

# Implementasi Metode Fuzzy Logic untuk Kontrol Pergerakan Autonomous Mobile Robot pada Aplikasi Soccer Robot

Moh. Hisyam Fithrony<sup>#1</sup>, Endah Suryawati Ningrum, ST, MT<sup>#2</sup>, Bambang Sumantri, ST, MSc<sup>#3</sup>

<sup>#</sup>Jurusan Teknik Elektronika, Politeknik Elektronika Negeri Surabaya  
Kampus PENS-ITS Sukolilo, Surabaya

<sup>1</sup>mhskibar@student.eepis-its.edu

<sup>2</sup>endah@eepis-its.edu

<sup>3</sup>bambang@eepis-its.edu

## Abstrak

Robot soccer merupakan aplikasi robotika di bidang *entertainment* dan edukasi. Penerapan kecerdasan buatan (*artificial intelligent*) sangat membantu untuk menjadikan robot lebih hidup. Obyek dari Proyek Akhir ini adalah sebuah Autonomous Mobile Robot dengan kemampuan untuk mengejar target berupa sebuah bola dengan warna tertentu (bola tenis meja warna oranye) dan kemampuan untuk menghindari halangan dengan sensor pendeteksi berupa ultrasonik. Untuk melakukan gerakan atau *manuver*, robot menggabungkan kedua kemampuan tersebut dengan teknik pengambilan keputusan memanfaatkan algoritma logika fuzzy pembuat keputusan (*fuzzy logic decision maker*). Dipilihnya logika fuzzy pada proyek akhir ini, karena logika fuzzy merupakan kontrol yang mengandalkan intuisi dari pembuat sistem, sehingga pengalaman pembuat sistem sangat berpengaruh dengan hasil perancangan aturan logika fuzzy. Dari hasil pengujian, diketahui bahwa keberhasilan sistem dapat mencapai 100% jika lapangan uji warnanya kontras dengan warna targetnya, pencahayaan konstan dan pandangan robot tidak tertutup oleh halangan.

Kata kunci : logika fuzzy, mobile robot, *autonomous*, pengambil keputusan, robot soccer, *obstacle avoidance*, *pursue the target*

## I. PENDAHULUAN

*Soccer robot* adalah jenis robot yang bergerak secara otonom yang dibuat layaknya pemain sepak bola, sehingga perlu adanya kecerdasan buatan yang ditanam dalam otak (prosesornya). *Soccer robot* dibuat untuk keperluan riset, hiburan, maupun industri. Dasar perancangan kecerdasan soccer robot bisa juga diadopsi dari mobile robot dengan beberapa modifikasi.

Merancang dan menerapkan kecerdasan buatan untuk *mobile robot* tidaklah mudah, tetapi dengan berkembangnya teknologi, banyak metode-metode yang dapat digunakan untuk mendukung pembuatan kecerdasan buatan tersebut. Jenis kontrolnya juga sangat beragam, bergantung dari obyek yang akan dibuat.

Kecerdasan buatan untuk *mobile robot* yang ada saat ini seperti yang telah diterapkan pada robot BRAM (Beginner's Robot Autonomous Mobile), masih parsial dan bekerja secara serial atau secara hirarki yang memiliki kelemahan dalam hal proses yang tidak

bisa bersamaan. Sehingga perlu dikembangkan teknologi untuk pengambilan keputusan pergerakan yang menggabungkan beberapa input data kemampuan-kemampuan yang dimiliki, dengan adanya teknik pengambilan keputusan tersebut, robot soccer dapat memiliki kemampuan mengejar target sambil menghindari halangan.

