

Makalah Seminar Tugas Akhir

PENGENDALIAN ORIENTASI *WEBCAM* SEBAGAI PENGAWAS RUANGAN DENGAN METODE KONTROL *FUZZY*

Masri'an^[1], Iwan Setiawan, ST, MT^[2], Darjat, ST, MT^[2]

Jurusan Teknik Elektro, Fakultas Teknik, Universitas Diponegoro
Jln. Prof. Sudharto, Tembalang, Semarang, Indonesia

Abstrak

Suatu sistem keamanan yang baik sangat diperlukan untuk membantu mengurangi, mencegah dan mengatasi permasalahan kriminalitas yang terus meningkat. Saat ini kamera video telah diterapkan untuk mendukung sistem keamanan. Kamera akan merekam gambar atau obyek didepannya serta menyimpannya dalam media penyimpanan yang dimiliki. Sistem keamanan yang diterapkan pada area yang luas akan membutuhkan perangkat monitoring dalam jumlah besar. Dengan menggunakan perangkat monitoring yang dapat bergerak bebas untuk memantau suatu ruangan, maka jumlah perangkat yang banyak dapat dikurangi, terlebih kalau perangkat tersebut dapat bergerak secara otomatis sehingga tidak diperlukan operator untuk menggerakkan perangkat monitoring tersebut.

Dalam tugas akhir ini akan dirancang sistem pengendalian orientasi webcam yang difungsikan sebagai alat monitoring ruangan yang mampu mengikuti suatu obyek yang bergerak didepannya serta memosisikan obyek tersebut ditengah-tengah perekaman gambar sehingga dihasilkan mutu perekaman gambar yang bagus. Metode kontrol yang dipakai dalam pengendalian webcam adalah Fuzzy Logic Controller. Sedangkan bahasa pemrograman yang digunakan adalah Microsoft Visual Basic pada PC dan bahasa C embedded dalam mikrokontroler ATMEGA8535.

Dari hasil pengujian didapatkan bahwa Pengendalian orientasi kamera mempunyai respon yang cukup baik terutama pada obyek diam yaitu dengan error steady state rata-rata ± 30 pixel. Error steady state terjadi karena karakteristik motor yang mempunyai dead zone yang cukup lebar. Motor bisa berputar jika diberikan minimal $\pm 33\%$ dutycycle PWM. Jika dalam area monitoring tidak terdapat obyek maka sistem akan melakukan scanning sampai ditemukan obyek. Pada pengawasan obyek bergerak, kamera akan melakukan pergerakan sama dengan arah pergerakan obyek baik untuk obyek berkecepatan lambat maupun cepat. Kecepatan maksimum webcam adalah $\pm 0,8$ RPM. Prinsip dan respon pada obyek mejemuk hampir sama dengan obyek tunggal.

Kata kunci : Pengendalian webcam, Fuzzy Logic Controller, mikrokontroler ATMEGA8535.

I PENDAHULUAN

1.1 Latar Belakang

Meningkatnya angka kriminalitas sekarang ini menuntut orang untuk lebih waspada dalam menjaga diri serta menjaga harta benda yang dimiliki. Suatu sistem keamanan yang baik sangat diperlukan untuk membantu mengurangi, mencegah dan mengatasi permasalahan kriminalitas. Sistem keamanan dapat diterapkan di rumah, tempat pameran, toko, kantor dan tempat-tempat lainnya.

Saat ini kamera video telah diterapkan untuk mendukung sistem keamanan. Kamera akan merekam gambar serta menyimpannya dalam media yang dimiliki. Untuk aplikasi yang lebih maju, kamera dapat difungsikan seperti mata pada manusia. Apabila diketahui ada orang tak dikenal masuk maka sistem keamanan akan mengambil suatu tindakan seperti menutup pintu, membunyikan sirene atau bahkan menembak pelaku.

Sistem keamanan yang diterapkan pada area yang luas akan membutuhkan perangkat *monitoring* dalam jumlah besar. Dengan menggunakan perangkat monitoring yang dapat bergerak bebas untuk memantau suatu ruangan, maka jumlah perangkat yang banyak dapat dikurangi, terlebih kalau perangkat tersebut dapat bergerak secara

otomatis sehingga tidak diperlukan operator untuk menggerakkan perangkat monitoring tersebut.

1.2 Tujuan

Tujuan yang ingin dicapai dalam pembuatan tugas akhir ini adalah untuk merencanakan, merancang dan membuat sistem pengendalian orientasi *webcam* yang difungsikan sebagai alat *monitoring* ruangan yang mampu mengikuti suatu obyek yang dikehendaki dalam area monitoring serta memosisikan obyek tersebut ditengah-tengah perekaman gambar.

1.3 Pembatasan Masalah

Pembatasan masalah pada tugas akhir ini yaitu:

1. Perancangan perangkat keras serta perangkat lunak untuk mengendalikan *webcam*.
2. Aktuator yang digunakan untuk menggerakkan *webcam* berupa motor DC yang dikendalikan dengan prinsip PWM menggunakan mikrokontroler ATMEGA8535.
3. Bahasa pemrograman yang digunakan untuk membangun *interfacing* adalah Visual Basic.
4. *Background* atau dinding ruangan yang digunakan mempunyai warna yang berlainan terhadap obyek.
5. *Webcam* hanya mempunyai 1 derajat kebebasan yaitu berputar dalam bidang horizontal.

¹ Mahasiswa Jurusan Teknik Elektro UNDIP

² Staf Pengajar Jurusan Teknik Elektro UNDIP

II DASAR TEORI

2.1 Webcam

Webcam adalah kamera kecil yang dapat menangkap video serta gambar, serta dapat menyimpannya dalam *hard drive* komputer. Memasang *webcam* sangatlah mudah karena *webcam* memiliki fitur fungsionalitas USB pasang dan penggunaan yang mudah, dan semuanya itu hanya membutuhkan beberapa langkah mudah untuk memasang dan mengoperasikan *webcam*.



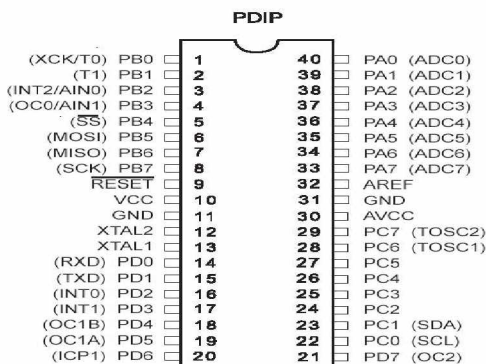
Gambar 2.1 contoh *webcam*.

