

## BAB II

### TINJAUAN PUSTAKA DAN LANDASAN TEORI

#### 2.1 Tinjauan Pustaka

Teori klasik tentang himpunan atau “set” didasarkan pada konsep fundamental himpunan bahwa suatu entiti dapat merupakan anggota himpunan tersebut atau bukan merupakan anggotanya. Perbedaan yang tajam, jelas dan tidak ambigu terdapat antara anggota dan bukan anggota dari suatu himpunan yang telah didefinisikan pada teori ini. Dan terdapat batas yang sangat jelas agar dapat mengindikasikan bahwa suatu entiti merupakan bagian dari himpunan ini. (Chen *and* Pham, 2001). Ketika terdapat pertanyaan mengenai suatu entiti ini merupakan anggota dari himpunan atau tidak, jawabannya adalah “Ya” atau “Tidak”. Dalam kasus ini jawabannya dapat berupa misalnya, “Kemungkinan bahwa entiti ini merupakan anggota dari suatu himpunan adalah 90%”, namun kesimpulannya masih juga dapat dikatakan bahwa entiti ini adalah anggota atau bukan anggota dari suatu himpunan. Kemungkinan untuk seseorang dalam membuat prediksi yang tepat bahwa “entiti ini anggota suatu himpunan “ adalah 90%, dimana hal ini bukan berarti bahwa entiti ini memiliki 90% keanggotaan dalam himpunan dan 10% bukan keanggotaan dari entiti ini. Dalam teori himpunan klasik, hal ini tidak diperbolehkan dimana sebuah elemen atau entiti ada dalam himpunan dan tidak ada dalam himpunan tersebut dalam waktu yang bersamaan. Sehingga, banyak kasus dalam aplikasi dunia nyata tidak dapat dijelaskan dan ditangani dengan teori himpunan klasik. Sebaliknya, teori himpunan fuzzy mengizinkan penggunaan keanggotaan sebagian dalam himpunan, yang dalam

teori himpunan klasik memiliki keterbatasan. (Chen *and* Pham, 2001). Berkenaan dengan kajian di atas maka persoalan-persoalan yang menyangkut sistem nonlinear, sistem yang tergantung pada waktu sangat sulit untuk di kendalikan, khususnya menggunakan sistem pengendalian konvensional. Oleh karena itu pemanfaatan metode fuzzy sangat tepat untuk menyelesaikannya. Salah satu persoalan tersebut adalah persoalan robot pengikut garis. yang merupakan bentuk robot bergerak otonom yang dirancang untuk bergerak mengikuti garis pemandu yang dibuat dengan tingkat presisi tertentu. Akash dkk (2009), meneliti robot penjejak garis dengan menelusuri garis putih pada permukaan hitam, dengan hardware menggunakan mikrokontroler, Sensor Inframerah penggerak motor Proximity, dengan bahasa pemrograman bahasa C. Penelitian ini menitikberatkan gerak robot pada mekanisme diferensial drive dengan sinyal PWM aktif tinggi sehingga penelitian ini bisa di disimpulkan pengaturan gerak motor drive dengan PWM yang dikontrol dari pemrograman tanpa ada data base logika, tidak menggunakan data latih ataupun model logika yang lain

Mayur Agarwal dkk(2008), Penelitian robot penjejak garis ini menggunakan logika PID untuk mengatur PWM sebagai kontrol penggerak motor robot., hardware yang di gunakan mikrokontroler, sensor infra merah, dengan bahasa pemrograman bahasa C. Penelitian ini bisa di simpulkan dalam pemrograman kendali robot menggunakan logika PID, namun belum ada data base logika dan hanya memfokuskan umpan balik dari respon motor dari sensor ke set point PID.

Mohammad Hadi Amoozgar (2011) Penelitian ini ini menyajikan pendekatan baru untuk mengontrol gerak robot beroda dengan menggunakan logika fuzi, juga memperhatikan aspek kinematika, performa dan konsumsi energi, hardware menggunakan mikrokontroler, sensor infra merah, penggerak motor. Kontrol logika gerak motor menggunakan logika fuzi. Untuk lebih memahami kajian pustaka yang ada maka di buat tabel sebagai berikut ;

Tabel 2-1. Tabel Pembeding Dengan Penelitian Terdahulu.

Komponen Pembeda	Nama Peneliti			
	Akash dkk (2009),	Mayur Agarwal dkk(2008),	Mohammad Amoozgar (2011)	Aji Joko*(2013)
Data base logika	Tidak menggunakan	PID	Logika fuzi tanpa PWM	Logika fuzi dengan PWM
Sensor	RF teknologi	Infra merah	Infra Merah	Ultra Bright Led
Pengamatan	Respon sensor terhadap gerakan robot	Respon sensor terhadap gerakan robot	Gerak manuver robot	Gerak robot dari tingkat kesulitan medan
Mikrokontroler	AT Mega 8	AT 89s51	Tidak di sebutkan	AT Mega 8
Bahasa Pemrograman	Bahasa C	Bahasa C	Bahasa C	Bahasa C++
Downloader	Serial Max 232	PararelPort	Tidak di sebutkan	Pararel Port

## 2.2 Landasan Teori

### 2.2.1 Himpunan Fuzi,

Konsep himpunan pada awalnya di mengerti secara intuitif saja, yaitu sebagai suatu kumpulan unsur yang di definisikan sedemikian rupa sehingga dapat di tentukan secara tegas apakah suatu obyek termasuk dalam kumpulan itu atau tidak. Hal semacam itu di sebut himpunan tegas, sebagai contohnya adalah sebagai berikut :

$$A = \{x \mid x > 6\}$$

Himpunan di atas mempunyai batas-batas yang tegas, yakni untuk nilai  $x$  lebih besar dari 6, maka  $x$  adalah anggota himpunan  $A$ , di luar nilai tersebut  $x$  tidak termasuk dalam himpunan  $A$ . Hal ini berbeda dengan himpunan Fuzzy yang berarti “kabur” atau “samar-samar”. Istilah fuzzy pada tulisan ini lebih menekankan pada bentuk kekaburan semantik. Suatu kata atau istilah dikatakan kabur (fuzzy) secara semantik apabila kata atau istilah tersebut tidak dapat didefinisikan secara tegas (benar atau salah) apakah suatu objek tertentu memiliki ciri atau sifat yang diungkapkan oleh kata atau istilah itu atau tidak. Contoh ungkapan yang menyatakan Himpunan fuzzy adalah “air itu panas” kalimat tersebut adalah relatif menurut masing-masing orang yang merasakan. Konsep tentang himpunan fuzzy pertama kali diperkenalkan oleh Profesor Lotfi A. Zadeh, seorang ilmuwan Amerika Serikat berkebangsaan Iran, dari Universitas California di Berkeley, melalui tulisannya “Fuzzy Sets” pada tahun 1965.

Secara matematis suatu himpunan kabur  $A$  dalam semesta  $X$  dapat dinyatakan sebagai himpunan pasangan berurutan seperti pada persamaan 2.1 berikut.

$$A = \{(x, \mu_A(x)) \mid x \in X\} \dots\dots\dots(2.1)$$

di mana  $\mu_A$  adalah fungsi keanggotaan dari himpunan kabur  $A$ , yang merupakan suatu pemetaan dari himpunan semesta  $X$  pada nilai keanggotaan kontinu dengan nilai antara 0 dan 1, definisi himpunan fuzzy merupakan pengembangan dari definisi himpunan tegas dalam arti jika nilai fungsi keanggotaan  $\mu_A(x)$  hanya bernilai 0 dan 1 maka  $A$  merupakan himpunan tegas dan  $\mu_A(x)$  adalah fungsi

karakteristik A. Biasanya  $x$  di sebut semesta pembicaraan atau di singkat semesta, dan  $x$  dapat berupa obyek diskrit atau kontinyu. Himpunan fuzi memiliki dua atribut, yaitu Linguistik, merupakan penamaan suatu grup yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alami, seperti mahal, sedang, murah dan sebagainya dan Numeris, yaitu suatu nilai (angka) yang menunjukkan ukuran dari suatu variable seperti 100 juta, 200 juta, 500 juta dan lain sebagainya.