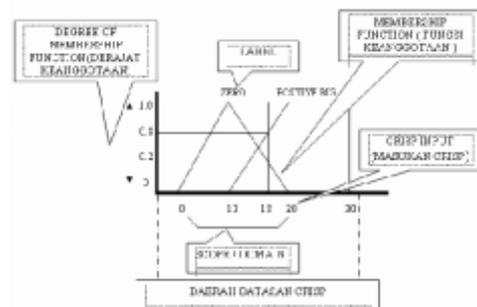
Satu diantara teknologi yang dapat digunakan untuk menggabungkan kemampuan-kemampuan tersebut adalah fuzzy logic. Dimana penggunaan teknologi *fuzzy logic* saat ini telah berkembang pesat dan meluas di berbagai aplikasi, mulai dari analisis informasi sampai proses industri, peralatan elektronika dan lain-lain. Pada tahun-tahun terakhir ini, *fuzzy logic* sudah menjadi bahan dasar bagi perancangan teknologi *mobile robot*, serta mempunyai nilai ekonomis tersendiri, setelah dilakukan pengembangan pada kemampuan robot tersebut. Karena dalam kenyataannya, manusia dalam melakukan aksi hanya berpikir atau memperhitungkan secara pendekatan dan secara linguistik dan intuitif.

Oleh karena itu, pada proyek akhir ini disajikan implementasi kendali *fuzzy logic* pada *mobile robot* untuk melakukan pengambilan keputusan (*decision maker*) dari berbagai masukan yang ada. Untuk saat ini, masukan yang digunakan adalah kemampuan mengejar target (*pursue the target*) berupa bola dan menghindari halangan (*obstacle avoidance*). Dimana *fuzzy logic* digunakan untuk mengolah kedua masukan dari data kemampuan mengejar bola dan menghindari halangan menjadi keluaran berupa aksi gerak robot melalui aktuatornya.

## II. TEORI PENUNJANG

### A. Fuzzy Logic [3][4][5]

Dalam pemanfaatannya, *fuzzy logic* memerlukan tiga proses dasar, yakni fuzzyfikasi, evaluasi rule, dan defuzzyfikasi.



Gambar 2.1 Konsep dasar logika fuzzy

o **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.

o **Label** adalah nama deskriptif yang digunakan untuk mengidentifikasi sebuah fungsi keanggotaan.

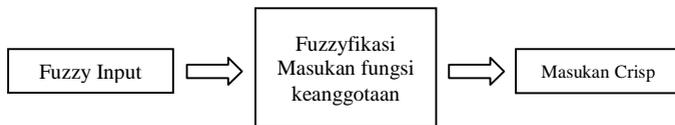
o **Fungsi Keanggotaan** adalah mendefinisikan fuzzy set dengan memetakan masukan crisp dari domainnya ke derajat keanggotaan.

o **Masukan Crisp** adalah masukan yang tegas dan tertentu

o **Lingkup/Domain** adalah lebar fungsi keanggotaan. Jangkauan konsep, biasanya bilangan, tempat dimana fungsi keanggotaan dipetakan.

o **Daerah Batasan Crisp** adalah jangkauan seluruh nilai yang mungkin dapat diaplikasikan pada variabel sistem.

Pada proses fuzzyfikasi, masukan berupa data valid yang biasa disebut *Crisp Input*, yang akan diolah menjadi *fuzzy input*



Gambar 2.2 Blok diagram fuzzyfikasi

Setelah proses fuzzyfikasi selesai, dilakukan evaluasi rule, rule yang akan dibuat ini, bergantung dari pengalaman dari perancang sistem, semakin banyak pengalaman, maka semakin baik pula hasil yang didapat. Berikut adalah bahasa yang biasa digunakan dalam tahap evaluasi rule fuzzy :

- Label /nilai linguistik : low, medium, high
- Operator penghubung : AND, OR, NOT
- Kondisi : baik, sedang, buruk

Contoh penggunaan rule dalam aplikasi :

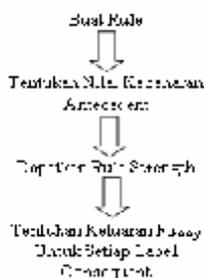
If halangan US kiri tidak ada *and* halangan US kanan tidak ada, *then* maju