2.2 Mikrokontroler ATmega8535

ATmega8535 merupakan mikrokontroler keluarga AVR (*Alf and Vegard's Risc Processor*). Pada mikrokontroler ini, hampir semua instruksi dieksekusi dalam satu siklus *clock*, hal ini berbeda dengan instruksi MCS51 yang membutuhkan 12 siklus *clock*.

Fitur yang dimiliki oleh ATmega8535 yaitu:

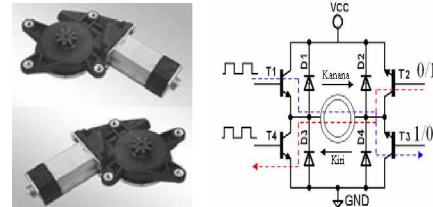
1. Saluran I/O sebanyak 32 buah, yaitu *Port A*, *Port B*, *Port C*, dan *Port D*.
2. ADC 10 bit sebanyak 8 saluran.
3. Tiga buah *Timer/Counter* dengan kemampuan perbandingan.
4. CPU yang terdiri atas 32 buah register.
5. *Watchdog Timer* dengan osilator internal.
6. SRAM sebesar 512 byte.
7. Memori *flash* sebesar 8kb dengan kemampuan *Read While Write*.
8. Unit interupsi internal dan eksternal.
9. *Port* antarmuka SPI.
10. EEPROM sebesar 512 byte yang dapat diprogram saat operasi.
11. Antarmuka komparator analog.
12. *Port* USART untuk komunikasi serial.
13. Empat kanal PWM.
14. Tegangan operasi sekitar 4.5-5.5V.



Gambar 2.2 Susunan kaki mikrokontroler ATmega8535.

2.3 Motor DC dan Driver Motor H-Bridge

Salah satu jenis motor yang sering digunakan dalam bidang kontrol yaitu Motor DC. Motor DC akan berputar jika dialiri tegangan dan arus DC.



Gambar 2.3 Motor DC dan jembatan H.

Sistem pengaturan motor DC yang sering digunakan pada sistem kontrol yaitu dengan *H-Bridge* yang pada dasarnya adalah 4 buah transistor yang difungsikan sebagai saklar. Pengaturan motor DC yaitu meliputi kecepatan dan arah. Pengaturan arah yaitu dengan cara membalik tegangan logika masukan *H-bridge*. Sedangkan sistem pengendalian kecepatan motor DC digunakan prinsip PWM (*Pulse Width Modulator*) yaitu suatu metode pengaturan kecepatan putaran motor DC dengan mengatur lamanya waktu pensaklaran aktif (*Duty Cycle*). Semakin besar *duty cycle* maka tegangan ekivalennya semakin besar, begitu pula sebaliknya.

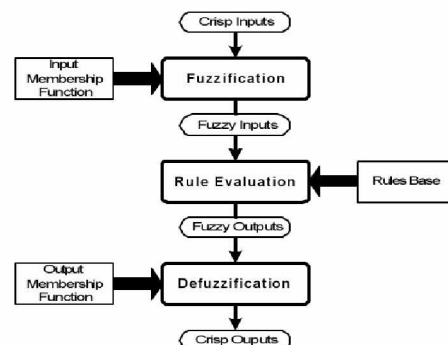
2.4 Pengendali Logika Fuzzy

Fuzzy berarti samar, kabur atau tidak jelas. *Fuzzy* adalah istilah yang dipakai oleh Lotfi A Zadeh pada bulan Juli 1964 untuk menyatakan kelompok / himpunan yang dapat dibedakan dengan himpunan lain berdasarkan derajat keanggotaan dengan batasan yang tidak begitu jelas (samar).

Pada teori himpunan *fuzzy* memungkinkan derajat keanggotaan (*member of degree*) suatu objek dalam suatu himpunan dinyatakan dalam interval antara "0" dan "1" atau [0 1]. Fungsi keanggotaan (*membership function*) dari himpunan *fuzzy* dapat disajikan dalam bentuk gabungan derajat keanggotaan tiap-tiap elemen pada semesta pembicaraan.

$$F = \sum \mu_F(u_i) / u_i$$

Pada Gambar 2.4 diperlihatkan diagram blok pengendali logika *fuzzy*.



Gambar 2.4 Diagram blok pengendali logika *fuzzy*.

Fuzzifikasi merupakan suatu proses untuk mengubah suatu peubah masukan dari bentuk tegas (*crisp*) menjadi peubah *fuzzy* (variable linguistik) yang biasanya disajikan dalam bentuk himpunan-himpunan *fuzzy* dengan fungsi keanggotaannya masing-masing.

Evaluasi aturan merupakan proses pengambilan keputusan (*inference*) yang berdasarkan aturan-aturan yang ditetapkan pada basis aturan (*rules base*) untuk menghubungkan antar peubah-peubah *fuzzy* masukan dan peubah *fuzzy* keluaran. Aturan-aturan ini berbentuk jika ... maka... (*IF ... THEN...*).

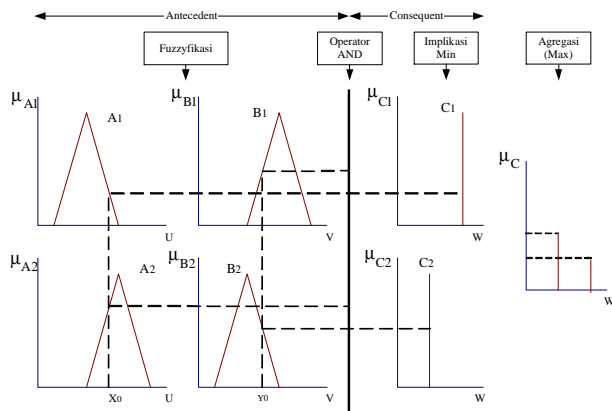
Proses pengambilan keputusan menggunakan metode Sugeno memberikan efisiensi komputasi karena dalam metode ini fungsi keanggotaan keluaran menggunakan fungsi *singleton* sehingga mempercepat penghitungan. Metode ini mempunyai bentuk aturan :

IF x is A and y is B then z = px+qy+k

dimana :

- x : variabel masukan 1
- A : himpunan *fuzzy* untuk variabel masukan X
- y : variabel masukan 2
- B : himpunan *fuzzy* untuk variabel masukan Y
- z : variabel keluaran
- p,q,k : konstanta tegas pada semesta pembicaraan W (p dan q bisa bernilai 0).

Implikasi dalam metode Sugeno yaitu merupakan suatu perkalian antara keluaran bagian *antecedent* dengan fungsi keanggotaan keluaran *singleton*. Sedangkan agregasi dalam metode Sugeno merupakan pengumpulan semua fungsi keanggotaan *singleton* hasil dari implikasi.



Gambar 2.5 Proses inferensi basis aturan.

