BAB II

LANDASAN TEORI

Bab ini berisi tentang teori mengenai permasalahan yang dibahas dalam tugas akhir ini secara garis besar dimulai dari definisi logika fuzzy, mikrokontroler ATMega-16, sensor ultrasonik ping, *motor DC*, *drivermotor*, *Liquid Crystal Display* (LCD), *buzzer*, *motor servo* sampai dengan perangkat lunak (*software*) Bahasa C sebagai pemrogram.

2.1. Logika Fuzzy

Teori himpunan logika fuzzy dikembangkan oleh Prof. Lofti Zadeh pada tahun 1965. Ia berpendapat bahwa logika benar dan salah dari logika Boolean tidak dapat mengatasi masalah gradasi yang berada pada dunia nyata. Untuk mengatasi masalah gradasi yang tidak terhingga tersebut, Zadeh mengembangkan sebuah himpunan fuzzy.

Tidak seperti logika Boolean, logika fuzzy mempunyai nilai yang kontinu. Fuzzy dinyatakan dalam derajat dari suatu keanggotaan dan derajat dari kebenaran. Oleh karena sebab itu sesuatu dapat dikatakan sebagian benar dan sebagian salah pada waktu yang sama. Logika fuzzy adalah suatu cara yang tepat untuk memetakan suatu ruang masukan *(input)* ke dalam suatu ruang keluaran *(output)*.

2.1.1. Alasan Penggunaan Logika Fuzzy

Adapun beberapa alasan dalam penggunaan logika fuzzy pada perancangan ini adalah:

1. logika fuzzy memiliki toleransi terhadap data-data yang tidak tepat,
2. logika fuzzy mampu memodelkan fungsi-fungsi non-linier yang sangat kompleks,
3. logika fuzzy dapat bekerja sama dengan teknik-teknik kendali cerdas secara konvensional,
4. pemakaian fungsi keanggotaan memungkinkan fuzzy untuk melakukan observasi obyektif terhadap nilai-nilai yang subyektif. Selanjutnya fungsi keanggotaan ini dapat dikombinasikan untuk membuat pengungkapan konsep yang lebih jelas,
5. beroperasi tanpa campur manusia manusia secara langsung, tetapi memiliki efektivitas yang sama dengan pengendali manusia.

2.1.2. Himpunan Fuzzy

Pada himpunan tegas (crisp), nilai keanggotaaan suatu item x dalam suatu himpunan A, yang sering ditulis dengan μA [x], memiliki dua kemungkinan, yaitu:

- satu (1), yang berarti bahwa suatu item menjadi anggota dalam suatu himpunan, atau
- nol (0), yang berarti bahwa suatu item tidak menjadi anggota dalam suatu himpunan.

Pada penggunaan himpunan crisp, adanya perubahan kecil saja pada suatu nilai mengakibatkan perbedaan kategori yang cukup signifikan. Himpunan fuzzy digunakan untuk mengantisispasi hal tersebut. Seberapa besar ekstensinya dalam himpunan tersebut dapat dilihat pada nilai keanggotaannya. Kalau pada himpunan crisp, nilai keanggotaan hanya ada dua kemungkinan, yaitu 0 atau 1, pada himpunan fuzzy nilai keanggotaan terletak pada rentang 0 sampai 1. apabila nilai keanggotaan fuzzy μ A[x] = 0 berarti x tidak menjadi anggota himpunan A, demikian pula apabila x memiliki nilai keanggotaan μA[x] = 1 berarti x menjadi anggota penuh himpunan A. Himpunan fuzzy memiliki dua atribut, yaitu :

1. linguistik, yaitu penamaan suatu grup yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alami, seperti : MUDA, PAROBAYA, TUA.
2. numeris, yaitu suatu nilai (angka) yang menunjukkan ukuran dari suatu variabel seperti 40, 25, 30.

Ada beberapa hal yang perlu diketahui dalam memahami sistem fuzzy, yaitu.

1. Variabel fuzzy

   Variabel fuzzy merupakan variabel yang hendak dibahas dalam suatu sistem fuzzy. Contoh: umur, temperatur, permintaan.

2. Himpunan fuzzy

   Himpunan fuzzy merupakan suatu grup yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel fuzzy. Contoh: Variabel umur, terbagi menjadi 3 himpunan fuzzy, yaitu: muda, parobaya, tua.

3. Semesta pembicaraan

Semesta pembicaraan adalah keseluruhan nilai yang diperbolehkan untuk dioperasikan dalam suatu variabel fuzzy. Contoh: Semesta pembicaraan untuk variabel umur: [0 + ~].

4. Domain

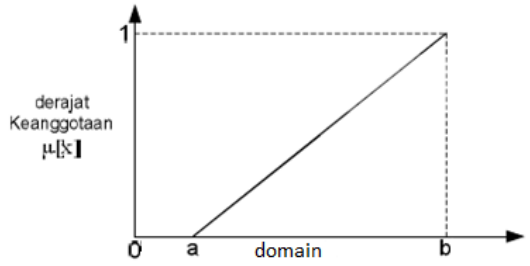
Domain himpunan fuzzy adalah keseluruhan nilai yang diijinkan dalam semesta pembicaraan dan boleh dioperasikan dalam suatu himpunan fuzzy. Contoh: MUDA = [0, 45] ,TUA = [45, + ~].

2.1.3. Fungsi Keanggotaan

Fungsi keanggotaan *(membership function)* adalah suatu kurva yang menunjukkan pemetaan titik-titik input data ke dalam nilai keanggotaannya (derajat keanggotaan). Ada beberapa fungsi yang digunakan pada perancangan ini, antara lain.

