

BAB II

TINJAUAN PUSTAKA

2.1. Definisi *Artificial Intelligence*

Definisi *Artificial Intelligence* merupakan salah satu bagian dari ilmu komputer yang mempelajari bagaimana membuat mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik apa yang dilakukan manusia. apabila dilakukan oleh manusia disebut kecerdasan atau *intelligence* (Efraim Turban, 1992). Sebagian kalangan menerjemahkan *Artificial Intelligence* sebagai kecerdasan buatan, kecerdasan artifisial, inteligensia artifisial, atau inteligensia buatan. Para ahli mendefinisikan AI secara berbeda-beda tergantung pada sudut pandang masing-masing. Ada yang fokus pada logika berpikir manusia saja, tetapi ada juga yang mendefinisikan AI secara lebih luas pada tingkah laku manusia. Definisi lain tentang *Artificial Intelligence* merupakan ilmu yang mempelajari bagaimana cara membuat komputer dapat melakukan pekerjaan-pekerjaan yang untuk saat ini manusia dapat mengerjakannya dengan baik (E.Rich, 1983). Tiga tujuan dari *Artificial Intelligence* yaitu membuat mesin menjadi lebih pintar, memahami apa itu kecerdasan (*intelligence*) dan membuat mesin lebih berguna (Dahria Muhamad, 2008).

2.2. Logika Fuzzy

Logika *fuzzy* adalah salah satu komponen pembentuk *soft computing*. Logika *fuzzy* pertama kali diperkenalkan oleh **Prof. Lotfi A. Zadeh** pada tahun 1965. Dasar logika *fuzzy* adalah teori himpunan *fuzzy*. Pada teori himpunan *fuzzy*, peranan derajat keanggotaan sebagai penentu keberadaan elemen dalam suatu himpunan sangatlah penting. Nilai keanggotaan atau derajat keanggotaan atau *membership function* menjadi ciri utama dalam penalaran dengan logika *fuzzy* tersebut. Logika *fuzzy* dapat dianggap sebagai kotak hitam yang berhubungan antara ruang *input* menuju ruang *output*. Kotak hitam tersebut berisi cara atau metode yang dapat digunakan untuk mengolah data *input* menjadi *output* dalam bentuk informasi yang baik. (Kusuma Dewi S, 2003).

2.2.1. Himpunan dan Fungsi Keanggotaan Logika Fuzzy

Pada himpunan tegas (Crisp), nilai keanggotaan suatu variabel x dalam suatu himpunan A , yang sering ditulis dengan $\mu_A[x]$, memiliki dua kemungkinan, (Kusumadewi S, Purnomo H, 2010) yaitu

1. Satu (1), yang berarti bahwa suatu item menjadi anggota dalam suatu himpunan atau
2. Nol (0), yang berarti bahwa suatu item tidak menjadi anggota dalam suatu himpunan.

Kalau pada himpunan tegas (crisp), nilai keanggotaan hanya ada dua kemungkinan yaitu 0 atau 1, maka pada himpunan fuzzy nilai keanggotaan terletak pada rentang 0 sampai 1. Apabila x memiliki nilai keanggotaan fuzzy $\mu_A[x]=0$ berarti x tidak menjadi anggota himpunan A , demikian pula apabila x memiliki nilai keanggotaan fuzzy $\mu_A[x]=1$ berarti x menjadi anggota penuh pada 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 : Dekat, Sedang, Jauh.
2. Numeris, yaitu suatu nilai (angka) yang menunjukkan ukuran dari suatu variabel seperti : 25, 35, 40 dan sebagainya.

Ada beberapa hal yang perlu diketahui dalam memahami sistem *fuzzy* (Aplikasi logika *fuzzy* untuk pendukung keputusan, Sri Kusumadewi, Hari Purnomo, Edisi kedua, Graha Ilmu, 2010), yaitu:

a. Variabel fuzzy

Variabel fuzzy merupakan variabel yang hendak dibahas dalam suatu sistem fuzzy, contoh : kecepatan dan jarak.

b. Himpunan fuzzy

Himpunan fuzzy merupakan suatu grup yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel fuzzy. Contohnya adalah variabel jarak terbagi menjadi tiga himpunan fuzzy, yaitu : Dekat, Sedang, Jauh.

c. Semesta Pembicaraan

Semesta pembicaraan adalah keseluruhan nilai yang diperbolehkan untuk dioperasikan dalam suatu variabel fuzzy. Semesta pembicaraan merupakan himpunan bilangan real yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan. Nilai semesta pembicaraan dapat berupa bilangan positif maupun negatif. Pada suatu kondisi tertentu nilai semesta pembicaraan ini tidak dibatasi batas atasnya.

Contoh :

1. Semesta pembicaraan untuk variabel kecepatan $[0 \ 100]$
2. Semesta pembicaraan untuk variabel jarak $[0 \ 20]$

d. Domain

Domain himpunan fuzzy adalah keseluruhan nilai yang diijinkan 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 negatif.