#### 2.2.1.1. Fungsi Keanggotaan (*Membership Function*)

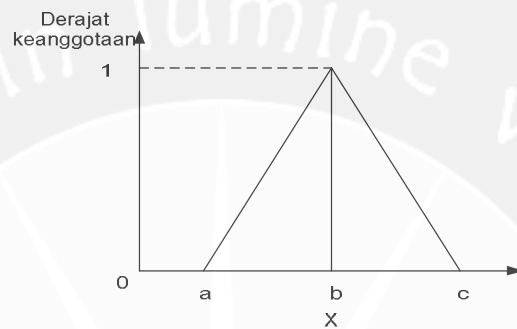
Fungsi keanggotaan adalah suatu kurva yang menunjukkan pemetaan titik-titik input data ke dalam nilai keanggotaannya yang memiliki interval antara nol sampai satu. Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi. Kebanyakan himpunan fuzi berada dalam semesta himpunan semua bilangan riil  $R$  dengan fungsi keanggotaan yang dinyatakan dalam bentuk suatu formula matematis. Formula matematis fungsi keanggotaan dalam himpunan fuzi tersebut diantaranya adalah fungsi keanggotaan segitiga, fungsi keanggotaan trapesium, fungsi keanggotaan Gauss, fungsi keanggotaan Cauchy, dan fungsi keanggotaan sigmoid.

#### 2.2.1.2. Fungsi Keanggotaan Segitiga (*Triangular Membership Function*)

Suatu fungsi keanggotaan himpunan fuzi disebut fungsi keanggotaan segitiga jika mempunyai tiga buah parameter, yaitu  $a, b, c \in R$  dengan  $a < b < c$ , dinyatakan dengan *Segitiga*  $(x; a, b, c)$  seperti pada persamaan 2.2 berikut.

$$\text{Segitiga } x; a, b, c = \begin{cases} \frac{x-a}{b-a} & \text{untuk } a \leq x \leq b \\ \frac{c-x}{c-b} & \text{untuk } b \leq x \leq c \\ 0 & \text{untuk } x \geq c \text{ atau } x \leq a \end{cases} \dots\dots\dots(2.2)$$

Gambar 2.1 menunjukkan representasi fungsi Keanggotaan Segitiga



Gambar 2.1 Jenis fungsi Keanggotaan Segitiga

2.2.1.3. Fungsi Keanggotaan Trapesium .

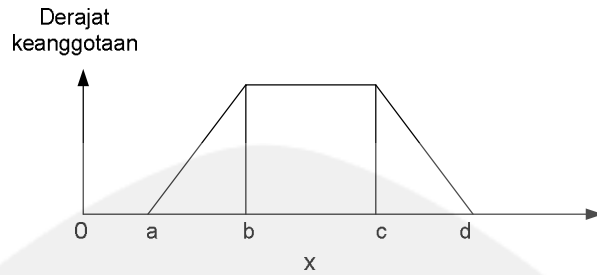
Suatu fungsi keanggotaan himpunan fuzi disebut fungsi keanggotaan trapesium jika mempunyai empat buah parameter, yaitu  $a, b, c, d \in \mathbb{R}$  dengan  $a < b < c < d$ , dinyatakan dengan *Trapesium*  $(x ; a, b, c, d)$  seperti pada persamaan 2.3 berikut.

$$\text{Trapesium} \begin{cases} \frac{x-a}{b-a} & \text{untuk } a \leq x \leq b \\ 1 & \text{untuk } b \leq x \leq c \\ \frac{d-x}{d-c} & \text{untuk } c \leq x \leq d \\ 0 & \text{untuk lainnya} \end{cases} \dots\dots\dots(2.3)$$

Fungsi keanggotaan tersebut dapat juga dinyatakan seperti persamaan 2.4 berikut.

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

Gambar 2.2 menunjukkan representasi fungsi Keanggotaan trapesium



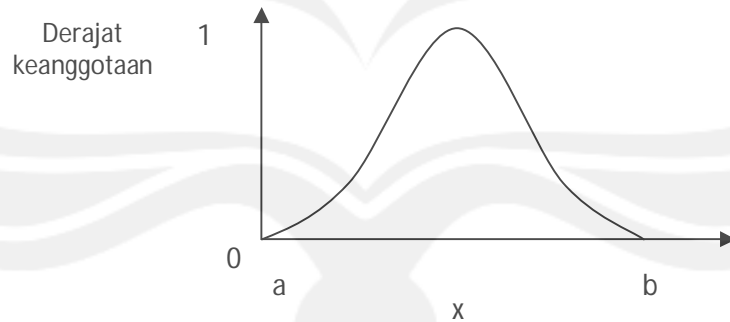
Gambar 2.2 Jenis fungsi Keanggotaan trapezium

2.2.1.4. Fungsi Keanggotaan Gauss

Suatu fungsi keanggotaan himpunan fuzzy disebut fungsi keanggotaan Gauss jika mempunyai dua buah parameter, yaitu  $a, b \in \mathbb{R}$ , dinyatakan dengan  $Gauss(x;a,b)$  seperti pada persamaan 2.5 berikut.

$$Gauss(x;a,b) = e^{-\left(\frac{x-a}{b}\right)^2} \dots\dots\dots(2.5)$$

Gambar 2.3 menunjukkan representasi fungsi Keanggotaan Gauss.



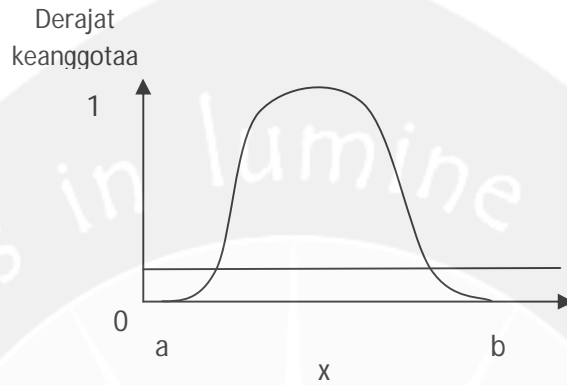
Gambar 2.3 Jenis fungsi Keanggotaan Gauss

2.2.1.5. Fungsi Keanggotaan Bell

Suatu fungsi keanggotaan himpunan fuzzy disebut fungsi keanggotaan Bell jika mempunyai tiga buah parameter, yaitu  $a, b, c \in \mathbb{R}$ , dinyatakan dengan  $bell(x ; a, b, c)$  seperti pada persamaan 2.6 sebagai berikut :

$$\text{Bell}(x; a, b, c) = \frac{1}{1 + \left| \frac{x-c^2b}{a} \right|} \dots\dots\dots(2.6)$$

Gambar 2.4 menunjukkan representasi fungsi Keanggotaan Bell



Gambar 2. 4 Jenis fungsi Keanggotaan Bell

2.2.1.6. Himpunan Bagian

Himpunan fuzi A adalah bagian dari himpunan fuzi B dapat pula di katakana A lebih kecil atau sama dengan B Jika dan hanya jika  $\mu_A(x) \leq \mu_B(x)$ . Dengan simbol matematika di nyatakan sebagai persamaan 2.7 berikut .

$$A \subseteq B \leftrightarrow \mu_A(x) \leq \mu_B(x) \dots\dots\dots(2.7)$$

2.2.1.7. Penjumlahan

Penjumlahan dua buah himpunan fuzi A dan B adalah himpunan fuzi A+B, yang didefinisikan dengan fungsi keanggotaan seperti pada persamaan 2.8 berikut.

$$\mu_{A+B}(z) = \sup_{x+y=z} \min\{\mu_A(x), \mu_B(y)\} \dots\dots\dots(2.8)$$

Misalkan dalam semesta  $X=\{1,2,3,4,5,6,7,8,9\}$  dan diketahui himpunan fuzi adalah ;

$$A = \{ (1,0.4), (2,1), (3,0.6), (4,0.1) \}.$$



$$B = \{ (2,0.2), (3,0.7), (4,1), (5,0.6) \}.$$

Maka diperoleh

$$A+B = \{ (3,0.2), (4,0.4), (5,0.7), (6,1), (7,0.6), (8,0.6), (9,0.1) \}$$

#### 2.2.1.8. Pengurangan

Pengurangan dua buah himpunan fuzzy  $A$  dan  $B$  adalah himpunan fuzzy  $A-B$ , yang didefinisikan dengan fungsi keanggotaan seperti pada persamaan 2.9 berikut.

