

**PERANCANGAN *MOBILE ROBOT*  
DENGAN SENSOR KAMERA  
MENGUNAKAN SISTEM KENDALI FUZZY**

**SKRIPSI**

**OLEH**

**LASTI WARASIH H.**

**04 04 03 057 1**



**DEPARTEMEN TEKNIK ELEKTRO  
FAKULTAS TEKNIK UNIVERSITAS INDONESIA  
GENAP 2007/2008**

**PERANCANGAN *MOBILE ROBOT*  
DENGAN SENSOR KAMERA  
MENGUNAKAN SISTEM KENDALI FUZZY**

**SKRIPSI**

**OLEH**

**LASTI WARASIH H.**

**04 04 03 057 1**



**SKRIPSI INI DIAJUKAN UNTUK MELENGKAPI SEBAGIAN  
PERSYARATAN MENJADI SARJANA TEKNIK**

**DEPARTEMEN TEKNIK ELEKTRO  
FAKULTAS TEKNIK UNIVERSITAS INDONESIA**

**GENAP 2007/ 2008**

## PERNYATAAN KEASLIAN SKRIPSI

Saya menyatakan dengan sesungguhnya bahwa skripsi dengan judul :

### **PERANCANGAN *MOBILE ROBOT* DENGAN SENSOR KAMERA MENGUNAKAN SISTEM KENDALI FUZZY**

yang dibuat untuk melengkapi sebagian persyaratan sebagai Sarjana Teknik pada Program Studi Teknik Elektro Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia, sejauh yang saya ketahui bukan merupakan tiruan atau duplikasi dari skripsi yang sudah dipublikasikan atau pernah dipakai untuk mendapatkan gelar kesarjanaan di lingkungan Universitas Indonesia maupun di Perguruan Tinggi atau Instansi manapun, kecuali bagian yang merupakan sumber informasinya dicantumkan sebagaimana mestinya .

Depok, 25 Juni 2008

Lasti Warasih H.

NPM 04 04 03 057 1

# PERSETUJUAN

Skripsi dengan judul:

## **PERANCANGAN *MOBILE ROBOT* DENGAN SENSOR KAMERA MENGUNAKAN SISTEM KENDALI FUZZY**

dibuat untuk melengkapi sebagian persyaratan menjadi Sarjana Teknik pada Program Studi Teknik Elektro Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia. dan disetujui untuk diajukan dalam sidang ujian skripsi.

Depok, 25 Juni 2008

Dosen Pembimbing

Dr. Abdul Muis, ST, M.Eng

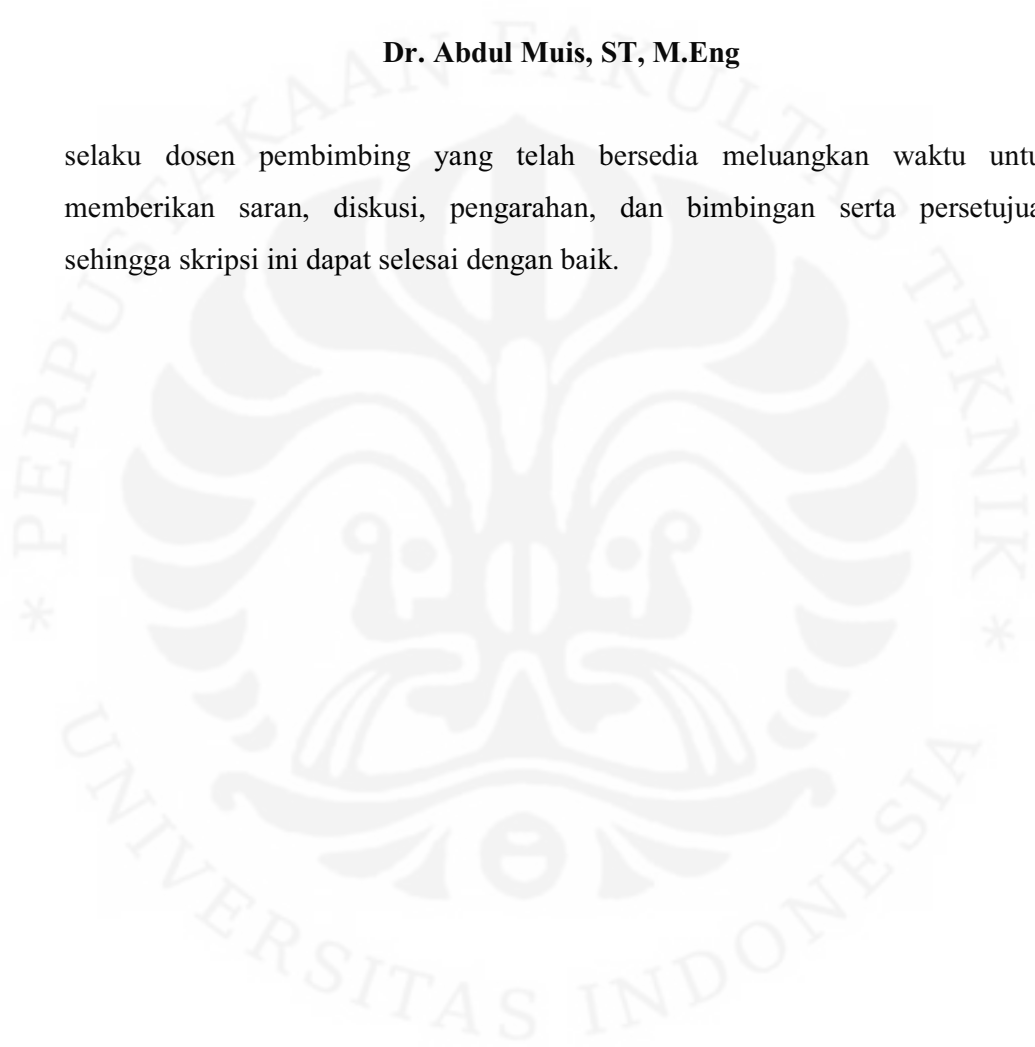
NIP. 132 233 210

## UCAPAN TERIMA KASIH

Puji syukur kehadiran Tuhan Yang Maha Esa atas berkat dan rahmat-Nya skripsi ini dapat diselesaikan dengan baik. Penulis juga mengucapkan terima kasih kepada:

**Dr. Abdul Muis, ST, M.Eng**

selaku dosen pembimbing yang telah bersedia meluangkan waktu untuk memberikan saran, diskusi, pengarahan, dan bimbingan serta persetujuan sehingga skripsi ini dapat selesai dengan baik.



# DAFTAR ISI

	Halaman
PERNYATAAN KEASLIAN SKRIPSI	ii
PERSETUJUAN	iii
UCAPAN TERIMA KASIH	iv
ABSTRAK	v
ABSTRACT	vi
DAFTAR ISI	vii
DAFTAR GAMBAR	x
DAFTAR TABEL	xii
DAFTAR LAMPIRAN	xiii
DAFTAR SINGKATAN	xiv
DAFTAR ISTILAH	xv
BAB I PENDAHULUAN	1
2.1 LATAR BELAKANG	1
2.2 PERUMUSAN MASALAH	1
2.3 TUJUAN PENULISAN	2
2.4 BATASAN MASALAH	2
2.5 METODOLOGI PENULISAN	2
2.6 SISTEMATIKA PENULISAN	2
BAB II DASAR TEORI	3
2.1 SISTEM KENDALI	3
2.1.1 Sistem Kendali Fuzzy	3
2.2 ROBOT	4
2.2.1 Mobile Robot	5
2.2.2 Manipulator Robot	5

2.3	SENSOR	5
2.3.1	Sensor Kamera	6
2.4	Pengolahan citra digital	6
2.4.1	Akuisisi Citra	7
2.4.2	Perbaikan Citra ( <i>Image enhancement</i> )	7
2.4.2.1	<i>Transformasi Grayscale</i>	7
2.4.2.2	<i>Contrast Stretching</i>	8
2.4.2.3	<i>Perbaikan Gamma</i>	9
2.4.3	Penyimpanan Citra ( <i>Storage</i> )	9
2.4.4	Pengenalan Citra Digital	9
2.4.4.1	<i>Pendekatan Struktural</i>	10
2.4.4.2	<i>Pendekatan Statistik</i>	11
2.5	Mikrokontroler AtMEGA8535 [2]	11
2.6	Standar komunikasi data RS-232	12
2.6.1	Karakteristik Sinyal RS-232 dan Tegangan	13
2.6.2	Tipe Konektor	14
2.7	Motor DC	15
BAB III PERANCANGAN SISTEM		18
3.1	PERANCANGAN KONSTRUKSI FISIK	19
3.1.1	Rangkaian Mikrokontroler	20
3.2.	PERANCANGAN SISTEM KENDALI MOTOR	23
3.2.1	Phase Width Modulation (PWM)	23
3.2.2	Rangkaian Kendali Motor	24
3.3	PERANCANGAN SISTEM SENSOR KAMERA	26
3.3.1	Penyesuaian Awal	27
3.3.1.1	<i>Perancangan Pola</i>	27
3.3.1.2	<i>Akuisisi Citra dan Kalibrasi</i>	28
3.3.2	Flowchart Program Sensor Kamera	29
3.3.3	Pemrosesan Citra Awal	29
3.3.4	Pengambilan Informasi Objek	30
3.3.5	Pengenalan Objek	32
3.4	PERANCANGAN SISTEM PENGENDALI MOBILE ROBOT	33
3.4.1	Sistem Pengendalian Logika Fuzzy	33
3.4.1.1	<i>Sistem Pengendalian Tanpa Objek Penghalang</i>	34
3.4.1.2	<i>Sistem Pengendalian Dengan Objek Penghalang</i>	38
BAB IV PENGUJIAN DAN ANALISA		40
4.1	PENGUJIAN KOMUNIKASI SERIAL	40
4.2	PENGUJIAN SISTEM KENDALI MOTOR	41
4.2	PENGUJIAN SISTEM SENSOR KAMERA	41

4.3	PENGUJIAN ALGORITMA PENGENDALI	42
4.3.1.	Pengujian Tanpa Objek Penghalang	42
4.3.2.	Pengujian Dengan Objek Penghalang	46
	BAB V KESIMPULAN	50
	DAFTAR ACUAN	51
	DAFTAR PUSTAKA	52
	LAMPIRAN	53

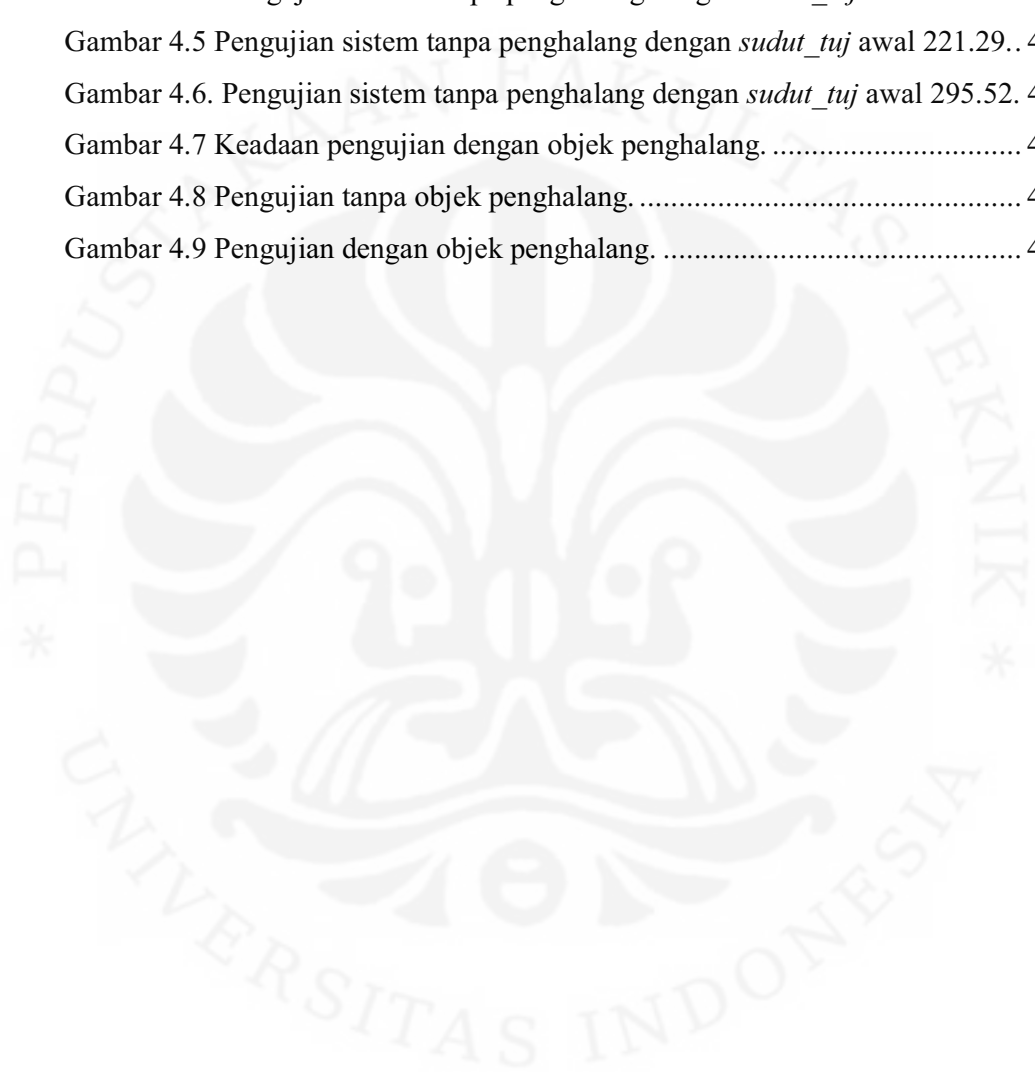




## DAFTAR GAMBAR

	Halaman
Gambar 2.1 Diagram blok sistem kendali tertutup.....	3
Gambar 2.2 Diagram Blok Sistem Kendali Fuzzy. ....	4
Gambar 2.3 Kamera digital dan citra digital.....	6
Gambar 2.4 Proses dasar pengolahan citra. ....	7
Gambar 2.5 Pemetaan intensitas citra.....	8
Gambar 2.6 Mikrokontroler Atmega8535. ....	12
Gambar 2.7 Karakteristik Sinyal RS-232. ....	13
Gambar 2.8 Konektor tipe: MIL-C-24308, no.DB-9(F).....	14
Gambar 2.9 Motor DC. [3].....	15
Gambar 2.10 Luas penampang penghantar. [3] .....	16
Gambar 3.1 Blok diagram sistem. ....	18
Gambar 3.2 Kedudukan komponen pada alat. ....	20
Gambar 3.3 Rangkaian mikrokontroler. ....	21
Gambar 3.4 Rangkaian RS-232.....	22
Gambar 3.5 Bentuk sinyal PWM.....	23
Gambar 3.6 Rangkaian optocoupler. ....	25
Gambar 3.7 Rangkaian H-bridge sederhana. ....	25
Gambar 3.8 Blok diagram sistem sensor kamera. ....	26
Gambar 3.9 Pola Objek; (a) Pola mobile robot; (b) Pola objek penghalang. ....	28
Gambar 3.10 Posisi kamera.....	28
Gambar 3.11 Flowchart program. ....	29
Gambar 3.13 Koordinat objek.....	31
Gambar 3.14 Sudut objek.....	32
Gambar 3.15 Blok diagram pengenalan objek. ....	32
Gambar 3.16 Fungsi keanggotaan berbentuk trapezoid.....	34
Gambar 3.17 Fungsi keanggotaan untuk <i>sudut_tuj</i> .....	35
Gambar 3.18 Fungsi keanggotaan untuk <i>jarak_tuj</i> .....	35
Gambar 3.19 Pembagian <i>sudut_tuj</i> .....	36

Gambar 3.20 Fungsi keanggotaan untuk kecepatan .....	37
Gambar 3.21 Flowchart pengendalian dengan penghalang. ....	38
Gambar 4.1 Tampilan teminal CodeVisionAVR. ....	40
Gambar 4.2 Pengujian sistem sensor kamera.....	41
Gambar 4.3 Keadaan pengujian tanpa objek penghalang.....	43
Gambar 4.4. Pengujian sistem tanpa penghalang dengan <i>sudut_tuj</i> awal 295.43. 44	44
Gambar 4.5 Pengujian sistem tanpa penghalang dengan <i>sudut_tuj</i> awal 221.29.. 45	45
Gambar 4.6. Pengujian sistem tanpa penghalang dengan <i>sudut_tuj</i> awal 295.52. 46	46
Gambar 4.7 Keadaan pengujian dengan objek penghalang. ....	47
Gambar 4.8 Pengujian tanpa objek penghalang. ....	48
Gambar 4.9 Pengujian dengan objek penghalang. ....	49



## DAFTAR TABEL

	Halaman
Tabel 2.1. Arah Sinyal .....	14
Tabel 3.1 Aturan fuzzy untuk kecepatan .....	37
Tabel 4.1. Hasil pengujian rangkaian pengendali motor. ....	41
Tabel 4.2. Hasil pengujian sistem sensor kamera.....	42
Tabel 4.3. Hasil pengujian tanpa objek penghalang.....	43
Tabel 4.4. Pengujian sistem tanpa penghalang dengan <i>sudut_tuj</i> awal 295.43. ...	44
Tabel 4.5. Pengujian sistem tanpa penghalang dengan <i>sudut_tuj</i> awal 221.29. ...	45
Tabel 4.6. Pengujian sistem tanpa penghalang dengan <i>sudut_tuj</i> awal 295.52. ...	45
Tabel 4.7. Pengujian tanpa objek penghalang.....	47
Tabel 4.8. Pengujian dengan objek penghalang.....	48

## DAFTAR LAMPIRAN

	Halaman
Lampiran 1 Listing Program <i>main_program.m</i> .....	54
Lampiran 2 Listing Program <i>pengenalan_cam.m</i> .....	55
Lampiran 3 Listing Program <i>kendali_fuzzy.m</i> .....	57
Lampiran 4 Skema Rangkaian .....	59
Lampiran 5 Foto Pengujian.....	61
Lampiran 6 Foto Alat .....	62

## DAFTAR SINGKATAN

DTE	Data Terminal Equipment
DCE	Data Circuit-Termination Equipment
IC	Integrated Circuit
TTL	Transistor–Transistor Logic
PWM	Pulse Width Modulation
USB	Universal Serial Bus
RGB	Red Green Blue
CPU	Central Processing Unit
SRAM	Static Read Only Memory
EEPROM	Electrically Erasable Programmable Read Only Memory
USART	Universal Serial Asynchronous Receive Transmit

## DAFTAR ISTILAH

Motor DC	Motor Arus Searah
SRAM	Memori untuk menyimpan data secara <i>volatile</i> oleh program yang sedang <i>running</i> dan sifatnya statis.
EEPROM	Memori untuk penyimpanan data secara permanen
<i>Flash</i>	Jenis ROM yang biasanya diisi dengan program hasil buatan yang harus dijalankan <i>microcontroller</i>
Timer	Modul yang bekerja untuk menghitung waktu

Lasti Warasih H.  
NPM 04 04 03 057 1

Dosen Pembimbing  
Dr. Abdul Muis, ST, M.Eng

Departemen Teknik Elektro

**PERANCANGAN *MOBILE ROBOT* DENGAN SENSOR KAMERA  
MENGUNAKAN SISTEM KENDALI FUZZY**

**ABSTRAK**

Manusia selalu ingin menciptakan robot yang dapat bernavigasi seperti dirinya. Manusia dapat bernavigasi dengan hanya menggunakan informasi yang didapat melalui indera penglihatan. Untuk itu perlu diterapkan sistem penginderaan yang didasarkan pada pengelihatn pada suatu sistem navigasi robot. Sistem penginderaan ini diterapkan pada robot menggunakan suatu jenis sensor untuk menangkap citra, yaitu sensor kamera.

Pada skripsi ini dibahas penggunaan kamera sebagai sensor pada sistem pengendalian *mobile robot* atau robot bergerak. Sebelum digunakan sebagai dasar pengendalian, informasi yang didapat dari sensor kamera harus diproses terlebih dahulu. Pengendalian yang digunakan untuk navigasi robot adalah pengendalian berdasarkan logika fuzzy.