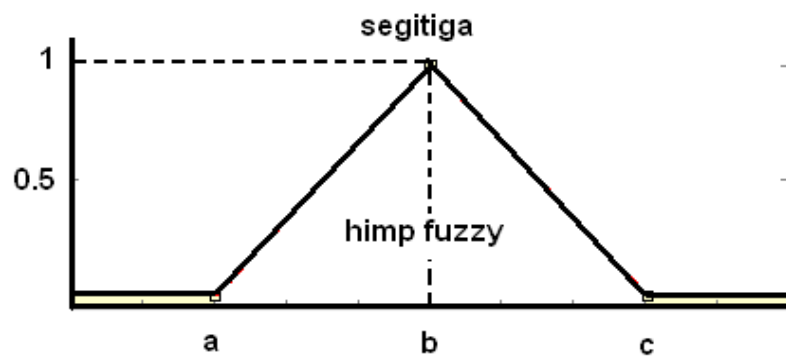
Contoh domain himpunan fuzzy :

1. DEKAT = $[0, \ 5]$
2. SEDANG = $[7, \ 10]$
3. JAUH = $[15, \ 20]$

Fungsi keanggotaan adalah suatu kurva yang menunjukkan pemetaan titik-titik *input* data kedalam 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. Apabila U menyatakan himpunan universal dan A adalah himpunan fungsi *fuzzy* dalam U , maka A dapat dinyatakan sebagai pasangan terurut (Wang, 1997 dari Wulandari, F., 2005). Ada beberapa fungsi yang bisa digunakan.

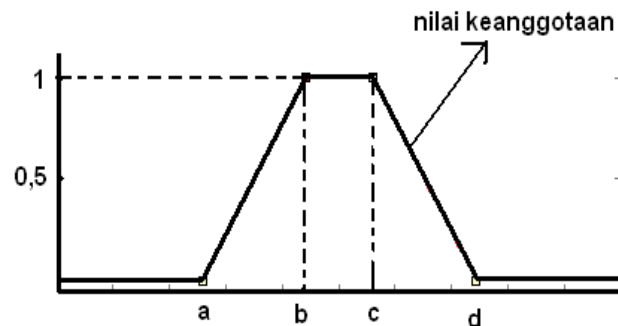
1. fungsi keanggotaan segitiga



Gambar 2.1. Himpunan fungsi keanggotaan

$$S(u; a, b, c) = \begin{cases} 0 & \text{untuk } u < a \\ (u - a)/(b - a) & \text{untuk } a \leq u \leq b \\ (c - u)/(c - b) & \text{untuk } b \leq u \leq c \\ 0 & \text{untuk } u > c \end{cases}$$

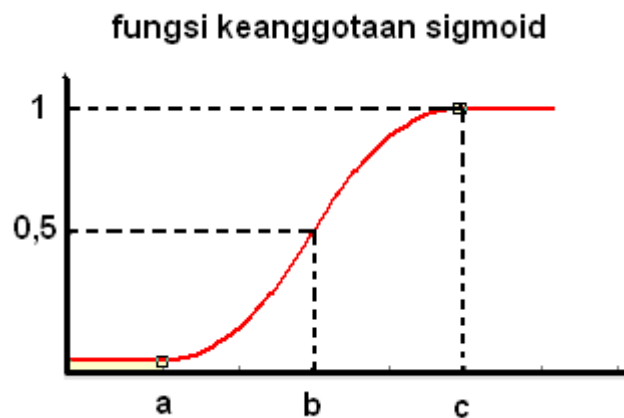
2. fungsi keanggotaan trapezium



Gambar 2.2. Himpunan Keanggotaan trapesium

$$S(u : a, b, c, d) = \begin{cases} 0 & \text{untuk } u < a \\ (u - a)/(b - a) & \text{untuk } a \leq u \leq b \\ 1 & \text{untuk } b \leq u \leq c \\ (d - u)/(d - c) & \text{untuk } c \leq u \leq d \\ 0 & \text{untuk } u > d \end{cases}$$

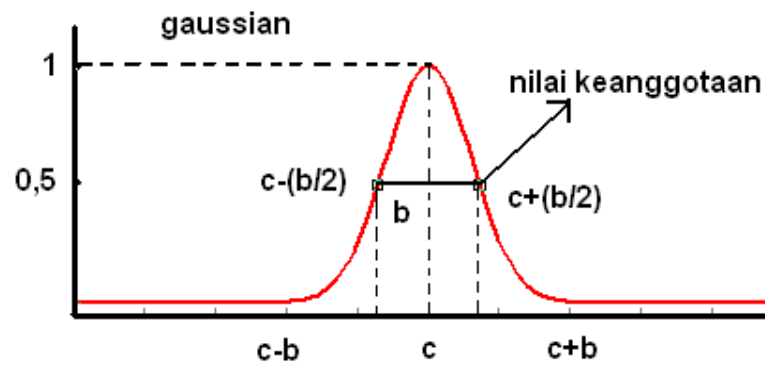
3. fungsi keanggotaan sigmoid



Gambar 2.3. Himpunan keanggotaan sigmoid

$$S(u; a, b, c) = \begin{cases} 0 & \text{untuk } u < a \\ 2[(u - a)/(c - a)]^2 & \text{untuk } a \leq u \leq b \\ 1 - 2[(u - c)/(c - u)]^2 & \text{untuk } b \leq u \leq c \\ 1 & \text{untuk } u > c \end{cases}$$

4. fungsi keanggotaan gaussian



Gambar 2.4. Himpunan keanggotaan gaussian

$$\pi(u; b, c) = \begin{cases} S(u; c - b, c - b/2, c) \\ 1 - S(u; c, c + b/2, c + b) \end{cases}$$

Untuk mengerti sistem fuzzy, kita harus mengenal konsep dasar yang berhubungan dengan logika fuzzy.

1. Derajat keanggotaan adalah derajat dimana nilai crisp compatible dengan fungsi keanggotaan (dari 0 sampai 1), juga mengacu sebagai tingkat keanggotaan, nilai kebenaran, atau masukan fuzzy.
2. Label adalah nama deskriptif yang digunakan untuk mengidentifikasi sebuah fungsi keanggotaan.
3. Fungsi keanggotaan adalah mendefinisikan fuzzy set dengan memetakan masukan crisp dari domainnya ke derajat keanggotaan.
4. Masukan crisp adalah masukan yang tegas dan tertentu.
5. Lingkup / domain adalah fungsi keanggotaan. Jangkauan konsep, biasanya bilangan tempat dimana fungsi keanggotaan dipetakan.
6. Daerah batasan crisp adalah jangkauan seluruh nilai yang mungkin dapat diaplikasikan pada variabel sistem. menggunakan logika fuzzy untuk mencapai penyelesaian crisp pada masalah khusus biasanya melibatkan tiga langkah : fuzzyfikasi, evaluasi rule, dan defuzzyfikasi.

