



**UNIVERSITAS INDONESIA**

**RANCANG BANGUN PROTOTIPE SISTEM AKTUATOR  
MENGUNAKAN *BRUSHED DC MOTOR* DENGAN  
PENGENDALIAN *FUZZY***

**SKRIPSI**

**NUGROHO NANDAR DYTO  
0906603386**

**FAKULTAS TEKNIK  
PROGRAM STUDI TEKNIK ELEKTRO  
DEPOK  
JUNI 2012**



**UNIVERSITAS INDONESIA**

**RANCANG BANGUN PROTOTIPE SISTEM AKTUATOR  
MENGUNAKAN *BRUSHED DC MOTOR* DENGAN  
PENGENDALIAN *FUZZY***

**SKRIPSI**

**DIAJUKAN SEBAGAI SALAH SATU SYARAT UNTUK MEMPEROLEH  
GELAR SARJANA TEKNIK**

**NUGROHO NANDAR DYTO  
0906603386**

**FAKULTAS TEKNIK  
PROGRAM STUDI TEKNIK ELEKTRO  
DEPOK  
JUNI 2012**

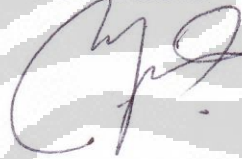
**HALAMAN PERNYATAAN ORISINALITAS**

**Skripsi ini adalah hasil karya saya sendiri,  
Dan semua sumber baik yang dikutip maupun dirujuk  
Telah saya nyatakan dengan benar.**

**Nama : Nugroho Nanda Dyto**

**NPM : 0906603386**

**Tanda Tangan :**



**Tanggal : 19 Juni 2012**

## HALAMAN PENGESAHAN

Skripsi ini diajukan oleh

Nama : Nugroho Nandar Dyto  
NPM : 0906603386  
Program Studi : Ekstensi Teknik Elektro  
Judul Skripsi : Rancang bangun prototipe sistem aktuator menggunakan *brushed dc motor* dengan pengendalian *fuzzy*.

Telah berhasil dipertahankan dihadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Elektro, Fakultas Teknik, Universitas Indonesia.

### DEWAN PENGUJI

Pembimbing : Dr. Abdul Halim M.Eng. (.....)

Penguji : Dr. Abdul Muis ST. M.Eng. (.....)

Penguji : Dr. Ir. Feri Yusivar M.Eng. (.....)

Ditetapkan di : Depok

Tanggal : 19 Juni 2012

## KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT karena atas limpahan rahmat dan karunia-Nya, penulis dapat menyelesaikan skripsi dengan judul “Rancang bangun prototipe sistem aktuator menggunakan *brushed dc motor* dengan pengendalian *fuzzy*” dengan baik. Penulisan skripsi ini dibuat untuk memenuhi salah satu persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Ekstensi Teknik Elektro, Fakultas Teknik, Universitas Indonesia.

Skripsi ini dapat tersusun dengan baik atas dukungan dan bimbingan dari berbagai pihak. Oleh karena itu, tidak lupa penulis sampaikan terima kasih yang sebesar-besarnya kepada:

1. Allah SWT yang telah memberikan rahmat dan ridho kepada penulis sehingga penulis dapat menyelesaikan penyusunan skripsi dengan baik.
2. Kedua orang tua penulis, bapak Sudarmo dan ibu Nandrawati serta kedua adik Laksmi dan Annisa yang senantiasa memberi dukungan dan motivasi tiada henti kepada penulis.
3. Dr. Abdul Halim, M. Eng., selaku dosen pembimbing penulis yang cukup sabar dalam memberikan arahan dan dukungan dalam penyusunan skripsi ini.
4. Rekan kerja penulis Ellan S Siregar, beserta rekan – rekan seperjuangan di Ekstensi Elektro 2009.
5. Senior dan junior di D3 Instrumentasi 2006, Slamet Mualif dan yang lainnya terima kasih atas masukan ide – idenya yang membantu penulis menyelesaikan skripsi ini.
6. Mitia Intansari yang setia menemani penulis dalam menyusun skripsi ini, terima kasih atas segala waktu dan pengertiannya.
7. Serta berbagai pihak yang penulis tidak dapat sebutkan satu – persatu tanpa mengurangi rasa terima kasih.

Semoga pihak-pihak yang terkait dalam penulisan skripsi ini Allah berikan balasan yang lebih atas kebaikan yang mereka berikan. Semoga skripsi ini memberikan manfaat dalam pengembangan ilmu pengetahuan, dan jika ada kekurangan dapat diperbaiki untuk penelitian selanjutnya.

Depok, Juni 2012

Penulis

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI UNTUK  
KEPENTINGAN AKADEMIS**

---

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : Nugroho Nandar Dyto  
NPM : 0906603386  
Program Studi : Ekstensi Teknik Elektro 2009  
Fakultas : Teknik  
Jenis Karya : Skripsi

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif** (*Non-exclusive Royalty-Free Right*) atas skripsi saya yang berjudul:

“Rancang Bangun Prototipe Sistem Aktuator Menggunakan *Brushed DC Motor* Dengan Pengendalian *Fuzzy*.”

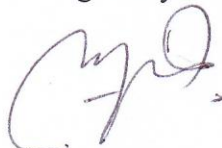
beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalih media/formatkan, mengelola dalam bentuk pangkalan data (database), merawat, dan mempublikasikan skripsi saya tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada tanggal : 19 Juni 2012

Yang Menyatakan



Nugroho Nandar Dyto

## ABSTRAK

Nama : Nugroho Nandar Dyto  
Program Studi : Ekstensi Teknik Elektro  
Judul : Rancang bangun prototipe sistem aktuator menggunakan *brushed dc motor* dengan pengendalian *fuzzy*.

Sebuah prototipe sistem pengendali posisi motor dc telah dirancang dan dibangun sebagai pengendali sistem aktuator pergerakan sirip pada roket kendali berbasis mikrokontroler ATmega yang menggunakan metode pengendalian logika fuzzy. Pengaturan posisi gerak motor dilakukan dengan mengatur tegangan motor dan menggunakan metode PWM (*Pulse Width Modulation*). Mekanisme umpan-balik sistem menggunakan sebuah sensor putaran yang membaca posisi dari motor dc. Metode fuzzy yang dirancang memiliki 2 nilai *crisp* input (*error* dan  $\Delta error$ ) dan satu nilai *crisp output* yaitu perubahan tegangan. Metode defuzzifikasi yang digunakan adalah metode *centre of gravity* (COG). Respon sistem ditampilkan dalam bentuk sudut posisi aktuator terhadap waktu dan didapatkan nilai  $T_r = 0,32$  detik,  $T_p = 0,47$  detik,  $T_s = 0,72$  detik dengan nilai persentase *overshoot* sebesar 21,57% dan kesalahan tunak sebesar 20 %.

Kata Kunci : sistem aktuator, pengendalian fuzzy, ATmega mikrokontroler, *brushed dc motor*, PWM.

## ABSTRACT

Name : Nugroho Nandar Dyto  
Study Program : Electrical Engineering – Extension  
Title : Development of brushed dc motor actuator system prototype using fuzzy control.

A prototype of dc motor position control system has been designed and built as a controller of fin control actuator system. This prototype uses fuzzy control method that has been embeded in ATmega microcontroller. Regulation of motor angular position has been implemented by adjusting motor voltage and used PWM (Pulse Width Modulation). Feedback mechanism has been done using rotation sensor that reads the angular position of dc motor. Fuzzy method is designed to have two crisp input (error and  $\Delta$ error) and one crisp output i.e voltage change. Defuzzification method used is Center Of Gravity (COG). From system respon, it has been shown that  $T_r = 0,32$  sec,  $T_p = 0,47$  sec,  $T_s = 0,72$  sec, percentage of overshoot 21,57 % and steady-state error of 20 %.

Keywords: actuator system, fuzzy controller, ATmega microcontroller, brushed dc motor, PWM



## DAFTAR ISI

HALAMAN JUDUL .....	i
HALAMAN PERNYATAAN ORISINALITAS.....	ii
HALAMAN PENGESAHAN .....	iii
KATA PENGANTAR .....	iv
LEMBAR PERSETUJUAN PUBLIKASI KARYA ILMIAH .....	v
ABSTRAK .....	vi
ABSTRACT .....	vii
DAFTAR ISI .....	viii
DAFTAR GAMBAR .....	x
DAFTAR TABEL .....	xii
<b>BAB 1 PENDAHULUAN</b>	
1.1 Latar Belakang .....	1
1.2 Perumusan Masalah .....	2
1.3 Tujuan Penelitian.....	2
1.4 Manfaat Penelitian .....	3
1.5 Batasan Penelitian .....	3
1.6 Metode Penelitian .....	3
1.7 Sistematika Penulisan .....	3
<b>BAB 2 DASAR TEORI</b> .....	5
2.1 Sistem Pengendalian Sirip Roket .....	5
2.2 Motor DC .....	6
2.2.1 Prinsip Kerja Brushed DC Motor .....	8
2.2.2 Brushed DC Motor Driver .....	9
2.3 Roda Gigi ( <i>Gear</i> ) .....	10
2.3.1 Planetary Gear .....	11
2.3.2 Bevel Gear .....	12
2.4 Mikrokontroler AVR ATmega8535 .....	13
2.4.1 Arsitektur ATmega8535 .....	14
2.4.2 Konfigurasi Pin ATmega8535 .....	15
2.5 Mekanisme Umpan Balik ( <i>Feedback</i> ) .....	17
2.5.3 <i>Variable Resistor</i> .....	17
2.6 Keypad .....	18
2.7 Metode Pengendalian <i>Fuzzy Logic</i> .....	19
2.7.1. Konsep Dasar Himpunan Fuzzy.....	19
2.7.2 Struktur Dasar Sistem Fuzzy.....	21
2.7.2.1 <i>Knowledge Base</i> .....	22
2.7.2.2 Inferensi .....	23
2.7.2.3 Defuzzifikasi .....	23
2.8 Teori Karakteristik Respon .....	24
<b>BAB 3 PERANCANGAN SISTEM</b> .....	26
3.1 Deskripsi Sistem .....	26
3.2 Perancangan Perangkat Keras .....	27

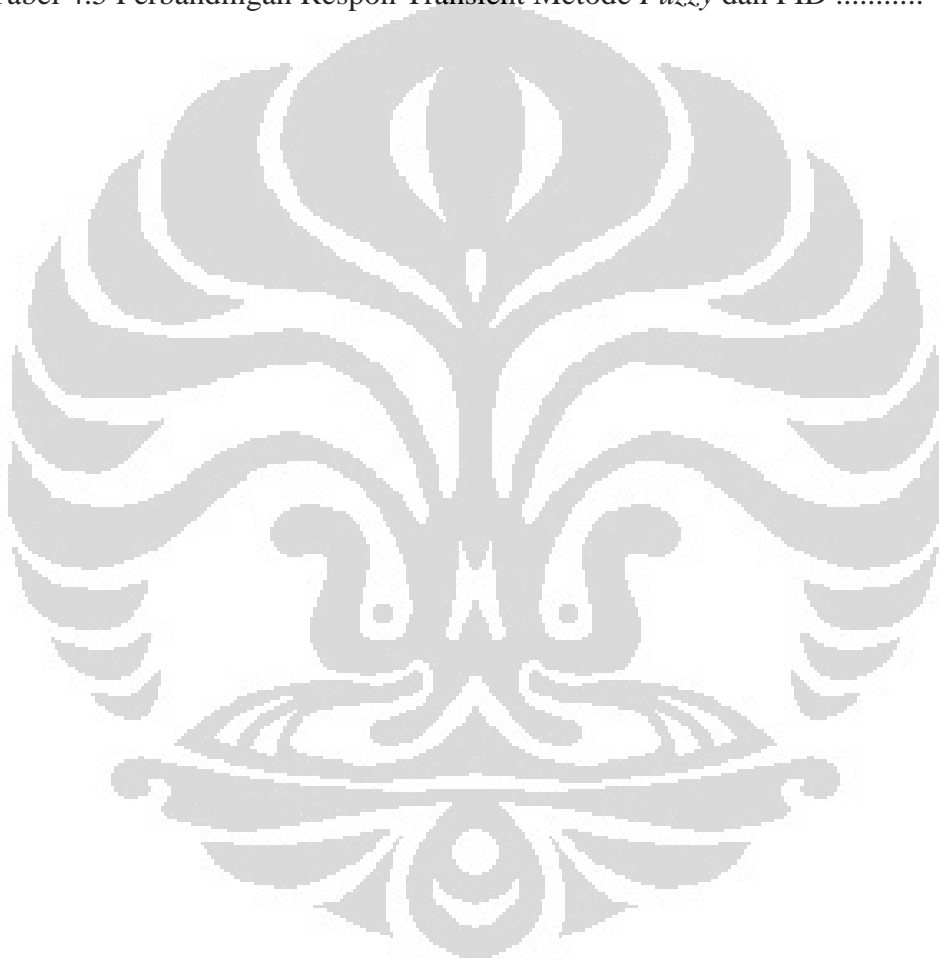
3.2.1 <i>Minimum System</i> DI-Smart AVR .....	27
3.2.2 Perancangan Aktuator Kendali Sirip .....	30
3.2.3 Motor DC yang digunakan .....	29
3.2.4 <i>Planetary Gear</i> yang digunakan .....	31
3.2.5 Driver Motor DC yang digunakan .....	31
3.3 Konfigurasi <i>Keypad</i> .....	32
3.4 Perancangan sistem Fuzzy Logic Control .....	35
3.4.1 Fuzzifikasi .....	35
3.4.2 Inferensi .....	37
3.4.3 Defuzzifikasi .....	38
3.5 Perancangan Mekanisme Umpan Balik .....	38
<b>BAB 4 PENGUJIAN DAN ANALISA HASIL</b> .....	<b>39</b>
4.1 Alat Ukur yang Digunakan .....	39
4.2 Pengujian Rangkaian Driver Motor DC .....	40
4.3 Pengujian Mekanisme Umpan-Balik .....	42
4.4 Pengujian Sistem Open-Loop .....	46
4.5 Pengujian Verifikasi Sistem Fuzzy Logic .....	47
4.6 Pengujian Pengendali Fuzzy Pada Sirip .....	50
4.7 Perbandingan Kinerja Pengendali .....	54
<b>BAB 5 PENUTUP</b>	
5.1 Kesimpulan .....	56
5.2 Saran .....	57
<b>DAFTAR REFERENSI</b> .....	<b>58</b>
<b>LAMPIRAN</b>	