Dalam tugas akhir ini diperlihatkan bagaimana kemampuan sistem dalam memproses masukan dari sensor kamera dan menterjemahkannya menjadi gerakan robot menggunakan pengendalian fuzzy.

Sistem yang ingin dicapai adalah sistem robot yang dapat bernavigasi berdasarkan pengendalian fuzzy dengan hanya menggunakan input dari kamera. Dari hasil pengujian dapat diketahui bahwa dapat diterapkan sebuah sistem kendali fuzzy untuk mengendalikan *mobile robot* dengan hanya menggunakan sensor kamera.

**Kata kunci : Robot, Robot bergerak, Sensor kamera, Sistem navigasi, Kendali fuzzy**

Lasti Warasih H.  
NPM 04 04 03 057 1

Counselor  
Dr. Abdul Muis, ST, M.Eng

Electrical Engineering Departement

## **MOBILE ROBOT DESIGN WITH CAMERA SENSOR USING FUZZY CONTROL SYSTEM**

### **ABSTRACT**

Human beings always try to create robot that can navigate like them self. Humans can navigate with information from their vision. To achieve it, a vision based system is needed for robot navigation. This vision based system is applied on a robot with a type of image capturing sensor, which is a camera.

This paper discusses the use of camera as a sensor in the mobile robot control system. Before used as the input of the navigation control system, the information from the camera sensor has to be processed beforehand. The control system that is used for the robot navigation is a control system based on fuzzy logic.

This paper will show the system ability to process input from the camera sensor and translate that input into robot movement using fuzzy control.

The final goal is a mobile robot with fuzzy control system that can navigate with input only from camera sensor. From the experiment we can see that a fuzzy control system can be applied on a mobile robot with camera sensor.

**Keywords : Robot, Mobile Robot, Camera sensor, Navigation system, Fuzzy control**



# BAB I

## PENDAHULUAN

### 2.1 LATAR BELAKANG

Sudah merupakan cita-cita banyak ilmuwan untuk dapat membuat robot yang menyerupai salah satu kemampuan manusia. Manusia dapat bernavigasi dengan hanya menggunakan informasi yang didapat melalui indra penglihatan.

Untuk dapat bernavigasi robot juga membutuhkan indra untuk mengetahui keadaan sekitarnya dengan menggunakan sensor. Sensor yang sering digunakan dalam sistem navigasi robot adalah sensor sentuh dan sensor jarak, tergantung pada kebutuhan dimana robot itu didesain untuk bekerja. Sedangkan sensor kamera masih jarang digunakan, karena informasi yang dihasilkan oleh sensor ini tidak dapat langsung digunakan.

Informasi yang dihasilkan dari sensor kamera harus diproses terlebih dahulu. Sedangkan pada umumnya robot hanya memiliki kemampuan pemrosesan yang sedikit/ terbatas.

Dengan digunakannya sensor kamera, maka dapat diperoleh banyak informasi yaitu jarak, sudut, jenis objek, bahkan kecepatan. Sehingga sensor ini sering disebut "all in one". Banyaknya informasi yang dapat diperoleh dari sensor ini membuat robot yang menggunakannya dapat beroperasi tidak hanya pada satu lingkungan, tapi dapat beradaptasi tergantung dengan lingkungan yang dihadapinya.

### 2.2 PERUMUSAN MASALAH

Masalah yang akan diteliti pada skripsi ini adalah perancangan *mobile robot* dengan menggunakan sensor kamera. Dimana informasi yang didapat dari sensor kamera ini menjadi dasar pengendalian *mobile robot* berdasarkan logika fuzzy.

### 2.3 TUJUAN PENULISAN

Tujuan dari penulisan ini adalah untuk merancang suatu *mobile robot* dengan sistem navigasi menggunakan sensor kamera.

### 2.4 BATASAN MASALAH

Dalam skripsi ini penulis memberikan batasan masalah untuk memudahkan pembahasan. Perancangan *mobile robot* ini dirancang untuk menemukan target dengan sensor kamera menggunakan sistem kendali fuzzy.

### 2.5 METODOLOGI PENULISAN

Langkah-langkah penelitian yang dilakukan adalah sebagai berikut:

1. Penelitian Pustaka, yang meliputi:
  - a. Perumusan ide pokok.
  - b. Mengumpulkan dan mempelajari teori dasar yang dapat dijadikan penunjang dari berbagai sumber, seperti buku-buku, jurnal-jurnal, artikel-artikel, dsb
2. Penelitian Laboratorium, meliputi:
  - a. Perancangan software dengan MATLAB.
  - b. Pengujian alat.

### 2.6 SISTEMATIKA PENULISAN

Adapun penulisan skripsi ini dibagi dalam lima bab. Bab I ini membahas tentang latar belakang, tujuan, pembatasan masalah, serta sistematika penulisan skripsi. Kemudian pada Bab II memuat teori-teori yang berhubungan dengan perancangan dan implementasi sistem. Bab III membahas langkah-langkah perancangan yang dilakukan. Pada Bab IV dilakukan analisa terhadap hasil pengujian sistem yang dirancang. Dan terakhir, Bab V berisi kesimpulan dari keseluruhan isi skripsi

## BAB II

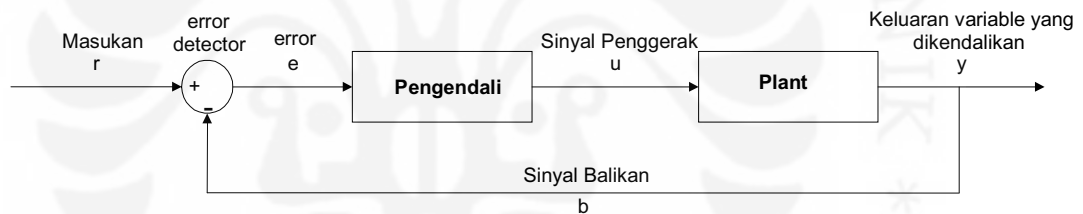
### DASAR TEORI

#### 2.1 SISTEM KENDALI

Pada umumnya sistem kendali dibagi menjadi dua kelompok :

1. Sistem Kendali Terbuka (*Open Loop Control System*)
2. Sistem Kendali Tertutup (*Close Loop Control System*)

Sistem kendali tertutup adalah sistem kendali dimana keluaran dari sistem akan mempengaruhi aksi pengendalian. Jadi keluarannya diumpanbalik untuk dibandingkan dengan masukan. Blok diagram sistem pengaturan tertutup dapat diperlihatkan pada gambar 2.1.



Gambar 2.1 Diagram blok sistem kendali tertutup.

Sistem ini merupakan perbaikan sistem kendali terbuka. Pada sistem kendali terbuka tidak ada elemen umpan balik dari variable yang dikendalikan ke masukan. Pada sistem kendali tertutup, sinyal umpan balik  $b$  dibandingkan dengan masukan  $r$  pada pembanding dan didapat sinyal  $e$  yang dimasukkan ke pengendali sehingga akan mengeluarkan sinyal penggerak  $u$  yang proporsional terhadap selisih masukan dan keluaran untuk memperbaiki galat.

##### 2.1.1 Sistem Kendali Fuzzy

Sistem kendali fuzzy adalah salah satu bentuk sistem kendali. Pada sistem kendali ini, nilai yang diproses didalamnya bukanlah nilai analog yang kontinu (*crisp*), tetapi merupakan nilai keadaan yang *fuzzy*.

Sama halnya dengan sistem kendali secara umum, pada sistem kendali fuzzy juga terdapat sistem kendali tertutup dan terbuka. Pada sistem kendali tertutup, terdapat sinyal umpan balik yang dibandingkan kembali.

Perbedaan antara sistem kendali fuzzy dengan sistem kendali biasa adalah sebelum masuk ke proses pengendalian, dilakukan proses fuzzifikasi. Proses fuzzifikasi adalah proses pengubahan nilai analog yang diterima dari sensor atau input lain ke dalam set nilai fuzzy. Setelah proses pengendalian selesai, sebelum masuk ke *plant*, dilakukan proses defuzzifikasi. Proses defuzzifikasi ini merupakan kebalikan dari fuzzifikasi, dimana nilai fuzzy yang didapatkan dari proses pengendalian diubah kembali ke nilai analog yang dapat diterima oleh *plant*.



Gambar 2.2 Diagram Blok Sistem Kendali Fuzzy.

## 2.2 ROBOT

Kata robot biasa digunakan untuk menjelaskan berbagai jenis mesin, yang memiliki kemampuan untuk bergerak dan dapat digunakan untuk melakukan pekerjaan fisik. Robot memiliki berbagai bentuk, mulai dari *humanoid*, menyerupai bentuk fisik dan cara bergerak manusia, sampai robot industri yang bentuknya sangat dipengaruhi oleh fungsinya.

Robot dapat dikendalikan secara langsung oleh manusia, seperti robot penjinak bom, atau bersifat *autonomous*, dapat mengambil keputusan sendiri dengan kepiintaran buatan.

Kata robot sendiri berasal dari bahasa Czech "*robota*", budak pekerja, diperkenalkan pertama kali oleh Karel Capek, seorang penulis drama berkebangsaan Czech, pada sebuah dramanya yang berjudul *Rossum's Universal Robots*(1920). [1]

Secara umum robot dapat dikelompokkan menjadi:

1. *mobile robot* ( robot bergerak)
2. manipulator robot

### 2.2.1 Mobile Robot

*Mobile robot* berarti robot yang dapat bergerak atau berpindah tempat.

Struktur sebuah mobile robot yang bergerak di darat dapat dipilah menjadi :

- Sistem mekanik, bagian ini meliputi bentuk dasar, sistem pergerakan, susunan *steering* sensor.
- Sistem pengendali pergerakan, bagian ini merupakan gabungan antara prinsip-prinsip algoritma dan alat yang bertujuan untuk melangkah pada arah tertentu, mengubah arah gerakan dan menambah atau mengurangi kecepatan. Oleh sebab itu kerja pengendalian sistem meliputi perencanaan pergerakan dan navigasi.
- Sistem sensor, bagian ini berfungsi untuk mengenali lingkungan dan menjadi masukan untuk sistem pengendali gerakan.
- Sistem pengetahuan, bagian ini merupakan algoritma pengambilan keputusan pada robot bergerak.

### 2.2.2 Manipulator Robot

Manipulator robot adalah robot yang didisain untuk memindahkan material, komponen-komponen, dan peralatan. Contoh dari maipulator robot adalah robot industri.

Manipulator robot memiliki *end effector* yang yang berfungsi untuk memanipulasi objek. *End effector* dapat berupa alat las, tangan penjepit, dll.

### 2.3 SENSOR

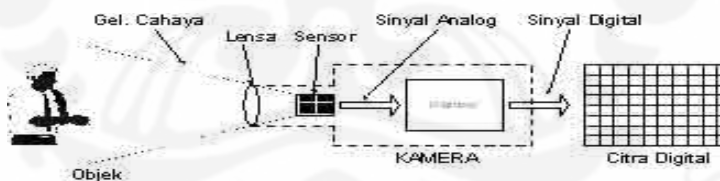
Sensor adalah alat untuk mengukur suatu nilai fisik untuk kemudian dijadikan suatu sinyal yang bisa terbaca oleh sistem. Untuk hubungannya dengan *mobile robot*, ada beberapa jenis sensor yang digunakan untuk mengetahui berbagai macam parameter penting. Pada *mobile robot*, parameter yang diukur diantaranya parameter kecepatan, parameter jarak, parameter posisi, dan seterusnya.

### 2.3.1 Sensor Kamera

Kamera adalah alat untuk mengakuisisi citra digital. Citra digital yang ditangkap oleh kamera tersebut dapat diproses sehingga didapatkan banyak parameter yang dibutuhkan oleh sistem pengendalian robot, diantaranya parameter posisi, parameter bentuk, bahkan parameter kecepatan.

Kamera adalah alat untuk menangkap gelombang cahaya yang dipantulkan oleh obyek, sehingga obyek tersebut dapat dipetakan dalam bentuk citra. Pada kamera digital citra yang dihasilkan merupakan citra dalam bentuk digital. Secara umum kamera digital terdiri dari lensa, sensor cahaya, dan *digitizer* seperti yang diperlihatkan pada gambar 2.3.

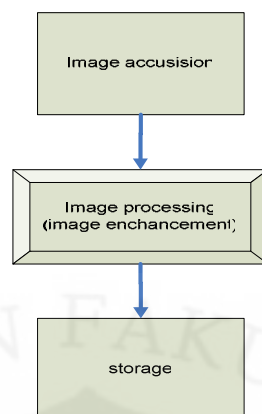
Lensa pada kamera berguna untuk memfokuskan cahaya yang diterima sehingga tepat mengenai bidang sensor cahaya. Sensor cahaya yang terdapat pada kamera merupakan kumpulan dari banyak sensor cahaya (phototransistor / photodiode) yang disusun dalam bentuk matrik. Sinyal analog yang dihasilkan oleh sensor lalu dikonversikan ke dalam bentuk digital oleh *digitizer* sehingga sinyal digital inilah yang dihasilkan oleh kamera digital.



Gambar 2.3 Kamera digital dan citra digital.

## 2.4 PENGOLAHAN CITRA DIGITAL

Pengolahan citra digital merupakan salah satu bentuk dari pemrosesan informasi digital dimana input dan outputnya merupakan citra, seperti foto atau video. Proses dasar dari pengolahan citra terdiri dari beberapa bagian, yaitu akuisisi citra, perbaikan citra, dan penyimpanan. Hal tersebut dapat dilihat pada gambar 2.4.



Gambar 2.4 Proses dasar pengolahan citra.

### 2.4.1 Akuisisi Citra

Sebuah alat akuisisi, seperti kamera digital, mikroskop elektronik, dan *scanner MRI* menghasilkan gambar digital. Gambar inilah yang akan digunakan sebagai input dari pengolahan citra digital.

Gambar yang dihasilkan dari proses akuisisi citra ini biasanya tidak sempurna, karena terdapat gangguan (*noise*) seperti distorsi, dan hilangnya sebagian dari sample.

### 2.4.2 Perbaikan Citra (*Image enhancement*)

Tujuan dari perbaikan citra adalah untuk memperbaiki tampilan visual dari citra. Pada tahap ini dilakukan perubahan persepsi dari citra, dengan cara memetakan kembali intensitas citra.

#### 2.4.2.1 Transformasi Grayscale

Transformasi grayscale sering disebut *point operation*, karena pada proses ini nilai intensitas suatu titik dipetakan ke titik itu kembali oleh suatu fungsi.

$$(n_1, n_2) \Rightarrow T(f(n_1, n_2)) \quad (2.1)$$

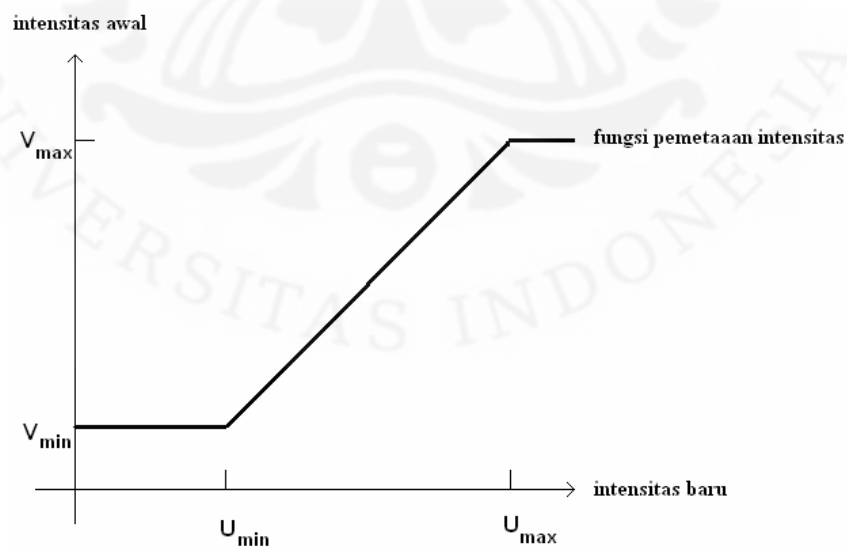
Transformasi grayscale hanya dapat menaikkan atau menurunkan intensitas dari citra. Transformasi ini biasa digunakan untuk mengembalikan informasi visual pada citra yang mengalami *overexposed* atau *underexposed*.

#### 2.4.2.2 Contrast Stretching

Pada proses ini intensitas dari citra dipetakan ke range yang baru dengan menggunakan fungsi sebagai berikut.

$$T(u) = \begin{cases} v_{\min} & u < u_{\min} \\ v_{\min} + \frac{(v_{\max} - v_{\min})(u - u_{\min})}{u_{\max} - u_{\min}} & u_{\min} < u < u_{\max} \\ v_{\max} & u > u_{\max} \end{cases} \quad (2.2)$$

Intensitas yang berada di antara  $u_{\min}$  dan  $u_{\max}$  dipetakan secara linier ke nilai intensitas yang baru yaitu antara  $v_{\min}$  dan  $v_{\max}$ . Seperti yang ditunjukkan pada gambar 2.3.



Gambar 2.5 Pemetaan intensitas citra.



### 2.4.2.3 Perbaikan Gamma

Proses perbaikan gamma dapat diwakilkan oleh fungsi nonlinier dibawah ini:

$$v = \frac{u^\gamma}{M^\gamma} \quad (2.3)$$

Dimana  $u$  fungsi intensitas awal pada range  $[0, M]$ . Jika  $\gamma$  lebih kecil dari 1, maka semua intensitas akan ditingkatkan, dan citra terlihat lebih terang. Nilai intensitas yang lebih kecil pada citra akan lebih diperbesar dibandingkan dengan intensitas yang besar. Jika  $\gamma$  lebih besar dari 1, maka citra terlihat lebih gelap, karena nilai intensitas yang besar lebih diperkecil dibandingkan dengan intensitas yang besar.

### 2.4.3 Penyimpanan Citra (*Storage*)

Penyimpanan citra meliputi kompresi citra (*image compression*) dengan cara pengkodean citra (*image coding*). Hal ini bertujuan untuk mengurangi kelebihan data, yang dianggap tidak penting, agar citra tersebut dapat disimpan secara efisien.

Citra digital secara umum terdiri dari ribuan pixel yang masing-masing membutuhkan banyak bit data. Sebagai contoh, citra dengan besar 500x500 pixel dengan 8 bit/pixel membutuhkan sebesar 2 Mbit untuk penyimpanan. Oleh karena itu citra digital perlu dikompresi, dikarenakan dua hal:

1. Untuk menghemat *bandwidth* dalam proses transmisi.
2. Untuk menghemat tempat pada media penyimpanan.

### 2.4.4 Pengenalan Citra Digital

Dalam mengenali citra digital dilakukan metode yang disebut dengan pengenalan pola (*Pattern Recognition*). Dengan mengenali suatu pola-pola tertentu dari suatu citra, maka bisa dilakukan klasifikasi dari citra tersebut.

Pengenalan pola tidak terbatas pada pengenalan citra digital saja, namun untuk berbagai macam sistem, misalnya pengenalan suara. Tujuan utama dari pengenalan pola adalah untuk mengenali suatu lingkungan, kemudian mempelajari suatu pola dari lingkungan tersebut, dan melakukan klasifikasi dari pola tersebut ke dalam kategori-kategori tertentu.

Metode pengenalan pola ini didasarkan pada kemampuan manusia yang mampu mengenali objek-objek yang ada. Manusia bisa mengenali objek berdasarkan ciri-ciri dan pengetahuan yang pernah diketahuinya dari objek tersebut.

Ada dua macam pendekatan pada *Pattern Recognition* yaitu :

- Pendekatan Struktural
- Pendekatan Statistik

#### 2.4.4.1 Pendekatan Struktural

Pendekatan struktural adalah pengenalan pola dimana data yang ada dibandingkan dengan aturan-aturan pola yang sudah diketahui sebelumnya. Aturan-aturan mengenai pola ini didapatkan dari hubungan kompleks yang ada antara bagian-bagian dari struktur data.