1. Representasi Linier

Pada pemetaan linear, pemetaan input ke derajat keanggotaannya digambarkan sebagai suatu garis lurus. Ada dua keadaaan himpunan fuzzy yang linear. Pertama, kenaikan himpunan dimulai pada nilai domain yang memiliki derajat keanggotaan nol [0] bergerak ke kanan menuju ke nilai domain yang memiliki derajat keanggotaan lebih tinggi. Gambar 2.1 dibawah ini menunjukan grafik dari kurva linier naik.

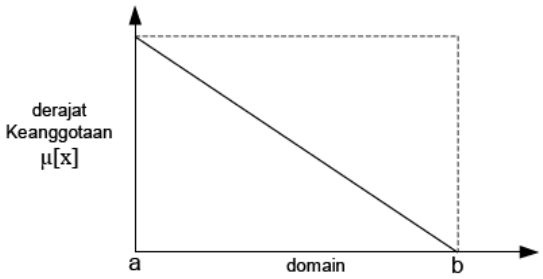


Gambar 2.1 Kurva Linear Naik

Fungsi Keanggotaan :

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

Kedua, merupakan kebalikan yang pertama. Garis lurus dimulai dari nilai domain dengan derajat keanggotaan tertinggi pada sisi kiri, kemudian bergerak menurun ke nilai domain yang memiliki derajat keanggotaan lebih rendah. Gambar 2.2 dibawah ini menunjukan grafik dari kurva linier turun.



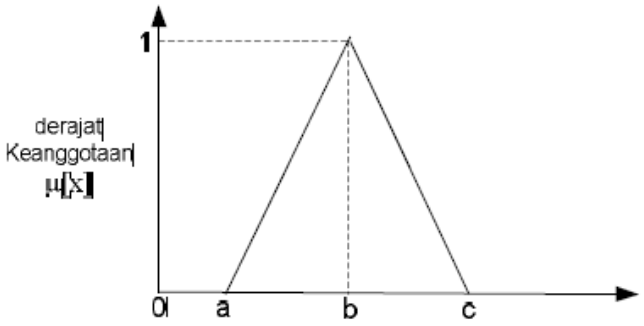
Gambar 2.2 Kurva Linear Turun

Fungsi keanggotaan :

$$\mu[x] = \begin{cases} 0 & ; x \geq b \\ \frac{(b-x)}{(b-a)} & ; a \leq x \leq b \\ 1 & ; x \leq a \end{cases} \text{.....(pers.2)}$$

2. Representasi Kurva Segitiga

Kurva segitiga pada dasarnya merupakan gabungan antara dua garis linear. Gambar 2.3 dibawah ini menunjukan grafik dari kurva segitiga.



Gambar 2.3 Kurva Segitiga

Fungsi Keanggotaan :

$$\mu[x] = \begin{cases} 0 \,; x \leq a \\ \frac{(x-a)}{(b-a)} \;; a \leq x \leq c \\ \frac{(c-x)}{(c-b)} \;; b \leq x \leq c \\ 0 \,; c \leq x \end{cases} \text{.....(pers.3)}$$

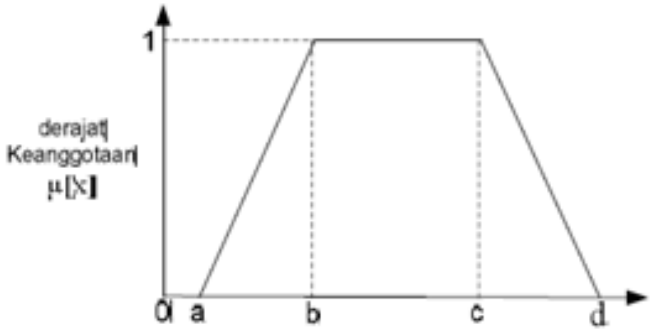
Untuk menentukan nilai minimum dan maximum fungsi keanggotaan pada representasi kurva segitiga, dapat menggunakan persamaan dibawah ini.

$$\mu[x] = max\left(min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right) \text{.....(pers.4)}$$

Parameter {a,b,c} dengan (a < b < c) menentukan sudut x dari tiga sudut penting dari fungsi keanggotaan.

3. Representasi Kurva Trapesium

Kurva trapesium pada dasarnya seperti bentuk segitiga, hanya saja ada beberapa titik yang memiliki nilai keanggotaan satu. Gambar 2.4 dibawah ini menunjukan grafik dari kurva trapesium.



Gambar 2.4 Kurva Trapesium

Fungsi Keanggotaan :

$$\mu[x] = \begin{cases} 0 & ; x \leq a \\ \frac{(x-a)}{(b-a)} & ; a \leq x \leq b \\ 1 & ; b \leq x \leq c \\ \frac{(d-x)}{(d-c)} & ; c \leq x \leq d \\ 0 & ; d \leq x \end{cases} \text{ ……..……...…(pers. 5)}$$

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

$$\mu[x] = max\left(min\left(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c}\right), 0\right) \text{.....(pers.6)}$$

Parameter {a,b,c,d} dengan (a < b < c < d) menentukan sudut x dari empat sudut penting dari fungsi keanggotaan.