## DAFTAR GAMBAR

Gambar 2.1	Posisi Sirip Kendali Pada Roket .....	5
Gambar 2.2	Proses Konversi Energi Pada Motor DC .....	6
Gambar 2.3	Brushed DC Motor Dua Kutub .....	7
Gambar 2.4	Prinsip Kerja Motor .....	8
Gambar 2.5	Rangkaian <i>H-Bridge Driver</i> .....	9
Gambar 2.6	Susunan <i>Planetary Gear</i> .....	11
Gambar 2.7	<i>Bevel Gear</i> .....	13
Gambar 2.8	Blok Diagram ATmega8535 .....	15
Gambar 2.9	Susunan Pin ATmega8535 .....	16
Gambar 2.10	Skematik <i>Variable Resistor</i> .....	18
Gambar 2.11	Rangkaian <i>Keypad</i> .....	18
Gambar 2.12	Contoh Penggunaan <i>Keypad</i> Matriks 4x4 .....	19
Gambar 2.13	Grafik Fungsi Keanggotaan .....	20
Gambar 2.14	Struktur Dasar Pengendali <i>Fuzzy Logic</i> .....	22
Gambar 3.1	Blok Diagram Sistem .....	26
Gambar 3.2	Rangkaian <i>Minimum System</i> DI-Smart AVR .....	28
Gambar 3.3	DI-Smart AVR .....	28
Gambar 3.4	Desain Mekanik Aktuator Pergerakan Sirip .....	29
Gambar 3.5	<i>Pitch Gauge</i> .....	29
Gambar 3.6	MAXON <i>Brushed DC Motor</i> .....	30
Gambar 3.7	MAXON <i>Planetary Gear</i> .....	31
Gambar 3.8	Rangkaian IC L298 .....	32
Gambar 3.9	DI-MDCD4A .....	32
Gambar 3.10	<i>Flowchart</i> Pengecekan <i>Keypad</i> dengan Metode <i>Scanning</i> ..	33
Gambar 3.11	Tampilan <i>Listing Program</i> untuk <i>Scanning</i> data <i>Keypad</i> ..	34
Gambar 3.12	<i>Flowchart</i> Struktur <i>Fuzzy Logic</i> .....	35
Gambar 3.13	Tipe Keanggotaan Segitiga .....	36
Gambar 3.14	Tipe Keanggotaan Trapesium .....	36
Gambar 3.15	DFRobot <i>Rotation Sensor v.1</i> .....	38
Gambar 4.1	Multimeter HELES UX37 .....	39
Gambar 4.2	Tampilan aplikasi <i>hyperterminal</i> .....	40
Gambar 4.3	Rangkaian IC L298N .....	41
Gambar 4.4	Konfigurasi Hubungan <i>Rotation Sensor</i> ke Mikrokontroler	43
Gambar 4.5	<i>Listing Program Basic Compiler</i> Untuk Menampilkan Nilai ADC .....	43
Gambar 4.6	Grafik Respon <i>Brushed DC Motor</i> Sistem <i>Open-Loop</i> ....	47
Gambar 4.7	Diagram Alir Metode <i>Fuzzy Logic</i> Pada Sistem .....	47
Gambar 4.8	Fungsi <i>Membership Function</i> pada Contoh Kasus .....	48
Gambar 4.9	Hasil Simulasi BASCOM .....	49
Gambar 4.10	<i>Flowchart</i> Sistem Pengendalian <i>Fuzzy Logic</i> .....	50
Gambar 4.11	Grafik Respon Pergerakan Aktuator Dari 0° Menuju 2° ....	51
Gambar 4.12	Grafik Respon Pergerakan Aktuator Dari 0° Menuju 5° ....	52
Gambar 4.13	Grafik Respon Pergerakan Aktuator Dari 0° Menuju 9° .....	52
Gambar 4.14	Grafik Respon Pergerakan Aktuator Dari 0° Menuju 10° ....	53

## DAFTAR TABEL

Tabel 3.1 Spesifikasi MAXON <i>Brushed DC Motor</i> .....	30
Tabel 3.2 Spesifikasi MAXON <i>Planetary Gear</i> .....	31
Tabel 3.3 Data Hasil Metode Scanning Pada Keypad Matriks 4x4 .....	34
Tabel 3.4 <i>Rule Base</i> Perancangan Sistem .....	37
Tabel 4.1 Pengujian Rangkaian Driver Motor L298N Tanpa Beban .....	41
Tabel 4.2 Pengujian Driver Motor DC Dengan Beban Motor DC .....	42
Tabel 4.3 Data ADC Pada Setiap Perubahan Sudut .....	44
Tabel 4.4 Respon Rata – Rata Aktuator Sirip Metode Fuzzy .....	54
Tabel 4.5 Perbandingan Respon Transient Metode <i>Fuzzy</i> dan PID .....	55



# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Kebutuhan suatu negara dalam bidang pertahanan dan keamanan khususnya dibidang militer seringkali menjadi pemicu dibuatnya penelitian tentang alat – alat atau komponen yang bisa menjadi penunjang kemajuan dari militer tersebut. Bukan rahasia lagi bahwa kemajuan elektronika saat ini banyak dimanfaatkan dibidang militer untuk membantu mengurangi peran manusia untuk menguji coba alat – alat penunjang perang terbaru dengan tujuan mengurangi resiko cederanya prajurit saat menguji coba senjata tersebut.

Kemajuan militer di suatu negara bisa membuat nama negara tersebut disegani di dunia internasional bahkan sekarang senjata – senjata tersebut bisa diperjualbelikan antar negara. Tentu saja hal ini cukup menggiurkan karena pemasukan yang bisa diterima negara tidak sedikit jumlahnya. Dewasa ini bidang militer sudah sampai dimana dilakukan penelitian – penelitian untuk membuat dan mengembangkan senjata – senjata pemusnah massal yang tidak saja berdaya ledak besar tetapi juga bisa dikendalikan dari jarak jauh.

Penelitian yang dilakukan pada skripsi ini merupakan salah satu contoh simulasi dari aplikasi elektronika yang digunakan di senjata militer. Penelitian ini dibatasi pada pengendalian pergerakan sudut dari aktuator sirip roket yang akan menentukan arah pergerakan dari roket tersebut. Pada skripsi ini penggerak yang digunakan adalah sebuah *brushed DC motor* dengan *planetary gear* yang dihubungkan dengan aktuator sirip yang terbuat dari bahan *fiber plasttic*. Sistem pengendalian dengan menggunakan Logika *Fuzzy* diterapkan untuk mengendalikan putaran dari *brushed DC motor* tersebut untuk mencapai posisi sudut yang diinginkan.

Sebagai pengendali utama digunakan sebuah minimum system dengan sebuah ic mikrokontroler ATmega8535 dengan menggunakan bahasa pemrograman *basic compiler* (BASCUM). Mikrokontroler ini berfungsi sebagai pengolah data yang diterima dari input yang diberikan dan akan memberikan keluaran yang terhubung ke aktuator sirip yang dalam hal ini menggunakan *brushed DC motor*. Data yang diterima oleh mikrokontroler adalah data yang

diberikan berupa besaran sudut yang ingin dicapai, lalu data tersebut diolah oleh mikrokontroler yang kemudian akan mengendalikan pergerakan motor sehingga mencapai hasil posisi sudut yang diinginkan. Untuk mengetahui apakah posisi dari sirip sudah sesuai yang diinginkan maka dibutuhkan sebuah data umpan balik yang dalam hal ini diberikan oleh sebuah rotation sensor untuk mendapatkan sudut pergerakan sirip yang akurat.

## 1.2 Perumusan Masalah

Berdasarkan latar belakang tersebut, maka perumusan masalah ini akan ditekankan pada :

1. Bagaimana mengendalikan sebuah DC motor dengan menggunakan metode pengendalian Logika *Fuzzy*.
2. Bagaimana mendapatkan pergerakan aktuator sirip yang akurat dengan rentang jarak antara -10 derajat sampai 10 derajat dengan pergerakan per 1 derajat.
3. Bagaimana menyusun program pada mikrokontroler AVR ATmega8535 dengan menggunakan bahasa *basic compiler* BASCOM AVR untuk pengendalian pergerakan sirip yang akurat dengan bantuan data umpan balik dari rotation sensor.
4. Bagaimana menentukan metode logika *fuzzy* yang sesuai sehingga mendapatkan respon sistem yang baik.

## 1.3 Tujuan Penelitian

1. Membuat prototipe aktuator sistem pengendali sirip.
2. Mempelajari dan mengaplikasikan metode pemrograman menggunakan mikrokontroler ATmega8535.
3. Mempelajari dan merancang pengendalian untuk brushed dc motor.
4. Mempelajari dan membuat aplikasi pengendalian menggunakan metode *fuzzy*.
5. Mempelajari dan mengaplikasikan sensor rotasi berbasis *variable resistor*.

#### 1.4 Manfaat Penelitian

Manfaat yang didapat dari penelitian ini antara lain:

1. Hasil yang didapatkan dari penelitian ini diharapkan bisa dikembangkan dipenelitian berikutnya seperti penyempurnaan di bagian mekanik, pemilihan sensor dan juga pemilihan metode pengendalian yang tepat agar bisa diaplikasikan di bidang transportasi, keamanan, maupun pertahanan.
2. Jika diterapkan untuk keperluan militer, akan membuat kekuatan militer kita disegani dan tidak menutup kemungkinan akan ada negara lain yang akan membeli alat ini sehingga akan memberikan pemasukan bagi negara.

#### 1.5 Batasan Penelitian

Pembahasan penelitian ini dibatasi oleh beberapa hal, diantaranya :

1. Pengendalian sudut aktuator sirip dengan metode pengendalian logika *fuzzy*.
2. Pembuatan program untuk mikrokontroler.

#### 1.6 Metode Penelitian

Metode penelitian yang digunakan dalam penelitian ini adalah:

1. Tinjauan Pustaka
2. Perancangan Sistem
3. Pengambilan Data
4. Analisa dan Pengolahan Data
5. Pembuatan Laporan

#### 1.7 Sistematika Penulisan

Pada penulisan skripsi ini, pembahasan penelitian akan diuraikan dalam lima bab sebagai berikut:

##### BAB 1 PENDAHULUAN

Bab ini berisi tentang latar belakang dibuatnya penelitian ini, tujuan yang akan dicapai, manfaat yang akan didapat, batasan –

batasan permasalahan yang akan dibahas serta metode penelitian yang akan digunakan.

## **BAB 2 TEORI DASAR**

Bab ini berisi tentang teori – teori pendukung yang akan membantu memecahkan masalah yang berkaitan dengan penelitian yang didapat dari buku maupun literatur.

## **BAB 3 PERANCANGAN SISTEM**

Pada bab ini membahas tentang perancangan sistem kendali sirip yang akan dibuat. Membahas tentang hardware maupun software serta mencakup segala komponen yang akan digunakan.

## **BAB 4 PENGUJIAN DAN ANALISA HASIL**

Membahas tentang proses pengujian dari sistem yang telah dibuat serta menganalisa segala kelebihan dan kekurangan berdasarkan data yang telah didapatkan.

## **BAB 5 KESIMPULAN DAN SARAN**

Bab ini berisi tentang kesimpulan dari hasil analisa yang didapat dari hasil penelitian yang dilakukan. Saran – saran tentang kekurangan yang dapat diperbaiki untuk penelitian selanjutnya agar didapati hasil seperti yang diinginkan juga dicantumkan dalam bab penutup ini.



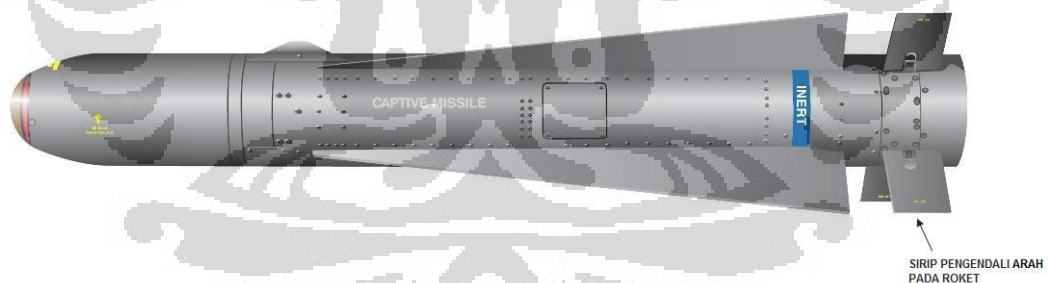
## BAB 2

### DASAR TEORI

Pada bab ini akan membahas tentang teori – teori dasar yang digunakan dalam perancangan sistem kendali sirip roket ini. Bahan – bahan yang menjadi acuan diambil dari beberapa buku, hasil skripsi maupun jurnal yang telah dikutip oleh penulis.