Sehingga pengenalan pola pendekatan struktural hanya bisa dilakukan jika pola yang ada mempunyai struktur yang jelas. Sehingga struktur tersebut bisa diidentifikasi keberadaannya pada data yang ada.

Pada pengenalan pola suatu data citra digital, pendekatan struktural dilakukan dengan membandingkan struktur matrik dari citra dengan struktur pola yang diketahui sebelumnya.

Proses pengenalan pola menggunakan pendekatan struktural sangat tergantung dari struktur pola yang ada. Semakin rumit pola yang ada, semakin rumit pula proses yang ada. Namun, jika pola terlalu sederhana, pola ini menjadi tidak representatif terhadap pola yang kita inginkan. Sehingga pada pendekatan struktural ini, penentuan pola memegang peranan penting.

#### 2.4.4.2 Pendekatan Statistik

Secara garis besar, pendekatan statistik adalah pengenalan pola berdasarkan informasi statistik yang didapat dari suatu pola. Berbeda dengan pendekatan struktural, pada pendekatan statistik ini, tidak diperlukan struktur yang jelas dari suatu pola.

Pendekatan statistik dilakukan dengan melihat kemiripan antara data yang ada dan pola yang diketahui, menggunakan informasi statistik seperti simpangan, sebaran, dan lain-lain. Dengan informasi tersebut, bisa dilakukan klasifikasi dari data-data yang ada.

Pendekatan statistik dapat dilakukan menggunakan berbagai metode, mulai yang paling sederhana sampai yang paling rumit seperti Jaringan Saraf Tiruan (JST). Pada Jaringan Saraf Tiruan, sistem pemrosesan informasinya diadaptasi dari arsitektur dan cara kerja dari jaringan saraf biologis.

### 2.5 MIKROKONTROLER ATMEGA8535 [2]

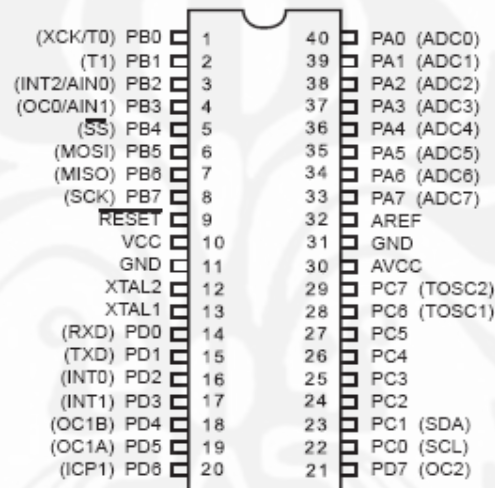
Mikrokontroler digunakan sebagai saluran masukan atau keluaran (I/O), transmisi data serial, dan pengolah / penyimpan data. Alasan penggunaan Mikrokontroler adalah dengan sudah tergabungnya saluran I/O dalam satu *Integrated Circuit* (IC), maka membuat sistem ini menjadi kompak. Mikrokontroler yang digunakan adalah tipe ATmega8535 yang termasuk dalam keluarga AVR. Mikrokontroler keluarga AVR ini menggunakan bahasa pemrograman tingkat tinggi yang mudah dipahami manusia yaitu bahasa C.

Jenis Mikrokontroler tersebut mempunyai spesifikasi sebagai berikut :

- Frekuensi clock maksimum 16 MHz
- Jalur I/O 32 buah, yang terbagi dalam PortA, PortB, PortC dan PortD
- Analog to Digital Converter 10 bit sebanyak 8 input
- Timer/Counter sebanyak 3 buah
- CPU 8 bit yang terdiri dari 32 register
- Watchdog Timer dengan osilator internal
- SRAM sebesar 512 byte

- Memori Flash sebesar 8 Kbyte dengan kemampuan *read while write*
- Interrupt internal maupun eksternal
- Port komunikasi SPI
- EEPROM sebesar 512 byte yang dapat diprogram saat operasi
- Analog Comparator
- Komunikasi serial standar USART dengan kecepatan maksimal 2,5 Mbps

Bentuk fisik unit Mikrokontroler tipe Atmega8535 beserta nama kakinya dapat dilihat pada Gambar 2.6



Gambar 2.6 Mikrokontroler Atmega8535.

## 2.6 STANDAR KOMUNIKASI DATA RS-232

RS-232 merupakan singkatan *Recommended Standard-232*, merupakan standar antar muka yang dikeluarkan oleh Electronic Industries Association (EIA) untuk komunikasi serial. Pada tahun 1991 EIA bekerjasama dengan Telecommunications Industry Association (TIA) mengeluarkan versi terbaru dengan nama EIA/TIA 232-E, tetapi banyak orang yang masih menyebutnya sebagai RS-232.

RS-232 merupakan bentuk komunikasi serial, ini berarti data dikirim satu-persatu melalui satu saluran. Peralatan yang menggunakan RS-232 dapat dibagi

menjadi dua kategori, yaitu *Data Terminal Equipment (DTE)* dan *Data Circuit-termination Equipment (DCE)*. DTE adalah peralatan yang berada jauh dari hubungan, biasanya computer atau terminal, sedangkan DCE adalah peralatan yang berada di ujung hubungan (antar muka saluran telpon), biasanya modem.

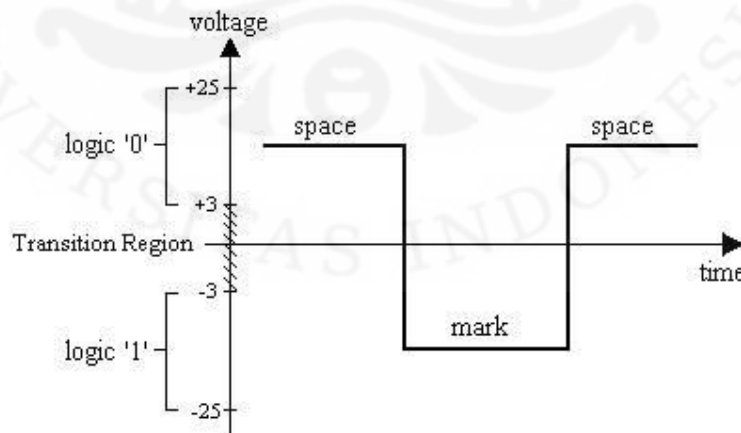
Tiga hal pokok yang diatur dalam RS-232 antara lain:

1. Bentuk sinyal dan level tegangan yang dipakai.
2. Penentuan jenis sinyal dan konektor yang dipakai, serta susunan sinyal pada kaki-kaki konektor.
3. Penentuan tata cara pertukaran informasi antara komputer dan alat-alat pelengkapanya.

### 2.6.1 Karakteristik Sinyal RS-232 dan Tegangan

RS-232 menggunakan standar yang berkebalikan dengan standar rangkaian digital sekarang, dimana tegangan negatif bernilai 1 dan tegangan positif bernilai 0, seperti terlihat pada gambar 2.7.

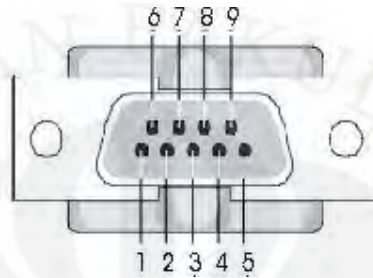
Tegangan -3 Volt sampai -25 Volt bernilai logika "1", tegangan +3 Volt sampai +25 Volt bernilai logika "0", dan tegangan +3 Volt sampai -3 Volt dinamakan *Transition Region*, berguna untuk menyerap *noise* yang terjadi selama komunikasi.



Gambar 2.7 Karakteristik Sinyal RS-232.

## 2.6.2 Tipe Konektor

RS-232 mempunyai dua jenis konektor, yaitu DB-25 dan DB-9, akan tetapi dalam perkembangan PC yang terakhir, hanya tipe DB-9 saja yang umum digunakan untuk komunikasi serial.



Gambar 2.8 Konektor tipe: MIL-C-24308, no.DB-9(F).

Arah sinyal konektor DB-9 dapat dilihat pada tabel 2.1.

Tabel 2.1. Arah Sinyal

Nama Sinyal	Arah Sinyal	Nomor Kaki Konektor DB-9
Signal Common	--	5
Transmitted Data (TD)	Ke DCE	3
Received Data (RD)	Dari DCE	2
Request to Send (RTS)	Ke DCE	7
Clear to Send (CTS)	Dari DCE	8
DCE Ready (DSR)	Dari DCE	6
DTE Ready (DTR)	Ke DCE	4
Ring Indicator (R)	Dari DCE	9
Data Carrier Detect (DCD)	Dari DCE	1

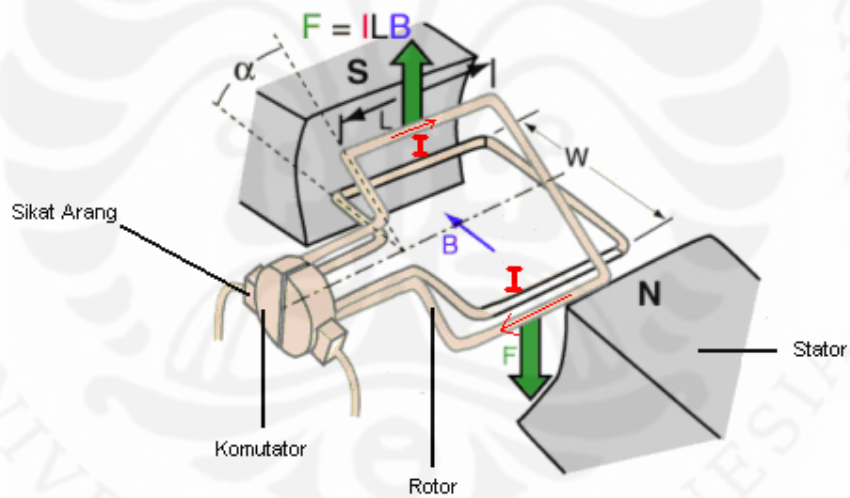
Pada tabel 2.1 terlihat arah sinyal-sinyal RS-232, ada yang menuju ke DTE ada pula yang berasal dari DTE. Bagi sinyal yang menuju ke DTE artinya DCE berfungsi sebagai output dan DTE berfungsi sebagai input, misalnya sinyal

*Transmitted Data* (TD), pada sisi DTE kaki TD adalah output, dan kaki ini dihubungkan ke kaki TD pada DCE yang berfungsi sebagai input. Kebalikan sinyal TD adalah *Received Data* (RD), sinyal ini berasal dari DCE dan dihubungkan ke kaki RD pada DTE yang berfungsi sebagai output.

## 2.7 MOTOR DC

Motor adalah suatu mesin listrik yang mengubah energi listrik menjadi energi mekanik. Energi listrik yang digunakan motor DC adalah listrik arus searah. Umumnya motor DC terdiri dari tiga bagian utama yaitu :

1. bagian yang diam disebut dengan stator
2. bagian yang berputar disebut dengan rotor
3. komutator dan sikat arang



Gambar 2.9 Motor DC. [3]

Adapun prinsip kerja dari motor DC adalah sebagai berikut: arus mengalir melalui kawat penghantar pada rotor berada di dalam medan magnet yang homogen akan timbul gaya Lorentz. Gaya ini timbul di tiap sisi yang tegak lurus terhadap medan magnet, yang besarnya dapat dirumuskan dengan :

$$F = B i l \sin\theta \quad (2.4)$$

Dimana :  $F$  = gaya Lorentz dalam Newton

$B$  = induksi magnet / medan magnet dalam Weber/m<sup>2</sup>

$I$  = kuat arus dalam Ampere

$L$  = panjang kawat dalam meter

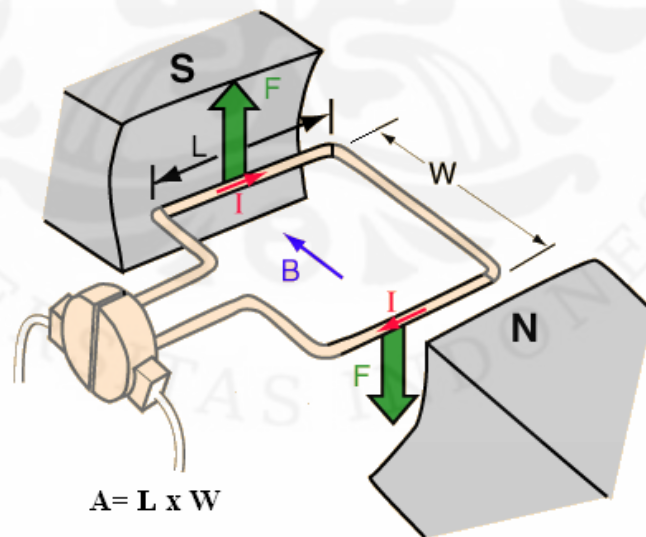
Gaya Lorentz yang timbul ini akan saling meniadakan karena pada dua sisi pada kawat penghantar yang berbentuk persegi empat itu akan timbul gaya Lorentz yang berlawanan. Oleh karena itu tidak akan menimbulkan gerak translasi, tetapi akan menimbulkan kopel dengan momen terhadap titik tengah dari sisi yang sejajar dengan arah  $B$ . Momen yang timbul itu besarnya adalah :

$$M = NiBA \sin \theta \quad (2.5)$$

Dimana :  $M$  = momen yang timbul

$N$  = jumlah lilitan

$A$  = luas penampang penghantar



Gambar 2.10 Luas penampang penghantar. [3]

Jika motor berputar, maka akan terjadi gaya gerak listrik (ggl) pada penghantar yang ada dalam jangkar. Ggl yang timbul ini dikarenakan kawat



penghantar yang berputar dalam medan magnet menyebabkan perubahan fluks magnetik yang dinyatakan dengan rumus di bawah ini:

$$\phi = BA \cos \theta \quad (2.6)$$

Dimana :  $\phi$  = fluks magnetik yang melalui suatu bidang  
 $B$  = komponen medan magnet yang tegak lurus bidang  
 $A$  = luas bidang yang dilalui medan magnetis  
 $\theta$  = sudut antara arah medan magnetis dan normal bidang

Oleh karena adanya perubahan medan magnet ini maka sesuai dengan hukum Faraday akan timbul suatu ggl imbas yang berbanding lurus dengan cepat perubahan fluks magnetis yang melalui kawat penghantar tersebut. Ggl imbas yang timbul dapat dituliskan dalam rumus sebagai berikut:

$$\varepsilon = \frac{-Nd\phi}{dt} \quad (2.7)$$

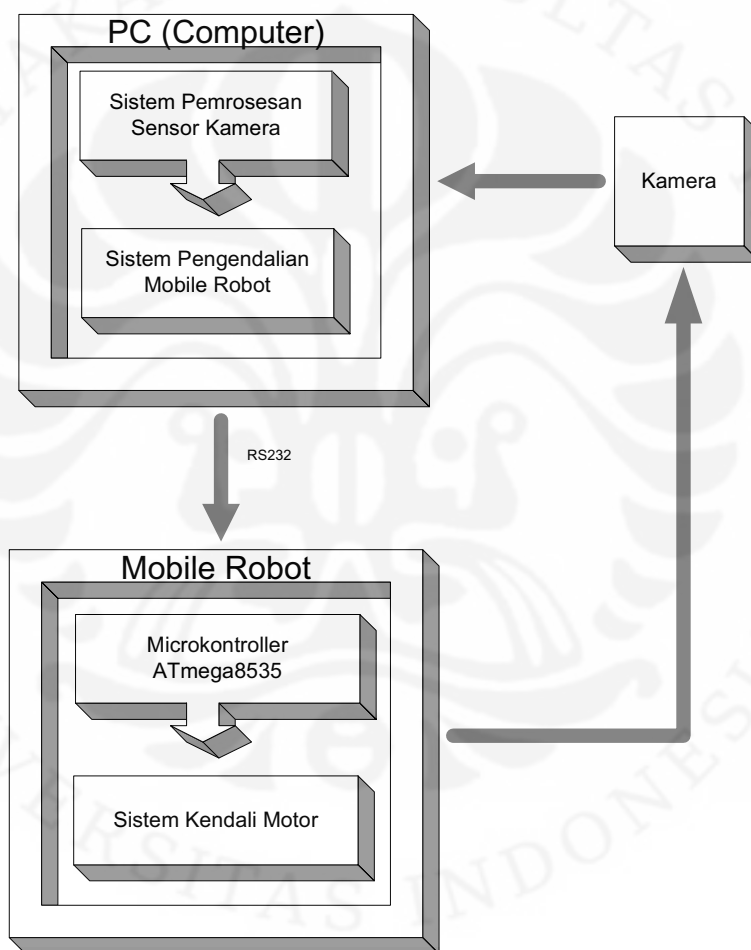
Dimana :  $N$  = jumlah lilitan kawat  
 $d\phi$  = perubahan fluks magnetis terhadap waktu

Tanda minus dalam persamaan Faraday dinyatakan oleh Lenz sesuai dengan hukum Lenz yaitu: Ggl imbas yang timbul akan menyebabkan arus induksi yang akan melawan penyebab timbulnya ggl imbas itu sendiri.

## BAB III

### PERANCANGAN SISTEM

Secara garis besar sistem dapat dibagi menjadi tiga bagian utama yaitu kamera, PC (*Personal Computer*), dan *mobile robot*.



Gambar 3.1 Blok diagram sistem.

Cara kerja sistem secara garis besar dapat dijelaskan sebagai berikut:

- Kamera berfungsi untuk sistem sensor yang akan mengambil citra sebagai umpan balik sistem.

- Pemrosesan hasil citra dari kamera dilakukan melalui komputer dengan menggunakan MATLAB, berfungsi untuk memproses gambar yang diterima dari kamera. Dan juga melakukan sistem pengendalian dari pergerakan *mobile robot*.
- *Mobile Robot*, melalui RS-232, menerima hasil pengendalian yang kemudian diterjemahkan oleh mikrokontroler ke sistem kendali motor menjadi kecepatan roda penggerak.

Perancangan sistem dapat dibagi menjadi 4 (empat) bagian yaitu perancangan konstruksi fisik, perancangan kendali motor, perancangan sistem sensor kamera, dan perancangan sistem pengendali *mobile robot*.

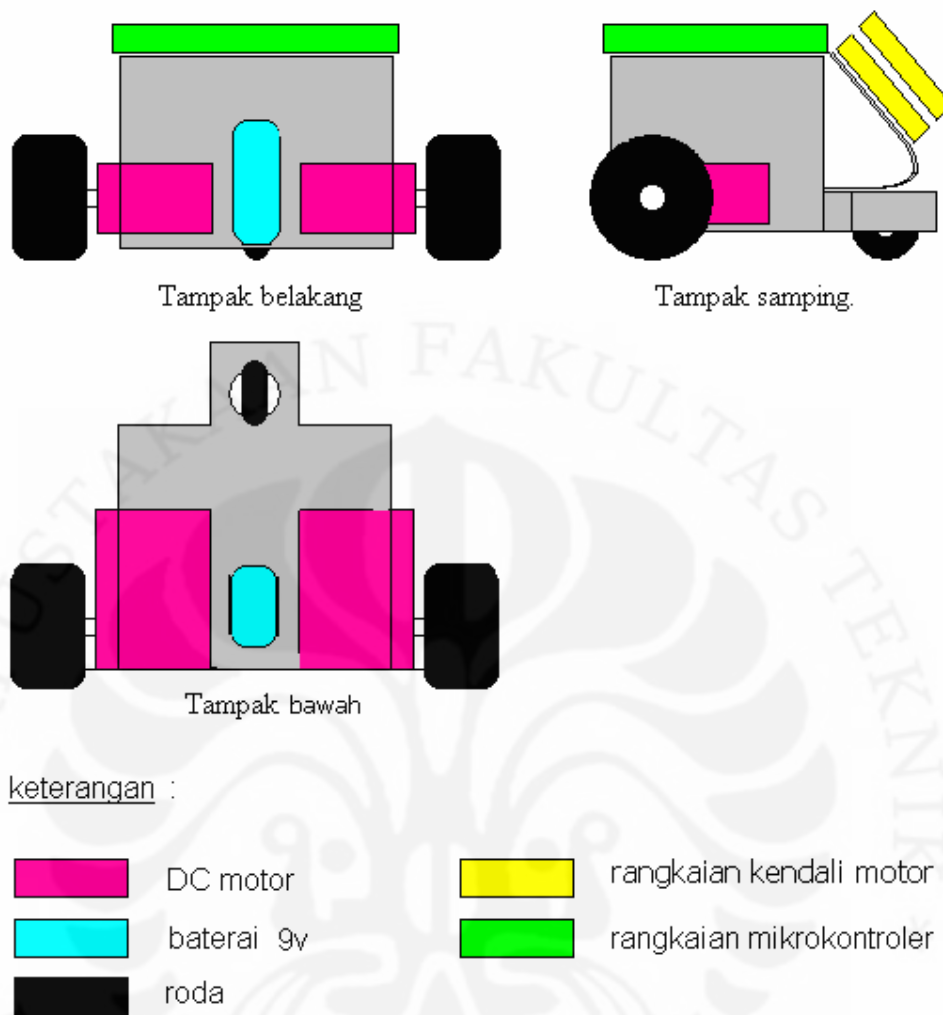
Sistem kendali motor merupakan pengendalian motor DC yang kemudian menggerakkan roda penggerak. Perancangan sensor kamera berkaitan dengan perancangan sistem pencarian dan pengenalan objek *mobile robot* dan objek sejenisnya. Sedangkan perancangan sistem pengendali *mobile robot* berkaitan dengan pengendalian *mobile robot* secara keseluruhan.