2.1.4. Operator Dasar 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.

1. Operator AND

   Operator ini berhubungan dengan operasi interseksi pada himpunan. A predikat sebagai hasil operasi dengan operator AND diperoleh dengan mengambil nilai keanggotaan terkecil antara elemen pada himpunan-himpunan yang bersangkutan.

   $$\mu_{AnB} = \text{Min}\ (\mu_A [x], \mu_B [y] \quad \text{(pers. 7)}$$

2. Operator OR

   Operator ini berhubungan dengan operasi union pada himpunan. α - predikat sebagai hasil operasi dengan operator OR diperoleh dengan mengambil nilai keanggotaan terbesar antara elemen pada himpunan-himpunan yang bersangkutan.

   $$\mu_{AuB} = \text{Max}\ (\mu_A [x], \mu_B [y]) \quad \text{(pers. 8)}$$

3. Operator NOT

   Operator ini berhubungan dengan operasi komplemen himpunan. α – predikat sebagai hasil operasi dengan operator NOT diperoleh dengan mengurangkan nilai keanggotaan elemen pada himpunan dengan 1.

   $$\mu_N = 1-\mu_A [x] \quad \text{(pers. 9)}$$

2.1.5. Fuzzy Inference System

Salah satu metode untuk penalaran *(Fuzzy Inference System)* adalah Metode Mamdani (Max-Min). Metode ini diperkenalkan oleh Ebrahim Mamdani pada tahun 1975. Untuk mendapatkan output, diperlukan empat tahap. Dapat digambarkan dengan block dibawah ini.

1. Pembentukan himpunan Fuzzy

   Pada Metode Mamdani, baik variabel input maupun variabel output dibagi menjadi satu atau lebih himpunan fuzzy.

2. Fuzzyfikasi *(Fuzzification)*

   Proses memetakan nilai tegas input *(crisp)* kedalam himpunan Fuzzy. Hasil dari proses ini berupa fuzzy input.

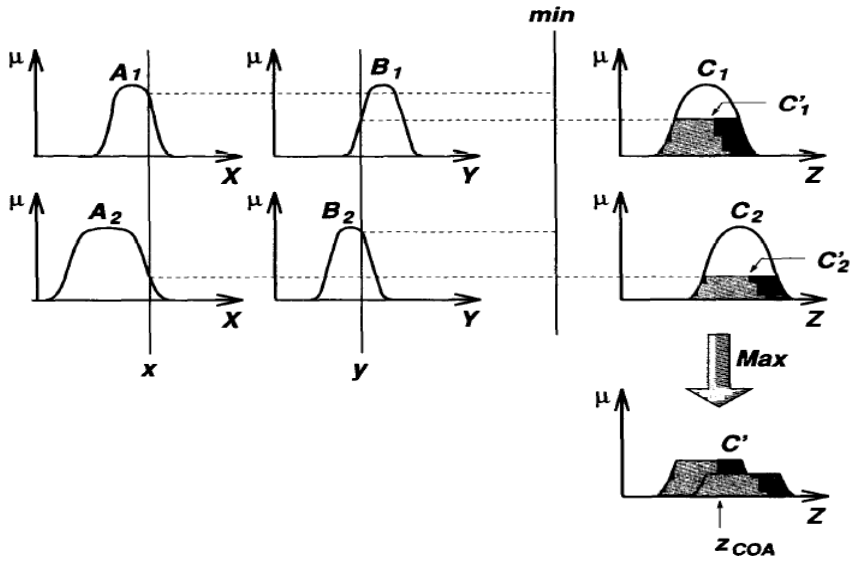
3. Rule evaluasi *(Rule evaluation)*

   Proses melakukan penalaran terhadap fuzzy input yang dihasilkan oleh proses *fuzzification* berdasarkan aturan fuzzy yang telah dibuat. Proses ini menghasilkan fuzzy output. Fungsi Implikasi yang digunakan adalah Min.

4. Penegasan *(Defuzzification)*

   Input dari proses penegasan *(defuzzification)* adalah suatu himpunan fuzzy yang diperoleh dari komposisi aturan-aturan fuzzy, sedangkan output yang dihasilkan merupakan suatu bilangan pada domain himpunan fuzzy tersebut. Sehingga jika diberikan suatu himpunan fuzzy dalam range tertentu, maka harus dapat diambil suatu nilai *crisp* tertentu sebagai output. Ada banyak metode penegasan *(defuzzification)* yang dapat digunakan untuk aturan Mamdani, salah satunya dengan mengunakan metode *centroid of area.*

$$^{Z}\mathrm{COA} = \frac{\int z\mu A(z) z\, dz}{\int z\mu A(z) z\, dz} \text{.....(pers.10)}$$

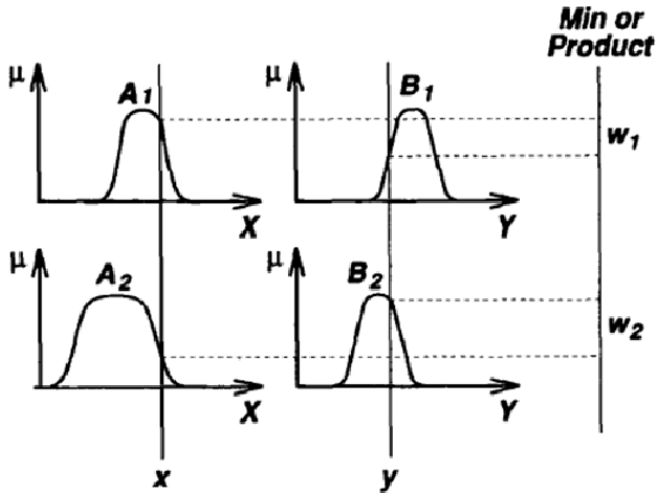


Gambar 2.5 *Fuzzy Inference System* (Mamdani)

Salah satu alternatif metode untuk penalaran *(Fuzzy Inference System)* adalah dengan menggunakan metode penalaran fuzzy metode Sugeno hampir sama dengan penalaran Mamdani, hanya saja output sistem tidak berupa himpunan fuzzy, melainkan berupa konstanta atau persamaan linier. Metode ini diperkenalkan oleh Takagi-Sugeno Kang pada tahun 1985.

