSC1	Kaki masukan Osilator
13	FOSC	Dipakai untuk menguji IC
14	Fungsi 5	Jika terhubung ke GND, mengaktifkan F 5 di RX

Cara kerja IC TX-2B adalah bila pin fungsi 1 pada IC TX-2B terhubung dengan GND (sumber negatif), maka akan keluar sinyal dari pin fungsi 1 pada IC RX-2B. Fungsi dari setiap pin IC TX-2B terdapat pada Tabel 2.6.

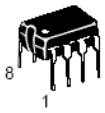
Tabel 2.6 Pin-Pin IC RX-2B

Pin	Simbol	Keterangan
1	VO 2	*Inverter* pin 2 *output* untuk penguat daya
2	GND	Untuk dihubungkan ke negatif (*ground*)
3	SI	Pin input sinyal terkode
4	OSC1	Kaki masukan osilator
5	OSC0	Kaki keluaran osilator
6	Fungsi 1	Aktualisasi F1 dari TX
7	Fungsi 5	Aktualisasi F5 dari TX
8	ROB	Jika dihubungkan ke GND, menonaktifkan pin 6
9	LDB	Jika dihubungkan ke GND, menonaktifkan pin 7
13	Fungsi 2	Aktualisasi F2 dari TX
11	Fungsi 3	Aktualisasi F3 dari TX
12	Fungsi 4	Aktualisasi F4 dari TX
13	VDD	Untuk dihubungkan ke positif *power supply*
14	VI 1	*Inverter* pin 1 *input* untuk penguat daya
15	VO 1	*Invrter* pin 1 *output* untuk penguat daya
16	VI 2	*Inverter* pin 2 *input* untuk penguat daya

2.11 Komparator

Komparator secara harfiah berarti membandingkan, dalam hal ini adalah membandingkan dua macam tegangan pada kedua masukannya. Dalam prakteknya tegangan yang satu dicatu oleh suatu acuan (*reference)* yang besarnya tetap, sedangkan yang lainnya oleh suatu masukan tegangan yang variable, bilamana tersambung (*on*). Apabila masukan *non inverting* lebih positif dari masukan *inverting*-nya, maka akan diperoleh keluaran maksismum positif. Dan bilamana masukan *inverting* lebih positif dibandingkan masukan *non invertingnya*, maka akan diperoleh keluaran maksimum negatif.

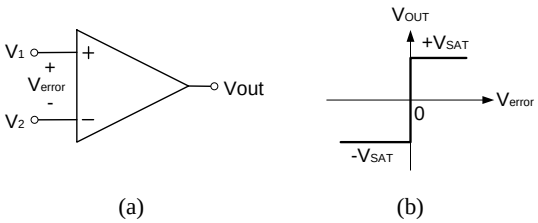
Komparator adalah suatu penguat operasional (opamp). Penguat operasional adalah penguat dengan gain tinggi dan terhubung secara langsung. Penguat ini biasanya digunakan untuk menguatkan sinyal berjalur frekuensi lebar dan digunakan bersama jaringan umpan balik eksternal. Penguat operasional memiliki 5 terminal dasar, yaitu 2 untuk memberikan daya, 2 untuk masukan dan satu untuk keluaran. Bentuk Fisik IC LM393N dapat dilihat pada Gambar 2.20.



Gambar 2.20 Bentuk Fisik IC LM393N

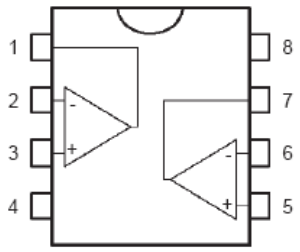
Cara yang termudah untuk menggunakan suatu penguat operasional adalah *loop* terbuka (tidak ada resistor umpan balik), seperti ditunjukkan dalam gambar 2.21(a) karena penguat yang tinggi dari penguat operasional tegangan kesalahannya sedikit (secara tipikal dalam mikrovolt) menimbulkan ayunan

(*swing*) *output* maksimum. Misalnya, jika V_1 lebih besar daripada V_2, tegangan kesalahan adalah positif dan tegangan *output* menuju harga positif maksimum secara tipikal 1 sampai 2 V kurang dari tegangan catu. Jika V_1 lebih kecil dari V_2, tegangan *output* berayun ke tegangan negatif maksimum.



Gambar 2.21 *OpAmp*

Gambar 2.21(b) meringkaskan gerak tersebut. Tegangan kesalahan positif mendorong *output* ke $+V_{SAT}$ harga maksimum dari tegangan *output*. Tegangan kesalahan negatif menimbulkan tegangan *output* $–V_{SAT}$. Jika sebuah penguat operasional digunakan seperti ini, maka disebut *komparator* karena semua yang dapat dilakukannya adalah membandingkan V_1 dan V_2 yang menghasilkan *output* positif atau negatif jenuh, tergantung pada apakah V_1 lebih besar atau lebih kecil daripada V_2. Konfigurasi pin IC LM393N dapat dilihat pada Gambar 2.22. Untuk fungsi dari setiap pin IC LM393N terdapat pada Tabel 2.7.



Gambar 2.22 Konfigurasi Pin IC LM393N

Komparator yang digunakan dalam tugas akhir ini adalah LM 393N. IC ini mempunyai kemampuan untuk menghasilkan dua keluaran (*output*) untuk dibaca oleh mikrokontroler.