$$\mu_{A-B}(z) = \sup_{x-y=z} \min\{\mu_A(x), \mu_B(y)\}. \dots\dots\dots(2.9)$$

Misalkan dalam semesta  $X = \{1,2,3,4,5,6,7,8,9\}$  diketahui himpunan fuzzy

$$A = \{ (6,0.4), (7,0.1), (8,0.6), (9,0.1) \}.$$

$$B = \{ (2,0.2), (3,0.7), (4,0.1), (5,0.6) \}.$$

Maka diperoleh

$$A-B = \{ (1,0.4), (2,0.6), (3,1), (4,0.7), (5,0.6), (6,0.6), (7,0.1) \}.$$

#### 2.2.1.9. Perkalian

Perkalian dua buah himpunan fuzzy  $A$  dan  $B$  adalah himpunan fuzzy  $A \cdot B$ , yang didefinisikan dengan fungsi keanggotaan seperti pada persamaan 2.10 berikut.

$$\mu_{A \cdot B}(z) = \sup_{x \cdot y = z} \min\{\mu_A(x), \mu_B(y)\}. \dots\dots\dots(2.10)$$

Misalkan dalam semesta  $X = \{0,1,2,3,4,5,6,7,8\}$  diketahui himpunan-himpunan fuzzy

$$A = \{ (1,0.4), (2,1), (3,0.6), (4,0.1) \}.$$

$$B = \{ (0,0.2), (1,1), (2,0.6) \}.$$

Maka diperoleh

$$A \cdot B = (0,0.2), (1,0.4), (2,1), (3,0.6), (4,0.6), (6,0.6), (8,0.1).$$

#### 2.2.1.10. Pembagian

Pembagian dua buah himpunan fuzzy  $A$  dan  $B$  adalah himpunan fuzzy  $A/B$ , yang didefinisikan dengan fungsi keanggotaan seperti pada persamaan 2.11 berikut.

$$\mu_{A/B} = \sup_{x/y=z} \min\{\mu_A(x), \mu_B(y)\}. \dots\dots\dots(2.11)$$

Misalkan dalam semesta  $X = \{0,1,2,3,4,5,6,7,8\}$  diketahui himpunan-himpunan fuzzy

$$A = \{ (0,0.4), (4,1), (8,0.6) \}.$$

$$B = \{ (1,0.3), (2,1), (4,0.7) \}.$$

Maka diperoleh ;

$$A/B = \{ (0,0.4), (1,0.7), (2,1), (4,0.6), (8,0.3) \}.$$

#### 2.2.1.11. Komplemen

Komplemen dari suatu himpunan fuzzy  $A$  adalah himpunan fuzzy  $A^-$ , diartikan sebagai “ $x$  tidak dekat  $A$ ”, dengan fungsi keanggotaan seperti pada persamaan 2.12 berikut.

$$\mu_{A^-} = 1 - \mu_A(x), \text{ untuk setiap } x \in X. \dots\dots\dots(2.12)$$

Misalkan dalam semesta  $X = \{-4,-3,-2,-1,0\}$  diketahui himpunan fuzzy

$$A = \{ (-4,0), (-3,0.3), (-2,0.5), (-1,0.7), (0,1) \}.$$

Maka diperoleh

$$A^- = \{ (-4,1), (-3,0.7), (-2,0.5), (-1,0.3), (0,0) \}.$$

### 2.2.1.12. Gabungan

Gabungan dua buah himpunan fuzzy  $A$  dan  $B$  adalah himpunan fuzzy  $A \cup B$ , diartikan sebagai “ $x$  dekat  $A$  atau  $x$  dekat  $B$ ”, dengan fungsi keanggotaan seperti pada persamaan 2.13 berikut.

$$\mu_{A \cup B} = \mu_A(x) \vee \mu_B(x) = \max(\mu_A(x), \mu_B(x)), \text{ untuk setiap } x \in X. \dots \dots \dots (2.13)$$

misalkan dalam semesta  $X = \{-3, -2, -1, 0, 1, 2\}$  diketahui himpunan-himpunan fuzzy ;

$$A = \{(-3, 0.3), (-2, 0.7), (-1, 1), (0, 0.5), (1, 0.2), (2, 0)\}.$$

$$B = \{(-3, 0), (-2, 0.1), (-1, 0.4), (0, 0.6), (1, 0.8), (2, 1)\}.$$

Maka diperoleh ;

$$A \cup B = \{(-3, 0.3), (-2, 0.7), (-1, 1), (0, 0.6), (1, 0.8), (2, 1)\}.$$

### 2.2.1.13. Irisan

Irisan dua buah himpunan fuzzy  $A$  dan  $B$  adalah himpunan fuzzy  $A \cap B$ , diartikan sebagai “ $x$  dekat  $A$  dan  $x$  dekat  $B$ ”, dengan fungsi keanggotaan seperti pada persamaan 2.14 berikut.

$$\mu_{A \cap B} = \mu_A(x) \wedge \mu_B(x) = \min(\mu_A(x), \mu_B(x)), \text{ untuk setiap } x \in X. \dots (2.14)$$

misalkan dalam semesta  $X = \{-3, -2, -1, 0, 1, 2\}$  diketahui himpunan-himpunan fuzzy

$$A = \{(-3, 0.3), (-2, 0.7), (-1, 1), (0, 0.5), (1, 0.2), (2, 0)\}.$$

$$B = \{(-3, 0), (-2, 0.1), (-1, 0.4), (0, 0.6), (1, 0.8), (2, 1)\}.$$

maka diperoleh ;

$$A \cap B = \{(-3, 0), (-2, 0.1), (-1, 0.4), (0, 0.5), (1, 0.2), (2, 0)\}.$$

Dua buah himpunan fuzzy dikatakan beririsan apabila irisan kedua himpunan fuzzy tersebut tidak sama dengan himpunan kosong. Apabila irisan dua buah himpunan

fuzi sama dengan himpunan kosong, maka kedua himpunan fuzi tersebut dikatakan lepas.

### 2.2.2. Aturan Fuzi JIKA-MAKA

Aturan fuzi JIKA-MAKA implikasi fuzi A dan B dengan fungsi keanggotaan  $\mu_A(x)$  dan  $\mu_B(x)$  merupakan himpunan fuzi dalam bentuk,

“ JIKA X adalah A MAKA Y adalah B” Dengan A dan B merupakan nilai linguistik, adalah himpunan fuzi pada semesta pembicaraan  $x$  dan  $y$ . Pernyataan  $x$  adalah A sering di sebut anteseden atau premis, sedangkan  $y$  adalah B di sebut konsekuen atau kesimpulan, contoh „JIKA tekanan tinggi Maka volume kecil. Sebelum menggunakan aturan jika maka untuk pemodelan dan analisis sebuah sistem, pertama harus di ketahui terlebih dahulu maksud dari ekspresi “JIKA  $x$  adalah A MAKA  $y$ ” adalah B yang dapat di gantikan dengan notasi  $A \rightarrow B$ . Pada dasarnya, ekspresi tersebut menguraikan hubungan antara dua variable  $x$  dan  $y$ , dan untuk situasi tersebut aturan fuzi JIKA-MAKA di pandang sebagai relasi fuzi biner R pada ruang perkalian  $X \times Y$ . Relasi biner R merupakan pengembangan perkalian Cartesian klasik, dengan masing masing elemen  $(x,y) \in X \times Y$ , berhubungan dengan tingkat keanggotaan yang di beri notasi  $\mu_R(x,y)$ . dengan kata lain, relasi fuzi biner R dapat di gambarkan sebagai himpunan fuzi dengan semesta  $X \times Y$  dan himpunan fuzi tersebut di tandai dengan adanya fungsi keanggotaan dua dimensi  $\mu_R(x, y)$ .