Pada metode Sugeno, fuzzifikasi, operasi fuzzy, dan implikasi sama seperti metode Mamdani, perbedaannya hanya pada agregasi dan defuzzifikasi. Jika pada metode Mamdani agregasi berupa daerah di bawah kurva, maka pada metode Sugeno agregasi berupa singleton. Defuzifikasi pada metode sugeno lebih sederhana, karena hanya menghitung titik rata-rata.

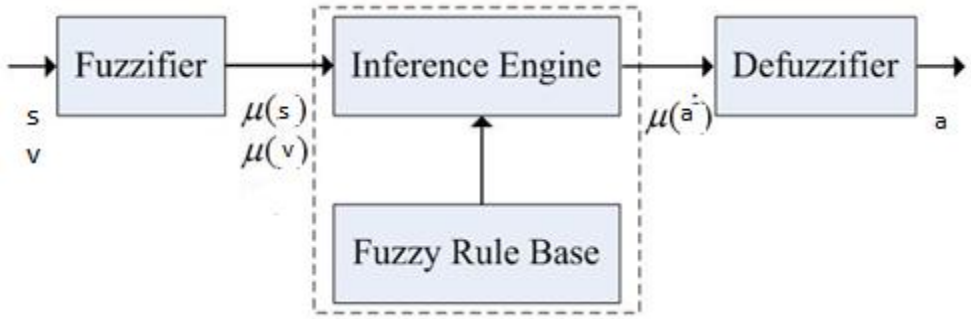
$$Z = \frac{W1Z1+W2Z2}{W1+W2} \text{.....(pers.11)}$$



Gambar 2.6 *Fuzzy Inference System* (Sugeno)

2.2. Pengereman Berbasis Kontrol Logika Fuzzy

Sebuah kontrol logika fuzzy untuk pengereman mempunyai dua variabel masukan s, v dan satu variabel keluaran a. Struktur kontrol logika fuzzy ditunjukkan pada Gambar 2.7. Aturan fuzzy dasar terdiri dari beberapa aturan dengan sebelumnya dan model inferensi fuzzy mamdani digunakan untuk menentukan nilai output, sedangkan defuzzifikasi dilakukan dengan hasil percepatan. Gambar 2.7 menunjukkan fungsi keanggotaan fuzzy yang dipakai.

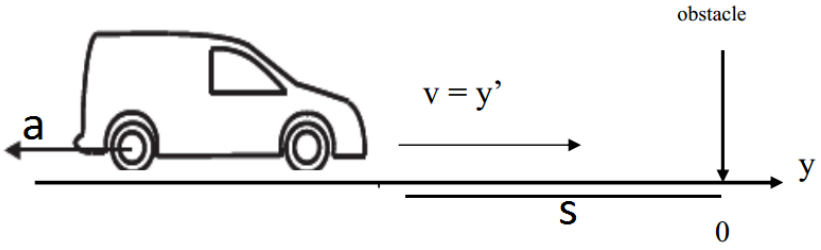


Gambar 2.7 Struktur Kontrol Fuzzy

2.3. Permasalahan Dalam Pengereman Otomatis

Pada perancangan ini pengereman mobil ditentukan oleh 2 variabel s, v. Dimana variabel s merupakan jarak pada rintangan *(obstacle),* dan untuk v merupakan sebagai kecepatan dimana mobil akan melaju. Sedangkan output dari perancangan ini yaitu a, percepatan digunakan untuk mengontrol pengereman mobil tersebut agar bisa mencapai pengereman yang ditentukan secara otomatis.

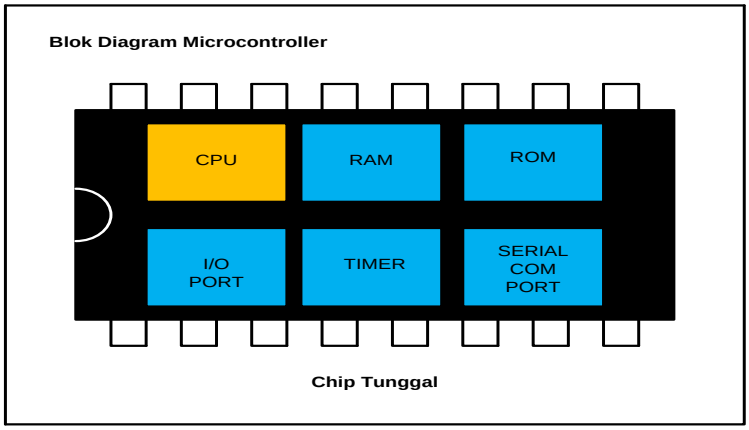
Mobil akan melakukan pengereman dan penurunkan kecepatan dari peraturan (*rules)* dan fungsi keanggotaan yang telah ditentukan**.** Sehingga akan menghasilkan kontrol pengereman mobil dengan rintangan (*obstacle)* depan mobil secara otomatis, agar mobil bisa mencapai pengereman yang telah ditentukan.



Gambar 2.8 Perancangan Pengereman Mobil

2.4. Mikrokontroler AVR ATmega16

Mikrokontroler adalah istilah yang merujuk pada *central processing unit* (CPU) komputer digital untuk tujuan umum. Untuk membuat sistem komputer, CPU harus ditambahkan *memory*, umumnya *read only memory* (ROM) dan *randomaccess memory* (RAM), *decoder memory*, *oscillator* dan sejumlah *input-output* (I/O) seperti *port* data *paralel* dan *serial*.