#### 2.1 Sistem Pengendalian Sirip Roket

Pergerakan dari sebuah rudal ditentukan dari pergerakan sirip yang terdapat pada badan rudal tersebut. Dengan kemajuan teknologi dewasa ini, sirip rudal tersebut bisa dikendalikan dari jarak jauh. Bahkan hanya dengan mengirim koordinat yang diinginkan rudal akan mencapainya dengan bantuan kendali dari *autopilot*. *Autopilot* telah diprogram agar tetap mengarahkan rudal ke koordinat yang diinginkan walaupun banyak gangguan dari beban angin maupun gravitasi yang akan merubah arah roket. Sirip yang berfungsi sebagai pengendali arah dari roket dapat dilihat pada gambar 2.1 berikut:

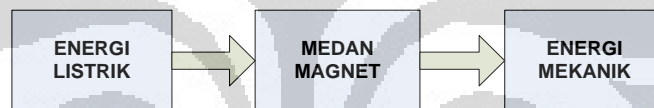


Gambar 2.1 Posisi Sirip Kendali Pada Roket<sup>[13]</sup>

Setiap  $1^\circ$  pergerakan dari sirip pengendali arah roket ini akan mengubah arah dari roket secara signifikan. Sirip ini digerakkan oleh sebuah motor yang dikendalikan oleh sebuah kontroler agar bergerak sesuai keinginan. Sebuah motor dengan torsi yang besar dipilih sebagai penggerak dari sirip roket ini agar mampu menahan beban di sirip yang dapat mengganggu arah roket yang biasanya disebabkan oleh kecepatan angin maupun gravitasi.

## 2.2 Motor DC

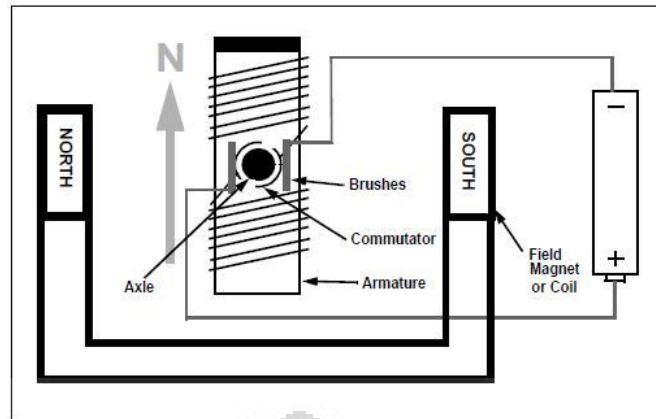
Motor DC secara umum adalah sebuah motor yang memerlukan sumber tegangan searah pada kumparan jangkar dan kumparan medan untuk diubah menjadi energi mekanik. Pada motor DC, kumparan medan yang dialiri arus listrik akan menghasilkan medan magnet yang melingkupi kumparan jangkar dengan arah tertentu. Konverter energi baik energi listrik menjadi energi mekanik (motor) maupun sebaliknya dari energi mekanik menjadi energi listrik (generator) berlangsung melalui medium medan magnet. Energi yang akan diubah dari suatu sistem ke sistem yang lain, sementara akan tersimpan pada medium medan magnet untuk kemudian dilepaskan menjadi energi system lainnya. Dengan demikian, medan magnet disini selain berfungsi sebagai tempat penyimpanan energi juga sekaligus proses perubahan energi, dimana proses perubahan energi pada motor arus searah dapat digambarkan pada gambar 2.2 berikut:



Gambar 2.2 Proses Konversi Energi Pada Motor DC

Salah satu jenis motor DC adalah motor DC dengan sikat (*brushed DC motor*). *Brushed DC motor* banyak digunakan dalam berbagai aplikasi mulai dari mainan anak-anak sampai menjadi penggerak dari sirip sebuah roket militer. *Brushed DC motor* banyak digunakan karena harganya yang cukup murah, banyak varian ukuran yang tersedia dan juga karena tidak terlalu sulit untuk dikendalikan.

Semua motor DC standar tersusun dari komponen dasar yang sama, yaitu stator, rotor, sikat, dan komutator. Contoh sederhana dari *brushed DC motor* dapat dilihat pada gambar 2.3 berikut:



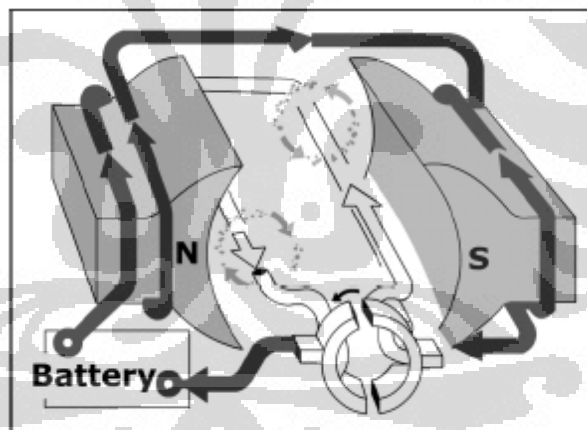
Gambar 2.3 Brushed DC Motor Dua Kutub<sup>[1]</sup>.

Bagian – bagian motor DC diantaranya adalah:

- Stator, bagian ini menghasilkan medan magnet yang mengelilingi rotor yang dihasilkan oleh magnet permanen maupun gulungan kumparan yang dialiri listrik. Perbedaan jenis motor dc bisa dibedakan dari rangka statornya atau cara gulungan kumparan yang terhubung ke sumber tegangan listrik searah.
- Rotor, yang juga biasa disebut angker terdiri dari satu atau lebih gulungan. Ketika gulungan ini teraliri listrik, maka akan menghasilkan medan magnet. Kutub magnetik dari rotor ini akan tertarik ke kutub berlawanan yang dihasilkan oleh stator dan akan menyebabkan rotor berputar. Rotor berputar dalam medan magnet yang dibentuk oleh kutub – kutub. Selagi motor berputar, gulungan harus terus diberi aliran arus dalam urutan yang berbeda sehingga kutub magnet yang dihasilkan oleh rotor tidak sama dengan yang kutub yang dihasilkan didalam stator. Pergantian kutub didalam gulungan kumparan rotor ini yang disebut komutasi<sup>[1]</sup>.
- Komutator, komponen ini terdapat pada motor DC dan berfungsi untuk membalikkan arah arus listrik dalam kumparan jangkar. Komutator juga membantu dalam transmisi arus antara kumparan jangkar dan saluran daya.
- Sikat (*Brush*), berfungsi untuk menghantarkan arus listrik untuk membangkitkan medan magnet yang diperlukan rotor untuk memberikan gaya tarik maupun tolak terhadap medan magnet stator agar rotor dapat berputar.

### 2.2.1 Prinsip Kerja Brushed DC Motor

Yang akan dibahas berikut adalah contoh prinsip kerja dari *brushed DC motor* dengan magnet permanen yang dapat dilihat pada gambar 2.4. *Brushed DC motor* jenis ini memiliki dua buah magnet permanen sehingga timbul medan magnet di antara kedua magnet tersebut. Di dalam medan magnet inilah rotor berputar. Rotor yang terletak di tengah motor memiliki jumlah kutub yang ganjil dan pada setiap kutubnya terdapat lilitan. Lilitan ini terhubung ke komutator. Sikat (*brushes*) yang terhubung ke kutub positif dan negatif motor memberikan daya ke lilitan sedemikian rupa sehingga kutub yang satu akan ditolak oleh magnet permanen yang berada di dekatnya, sedangkan lilitan lain akan ditarik ke magnet permanen yang lain sehingga menyebabkan jangkar berputar. Ketika jangkar berputar, komutator mengubah lilitan yang mendapat pengaruh polaritas medan magnet sehingga rotor akan terus berputar selama kutub positif dan negatif motor diberi daya<sup>[1]</sup>.



Gambar 2.4 Prinsip Kerja Motor<sup>[1]</sup>

Kecepatan putar motor DC ( $N$ ) dirumuskan dengan persamaan berikut<sup>[1]</sup>:

$$N = \frac{V_{TM} - I_A R_A}{K\phi} \quad (2.1)$$

Keterangan :

$V_{TM}$  = Tegangan Terminal

$I_A$  = Arus Jangkar (Rotor)

$R_A$  = Hambatan Jangkar

$K$  = Konstanta Motor

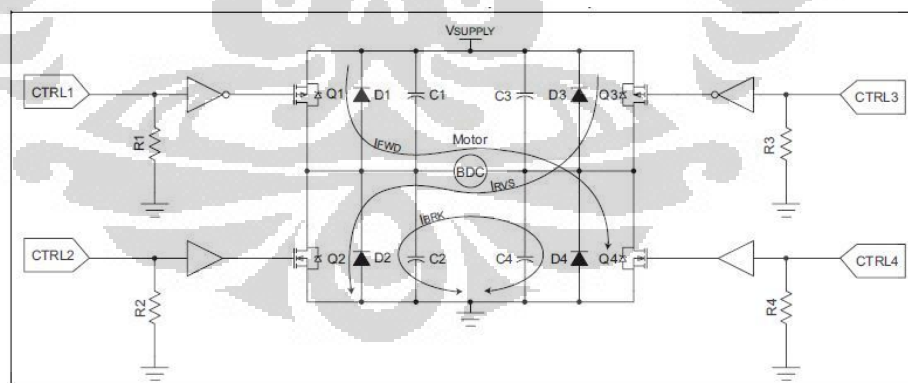
$\Phi$  = Fluks magnet yang terbentuk pada motor

Pengendalian kecepatan putar motor DC dapat dilakukan dengan mengatur besar tegangan terminal motor  $V_{TM}$ . Metode lain yang biasa digunakan untuk mengendalikan kecepatan motor DC adalah dengan teknik modulasi lebar pulsa atau *Pulse Width Modulation (PWM)*, *Proportional, Integral, Derivative (PID)*, dan *fuzzy logic*.

### 2.2.2 Brushed DC Motor Driver

Untuk menggerakkan sebuah motor DC diperlukan sebuah driver untuk mengatur arah putaran dari motor tersebut. Setiap jenis motor pasti mempunyai rangkaian driver yang berbeda pula. Untuk *brushed DC motor*, rangkaian yang paling sering digunakan untuk menggerakkan atau mengatur arahnya adalah rangkaian *H-bridge*.

*H-bridge* adalah sebuah rangkaian yang biasanya terdiri dari transistor mosfet dan diode yang berfungsi untuk menggerakkan motor. Rangkaian ini diberi nama *H-bridge* karena bentuk rangkaiannya yang menyerupai huruf H seperti pada gambar 2.5 berikut:



Gambar 2.5 Rangkaian *H-Bridge Driver*<sup>[1]</sup>

Pada rangkaian *H-bridge* biasanya menggunakan transistor MOSFET sebagai *switching*. Transistor jenis MOSFET dipilih karena transistor ini terkenal karena kesanggupan dilalui arus yang relatif besar jika dibandingkan dengan transistor lain, serta memiliki daya disipasi yang kecil. Sehingga Transistor ini

dapat menghemat pemakaian daya. Pada rangkaian diatas menggunakan dua buah MOSFET tipe P dan dua buah MOSFET tipe N. Prinsip kerja rangkaian ini adalah mengatur urutan aktif dari keempat MOSFET tersebut. MOSFET Q1 dan Q3 adalah MOSFET tipe P yang aktif apabila diberikan logika 0 (aktif low) dan akan meneruskan tegangan kutub positif dari catu daya ke input motor. Sementara MOSFET Q2 dan Q4 adalah MOSFET tipe N yang aktif apabila diberikan logika 1 (aktif hi) dan akan memberikan input negatif dari catu daya ke input motor.

Pada gambar diatas dapat dilihat saat MOSFET Q1 dan Q4 dalam keadaan *on* dan MOSFET Q2 dan Q3 dalam keadaan *off*, maka sisi kiri motor akan terhubung dengan tegangan positif dan sisi kanan motor akan terhubung dengan kutub negatif dari catu daya. Dengan keadaan seperti ini maka motor akan bergerak kekanan atau searah dengan arah jarum jam (*clockwise*). Sebaliknya, MOSFET Q1 dan Q4 dalam keadaan *off* dan MOSFET Q2 dan Q3 dalam keadaan *on*, maka sisi kanan motor akan terhubung dengan tegangan positif sedangkan sisi kiri motor akan terhubung dengan kutub negatif dari catu daya dan akan mengakibatkan motor akan bergerak berlawanan arah jarum jam (*counter clockwise*).

Konfigurasi lain yang dapat terjadi adalah apabila MOSFET Q1 dan Q3 dalam keadaan *on* sedangkan MOSFET Q2 dan Q4 keadaan *off* ataupun sebaliknya MOSFET Q1 dan Q3 dalam keadaan *off* sedangkan MOSFET Q2 dan Q4 dikeadaan *on*. Konfigurasi ini akan menyebabkan sisi kiri dan kanan motor terhubung pada kutub yang sama sehingga tidak ada perbedaan tegangan diantara dua buah polaritas motor, sehingga motor akan diam. Konfigurasi seperti ini disebut dengan konfigurasi *break*. Konfigurasi yang harus dihindari adalah pada saat MOSFET Q1 dan Q2 dalam keadaan *on* secara bersamaan atau MOSFET Q3 dan Q4 dikeadaan *on* secara bersamaan. Pada konfigurasi ini akan terjadi hubungan arus singkat antara kutub positif catu daya dengan kutub negatif catu daya, dan akan mengakibatkan kerusakan pada MOSFET tersebut.

### 2.3 Roda Gigi (*Gear*)

Rodagigi (*Gear*) digunakan untuk mentransmisikan daya besar dan putaran yang tepat. Rodagigi memiliki gigi di sekelilingnya, sehingga penerusan

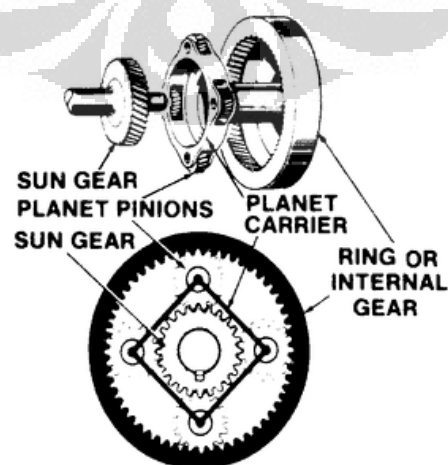
daya dilakukan oleh gigi-gigi kedua roda yang saling berkait. Rodagigi sering digunakan karena dapat meneruskan putaran dan daya yang lebih bervariasi dan lebih kompak daripada menggunakan alat transmisi yang lainnya, selain itu rodagigi juga memiliki beberapa kelebihan jika dibandingkan dengan alat transmisi lainnya, yaitu<sup>[2]</sup>:

1. Sistem transmisinya lebih ringkas, putaran lebih tinggi dan daya yang besar.
2. Sistem yang kompak sehingga konstruksinya sederhana.
3. Kemampuan menerima beban lebih tinggi.
4. Efisiensi pemindahan dayanya tinggi karena faktor terjadinya slip sangat kecil.
5. Kecepatan transmisi rodagigi dapat ditentukan sehingga dapat digunakan dengan pengukuran yang kecil dan daya yang besar.