If halangan US kiri ada *and* halangan US kanan tidak ada, *then* belok kanan

If halangan US kiri tidak ada *and* halangan US kanan ada, *then* belok kiri

If halangan US kiri ada *and* halangan US kanan ada, *then* mundur

**Prosedur Evaluasi Rule**



Gambar 2.3 Prosedur evaluasi rule

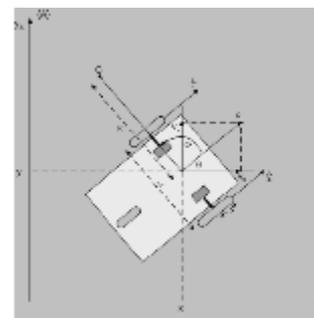
Tahap terakhir adalah defuzzyfikasi, yang mengambil keluaran fuzzy dari setiap label, untuk dijadikan *output crisp*. Menggunakan metode defuzzyfikasi COG, keluaran nilai *singleton* dikombinasikan menggunakan bobot rata-rata. Rumus COG, untuk perhitungan *reduksi singleton* adalah

$$KeluaranCrisp(y) = \frac{\sum (keluaranFuzzy[i]) \times (posisiSingletonSumbuX[i])}{\sum (keluaranFuzzy[i])}$$

**B. Differential drive [1][2]**

Salah satu jenis *mobile robot* yang umum digunakan, terutama untuk dioperasikan dalam ruangan adalah *mobile robot* dengan pengemudian atau sistem penggerak diferensial (*differential drive*).

Secara teknis, robot jenis ini pada dasarnya memiliki dua roda utama yang masing-masing digerakan oleh penggerak tersendiri (umumnya berupa motor DC magnet permanent dengan gear-pereduksi yang berfungsi untuk memperkuat torsi motor), selain itu robot ini dilengkapi juga dengan satu atau dua buah roda castor yang ditempatkan dibagian belakang robot yang berfungsi sebagai penyeimbang. Gambar 4 memperlihatkan arsitektur robot dilihat dari bagian atas: Jika kedua roda penggerak tersebut berputar dengan kecepatan yang sama maka robot tersebut akan bergerak dengan arah yang lurus, sedangkan jika kecepatan salah satu roda lebih lambat maka robot akan bergerak membentuk kurva dengan arah lintasan menuju salah satu roda yang bergerak lebih lambat.



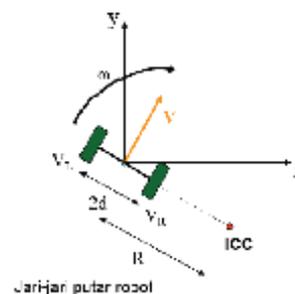
Gambar 2.4 Posisi dan Orientasi Mobile Robot Dalam Sistem Koordinat Cartesian

Dengan mengetahui jari-jari roda (r), maka kita dapat mengubah kecepatan linear dari masing-masing roda ( $V_r$  &  $V_l$ ) menjadi kecepatan angular roda kiri ( $\omega_l$ ) dan roda kanan ( $\omega_r$ ), menggunakan dua rumus berikut :

$$\omega_r(t) = V_r(t) / r \dots\dots\dots (1)$$

$$\omega_l(t) = V_l(t) / r \dots\dots\dots (2)$$

Ketika robot melakukan gerak memutar (rotasi) sesaat pada titik pusat rotasi tertentu (ICC), maka kecepatan rotasi di setiap titik robot tersebut akan selalu sama. Ilustrasinya seperti pada gambar berikut :



Gambar 2.5 Pergerakan robot berdasar titik pusat rotasi tertentu

Dengan demikian, berlaku persamaan untuk menghitung kecepatan rotasi dari robot tersebut.