2.2.2. Cara Kerja Kontrol Logika Fuzzy

Dalam system control logika fuzzy terdapat beberapa tahapan operasional yang meliputi :

1. Fuzzyfikasi
2. Penalaran (Inference Machine)
3. Aturan Dasar (Rule Baseed)
4. Defuzzyfikas.

1. Fuzzyfikasi

Fuzzifikasi yaitu suatu proses untuk mengubah suatu masukan dari bentuk tegas (crisp) menjadi fuzzy (variabel linguistik) yang biasanya disajikan dalam bentuk himpunan-himpunan fuzzy dengan suatu fungsi keanggotaannya masing-masing. Masukan crisp ditransformasikan kedalam masukan fuzzy. misalkan masukan crisp 9 cm akan ditransformasikan sebagai dekat dalam bentuk fuzzy. Untuk mengubah bentuk masukan crisp kedalam masukan fuzzy, hal yang perlu dilakukan menentukan fungsi keanggotaan untuk tiap masukan. Sekali fungsi keanggotaan ditentukan, fuzzifikasi mengambil nilai masukan secara *realtime*, seperti jarak, dan membandingkannya dengan informasi fungsi keanggotaan yang tersimpan untuk menghasilkan nilai masukan fuzzy.

Langkah pertama dalam fuzzifikasi adalah menentukan label-label fuzzy pada daerah batasan crisp dari setiap masukan crisp. Masukan fungsi keanggotaan dihasilkan dengan menuliskan satu demi satu bilangan, yaitu derajat keanggotaan, untuk setiap masukan yang mungkin dari label yang diberikan.

2. Evaluasi Rule

Sistem fuzzy menggunakan aturan linguistik untuk menentukan aksi kontrol apa yang harus dilakukan dalam merespon nilai masukan yang diberikan. Evaluasi rule terdiri dari sejumlah aturan yang biasanya dinyatakan secara *linguistic*. Evaluasi rule juga mengacu pada fuzzy interface, mengaplikasikan aturan pada masukan fuzzy yang dihasilkan dalam proses fuzzifikasi, kemudian mengevaluasi tiap aturan dengan masukan yang dihasilkan dari proses fuzzifikasi. Aturan fuzzy seringkali dinyatakan dengan “IF...THEN....”.

Pada sistem fuzzy alat ini menggunakan aturan-aturan dalam bentuk bahasa alami yangn dibatasi oleh istilah linguistik, serta sintaksis yang baku. Sintaksi tersebut adalah :

If antecedent 1 And antecedent 2 Then consequent 1 And consequent 2

Dimana :

And adalah salah satu operator logika fuzzy yang diizinkan

Antecedent adalah bentuk dari : Variabel masukan = Label (contohnya : jarak sensor kiri = dekat, dimana jarak sensor kiri adalah variabel masukan dan dekat adalah salah satu label fungsi keanggotaan yang berhubungan dengan jarak sensor kiri).

Consequent adalah bentuk : variabel keluaran = label (contohnya : kecepatan motor = cepat)

2.2.3. Mesin Penalaran Kontrol Logika Fuzzy

Mesin Penalaran adalah proses implikasi dalam menalar nilai masukan guna penentuan nilai keluaran sebagai bentuk pengambilan keputusan. Pada umumnya tiap-tiap aturan (*proposisi*) *fuzzy* dinyatakan dalam bentuk IF..THEN.. dan menyatakan suatu hubungan tertentu. Hubungan *fuzzy* ini sering disebut implikasi. Hubungan *fuzzy* dalam *knowledge base* dapat didefinisikan sebagai himpunan implikasi *fuzzy*.

Ada 2 jenis *proposisi fuzzy* yaitu “*condition fuzzy proposition*” dan “*uncondition fuzzy proposition*”.

1) *Condition Fuzzy Proposition*

Jenis ini dicirikan dengan penggunaan IF.

IF x is A THEN y is B

2) *Uncondition Fuzzy Proposition.*

Jenis *uncondition* ditandai dengan tidak adanya pernyataan IF.

x is A

proposisi uncondition selalu diaplikasi dengan model AND.

Jika dalam *system fuzzy* terdapat beberapa aturan, maka ada 3 metode yang dipakai dalam menentukan inferensi yaitu : *max-min*, *additive* dan *probabilistic OR (probor)*

A. Metode *Max-Min*

Max dapat dianalogikan dengan operasi logika OR sedangkan Min dianalogikan dengan operasi logika AND.