### 3.1 PERANCANGAN KONSTRUKSI FISIK

Konstruksi fisik dari *mobile robot* terdiri dari:

1. Sepasang motor DC.
2. Rangkaian mikrokontroler.
3. Rangkaian Kendali Motor.
4. Baterai 9V.
5. Sumber Catu Daya 12 volt.

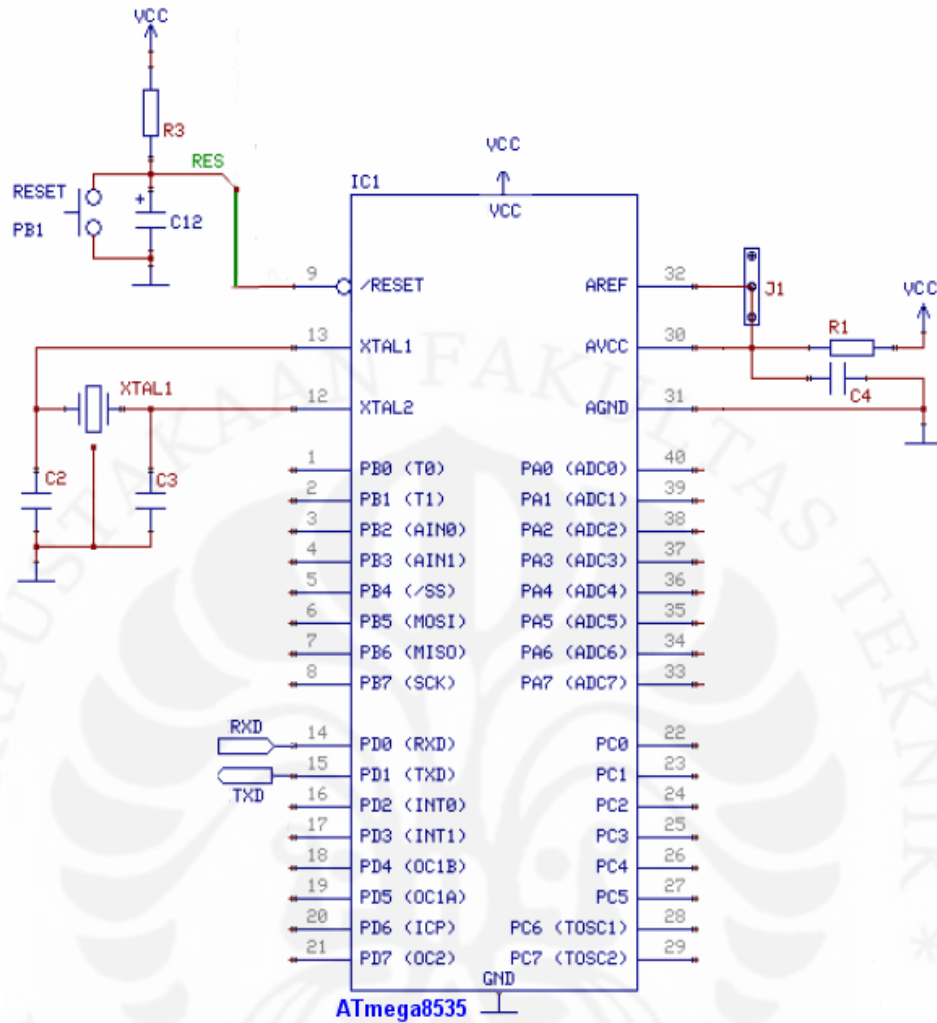
Motor DC diletakkan pada sisi kanan dan kiri dari badan robot, masing-masing langsung dihubungkan dengan roda penggerak. Rangkaian mikrokontroler berfungsi menerima data dari komputer dan menghasilkan sinyal untuk rangkaian kendali motor. Rangkaian kendali motor berfungsi untuk menghasilkan sinyal untuk mengendalikan motor. Sedangkan baterai 9V, digunakan sebagai sumber daya bagi rangkaian mikrokontroler.



Gambar 3.2 Kedudukan komponen pada alat.

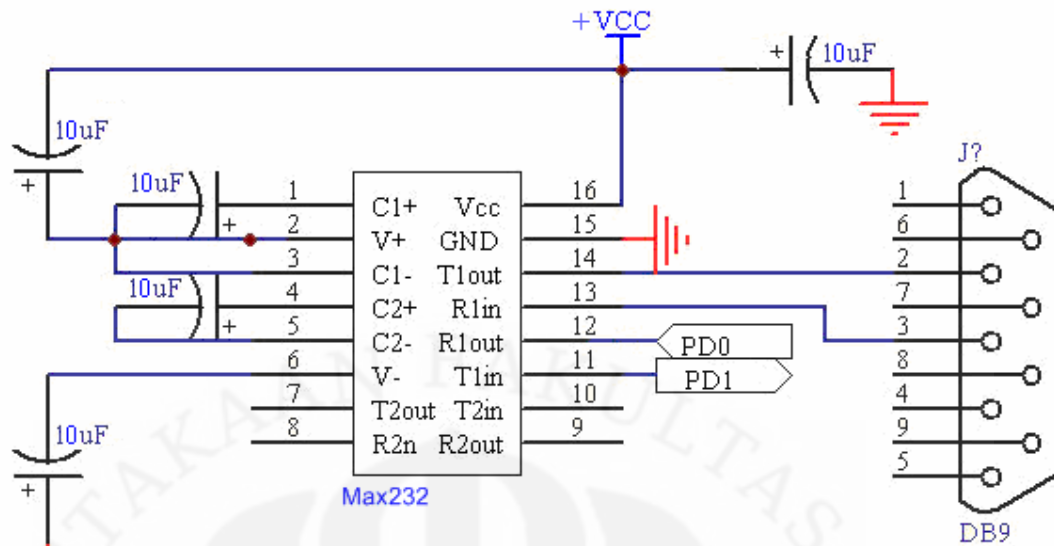
### 3.1.1 Rangkaian Mikrokontroler

Sebagai pengendali digunakan sebuah IC mikrokontroler ATmega8535. Seperti terlihat pada Gambar 3.3, rangkaian ini berfungsi untuk menerima dan mengolah data yang diterima dari komputer, dan menterjemahkannya menjadi sinyal untuk dikirimkan ke sistem penggerak.



Gambar 3.3 Rangkaian mikrokontroler.

Agar Mikrokontroler dapat berkomunikasi dengan komputer maka diperlukan sebuah penghubung yaitu RS-232. Untuk itu, dibutuhkan tambahan rangkaian yang terdiri dari kabel serial, IC Max232, dan kapasitor sebesar 10 uF



Gambar 3.4 Rangkaian RS-232.

Pada dasarnya kabel serial bisa langsung dipakai pada setiap aplikasi untuk mengadakan komunikasi antara alat dengan komputer, namun pada Mikrokontroler kabel serial tidak bisa begitu saja digunakan untuk melakukan komunikasi dengan komputer. Hal tersebut disebabkan karena komputer menggunakan tingkat sinyal yang berbeda. Rangkaian mikrokontroler menggunakan tingkat sinyal yang kompatibel dengan TTL (0 – 5 Volt) sedangkan komputer menggunakan tingkat sinyal yang tidak kompatibel dengan TTL. Dalam tingkat sinyal TTL, tegangan 0 Volt akan diartikan sebagai 0 (*low*) dan tegangan +5 Volt akan diartikan sebagai 1 (*high*) dalam format bilangan biner. Dalam sistem ini digunakan standar EIA RS-232 yang terdapat pada komputer, dimana tingkat sinyal yang digunakan adalah -25 sampai +25 Volt. Dalam standar RS-232 tegangan -25 sampai -3 Volt akan diartikan sebagai 1 (*high*) dan tegangan +25 sampai +3 Volt akan diartikan sebagai 0 (*low*) dalam format bilangan biner.

Untuk mengatasi perbedaan tingkat sinyal tersebut, maka diperlukan sebuah rangkaian konversi. Rangkaian ini terdiri dari IC Max 232 dan kapasitor 10 uF. Kaki 5 pada kabel serial dihubungkan ke *ground*.

### 3.2. PERANCANGAN SISTEM KENDALI MOTOR

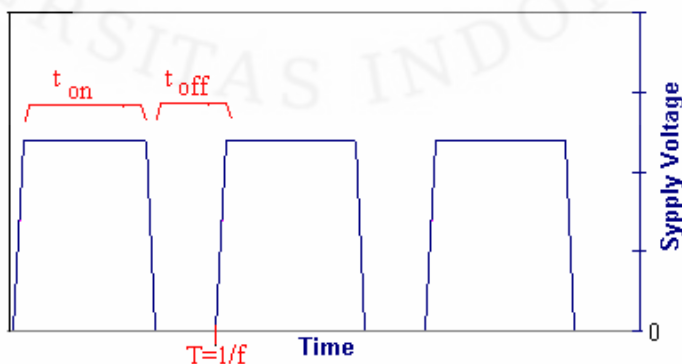
Sistem kendali motor berfungsi menerima sinyal dari mikrokontroler dan mengeluarkan sinyal yang digunakan langsung untuk menggerakkan motor. Untuk melakukan hal tersebut, dilakukan pengendalian kecepatan motor menggunakan PWM (*Phase Width Modulation*) yang kemudian menjadi masukan rangkaian kendali motor [5].

#### 3.2.1 Phase Width Modulation (PWM)

Dalam pengendalian motor, nilai yang harus dikendalikan adalah nilai kecepatan motor. Untuk mengendalikan kecepatan motor digunakan suatu sinyal yang merepresentasikan kecepatan motor yang diinginkan, sehingga motor diharapkan dapat bekerja dengan kecepatan tersebut.

Dalam mengendalikan kecepatan motor DC, dialirkan daya yang besarnya dapat diubah-ubah sesuai dengan kecepatan motor yang diinginkan. Metode yang digunakan untuk mengaplikasikan daya yang berubah-ubah ini berupa metode PWM (*Phase Width Modulation*).

Metode PWM ini merupakan metode yang menyerupai konsep *switching on-off*. Jika *switching on-off* ini dilakukan dengan frekuensi yang lebih cepat dari reaksi motor, maka akan didapatkan kecepatan motor yang stabil pada suatu nilai. Jadi, dengan mengatur perbandingan sinyal *on* dan *off* tersebut, bisa didapatkan rata-rata daya yang diinginkan untuk mengatur kecepatan motor. Dengan menambah lamanya sinyal *on*, akan didapatkan kecepatan motor yang bertambah dikarenakan rata-rata daya yang disuplai bertambah.



Gambar 3.5 Bentuk sinyal PWM.

Untuk mengukur besar PWM, digunakan perhitungan *duty cycle*. *Duty cycle* merupakan perbandingan antara waktu menyala ( $t_{on}$ ) dengan periode PWM tersebut ( $T$ ), biasa direpresentasikan dalam bentuk persen.

$$DutyCycle = \frac{t_{on}}{T} \times 100\% \quad (3.1)$$

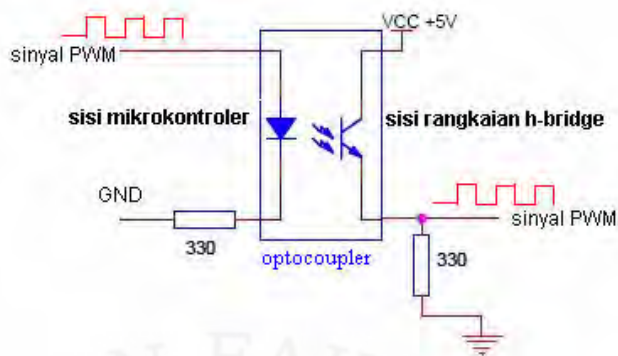
Pada perancangan ini, sinyal PWM dibangkitkan dengan menggunakan pemrograman mikrokontroler. Dengan menggunakan mikrokontroler, bisa didapatkan sinyal PWM yang lebih sederhana dan fleksibel besarnya.

### 3.2.2 Rangkaian Kendali Motor

Rangkaian sistem penggerak ini terbagi dari tiga bagian, yaitu rangkaian optocoupler, *H-bridge*, dan motor DC. Optocoupler berfungsi untuk menghubungkan antara pin mikrokontroler dengan rangkaian *H-bridge*. Sedangkan rangkaian *H-bridge* mempunyai fungsi utama agar motor DC yang dihubungkan, bisa bergerak dalam dua arah yang berlawanan. Motor yang mampu bergerak dalam dua arah yang berlawanan ini kemudian dihubungkan langsung dengan roda penggerak sehingga *mobile robot* bisa bergerak.

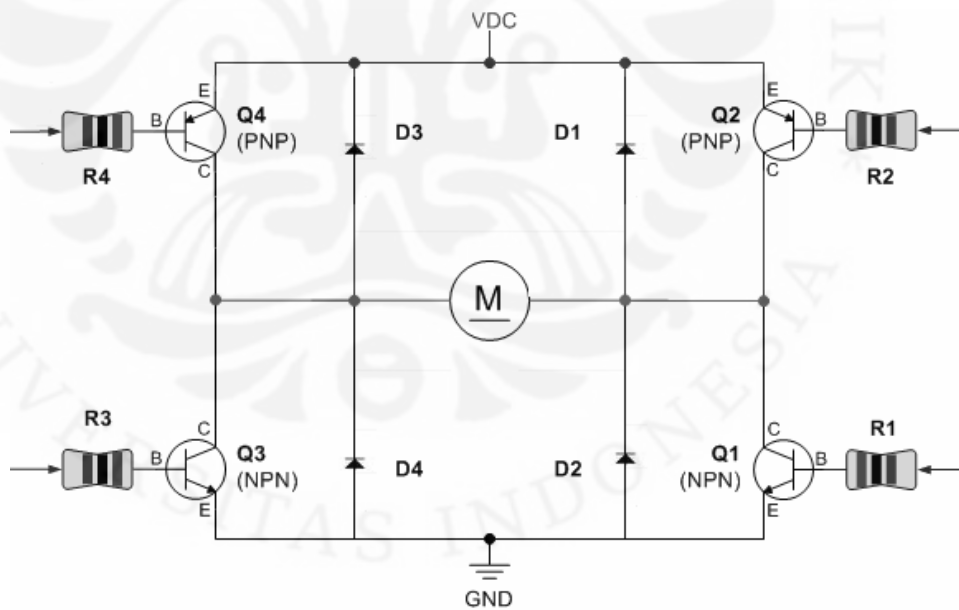
Agar sumber tegangan antara mikrokontroler dengan rangkaian *H-bridge* tidak terbagi, digunakan optocoupler untuk memisah kedua rangkaian tersebut. Optocoupler adalah suatu alat elektronika yang antara satu bagian dengan bagian lain tidak mempunyai hubungan listrik, namun terhubung secara optis. Sehingga dengan memasang optocoupler pada masing-masing pin mikrokontroler yang menjadi masukan rangkaian *H-bridge*, juga akan mencegah timbulnya arus balik yang bisa masuk ke mikrokontroler.





Gambar 3.6 Rangkaian optocoupler.

Prinsip dasar dari *H-bridge* adalah dengan menghubungkan motor dengan empat *switch*. *H-bridge* sederhana dengan menggunakan dua PNP transistor dan dua NPN transistor sebagai *switch* digambarkan pada gambar 3.7. Dengan mengubah keadaan dari empat *switch* tersebut didapatkan aliran arus ke motor yang berbeda arah, sehingga akan membuat motor berputar searah jarum jam atau sebaliknya.



Gambar 3.7 Rangkaian H-bridge sederhana.

Untuk memudahkan perancangan, rangkaian *H-bridge* digantikan dengan IC L298 yang bekerja dengan prinsip dasar yang sama. L298 adalah rangkaian terintegrasi yang merupakan *dual full-bridge driver*, dan dapat digunakan untuk

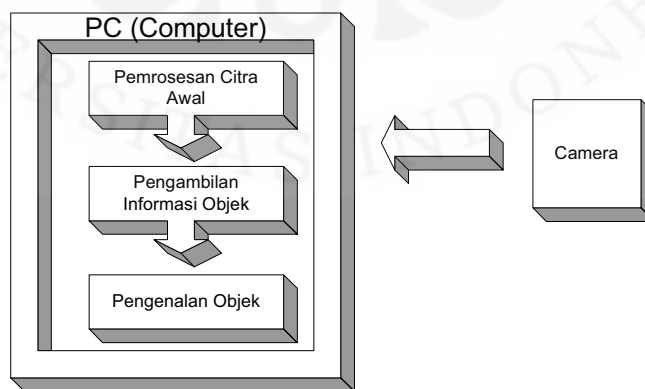
mengendalikan beban induktif, diantaranya motor DC. Untuk satu motor, input utamanya terdiri dari dua input *enable* dan satu input untuk sinyal. L298 bisa digunakan untuk mengendalikan hingga dua buah motor DC.

Pada rancangan ini, untuk masing-masing motor, kedua input *enable* diberi masukan PWM yang saling berlawanan. PWM ini merupakan output yang dihasilkan oleh mikrokontroler. PC.0 dan PC.2 menghasilkan dua saluran PWM yang saling berlawanan untuk motor kiri, sedangkan PC.4 dan PC.6 untuk motor kanan. Sedangkan input untuk sinyal dijadikan *enable*, sehingga diset *high* terus menerus.

### 3.3 PERANCANGAN SISTEM SENSOR KAMERA

Sistem sensor kamera ini akan menangkap citra yang didalamnya terdapat beberapa objek yang berbeda, yaitu objek *mobile robot* dan objek penghalang, jika ada. Tujuan sistem ini adalah mengenali masing-masing objek yang ditangkap kamera, selain itu juga untuk mendapatkan informasi koordinat dan sudut kemiringan masing-masing objek terhadap garis acuan. Karena itu, untuk mendapatkan informasi yang diinginkan dari sensor kamera, perlu dilakukan pemrosesan citra.

Sistem sensor kamera dilakukan dengan menggunakan kamera sebagai alat akuisisi citra, dan PC (*Personal Computer*) dengan MATLAB R2006b sebagai perangkat lunak pemrosesan citra.



Gambar 3.8 Blok diagram sistem sensor kamera.

Proses yang dilakukan oleh komputer yang didapat dari kamera dibagi menjadi urutan berikut :

- Pemrosesan citra awal
- Pengambilan informasi sudut dan posisi objek, dan
- Pengenalan objek

Namun, sebelum pemrosesan yang dilakukan oleh komputer dimulai, perlu dilakukan beberapa penyesuaian awal. Pertama-tama dilakukan perancangan pola dari objek yang akan dideteksi, dan juga perlu dilakukan kalibrasi ukuran antara keadaan sebenarnya dengan citra yang diakuisisi komputer.