$$V_R = (R+d) \quad (3)$$

$$V_L = (R-d) \quad (4)$$

kemudian,

$$= (V_R - V_L) / 2d \quad (5)$$

$$R = d (V_R + V_L) / (V_R - V_L) \quad (6)$$

sehingga kecepatan robot :

$$V = R = (V_R + V_L) / 2 \quad (7)$$

Untuk mendapatkan integrasi kecepatan robot adalah sebagai berikut :

$$V_x = V(t) \cos ( t) \quad (8)$$

$$V_y = V(t) \sin ( t) \quad (9)$$

Dimana :  $V_x$  = Kecepatan di sumbu x

$V_y$  = Kecepatan di sumbu y

$$x(t) = \int V(t) \cos( t) dt \quad (10)$$

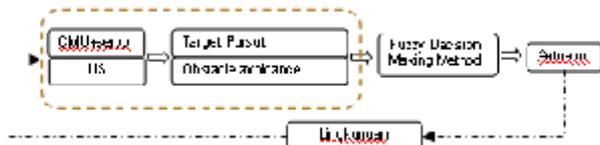
$$y(t) = \int V(t) \sin( t) dt \quad (11)$$

$$t) = ( t) dt \quad (12)$$

### III. PERENCANAAN SISTEM

Pada perencanaan sistem ini, terdiri dari blok diagram perencanaan sistem, perencanaan mekanik, perencanaan hardware dan perencanaan software

#### A. Perencanaan sistem

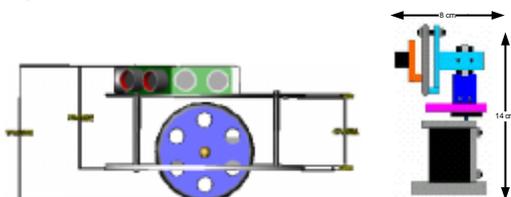


Gambar 3.1 Blok diagram sistem

Sesuai dengan gambar diatas, sistem akan menggunakan visual servo untuk mendapatkan informasi target, kemudian ditambahkan dengan SRF04 untuk mendapatkan informasi halangan. Sehingga kedua data tersebut dapat digunakan sebagai masukan fuzzy logic untuk menentukan aksi gerak robot.

#### B. Perencanaan Mekanik

Perancangan mekanik pada sistem ini, didasarkan pada desain yang compact dan efisien, sehingga dibuat dengan ukuran yang relatif kecil berdasarkan batasan lebar lapangan uji coba dan bentuk silinder agar penempatan sensor lebih mudah dan robot lebih lincah dalam bermanuver. Badan robot menggunakan bahan acrylic dan aluminium, karena bahan-bahan tersebut cocok untuk digunakan dalam mendesain robot, karena ringan tapi kuat dan mudah dibentuk. Untuk meminimalisir bobot robot, PCB hardware utama juga digunakan sebagai bagian dari badan robot. Untuk mendukung manuver geraknya, maka robot ini di desain menggunakan kinematika gerak differential drive, sehingga lebih mudah dalam kontrolnya. Penggunaan bahan-bahan tersebut dikarenakan bahan tersebut dapat dengan mudah didapat dan dengan harga yang terjangkau

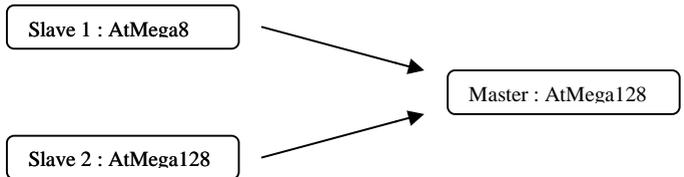


Gambar 3.2 Desain mekanik

#### C. Perencanaan Hardware

Pada sistem ini menggunakan ATMEGA 128 sebagai master untuk melakukan akuisisi data, penyimpanan data maupun untuk melakukan komunikasi dengan hardware lainnya. Pada ATMEGA 128 mempunyai konfigurasi sebagai berikut :

- Frekuensi clock sebesar 11.0592 MHz agar komunikasi serial tidak mengalami error yang mengakibatkan lost data serial
- Menggunakan 2 buah komunikasi serial USART untuk komunikasi dengan slave ATMEGA128 dan slave ATMEGA8.