B. Metode *additive*

metode *additive* dilakukan dengan melakukan *bounded-sum* terhadap semua output daerah fuzzy. Secara umum dapat dituliskan dengan L:

$$\mu_{sf}[x_i] \leftarrow \min (1, \mu_{sf}[x_i] + \mu_{kf}[x_i])$$

$\mu_{sf}[x_i]$ = nilai keanggotaan solusi fuzzy sampai aturan ke-i

$\mu_{kf}[x_i]$ = nilai keanggotaan konsekuen (output) fuzzy sampai aturan ke-i

C. Metode Probor

Metode probor diperoleh dengan melakukan *product* (perkalian) terhadap semua output daerah fuzzy. Secara umum dituliskan dengan :

$$\mu_{sf}[x_i] \leftarrow (\mu_{sf}[x_i] + \mu_{kf}[x_i]) - \leftarrow (\mu_{sf}[x_i] * \mu_{kf}[x_i])$$

$\mu_{sf}[x_i]$ = nilai keanggotaan solusi *fuzzy* sampai aturan ke-i

$\mu_{kf}[x_i]$ = nilai keanggotaan konsekuen (*output*) *fuzzy* sampai aturan ke-i

Salah satu model yang banyak dipakai adalah penalaran *Max-Min*. Dalam penalaran *max-min* proses pertama yang dilakukan adalah melakukan operasi min sinyal keluaran lapisan *fuzzyfikasi*, yang diteruskan dengan operasi *max* untuk mencari nilai keluaran yang selanjutnya akan difuzzyfikasikan sebagai bentuk keluaran pengontrol *Defuzzifikasi*.

3. *Defuzzifikasi*

Merupakan proses pemetaan himpunan *fuzzy* ke himpunan tegas (*crisp*). Proses ini merupakan kebalikan dari proses *fuzzyfikasi*.

Proses *defuzzifikasi* diekspresikan sebagai berikut :

$$Z^* = \text{defuzzifier}(Z)$$

Dimana :

Z = Hasil penalaran *fuzzy*

Z* = Keluaran Kontrol Logika *Fuzzy*

Defuzzifier = Operasi *defuzzier*

Ada beberapa metode *defuzzifikasi* antara lain :

a. Metode centroid

metode centroid dilakukan dengan mengambil titik pusat daerah *fuzzy*. Secara umum dirumuskan dengan :

$$z = \frac{\int z\mu(z)dz}{\int \mu(z)dz} \quad \text{atau} \quad z = \frac{\sum_{j=1}^n z_j\mu(z_j)}{\sum_{j=1}^n \mu(z_j)}$$

b. Metode *bisector*

metode *bisector* dilakukan dengan mengambil nilai dari domain *fuzzy* yang memiliki nilai keanggotaan separo dari nilai keanggotaan *fuzzy*. Secara dirumuskan dengan :

$$z_p \text{ -- sedemikian -- hingga -- } \int_{R1}^p \mu(z)dz = \int_p^{Rn} \mu(z)dz$$

c. Metode *Mean of Maximum* (MOM)

metode MOM dilakukan dengan mengambil nilai rata-rata domain yang memiliki nilai keanggotaan maksimum

d. Metode *Largest of Minimum* (LOM)

metode LOM dilakukan dengan mengambil nilai terbesar dari domain yang memiliki nilai keanggotaan maksimum.

e. Metode *Smallest of Maximum* (SOM)]

metode LOM dilakukan dengan mengambil nilai terkecil dari domain yang memiliki nilai keanggotaan maksimum.

Contoh :

Pembentukan himpunan *fuzzy*, dimana baik variabel *input* maupun variabel *output* masing-masing dibagi 3 himpunan *fuzzy*. Ada 3 variabel *fuzzy* yang digunakan, yaitu:

1. Jarak kendaraan adalah jarak antara mobil dengan target, yang terdiri dari 3 himpunan *fuzzy*, yaitu: DEKAT, SEDANG dan JAUH.
2. Jarak tikungan adalah jarak antara mobil dan tikungan, yang terdiri dari 3 himpunan *fuzzy*, yaitu: DEKAT, SEDANG dan JAUH.
3. *Output* yang berupa kecepatan ideal suatu mobil, yang terdiri dari 3 himpunan *fuzzy*, yaitu: LAMBAT, SEDANG dan CEPAT.

Data yang akan digunakan untuk membantu penelitian ini adalah data jarak dan kecepatan rata-rata dari suatu mobil yaitu :

- a. Untuk x berupa jarak kendaraan, datanya adalah sebagai berikut:

Dekat : $x \leq 5m$

Jauh : $15 m \leq x \leq 20m$

- b. Untuk y berupa jarak tikungan, datanya adalah sebagai berikut:

Dekat : $y \leq 5m$

Jauh : $10m \leq y \leq 15m$

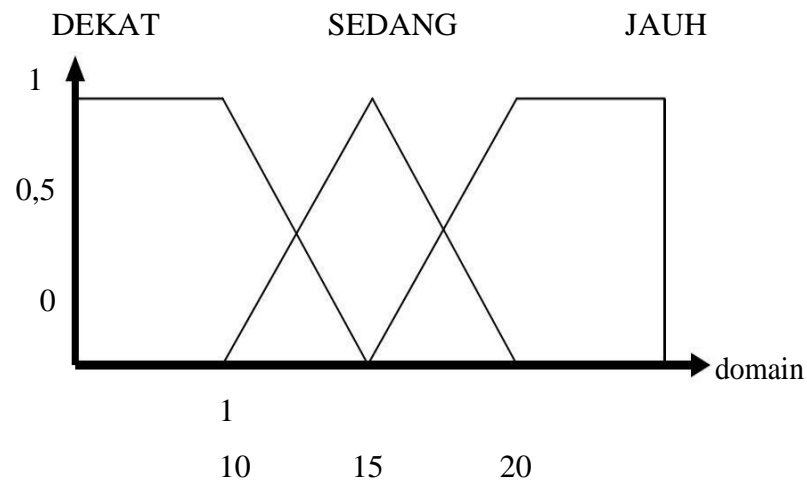
- c. Untuk z berupa kecepatan, datanya adalah sebagai berikut:

Lambat : $z \leq 40 \text{ km/jam}$

Cepat : $80 \text{ km/jam} \leq z \leq 100 \text{ km/jam}$

Berdasarkan data jarak dan kecepatan, maka dibuatlah sebuah fungsi keanggotaan dari masing-masing variabel yang akan digunakan. Fungsi keanggotaan merupakan suatu kurva yang menunjukkan pemetaan titik-titik input data ke dalam nilai keanggotaannya atau derajat keanggotaan. Adapun fungsi keanggotaan dari masing-masing variabel adalah sebagai berikut.