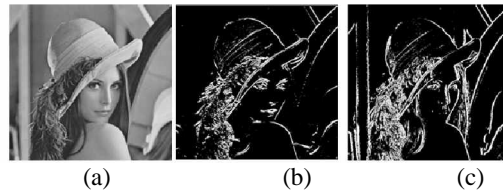
Defuzzifikasi merupakan proses perubahan besaran *fuzzy* yang disajikan dalam bentuk himpunan-himpunan *fuzzy* keluaran dengan fungsi keanggotaannya untuk mendapatkan kembali bentuk tegasnya. Hal ini diperlukan karena plant hanya mengenal nilai tegas sebagai besaran sebenarnya untuk regulasi prosesnya.

2.6 Deteksi Tepi Citra

Citra merupakan penggambaran dua dimensi dari bentuk fisik tiga dimensi. Citra digital merupakan suatu matriks yang elemen-elemennya menyatakan tingkat keabuan dari elemen gambar. Untuk mendapatkan suatu citra digital diperlukan suatu proses konversi, sehingga citra tersebut selanjutnya dapat diproses dengan komputer.

Pengolahan Citra merupakan proses pengolahan dan analisis citra yang banyak melibatkan persepsi visual. Istilah pengolahan citra digital secara umum didefinisikan sebagai pemrosesan citra dua dimensi dengan komputer.

Deteksi tepi merupakan salah satu proses pengolahan yang sering dilakukan pada analisis citra. Proses tersebut bertujuan untuk meningkatkan penampakan garis pada citra. Proses penentuan garis batas suatu wilayah yang homogen dengan cara membandingkan nilai *pixel* suatu titik terhadap *pixel* tetangga.

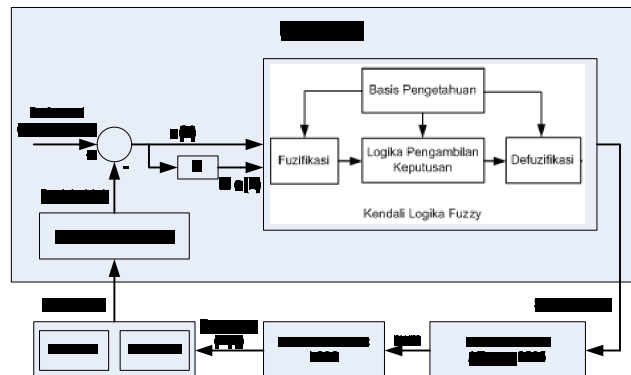


Gambar 2.6 Contoh deteksi tepi suatu citra digital.

- (a) Citra asli
- (b) Deteksi tepi horizontal
- (c) Deteksi tepi Vertikal

III PERANCANGAN SISTEM

Pada sistem pengendalian kamera ini dirancang untuk mengawasi suatu obyek yang telah ditentukan berdasarkan intensitas warna (RGB) yang dimiliki. Sistem pengendalian ini digerakkan oleh kontroler dengan metode kontrol *fuzzy*. Dimana sinyal kontrol akan mengatur posisi orientasi (arah/pandangan) *webcam* yaitu dengan memberikan suatu nilai dari kecepatan angular ($\omega(t)$) pada motor DC. Secara umum, diagram blok perancangan sistem pengendalian orientasi *webcam* menggunakan kendali logika *fuzzy* ditunjukkan seperti pada Gambar 3.1



Gambar 3.1 Blok diagram pengendalian orientasi *webcam* menggunakan kendali logika *fuzzy*

Referensi (*set point* posisi) adalah posisi obyek yang diinginkan yaitu ditengah-tengah layar monitoring (horizontal). Resolusi gambar yang digunakan adalah 640x480, dengan demikian referensinya yaitu 320. *Error*($e(k)$) didapat dengan membandingkan referensi dan jarak aktual obyek. Sedangkan *delta error* didapatkan dari nilai kesalahan sekarang dikurangi dengan nilai kesalahan sebelumnya. Kedua nilai ini (*error* dan *delta error*) digunakan sebagai masukan kendali logika *fuzzy*.

Perancangan pada sistem “ Pengendalian Orientasi *Webcam* sebagai Pengawas Ruang dengan Metode Kontrol *Fuzzy* ” akan dibagi menjadi 2 yaitu perancangan perangkat lunak (*Software*) dan perangkat keras (*Hardware*).

3.1 Perancangan Hardware

Secara umum perancangan sistem ini dapat dijelaskan sebagai berikut :

1. PC digunakan sebagai pusat pemrosesan data video dan pemrosesan sinyal kontrol yang akan dikirim ke mikrokontroler.
2. *webcam* digunakan sebagai media perekam video yang berfungsi sebagai alat pengawas sekaligus sebagai masukan untuk kontroler yaitu koordinat atau posisi obyek yang ditangkap.
3. Mikrokontroler ATmega8535 digunakan untuk mengolah sinyal kontrol dari PC yang digunakan sebagai masukan sinyal PWM.
4. *Driver* motor DC L293D untuk menguatkan arus keluaran dari mikrokontroler.
5. Motor DC sebagai penggerak atau pemutar *webcam*. Kecepatan putarnya dikendalikan melalui perubahan *duty cycle* PWM.

3.1.1 Komputer / PC

Merupakan perangkat utama pada sistem pengendalian orientasi *webcam* karena disinilah pusat pengolahan data. Komputer menerima data gambar dari *webcam* kemudian melakukan pemrosesan citra untuk mendapatkan nilai posisi dari obyek yang diamati. Komputer terhubung dengan *webcam* melalui *port* USB. Hasil pemrosesan citra akan diperoleh sinyal *error*. Sinyal *error* bersama-sama dengan *delta error* (selisih *error* sekarang dan *error* sebelumnya) akan diproses kembali yaitu dalam perhitungan kendali *fuzzy*. Keluaran dari kendali *fuzzy* berupa suatu nilai masukan untuk PWM. Data ini dikirim ke mikrokontroler melalui komunikasi serial RS232.

3.1.2 Sistem Mikrokontroler ATmega8535

Mikrokontroler ATmega8535 berfungsi sebagai penerima sinyal kontrol dari komputer melalui komunikasi serial RS232. Sinyal kontrol ini kemudian diolah dan digunakan sebagai masukan PWM. Selain menerima data masukan untuk PWM, mikrokontroler juga menerima data arah perputaran motor DC.

Output dari mikrokontroler yaitu sinyal PWM pada *pin* B.3 (timer0) yang berfungsi memberikan pulsa kotak ke *driver* motor yang untuk selanjutnya digunakan untuk mengatur kecepatan putar motor DC. *Pin* B.0 dan B.1 digunakan sebagai masukan *driver* motor untuk menentukan arah putar motor DC yaitu ke kiri maupun ke kanan.