### 3.3.1 Penyesuaian Awal

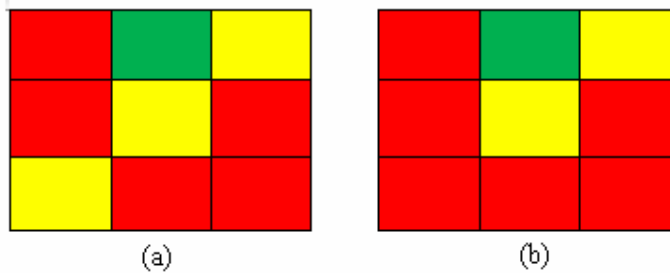
#### 3.3.1.1 Perancangan Pola

Dalam penyesuaian awal ini, pertama-tama ditentukan pola dari objek-objek yang akan dipakai pada sistem ini. Pola dari objek-objek ini harus mempunyai ciri-ciri khusus yang untuk kemudian dapat menjadi ciri-ciri yang membedakan antar objek, dan juga dapat mempermudah sistem untuk mendapat informasi posisi dan sudut objek

Objek berbentuk segiempat yang dibagi menjadi sembilan bagian kecil (blok). Sembilan bagian tersebut masing-masing diberi warna dengan tiga kemungkinan warna, yaitu hijau, merah, dan kuning, sehingga menghasilkan pola yang unik.

Warna hijau diletakkan pada salah satu blok untuk menunjukkan bagian depan dari objek, dimana hal ini berguna untuk menentukan sudut objek. Sedangkan komposisi warna merah dan kuning yang diletakkan pada kedelapan blok lainnya menunjukkan identitas dari objek.

Dengan kombinasi warna merah dan kuning pada kedelapan blok yang berbeda-beda, maka rancangan pola ini akan mampu merepresentasikan 28 objek yang berbeda. Pada perancangan ini ditentukan dua pola yang digunakan pada sistem, dimana pola pertama untuk *mobile robot* dan pola kedua untuk objek penghalang.

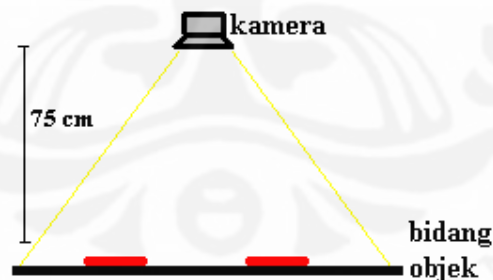


Gambar 3.9 Pola Objek; (a) Pola mobile robot; (b) Pola objek penghalang.

### 3.3.1.2 Akuisisi Citra dan Kalibrasi

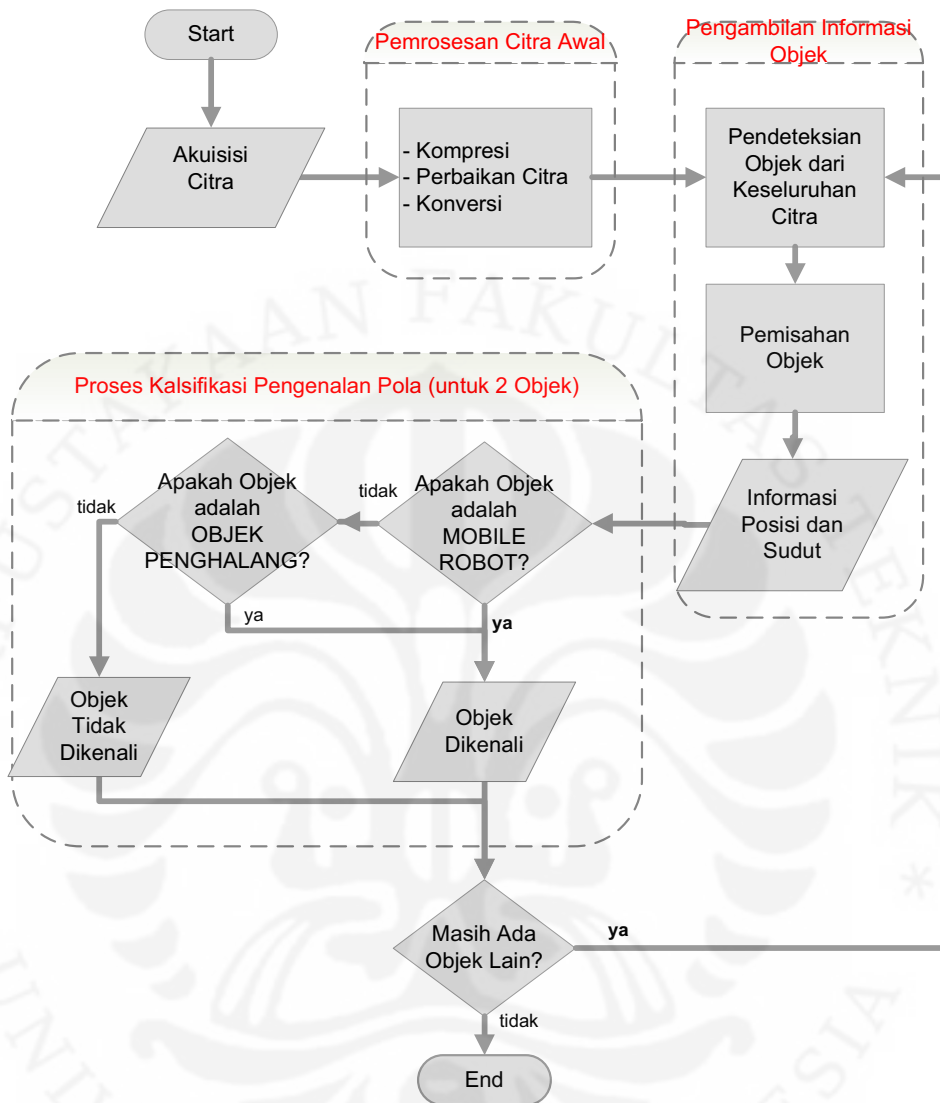
Akuisisi citra dilakukan menggunakan kamera Logitech QuickCam Pro 4000, selanjutnya citra dikirimkan ke komputer melalui komunikasi data USB. Citra yang dihasilkan dari kamera merupakan citra tiga layer RGB dengan ukuran 240 x 320 pixel.

Untuk mewujudkan sistem ini, kamera diletakkan menghadap tegak lurus permukaan bidang dengan jarak 75 cm. Setelah jarak antara kamera dengan objek, kemudian dilakukan kalibrasi ukuran antara pixel citra yang ditangkap dengan ukuran sebenarnya, 1 pixel sebanding dengan 0,27cm.



Gambar 3.10 Posisi kamera.

### 3.3.2 Flowchart Program Sensor Kamera



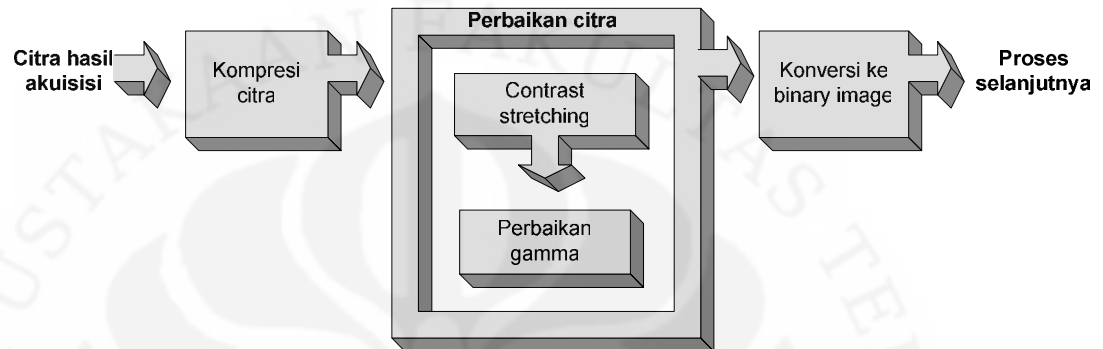
Gambar 3.11 Flowchart program.

### 3.3.3 Pemrosesan Citra Awal

Citra digital yang dihasilkan oleh kamera tidak selalu sempurna, kadang terdapat gangguan berupa *noise*, distorsi, atau hilangnya sebagian dari data. Hal ini sangat mengganggu karena citra yang tidak sempurna tidak dapat diterima oleh proses selanjutnya. Oleh sebab itu perlu dilakukan perbaikan citra terlebih dahulu.

Proses yang perlu dilakukan agar citra digital dari kamera dapat diterima oleh proses selanjutnya adalah:

1. Kompresi
2. Perbaikan Citra, dan
3. Konversi ke Citra Hitam-Putih (Citra Biner)



Gambar 3.12 Blok diagram pemrosesan citra awal.

Kompresi dilakukan untuk menurunkan ukuran citra, sehingga didapatkan ukuran citra yang lebih efisien untuk diproses. Namun agar informasi-informasi yang diinginkan dari citra tersebut tidak hilang, proses kompresi ini harus diperhatikan.

Berikutnya, dilakukan perbaikan yang dilakukan dengan melakukan *contrast stretching* dan perbaikan gamma atau pemetaan batas intensitas lama ke batas intensitas baru, sehingga didapatkan citra baru yang intensitasnya lebih baik dan sesuai untuk proses berikutnya.

Sedangkan konversi dilakukan untuk lebih menyederhanakan citra yang ada tanpa menghilangkan informasi didalamnya. Dalam hal ini, citra RGB dikonversi menjadi citra biner, dimana informasi di dalam citra hanya terdiri dua nilai saja yaitu putih dan hitam.

### 3.3.4 Pengambilan Informasi Objek

Langkah pengambilan informasi objek terbagi menjadi tiga bagian yaitu pendeteksian masing-masing objek pada citra, pemisahan masing-masing objek

dari citra, dan terakhir pencarian informasi posisi dan sudut dari masing-masing objek

Pendeteksian dan pemisahan masing-masing objek berguna untuk mendapatkan citra yang hanya berisi objek saja. Citra ini kemudian akan digunakan pada proses selanjutnya yaitu proses pengambilan informasi dan pengenalan pola objek.

Pendeteksian objek dilakukan dengan cara pengelompokan (*grouping*). Hal ini dilakukan dengan menggunakan fungsi matlab *label grouping*, dimana fungsi tersebut akan mencari kelompok nilai '1' yang berdekatan di dalam matriks citra biner yang didapat sebelumnya. Nilai '1' adalah warna putih yang merupakan objek yang kita cari. Dari fungsi ini didapat posisi dari kelompok-kelompok nilai '1' tersebut yang merupakan objek yang dicari. Setelah itu dilakukan pemisahan objek dengan cara *cropping*, sehingga didapatkan suatu citra baru yang hanya berisi objek tersebut saja.

Berikutnya dibutuhkan informasi dari objek yang sudah dilakukan pemisahan. Informasi tersebut adalah letak koordinat objek dan juga sudut kemiringan objek terhadap garis acuan.

Letak koordinat titik tengah objek didapat cukup dengan mencari titik tengah dari keseluruhan koordinat objek yang sudah didapat dari hasil pemisahan objek seperti ditunjukkan gambar 3.13.



Gambar 3.13 Koordinat objek

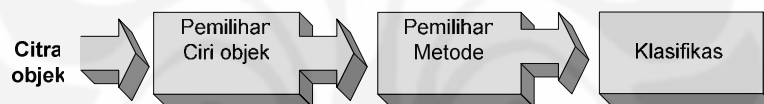
Kemudian untuk mencari sudut objek, dicari koordinat dari blok berwarna hijau. Seperti dijelaskan sebelumnya, koordinat berwarna hijau menunjukkan bagian depan dari objek. Sudut objek didapat dari perhitungan *arc tan* dari kedua koordinat tersebut seperti ditunjukkan gambar 3.14.



Gambar 3.14 Sudut objek.

### 3.3.5 Pengenalan Objek

Pengenalan objek dilakukan dengan metode pengenalan pola. Langkah-langkah dalam melakukan pengenalan objek adalah pertama-tama pemilihan ciri-ciri objek, kemudian pemilihan metode pengenalan pola, lalu membuat klasifikasi, dan terakhir mengetes hasil klasifikasi.



Gambar 3.15 Blok diagram pengenalan objek.

Pemilihan ciri-ciri ini didapatkan dari informasi tentang objek yang sudah kita ketahui sebelumnya. Ciri-ciri dari objek yang dipilih harus merepresentasikan objek tersebut, sehingga dengan ciri-ciri tersebut suatu objek bisa dibedakan dengan objek lain.

Dengan kata lain, ciri tersebut harus bersifat diskriminatif, yaitu harus mempunyai nilai yang sama (atau mirip) untuk pola yang sama, dan nilai yang berbeda untuk pola yang berbeda.

Ciri-ciri yang dipilih harus bisa tetap dikenali walaupun terjadi translasi, rotasi, dan perubahan ukuran citra. Selain itu, ciri yang ada juga harus kebal terhadap distorsi, deformasi, dan perubahan keadaan sekitar dari citra.



Karena perancangan pola sudah dilakukan sebelumnya, jadi bisa langsung didapat bahwa ciri yang dipilih untuk membedakan objek adalah kombinasi warna kuning dan merah pada blok-blok pola.

Dua metode yang bisa dipilih dalam pengenalan pola adalah metode pendekatan struktural dan statistikal. Pemilihan metode dilakukan berdasarkan struktur dari pola yang ingin dikenali. Untuk sistem ini, metode pendekatan struktural dipilih karena ada hubungan struktural yang jelas berdasarkan ciri-ciri yang didapat dari proses sebelumnya.

Klasifikasi objek merupakan tahap akhir dari pengenalan objek. Pada proses klasifikasi ini, dilakukan pengelompokan didasarkan pada ciri-ciri yang sudah didapat sebelumnya.

Klasifikasi bisa dilakukan dalam dua kondisi, yaitu *supervised* dan *unsupervised*. Pada kondisi *supervised*, sistem klasifikasi akan mengelompokkan objek-objek yang ada ke dalam kelompok yang sudah ditentukan sebelumnya. Sedangkan pada kondisi *unsupervised*, sistem akan langsung membentuk kelompok-kelompok berdasarkan ciri yang ada. Karena pada sistem ini, kelompok-kelompok objek sudah ditentukan sebelumnya, maka proses klasifikasi dilakukan pada kondisi *supervised*.

### **3.4 PERANCANGAN SISTEM PENGENDALI MOBILE ROBOT**

#### **3.4.1 Sistem Pengendalian Logika Fuzzy**

Agar sistem yang dirancang dapat berjalan sesuai dengan perancangan, maka dibutuhkan sistem pengendalian yang ada pada mikrokontroler. Seperti dijelaskan sebelumnya, mikrokontroler disini berfungsi untuk memproses informasi yang dikirimkan oleh komputer, melalui RS-232, menjadi gerakan pada *mobile robot*. Untuk dapat memproses data ini menjadi gerakan maka dibutuhkan suatu algoritma pengendalian yang berupa pengendalian fuzzy.

Dari sistem sensor kamera, didapatkan posisi dan sudut dari objek *mobile robot* dan juga objek penghalang. Selain itu, ada juga data posisi target yang ditentukan, target ini merupakan tujuan akhir dari *mobile robot*. Kemudian data-data tersebut diolah sehingga didapatkan empat parameter berikut:

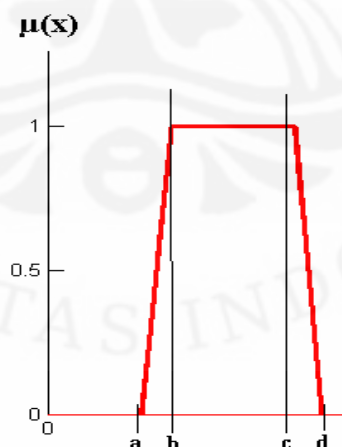
1. Jarak *mobile robot* dengan target (*jarak\_tuj*)
2. Sudut *mobile robot* dengan target (*sudut\_tuj*)
3. Jarak *mobile robot* dengan objek penghalang (*jarak\_obs*)
4. Sudut *mobile robot* dengan objek penghalang (*sudut\_obs*)

Dari keempat parameter diatas, dibuat suatu pengendalian berbasis logika fuzzy dengan dua macam sistem berbeda. Sistem pertama adalah sistem pengendalian tanpa objek penghalang dan sistem kedua merupakan sistem pengendalian dengan objek penghalang.

### 3.4.1.1 Sistem Pengendalian Tanpa Objek Penghalang

Untuk sistem pengendalian tanpa objek penghalang, parameter yang digunakan hanya parameter *jarak\_tuj* dan *sudut\_tuj*. *Jarak\_tuj* menunjukkan jarak yang masih harus ditempuh *mobile robot* untuk sampai ke tujuan akhir. Sedangkan *sudut\_tuj* menunjukkan sudut *mobile robot* saat itu terhadap titik tujuan.

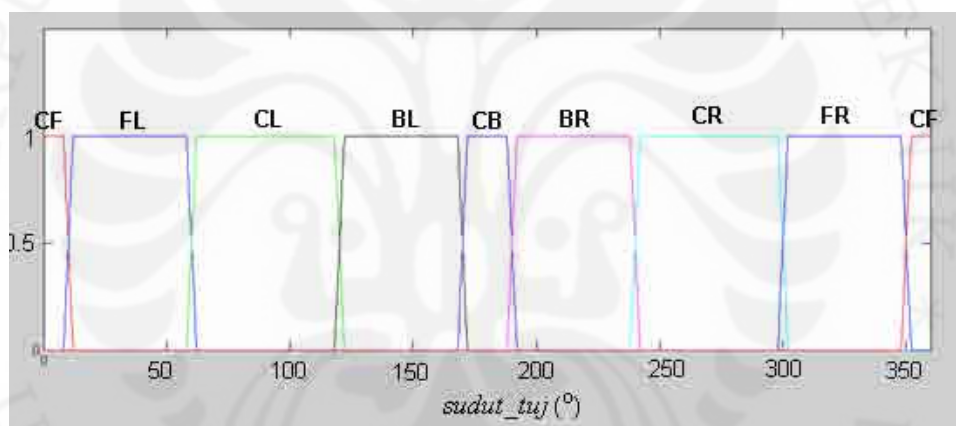
Untuk melakukan pengendalian [6], nilai kedua parameter yang kontinu tersebut dikonversi terlebih dahulu ke suatu set nilai fuzzy. Set nilai fuzzy yang digunakan adalah bentuk trapezoid seperti pada gambar 3.16, dan ditunjukkan dengan persamaan 3.2.



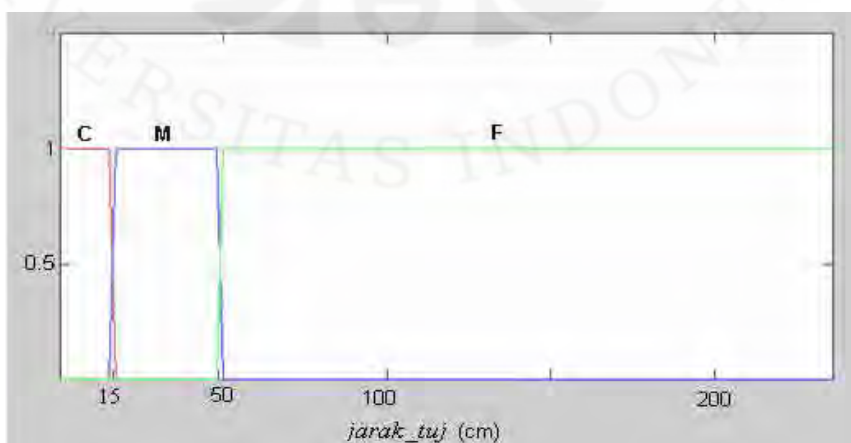
Gambar 3.16 Fungsi keanggotaan berbentuk trapezoid

$$\mu(x) = \begin{cases} 0 & , x \leq a \\ \frac{x-a}{b-a} & , a \leq x \leq b \\ 1 & , b \leq x \leq c \\ \frac{d-x}{d-c} & , c \leq x \leq d \\ 0 & , x \geq d \end{cases} \quad (3.2)$$

Proses konversi dari nilai kontinu ke nilai fuzzy ini disebut juga dengan fuzzifikasi. Fuzzifikasi dilakukan sesuai dengan grafik fungsi keanggotaan yang ditunjukkan pada gambar 3.17 dan 3.18.