- a. Variabel Jarak Kendaraan



Gambar 2.5 Fungsi Keanggotaan Variabel Jarak

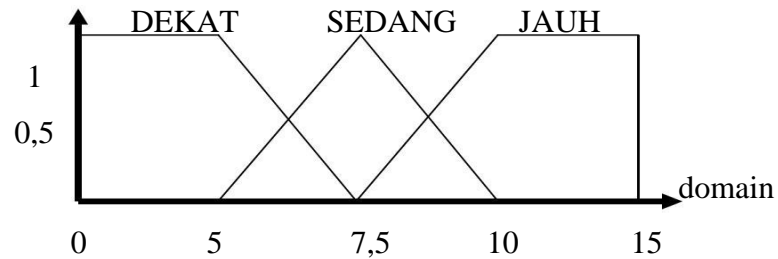
Fungsi keanggotaannya adalah sebagai berikut:

$$\mu_{JKDekat}(x) = \begin{cases} 1 & ; x \leq 5 \\ \frac{(10-x)}{(10-5)} & ; x \leq 5 \leq 10 \\ 0 & ; x \geq 10 \end{cases}$$

$$\mu_{JKSedang}(x) = \begin{cases} 0 & ; x \leq 5 \text{ atau } x \geq 15 \\ \frac{(x-5)}{(10-5)} & ; 5 \leq x \leq 10 \\ \frac{(15-x)}{(15-10)} & ; 10 \leq x \leq 15 \end{cases}$$

$$\mu_{JKJauh}(x) = \begin{cases} 1 & ; x \leq 5 \\ \frac{(10-x)}{(10-5)} & ; x \leq 5 \leq 10 \\ 0 & ; x \geq 10 \end{cases}$$

b. Variabel Jarak Tikungan



Gambar 2.6 Fungsi Keanggotaan Variabel Jarak terhadap Tikungan

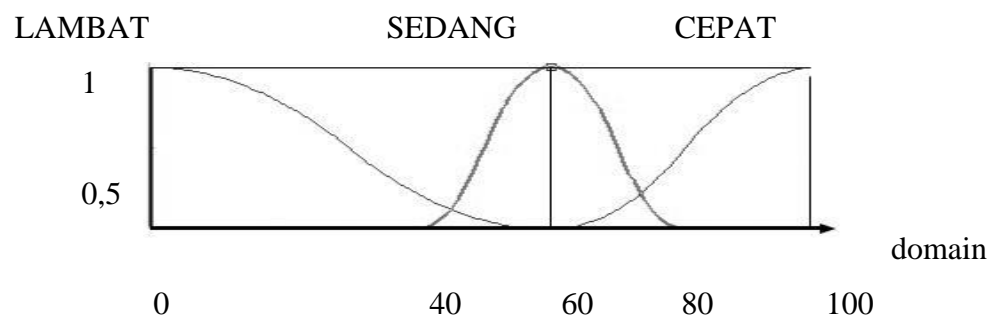
Fungsi keanggotaannya adalah sebagai berikut:

$$\mu_{JKDekat}(x) = \begin{cases} 1 & ; x \leq 5 \\ \frac{(10-x)}{(10-5)} & ; x \leq 5 \leq 10 \\ 0 & ; x \geq 10 \end{cases}$$

$$\mu_{JKSedang}(x) = \begin{cases} 0 & ; x \leq 5 \text{ atau } x \geq 15 \\ \frac{(x-5)}{(10-5)} & ; 5 \leq x \leq 10 \\ \frac{(15-x)}{(15-10)} & ; 10 \leq x \leq 15 \end{cases}$$

$$\mu_{JKJauh}(x) = \begin{cases} 0 & ; x \leq 10 \\ \frac{(x-10)}{(15-10)} & ; 10 \leq x \leq 15 \\ 1 & ; x \geq 15 \end{cases}$$

c. Variabel Kecepatan



Gambar 2.7 Fungsi Keanggotaan Variabel

Kecepatan Fungsi keanggotaannya adalah sebagai berikut:

$$\mu_{KLambat}(x) = \begin{cases} 1 & ; z \leq 0 \\ 1 - 2 \left(\frac{(z-0)}{(60-0)} \right)^2 & ; 0 \leq z \leq 30 \\ 2 \left(\frac{(60-z)}{(60-0)} \right)^2 & ; 30 \leq z \leq 60 \\ 0 & ; z \geq 60 \end{cases}$$

$$\mu_{KSedang}(x) = \begin{cases} 0 & ; z \leq 40 \text{ atau } z \geq 80 \\ 2 \left(\frac{(z-40)}{(60-40)} \right)^2 & ; 40 \leq z \leq 50 \\ 1 - 2 \left(\frac{(z-0)}{(60-0)} \right)^2 & ; 50 \leq z \leq 60 \\ 1 - 2 \left(\frac{(z-60)}{(80-60)} \right)^2 & ; 60 \leq z \leq 70 \\ 0 & ; 70 \leq z \leq 80 \\ 2 \left(\frac{(80-z)}{(80-60)} \right)^2 & \end{cases}$$

$$\mu_{KCepat}(x) = \begin{cases} 0 & ; z \leq 60 \\ 2 \left(\frac{(z-60)}{(100-60)} \right)^2 & ; 60 \leq z \leq 80 \\ 1 - 2 \left(\frac{(100-z)}{(100-60)} \right)^2 & ; 80 \leq z \leq 100 \\ 1 & ; z \geq 100 \end{cases}$$

Aplikasi fungsi implikasi, dimana pada metode *Mamdani* menggunakan fungsi implikasi *min*. Untuk mendapatkan daerah hasil implikasi diperlukan sejumlah aturan/*rules*. Jumlah aturan yang terbentuk berdasarkan 3 himpunan *fuzzy* adalah sebanyak 27 aturan. Dari 27 aturan hanya 9 aturan yang masuk akal dan layak digunakan, karena jika semua aturan digunakan maka hasilnya jauh dari yang diharapkan. 9 aturan yang digunakan diantaranya adalah sebagai berikut.