3.1.3 Webcam

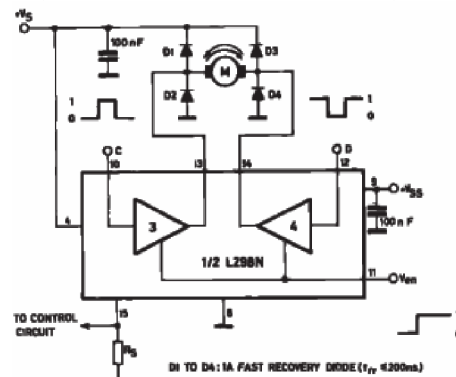
Webcam yang digunakan pada sistem tugas akhir ini yaitu produk dari *Sun Flower* dengan tipe *SF-1007*. Spesifikasi dari *webcam* ini yaitu :

1. Resolusi : 160x120, 176x144, 320x240, 352x288, 640x480.
2. Hasil *record* berformat data AVI.
3. Pengaturan fokus kamera terdapat pada sisi depan kamera yaitu secara manual.
4. Pengaturan pencahayaan putih (*white balance*) secara otomatis.
5. Dilengkapi *built-in image compression*.
6. Pengaturan keseimbangan warna (*color compensated*) secara otomatis.

3.1.4 Motor DC dan driver motor L298

Motor DC merupakan sebuah komponen yang memerlukan arus yang cukup besar untuk menggerakannya. Oleh karena itu motor DC biasanya memiliki penggerak tersendiri. Pada tugas akhir ini motor DC akan digerakkan dengan menggunakan PWM yang telah terintegrasi dengan rangkaian *H-Bridge*.

Dengan rangkaian *H-Bridge* yang memiliki input PWM ini, maka selain arah kita juga bisa mengendalikan kecepatan putar motor DC tersebut. Perancangan *driver* motor DC yaitu sebagai berikut :

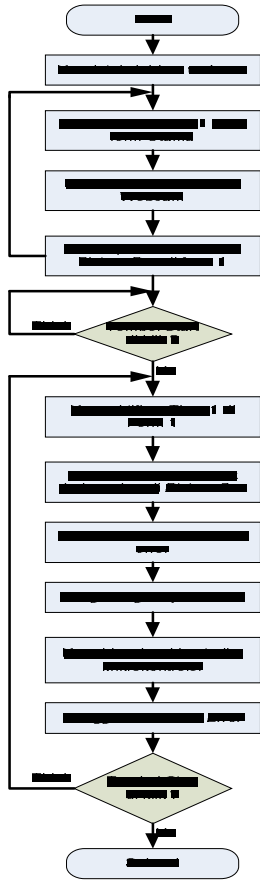


Gambar 3.2 Wiring Driver Motor DC.

3.2 Perancangan Software

3.2.1 Perancangan Software pada Komputer

Perancangan perangkat lunak (*software*) pada komputer menggunakan bahasa pemrograman Visual Basic 6.0. *Flow chart* program utama dapat dilihat pada Gambar 3.3.



Gambar 3.3 *Flowchart* program utama.

3.2.1.1 *Mainform*

Prosedur pada *mainform* akan menjalankan fungsi pendeteksian driver, kemudian dilanjutkan dengan pemanggilan *procedure create capture window* dan yang terakhir adalah membuka *form1*.

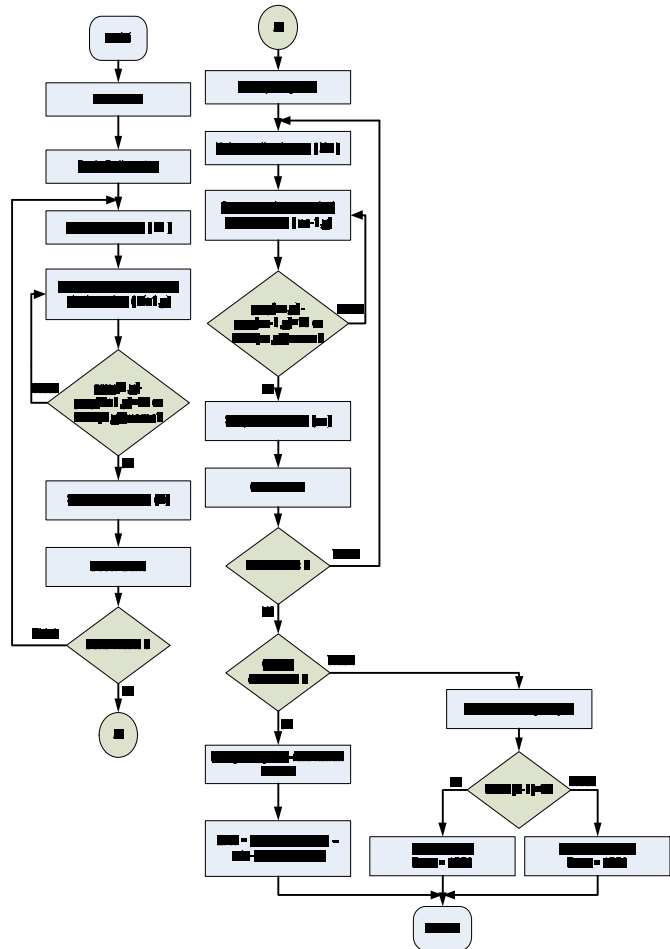
Pada *form* utama terdapat sebuah timer yang berfungsi untuk menjalankan suatu prosedur penyimpanan gambar secara terus menerus dan meng-*overwrite* gambar sebelumnya, serta menampilkannya secara terus menerus pada *picture box* di *form1* yang merupakan *form interfacing* pada sistem pengendalian orientasi *webcam* ini.

3.2.1.2 *Form1*

Program yang ditangani pada *Form1* yaitu melakukan pengolahan citra (perhitungan *error*), *procedure* pemilihan obyek, melakukan perhitungan sinyal kontrol berdasarkan algoritma kendali *fuzzy*, serta mengirim sinyal kontrol ke mikrokontroler melalui komunikasi serial RS232.

a. Perhitungan Error

Flowchart pembacaan gambar sehingga dihasilkan sinyal *error* adalah sebagai berikut :



Gambar 3.4 *Flowchart* perhitungan *error*.

Mode pemilihan obyek pada tugas akhir ini yaitu: HITAM, MERAH, KUNING, HIJAU, BIRU, PUTIH, dan KONTRAS (terhadap *background*).