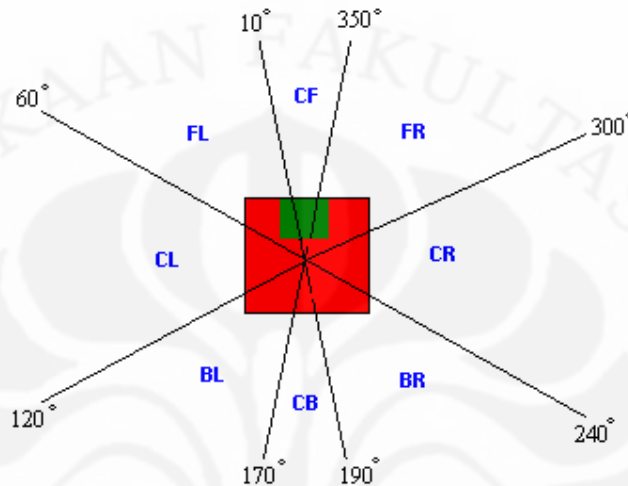


Gambar 3.17 Fungsi keanggotaan untuk *sudut\_tuj*



Gambar 3.18 Fungsi keanggotaan untuk *jarak\_tuj*

Nilai fuzzy dari *jarak\_tuj* terbagi menjadi tiga yaitu *F* (*far*), *M* (*medium*), dan *C* (*close*). Sedangkan nilai fuzzy dari *sudut\_tuj* terbagi menjadi delapan yaitu *CF* (*center-front*), *CB* (*center-back*), *CR* (*center-right*), *CL* (*center-left*), *FR* (*front-right*), *FL* (*front-left*), *BR* (*back-right*), dan *BL* (*back-left*).

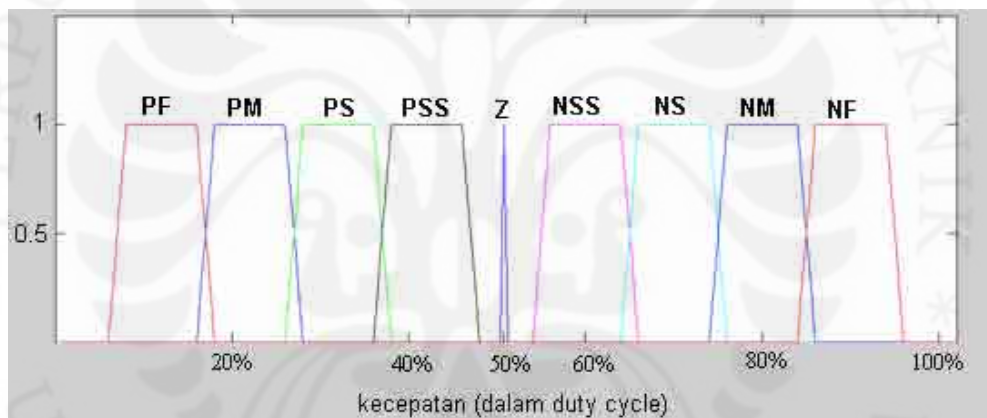


Gambar 3.19 Pembagian *sudut\_tuj*.

Kemudian dari kedua kondisi fuzzy dari *jarak\_tuj* dan *sudut\_tuj* yang didapatkan, diterapkan aturan-aturan fuzzy untuk mendapatkan nilai *duty cycle* yang harus diaplikasikan ke sistem kendali motor untuk menggerakkan *mobile robot* [7]. Aturan fuzzy yang diterapkan pada pengendalian sistem ini ditunjukkan pada tabel 3.3. Dari aturan fuzzy tersebut, terdapat sepuluh nilai kecepatan yang mungkin didapatkan yaitu *PF* (*positive-fast*), *PM* (*positive-medium*), *PS* (*positive-slow*), *PSS* (*positive-really-slow*), *Z* (*zero*), *NSS* (*negative-really-slow*), *NS* (*negative-slow*), *NM* (*negative-medium*), dan *NF* (*negative-fast*). Fungsi keanggotaan dari nilai fuzzy kecepatan ini ditunjukkan pada gambar 3.20.

Tabel 3.1 Aturan fuzzy untuk kecepatan

	Jarak_tuj= FAR		Jarak_tuj= CLOSE		Jarak_tuj=VERYCLOSE	
	kiri	kanan	kiri	kanan	kiri	kanan
<i>Sudut_tuj</i> = CF	+F	+F	+M	+M	+S	+S
<i>Sudut_tuj</i> = CB	-F	-F	-M	-M	-S	-S
<i>Sudut_tuj</i> = CR	+F	-RS	+M	-RS	+S	-RS
<i>Sudut_tuj</i> = CL	-RS	+F	-RS	+M	-RS	+S
<i>Sudut_tuj</i> = FR	+F	+S	+M	+RS	+S	Z
<i>Sudut_tuj</i> = FL	+S	+F	+RS	+M	Z	+S
<i>Sudut_tuj</i> = BR	-S	-F	-RS	-M	Z	-S
<i>Sudut_tuj</i> = BL	-F	-S	-M	-RS	-S	Z



Gambar 3.20 Fungsi keanggotaan untuk kecepatan

Langkah terakhir dari pengendalian sistem ini adalah proses defuzzifikasi pada nilai fuzzy kecepatan yang didapatkan dari aturan fuzzy. Proses defuzzifikasi ini dilakukan dengan metode centroid yang dilakukan dengan persamaan 3.3 [4].

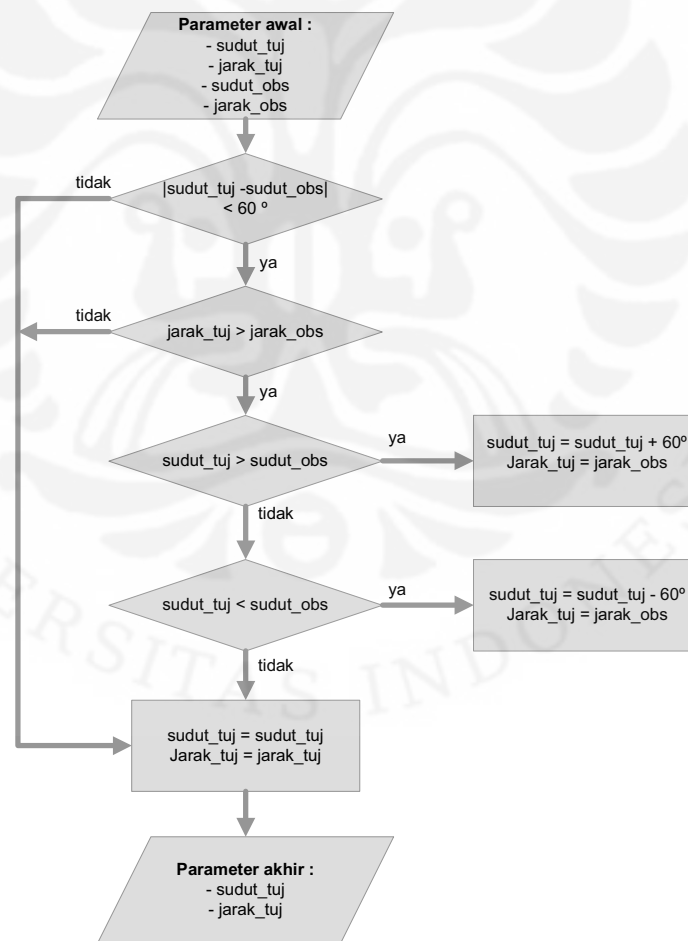
$$Z = \frac{\int \mu_{kecepatan}(z) \cdot z \cdot dz}{\int \mu_{kecepatan}(z)} \tag{3.3}$$

Hasil akhir dari proses defuzzifikasi adalah nilai kecepatan kontinu berupa nilai *duty cycle* dari PWM.

Pergerakan *mobile robot* akan dianggap selesai jika nilai *jarak\_tuj* atau nilai jarak antara *mobile robot* dan titik tujuan kurang dari 3% dari panjang bidang. Dalam perancangan ini, 3% dari panjang bidang adalah bernilai 2,61 cm.

### 3.4.1.2 Sistem Pengendalian Dengan Objek Penghalang

Untuk sistem dengan objek penghalang, digunakan keempat parameter yang ada. Sehingga bisa dilakukan pengendalian untuk menghindari objek penghalang dengan algoritma yang digambarkan flowchart pada gambar 3.21.



Gambar 3.21 Flowchart pengendalian dengan penghalang.

Setelah algoritma diatas dijalankan didapatkan parameter akhir yaitu parameter *jarak\_tuj* dan *sudut\_tuj* baru. Dari kedua parameter ini, selanjutnya dilakukan fuzzifikasi, penerapan aturan fuzzy, dan defuzzifikasi yang sama seperti pada sistem pengendalian tanpa objek penghalang.



## BAB IV

### PENGUJIAN DAN ANALISA

#### 4.1 PENGUJIAN KOMUNIKASI SERIAL

Pengujian dilakukan dengan menguji terlebih dahulu rangkaian serial yang berfungsi sebagai alat konversi komunikasi antara komputer dengan mikrokontroler. Program yang digunakan adalah program terminal pada CodeVisionAVR dengan *baud rate* 9600.

Dengan program *komunikasi.cpp* di dalam mikrokontroler, maka jika diketikkan suatu karakter pada terminal, karakter tersebut akan dikirimkan ke mikrokontroler. Kemudian mikrokontroler akan mengirim kembali karakter tersebut, sehingga karakter tersebut muncul pada layar terminal.



Gambar 4.1 Tampilan teminal CodeVisionAVR.

Window output menampilkan karakter 'a' yang dikirimkan sebelumnya. Ini berarti telah terjadi komunikasi serial antara komputer dengan mikrokontroler.

Untuk aplikasinya pada *mobile robot*, pada sistem ini digunakan komunikasi serial menggunakan kabel. Untuk mengembangkan sistem komunikasi serial tersebut, dapat diubah penggunaan kabel dengan sistem nirkabel yang akan memudahkan *mobile robot* untuk bergerak lebih leluasa.



## 4.2 PENGUJIAN SISTEM KENDALI MOTOR

Pengujian dilakukan dengan memberikan input ke rangkaian pengendali motor DC. Input berupa sinyal PWM pada port C (PC.0, PC.2, PC.4, PC.6) mikrokontroler. Output pengendali motor dihubungkan ke motor DC 1 dan motor DC 2. Sedangkan pengukuran kecepatan dilakukan manual dengan perhitungan jarak tempuh dibagi waktu tempuh.

Tabel 4.1. Hasil pengujian rangkaian pengendali motor.

Duty Cycle PWM				Putaran Motor		Kecepatan Motor	
PC.0	PC.2	PC.4	PC.6	1	2	1	2
0%	100%	100%	0%	cw	ccw	20.5 cm/s	20.2 cm/s
11,76%	88,24%	88,24%	11,76%	cw	ccw	12,7 cm/s	12,5 cm/s
21,57%	78,43%	78,43%	21,57%	cw	ccw	7,5 cm/s	7,5 cm/s
31,37%	68,63%	68,63%	31,37%	cw	ccw	4,5 cm/s	4,5 cm/s
50%	50%	50%	50%	Stop	Stop	-	-

Agar pergerakan roda penggerak lebih stabil, motor DC yang digunakan dapat ditambahkan *controller*.

## 4.2 PENGUJIAN SISTEM SENSOR KAMERA

Pengujian dilakukan dengan menjalankan sistem sensor kamera. Hasil keluaran yang diharapkan adalah identitas objek, juga nilai posisi dan sudut dari masing-masing objek.



Gambar 4.2 Pengujian sistem sensor kamera.

Pengujian dilakukan sebanyak enam kali, dengan dua macam objek yang diletakkan bervariasi pada kondisi pencahayaan yang berbeda-beda. Dua kali pengujian untuk masing-masing kondisi pencahayaan. Dari pengujian tersebut diambil data yang ditunjukkan pada tabel 4.2, keberhasilan pengenalan identitas objek ditentukan oleh hasil identifikasi untuk masing-masing objek. Sedangkan parameter keberhasilan untuk informasi sudut dan posisi adalah didapatkannya nilai sudut dan posisi.

Tabel 4.2. Hasil pengujian sistem sensor kamera.

Waktu	Informasi Sudut & Posisi	Pengenalan Identitas Objek	Keadaan
0.4219 s	Berhasil	Berhasil	Cahaya = terang Bayangan = sedikit
0.4063 s	Berhasil	Berhasil	
0.3906 s	-	-	Cahaya = terang Bayangan = banyak
0.4375 s	-	Berhasil	
0.3750 s	-	-	Cahaya = redup
0.4375 s	-	-	

## 4.3 PENGUJIAN ALGORITMA PENGENDALI

### 4.3.1. Pengujian Tanpa Objek Penghalang

Pengujian dilakukan dengan menjalankan sistem pada kondisi objek *mobile robot* tanpa penghalang. Percobaan dilakukan sebanyak lima kali dengan koordinat *mobile robot* dan koordinat titik tujuan yang berbeda-beda.



Gambar 4.3 Keadaan pengujian tanpa objek penghalang.

Data yang diambil adalah posisi pada keadaan awal dan akhir, sudut antara *mobile robot* dengan titik tujuan (*sudut\_tuj*), dan juga jarak akhir antara *mobile robot* dengan titik tujuan (*jarak\_tuj*).

Tabel 4.3. Hasil pengujian tanpa objek penghalang.

tujuan		Awal			Akhir			jarak_tuj
X	Y	X mr	Y mr	sudut_tuj	X mr	Y mr	sudut_tuj	akhir
8	8	73.6374	38.5788	42.9101	9.641	10.9232	338.0473	1.9865
8	8	72.9268	48.2559	187.6355	11.773	7.3451	176.9031	2.2693
8	8	75.9826	46.8916	79.141	10.5766	9.411	0.0051	1.7408
60	8	17.1573	35.4191	221.2961	60.783	10.4531	191.3882	1.5259
60	8	53.0381	36.2751	295.5216	58.6784	10.1469	330.2549	1.494

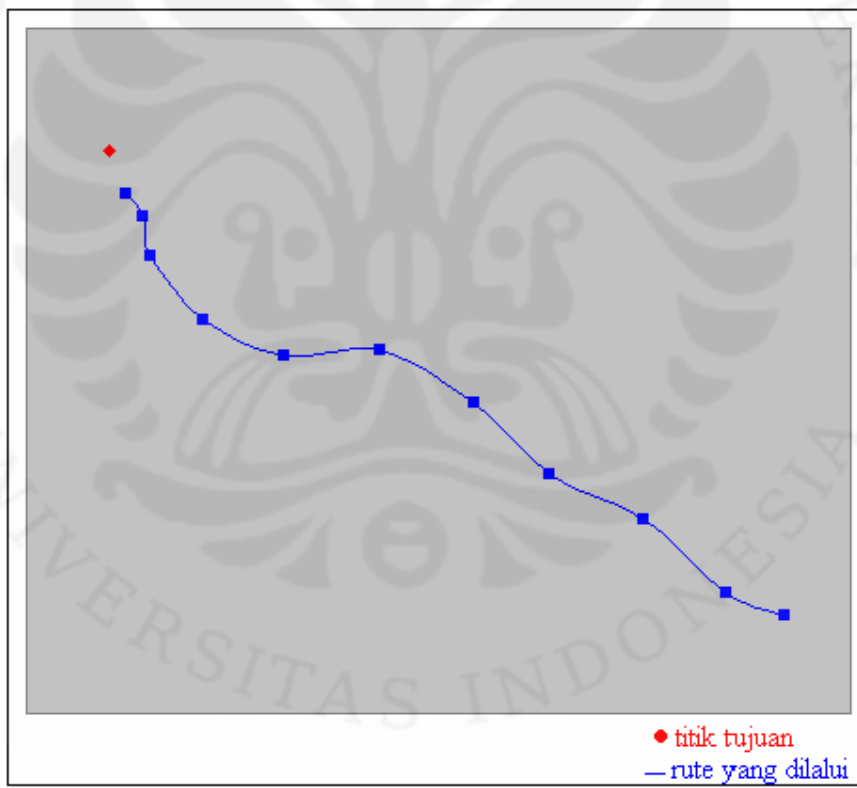
Dari data yang didapat dari pengujian, terlihat bahwa nilai *jarak\_tuj* akhir kurang dari 2,3 cm. Secara umum, hal ini sudah menunjukkan bahwa pengendalian yang dilakukan sudah cukup baik.

Sistem yang dijalankan masih menunggu lebih dari 3 s untuk setiap *loop*-nya. Hal ini dikarenakan sistem harus menunggu akuisisi citra dan jalannya program. Jadi, algoritma pemrograman yang lebih singkat, dapat membuat proses kendali dapat dilakukan secara kontinu.

Untuk melihat sistem secara lebih detil, dilakukan pengamatan terhadap pergerakan yang dilakukan oleh sistem pada masing-masing *loop*. Maka dilakukan tiga kali pengujian yang dapat mewakili pengujian sistem keseluruhan.

Tabel 4.4. Pengujian sistem tanpa penghalang dengan *sudut\_tuj* awal 295.43.

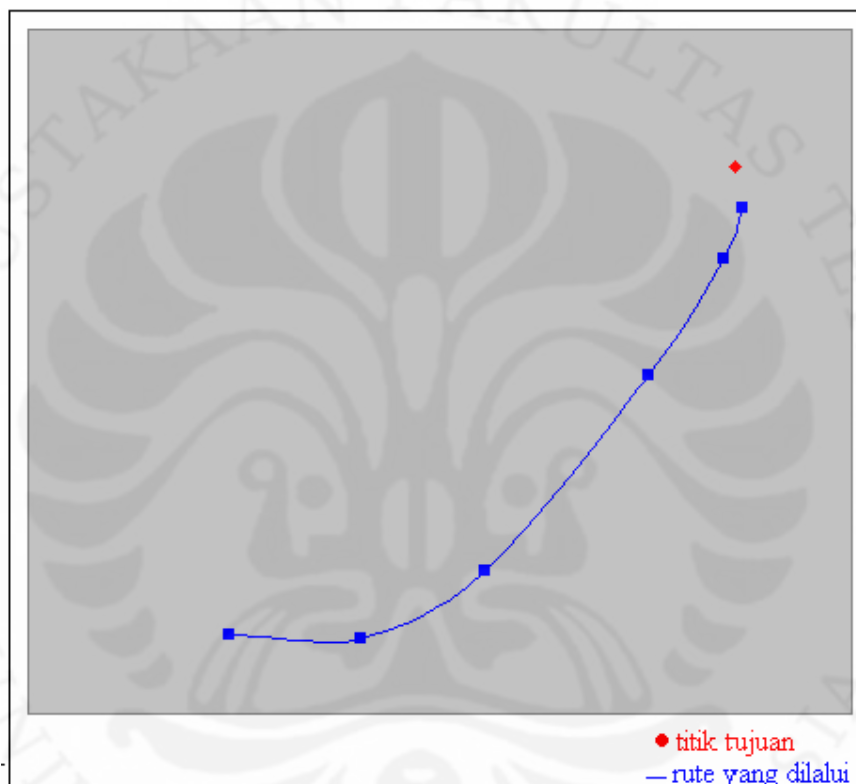
X tujuan	Y tujuan	X mr	Y mr	jarak_real	sudut_tuj
8	8	73.6374	38.5788	42.9101	295.4328
8	8	67.9068	37.0864	39.4635	345.2947
8	8	59.8453	32.2742	33.9239	11.8067
8	8	50.7291	29.3388	28.3028	347.764
8	8	43.4352	24.5548	23.1772	14.3031
8	8	34.408	21.1923	17.4932	351.4499
8	8	25.0781	21.4657	12.8878	317.6813
8	8	17.2766	19.2221	8.6281	329.233
8	8	12.1661	14.9807	4.8174	344.3554
8	8	11.4172	12.4022	3.3024	10.1743
8	8	9.641	10.9232	1.9865	338.0473



Gambar 4.4. Pengujian sistem tanpa penghalang dengan *sudut\_tuj* awal 295.43.