### 2.2.3. Penalaran Fuzi

Pada Tahapan ini menalar nilai masukan untuk menentukan nilai keluaran sebagai bentuk keputusan. Ada 2 cara penalaran yaitu menurut operator zadeh yang terdiri dari operator yang menggunakan logika AND, OR dan NOT yang di realisasikan dengan maks –min, maks-dot dan sebagainya, dan penalaran Non Zadeh yang menggunakan operator yang di dasarkan pada tranformasi aritmatika, seperti mean, product dan bounded sum. Penggunaan operator mean mempunyai dua kriteria yaitu tidak terlalu sensitive terhadap nilai minimum dan maksimum, dan distribusi urutan nilai kebenaran yang di evaluasi tidak mempengaruhi operasi. Rumus yang di gunakan untuk operator mean ini adalah seperti pada persamaan 2.15 berikut .

$$\mu_{A \cap B} = (\mu_A [x] + \mu_B [y])/2 \dots \dots \dots (2.15)$$

### 2.2.4. Pengendalian Logika Fuzi

Pengendali logika fuzi (PLF) merupakan suatu teknik pengendalian yang menggunakan logika fuzi. Pada saat ini sistem pengendalian berdasarkan logika fuzi banyak sekali di gunakan dalam kehidupan sehari hari. Sebagai contoh di gunakan dalam terknologi otomotif, transportasi, peralatan rumah tangga, aplikasi industri dan lain sebagainya. Pada perancangan sebuah pengendali logika fuzi, ada beberapa faktor yang harus di perhatikan :

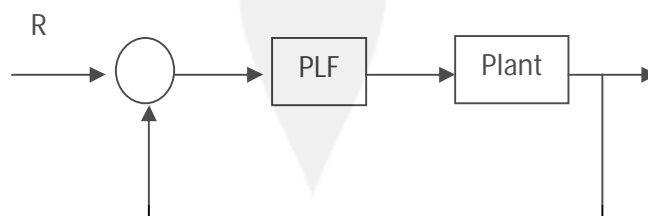
- 1) Masukan dan keluaran yang sesungguhnya serta semesta pembicaraan, yaitu rentang nilai yang mungkin di ambil.
- 2) Faktor-faktor skala dari variabel masukan dan keluaran.

- 3) Fungsi keanggotaan fuzzy yang di gunakan dalam menyusun nilai fuzzy untuk setiap variable masukan keluaran.
- 4) Basis aturan pengendalian fuzzy

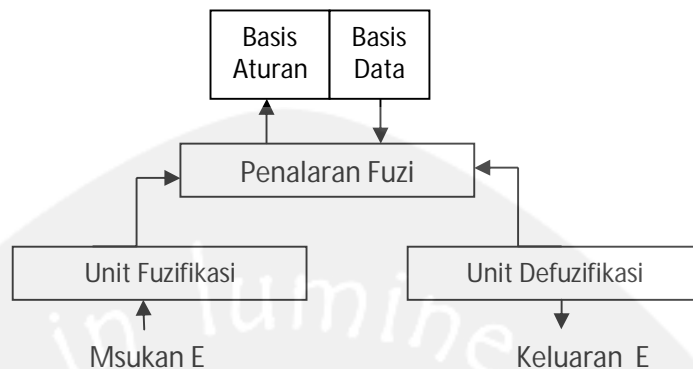
Konstruksi fungsi keanggotaan dan aturan pengendalian fuzzy merupakan inti dalam perancangan pengendali logika fuzzy. Bergantung pada tujuan perancangan , suatu pengendali logika fuzzy mempunya pengaturan dini atau kemampuan belajar.

#### 2.2.4.1. Struktur Dasar dari Pengendalian Logika Fuzzy

Suatu pengendali logika fuzzy dapat di tunjukan pada gambar 2.5 yang merupakan suatu sistem pengendali untai tertutup. Pada gambar 2.6 di perlihatkan elemen – elemen utama pengendali logika fuzzy yaitu unit fuzifikasi, unit penalaran logika fuzzy, basis pengetahuan dan unit defuzifikasi. Bergantung pada tujuan perancangan, perbedaan tipe pengendali logika fuzzy dapat di susun. Sebagai contoh pengendali logika fuzzy boleh mempunyai bilangan yang pasti dari aturan kendali fuzzy ( sebuah basis pengetahuan statis). Atau boleh mempunyai kemampuan belajar melalui modifikasi dari basis pengetahuan ( sebuah basis pengetahuan dinamis)



Gambar 2.5 Sistem Pengendalian Logika fuzzy



Gambar 2.6. Struktur dasar dari pengendali logika fuzi

Basis pengetahuan fuzi terdiri dari dua tipe informasi utama :

1. Suatu basis data yang mendefinisikan fungsi keanggotaan dari himpunan himpunan fuzi yang di gunakan sebagai nilai untuk setiap variabel sistem.
2. Basis aturan yang pada dasarnya untuk memetakan nilai fuzi masukan ke nilai fuzi keluaran.

Variabel sistem terdiri dari dua tipe utama , variable masukan (E) di ukur dari proses pengontrolan dan variable keluaran (U) di gunakan oleh pengendali logika fuzi untuk mengontrol proses. Setiap Variabel sistem di gunakan untuk menampilkan aturan, nilai nilai yang di perbolehkan harus di definisikan sebagai himpunan fuzi. Nilai aktual di kirim ke proses pengendalian yang biasanya tegas, dan operasi fuzifikasi dan defuzifikasi dibutuhkan untuk memetakan ke dan dari nilai fuzi yang digunakan oleh pengendali logika fuzi. Untuk data masukan yang tegas, dasar strategi fuzifikasi meliputi hal-hal sebagai berikut :

1. Mendapatkan nilai-nilai tegas dari variabel-variabel masukan .
2. Memetakan nilai-nilai tegas dari variabel masukan ke dalam semesta pembicaraan

3. Mengubah data yang dipetakan ke dalam fuzi tunggal apabila dikehendak, selain itu, menerjemahkan setiap data ke dalam bentuk linguistic yang sesuai sebagai label dari suatu himpunan fuzi yang didefinisikan untuk variabel.

Untuk keluaranya, proses defuzifikasi lebih terlibat, biasanya terjadi sebagai bagian dari tahap akhir dalam penarikan kesimpulan fuzi. Hal tersebut meliputi bobot dan kombinasi bilangan hasil himpunan fuzi dari proses penarikan kesimpulan fuzi dalam suatu perhitungan yang memberikan suatu nilai tegas tunggal untuk setiap keluaran.

Bagian basis aturan dari basis pengetahuan terdiri dari sejumlah aturan fuzi yang menunjukkan hubungan pengendalian. Aturan-aturan tersebut ditunjukkan dalam suatu format JIKA-MAKA. Sebagai contoh, untuk single input-Single-output (SISO) pengendali logika fuzi, aturan fuzi dapat di tuliskan “ JIKA  $x$  adalah PB maka  $y$  adalah NB” dengan  $x$  merupakan variabel masukan,  $y$  adalah variabel keluaran, PB adalah satu dari himpunan fuzi yang di defisikan untuk  $x$  dalam semesta  $x$ , dan NB adalah satu dari himpunan fuzi yang di defisikan untuk  $y$  dalam semesta  $y$ .

Basis pengetahuan fuzi mungkin statis atau dinamis. Basis pengetahuan fuzi dinamis di butuhkan untuk mempelajari atau mengatur sendiri tingkah laku oleh pengendali logika fuzi .

Modul penalaran logika fuzi dalam aspek pengambilan keputusan. Modul ini memberikan kesimpulan fuzi untuk aksi kendali fuzi dengan megevaluasi basis pengetahuan untuk masukan yang di fuzifikasikan. Selama menarik kesimpulan fuzi langkah-langkah berikut dilibatkan untuk setiap aturan fuzi.



1. Menentukan tingkatan yang sesuai antara data masukan fuzzy dan definisi himpunan fuzzy untuk setiap variabel masukan .
2. Menyelesaikan perhitungan kekuatan penyulutan untuk setiap baris aturan pada tingkatan yang sesuai dan penghubung (misalnya AND, OR) yang digunakan oleh variabel masukan dalam kalimat-kalimat anteseden dari aturan atau dengan operator pengganti zadeh yaitu mean
3. memperoleh keluaran pengendalian yang berdasarkan pada perhitungan kekuatan penyulutan dan definisi himpunan fuzzy untuk setiap variabel keluaran pada kalimat konsekuen dari setiap aturan .

#### 2.2.4.2. Variabel Sistem dan Parameter Fuzzy

Kompleksi sistem dari sistem fuzzy bergantung pada variasi jumlah variabel masukan dan keluaran. Suatu sistem dengan  $n$  variabel masukan dan  $m$  variabel keluaran dapat dikatakan sebagai suatu sistem  $n$ -masukan- $m$ -keluaran. Pada kasus dengan  $n = 1$  dan  $m = 1$ , sistem dapat disebut sebagai Single-Input-Single-Output (SISO) atau Masuk-Tunggal-Keluaran-Tunggal. Dalam kasus dengan  $n \geq 2$  dan  $m \geq 1$ , sistem fuzzy dikatakan Multi-Input-Single-Output (MISO) atau Masukan-Banyak-Keluaran-Banyak. Apabila pengendali logika fuzzy dirancang untuk menggantikan pengendali PID konvensional dalam sistem MISO, variabel-variabel masukan pengendali logika fuzzy adalah keadaan galat (state error) dengan notasi  $e$ , jumlah Keadaan galat (sum of state error) notasinya  $\Sigma e$ , dan turunan dari keadaan galat (derivative of state error) dengan notasi  $de$ . Semuanya dapat ditulis sebagai vektor variabel masukan  $E$ , dengan  $E = ( e, \Sigma e, de)$ .