Gambar 3.3 Blok diagram sistem perangkat elektronika

#### D. Perencanaan Software

Software yang digunakan dalam proyek akhir ini dibangun dengan menggunakan bahasa pemrograman jenis C, untuk compiler yang dipakai menggunakan codevisionAVR karena memiliki wizard untuk menghasilkan konfigurasi yang dibutuhkan. Untuk membantu perancangan sistem fuzzynya, digunakan software Fuzzy Generator.

#### SOFTWARE FUZZY KONTROL

Untuk software fuzzy ini, memiliki empat buah input crisp, yakni posisi target, jarak target, posisi halangan dan jarak halangan.

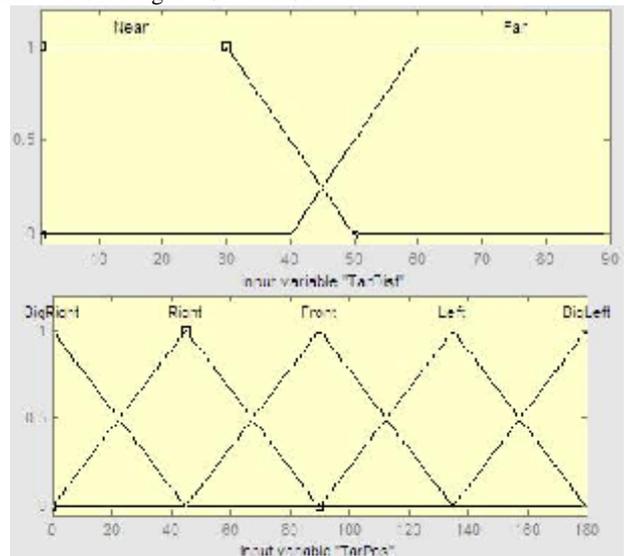
Berikut rincian membership function inputnya :

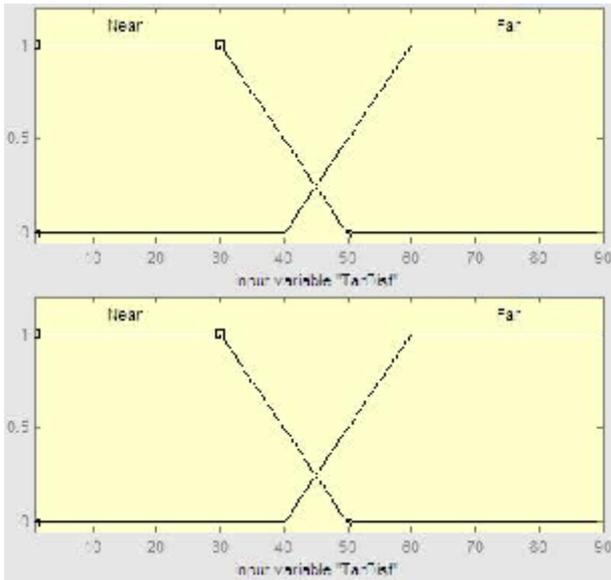
Posisi target : TBigL, TL, TF, TR, TBigR

Arah hindar halangan : OBigL, OL, OF, OR, OBigR

Jarak target : TN dan TF

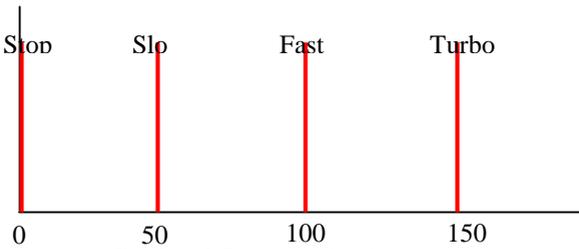
Jarak bebas halangan : ON dan OF





Gambar 3.4 Membership function input

Kemudian untuk membership outputnya, digunakan singleton dengan 4 anggota. Berikut rincian membership function outputnya :



Gambar 3.5 Membership function output

Dari membership function yang sudah dibuat, maka dapat dirumuskan rule fuzzyfikasinya sesuai kondisi yang ada.