Tabel 2.7 Fungsi Pin-Pin IC LM393N

Pin	Keterangan
1	Output 1
2	Inverting Input 1
3	Non-Inverting Input 1
4	Gnd
5	Non-Inverting Input 2
6	Inverting Input 2
7	Output 2
8	Vcc

2.12 Pemrograman Bahasa C

Bahasa pemrograman yang digunakan dalam tugas akhir ini adalah program Bahasa C. Bahasa pemrograman Bahasa C dikenal sebagai bahasa pemrograman handal, cepat, mudah, dan tergolong ke dalam bahasa pemrograman tingkat menengah. Konstruksi dari program bahasa C harus mengikuti aturan sebagai berikut.

a. Komentar

   Komentar digunakan untuk memberi keterangan pada program agar mudah dibaca dan akan diabaikan oleh *compiler*.

   Contoh penulisan :

   // hanya satu ini sebelum enter yang dianggap komentar.

   * kata-kata yang berbeda dalam tanda garis miring dengan bintang dianggap komentar berapa pun panjangnya dan berapa pun barisnya*/.

b. *Preprocessor*

Biasanya digunakan untuk menyertakan file header (.h) atau file *library*. File *include* berguna untuk memberitahu *compiler* agar membaca file yang di *include*- kan lebih dahulu agar mengenali definisi-definisi yang digunakan dalam program sehingga tidak dianggap *error*.

Cara penulisan:

#include <…….> untuk lokasi standar file yang telah disetting oleh *tools* biasanya pada *folder include* atau *folder directori complier.*

#include"……." Untuk lokasi file yang yang kita tentukan sendiri.

File *header* io.h adalah file yang segala informasi/definisi tentang register-register fungsi khusus (SFR) dan bit-bit atau pin-pin mikrokontroler AVR.

c. Pengenal (*Identifier*)

Pengenal digunakan untuk memberi nama variabel, konstanta, dll. Bahasa C bersifat *case sensitive* (huruf kapital dan huruf kecil dianggap berbeda). Konstruksi pengenal adalah huruf, angka, garis bawah (_). Tiap pengenal bisa menggunakan gabungan ketiga hal tersebut dengan catatan tidak boleh diawali dengan angka.

Cara penulisan:

Menit //benar

MeniT //benar dan berbeda dengan pengenal Menit

60detik //salah

_60detik //benar

d. Variabel

Variabel adalah tempat untuk menyimpan dan mengakses data yang mewakili memori dalam mikrokontroler. Variabel harus dideklarasikan (memberitahu kompiler) dengan tipe data beserta nama variabel yang akan digunakan.Tiap tipe data mempunyai jangkauan bilangan yang dapat disimpan, hal ini akibat dari *byte* memori yang dipesan dan bentuk bilangan bertanda atau tidak. Seperti pada Tabel 2.8 berisi jenis tipe data dan banyaknya bilangan yang dapat ditampung.

Tabel 2.8 Tipe Data

Tipe Data	*byte*	bit	Minimum	Maksimum
Char	1	8	-128	127
Signed char	1	8	-128	127
Unsigned char	1	8	0	255
Int	2	16	-32768	32767
Signed int	2	16	-32768	32767
Unsigned int	2	16	0	65535
Long	3	32	-2147483648	2147483647
Signed long	3	32	-2147483648	2147483647
Unsigned long	3	32	0	4294967295
Float	3	32	1.28E-38	3.4E38

e. Operator

Operator adalah karakter-karakter khusus untuk memanipulasi variable. *Operand* adalah variabel atau konstanta yang merupakan bagian dari pernyataan.

Aritmatika :

- + adalah penjumlahan.
- - adalah pengurangan.
- * dalah perkalian.

- / adalah pembagian.
- ++ adalah increment.
- -- adalah decrement.

Logika :

- == adalah logika sama dengan.
- != adalah logika tidak sama dengan.
- < adalah logika lebih kecil.
- <= adalah logika lebih kecil sama dengan.
- > adalah logika lebih besar.
- >= adalah logika lebih besar sama dengan.
- ! adalah logika not.
- && adalah logika AND.
- || adalah logika OR.

Manipulasi Bit :

- ~ adalah mengkomplementkan.
- & adalah mengANDkan.
- | adalah mengORkan.
- ^ adalah mengXORkan.
- << adalah shift left.
- >> adalah shift right.

f. Kontrol Aliran Program

Untuk mencerdaskan sebuah program, maka diperlukan algoritma dimana terdiri dari berbagai pengendalian aliran program.

- If (...){…}

  Digunakan untuk mengecek satu kondisi satu blok jawaban.

- If (…) {…} else {…}

  Digunakan untuk mengecek satu kondisi dua blok jawaban.

- If (…) {…} else if (…) {…} else {…}

  Digunakan untuk mengecek beberapa kondisi yang berkaitan.

- While (…) {…}

  Digunakan untuk perulangan/looping/iterasi jika kondisi yang diuji bernilai benar.

- Do {…} while (…)

  Digunakan untuk perulangan/looping/iterasi jika kondisi yang diuji bernilai benar. Perbedaanya dengan *while* tanpa *do* adalah blok di eksekusi terlebih dahulu baru diuji, hal ini dapat terjadi kemungkinan yang diuji salah namun blok tetap di eksekusi.

- For (… ; … ; … ;…) {…}

  Digunakan untuk perulangan/looping/iterasi jika kondisi dan syarat yang ditentukan.