Variabel keluaran pengendali logika fuzzy adalah isyarat kendali U seperti tegangan, arus, dan lain sebagainya .

Himpunan fuzzy untuk setiap variabel sistem didefinisikan dalam bentuk linguistik seperti PB (positive big) atau positif besar, PM (positive medium) atau positif sedang, PS (positive small) atau positif kecil, ZE (zero) atau nol, NM (negative medium) atau negatif sedang, NB (negative big) atau negatif besar, dan lain sebagainya. Fungsi keanggotaan untuk setiap himpunan fuzzy menentukan arti dari nilai linguistik, dan didefinisikan pada semesta pembicaraan dari variabel yang dibicarakan .

Ada dua cara untuk mendefinisikan keanggotaan himpunan fuzzy, yaitu numeris dan fungsi. Pendefinisian cara numeris menyatakan derajat fungsi keanggotaan dari himpunan fuzzy sebagai vektor bilangan yang besarnya bergantung pada tingkat diskritisasi dalam semesta pembicaraan. Sedangkan pendefinisian cara fungsi menyatakan fungsi keanggotaan suatu himpunan fuzzy dalam bentuk fungsi, seperti fungsi S, fungsi  $\pi$ , fungsi segitiga, dan lain sebagainya. Pendefinisian numeris atau fungsi boleh untuk menentukan derajat keanggotaan pada himpunan fuzzy dalam pengendali logika fuzzy

#### 2.2.4.3. Basis Pengetahuan

Basis pengetahuan terdiri dari basis aturan dan basis data. Basis aturan berisi aturan kendali fungsi untuk memenuhi tujuan pengendalian. Basis data berisi definisi yang penting dari parameter parameter fuzzy sebagai himpunan

fuzi dengan fungsi keanggotaan yang didefinisikan dalam semesta pembicaraan untuk setiap variabel.

#### 2.2.4.4. Basis Aturan

Basis aturan pengendalian fuzi adalah kumpulan aturan pengendalian dan kebijakan pengendalian oleh pakar. Pada pengendalian logika fuzi, aturan pengendalian fuzi biasanya berbentuk turan JIKA-MAKA. Sebagai contoh adalah untuk sistem MISO basis aturan pengendalian fuzi dapat ditulis “Jika  $x_1$  adalah  $A_{11}$  dan  $x_m$  adalah  $A_{1m}$  maka  $Y$  adalah  $B_1$ ” Dengan  $x_1$  sebagai variabel masukan sistem, seperti galat (error) galat derivatif, dan lain lain.  $A_{ij}$  sebagai himpunan fuzi untuk  $x_j$ , seperti PB, PM, PS, ZE dan lain sebagainya.  $Y$  melambangkan variabel keluaran, misalnya arus isyarat penggerak motor DC,  $B_i$  adalah himpunan fuzi untuk  $y$ , seperti PB, PM, ZE, NS, NB dan lain lain.

#### 2.2.4.5. Jumlah Aturan Pengendali

Algoritma pengendali fuzi seharusnya selalu mampu mampu menyimpulkan yang layak untuk sembarang dalam semesta pembicaraan. Sifat ini disebut sebagai sifat kelengkapan. Sifat kelengkapan ini dimasukkan ke dalam aturan pengendalian fuzi melalui pengalaman perancangan dan pengetahuan rekayasa. Selain itu biasanya juga membuat himpunan fuzi saling tumpang tindih untuk memastikan bahwa setiap nilai dapat masuk ke dalam beberapa himpunan fuzi. Di samping itu perancangan aturan perlu dilakukan dengan hati hati untuk menjamin bahwa setiap masukan akan menghasilkan

beberapa tanggapan. Jika banyaknya himpunan fuzzy atau predikat bagi setiap variabel masukan dilambangkan dengan  $m$  dan banyaknya variabel masukan sistem diberi lambang  $n$ , maka jelaslah bahwa  $m^n$  aturan yang berbeda diperlukan untuk kelengkapan dalam pendekatan sistem pakar konvensional. Sebagai contoh, jika banyak himpunan fuzzy tiap variabel masukan sistem  $m$  adalah tujuh dan banyak variabel masukan  $n$  adalah tiga, maka dibutuhkan 343 aturan.

Berbeda dengan sistem pakar konvensional, basis aturan pengendali logika fuzzy hanya menggunakan aturan yang sedikit. Disamping aturan pengendalian dalam pengendalian logika fuzzy masih dapat dikurangi dengan adanya himpunan fuzzy yang saling tumpang tindih dan penggunaan pendekatan yang sesuai dalam penarikan kesimpulan fuzzy. Hal ini akan mengakibatkan waktu perhitungan yang pendek. Namun sampai saat ini belum ada prosedur khusus untuk menentukan beberapa aturan pengendalian fuzzy yang optimal.

#### 2.2.5. Diskritisasi Semesta Pembicaraan

Pada beberapa aplikasi industri, sebagian besar nilai yang diukur bersifat analog dan dapat dianggap kontinu dalam rentang tertentu. Nilai-nilai tersebut dapat dibuat diskrit dengan menggunakan pengubah analog ke digital agar dapat digunakan sebagai masukan bagi sistem komputer digital. Proses ini sering ditunjukkan pada kuantisasi, dan diskretisasi semesta pengukuran ke dalam jumlah segmen tertentu. Segmen ini secara efektif mengganti nilai-nilai analog yang asli, dan membentuk semesta pembicaraan untuk variabel, yang bersifat diskret.

himpunan fuzzy sekarang didefinisikan dengan menentukan nilai derajat keanggotaan untuk setiap elemen dari semesta diskret baru.

Pemilihan jumlah sekmen berpengaruh dalam menentukan seberapa baik level pengendalian yang dihasilkan. Oleh karena itu jumlah segmen akan mempengaruhi resolusi pengendalian. Sedangkan pemilihan level kuantisasi berdasarkan pada resolusi pengendalian yang akan di capai.

#### 2.2.6. Pembagian Fuzzy Ruang Masukan / Keluaran

Berbentuk ruang masukan fuzzy dengan mempertimbangkan semesta pembicaraan masukan sedangkan dalam kalimat konsekuen dari aturan pengendalian fuzzy berbentuk ruang keluaran fuzzy. Banyaknya himpunan fuzzy yang di definisikan untuk setiap variable merupakan penentu kecermatan pengendalian yg di berikan oleh pengedali logika fuzzy. Kemudian, jumlah total dari himpunan fuzzy untuk variable masukan sistem akan menentukan batasan pada jumlah maksimum aturan pengendalian fuzzy yang dapat di buat. Akhirnya, untuk menentukan pembagian ruang yang optimal dapat dilakukan dengan cara coba – coba.

#### 2.2.7. Strategi Defuzzifikasi

Proses defuzzifikasi adalah suatu himpunan fuzzy yang diperoleh dari komposisi aturan-aturan fuzzy, sedangkan *output* yang dihasilkan merupakan suatu bilangan pada domain himpunan fuzzy dalam *range* tertentu, maka harus dapat

diambil suatu nilai *crisp* tertentu sebagai *output*. Model strategi defuzzifikasi adalah sebagai berikut

## 2. Metode Centroid

Pada metode ini, solusi *crisp* diperoleh dengan cara mengambil titik pusat ( $z^*$ ) daerah fuzi. Secara umum dirumuskan seperti pada persamaan 2.14. berikut.

$$z^* = \frac{\int z \cdot \mu(z) dz}{\int \mu(z) dz} \dots\dots\dots (2.14)$$

## 3. Metode COG (*Centre Of Gravity*)

Metode COG ini nilai keluaran tegas diperoleh dari jumlah hasil kali keluaran Fuzi untuk setiap himpunan fuzi keluaran dengan posisi singleton pada sumbu x dibagi dengan jumlah keluaran Fuzi untuk setiap himpunan Fuzi keluaran. Rumus metode defuzzifikasi COG adalah sebagai.

$$COG = \frac{(FUZZY OUTPUT) \times (Sglt n_{Pos})}{\Sigma(Fuzi Output)} \dots\dots\dots (2.15)$$

## 4. Metode *Mean of Maximum* (MOM)

Pada metode ini, solusi *crisp* diperoleh dengan cara mengambil nilai rata-rata domain yang memiliki nilai keanggotaan maksimum.