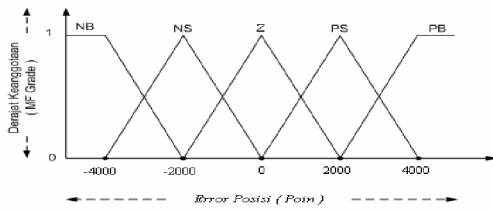
Sistem deteksi dilakukan pada obyek warna dengan cara filterisasi nilai RGB pada citra. Cara menetapkan nilai batas pada filter dengan cara coba-coba (*trial-error*). Sedangkan pada mode kontras yaitu dengan deteksi tepi.

b. *Prosedure* fungsi *fuzzy*

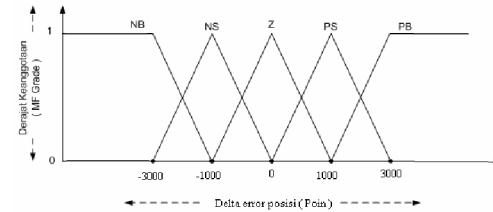
Pengendali logika *fuzzy* akan mengevaluasi tiap masukan *fuzzy* yaitu *error* dan *delta error* dari hasil pengolahan citra gambar kemudian melakukan perhitungan sinyal kontrol melalui tahapan *Fuzzifikasi*, evaluasi *rule* dan *defuzzifikasi*. Pengendali *fuzzy* yang digunakan dalam sistem ini adalah sugeno *fuzzy* model.

Ø *Fuzzifikasi*

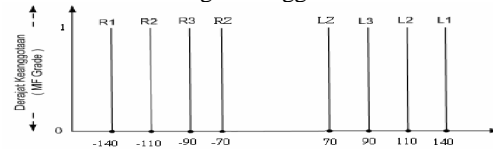
Dalam perancangan kendali logika *fuzzy* ini terdapat 2 fungsi keanggotaan masukan (*Error* dan *delta error*) dan 1 fungsi keanggotaan keluaran.



Gambar 3.5 Fungsi keanggotaan error.



Gambar 3.6 Fungsi keanggotaan delta error.



Gambar 3.7 Fungsi keanggotaan keluaran fuzzy.

Ø Evaluasi aturan

Metode pengambilan keputusan (*inferensi*) yang digunakan dalam pemrograman ini adalah metode *product* dimana hasil *Fuzzifikasi* masukan *error* dan *delta error* dikalikan.

Tabel 3.1 Basis aturan Fuzzy

Error / DError	NB	NS	NZ	PS	PB
NB	R1	R1	R3	L3	L3
NS	R1	R2	LZ/RZ	L3	L3
Z	R2	R2	LZ/RZ	L2	L2
PS	R3	R3	LZ/RZ	L2	L1
PB	R3	R3	L3	L1	L1

Ø Defuzzifikasi

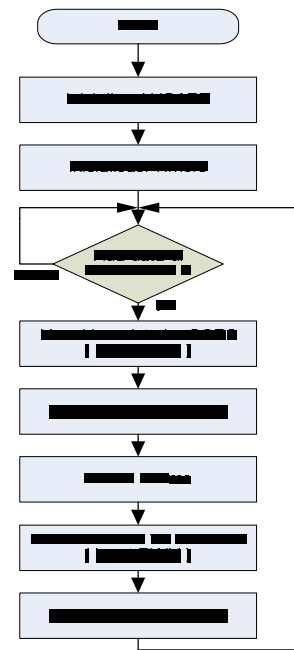
Hasil keluaran *fuzzy* (*crisp output*) akan menentukan besar kecepatan angular motor sebagai penggerak *webcam*. Untuk mendapatkan nilai tegas (*crisp*) pada model *fuzzy* sugeno dengan memasukkan nilai-nilai hasil evaluasi aturan kedalam rumus :

$$Crisp_Out = \frac{\sum_i (Fuzzy\ Output_i) \times (Posisi\ singleton\ di\ X\ axis_i)}{\sum_i (Fuzzy\ Output_i)}$$

3.2.2 Perancangan Software pada Mikrokontroler ATmega8535

Fungsi mikrokontroler adalah menerima data hasil output *fuzzy* kemudian pengolahan data tersebut sebagai masukan PWM.

Flowchart pada Mikrokontroler untuk menangani fungsinya adalah sebagai berikut :



Gambar 3.8 Flowchart program pada mikrokontroler.

Komunikasi serial yang digunakan yaitu mode *Asynchronous*, *Baud rate* 9600 dengan parameter komunikasi 8 Data, 1 *Stop bit* dan tidak ada *Parity*.

PWM digunakan adalah PWM0 dengan *Clock value* 500.000 kHz dan *Mode Phase correct PWM top=FFh*. PINB.3 (OC0) *output* yang terhubung adalah *Non-Inverted PWM*.

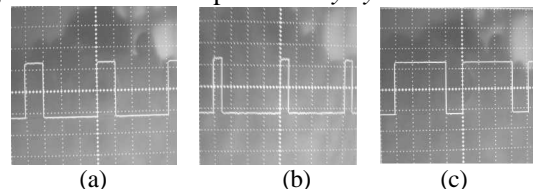
IV PENGUJIAN DAN ANALISA

4.1 Pengujian Perangkat Keras (Hardware)

4.1.2 Pengujian PWM Mikrokontroler

Pengujian PWM dilakukan dengan menggunakan program yang dibuat secara khusus agar mikrokontroler dapat berkomunikasi secara serial dengan PC. Dengan demikian data nilai PWM pada register OCR0 dapat diganti-ganti melalui PC.

Pengujian PWM dilakukan dengan memasukkan nilai PWM tertentu dan menghitung *duty cycle*-nya. Berikut adalah beberapa *screen shoot* pada osiloskop guna memperlihatkan pengaruh perubahan data pada register OCR0 terhadap lebar *duty cycle*.



Gambar 4.1 Sinyal PWM pada mikrokontroler

- OCR0 = 64 (25% *duty cycle*)
- OCR0 = 32 (13% *duty cycle*)
- OCR0 = 192 (75% *duty cycle*)

Output PWM telah sesuai dengan masukan yang diberikan dengan demikian PWM telah berfungsi dengan baik.