### 2.3.1 Planetary Gear

*Planetary gear* digunakan untuk mereduksi kecepatan dan meningkatkan torsi yang dihasilkan oleh sebuah motor listrik. Hal ini dibutuhkan untuk dapat menyesuaikan RPM motor sesuai dengan kecepatan dari poros aktuator sirip roket kendali. Rodagigi ini dapat memberikan torsi yang kuat bahkan pada kecepatan yang sangat rendah.

Berikut ini adalah bagian – bagian dari *planetary gear* :



Gambar 2.6 Susunan *Planetary Gear*<sup>[14]</sup>

Seperti terlihat pada gambar 2.6 diatas, sebuah *planetary gear* sederhana terdiri dari *sun gear*, *planetary pinions*, dan *ring gear*. *Sun gear* terletak dipusat susunan sebagai poros putaran. *Sun gear* dapat berupa spur atau *helical gear*. *Sun gear* bertautan dengan gigi pada *planetary pinion gear*.

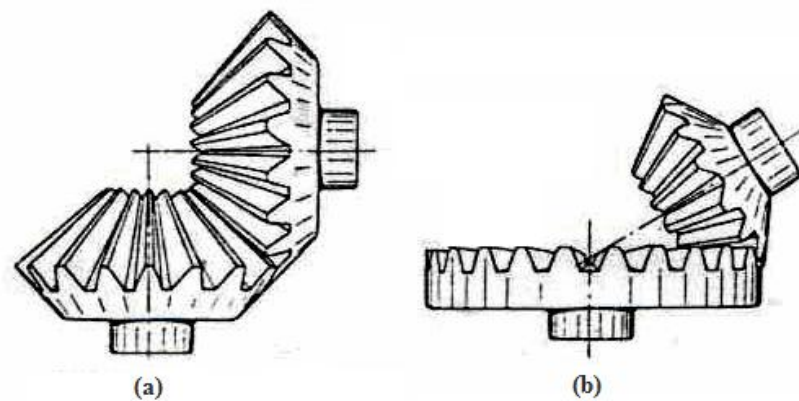
*Planetary pinion gear* adalah gear kecil yang disusun dalam kerangka yang disebut *planetary carrier*. *Planetary carrier* dapat terbuat dari besi tuang, alumunium atau pelat baja dan dirancang dengan sebuah *shaft* untuk masing-masing *planetary pinion gear*. *Planetary pinion* berputar pada *needle bearing* yang diposisikan diantara *shaft planetary carrier* dan *planetary pinion*. *Planetary pinion* mengelilingi poros tengah *sun gear* dan dilingkari oleh *annulus* atau *ring gear*. *Ring gear* bertindak seperti sebuah pengikat yang menahan keseluruhan gear set bersama dan memberikan kekuatan yang besar pada unit.

*Ring gear* diletakkan pada jarak terjauh dari poros pusat dan karena itu berfungsi sebagai tuas terbesar pada poros pusat. Untuk membantu mengingat rancangan *planetary gear set*, maka digunakan sistem tata surya sebagai contohnya. Sun adalah pusat tata surya dengan planet berputar disekelilingnya. Karena itu disebut *planetary gear set*.

### 2.3.2 *Bevel Gear*

*Bevel gear* digunakan untuk memindahkan arah gerakan antara dua poros yang berpotongan dengan sudut perpotongan sekitar  $90^{\circ}$ . Rodagigi kerucut lurus adalah jenis rodagigi kerucut yang mudah dan sederhana pembuatannya dan memberikan hasil yang baik dalam pemakaiannya bila pemasangan dilakukan secara tepat dan presisi. Susunan pada bevel gear dapat dilihat pada gambar 2.7 dibawah ini:





Gambar 2.7 *Bevel Gear*<sup>[15]</sup>

Jenis *bevel gear* ada dua macam, yaitu *miter gear* yang dapat dilihat pada gambar 2.7a yaitu pemasangan rodagigi kerucut secara berpotongan yang memiliki jumlah gigi yang sama yang biasanya digunakan untuk mengubah poros dengan sudut perpotongan  $90^\circ$  dan juga *skew bevel gear* yaitu gabungan dari rodagigi mahkota dengan rodagigik kerucut miring yang biasanya digunakan untuk sudut perpotongan lebih dari  $90^\circ$  yang dapat dilihat pada gambar 2.7b.

#### 2.4 Mikrokontroler AVR ATmega8535

Mikrokontroler adalah suatu piranti yang digunakan untuk mengolah data – data biner (*digital*) yang didalamnya merupakan gabungan dari rangkaian – rangkaian elektronik yang dikemas dalam bentuk suatu chip IC (*Integrated Circuit*) yang didalamnya terdapat mikroprosesor RAM (*Random Access Memory*), dan ROM (*Read-Only Memory*).

Salah satu jenis mikrokontroler yang dipakai adalah mikrokontroler ATmega8535 produksi Atmel. ATmega8535 merupakan alah satu mikrokontroler 8 bit dari keluarga AVR yang diproduksi secara massal pada tahun 2006<sup>[3]</sup>. Karena merupakan keluarga dari AVR (*Alf and Vegard's Risc*) dinamai berasal dari kedua nama penemunya yaitu Alf Egil Bogen dan Vegard Wollan, maka ATmega8535 juga menggunakan arsitektur RISC (*Reduced Instruction Set Computing*) bit, dimana semua instruksi dikemas dalam kode 16 bit dan sebagian besar instruksi dieksekusi dalam satu siklus *clock*. Berbeda dengan

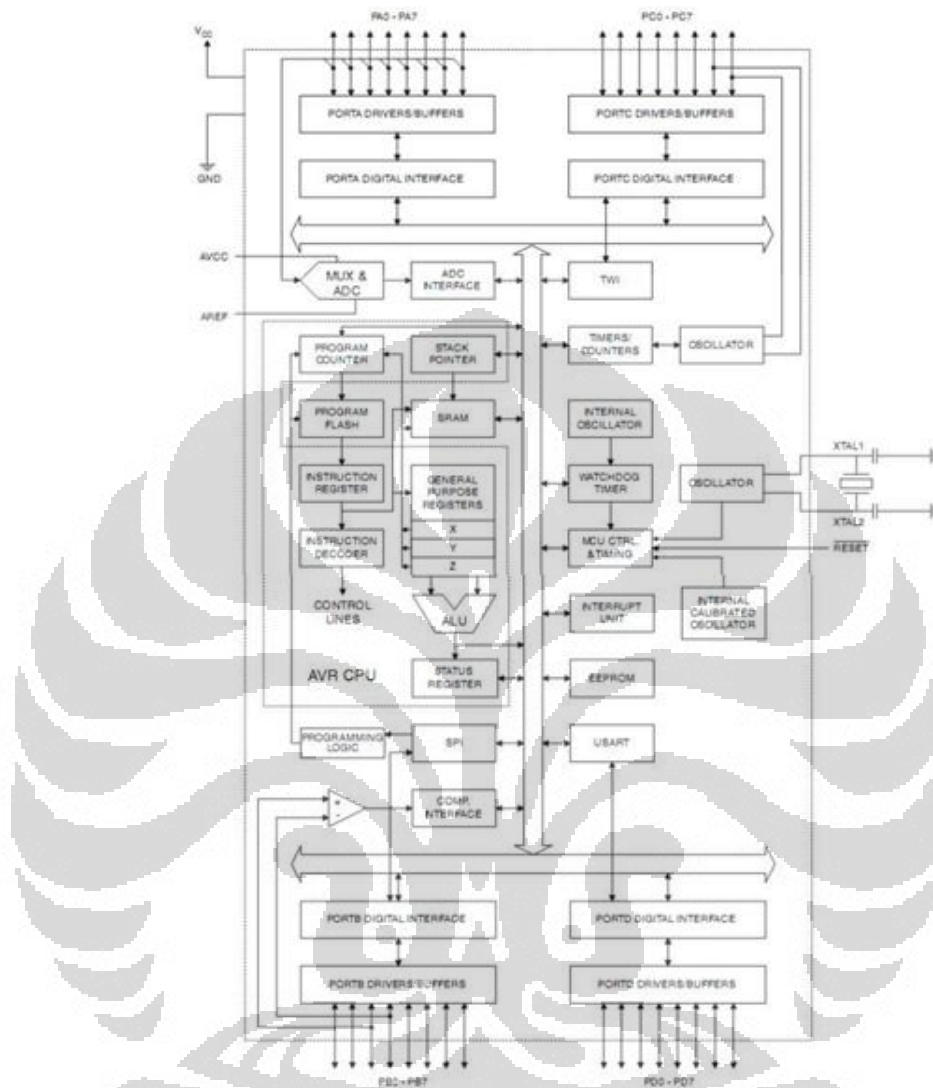
mikrokontroler jenis MCS51, dengan teknologi CISC (*Complex Instruction Set Computing*) membutuhkan 12 siklus clock<sup>[3]</sup>. Secara umum mikrokontroler Atmel AVR dikelompokkan menjadi 4 kelas yaitu ATtiny, AT90Sxx, ATmega, dan AT86RFx. Pada dasarnya yang membedakan masing-masing kelas adalah kapasitas dari memori, peripheral, dan fungsinya. Sedangkan dari segi arsitektur dan instruksi yang digunakan bisa dikatakan sama<sup>[3]</sup>.

#### 2.4.1 Arsitektur ATmega8535

Di dalam sebuah *chip* mikrokontroler AVR ATmega 8535 memiliki beberapa fitur sebagai berikut :

1. Saluran (*channel*) Input/Output sebanyak 32 buah yang dibagi menjadi 4 port, yaitu Port A, Port B, Port C, dan Port D yang masing-masing port memiliki 8 buah saluran I/O.
2. ADC 10 bit sebanyak 10 saluran.
3. Tiga buah *Counter/Timer* dengan kemampuan perbandingan.
4. CPU (*Central Processing Unit*) yang memiliki 32 buah register.
5. 131 instruksi yang hanya membutuhkan 1 siklus *clock*.
6. *Watchdog Timer* dengan *osilator internal*.
7. Tegangan operasi 2,7 V – 5,5 V.
8. Internal SRAM (*Static Random-Access Memory*) sebesar 512 byte.
9. Memory Flash sebesar 8 kB dengan kemampuan *Read While Write*.
10. Unit interupsi internal dan eksternal.
11. Port antarmuka SPI (*Serial Peripheral Interface*).
12. Kecepatan hampir mencapai 16 MPIS pada kristal 16MHz.
13. Internal downloader USB AVR (*In-System Programming* dilengkapi *LED programming indicator*).
14. Tidak membutuhkan power tambahan saat melakukan download program.
15. EEPROM (*Electrically Erasable Programmable Read Only Memory*) sebesar 512 byte yang dapat diprogram saat operasi.
16. Antarmuka komparator analog.
17. Port USART untuk komunikasi serial.

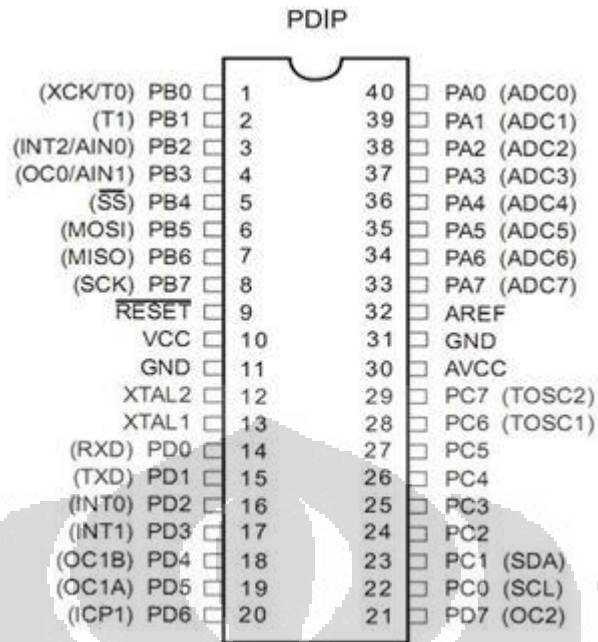
Blok diagram ATmega8535 ditunjukkan pada Gambar 2.8 dibawah ini:



Gambar 2.8 Blok Diagram ATmega8535<sup>[3]</sup>

#### 2.4.2 Konfigurasi Pin ATmega8535

Sebuah Mikrokontroler ATmega8535 memiliki 40 susunan pin seperti yang terdapat pada Gambar 2.9 dibawah ini:



Gambar 2.9 Susunan Pin ATmega8535<sup>[3]</sup>

Berikut ini adalah penjelasan dari susunan pin standar dari ATmega8535 :

1. 1 buah pin VCC yang berfungsi untuk pin masukan catu daya.
2. 1 buah pin GND yang merupakan pin ground.
3. 8 buah pin di Port A (PA0-PA7) yang merupakan pin I/O dua arah dan pin masukan ADC (*Analog to Digital Converter*).
4. 8 buah pin di Port B (PB0-PB7) yang merupakan pin I/O dua arah dan pin fungsi khusus yaitu *Counter/Timer*, komparator analog, dan SPI.
5. 8 buah pin di Port C (PC0-PC7) yang merupakan pin I/O dua arah dan pin fungsi khusus yaitu TWI, komparator analog, dan *Timer Oscillator*.
6. 8 buah pin di Port D (PD0-PD7) yang merupakan pin I/O dua arah dan pin fungsi khusus yaitu komparator analog, interupsi eksternal dan komunikasi serial.
7. 1 buah pin RESET yang digunakan untuk mereset mikrokontroler.
8. 2 buah pin XTALL (XTALL1 dan XTALL2) yang merupakan pin masukan *clock* eksternal.
9. 1 buah pin AVCC yang merupakan pin masukan tegangan untuk ADC.