## 5. Metode *Largest of Maximum* (LOM)

Pada metode ini, solusi *crisp* diperoleh dengan cara mengambil nilai terbesar dari domain yang memiliki nilai keanggotaan maksimum.

### 2.2.8. Robotika

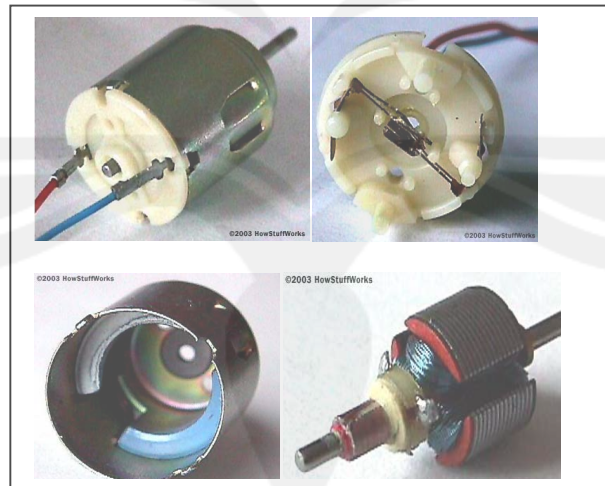
Robot adalah sebuah alat mekanik yang dapat melakukan tugas fisik, baik menggunakan pengawasan dan kontrol manusia, ataupun menggunakan program yang telah didefinisikan terlebih dulu (kecerdasan buatan). Robot biasanya digunakan untuk tugas yang berat, berbahaya, pekerjaan yang berulang dan kotor. Biasanya kebanyakan robot industri digunakan dalam bidang produksi. Penggunaan robot lainnya termasuk untuk pembersihan limbah beracun, penjelajahan bawah air dan luar angkasa, pertambangan, pekerjaan "cari dan tolong" (*search and rescue*), dan untuk pencarian tambang. Belakangan ini robot mulai memasuki pasaran konsumen di bidang hiburan, dan alat pembantu rumah tangga, seperti penyedot debu, pemotong rumput, dan pengantar barang.

Dalam penelitian ini di gunakan sebuah “Robot Pengikut Garis (*Line Follower*)” Robot pengikut garis (*line follower*) adalah robot yang dapat berjalan secara otomatis mengikuti garis yang mempunyai warna berbeda dengan backgroundnya (Garis hitam dengan background putih). Bagian – bagian robot pengikut garis adalah sebagai berikut,

#### 2.2.8.1. Motor DC

Motor penggerak menggunakan motor DC, motor DC digunakan karena mudah dikontrol arah putaran dan kecepatannya. Untuk bergerak bebas maju, mundur, kanan dan kiri, maka digunakan konfigurasi dua buah motor DC, motor satu menggerakkan roda kanan dan motor dua menggerakkan roda kiri. Motor DC memiliki 2 pin input, yaitu tegangan dan ground. Dengan membalik masukan

tegangan dan ground kita akan membalik putaran dari motor DC. Sebuah motor DC memiliki kumparan kawat di dalam slot sebuah silinder yang terbuat dari bahan feromagnetik, silinder ini diberi nama *armature*. Armature dipasang pada suatu bentuk dudukan (*bearing*) dan bebas berputar. Dudukan *armature* adalah sebuah medan magnet yang dihasilkan oleh magnet permanent atau arus yang dialirkan melalui kumparan kawat yang dinamakan kumparan medan. Kedua magnet ini, magnet permanent maupun *elegtromagnet*, disebut sebagai *stator* (bagian yang diam). Ketika arus mengalir melalui kumparan armature, sebuah konduktor berarus yang berada tegak lurus terhadap sebuah medan magnet akan mengalami gaya. Gaya magnet akan bekerja pada kumparan tersebut dan mengakibatkan putaran. Gambar motor DC beserta komponennya di tunjukan pada Gambar 2.7 berikut.



Gambar 2.7 Motor DC biasa beserta komponennya  
(Sumber: <http://electronics.howstuffworks.com/motor2.htm>)

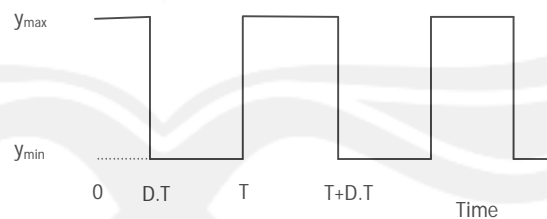
Kecepatan putaran motor DC dapat diubah dengan cara mengubah besar arus pada kumparan *armature*. Akan tetapi, karena sumber tegangan tetap biasanya



digunakan sebagai input ke kumparan, perubahan arus yang diperlukan seringkali diperoleh melalui penggunaan sebuah rangkaian elektronik. Rangkaian ini dapat mengontrol nilai rata-rata tegangan, dengan cara mengubah interval waktu untuk menghasilkan tegangan DC yang bervariasi, misalnya menggunakan rangkaian *PWM ( Pulse Width Mode)* .

#### 2.2.8.2. *PWM (Pulsa With Mode)*

Metode pengendalian motor DC dengan PWM adalah metode yang cukup efektif untuk mengendalikan kecepatan motor DC. PWM bekerja dengan cara membuat gelombang persegi yang memiliki perbandingan pulsa *high* terhadap pulsa *low* yang telah tertentu, biasanya diskalakan dari 0 hingga 100%. Perbandingan pulsa *high* terhadap *low* ini di kenal dengan nama gelombang kotak atau *duty cycle* seperti pada gambar 2.8 sebagai berikut.



Gambar 2.8. Gelombang kotak dengan  $y_{\max}$  dan  $y_{\min}$

dengan memperhatikan gambar 2.8. gelombang kotak dengan  $y_{\max}$  dan  $y_{\min}$ , maka nilai rata rata dari gelombang kotak tersebut di tuliskan seperti pada persamaan 2.16 berikut.

$$\bar{y} = \frac{1}{T} \int_0^T f(t) dt \dots \dots \dots (2.16)$$

Jika  $f(t)$  adalah gelombang kotak, maka nilai  $y_{max}$  adalah dari  $0 < t < D.T$  dan nilai  $y_{min}$  dari  $D.T < t < T$ , dari pernyataan di atas di dapat :

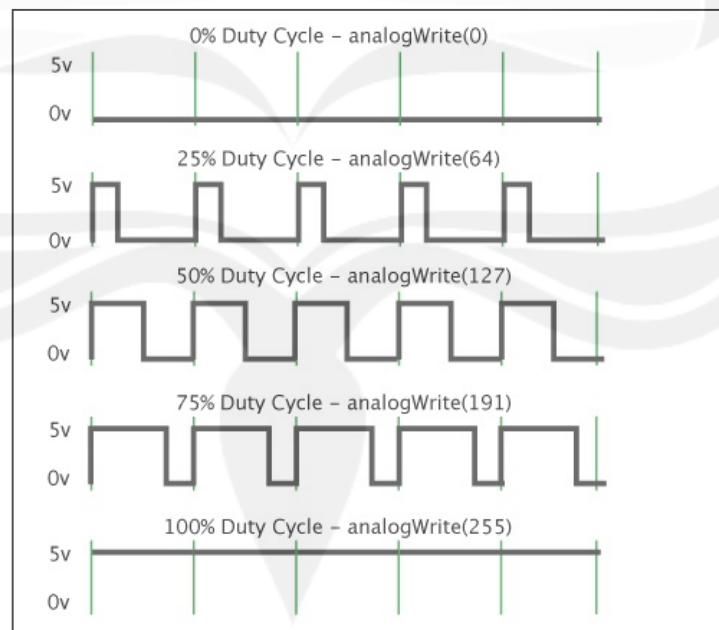
$$\bar{y} = \frac{1}{T} \left( \int_0^{DT} y_{max} dt + \int_{DT}^T y_{min} dt \right)$$

$$\bar{y} = \frac{D.T y_{max} + T(1-D)y_{min}}{T}$$

$\bar{y} = D.y_{max} + (1-D)y_{min}$ , bila  $y_{min} = 0$ , maka persamaan menjadi,

$$\bar{y} = D.y_{max}, \dots\dots\dots(2.17)$$

sehingga dari persamaan 2.17 maka nilai  $\bar{y}$  tergantung pada *duty cycle*. *Duty cycle* merupakan prosentase keadaan logika *high (pulse)* dalam satu periode sinyal. Satu siklus diawali oleh transisi *low to high* dari sinyal dan berakhir pada transisi berikutnya. Gambar 2.9 menunjukkan gambar PWM dengan *Duty Cycle* yang berbeda untuk menghasilkan nilai analog secara digital