Tabel 4.5. Pengujian sistem tanpa penghalang dengan *sudut\_tuj* awal 221.29.

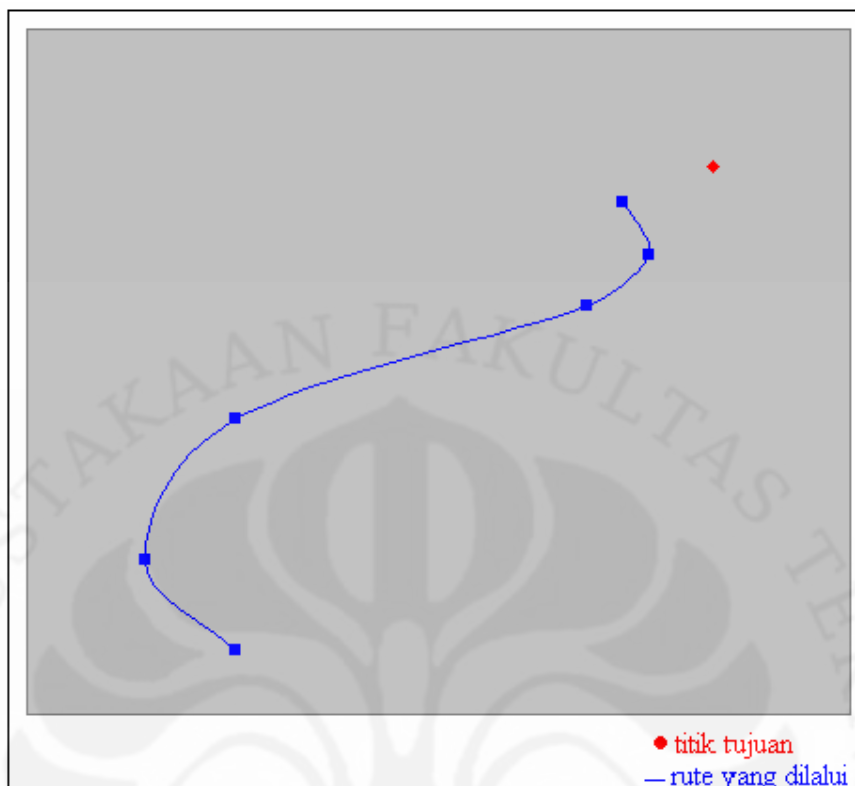
<b>X tujuan</b>	<b>Y tujuan</b>	<b>X mr</b>	<b>Y mr</b>	<b>jarak_tuj</b>	<b>sudut_tuj</b>
60	8	17.1573	35.4191	30.1425	221.2961
60	8	28.4107	35.6354	24.872	205.7472
60	8	38.7838	31.657	18.8309	187.5666
60	8	52.7721	20.2081	8.4073	198.4602
60	8	59.1814	13.4144	3.245	193.518
60	8	60.783	10.4531	1.5259	191.3882



Gambar 4.5 Pengujian sistem tanpa penghalang dengan *sudut\_tuj* awal 221.29.

Tabel 4.6. Pengujian sistem tanpa penghalang dengan *sudut\_tuj* awal 295.52.

<b>X tujuan</b>	<b>Y tujuan</b>	<b>X mr</b>	<b>Y mr</b>	<b>jarak_tuj</b>	<b>sudut_tuj</b>
60	8	53.0381	36.2751	17.256	295.5216
60	8	51.7392	30.9925	14.4779	335.0565
60	8	53.0378	22.7949	9.6896	356.0053
60	8	58.1651	16.1682	4.961	34.5348
60	8	59.0555	13.1696	3.1142	14.7019
60	8	58.6784	10.1469	1.494	330.2549



Gambar 4.6. Pengujian sistem tanpa penghalang dengan *sudut\_tuj* awal 295.52.

Dari ketiga kali pengujian, terlihat bentuk rute yang berbeda-beda. Pada pengujian pertama, jarak awal antara *mobile robot* dengan titik tujuan cukup jauh. Hal ini menyebabkan pada awal, pergerakan yang dilakukan cukup signifikan. Namun kemudian ketika *mobile robot* mulai mendekati ke tujuan, pergerakannya lebih sedikit. Lain halnya dengan pengujian kedua dan ketiga, dimana jarak awal antara *mobile robot* dengan titik tujuan lebih dekat.

Untuk meningkatkan keakuratan gerakan robot ke titik tujuan, variabel fuzzy pada pengendali dapat disempurnakan dengan menambahkan variabel-variabel baru diantara variabel yang ada.

#### 4.3.2. Pengujian Dengan Objek Penghalang

Pengujian dilakukan dengan menjalankan sistem pada kondisi objek *mobile robot* dengan penghalang. Jadi objek yang ditangkap oleh sensor ada dua yaitu objek *mobile robot* dan objek penghalang.

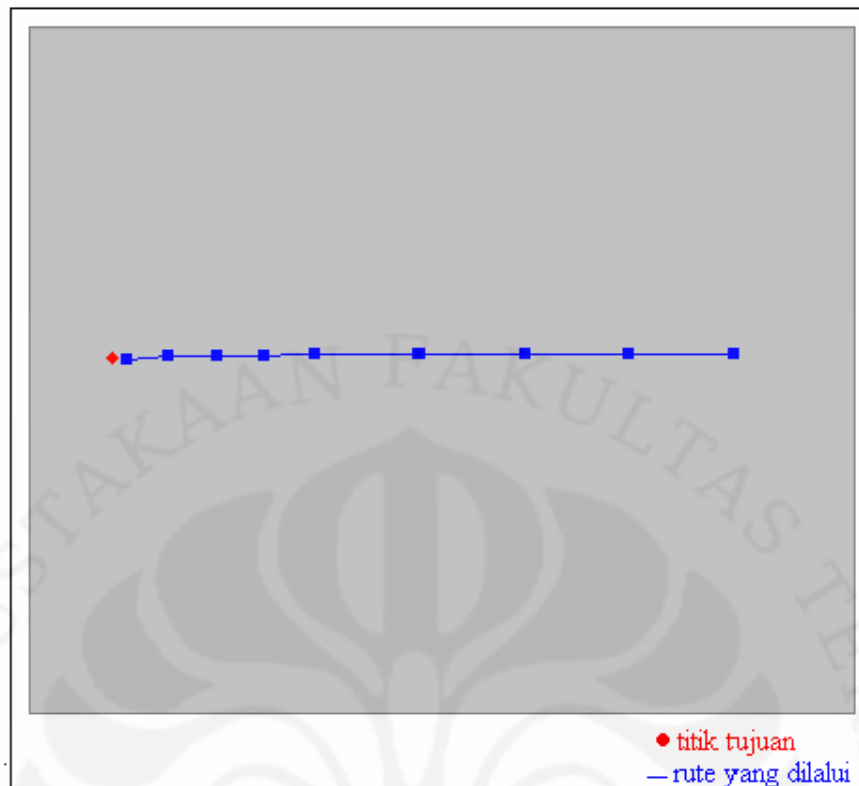


Gambar 4.7 Keadaan pengujian dengan objek penghalang.

Untuk menguji algoritma sistem ini diambil dua data pengujian. Kedua pengujian dilakukan dengan kondisi awal *mobile robot* dan titik tujuan sama, namun pengujian pertama dilakukan tanpa halangan dan pengujian kedua dilakukan dengan halangan.

Tabel 4.7. Pengujian tanpa objek penghalang

<b>X tujuan</b>	<b>Y tujuan</b>	<b>X mr</b>	<b>Y Mr</b>	<b>jarak_tuj</b>	<b>sudut_tuj</b>
8	31	68.3201	31.3188	35.7457	2.7912
8	31	58.1196	31.4136	29.7015	3.1586
8	31	48.0121	31.435	23.7123	2.8108
8	31	37.8239	31.4057	17.675	3.6508
8	31	27.6876	31.4406	11.6696	2.7852
8	31	22.787	31.3144	8.7647	4.6708
8	31	18.3522	31.2307	6.1362	4.1095
8	31	13.5284	31.2936	3.2807	2.0796
8	31	9.4427	30.9615	0.8552	3.8115

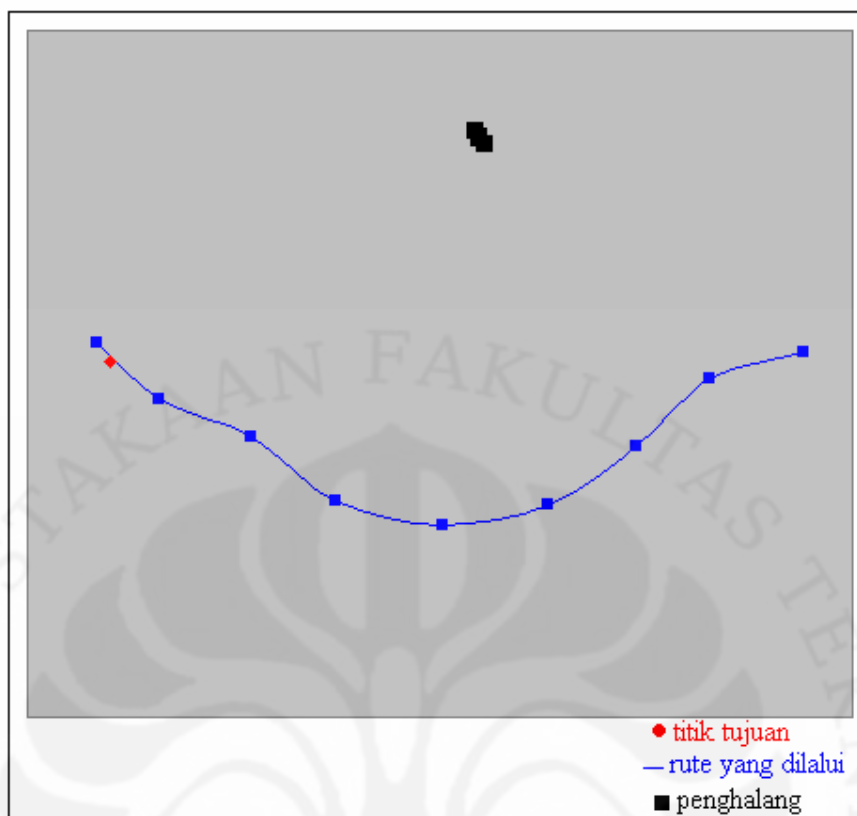


Gambar 4.8 Pengujian tanpa objek penghalang.

Tabel 4.8. Pengujian dengan objek penghalang

<i>X tujuan</i>	<i>Y tujuan</i>	<i>X pengh</i>	<i>Y pengh</i>	<i>X mr</i>	<i>Y mr</i>	<i>jarak_tuj</i>	<i>sudut_tuj</i>
8	31	43.7334	50.6668	75.2279	31.8842	21.7303	318.4255
8	31	44.2862	50.2075	66.1458	29.5751	17.8127	349.0722
8	31	43.7087	50.6839	59.0684	23.6806	18.4095	22.3524
8	31	43.7066	50.7158	50.3923	18.5421	26.1836	46.7281
8	31	43.7139	50.7169	40.156	16.7664	20.8387	31.8469
8	31	43.6978	50.7108	29.9832	18.8516	14.8839	14.7399
8	31	43.5935	50.8312	21.6471	24.4292	8.9757	349.5017
8	31	43.3518	51.2891	12.7047	27.8555	3.3534	23.3274
8	31	43.3148	51.3609	6.7946	32.7101	2.092	2.5165





Gambar 4.9 Pengujian dengan objek penghalang.

Pada pengujian dengan halangan, pergerakan *mobile robot* ke titik tujuan tidak dilakukan dengan garis lurus. Pergerakan *mobile robot* ini mengambil rute lebih ke kiri dari seharusnya karena adanya pengaruh dari objek penghalang yang ada. Hasil akhir dari kedua sistem yang diujikan tidak jauh berbeda, dengan jarak ke tujuan pada keadaan akhir kurang dari 3 cm.

Secara umum, dilakukan juga lima kali pengujian terhadap sistem dengan objek penghalang. Dari lima pengujian tersebut, dihasilkan diantaranya dua yang menghasilkan sistem error. Sistem error ini terjadi dikarenakan *mobile robot* keluar dari area yang ditangkap sensor kamera karena algoritma sistem dengan halangan ini tidak mempertimbangkan ujung-ujung area yang ditangkap sensor.

Jadi, algoritma sistem penghalang bisa dikembangkan lebih baik lagi dengan mempertimbangkan ujung-ujung area yang ditangkap oleh sensor kamera.

## BAB V

### KESIMPULAN

1. Sistem navigasi yang dirancang pada *mobile robot* dalam skripsi ini dengan menggunakan sensor kamera sudah dapat bekerja dengan baik.
2. Sistem pengendalian fuzzy secara umum sudah dapat diaplikasikan, baik untuk sistem tanpa penghalang maupun sistem dengan penghalang.
3. Algoritma sistem dengan penghalang masih kurang baik karena tidak mempertimbangkan ujung-ujung area yang ditangkap sensor kamera.
4. Sistem sensor kamera yang digunakan sebagai umpan balik membutuhkan waktu cukup lama (sekitar 0,4 s). Ini menyebabkan, sensor tidak dapat mengimbangi pergerakan robot, sehingga proses kendali harus dilakukan secara bertahap (tidak kontinu).
5. Dua motor DC yang digunakan, dengan input *duty cycle* yang sama memiliki perbedaan kecepatan yang kecil dan tidak signifikan dalam pergerakan *mobile robot*.

## DAFTAR ACUAN

- [1] \_\_\_\_\_, “*Definition of Robot.*” Diakses 19 Maret 2008, dari Dictionary.  
<http://dictionary.reference.com/browse/robot>
- [2] \_\_\_\_\_, “Atmega8535 / Atmega8535L,” Datasheet, ATMEL Corporation, Juli 2006.
- [3] \_\_\_\_\_, “*DC Motor Operation.*” Diakses 22 Februari 2008, dari Hyperphysics.  
<http://hyperphysics.phy-astr.gsu.edu/hbase/magnetic/motdc.html>
- [4] J. Ross, Timothy, *Fuzzy Logic With Engineering Applications*, (USA: McGraw-Hill Book Company, 1995), hal 136.
- [5] Kusdarwanto, Hendra, “*Pengendalian Motor DC Menggunakan Fuzzy Logic Berbasis Mikrokontroler*”
- [6] Marhaban, *et al*, “*Real Time Navigation of an Autonomous Robot using Fuzzy Controller*”, 2007
- [7] F. Cupertino, *et al*, “*Fuzzy Control of Mobile Robot*” *Jurnal IEEE*, 2006.

## DAFTAR PUSTAKA

- \_\_\_\_\_, “*AVR Microcontroller*,” Application Note, ATMEL Corporation, Februari 2005.
- R. Siegwart, I. Nourbakhsh, *Introduction to Autonomous Mobile Robot*, (MIT Press, 2004)
- Bose, Tamal, *Digital Signal and Image Processing*, (New Jersey: Wiley, 2003)
- Bing, Hao, “*H-Bridge Solution*”. Diakses 9 Mei 2008, dari Urobotbots.  
<http://urrobotbots.googlepages.com/h-bridgesolution>
- F. Pezeshkpour, “*VFM for MATLAB*,” Release Note, 1998. Diakses 22 Februari 2008, dari MATLAB Central File.  
<http://www.sys.uea.ac.uk/~fuzz/vfm/default.html>
- \_\_\_\_\_, “*Scientific / Educational MATLAB Database*.” Diakses 22 Februari 2008, dari University of Stuttgart.  
<http://matlabdb.mathematik.uni-stuttgart.de/index.jsp>

## LAMPIRAN



## Lampiran 1 Listing Program *main\_program.m*

```
s = serial('COM1');
set(s, 'BaudRate', 9600);
fopen(s);

%kalibrasi
%ukuran sebenarnya 1 pixel=n cm
perb=0.27
koorx_real=koorx*perb;
koory_real=koory*perb;

%tentukan tujuan (real)
tujuan_real=[31 8];
tujuan=tujuan_real/perb;

jarak=1;
sdt_tuj=0;
data_jarak=[1];
data_sdt_tuj=[1];
sbl=2;
while jarak>0 && sdt_tuj>=-1;
t=cputime;
pengenalan_cam;
data_jarak=[data_jarak; jarak_real]
sel_sbl=abs(data_jarak(sbl)-data_jarak(sbl-1));
data_sdt_tuj=[data_sdt_tuj; sdt_tuj]
sdt_sbl=abs(data_sdt_tuj(sbl)-data_sdt_tuj(sbl-1));
if sel_sbl<0.1 & sdt_sbl<2
    jarak=0;
    sdt_tuj=-2;
end
if jarak_real<(perb*320*0.03)
    jarak=0;
    sdt_tuj=-2;
end
sbl=sbl+1;
%fuzz;
kendali_fuzzy;

%kirimdata;
fwrite(s,xx_kiri)
fwrite(s,xx_kanan)

e=cputime-t
%delay sampe 2 sekon
while (cputime-t)<2
end

end

fclose(s);
```

## Lampiran 2 Listing Program pengenalan\_cam.m

```

vfm('show',0);
aa=vfm('grab');
aa=imresize(aa,[240 320]);
[baris,kolom,aaa]=size(aa);
imgCap=imadjust(aa,[0 0.9],[0 1]);
img_L=imgCap(:,:,3);
baris_n=baris; kolom_m=kolom;
%berdasarkan intensitas cahaya
ruangan
b_bwh=10; b_ats=150;

filter_binaryimage;
mr=0;
tuj=0;
group=bwlabel(pixImgBW,4);
nilaimaxgroup=max(max(group));
koorx=[];koory=[];
maxmin=[];
heiii=[];widddd=[];
byk_group=0;
for nilaimax=1:nilaimaxgroup;
[r,c]= find(group == nilaimax);
sizer=length(r);
hei=max(r)-min(r);
wid=max(c)-min(c);
%luas min dan maks area
for xx=1:l:sizer
nilai_image=imgCap(r(xx),c(xx),1);
end
if (hei <= 150)&(wid <= 150)&(hei
> 50)&(wid > 50)&(sizer >=
500)&(sizer <= 8000) &
max(nilai_image)>100
%max(nilai_image)
koorx_s=sum(c)/length(c);
koory_s=sum(r)/length(r);
groupa=
group(min(r):max(r),min(c):max(c));
heiii=[heiii;hei];
widddd=[widddd;wid];
maxmin=[maxmin; min(r) max(r)
min(c) max(c)];
% Mencari letak
koorx=[koorx;koorx_s];
koory=[koory;koory_s];
byk_group=byk_group+1;
else
groupa=[];
end
end
sudut=[];
objek_id=[];
objekdoang=[];
[ro kol]=size(maxmin);
for l=1:ro
objekdoang=imgCap(maxmin(l,1):maxmin(
l,2),maxmin(l,3):maxmin(l,4),:);
img_L=objekdoang(:,:,1);
baris_n=heiii(l);
kolom_m=widddd(l);
b_bwh=0; b_ats=100;
filter_binaryimage;
sdt=[];
group2=bwlabel(pixImgBW,4);
nilaimaxgroup2=max(max(group2));
for nilaimax2=1:nilaimaxgroup2;
[r2,c2]= find(group2 ==
nilaimax2);
%luas min dan maks area
sizer2=length(r2);
hei2=max(r2)-min(r2);
wid2=max(c2)-min(c2);
if (hei2 <= 150)&(wid2 <=
150)&(sizer2 >= 50)&(sizer2 <= 8000)
groupa2=group2(min(r2):max(r2),min(c2)
):max(c2));
%imshow(groupa2)
% Mencari letak
koorx2=maxmin(l,3)+(sum(c2)/length(c2)
));
koory2=maxmin(l,1)+(sum(r2)/length(r2)
));
else
groupa2=[];
end
end
radian=atan2((koory(1)-
koory2),(koorx(1)-koorx2));
sdt=(radian*180)/(pi);
sdt=90-sdt;
if sdt<0
sdt=360+sdt;
end
sudut=[sudut;sdt];
putar=imrotate(objekdoang,-
sdt,'crop');
img_L=putar(:,:,3);
b_bwh=10; b_ats=200;
[baris_n,kolom_m,aaa]=size(putar);
filter_binaryimage;
group3=bwlabel(pixImgBW,4);
nilaimaxgroup3=max(max(group3));
for nilaimax3=1:nilaimaxgroup3;
[r3,c3]= find(group3 ==
nilaimax3);
%luas min dan maks area
sizer3=length(r3);
hei3=max(r3)-min(r3);
wid3=max(c3)-min(c3);
if (hei3 <= 150)&(wid3 <=
150)&(sizer3 >= 500)&(sizer3 <= 8000)
objekputar=
putar(min(r3):max(r3),min(c3):max(c3)
,:);
%figure;
%imshow(objekputar);
h=1; w=1;
[hhh www
ttd]=size(objekputar);
hh=double(int8((hhh-
1)/3));
ww=double(int8((www-
1)/3));
blok=[];
warnaa=[];
for n=1:3
for m=1:3