Gambar 2.9 Blok Diagram *Chip* Mikrokontroler

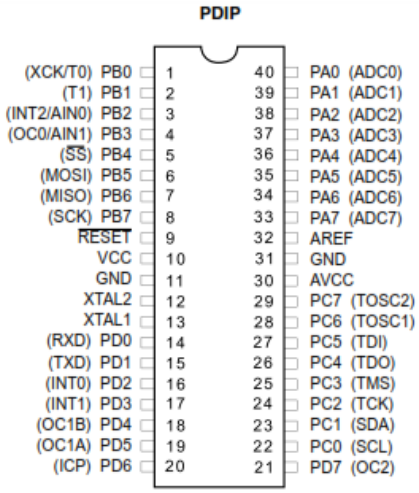
Pada gambar diatas ditunjukkan blok diagram dari mikrokontroler yang khas, yang merupakan sebuah sistem komputer dalam bentuk *chip* tunggal. Desain tersebut menggabungkan semua fitur yang diperoleh dari sebuah sistem komputer yaitu *central processing unit* (CPU) dan fitur-fitur lain yang dibutuhkan untuk membuat sebuah komputer yang lengkap yaitu: ROM, RAM, I/O *Port*, Serial I/O, Timer dan *Clock Circuit*.

Seperti halnya mikroprosesor, mikrokontroler juga adalah *general purpose device*, tetapi hanya difungsikan untuk membaca data dan melakukan kalkulasi terbatas pada data serta mengendalikan lingkungannya berdasarkan kalkulasi tersebut. Penggunaan utama mikrokontroler adalah untuk mengontrol operasi sebuah mesin yang menggunakan program yang tetap yang disimpan di dalam ROM dan tidak akan berubah sepanjang umur sistem tersebut.

Jenis mikrokontroler yang digunakan dalam perancangan ini ialah tipe ATmega16. Pemilihan mikrokontroler ini didasarkan pada kebutuhan penggunaan. Di dalam sebuah *chip* mikrokontroler ATmega16 terdapat fitur utama diantaranya sebagai berikut.

- 16K *Bytes Flash Memory*

- 1K *Byte Internal SRAM*
- 512 *bytes* EEPROM
- *Port* I/O 32 *bit*, yang dikelompokkan dalam PortA, PortB, PortC dan PortD.
- 8 bit *timer/counter*.
- 16 bit *timer/counter*.
- Terdapat fasilitas JTAG.
- 4 *channel* PWM.
- 8-*channel*, 10-*bit* ADC.
- *On-Chip Analog Comparator*.
- *Programmable Serial* USART.
- *Real Time Counter with Separate Oscillator*.
- Konfigurasi pin mikrokontroler AVR ATmega16 untuk 40 pin DIP (*dual line package*) ditunjukkan pada Gambar 2.10 berikut.



Gambar 2.10 Konfigurasi *Pin* ATmega16

ATmega16 memiliki 40 *Pin*, yang masing-masing pin nya memiliki fungsi yang berbeda-beda baik sebagai port maupun fungsi yang lainnya. Berikut ini akan dijelaskan fungsi dari masing-masing kaki ATmega16.

Tabel 2.1 Fungsi *Pin* ATmega16

No. Pin	**Nama Pin**	**Keterangan**
10	VCC	Catu Daya
11	GND	Ground
40 - 33	PortA: PA0 – PA7 (ADC0 – ADC7)	*Port I/O* dua arah yang dilengkapi *internal pull-up resistor*, Port ini juga dimultipleks dengan masukan *analog* ke ADC 8 *channel*
1 - 7	PortB: PB0 – PB7	*Port I/O* dua arah yang dilengkapi *internal pull-up resistor*. Fungsi lain dari masing-masing adalah: PB0:T0(*timer/counter0 external counter input*) PB1:T1(*timer/counter1 external counter input*) PB2: AIN0 (*analog comparator positive input*) PB3:AIN1 (*analog comparator positive input*) PB4:SS (SPI *slave select input*) PB5:MOSI (SPI *bus master input/slave input*) PB6:MISO (SPI *bus master input/slave output*) PB7:SCK (SPI *bus serial clock*)
22- 29	PortC: PC0 – PC7	*Port I/O* dua arah yang dilengkapi *internal pull-up resistor*. Dua pin yaitu PC6 dan PC7 berfungsi sebagai *oscillator* eksternal untuk *timer/counter2*
14 - 21	PortD: PD0 – PD7	*Port I/O* dua arah yang dilengkapi *internal pull-up resistor*. Fungsi lain dari *port* ini masing-masing adalah: PD0:RXD (UART *input line*) PD1:TXD (UART *input line*) PD2:INT0 (*External Interupt 0 input*) PD3:INT1 (*External Interupt 1 Input*) PD4:OC1A (*timer/counter1 output compareA match output*) PD5:OC1B (*timer/counter1 output compareB match output*) PD6:ICP (*timer/counter1 input capture pin*) PD7:OC2 (*timer/counter2 output compare match output*)
13	XTAL1	Masukan ke *inverting oscillator amplifier* dan masukan ke rangkaian *internal clock*
12	XTAL2	Keluaran dari *inverting oscillator amplifier*
30	AVCC	Catu Daya untuk *Port*A dan ADC

31	AGND	*Analog Ground*
32	AREF	Referensi masukan *analog* untuk ADC

2.5. Sensor Ultrasonik PING

Gelombang ultrasonik adalah gelombang yang memiliki frekuensi diatas 20 KHz dan diluar jangkauan pendengaran manusia. Sensor ultrasonik bekerja berdasarkan prinsip pantulan gelombang suara, dimana sensor ini menghasilkan gelombang suara yang kemudian menangkapnya kembali dengan perbedaan waktu sebagai dasar penginderaannya. Perbedaaan waktu antara gelombang suara yang dipancarkan dengan ditangkapnya kembali gelombang suara tersebut merupakan representasi jarak. Kecepatan rambat gelombang ultrasonik dipengaruhi oleh beberapa faktor antara lain adalah suhu, tekanan, kelembaban, dan sinyal radio.