10. 1 buah pin AREF yang merupakan pin masukan tegangan referensi ADC.

## 2.5 Mekanisme Umpan Balik (*Feedback*)

Secara teori, tidak ada motor yang ideal. Karena adanya panas, tambahan beban, dan keausan yang terjadi pada komutator akan mempengaruhi kecepatan pada motor tersebut. Untuk memaksimalkan kecepatan maupun posisi dari pergerakan motor agar lebih presisi, dibutuhkan sebuah mekanisme umpan balik pada sistem agar pergerakan motor tersebut bisa dikendalikan. Biasanya ditambahkan sebuah sensor untuk mengukur pergerakan dari motor tersebut.

Ada berbagai macam sensor yang digunakan untuk mekanisme umpan-balik pada motor, pada skripsi ini sensor yang digunakan adalah sensor rotai dengan *variable resistor*.

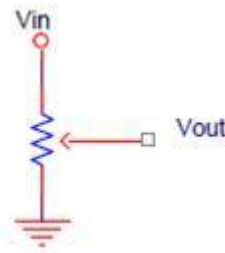
### 2.5.1 *Variable Resistor*

Sebenarnya dasar dari sensor ini adalah sebuah potensiometer yang digunakan sebagai sensor putaran. Potensiometer merupakan sebuah resistor tiga terminal dengan sambungan geser yang membentuk pembagi tegangan yang dapat diatur. Jika pada terminal keluaran dan terminal *ground* diberikan beda potensial yang tetap maka tegangan pada pin keluaran dapat diubah – ubah sesuai putaran dari tuas putarnya. Jika tuas putar ini dihubungkan dengan *shaft* pada motor, maka jika motor bergerak akan menghasilkan tegangan yang berbeda. Tegangan inilah yang digunakan untuk mengidentifikasi posisi sudut pada aktuator yang digunakan. Hubungan antara tegangan dan hambatan dapat dilihat pada persamaan 2.1 yang menjelaskan hukum ohm dibawah ini:

$$V = I \times R \quad (2.1)$$

Gambar 2.10 dibawah ini menunjukkan bahwa dengan besar arus yang konstan, maka perubahan tegangan akan linier terhadap hambatan,

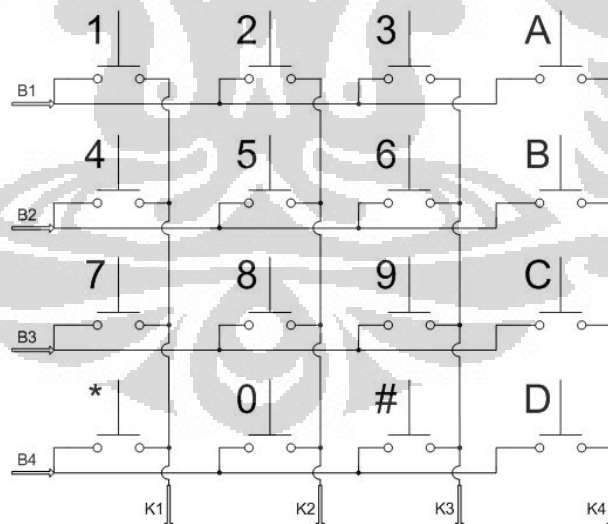
sehingga perubahan sudut putar pada potensiometer akan menyebabkan perubahan tegangan keluaran potensiometer secara linier:



Gambar 2.10 Skematik *Variable Resistor*

## 2.6 Keypad

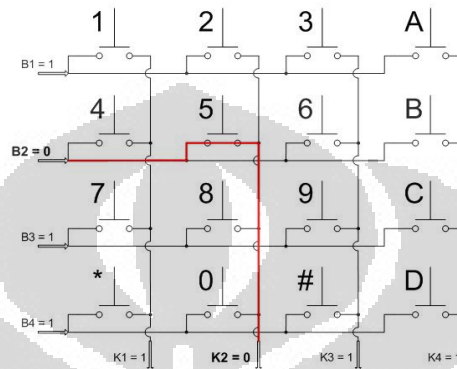
Keypad yang dipakai pada alat ini adalah susunan tombol-tombol yang disusun secara maktriks (baris x kolom) sehingga dapat mengurangi penggunaan pin input. Sebagai contoh, Keypad 4×4 cukup menggunakan 8 pin untuk 16 tombol. Hal tersebut dimungkinkan karena rangkaian tombol disusun secara horizontal membentuk baris dan secara vertikal membentuk kolom seperti yang dapat dilihat pada gambar 2.11 dibawah ini:



Gambar 2.11 Rangkaian *Keypad*<sup>[4]</sup>

Proses pengecekan dari tombol yang dirangkai secara maktriks adalah dengan teknik *scanning*, yaitu proses pengecekan yang dilakukan dengan cara memberikan umpan-data pada satu bagian dan mengecek *feedback* (umpan-balik)

pada bagian yang lain. Dalam hal ini, pemberian umpan-data dilakukan pada bagian baris dan pengecekan umpan-balik pada bagian kolom. Pada saat pemberian umpan-data pada satu baris, maka baris yang lain harus dalam kondisi inversi-nya. Tombol yang ditekan dapat diketahui dengan melihat asal data dan di kolom mana data tersebut terdeteksi:



Gambar 2.12 Contoh Penggunaan Keypad Matriks  $4 \times 4$ <sup>[4]</sup>

Pada gambar 2.12 di atas, tombol yang ditekan adalah tombol “5”. Terlihat bahwa B2 bernilai nol, sedangkan B1, B3, dan B4 adalah satu. Kemudian dengan mengetahui bahwa asal data dari B2, dan umpan-baliknya terdeteksi pada K2, maka dapat disimpulkan bahwa tombol yang ditekan adalah tombol “5”.

## 2.7 Metode Pengendalian *Fuzzy Logic*

Dasar teori fuzzy membahas tentang konsep dasar himpunan fuzzy, yang mencakup pembahasan himpunan fuzzy, operasi logika pada fuzzy dan hukum-hukum pada himpunan fuzzy.

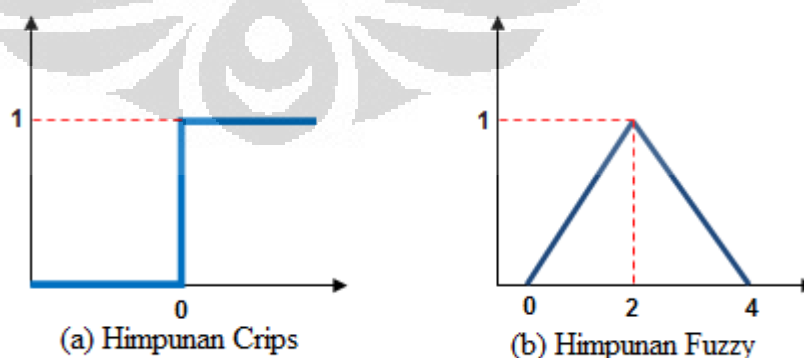
### 2.7.1. Konsep Dasar Himpunan Fuzzy

Pada teori himpunan klasik (Crisp) suatu variabel hanya mempunyai dua kemungkinan, menjadi anggota himpunan atau tidak menjadi anggota himpunan. Dalam teori himpunan crisp ini batasan-batasan antara anggota dan bukan anggota jelas sekali. Batasan nilai error yang dianggap “zero” adalah 0 derajat. Jika dalam pengaplikasian ditemui error sebesar 0,01 derajat saja maka derajat error tersebut sudah bukan

merupakan anggota dari “zero” error. Dalam hal ini himpunan klasik (crisp) hanya mempunyai 2 kemungkinan yang terjadi, yaitu nilai error disebut “positif” apabila nilai error tersebut lebih besar dari 0 dan nilai error yang disebut “negatif” adalah nilai errornya lebih kecil dari 0. Sehingga jika pada percobaan ditemukan nilai error sebesar 0,1 maka tetaplah bukan termasuk bagian dari anggota nilai “zero” error.

Didalam himpunan fuzzy terdapat perbedaan dengan himpunan klasik. Himpunan fuzzy merupakan perluasan dari himpunan klasik, sehingga dalam himpunan fuzzy dapat mempunyai beberapa kemungkinan, bukan hanya 2 kemungkinan seperti didalam himpunan klasik. Misalnya error yang masuk kedalam himpunan “positif small” error adalah “sekitar” 2 derajat, maka jika pada keadaan error sebesar 2,9 dapat dikatakan sebagai anggota dari “positif small” error, bahkan jika ditemukan error sebesar 3,4 derajat masih bisa dikatakan sebagai anggota himpunan “positif small” error.

Untuk persoalan ini himpunan fuzzy membedakan nilai anggota “positif small” error itu dengan menggunakan nilai keanggotaannya, yaitu dari nilai keanggotaan “0” sampai nilai keanggotaan “1”. Nilai atau derajat keanggotaan ini dapat dinyatakan sebagai fungsi keanggotaan. Fungsi keanggotaan didalam himpunan crisp (Gambar 2.13.a) dan fungsi keanggotaan himpunan *fuzzy* (Gambar 2.13.b) dapat digambarkan seperti berikut:



Gambar 2.13 Grafik Fungsi Keanggotaan.



Himpunan pada *fuzzy logic* menggunakan 3 parameter untuk membentuk keanggotaan dalam himpunannya. Parameter-parameter yang digunakan untuk membentuk himpunan *fuzzy logic* adalah:

a. Variabel Linguistic

Variabel yang digunakan pada pengendalian *fuzzy logic* untuk menggantikan variabel kuantitatif yang digunakan pada logika crisp. Variabel linguistik mempunyai nilai yang dinyatakan dengan kata - kata, misalnya untuk variabel linguistik 'nilai error' akan mempunyai nilai berupa nilai linguistik seperti: Zero (Z), Small Positif (SP), Large Positif (LP), Small Negatif (SN) dan Large Negatif (LP).

b. Derajat Keanggotaan

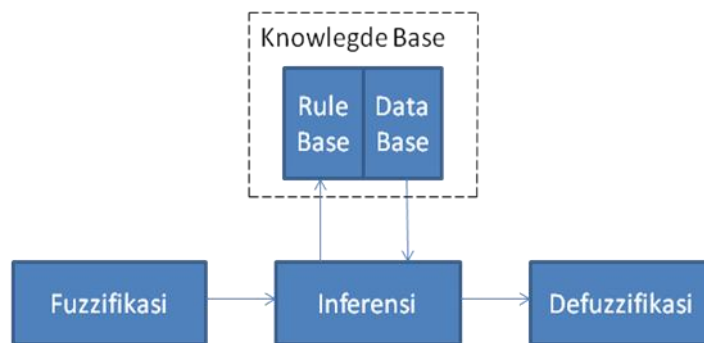
Derajat keanggotaan, yaitu nilai-nilai yang terdapat pada variabel linguistik yang dipetakan ke interval  $[0,1]$ . Nilai pemetaan inilah yang disebut sebagai nilai keanggotaan atau derajat keanggotaan.

c. Fungsi Keanggotaan

Hubungan-hubungan pemetaan pada nilai linguistik dan nilai keanggotaan (dari 0 sampai 1) yang digambarkan kedalam grafik fungsi sehingga didapatkan suatu fungsi. Fungsi inilah yang disebut sebagai fungsi keanggotaan dalam himpunan fuzzy.

### 2.7.2 Struktur Dasar Sistem Fuzzy

Didalam struktur dasar sistem pengendalian pada *fuzzy logic*, terdapat empat komponen atau bagian utama yang sangat penting. Gambar 2.14 menunjukkan struktur dasar dari pengendali *fuzzy logic*, yang terdiri dari Fuzzifikasi, Knowledge Base, Inferensi dan Defuzzifikasi.



Gambar 2.14 Struktur Dasar Pengendali *Fuzzy Logic*.

### 2.7.2.1 *Knowledge Base*

*Knowledge base* mempunyai fungsi penting dalam pengendalian dengan logika *fuzzy* karena semua proses: fuzzifikasi, inferensi dan defuzzifikasi bekerja berdasarkan pengetahuan yang ada pada *knowledge base*. *Knowledge base* dibagi dua, yaitu data base dan rule base. *Data Base* berisi definisi-definisi penting mengenai parameter *fuzzy* seperti himpunan *fuzzy* dengan fungsi keanggotaannya yang telah didefinisikan untuk setiap variabel linguistik yang ada. Pembentukan *data base* meliputi pendefinisian ruang semesta, penentuan banyaknya nilai linguistik yang digunakan untuk setiap variabel linguistik, dan membentuk fungsi keanggotaan. *Rule base* berisi aturan kendali *fuzzy* yang dijalankan untuk mencapai tujuan pengendalian. Tiap rule kendali berupa implikasi dan pernyataan kondisional *If – Then*. Aturan - aturan *If – Then* yang ada dikelompokkan dan disusun kedalam bentuk *Fuzzy Associative Memory* (FAM). FAM ini berupa suatu matriks yang menyatakan input-output sesuai dengan aturan *If – Then* pada basis aturan yang ada. Bentuk matrik dari FAM akan dibahas kemudian. Aturan yang telah dibuat harus dapat mengatasi semua kombinasi-kombinasi input yang mungkin terjadi, dan harus dapat menghasilkan sinyal kendali yang sesuai agar tujuan pengendalian tercapai. Oleh karena itu, maka pembentukan basis aturan ini sangat penting.