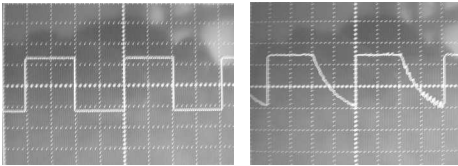
4.1.3 Pengujian Driver Motor DC

Rangkaian *driver* L298 berfungsi sebagai penggerak motor dc. V_{motor} yang digunakan adalah 13,6 Volt. Hasil Pengujian *driver* motor DC L298 dapat dilihat pada Tabel 4.2 berikut:

Tabel 4.1 Hasil Pengujian *Driver* L298

Nilai OCR0	$V_{keluaran}$ (volt) Channel B	$V_{keluaran}$ (volt) perhitungan
255	13,6	13,6
224	13,0	11,95
192	12,2	10,24
160	11,5	8,53
128	10,7	6,83
96	9,6	5,22
64	8,3	3,41
32	6,7	1,71
0	0,3	0

Dari Tabel 4.1 dapat dilihat bahwa tegangan efektif keluaran *driver* berbeda dengan nilai perhitungan, hal ini dikarenakan saat pulsa pwm aktif, nilai *driver* juga langsung aktif, sedangkan saat sinyal output pwm turun, tegangan output *driver* tidak langsung turun melainkan secara perlahan-lahan dikarenakan ada penyimpanan muatan oleh kapasitor. Dengan demikian, tegangan efektif (V_{rms}) yang terukur lebih besar dari pada teori. Hal ini dapat ditunjukkan seperti gambar dibawah ini :

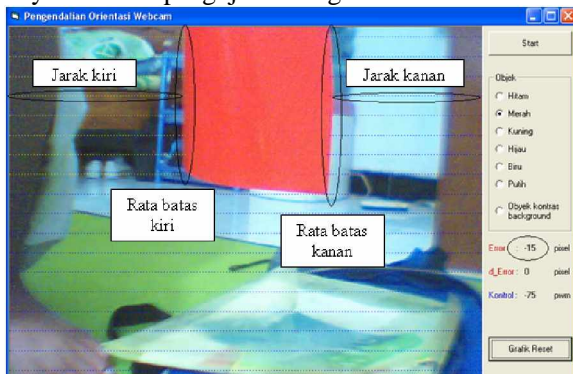


Gambar 4.2 Output PWM (kiri) dan *driver* motor (kanan) dengan OCR0=128 (Duty cycle 50 %).

4.2 Pengujian Software

4.2.1 Perhitungan Posisi Obyek

Pengujian perhitungan posisi obyek yaitu dengan cara menggunakan algoritma dan listing yang sama, tetapi pada proses *scanning* gambar akan ditandai, dengan demikian akan lebih mudah apakah ada kesalahan pada pembacaan posisi obyek. Proses pengujian sebagai berikut :



Gambar 4.3 Proses perhitungan posisi obyek dan *error* terhadap garis tengah.

Proses perhitungan obyek dimulai dari pembacaan tiap *pixel* dari koordinat kiri atas kemudian bergeser kekanan (dalam 1 baris deteksi kolom) sampai didapatkan obyek yang diinginkan, setelah mendapatkan obyek atau *scanning* baris pertama selesai kemudian dilanjutkan ke baris selanjutnya, dan proses ini akan terus berlanjut sampai proses *scanning* seluruh gambar selesai. Pada sistem ini, deteksi gambar tidak dilakukan pada seluruh *pixel* gambar, tetapi dengan sistem sampling. Hal ini dikarenakan untuk mempercepat proses pembacaan gambar dan sistem lebih responsif. *Scanning* kolom tiap 4 *pixel* (60 poin pada VB) dan 27 *pixel* (400 poin VB) untuk *scanning* tiap baris.

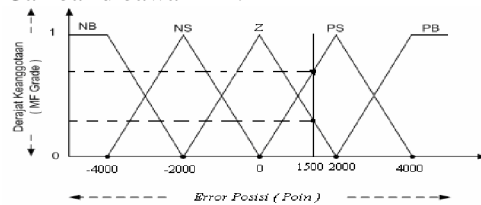
Posisi suatu obyek yaitu rata-rata jarak sebelah kanan dan sebelah kiri. Sedangkan *error* yaitu referensi dikurangi posisi obyek. Dan dapat dirumuskan sebagai berikut:

$$Error = ref (4776) - \text{posisi obyek}$$

4.2.2 Pengujian Pemrograman Kendali Logika Fuzzy

Untuk menguji pemrograman kendali logika *fuzzy* akan dilakukan secara bertahap. Tahap pertama akan dilakukan pengujian pemrograman *fuzzifikasi*, tahap kedua dilakukan pengujian pemrograman evaluasi aturan dan tahap terakhir adalah pengujian pemrograman *defuzzifikasi*.

Dalam pengujian *fuzzy* menggunakan masukan text pada program dan membandingkannya dengan perhitungan manual. Sebagai contoh akan dicari derajat keanggotaan dari suatu masukan yaitu *error* = 1500 poin dengan fungsi keanggotaan diperlihatkan pada Gambar dibawah ini :



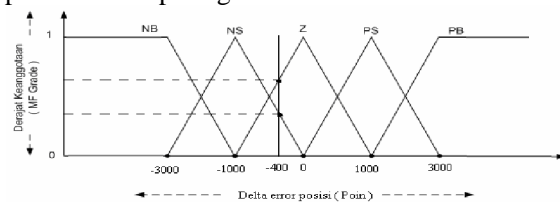
Gambar 4.4 Masukan *Error* = 1500.

Pada Gambar diatas, masukan *Error* = 1500 akan memotong MF Z dan MF PS. Maka derajat keanggotaan dapat ditentukan sebagai berikut :

$$Error_Z (w1(3)) = 1 - 1500/2000 = 0,25$$

$$Error_PS (w1(4)) = 1500/2000 = 0,75$$

Sedangkan untuk masukan *delta error* yaitu dengan masukan -400 poin dengan fungsi keanggotaan diperlihatkan seperti gambar dibawah ini :



Gambar 4.5 Masukan *delta error* = -400.

Pada Gambar diatas, masukan $\Delta error = -400$ akan memotong MF NZ dan MF Z. Maka derajat keanggotaan dapat ditentukan sebagai berikut :

$$\begin{aligned} \text{Error_NZ} (v1(2)) &= -(-400)/1500 = 0,2667 \\ \text{Error_Z} (v1(3)) &= 1 + (-400)/1500 = 0,733 \end{aligned}$$

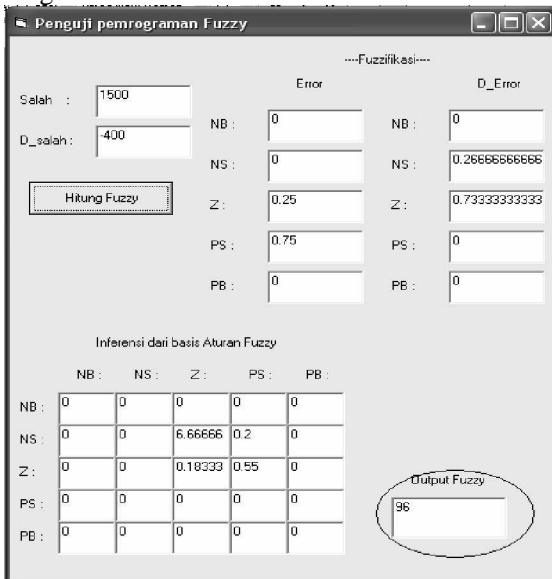
Langkah selanjutnya yaitu melakukan Pengujian pemrograman evaluasi aturan. Karena pada *Fuzzifikasi error* hanya ada 2 fungsi keanggotaan yang tidak bernilai 0 dan pada $\Delta error$ juga hanya ada 2 fungsi keanggotaan yang tidak bernilai 0 maka hanya ada 4 rule saja yang tidak bernilai 0 yaitu :