Hal yang harus diperhatikan ketika menggunakan sensor ultrasonik terhadap suatu objek yaitu sebagai berikut.

a. Bentuk dan Ukuran Objek

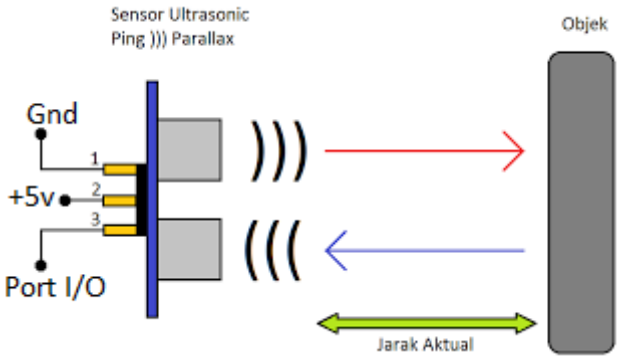
Objek dengan ukuran yang lebih besar akan memantulkan lebih banyak gelombang dibandingkan dengan objek yang berukuran kecil sehingga memberikan hasil yang lebih presisi. Bentuk suatu objek mempengaruhi banyaknya pantulan yang dipantulkan objek. Objek dengan bentuk bulat akan memantulkan gelombang ke segala arah sehingga pantulan yang ditangkap menjadi lemah. Objek dengan bentuk datar akan memantulkan gelombang dengan lebih baik.

b. Jenis Material

Objek padat yang terbuat dari besi atau baja akan memberikan pantulan yang lebih baik dibandingkan dengan objek yang terbuat dari kayu atau gabus.

c. Pola Permukaan

Objek dengan permukaan datar, halus, dan tegak lurus terhadap sinyal ultrasonik yang dipancarkan akan memberikan pantulan yang lebih kuat dibandingkan dengan permukaan tidak rata.



Gambar 2.11 Prinsip Kerja ULTRASONIK PING



Gambar 2.12 Sensor ULTRASONIK PING

2.6. *Pulse Width Mudulation* (PWM)

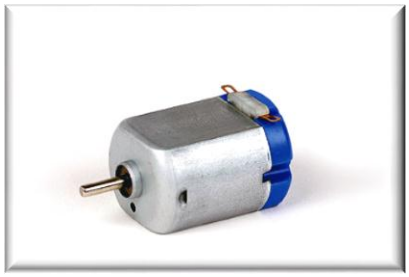
PWM merupakan suatu teknik teknik dalam mengatur kerja suatu peralatan yang memerlukan arus pull in yang besar dan untuk menghindari disipasi daya yang berlebihan dari peralatan yang akan dikontrol. PWM merupakan suatu metoda untuk mengatur kecepatan perputaran motor dengan cara mengatur prosentase lebar pulsa high terhadap perioda dari suatu sinyal persegi dalam bentuk tegangan periodik yang diberikan ke motor sebagai sumber daya. Semakin besar perbandingan lama sinyal high dengan perioda sinyal maka semakin cepat motor berputar.

Sinyal PWM dapat dibangun dengan banyak cara, dapat menggunakan metode analog menggunakan rankaian op-amp atau dengan menggunakan metode digital. Dengan metode analog setiap perubahan PWM-nya sangat halus, sedangkan menggunakan metode digital setiap perubahan PWM dipengaruhi oleh resolusi dari PWM itu sendiri. Misalkan PWM digital 8 bit berarti PWM tersebut memiliki resolusi 2 pangkat 8 = 256, maksudnya nilai keluaran PWM ini memiliki 256 variasi, variasinya mulai dari 0 – 255 yang mewakili duty cycle 0 – 100% dari keluaran PWM tersebut. Pada perancangan driver ini, sinyal PWM akan diatur secara digital yang dibangkitkan oleh mikrokontroler.

2.7. Motor Penggerak Roda

Motor DC adalah salah satu komponen yang dapat difungsikan sebagai penggerak roda-roda pada *prototype* sistem otomatis pengaturan rem pada mobil. Motor DC memiliki fungsi untuk mengubah energi listrik menjadi energi mekanik. Pada aplikasinya pergerakan pada sebuah mobil listrik umumnya

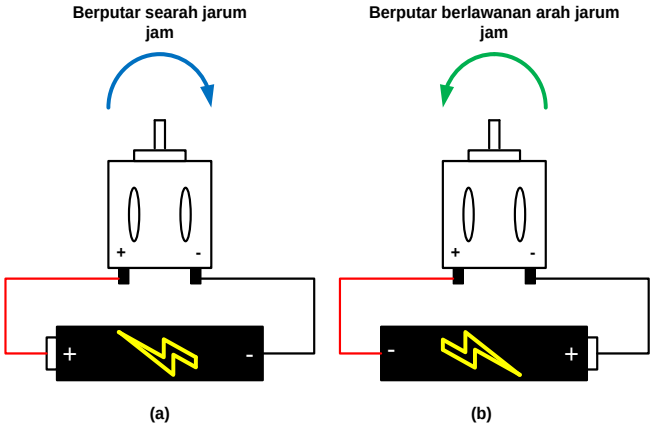
menggunakan motor DC sebagai komponen penggeraknya, karena jenis motor ini lebih mudah untuk dikendalikan. Kecepatan yang dihasilkan oleh motor DC berbanding lurus dengan potensial yang diberikan.