- Jika (Jarak kendaraan adalah DEKAT) dan (Jarak tikungan adalah DEKAT) maka (Kecepatan adalah LAMBAT).
- Jika (Jarak kendaraan adalah DEKAT) dan (Jarak tikungan adalah SEDANG) maka (Kecepatan adalah LAMBAT)
- Jika (Jarak kendaraan adalah DEKAT) dan (Jarak tikungan adalah JAUH) maka (Kecepatan adalah LAMBAT). Jika (Jarak kendaraan adalah SEDANG) dan (Jarak tikungan adalah DEKAT) maka (Kecepatan adalah LAMBAT)

- d. Jika (Jarak kendaraan adalah SEDANG) dan (Jarak tikungan adalah SEDANG) maka (Kecepatan adalah SEDANG)
- e. Jika (Jarak kendaraan adalah SEDANG) dan (Jarak tikungan adalah JAUH) maka (Kecepatan adalah SEDANG)
- f. Jika (Jarak kendaraan adalah JAUH) dan (Jarak tikungan adalah DEKAT) maka (Kecepatan adalah SEDANG)
- g. Jika (Jarak kendaraan adalah JAUH) dan (Jarak tikungan adalah SEDANG) maka (Kecepatan adalah SEDANG)
- h. Jika (Jarak kendaraan adalah JAUH) dan (Jarak tikungan adalah JAUH) maka (Kecepatan adalah CEPAT)

Komposisi antar aturan yang diperoleh dengan cara mengambil nilai maksimum aturan, kemudian mengaplikasikannya ke *output* dengan menggunakan operator OR. Proses *defuzzifikasi* adalah suatu proses dimana *input* nya adalah suatu himpunan *fuzzy* yang diperoleh dari komposisi aturan-aturan *fuzzy*, sedangkan *output* yang dihasilkan merupakan suatu bilangan pada *domain* himpunan himpunan *fuzzy* tersebut. Sehingga jika diberikan suatu himpunan *fuzzy* dalam *range* tertentu, maka harus dapat diambil suatu nilai *crisp* tertentu sebagai *output*. Pada penelitian ini, proses *defuzzifikasi* menggunakan metode *centroid* dimana nilai tegas diperoleh dengan cara mengambil titik pusat daerah *fuzzy*.

2.3. Mobile Robot

Robot mobil atau *mobile robot* adalah konstruksi robot yang ciri khasnya adalah mempunyai aktuator berupa roda untuk menggerakkan keseluruhan badan robot tersebut, sehingga robot tersebut dapat melakukan perpindahan posisi dari satu titik ke titik lain.

Mobile robot adalah tipe robot yang paling populer dalam dunia penelitian robot. Sebutan ini biasa digunakan sebagai kata kunci utama untuk mencari rujukan atau referensi yang berkaitan dengan robotik di internet. Publikasi dengan judul yang berkaitan *mobile robot* sering menjadi daya tarik, tidak hanya bagi kalangan peneliti, tapi juga bagi kalangan awam. Dari segi manfaat, penelitian tentang berbagai tipe *mobile robot* diharapkan dapat

membantu manusia dalam melakukan otomasi dalam transportasi, *platform* bergerak untuk robot industri, eksplorasi tanpa awak, dan masih banyak lagi.



Gambar 2.8 *Mobile Robot*

(sumber : <http://www.generationrobots.com> diakses 3 Maret 2016 pukul 19.18 WIB)

2.4. Komponen Penyusun Mobile Robot

2.4.1. Arduino Mega 2560

Arduino mega 2560 merupakan sebuah board mikrokontroler berbasis ATmega2560. Modul ini memiliki 54 *digital input/output* dimana 14 digunakan untuk *PWM output* dan 16 digunakan sebagai *analog input*, 4 port serial, 16 MHz osilator Kristal, koneksi USB, *power jack*, *ICISP Header*, dan tombol *reset*. Memiliki *flash memory* sebesar 256KB sangat cukup untuk menampung program yang banyak.