- E_Z dan DE_NZ = $0,25 \times 0,2667 = 0,06667$
- E_Z dan DE_Z = $0,25 \times 0,7333 = 0,183$
- E_PS dan DE_NZ = $0,75 \times 0,2667 = 0,2$
- E_PS dan DE_Z = $0,75 \times 0,7333 = 0,55$

Setelah tahap evaluasi aturan, tahap berikutnya yaitu Pengujian program *defuzzifikasi* Pada model *fuzzy sugeno*, *output fuzzy* didapatkan dari rumus sebagai berikut :

$$\begin{aligned} \text{Crisp_Out} &= \frac{\sum_i (\text{Fuzzy Output}_i) \times (\text{Posisi singleton di X axis}_i)}{\sum_i (\text{Fuzzy Output}_i)} \\ \text{Crisp_Out} &= \frac{(0,06667 \times 70) + (0,2 \times 90) + (0,1833 \times 70) + (0,55 \times 110)}{0,06667 + 0,2 + 0,1833 + 0,55} \\ \text{Crisp_Out} &= \frac{95,998}{0,99997} = 96 \end{aligned}$$

Hasil diatas selanjutnya dibandingkan dengan hasil keluaran pada program penguji yaitu sebagai berikut :



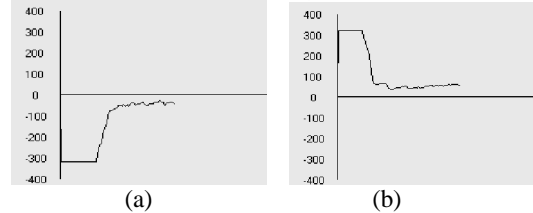
Gambar 4.6 Tampilan Hasil Output *Fuzzy*.

Dari hasil eksekusi program didapatkan nilai yang sama dengan perhitungan.

4.3 Pengujian Pengendalian Orientasi Kamera Pengawas Ruangan

4.3.1 Pengawasan Obyek Diam Tunggal

Pada pengujian ini dikondisikan obyek yang dicari tidak berada dalam area monitoringnya, sehingga kamera akan masuk ke mode *scanning* obyek. Pada ruangan hanya ada 1 obyek saja (sesuai dengan mode obyek yang diinginkan). Setelah mendapatkan obyek yang diinginkan maka kamera akan menempatkan obyek tersebut ditengah pandangan (pengawasan) kamera. Grafik respon sistem dapat ditunjukkan oleh gambar dibawah ini :



Gambar 4.7 Respon pengawasan obyek diam tunggal:
a. scanning kanan.
b. scanning kiri.

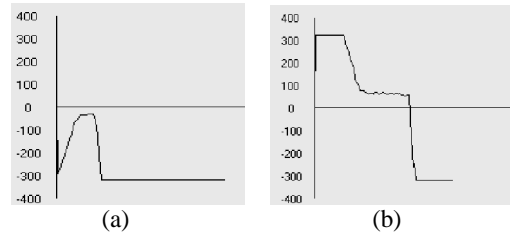
Mode *scanning* saat pertama kali yaitu scanning kekanan, tetapi setelah menemukan obyek kemudian kehilangan obyek maka scanning dapat kekiri maupun kekanan tergantung dari posisi obyek sebelumnya. *Error steady state* pada grafik 4.7(a) yaitu -43pixel, dan 4.7(b) yaitu +40pixel. *Error* disini dapat ditoleransi karena jarak -43 dan 40 pixel adalah jarak yang cukup kecil.

Pengujian untuk berbagai mode obyek mempunyai respon yang hampir sama seperti pada grafik diatas.

4.3.2 Pengawasan Obyek Bergerak Tunggal

4.3.2.1 Obyek Diam kemudian Bergerak

Kondisi awal obyek yaitu tidak berada dalam area pengawasan kamera, sehingga kamera harus melakukan *scanning* untuk mencari obyek yang diinginkan. Respon sistem dapat ditunjukkan oleh gambar berikut :

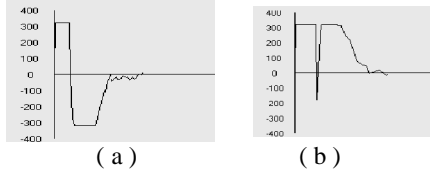


Gambar 4.8 Respon pengawasan obyek tunggal diam kemudian bergerak kekanan
(a) *scanning* awal kamera kekanan
(b) *scanning* awal kamera kekiri.

Dari gambar diatas terlihat bahwa sistem kehilangan obyek karena obyek bergerak lebih cepat daripada kecepatan kamera.

4.3.2.2 Obyek Bergerak kemudian Diam

Scanning awal sistem pada pengujian ini yaitu kearah kanan. Respon sistem seperti ditunjukkan oleh gambar dibawah ini :



Gambar 4.9 Respon pengawasan obyek tunggal bergerak kemudian diam

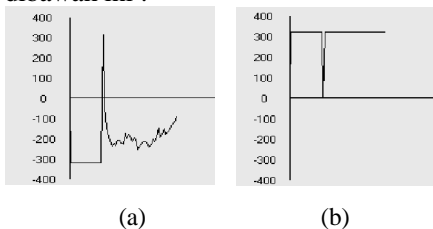
- (a) arah obyek belawan arah *scanning*
- (b) arah obyek se-arah dengan arah *scanning*.

Dari gambar diatas terlihat bahwa sistem dapat menempatkan posisi pengawasan ditengan area *monitoring* baik gerakan benda yang searah *scanning* kamera maupun yang berlawanan arah *scanning*.

4.3.2.3 Obyek Bergerak Terus-menerus

Ø Gerak Obyek Searah *Scanning* Kamera

Respon sistem seperti ditunjukkan oleh gambar dibawah ini :

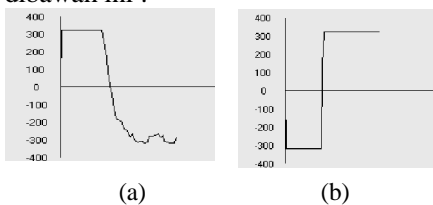


Gambar 4.10 Respon sistem untuk obyek yang bergerak searah *scanning*

- (a) kecepatan lambat ($\omega < 0,8$ RPM).
- (b) Kecepatan cepat ($\omega > 0,8$ RPM).