Gambar 2.9 Gambar Pulsa PWM

Selama satu siklus jika waktu sinyal pada keadaan *high* sama dengan *low* maka sinyal mempunyai *duty cycle* 50%, Bila  $\bar{y}$  dianggap sebagai tegangan rata

rata keluaran dari  $D$  dan  $y_{max}$ , yang merupakan Prosentase *Duty Cycle* dan tegangan referensi maka kita bisa menghitung tegangan keluaran sebagai berikut :

$$V_r = DC\% \times V_{ref} \dots \dots \dots (2.18)$$

$V_r$  = Tegangan keluaran

DC% = Prosentase Duty

$V_{ref}$  = Cycle x Tegangan referensi

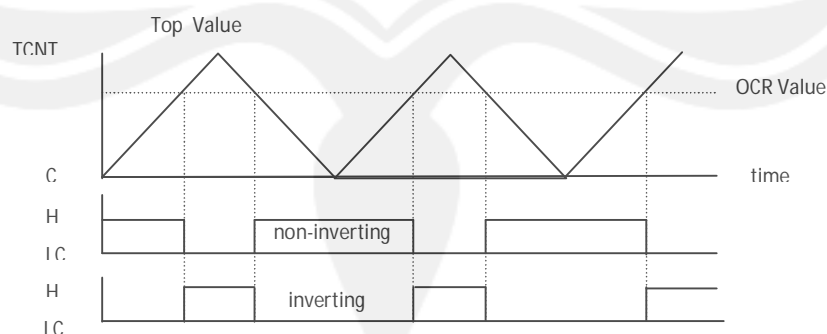
Sehingga jika *Duty Cycle* = 0%, dengan tegangan referensi = 5 volt, maka nilai

tegangan yang di dihasilkan adalah  $\frac{0}{100} \times 5 \text{ volt} = 0 \text{ volt}$ , Jika *Duty Cycle* = 25 %,

maka nilai tegangan yang di dihasilkan adalah  $\frac{25}{100} \times 5 \text{ volt} = 1,25 \text{ volt}$

### 2.2.8.3. Pengaturan PWM pada Mikrokontroler ATMEGA 8.

PWM yang di dikeluarkan oleh mikrokontroler ATMEGA 8 di lakukan dengan mengatur nilai dari OCR pada timer yang bersangkutan, mode PWM ATMEGA 8 mempunyai 2 tipe keluaran yaitu *non inverting* dan *inverting* PWM seperti pada gambar 2. 10 berikut.



Gambar 2.10. non inverting dan inverting PWM

Dari gambar 2.9 tampak jelas jika nilai OCR di rubah maka langsung mengubah lebar pulsa. Dalam mikrokontroler ATMEGA 8 bagian yang berperan untuk

membangkitkan dan pengaturan PWM adalah .

1. *Timer /Counter* untuk membangkitkan 2 pulsa PWM
2. Port/Pin keluaran yang bersangkutan
3. Register TCNT1, OCR1A, OCR1B untuk menghasilkan dan mengatur *duty cycle* keluaran PWM
4. Register TCCR1A, TCCR1B untuk mengatur mode dan cara kerja timer

Frekuensi PWM di formulasikan dengan persamaan 2.3 sebagai berikut ;

$$f_{pwm} = \frac{f_{osc}}{N(1+TOP)} \dots \dots \dots (2.19)$$

$f_{pwm}$  = frekuensi pwm

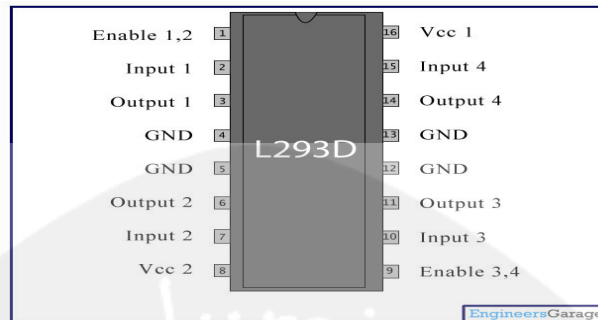
$f_{osc}$  = frekuensi osilator

N = skala clock

TOP = Nilai TCNT1 maximum

#### 2.2.8.4. *Driver* IC LM 293D

*Driver* motor pada robot pengikut garis dapat dibuat dengan menggunakan sebuah *IC H-Bridge* jenis L293D seperti pada gambar 2.11, dengan table spesifikasi maksimum seperti pada table 2.2. *Driver* motor inilah yang memproses sinyal pulsa PWM (*Pulse Width Modulation*) dari mikrokontroler.



Gambar 2.11 Konfigurasi PIN IC LN293D

Table 2.2. Parameter maksimal IC L293D

Pin No	Function	Name
1	Enable pin for Motor 1; active high	Enable 1,2
2	Input 1 for Motor 1	Input 1
3	Output 1 for Motor 1	Output 1
4	Ground (0V)	Ground
5	Ground (0V)	Ground
6	Output 2 for Motor 1	Output 2
7	Input 2 for Motor 1	Input 2
8	Supply voltage for Motors; 9-12V (up to 36V)	Vcc <sub>2</sub>
9	Enable pin for Motor 2; active high	Enable 3,4
10	Input 1 for Motor 1	Input 3
11	Output 1 for Motor 1	Output 3
12	Ground (0V)	Ground
13	Ground (0V)	Ground
14	Output 2 for Motor 1	Output 4
15	Input2 for Motor 1	Input 4
16	Supply voltage; 5V (up to 36V)	Vcc <sub>1</sub>

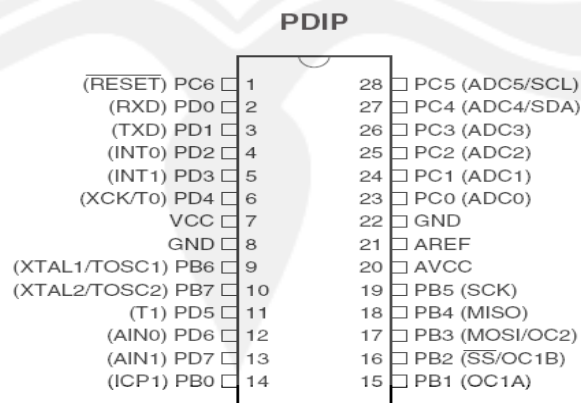
#### 2.2.8.5. Sensor Photodiode ( detector masukan)

Malvino (1996), *photodiodes* dibuat dari semikonduktor dengan bahan yang populer adalah *silicon (Si)* atau *galium arsenida (GaAs)*, dan yang lain meliputi *InSb*, *InAs*, *PbSe*. Ketika sebuah photon (satu satuan energi dalam cahaya) dari sumber cahaya diserap, hal tersebut membangkitkan suatu elektron dan menghasilkan sepasang pembawa muatan tunggal, sebuah elektron dan sebuah hole, di mana suatu hole adalah bagian dari kisi-kisi semikonduktor yang kehilangan elektron. Arah arus yang melalui sebuah semikonduktor adalah kebalikan dengan gerak muatan pembawa. cara tersebut didalam sebuah

photodiode digunakan untuk mengumpulkan photon yang menyebabkan pembawa muatan (seperti arus atau tegangan) mengalir atau terbentuk di bagian bagian elektroda.