Gambar 2.13 Bentuk Fisik Motor *DC*

2.7.1. Ilustrasi Sederhana Untuk Menguji Motor DC

Pengaturan arah putaran pada motor DC dilakukan dengan mengubah arah polaritas yang mengalir melalui motor. Secara sederhana dapat dilihat pada Gambar 2.15 di bawah ini bahwa hal tersebut dapat dilakukan dengan cara mengubah polaritas tegangan pada *motor* DC.



Gambar 2.14 Pengaturan Arah Putaran pada *Motor DC*

Bila kita lihat pada gambar (a) di atas dijelaskan bahwa perputaran motor DC dikatakan searah jarum jam jika diberikan tegangan sumber dengan polaritas yang searah maka arah putaran motor akan mengarah ke (kanan). Sedangkan

pada gambar (b) dijelaskan bahwa perputaran *motor DC* dikatakan berlawanan arah jarum jam jika diberikan tegangan sumber dengan polaritas terbalik, maka arah putaran motor akan mengarah ke (kiri).

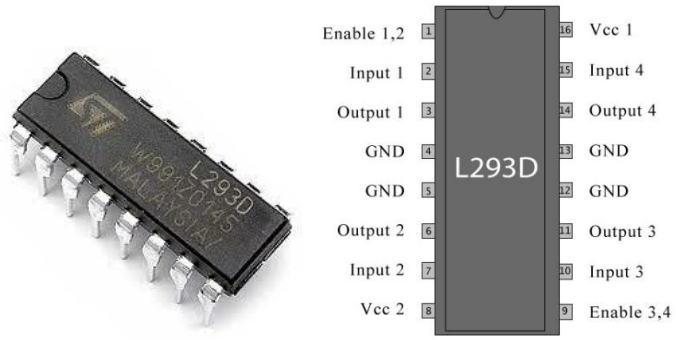
Kecepatan pada *motor DC* dapat diatur dengan beberapa cara, yaitu dengan mengatur fluks medan dan dengan mengaut tegangan sumber. Cara lain yang sering digunakan yaitu dengan melakukan pengaturan tegangan sumber dengan menggunakan metode PWM (*Pulse Width Modulation*). Sebenarnya *motor DC* tidak dapat dikendalikan secara langsung oleh mikrokontroler, karena kebutuhan arus listrik yang besar pada *motor DC* sedangkan arus keluaran pada mikro sangat kecil.

Drivermotor merupakan pilihan alternatif yang dapat digunakan karena dengan adanya *drivermotor* ini dapat membantu untuk mengontrol/ mengendalikan *motor DC* pada *prototype* sistem otomatis pengaturan rem pada mobil. Ada beberapa jenis *drivermotor DC* yang sering digunakan diantaranya, yaitu menggunakan rangkaian *H-bridge transistor, H-bridge MOSFET* dan *IC drivermotor DC*. *Drivermotor* yang mudah digunakan yaitu menggunakan *IC drivermotor DC* seperti jenis L293D dan L298N.

2.8. *DriverMotor* L293D

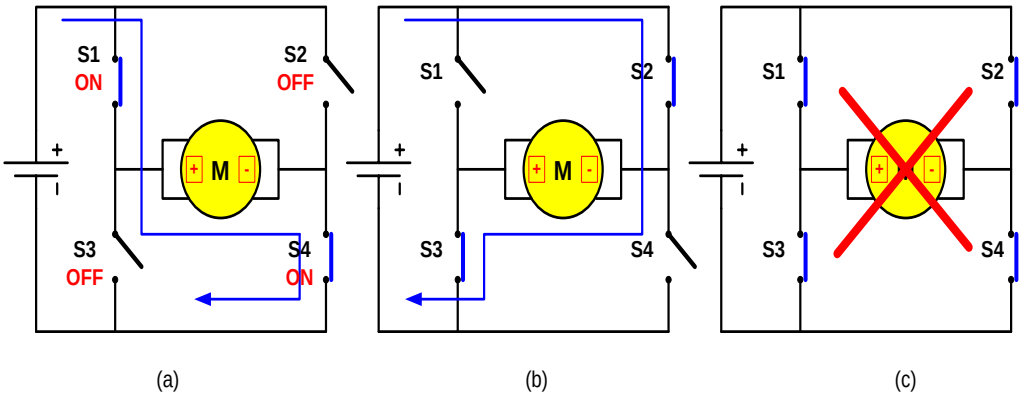
L293D diproduksi oleh *SGS-Thomson* merupakan sebuah *chip* yang di dalamnya terdapat dua rangkaian *H-Bridge*. Komponen ini bisa men-*drive* (mengendalikan) dua buah *motor DC* sampai tegangan 4,5 VDC, arus sebesar 1A untuk tiap kanal dengan arus puncak 1,2A. Untuk satu bagian *motor DC* dapat dikontrol dengan 3 *pin* seperti 1(*enable*) dan 2 *inputan* . Kemudian pin *output*

yang menuju ke *motor DC* dihubungkan dengan *pin outputan*. Di bawah ini terdapat gambar IC *drivermotor DC* dan konfigurasi *pin* nya.