Ø Gerak Obyek Berlawanan arah *Scanning* Kamera

Respon sistem seperti ditunjukkan oleh gambar dibawah ini :



Gambar 4.22 Respon sistem untuk obyek yang bergerak berlawanan arah *scanning*

- (c) kecepatan lambat ($\omega < 0,8$ RPM).
- (d) Kecepatan cepat ($\omega > 0,8$ RPM).

4.3.3 Pengawasan Obyek Majemuk Diam

Pengujian disini dilakukan untuk melihat respon sistem pengendalian kamera jika dalam ruangan tersebut terdapat lebih dari satu obyek dengan warna yang sama. Prinsip pendeteksian obyek yang berjumlah lebih dari 1 yaitu seperti dijelaskan dengan gambar berikut :



Gambar 4.26 Pendeteksian obyek majemuk.

Dari gambar di atas dapat dilihat bahwa terdapat 2 buah obyek yang mempunyai bentuk dan ukuran yang berbeda. Kedua obyek yang terdeteksi akan ditempatkan ditengah area *monitoring*. jika ukuran obyek lebih besar dari pada area monitoring maka bagian obyek yang terdeteksi akan ditempatkan di tengah area pengawasan. Sistem seolah-olah hanya mendeteksi bahwa obyek tersebut hanya satu dengan ukuran gabungan dari semua obyek.

4.3.4 Pengawasan Obyek Majemuk Bergerak

Ø Obyek Bergerak dan Obyek Diam

Hasil pengujian yaitu :

- a. sistem akan mengikuti obyek yang lebih dominan (ukuran lebih besar) baik obyek itu yang bergerak maupun obyek tersebut diam.
- b. Jika gerakan salah satu obyek sangat cepat maka sistem akan mengawasi obyek yang diam, sedangkan obyek yang bergerak cepat akan diabaikan walaupun memiliki ukuran lebih besar.

Respon dari sistem pada pengujian ini hampir sama dengan obyek tunggal. Pada pengujian ini digunakan obyek berwarna merah, sedangkan untuk jenis warna yang lain akan mempunyai respon yang hampir sama.

Ø Obyek Bergerak Semua

Hasil yang didapat pada pengujian ini yaitu:

- a. sistem akan mengikuti obyek yang lebih dominan (ukuran lebih besar).
- b. Jika terdapat obyek yang bergerak sangat cepat maka sistem akan mengabaikan obyek tersebut walaupun ukurannya lebih besar. Sistem akan mengikuti obyek yang bergerak dibawah kecepatan kamera.
- c. Jika semua obyek yang terdeteksi mempunyai kecepatan tinggi maka sistem akan melakukan *scanning* searah dengan obyek yang terakhir terdeteksi.

Respon dari sistem pada pengujian ini hampir sama dengan obyek tunggal. Pada pengujian ini digunakan obyek berwarna merah, sedangkan untuk jenis warna yang lain akan mempunyai respon yang hampir sama.

V PENUTUP

5.1 KESIMPULAN

Berdasarkan perancangan, pengujian dan analisa yang telah dilakukan, maka dapat disimpulkan beberapa hal sebagai berikut:

1. Pengendalian orientasi kamera mempunyai respon yang cukup baik terutama pada obyek diam yaitu dengan *error steady state* rata-rata ± 30 pixel.
2. *Error steady state* terjadi karena karakteristik motor yang mempunyai *dead zone* yang cukup lebar. Motor bisa berputar jika diberikan minimal $\pm 33\%$ *dutycycle* PWM. *Error steady state* ± 30 pixel adalah angka yang sangat kecil karena bila ubah ke sudut hanya $\pm 2^\circ$.
3. Jika dalam area *monitoring* tidak terdapat obyek yang harus diawasi maka sistem akan melakukan *scanning* sampai ditemukan obyek.
4. Pada pengawasan obyek bergerak, kamera akan melakukan pergerakan sama dengan arah pergerakan obyek baik untuk obyek berkecepatan lambat maupun cepat.
5. Kecepatan maksimum *webcam* adalah $\pm 0,8$ RPM.
6. Prinsip dan respon pada obyek mejemuk hampir sama dengan obyek tunggal.

5.2 SARAN

Untuk pengembangan sistem lebih lanjut, maka penulis memberikan saran-saran sebagai berikut:

1. Dapat dilakukan perancangan kembali pengendali logika *fuzzy*, dengan basis aturan maupun fungsi keanggotaan (masukan/keluaran) yang berbeda untuk mendapatkan hasil yang paling memuaskan. Serta dapat dicoba model *fuzzy* yang lain.
2. Pengembangan karakteristik obyek yaitu dari bentuknya melalui pengenalan pola misalnya hewan, manusia (pria/wanita) serta benda mati.

DAFTAR PUSTAKA

- [1] Wahyudi, *Implementasi Fuzzy Logic Controller Pada Sistem Pengereman Kereta Api*, Universitas Diponegoro, Semarang, 2005
- [2] Heryanto, M.Ary & Wisnu Adi P., *Pemrograman Bahasa C untuk Mikrokontroler ATMEGA8535*, Penerbit Andi, Yogyakarta, 2008.
- [3] Munir, Rinaldi, *Pengolahan Citra Digital dengan Pendekatan Algoritmik*, Informatika, Bandung, 2004.
- [4] -----, *vbVidCap for Windows*, <http://www.shrinkwrapvb.com>. 2000.

- [5] Basuki, Achmad; Josua F. Palandino; and Fatchurrochman, *Pengolahan Citra Digital Menggunakan Visual Basic*, Graha Ilmu, Yogyakarta, 2005.
- [6] Prasetia, Retna and Catur Edi Widodo, *Interfacing Port Paralel dan Port Serial Komputer dengan Visual Basic 6.0*, Andi, Yogyakarta, 2004.
- [7] Fallahi, Andi., *Perancangan Robot Mobile Penjelaj Dinding Ruang Menggunakan Kendali Logika Fuzzy*, Skripsi S-1, Universitas Diponegoro, Semarang, 2005.
- [8] Santoso, Junaidi., *Rancang Bangun Robot Mobil Penghindar Rintangan Menggunakan Kendali Logika Fuzzy*, Skripsi S-1, Universitas Diponegoro, Semarang, 2007.
- [9] -----, *ATmega8535 Data Sheet*, <http://www.atmel.com>.
- [10] -----, *L298 Data Sheet*, <http://www.st.com>, januari, 2000.

Masri'an (L2F 004 491)



Lahir di Grobogan, Saat ini sedang melanjutkan studi pendidikan strata I di Jurusan Teknik Elektro, Fakultas Teknik Universitas Diponegoro Konsentrasi Kontrol.

Mengetahui dan mengesahkan,

Dosen Pembimbing I

Dosen Pembimbing II

Iwan Setiawan, ST, MT

NIP. 132 283 183

Tanggal: _____

Darjat, ST, MT

NIP. 132 231 135

Tanggal: _____