#### 2.2.8.6. Control Unit (Mikrokontroler AVR ATMEGA 8)

Mikrokontroler sebagai sebuah “*one chip solution*” pada dasarnya adalah rangkaian terintegrasi (*Integrated Circuit-IC*) yang telah mengandung secara lengkap berbagai komponen pembentuk sebuah komputer. Seperti *RAM*, *ROM*. AVR ATMEGA8 adalah mikrokontroler CMOS 8-bit berarsitektur AVR RISC yang memiliki 8K byte *in-Sistem Programmable Flash*. Jika dibandingkan dengan ATMEGA8L perbedaannya hanya terletak pada besarnya tegangan yang diperlukan untuk bekerja. Untuk ATMEGA8 tipe L, mikrokontroler ini dapat bekerja dengan tegangan antara 2,7 - 5,5 V sedangkan untuk ATMEGA8 hanya dapat bekerja pada tegangan antara 4,5 – 5,5 V. Konfigurasi Pin mikrokontroler ATMEGA 8 di tunjukan pada gambar 2.11 berikut.



Gambar 2.12. Konfigurasi Pin ATMEGA 8

ATMEGA8 memiliki 28 pin, yang masing-masing pinnya memiliki fungsi yang berbeda-beda baik sebagai port maupun fungsi yang lainnya. Berikut penjelasan fungsi dari masing-masing kaki ATMEGA8.

1. VCC, Merupakan *supply* tegangan
2. GND, Merupakan ground untuk semua komponen yang membutuhkan grounding.
3. Port B (PB7...PB0), Didalam Port B terdapat XTAL1, XTAL2, TOSC1, TOSC2. Jumlah Port B adalah 8 buah pin, mulai dari pin B.0 sampai dengan B.7. Tiap pin dapat digunakan sebagai *input* maupun *output*. Port B merupakan sebuah 8-bit *bi-directional* I/O dengan internal pull-up resistor. Sebagai *input*, pin-pin 7 yang terdapat pada port B yang secara eksternal diturunkan, maka akan mengeluarkan arus jika *pull-up* resistor diaktifkan. Khusus PB6 dapat digunakan sebagai *input* Kristal (*inverting oscillator amplifier*) dan *input* ke rangkaian *clock* internal, bergantung pada pengaturan *Fuse bit* yang digunakan untuk memilih sumber *clock*. Sedangkan untuk PB7 dapat digunakan sebagai *output* Kristal (*output oscillator amplifier*) bergantung pada pengaturan *Fuse bit* yang digunakan untuk memilih sumber *clock*. Jika sumber clock yang dipilih dari *oscillator internal*, PB7 dan PB6 dapat digunakan sebagai I/O atau jika menggunakan *Asynchronous Timer/Counter2* maka PB6 dan PB7 (TOSC2 dan TOSC1) digunakan untuk saluran *input timer*.
4. Port C (PC5...PC0); Port C merupakan sebuah 7-bit *bi-directional* I/O port yang di dalam masingmasingpin terdapat *pull-up* resistor. Jumlah pin nya

hanya 7 buah mulai dari *pin* C.0 sampai dengan *pin* C.6. Sebagai keluaran/*output port* C memiliki karakteristik yang sama dalam hal menyerap arus (*sink*) ataupun mengeluarkan arus (*source*).

5. RESET/PC6, Jika RSTDISBL *Fuse* diprogram, maka PC6 akan berfungsi sebagai *pin* I/O. *Pin* ini memiliki karakteristik yang berbeda dengan *pin-pin* yang terdapat pada *port* C lainnya. Namun jika RSTDISBL *Fuse* tidak diprogram, maka *pin* ini akan berfungsi sebagai input reset. Dan jika *level* tegangan yang masuk ke *pin* ini rendah dan pulsa yang ada lebih pendek dari pulsa minimum, maka akan menghasilkan suatu kondisi reset meskipun *clock*-nya tidak bekerja.
6. Port D (PD7...PD0); Port D merupakan 8-bit *bi-directional* I/O dengan internal *pull-up* resistor. Fungsi dari port ini sama dengan port-port yang lain. Hanya saja pada port ini tidak terdapat kegunaan-kegunaan yang lain. Pada *port* ini hanya berfungsi sebagai masukan dan keluaran saja atau biasa disebut dengan I/O.
7. AVcc, *Pin* ini berfungsi sebagai *supply* tegangan untuk ADC. Untuk *pin* ini harus dihubungkan secara terpisah dengan VCC karena *pin* ini digunakan untuk analog saja. Bahkan jika ADC pada AVR tidak digunakan tetap saja disarankan untuk menghubungkannya secara terpisah dengan VCC. Jika ADC digunakan, maka AVcc harus dihubungkan ke VCC melalui *low pass filter*.
8. AREF, adalah kaki masukan referensi bagi A/D Converter. Untuk operasionalisasi ADC, suatu level tegangan antara AGND dan AVcc harus diberikan ke kaki ini.



9. AGND, adalah kaki untuk analog ground. Hubungkan kaki ini ke GND, kecuali jika board memiliki analog ground yang terpisah.

10. PORTx, adalah register data untuk output di PORTx. I/O-pin yang termasuk PORTx dan telah dikonfigurasi sebagai output akan memiliki nilai yang sesuai di PORTx bit register. Ketika sebuah pin dikonfigurasi sebagai input maka bit untuk PORTx pin digunakan untuk mengaktifkan internal pull-up resistor untuk pin. Nilai-nilai input tidak dapat dibaca melalui register PORTx.

11. PINx digunakan untuk membaca nilai dari I/O-pin yang dimiliki PORTx.

Nilai yang dibaca dari PINx akan termasuk state apa pun ketika I/O-pin dikonfigurasi sebagai output. Ketika melangkah melalui source code dalam simulator menekan F11, akan terlihat paling sedikit PORTB toggling signifikan antara 0 dan 1.

12. DDRx menentukan arah data setiap bit di PORTx.

Bila bit pada DDRx diset (aktif) maka pin di PORTx berfungsi sebagai output. PORTx akan berfungsi sebagai input, apabila DDRx-nya di clear

#### 2.2.9.6. Bahasa Pemrograman Mikrokontroler

Dewasa ini penggunaan bahasa pemrograman kelas tinggi (seperti C++, Basic, Pascal, Forth dan sebagainya) semakin populer dan banyak digunakan untuk memprogram sistem mikrokontroler. Berdasarkan sifatnya yang sangat fleksibel dalam hal keleluasaan pemrogram untuk mengakses perangkat keras, Bahasa C++ paling cocok dibandingkan bahasa-bahasa pemrograman kelas tinggi lainnya. Bahasa C++ dikembangkan pertama kali oleh Dennis Ritchie dan Ken Thomson pada tahun 1972, Bahasa C++ merupakan salah satu bahasa

pemrograman yang paling populer untuk pengembangan program-program aplikasi yang berjalan pada sistem mikroprocessor (komputer). Karena kepopulerannya, vendor-vendor perangkat lunak kemudian mengembangkan kompiler C++ sehingga menjadi beberapa varian berikut: Turbo C, Borland C, Microsoft C, Power C, Zortech C dan lain sebagainya. Untuk menjaga portabilitas, compilercompiler C++ tersebut menerapkan ANSI C (ANSI: *American National Standards Institute*) sebagai standar bakunya. Perbedaan antara compiler-compiler tersebut umumnya hanya terletak pada pengembangan fungsi-fungsi library serta fasilitas IDE (*Integrated Development Environment*)–nya saja. CodeVisionAVR pada dasarnya merupakan perangkat lunak pemrograman microcontroller keluarga AVR berbasis bahasa C++. Ada tiga komponen penting yang telah diintegrasikan dalam perangkat lunak ini, kompiler C++, IDE dan program generator. Berdasarkan spesifikasi yang dikeluarkan oleh perusahaan pengembangnya, Kompiler C++ yang digunakan hampir mengimplementasikan semua komponen standar yang ada pada bahasa C++ standar ANSI (seperti struktur program, jenis tipe data, jenis operator, dan library fungsi standar-berikut penamaannya). Tetapi walaupun demikian, dibandingkan bahasa C++ untuk aplikasi komputer, compiler C++ untuk mikrokontroller ini memiliki sedikit perbedaan yang disesuaikan dengan arsitektur AVR tempat program C+ tersebut ditanamkan (*embedded*). Khusus untuk library fungsi, disamping library standar (seperti fungsi-fungsi matematik, manipulasi String, pengaksesan memori dan sebagainya), CodeVisionAVR juga menyediakan

fungsi-fungsi tambahan yang sangat bermanfaat dalam pemrograman antarmuka AVR dengan perangkat luar yang umum digunakan dalam aplikasi kontrol