Tabel 3.1 Rule Fuzzyfikasi

```

rule[ 1] = and(T_Near,O_Near);
rule[ 2] = and(T_Far,O_Near,O_BigL);
rule[ 3] = and(T_Far,T_Near,O_Left);
rule[ 4] = and(T_Far,O_Near,O_Front);
rule[ 5] = and(T_Far,O_Near,O_Right);
rule[ 6] = and(T_Far,O_Near,O_BigR);
rule[ 7] = and(T_Near,T_BigL,O_Far);
rule[ 8] = and(T_Near,T_Left,O_Far);
rule[ 9] = and(T_Near,T_Front,O_Far);
rule[10] = and(T_Near,T_Right,O_Far);
rule[11] = and(T_Near,T_BigR,O_Far);
rule[12] = and(T_Far,T_BigL,O_Far);
rule[13] = and(T_Far,T_Left,O_Far);
rule[14] = and(T_Far,T_Front,O_Far);
rule[15] = and(T_Far,T_Right,O_Far);
rule[16] = and(T_Far,T_BigR,O_Far);

```

Karena *membership function inputnya* masing-masing memiliki lima buah member untuk masukan posisi dan dua buah member untuk masukan jarak, sehingga didapatkan 100 kondisi, maka untuk mendapatkan keseluruhan kondisi, ada 16 rule kemungkinan yang dikombinasikan dengan tujuh buah membership output.

$$\text{Out} = \frac{o\text{Stop}.p\text{Stop} + o\text{Slow}.p\text{Slow} + o\text{Fast}.p\text{Fast} + o\text{Turbo}.p\text{Turbo}}{o\text{Stop} + o\text{Slow} + o\text{Fast} + o\text{Turbo}}$$

#### IV. PENGUJIAN

Pada bab ini akan dibahas tentang pengujian yang telah dilakukan. Untuk tahap-tahap pengujian yang akan dilakukan adalah sebagai berikut :

- Pengujian halangan u-trap  
 Dari hasil pengujian (tabel 4.1), dapat dilihat keberhasilannya 80% untuk menghindari halangan model u-trap. Error yang terjadi karena robot terjebak pada sudut, sehingga pembacaan kacau atau terlalu dekat dengan halangan.

Tabel 4.1 Tabel keberhasilan menghindari u-trap

Pengujian	Hasil	% berhasil
1	Sukses	100%
2	Sukses	100%
3	Sukses	100%
4	Sukses	100%
5	Gagal	0%
6	Sukses	100%
7	Sukses	100%
8	Gagal	0%
9	Sukses	100%
10	Sukses	100%
<b>Total % keberhasilan</b>		<b>80%</b>

- Pengujian pencarian target  
 Dari hasil pengujian (tabel 4.2), dapat dilihat keberhasilannya 100% untuk mencari target pada lantai gelap, namun pada lantai terang, keberhasilan terbesar dari lima kali pengambilan sampel adalah 80%. Hal ini dikarenakan lantai yang terang, membuat pembacaan RGB bola atau target membias atau terpantul cahaya.

Tabel 4.2 Tabel keberhasilan mencari target

Pengujian	Lantai	% berhasil
1	Terang	20%
2	Terang	60%
3	Terang	80%
4	Terang	80%
5	Terang	40%
6	Gelap	100%
7	Gelap	100%
8	Gelap	100%
9	Gelap	100%
10	Gelap	100%

➤ Pengujian mengejar target

Dari hasil pengujian (tabel 4.3), dapat dilihat keberhasilannya 100% untuk mengejar target, baik target diam maupun bergerak.