Gambar 2.15 Bentuk Fisik IC *DriverMotor* L298D

Secara konsep rangkaian *H-bridge* ini dapat diilustrasikan sebagai 4 saklar yang tersusun sedemikian rupa sehingga memungkinkan motor dapat teraliri arus dengan arah yang berkebalikan. Cara kerja dari rangkaian *H-bridge* terdapat pada Gambar 2.17 berikut ini:



Gambar 2.16 Cara Kerja *H-bridge*

Dari gambar diatas, berikut *H-bridge* bekerja:

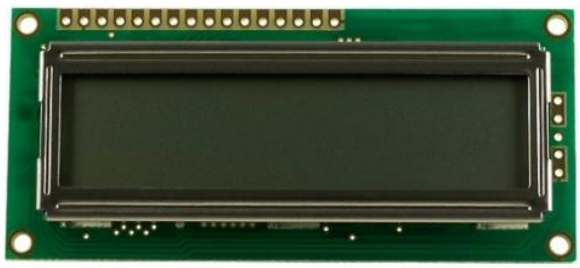
a) Ketika S1 dan S4 tertutup (*diagonal*) dan S2 dan S3 terbuka, maka arus akan mengalir dari baterai menuju ke kutub positif motor kemudian keluar ke kutub negatif motor, maka motor akan berputar ke arah kanan.

b) Ketika S2 dan S3 tertutup (*diagonal*) dan S1 dan S4 terbuka, maka arus akan mengalir sebaliknya yaitu mengalir dari baterai kenuju ke kutub negatif motor kemudian keluar ke kutup positif motor, maka motor akan berputar ke arah kiri.

c) Jika semua saklar tertutup, maka motor akan berhenti, dan jika diteruskan maka akan menyebabkan "*short circuit*".

2.9. *Liquid Crystal Display* (LCD)

Liquid Crystal Display (LCD) merupakan *display* dot matriks yang difungsikan untuk menampilkan tulisan berupa angka atau huruf sesuai dengan yang diinginkan, angka atau huruf yang ditampilkan sesuai dengan program yang digunakan untuk mengontrolnya. LCD *Character* memiliki banyak jenis dilihat dari jumlah bit-nya. LCD dot matriks dengan karakter 16x2 dan memiliki jumlah pin sebanyak 16 pin. Dalam tugas akhir ini LCD *Character* digunakan untuk menampilkan jarak antara mobil dengan *obstacle* di depannya dan sebagai indikator kecepatan, dalam arti lain LCD berfungsi mengantikan peran speedometer pada *prototype* mobil .



Gambar 2.17 16x2 *Character* LCD *Module*

2.10. *Buzzer*

Buzzer adalah sebuah komponen elektronika yang berfungsi untuk mengubah getaran listrik menjadi getaran suara. Pada dasarnya prinsip kerja *buzzer* hampir sama dengan loud speaker, jadi *buzzer* juga terdiri dari kumparan yang terpasang pada diafragma dan kemudian kumparan tersebut dialiri arus sehingga menjadi elektromagnet, kumparan tadi akan tertarik ke dalam atau keluar, tergantung dari arah arus dan polaritas magnetnya, karena kumparan dipasang pada diafragma maka setiap gerakan kumparan akan menggerakkan diafragma secara bolak-balik sehingga membuat udara bergetar yang akan menghasilkan suara. *Buzzer* biasa digunakan sebagai indikator bahwa proses telah selesai atau terjadi suatu kesalahan pada sebuah alat (alarm). Dalam tugas akhir ini *buzzer* digunakan sebagai indikator ketika mobil telah mencapai jarak maksimum dengan *obstacle* didepannya .



Gambar 2.18 *Buzzer*

2.11. *Motor Servo*

Motor Servo adalah sebuah motor dengan sistem umpan balik tertutup dimana posisi dari motor 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.



Gambar 2.19 *Motor Servo*

2.12. Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan dalam tugas akhir ini adalah program Bahasa C. Berikut adalah uraian dari masing-masing perangkat lunak tersebut.

2.12.1. Bahasa C

Pemrograman menggunakan Bahasa C adalah salah satu dari sekian banyak bahasa dasar untuk pemrograman mikrokontroler, misalnya bahasa *Assembly, Basic*, dan lain-lain. Bahasa pemrograman Bahasa C dikenal di seluruh dunia sebagai bahasa pemrograman handal, cepat, mudah, dan tergolong ke dalam bahasa pemrograman tingkat menengah. Bahasa C banyak digunakan untuk aplikasi mikrokontroler karena kemudahan dan kompatibel terhadap mikrokontroler jenis AVR. Setiap bahasa pemrograman mempunyai standar penulisan program. Konstruksi dari program bahasa C harus mengikuti aturan sebagai berikut.

1. *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 *forder 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.

2. Komentar

Komentar merupakan bagian kode program yang tidak dieksekusi saat program dijalankan. Komentar diperlukan untuk memperjelas jalannya program agar lebih mudah dimengerti. Komentar dapat dimanfaatkan untuk penuangan ide yang belum sempat diimpelementasikan.

Contoh penulisan :

// hanya satu ini sebelu enter yang dianggap komentar.

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

Komentar adalah bagian yang tidak dieksekusi oleh kompiler. Ini mempertegas bahwa kode program dibuat untuk dibaca dan dimengerti oleh manusia. Biasakan untuk menuliskan komentar dibagian tertentu dari program.

3. Nama

Nama digunakan untuk mengidentifikasi variabel, konstanta, tipe, fungsi dan prosedur. Penamaan memiliki aturan.

- Dalam C, nama membedakan antara huruf besar dan huruf kecil.
- Nama harus dimulai dengan alfabet, tidak boleh angka, spasi atau karakter khusus.
- Karakter penyusun nama hanya huruf alfabet, angka dan underscore.
- Tidak boleh mengandung operator aritmatik, relasional, tanda baca dan karakter khusus.

Setiap nama harus ditulis secara utuh, tidak boleh diberi spasi.

Contohnya penulisan : int x, y, jumlah; // x, y dan jumlah adalah nama dari variabel intnilai(const int angka, int test){} // nilai adalah nama dari fungsi.

4. Operator

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

Aritmatika :

- + adalah penjumlahan.
- - adalah pengurangan.
- * adalahperkalian.
- / 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.

5. 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.