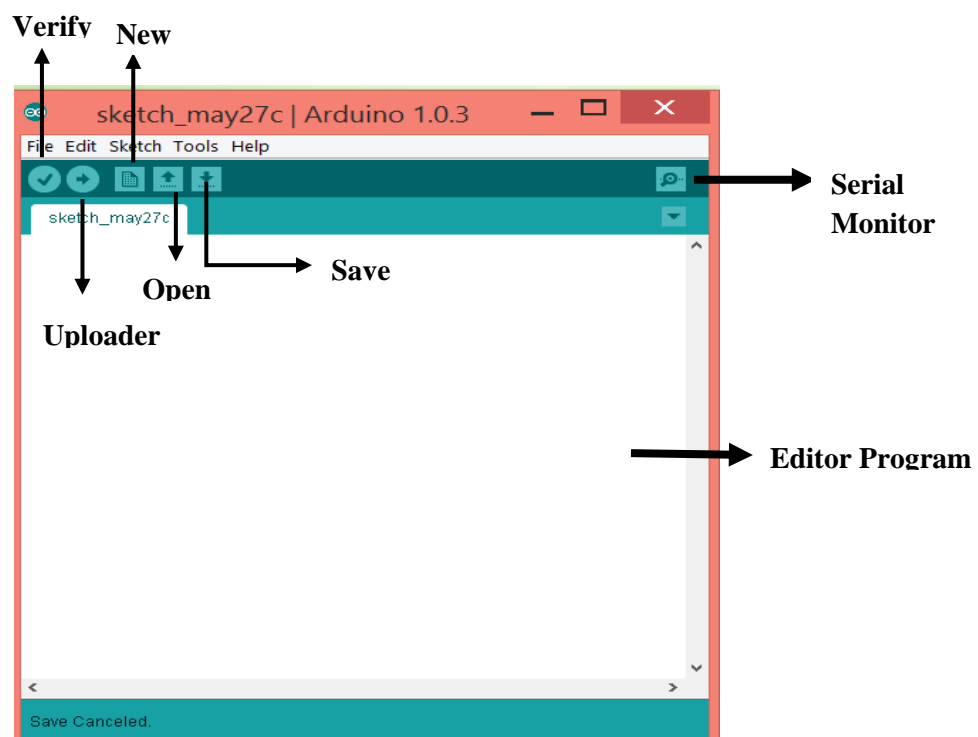
Arduino mega 2560 tidak memerlukan *flash program external* karena didalam *chip* mikrokontroler Arduino telah diisi dengan *bootloader* yang membuat proses *upload* program yang kita buat menjadi lebih sederhana dan cepat. Untuk koneksi dengan komputer sudah tersedia RS232 to TTL *converter* atau menggunakan chip USB ke serial converter.



Gambar 2.9 Arduino Mega 2560

2.4.2. Software Arduino

Software arduino yang digunakan adalah *driver* dan IDE, walaupun masih ada beberapa *software* lain yang sangat berguna selama pengembangan arduino. IDE atau *Integrated Development Environment* merupakan suatu program khusus untuk suatu komputer agar dapat membuat suatu rancangan atau sketsa program untuk papan Arduino. IDE arduino merupakan *software* yang sangat canggih ditulis dengan menggunakan java. IDE arduino terdiri dari :



Gambar 2.10 Tampilan Utama Pada Software Arduino

(Syahwil,2013:42)

Keterangan :

1. Editor Program

Sebuah *window* yang memungkinkan pengguna menulis dan mengedit program dalam bahasa *processing*.

2. *Verify*

Mengecek kode sketch yang *error* sebelum mengupload ke *board* arduino.

3. *Uploader*

Sebuah modul yang memuat kode biner dari komputer ke dalam memori di dalam papan arduino.

4. *New*

Membuat sebuah sketch baru.

5. *Open*

Membuka daftar sketch pada *sketchbook* arduino.

6. *Save*

Menyimpan kode sketch pada *sketchbook*.

7. Serial Monitor

Menampilkan data serial yang dikirimkan dari *board* arduino.

2.4.3. LCD (*Liquid Crystal Display*)

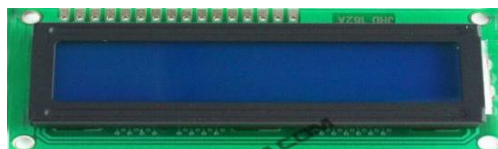
Display LCD sebuah *liquid crystal* atau perangkat elektronik yang dapat digunakan untuk menampilkan angka atau teks. Ada dua jenis utama layar LCD yang dapat menampilkan numerik (digunakan dalam jam tangan, kalkulator dll) dan menampilkan teks *alfanumerik* (sering digunakan pada mesin foto kopi dan telepon genggam). Dalam menampilkan numerik ini kristal yang dibentuk menjadi bar, dan dalam menampilkan *alfanumerik* kristal hanya diatur kedalam pola titik. Setiap kristal memiliki sambungan listrik individu sehingga dapat dikontrol secara independen. Ketika kristal *off* (yakni tidak ada arus yang melalui kristal) cahaya kristal terlihat sama dengan bahan latar belakangnya, sehingga kristal tidak dapat terlihat. Namun ketika arus listrik melewati kristal, itu akan merubah bentuk dan menyerap lebih banyak cahaya. Hal ini membuat kristal terlihat lebih gelap dari englihatan mata manusia sehingga bentuk titik atau bar dapat dilihat dari perbedaan latar belakang. Sangat penting untuk menyadari perbedaan antara layar LCD dan layar LED. Sebuah LED *display* (sering digunakan dalam radio jam) terdiri dari sejumlah LED yang benar-benar mengeluarkan cahaya (dan dapat dilihat dalam gelap). Sebuah layar LCD hanya mencerminkan cahaya, sehingga tidak dapat dilihat dalam gelap. LMB162A adalah modul LCD matrix

dengan konfigurasi 16 karakter dan 2 baris dengan setiap karakternya dibentuk oleh 8 baris *pixel* dan 5 kolom *pixel* (1 baris terakhir adalah kursor).

Memori LCD terdiri dari 9.920 bit CGROM, 64 byte CGRAM dan 80x8 bit DDRAM yang diatur pengalamatannya oleh *Address Counter* dan akses datanya (pembacaan maupun penulisan datanya) dilakukan melalui register data. Pada LMB162A terdapat register data dan register perintah. Proses akses data ke atau dari register data akan mengakses ke CGRAM, DDRAM atau CGROM bergantung pada kondisi *Address Counter*, sedangkan proses akses data ke atau dari Register perintah akan mengakses *Instruction Decoder* (dekoder instruksi) yang akan menentukan perintah-perintah yang akan dilakukan oleh LCD.