Tabel 4.3 Tabel keberhasilan mengejar target

Pengujian	Target	% berhasil
1	Diam	100%
2	Diam	100%
3	Diam	100%
4	Diam	100%
5	Diam	100%
6	Gerak	100%
7	Gerak	100%
8	Gerak	100%
9	Gerak	100%
10	Gerak	100%

➤ Pengujian gabungan

Dari hasil pengujian (tabel 4.4), dapat dilihat keberhasilannya 80% untuk kemampuan gabungan menggunakan fuzzy, dari 10 kali percobaan, dua kali kegagalan dikarenakan target tertutupi halangan setelah running (percobaan 1) dan halangan terlalu dekat setelah running (percobaan 3).

Tabel 4.4 Tabel keberhasilan uji gabungan

Pengujian	Hasil	% berhasil
1	Gagal	0%
2	Sukses	100%
3	Gagal	0%
4	Sukses	100%
5	Sukses	100%
6	Sukses	100%
7	Sukses	100%
8	Sukses	100%
9	Sukses	100%
10	Sukses	100%
<b>Total % keberhasilan</b>		<b>80%</b>

titik tengah robot dan menghindari halangan jika halangan tidak kurang dari 5 cm

4. Kalibrasi RGB CMUcam berpengaruh terhadap sistem, karena ketidakstabilan data path mengejar target membuat data pembacaan target tidak sesuai pula
5. Prosentase keberhasilan sistem mencapai 100% jika lapangan uji warnanya kontras (gelap) dengan warna target dan pandangan robot terhadap target tidak tertutup oleh halangan

Dan saran untuk pengembangan sistem ini adalah :

1. Perlu adanya penyempurnaan *path obstacle avoidance* dan *path pursue the target*, baik secara *hardware* (jenis sensor) maupun *software* (metode dan algoritma)

#### DAFTAR PUSTAKA

- [1] Borenstein, Johan, “ *Navigating Mobile Robots Systems and Techniques*”, 1996.
- [2] Budiharto, Widodo : “*Robotika (teori+implementasi)*”, Penerbit Andi, 2010.
- [3] Jamsidi M.;Vadie, N.;Ross, T.; “*Fuzzy Logic And Control*”, Prentice Hall, New Jersey, 1993
- [4] Klir, George J dan Yuan Bo, “*Fuzzy sets and Fuzzy Logic: Theory And Applications*”, New Jersey: Prentice-Hall. Inc, 1995.
- [5] Kusumadewi, Sri dan Hari Purnomo : “*Aplikasi Logika Fuzzy untuk pendukung keputusan*”, Graha Ilmu, 2010.
- [6] Kuswadi, Son ; “*Kendali Cerdas*”, EEPIS Press, 2000.
- [7] Naba, Agus : “*Belajar Cepat Fuzzy Logic menggunakan MATLAB*”, Penerbit Andi, 2009
- [8] Sousa, J.M.C and U. Kaymak ”*Fuzzy Decision Making in Modelling and Control*”, World Scientific Pub. Co. : Singapore,2002.
- [9] [www.mathworks.com/help/toolbox/fuzzy/fp4856.html](http://www.mathworks.com/help/toolbox/fuzzy/fp4856.html) diakses pada 10 Januari 2011.

#### V. PENUTUP

Kesimpulan dari keseluruhan perancangan, pengujian sistem ini adalah sebagai berikut :

1. Dari pengujian sensor ultrasonik, didapatkan pembatasan bahwa respon kritis jarak halangan minimal 3 cm.
2. Dari pengujian sensor CMUcam, didapatkan kesimpulan bahwa pemberian led ultrabright membantu dalam menstabilkan pencahayaan local, dan pengambilan region bawah lebih stabil dari region atas, namun memiliki bias yang besar, sehingga menjadi kurang fokus.
3. Dari pengujian free running, didapatkan kesimpulan bahwa robot dapat mengejar target dalam jarak 5-60 cm radius