### 2.7.2.2 Inferensi

Inferensi adalah proses transformasi dari suatu input dalam domain fuzzy ke suatu output (sinyal kendali) dalam domain *fuzzy*. Proses transformasi pada bagian inferensi membutuhkan aturan-aturan *fuzzy* yang terdapat didalam basis – basis aturan. Blok inferensi menggunakan teknik penalaran untuk menyeleksi basis-basis aturan dan rule dari blok knowledge base. Teknik penalaran yang digunakan adalah teknik penalaran *max – min* yang berfungsi sebagai logika pengambil keputusan. Proses inferensi dengan metode penalaran *max - min* menggunakan inputan besaran sudut posisi sirip roket. Langkah awal dalam proses penalaran *max - min* adalah pembacaan nilai - nilai yang masuk dari sensor yaitu *rotation sensor* lalu ditempatkan dalam grafik keanggotaan yang telah dibuat sebelumnya. Setelah didapatkan hasil penempatannya didalam grafik keanggotaan lalu dilakukan proses penyeleksian dengan mengambil nilai minimum dari derajat keanggotaannya. Setelah didapatkan hasil seleksi nilai minimum, penalaran *max - min* menyeleksi kembali dengan mengambil nilai maximum untuk mendapatkan hasil akhir berupa nilai output inferensi dalam domain fuzzy.

### 2.7.2.3 Defuzzifikasi

Defuzzifikasi merupakan proses pengubahan kembali besaran fuzzy yang disajikan dalam bentuk himpunan – himpunan fuzzy keluaran dengan fungsi keanggotaannya untuk mendapatkan kembali bentuk dari nilai tegasnya (*crisp*). Hal ini dilakukan karena dalam aplikasinya yang dibutuhkan adalah nilai tegasnya. Suatu nilai fuzzy output yang berasal dari proses inferensi dimasukkan kedalam suatu rumus yang dinamakan Centre of Gravity (COG) untuk mendapatkan hasil yang disebut *crisp output*. *Crisp output* adalah suatu nilai analog yang dibutuhkan untuk menolah data pada sistem yang telah dirancang.

## 2.8 Teori Karakteristik Respon

Karakteristik respon adalah ciri-ciri khusus perilaku dinamik (spesifikasi performansi) respon output sistem yang muncul akibat diberikannya suatu sinyal masukan tertentu yang khas bentuknya (disebut sebagai sinyal uji)<sup>[16]</sup>. Karakteristik respon sistem dapat diklasifikasi menjadi dua macam, yaitu karakteristik respon waktu dan karakteristik respon frekuensi. Tetapi pada skripsi ini, respon sistem yang digunakan adalah respon sistem waktu (time response).

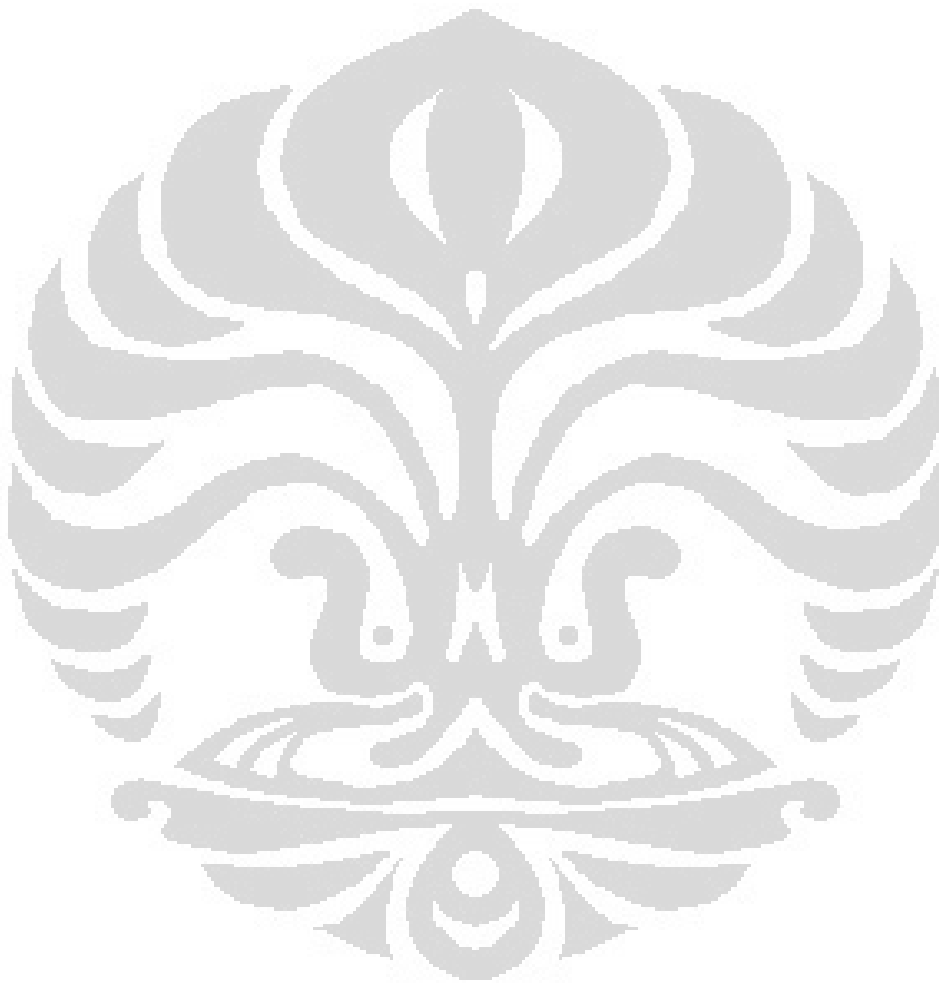
Karakteristik respon waktu adalah karakteristik respon yang spesifikasi performansinya didasari pada pengamatan bentuk respon output sistem terhadap waktu. Secara umum spesifikasi performansi respon waktu dapat dibagi menjadi dua tahapan pengamatan, yaitu:

- Spesifikasi Respon *Transient*, adalah spesifikasi respon sistem yang diamati mulai saat terjadinya perubahan sinyal input sampai respon masuk dalam keadaan *steady state* (tunak). Tolok ukur yang digunakan untuk mengukur kualitas respon transient ini antara lain *rise time*, *delay time*, *peak time*, *settling time* dan *%overshoot*.
- Spesifikasi Respon *Steady State*, adalah spesifikasi respon sistem yang diamati mulai saat respon masuk dalam keadaan *steady state* sampai waktu tak terbatas. Tolok ukur yang digunakan untuk mengukur kualitas respon *steady state* ini antara lain *%error steady state* baik untuk error posisi, error kecepatan maupun error percepatan.

Beberapa macam ukuran kualitas respon yang lazim digunakan adalah sebagai berikut:

- *Rise Time*, adalah ukuran waktu yang menyatakan keberadaan suatu respon, yang diukur mulai respon 5% sampai dengan respon 95% dari keadaan *steady state*.
- *Settling Time*, adalah ukuran waktu yang menyatakan respon telah masuk sekitar 5% dari keadaan *steady state*.
- *Overshoot*, adalah nilai relatif yang menyatakan perbandingan nilai maksimum respon yang melebihi nilai *steady state* berbanding dengan nilai *steady state*.

- *Peak Time*, adalah ukuran waktu diukur mulai  $t=0$  sampai dengan respon mencapai puncak pertama kali atau yang paling besar.
- *Error Steady State*, adalah perbedaan nilai output yang didapat dengan nilai output yang diinginkan. Analisa *error steady state* dilakukan ketika sistem sudah dalam keadaan stabil.



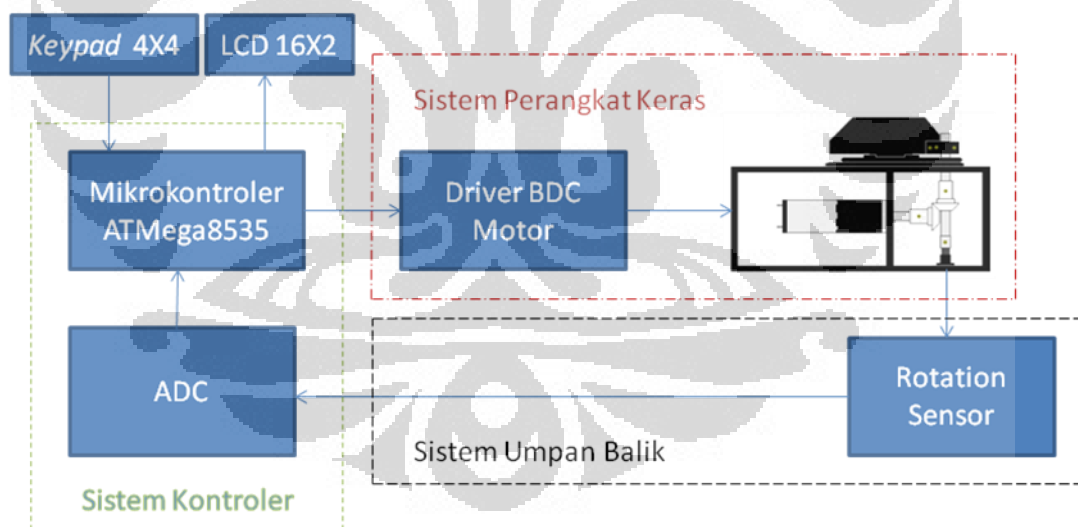
## BAB 3

### PERANCANGAN DAN IMPLEMENTASI SISTEM

Pada bab ini akan dibahas tentang gambaran sistem yang akan dibuat. Segala perancangan yang meliputi perancangan perangkat keras seperti pembuatan mekanik dari aktuator pergerakan sirip, jenis motor, roda gigi dan komponen elektronika yang akan digunakan maupun perancangan perangkat lunak seperti metode pengendalian dan bahasa pemrograman yang digunakan akan dibahas pada bab ini.

#### 3.1 Deskripsi Sistem

Sistem yang akan dibangun pada makalah ini adalah sebuah sistem pengendali posisi sirip pada roket yang menggunakan sebuah motor DC 24 volt yang terhubung dengan planetary gear sebagai penggerakannya. Keseluruhan blok diagram dari sistem yang dirancang dapat dilihat pada gambar 3.1 berikut:



Gambar 3.1 Blok Diagram Sistem

Pada blok diagram diatas terlihat keseluruhan sistem yang akan dibangun. Sebuah CPU akan menjadi pemberi masukan yang akan dihubungkan ke Mikrokontroler ATmega8535 dengan menggunakan koneksi serial USB. Mikrokontroler yang sudah diprogram akan memberi input kepada motor yang akan diterjemahkan oleh driver yang menggunakan IC (*Integrated Circuit*) L298

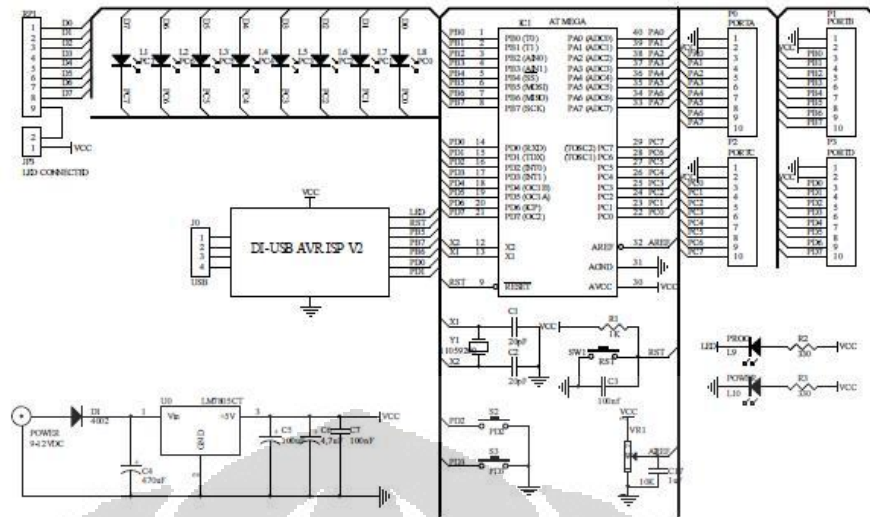
yang akan mengatur keluaran tegangan yang dihubungkan ke kaki – kaki motor DC sehingga pergerakan dari motor DC tersebut dapat dikendalikan sesuai dengan keinginan. Pergerakan dari motor DC tersebut akan diketahui dengan adanya sebuah *rotation sensor* yang dipasang pada mekanik. Data analog yang didapat dari *rotation sensor* akan dikirim kembali menuju mikrokontroler dengan diubah terlebih dahulu menjadi data digital dengan menggunakan ADC (*Analog to Digital Converter*) yang sudah terdapat pada ATmega8535 tersebut. Data tersebut akan digunakan sebagai umpan balik agar pengendalian dari motor tersebut bisa lebih akurat.

### 3.2 Perancangan Perangkat Keras

Berikut ini yang akan dibahas adalah tahap perancangan dan pembuatan dari perangkat keras yang meliputi tentang rancangan *minimum system* dari mikrokontroler, jenis motor yang akan dipakai, jenis sensor umpan balik, serta perancangan mekanik secara umum.