```

```

blok=objekputar(h:(h+hh-1),w:(w+ww-1),:);
cent=blok(int8(hh/2),int8(ww/2),2);
if cent>150
    kuning
else
    merah
end
warna=[warna;warna];
w=w+ww;
end
w=1;
h=h+hh;
end
else
    groupa=[];
end
end
cek_warna;
end
for v=1:size(objek_id)
    objek_identitas=objek_id(v)
    koordinat_x=koorx_real(v)
    koordinat_y=koory_real(v)
    sudut_objek=sudut(v)
    if objek_id(v)=='1'
        mr=v;
    end
    if objek_id(v)=='2'
        tuju=v;
    end
end
if ro==1
    radian_tuj=atan2((koory(mr)-tujuan(1)),(koorx(mr)-tujuan(2)));
    sdt_tuj=(radian_tuj*180)/(pi);
    sdt_tuj=90-sdt_tuj;
    if sdt_tuj<0
        sdt_tuj=360+sdt_tuj;
    end
    sdt_tuj=sdt_tuj-sudut(mr);
    if sdt_tuj<0
        sdt_tuj=360+sdt_tuj;
    end
    sdt_tuj
    jarak=((koory(mr)-tujuan(1))^2+((koorx(mr)-tujuan(2))^2)^0.5;
    jarak_real=jarak*0.16
elseif ro==2
    objek_y=tujuan(1);
    objek_x=tujuan(2);
    hitung;
    sudut_t=sdt_n
    jarak_t=jarak_n
    objek_y=koory(obs);
    objek_x=koorx(obs);
    hitung;
    sudut_o=sdt_n
    jarak_o=jarak_n
    if abs(sudut_t-sudut_o)<60 ||
    abs(sudut_t-sudut_o)>300
        if jarak_o<=jarak_t
            if (sudut_t-
            sudut_o)>0
                banding=[abs(sudut_t-
                sudut_o);abs(sudut_t+sudut_o-360)];
            [minn,min_banding]=min(banding
            );
            elseif
            (sudut_t-sudut_o)<0
                banding=[abs(sudut_t+sudut_o-
                360);abs(sudut_o-sudut_t)];
            [minn,min_banding]=min(banding
            );
            end
            if
            min_banding==1
                sdt_tuj=sudut_t+40;
            elseif
            min_banding==2
                sdt_tuj=sudut_t-40;
            end
            if sdt_tuj>=360
                sdt_tuj=sdt_tuj-360;
            end
            if sdt_tuj<0
                sdt_tuj=360+sdt_tuj;
            end
            jarak_real=jarak_o;
            else sdt_tuj=sudut_t;
            jarak_real=jarak_t;
            end
            else sdt_tuj=sudut_t;
            jarak_real=jarak_t;
            end
            end
end
end

```



### Lampiran 3 Listing Program kendali\_fuzzy.m

```

%FUZZIFICATION
x_jarak=0:1:320;
jarak_near=trapmf(x_jarak,[0 0 15
17]);
jarak_medium=trapmf(x_jarak,[15 17 48
50]);
jarak_far=trapmf(x_jarak,[48 50 320
320]);

x_sdt=0:1:360;
sdt_CF1=trapmf(x_sdt,[0 0 8 12]);
sdt_FL=trapmf(x_sdt,[8 12 58 62]);
sdt_CL=trapmf(x_sdt,[58 62 118 122]);
sdt_BL=trapmf(x_sdt,[118 122 168
172]);
sdt_CB=trapmf(x_sdt,[168 172 188
192]);
sdt_BR=trapmf(x_sdt,[188 192 238
242]);
sdt_CR=trapmf(x_sdt,[238 242 298
302]);
sdt_FR=trapmf(x_sdt,[298 302 348
352]);
sdt_CF2=trapmf(x_sdt,[348 352 360
360]);

x_speed=0:1:255;
speed_PF=trapmf(x_speed,[15 20 40
45]);
speed_PM=trapmf(x_speed,[40 45 65
70]);
speed_PS=trapmf(x_speed,[65 70 90
95]);
speed_PSS=trapmf(x_speed,[90 95 115
120]);
speed_Z=trapmf(x_speed,[127 127 127
127]);
speed_NSS=trapmf(x_speed,[135 140 160
165]);
speed_NS=trapmf(x_speed,[160 165 185
190]);
speed_NM=trapmf(x_speed,[185 190 210
215]);
speed_NF=trapmf(x_speed,[210 215 235
240]);

%IF-THEN RULES
rulehasil=[];
%untuk kiri ada 24 rules

A=jarak_far(jarak_real);
B=max(sdt_CF1(sdt_tuj),sdt_CF1(sdt_tu
j));
ruleand=min(A,B);
rulehasil=[rulehasil;
min(ruleand,speed_PF)];
B=sdt_CR(sdt_tuj);
ruleand=min(A,B);
rulehasil=[rulehasil;
min(ruleand,speed_PF)];
B=sdt_FR(sdt_tuj);
ruleand=min(A,B);
rulehasil=[rulehasil;
min(ruleand,speed_PF)];
B=sdt_CB(sdt_tuj);
ruleand=min(A,B);
rulehasil=[rulehasil;
min(ruleand,speed_NF)];
B=sdt_FL(sdt_tuj);
ruleand=min(A,B);
rulehasil=[rulehasil;
min(ruleand,speed_NF)];
B=sdt_BR(sdt_tuj);
ruleand=min(A,B);
rulehasil=[rulehasil;
min(ruleand,speed_PS)];
B=sdt_CL(sdt_tuj);
ruleand=min(A,B);
rulehasil=[rulehasil;
min(ruleand,speed_NS)];
B=sdt_CR(sdt_tuj);
ruleand=min(A,B);
rulehasil=[rulehasil;
min(ruleand,speed_NM)];
B=sdt_FR(sdt_tuj);
ruleand=min(A,B);
rulehasil=[rulehasil;
min(ruleand,speed_PSS)];
B=sdt_BR(sdt_tuj);
ruleand=min(A,B);
rulehasil=[rulehasil;
min(ruleand,speed_NM)];

A=jarak_medium(jarak_real);
B=max(sdt_CF1(sdt_tuj),sdt_CF1(sdt_tu
j));
ruleand=min(A,B);
rulehasil=[rulehasil;
min(ruleand,speed_PM)];
B=sdt_CB(sdt_tuj);
ruleand=min(A,B);
rulehasil=[rulehasil;
min(ruleand,speed_NM)];
B=sdt_CR(sdt_tuj);
ruleand=min(A,B);
rulehasil=[rulehasil;
min(ruleand,speed_NM)];
B=sdt_CL(sdt_tuj);
ruleand=min(A,B);
rulehasil=[rulehasil;
min(ruleand,speed_PM)];
B=sdt_CL(sdt_tuj);
ruleand=min(A,B);
rulehasil=[rulehasil;
min(ruleand,speed_NSS)];
B=sdt_FR(sdt_tuj);
ruleand=min(A,B);
rulehasil=[rulehasil;
min(ruleand,speed_PM)];
B=sdt_FL(sdt_tuj);
ruleand=min(A,B);
rulehasil=[rulehasil;
min(ruleand,speed_PSS)];
B=sdt_BR(sdt_tuj);
ruleand=min(A,B);
rulehasil=[rulehasil;
min(ruleand,speed_NM)];

A=jarak_near(jarak_real);
B=max(sdt_CF1(sdt_tuj),sdt_CF1(sdt_tu
j));
ruleand=min(A,B);
rulehasil=[rulehasil;
min(ruleand,speed_PS)];
B=sdt_CB(sdt_tuj);
ruleand=min(A,B);
rulehasil=[rulehasil;
min(ruleand,speed_NS)];
B=sdt_CR(sdt_tuj);
ruleand=min(A,B);
rulehasil=[rulehasil;
min(ruleand,speed_PS)];
B=sdt_CL(sdt_tuj);
ruleand=min(A,B);
rulehasil=[rulehasil;
min(ruleand,speed_PS)];

```

```

rulehasil=[rulehasil;
min(ruleand, speed_NSS)];
B=sdt_FR(sdt_tuj);
ruleand=min(A,B);
rulehasil=[rulehasil;
min(ruleand, speed_PS)];
B=sdt_FL(sdt_tuj);
ruleand=min(A,B);
rulehasil=[rulehasil;
min(ruleand, speed_Z)];
B=sdt_BR(sdt_tuj);
ruleand=min(A,B);
rulehasil=[rulehasil;
min(ruleand, speed_Z)];
B=sdt_BL(sdt_tuj);
ruleand=min(A,B);
rulehasil=[rulehasil;
min(ruleand, speed_NS)];

%DEFUZZIFICATION UNTUK MOTOR KIRI
yhasil_kiri=max(rulehasil);
xx_kiri =
defuzz(x_speed,yhasil_kiri,'centroid'
)

rulehasil=[];
%untuk kanan ada 24 rules

A=jarak_far(jarak_real);
B=max(sdt_CF1(sdt_tuj),sdt_CF1(sdt_tu
j));
ruleand=min(A,B);
rulehasil=[rulehasil;
min(ruleand, speed_PF)];
B=sdt_CB(sdt_tuj);
ruleand=min(A,B);
rulehasil=[rulehasil;
min(ruleand, speed_NF)];
B=sdt_CR(sdt_tuj);
ruleand=min(A,B);
rulehasil=[rulehasil;
min(ruleand, speed_NSS)];
B=sdt_CL(sdt_tuj);
ruleand=min(A,B);
rulehasil=[rulehasil;
min(ruleand, speed_PF)];
B=sdt_FR(sdt_tuj);
ruleand=min(A,B);
rulehasil=[rulehasil;
min(ruleand, speed_PS)];
B=sdt_FL(sdt_tuj);
ruleand=min(A,B);
rulehasil=[rulehasil;
min(ruleand, speed_PF)];
B=sdt_BR(sdt_tuj);
ruleand=min(A,B);
rulehasil=[rulehasil;
min(ruleand, speed_NF)];
B=sdt_BL(sdt_tuj);
ruleand=min(A,B);
rulehasil=[rulehasil;
min(ruleand, speed_NS)];

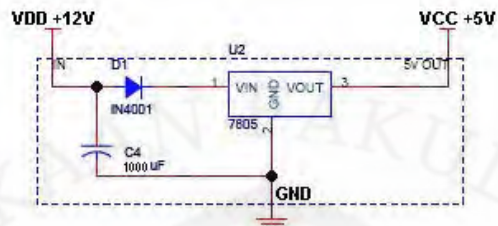
A=jarak_medium(jarak_real);
B=max(sdt_CF1(sdt_tuj),sdt_CF1(sdt_tu
j));
ruleand=min(A,B);
rulehasil=[rulehasil;
min(ruleand, speed_PM)];

B=sdt_CB(sdt_tuj);
ruleand=min(A,B);
rulehasil=[rulehasil;
min(ruleand, speed_NM)];
B=sdt_CR(sdt_tuj);
ruleand=min(A,B);
rulehasil=[rulehasil;
min(ruleand, speed_NSS)];
B=sdt_CL(sdt_tuj);
ruleand=min(A,B);
rulehasil=[rulehasil;
min(ruleand, speed_PM)];
B=sdt_FR(sdt_tuj);
ruleand=min(A,B);
rulehasil=[rulehasil;
min(ruleand, speed_PSS)];
B=sdt_FL(sdt_tuj);
ruleand=min(A,B);
rulehasil=[rulehasil;
min(ruleand, speed_PM)];
B=sdt_BR(sdt_tuj);
ruleand=min(A,B);
rulehasil=[rulehasil;
min(ruleand, speed_NM)];
B=sdt_BL(sdt_tuj);
ruleand=min(A,B);
rulehasil=[rulehasil;
min(ruleand, speed_NSS)];

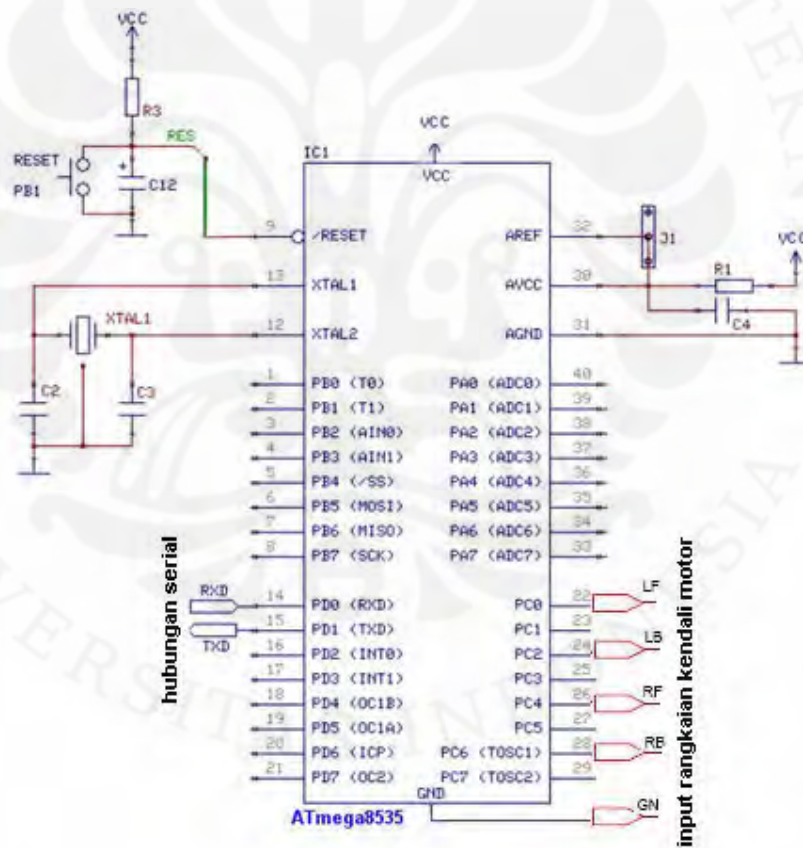
A=jarak_near(jarak_real);
B=max(sdt_CF1(sdt_tuj),sdt_CF1(sdt_tu
j));
ruleand=min(A,B);
rulehasil=[rulehasil;
min(ruleand, speed_PS)];
B=sdt_CB(sdt_tuj);
ruleand=min(A,B);
rulehasil=[rulehasil;
min(ruleand, speed_NS)];
B=sdt_CR(sdt_tuj);
ruleand=min(A,B);
rulehasil=[rulehasil;
min(ruleand, speed_NSS)];
B=sdt_CL(sdt_tuj);
ruleand=min(A,B);
rulehasil=[rulehasil;
min(ruleand, speed_PS)];
B=sdt_FR(sdt_tuj);
ruleand=min(A,B);
rulehasil=[rulehasil;
min(ruleand, speed_Z)];
B=sdt_FL(sdt_tuj);
ruleand=min(A,B);
rulehasil=[rulehasil;
min(ruleand, speed_PS)];
B=sdt_BR(sdt_tuj);
ruleand=min(A,B);
rulehasil=[rulehasil;
min(ruleand, speed_NS)];
B=sdt_BL(sdt_tuj);
ruleand=min(A,B);
rulehasil=[rulehasil;
min(ruleand, speed_Z)];

%DEFUZZIFICATION UNTUK MOTOR KANAN
yhasil_kanan=max(rulehasil);
xx_kanan =
defuzz(x_speed,yhasil_kanan,'centroid'
)
    
```

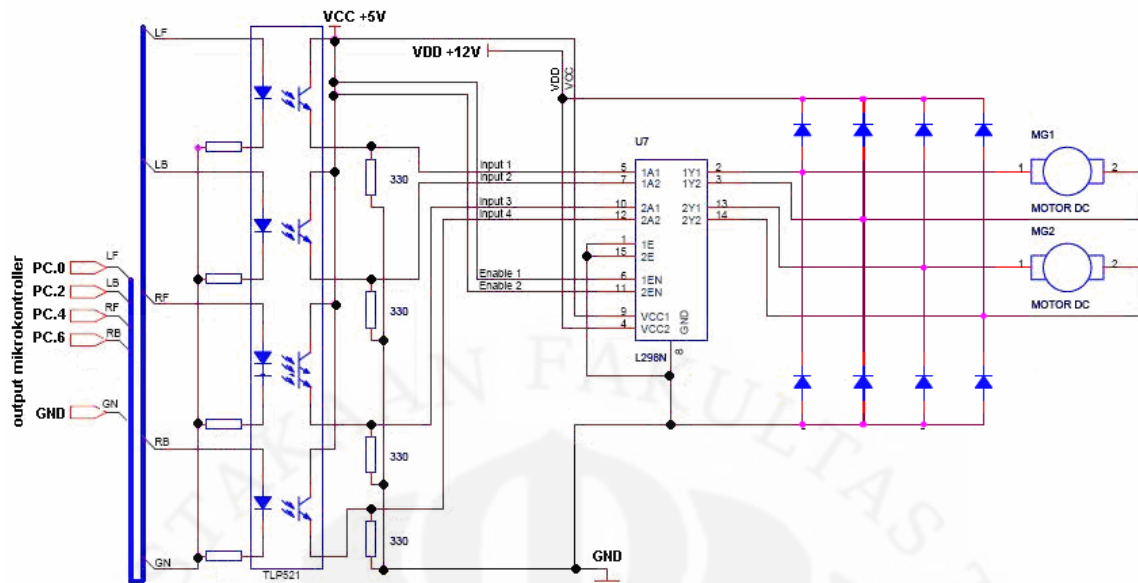
### Lampiran 4 Skema Rangkaian



Rangkaian Suplai Tegangan 12V dan 5V



Rangkaian Mikrokontroler



Rangkaian Kendali Motor

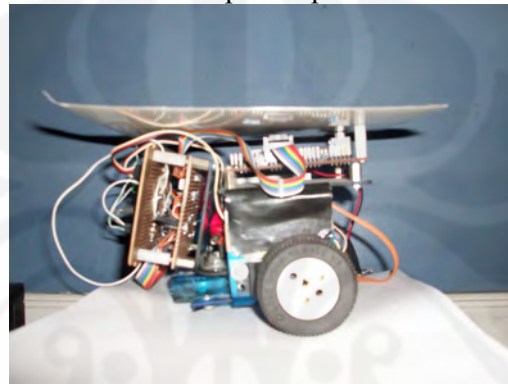
Lampiran 5 Foto Pengujian



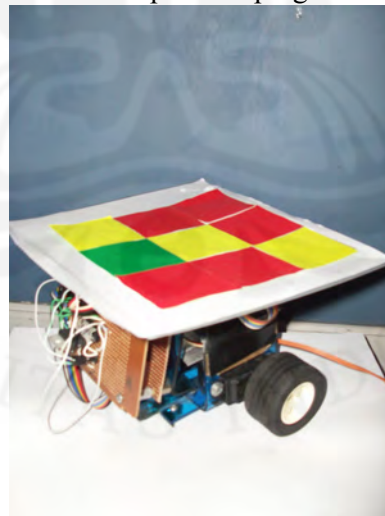
Lampiran 6 Foto Alat



Tampak Depan



Tampak Samping



Tampak Atas



Tampak Belakang