Klasifikasi *LED Display 16x2 Character* :

- a. 16 karakter x 2 baris
- b. 5x7 titik Matrix karakter + kursor
- c. HD44780 *Equivalent LCD controller/driver Built-In*
- d. 4-bit atau 8-bit MPU *Interface*
- e. Tipe standar
- f. Bekerja hampir dengan semua Mikrokontroler.

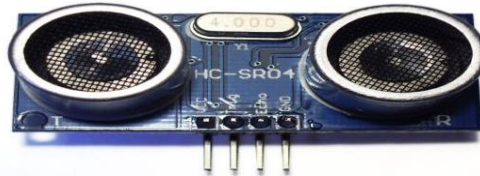


Gambar 2.11 LCD (*Liquid Crystal Display*)

2.4.4. Sensor Ultrasonik HC-SR04

Sensor HC-SR04 adalah sensor pengukur jarak berbasis gelombang ultrasonik. Prinsip kerja sensor ini mirip dengan radar ultrasonik. Gelombang ultrasonik dipancarkan kemudian diterima balik oleh *receiver* ultrasonik. Jarak antara waktu pancar dan waktu terima adalah representasi dari jarak objek. Sensor HC-SR04 adalah versi *low cost* dari sensor ultrasonik PING buatan parallax. Pada Sensor HC-SR04 pin trigger dan output diletakkan terpisah. Sensor ini mempunyai kisaran jangkauan maksimal 400-500 cm. Selain itu sensor HC-SR04

memiliki sudut deteksi terbaik pada 15 derajat, dengan tegangan kerja 5V DC (Lilik Gunarta 2011 : 2).



Gambar 2.12 Sensor Ultrasonik HC-SR04

(sumber : fjb.kaskus.co.id diakses 20 Maret 2016 pukul 21.09 WIB)

2.4.5. Baterai *Lithium Polymer* (LiPo)

Baterai LiPo tidak menggunakan cairan sebagai elektrolit melainkan menggunakan elektrolit polimer kering yang berbentuk seperti lapisan plastik film tipis. Lapisan film ini disusun berlapis-lapis diantara anoda dan katoda yang mengakibatkan pertukaran ion. Dengan metode ini baterai LiPo dapat dibuat dalam berbagai bentuk dan ukuran. Diluar dari kelebihan arsitektur baterai LiPo, terdapat juga kekurangan yaitu lemahnya aliran pertukaran ion yang terjadi melalui elektrolit polimer kering.



Gambar 2.13 Baterai LiPo

2.4.6. Motor DC

Motor DC memerlukan suplai tegangan yang searah pada kumparan medan untuk diubah menjadi energi mekanik. Kumparan medan pada motor dc disebut stator (bagian yang tidak berputar) dan kumparan jangkar disebut rotor (bagian yang berputar). Jika terjadi putaran pada kumparan jangkar dalam pada medan magnet, maka akan timbul tegangan (GGL) yang berubah-ubah arah pada

setiap setengah putaran, sehingga merupakan tegangan bolak-balik. Prinsip kerja dari arus searah adalah membalik fasa tegangan dari gelombang yang mempunyai nilai *positif* dengan menggunakan komutator, dengan demikian arus yang berbalik arah dengan kumparan jangkar yang berputar dalam medan magnet. Bentuk motor paling sederhana memiliki kumparan satu lilitan yang bisa berputar bebas di antara kutub-kutub magnet permanen. Jika arus lewat pada suatu konduktor, timbul medan magnet di sekitar konduktor. Arah medan magnet ditentukan oleh arah aliran arus pada konduktor.



Gambar 2.14 Motor DC

2.4.7. IC Motor Driver L298

L298 adalah IC yang dapat digunakan sebagai *driver* motor DC. IC ini menggunakan prinsip kerja *H-Bridge*. Tiap *H-Bridge* dikontrol menggunakan level tegangan TTL yang berasal dari *output* mikrokontroler. L298 dapat mengontrol 2 buah motor DC. Tegangan yang dapat digunakan untuk mengendalikan robot bisa mencapai tegangan 46 V dan arus 2 A untuk setiap kanalnya. Berikut ini bentuk IC L298 yang digunakan sebagai motor *driver*. Pengaturan kecepatan kedua motor dilakukan dengan cara pengontrolan lama pulsa aktif (mode PWM – *Pulse width Modulation*) yang dikirimkan ke rangkaian *driver* motor oleh pengendali (mikrokontroler *basic stamp*). *Duty cycle* PWM yang dikirimkan menentukan kecepatan putar motor DC.



Gambar 2.15 IC Motor Driver L298

2.4.8. DC-DC Step Down Module

DC to DC Converter ada 2 jenis, yaitu Step down DC Converter dan boost (step up) DC Converter. Pada step down DC Converter berfungsi seperti regulator biasa, menurunkan tegangan menjadi level tertentu dengan kestabilan tinggi, biasanya ripple maximum 1%. Dan input tegangan dengan range yg lebar. Pada 12A Adjustable DC-DC Step Down Module input voltage 4,5 - 30 VDC, output voltage 0.8-28 VDC 12A.

Dan sebaliknya boost (Step up) DC Converter, berfungsi untuk menaikkan atau menstabilkan tegangan pada level tertentu dengan output yg bisa lebih tinggi dari pada input, dan tetap stabil dengan ripple max 1%. Misalnya DC-DC Converter Input 10VDC sd 60VDC output 24VDC 10A atau Input 3VDC sd 15VDC output dual 15VDC 500mA.



Gambar 2.16 DC to DC Converter Module

(sumber : <https://ae01.alicdn.com> diakses 20 juni 2016)