#### 3.2.1 *Minimum System* DI-Smart AVR

DI-Super Smart AVR adalah sebuah rangkaian *minimum system* buatan Depok Instrumen yang merupakan rangkaian pengendali yang menggunakan sebuah mikrokontroler ATmega8535. Dalam rangkaian ini terdapat juga sebuah modul *downloader* berbasis USB yang juga bisa digunakan sebagai antar muka antara mikrokontroler dengan komputer dengan hanya mengubah posisi *jumper* pada rangkaian *minimum system* tersebut. Gambar rangkaian *minimum system* DI-Smart AVR ini dapat dilihat pada gambar 3.2 berikut:



Gambar 3.2 Rangkaian *Minimum System* DI-Smart AVR<sup>[5]</sup>

Pada rangkaian ini terdapat 40 pin keluaran dari mikrokontroler ATmega8535 yang dibagi menjadi 4 port. Pada port A akan digunakan sebagai pin masukan untuk data ADC yang dikirim dari rangkaian *rotation sensor* yang akan digunakan sebagai umpan balik dari sistem. Lalu pada port B akan digunakan sebagai port masukan dari *keypad* untuk menentukan posisi sudut dari aktuator sirip roket yang diinginkan. Port C digunakan sebagai pin keluaran menuju *driver* motor DC yang akan mengendalikan pergerakan motor DC tersebut. Lalu port D akan digunakan sebagai port keluaran untuk tampilan pada LCD. Berikut adalah gambar modul minimum system DI-Smart AVR yang digunakan:

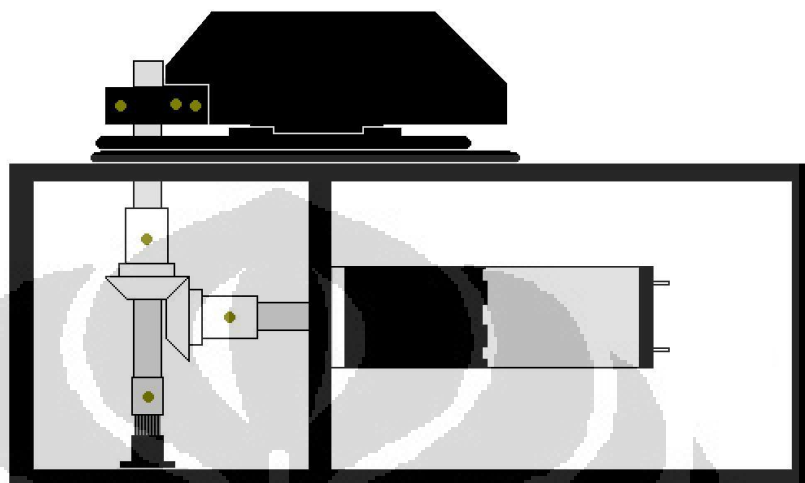


Gambar 3.3 DI-Smart AVR<sup>[5]</sup>



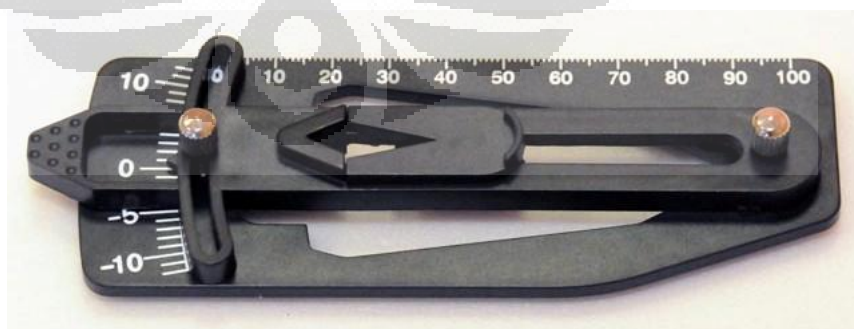
### 3.2.2 Perancangan Aktuator Kendali Sirip

Rancang bangun mekanik dari aktuator sirip ini dapat dilihat pada gambar 3.4 dibawah ini :



Gambar 3.4 Desain Mekanik Aktuator Pergerakan Sirip

Mekanik dibuat dari bahan plat besi setebal 3 mm yang dilipat. Lalu siripnya menggunakan bahan fiber. Sebagai penransmisi daya digunakan sepasang roda gig jenis kerucut lurus dengan bahan plastik dihubungkan dengan besi *shaft* setebal 6,5 mm dengan panjang 8 mm untuk poros sirip tersebut. Sebagai indikator pergerakan digunakan sebuah pitch gauge seperti yang dapat dilihat pada gambar 3.5 berikut:



Gambar 3.5 *Pitch Gauge*<sup>[9]</sup>

Alat ini dihubungkan dengan besi shaft yang terhubung dengan motor dengan bantuan roda gigi kerucut. Jadi setiap pergerakan motor akan bisa dilihat pada pitch gauge ini. Pitch gauge ini memiliki rentang indikator dari -10 hingga 10 derajat.

### 3.2.3 Motor DC yang digunakan

Dalam perancangan aktuator sirip ini telah digunakan sebuah *brushed DC motor* buatan MAXON yang akan dihubungkan dengan sebuah planetary gear yang juga buatan MAXON dengan reduksi putaran 1:246. Gambar 3.6 berikut adalah gambar dari motor yang digunakan:



Gambar 3.6 MAXON *Brushed DC Motor*<sup>[10]</sup>

*Brushed DC motor* ini produksi MAXON jenis A-max dengan kode 242472 yang memiliki spesifikasi sebagai berikut:

Tabel 3.1 Spesifikasi MAXON *Brushed DC Motor*<sup>[10]</sup>

Spesifikasi MAXON DC Motor	
Tegangan Nominal	24 Volt
Daya yang dihasilkan	15 Watt
Kecepatan tanpa beban	5860 rpm
Diameter	32 mm
Kecepatan Konstan	250 rpm / V
Torsi Konstan	38,3 mNm / A
Panjang badan motor	62,9 mm

### 3.2.4 Planetary Gear yang digunakan.

Motor MAXON tersebut dihubungkan dengan sebuah *planetary gear* buatan MAXON juga dengan penampakan dapat dilihat pada gambar 3.7 dan spesifikasinya dapat dilihat pada tabel 3.2 sebagai berikut:



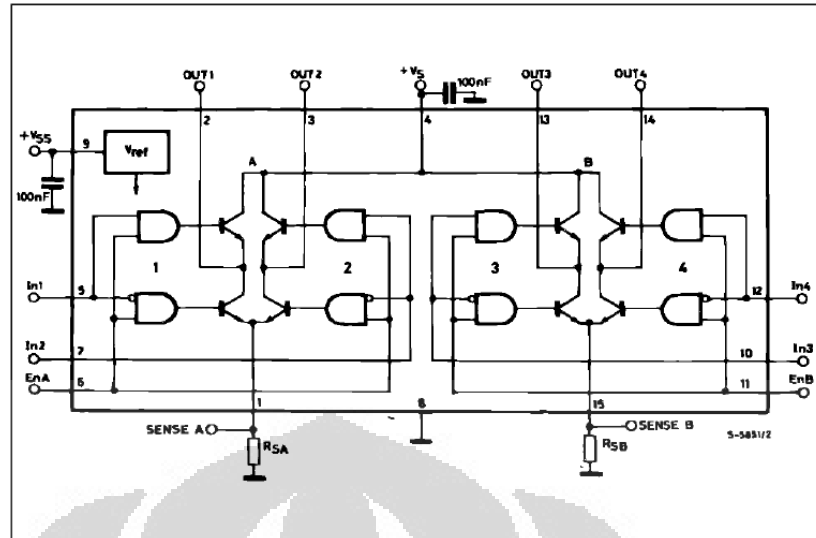
Gambar 3.7 MAXON *Planetary Gear*<sup>[11]</sup>

Tabel 3.2 Spesifikasi MAXON *Planetary Gear*<sup>[11]</sup>

Spesifikasi MAXON Planetary Gear	
Panjang Badan	49,8 mm
Diameter	32 mm
Panjang Shaft	21 mm
Diameter Shaft	4 mm
Reduksi Kecepatan	246 : 1
Reduksi Mutlak	421824 : 1715

### 3.2.5 Driver Motor DC yang digunakan

Untuk menggerakkan motor DC diperlukan sebuah rangkaian driver yang digunakan untuk mengendalikan arah putaran maupun kecepatan putaran. Rangkaian driver yang digunakan merupakan rangkaian buatan Depok Instrumen yang menggunakan IC L298. IC L298N ini memiliki konfigurasi 2 buah H-bridge driver motor dengan 4 input dan 4 output untuk mengendalikan arah putaran motor. IC driver motor ini dapat mengendalikan motor DC dari inputan yang berlogika *high* atau *low*. Gambar 3.8 dibawah ini adalah gambar dari rangkaian driver motor menggunakan IC L298 :



Gambar 3.8 Rangkaian IC L298<sup>[6]</sup>

Rangkaian driver ini berfungsi untuk menggerakkan motor sesuai dengan input yang diterimanya. Pin input dari driver dihubungkan ke mikrokontroler pada portd.6 dan portd.7 yang akan memberi perintah kepada driver tersebut untuk kemudian dikirim melalui output driver tersebut yang dihubungkan ke kaki - kaki motor DC yang digunakan. Driver motor DI-MDCD4A yang digunakan dapat dilihat pada gambar 3.9 dibawah ini:

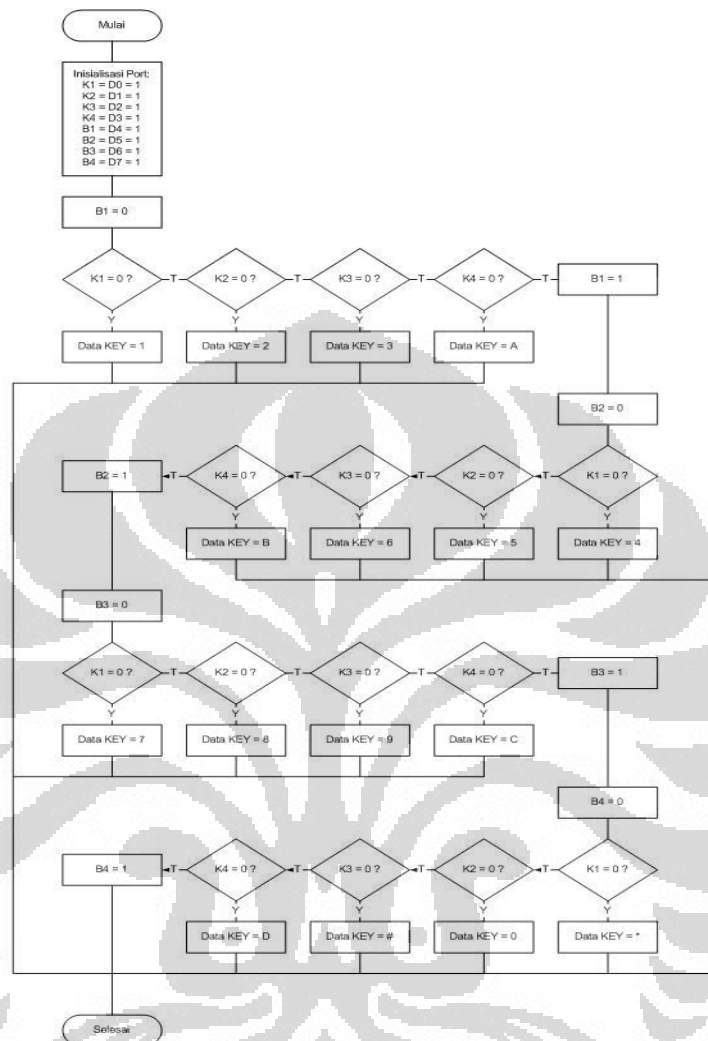


Gambar 3.9 DI-MDCD4A<sup>[8]</sup>

### 3.4 Konfigurasi Keypad

Pada pembuatan penelitian ini digunakan bantuan *keypad* yang digunakan untuk memberi masukan *setpoint* sudut yang diinginkan. Proses pengujian *keypad*

dilakukan dengan metode *scanning* yang *flowchart*nya dapat dilihat pada gambar 3.10 dibawah ini:



Gambar 3.10 *Flowchart* Pengecekan *Keypad* dengan Metode *Scanning*<sup>[4]</sup>

Proses pengecekan dari keypad yang dirangkai secara matriks adalah dengan teknik *scanning* yaitu dengan cara memberikan umpan-data pada satu bagian dan mengecek umpan-balik nya pada bagian yang lain. Dalam hal ini, pemberian umpan-data dilakukan pada bagian baris dan pengecekan umpan-balik pada bagian kolom. Ketika satu tombol ditekan maka akan diketahui tombol mana yang ditekan dengan melihat asal baris dan di kolom mana data tersebut terdeteksi. Berikut gambar 3.11 yang merupakan listing program dengan menggunakan bahasa *basic compiler* yang digunakan untuk melakukan pembacaan tombol pada *keypad*:

```

Getkey = "$"
B0 = 0
If K0 = 0 Then
  Getkey = "1"
Elseif K1 = 0 Then
  Getkey = "2"
Elseif K2 = 0 Then
  Getkey = "3"
Elseif K3 = 0 Then
  Getkey = "A"
End If

```

'pemberian umpan-data pada baris B0  
'jika ada umpan balik pada kolom K0  
'maka data yang keluar adalah angka "1"  
'jika ada umpan balik pada kolom K1  
'maka data yang keluar adalah angka "2"  
'jika ada umpan balik pada kolom K2  
'maka data yang keluar adalah angka "3"  
'dan seterusnya

Gambar 3.11 Tampilan *Listing Program* untuk Scanning data *Keypad*

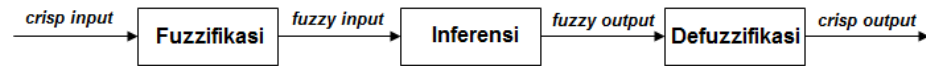
Pada tabel 3.3 berikut akan ditampilkan keseluruhan hasil data *scanning* yang telah dilakukan pada *keypad* matriks 4x4 yang digunakan.

Tabel 3.3 Data Hasil Metode Scanning Pada Keypad Matriks 4x4

Baris B	Kolom K	Data Keluaran
B0	K0	"1"
	K1	"2"
	K2	"3"
	K3	"A"
B1	K0	"4"
	K1	"5"
	K2	"6"
	K3	"B"
B2	K0	"7"
	K1	"8"
	K2	"9"
	K3	"C"
B3	K0	"*"
	K1	"0"
	K2	"#"
	K3	"D"

### 3.4 Perancangan sistem Fuzzy Logic Control

Fuzzy logic control memiliki tiga bagian utama dalam pembuatan struktur dasar sistem kendali *fuzzy*, yaitu: Fuzzifikasi, Inferensi dan Defuzzifikasi. Berikut ini adalah penjelasan dari masing – masing strukturnya:



Gambar 3.12 Flowchart Struktur *Fuzzy Logic*

Seperti terlihat pada gambar 3.12, pada sistem fuzzy logic terdapat 3 struktur dasar metode yang digunakan. Awalnya nilai tegas akan digunakan sebagai masukan pada tahap fuzzifikasi, yang akan dicari derajat keanggotaannya pada proses fuzzifikasi dan hasil keluarannya akan menghasilkan nilai *fuzzy*. Pada tahap inferensi inputan yang diberikan berupa nilai *fuzzy* dari hasil fuzzifikasi dan akan dihasilkan *output* berupa nilai *fuzzy* yang akan diubah menjadi nilai *crisp* kembali pada proses defuzzifikasi. Masing – masing tahap pada proses *fuzzy logic* akan dibahas lebih jelas pada bagian berikutnya.

#### 3.4.1 Fuzzifikasi

Pada fuzzifikasi ini, *input* error dan  $\Delta$ error (de) masing-masing dibagi dalam 3 fungsi keanggotaan sebagai berikut:

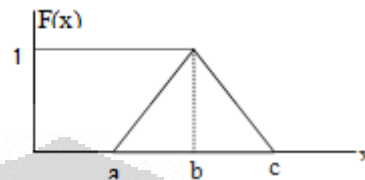
- Error : N (Negatif), Z (Zero), P (Positif)
- $\Delta$ error : N (Negatif), Z (Zero), P (Positif)

Sedangkan tipe keanggotaan fuzzy yang digunakan dalam skripsi ada dua, yaitu segitiga dan trapesium. Berikut ini persamaan dari segitiga dan trapesium:

Tipe Keanggotaan segitiga:

$$F(x; a, b, c) = \begin{cases} 0 & x < a \\ (x-a) / (b-a) & a \leq x < b \\ (c-x) / (c-d) & b \leq x < c \\ 0 & x \geq c \end{cases}$$

Dengan nilai  $x$ ,  $a$ ,  $b$ , dan  $c$  diambil dari gambar berikut:



Gambar 3.13 Tipe keanggotaan segitiga

Tipe keanggotaan trapesium:

$$F(x; a, b, c, d) = \begin{cases} 0 & x < a \\ (x-a) / (b-a) & a \leq x < b \\ 1 & b \leq x < c \\ (c-x) / (c-d) & c \leq x < d \\ 0 & x \geq d \end{cases}$$

Dengan nilai  $x$ ,  $a$ ,  $b$ ,  $c$ , dan  $d$  didapat dari gambar 3.14 berikut:



Gambar 3.14 Tipe Keanggotaan Trapesium

Nilai  $x$  ini merupakan nilai error atau  $\Delta$ error yang dimasukkan dan  $F(x)$  adalah nilai keanggotaannya. Berikut ini adalah nilai keanggotaan himpunan fuzzy, untuk himpunan Error ( $er$ ):

- Negatif (N) :  $(x; -4; -2, 0)$
- Zero (Z) :  $(x; -2; 0; 2)$
- Positif (P) :  $(x; 0; 2; 4)$

Sedangkan himpunan fuzzy untuk  $\Delta$ error ( $der$ ) adalah:



- Negatif (N) : (x; -10; -5, 0)
- Zero (Z) : (x; -5; 0; 5)
- Positif (P) : (x; 0; 5; 10)

### 3.4.2 Inferensi

Dari membership function yang sudah dibuat, maka dapat dirumuskan rule fuzzifikasinya sesuai kondisi yang ada. Karena *membership function* inputnya masing-masing memiliki tiga buah member untuk masukan *error* dan  $\Delta error$ . Sehingga didapatkan 9 rule kemungkinan yang dikombinasikan dengan 3 buah *membership output*. Berikut rule yang dirancang:

**Tabel 3.4 Rule Base Perancangan Sistem**

<b>E</b>	<b>Negatif</b>	<b>Zero</b>	<b>Positif</b>
<b>de</b>			
<b>Negatif</b>	Positif	Positif	Positif
<b>Zero</b>	Negatif	Zero	Positif
<b>Positif</b>	Negatif	Negatif	Positif

Berdasarkan tabel 3.4 diatas akan dihasilkan aturan – aturan yang dapat mengatasi semua kombinasi-kombinasi input yang mungkin terjadi, dan harus dapat menghasilkan sinyal kendali yang sesuai agar tujuan pengendalian tercapai. Karena kombinasi dari input masing – masing tiga, maka didapatkan 9 aturan dengan metode penalaran *If – Then* sebagai berikut:

1. If ( $e < 0$ ) AND ( $de < 0$ ) then Output Positif
2. If ( $e = 0$ ) AND ( $de < 0$ ) then Output Positif
3. If ( $e > 0$ ) AND ( $de < 0$ ) then Output Positif
4. If ( $e < 0$ ) AND ( $de = 0$ ) then Output Negatif
5. If ( $e = 0$ ) AND ( $de = 0$ ) then Output Zero
6. If ( $e > 0$ ) AND ( $de = 0$ ) then Output Positif

7. If ( $e < 0$ ) AND ( $de > 0$ ) then Output Negatif
8. If ( $e = 0$ ) AND ( $de > 0$ ) then Output Negatif
9. If ( $e > 0$ ) AND ( $de > 0$ ) then Output Positif

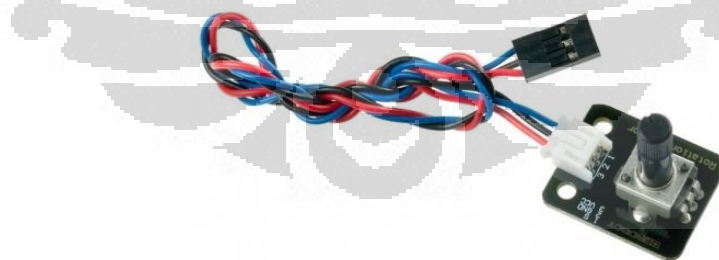
### 3.4.3 Defuzzifikasi

Defuzzifikasi yang digunakan dalam sistem ini menggunakan penalaran fuzzy metode mamdani, *center of gravity* (COG) dengan rumusan dasar seperti pada persamaan 3.1 berikut:

$$y = \frac{\sum_j^R = 1 c^j \int \mu_{\hat{v}} J(u) du}{\sum_j^R = 1 \int \mu_{\hat{v}} J(u) du} \quad (3.1)$$

### 3.5 Perancangan Mekanisme Umpan Balik

Pada pembuatan aktuator posisi sirip ini digunakan sebuah sensor rotasi sebagai mekanisme umpan balik dari sistem ini. Sensor ini berbasis variable resistor dengan tiga buah pin yaitu pin vcc, pin output data dan pin ground. Sensor ini dapat berputar hingga 300 derajat jenis single turn buatan DFRobot. Prinsip kerja sensor ini sama dengan prinsip kerja dari potensiometer, nilai resistansinya akan berubah secara linier ketika diputar. Setiap perubahan sudut dari aktuator sirip akan diketahui dengan perubahan tegangan yang dihasilkan oleh *rotation sensor* ini. Berikut ini adalah gambar dari DFRobot *rotation sensor*:



**Gambar 3.15** DFRobot *Rotation Sensor v.1*<sup>[7]</sup>

## BAB 4

### PENGUJIAN DAN ANALISIS HASIL

Setelah melakukan perencanaan dan pembuatan sistem, maka langkah selanjutnya yang akan dibahas pada bab ini adalah melakukan pengujian dan analisa terhadap sistem yang telah dibuat baik perangkat keras maupun perangkat lunaknya untuk mengetahui apakah sistem yang digunakan sudah bekerja sesuai dengan yang diinginkan. Pengujian perangkat keras terdiri dari pengujian rangkaian driver motor dc, pengujian keypad, dan pengujian terhadap rotation sensor yang digunakan. Perangkat lunak yang diuji meliputi pengujian terhadap metode pengendalian yang digunakan, yang dalam skripsi ini menggunakan metode pengendalian fuzzy logic untuk mengatur pergerakan dari aktuator sirip.

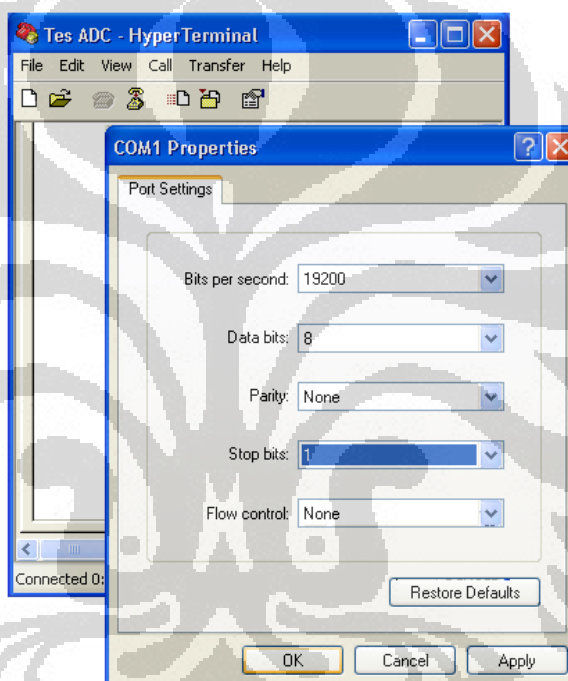
#### 4.1 Alat Ukur Yang Digunakan

Pada skripsi ini telah digunakan alat bantu untuk mengukur besaran tegangan, arus maupun tahanan yang berada pada rangkaian elektronik dengan menggunakan sebuah multimeter digital produksi Heles dengan tipe UX 37 seperti pada gambar 4.1 dibawah ini:



Gambar 4.1 Multimeter HELES UX37

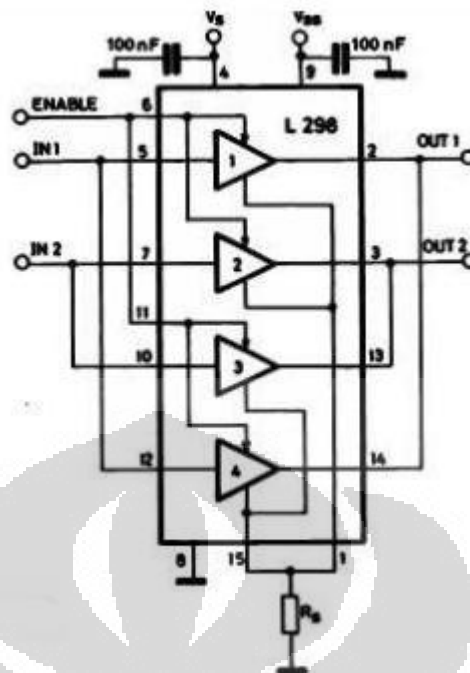
Sebuah aplikasi bernama *hyperterminal* digunakan untuk membaca nilai tegangan yang diinputkan oleh *rotation sensor* yang nilainya telah diubah sebelumnya oleh mikrokontroler menjadi nilai *digital*. Koneksi antara *hyperterminal* dengan mikrokontroler dihubungkan dengan koneksi USB. yang dihubungkan secara serial dengan mikrokontroler dengan bantuan USB. Data yang diterima oleh *hyperterminal* merupakan data dari pergerakan aktuator sirip yang akan diolah oleh metode pengendalian *fuzzy logic*. Gambar 4.2 dibawah ini merupakan tampilan dari aplikasi *hyperterminal* yang digunakan:



Gambar 4.2 Tampilan aplikasi *hyperterminal*

## 4.2 Pengujian Rangkaian Driver Motor DC

Pengujian rangkaian *driver* dengan beban untuk mengetahui apakah rangkaian *driver* digunakan dapat bekerja dengan baik. Rangkaian yang digunakan sebagai driver motor dc adalah rangkaian yang menggunakan IC L298N yang skematiknya dapat dilihat pada gambar 4.3 berikut:



Gambar 4.3 Rangkaian IC L298N<sup>[6]</sup>

Pengujian dilakukan dengan cara memberi kombinasi masukan dari adaptor sebesar 5 volt sebagai pengganti dari masukan dari mikrokontroler. Dengan variasi input yang diberikan, maka keluaran dari IC L298N juga akan berubah – ubah. Tabel 4.1 berikut akan memberikan contoh variasi input yang diberikan.

Tabel 4.1 Pengujian Rangkaian Driver Motor L298N Tanpa Beban

Enable	INPUT 1	INPUT 2	OUTPUT 1	OUTPUT 2
$V_{en} = \text{Low}$	Low	Low	Low	Low
	Low	High	Low	Low
	High	Low	Low	Low
	High	High	Low	Low
$V_{en} = \text{High}$	Low	Low	Low	Low
	Low	High	Low	High
	High	Low	High	Low
	High	High	High	High

Pada tabel terlihat bahwa ketika kaki *enable* (*en*) pada rangkaian dalam kondisi *low*, maka segala kombinasi yang dihasilkan oleh input tetap akan membuat kedua kaki output dalam keadaan *low*. Sedangkan ketika kaki *enable* dikondisikan *high* maka ketika input 1 diberikan kondisi *low* dan input 2 diberikan kondisi *high* maka kaki di output 2 dalam kondisi *high*. Dan sebaliknya jika input 1 dalam kondisi *high* sedangkan input 2 dalam kondisi *low* maka kaki di output 1 akan berada pada kondisi *high*. Ketika kedua masukan dalam kondisi *low* atau *high*, maka keluaran juga akan mengikuti logika dari masukan tersebut.

Ketika pada output dari driver motor L298N tersebut diberikan beban berupa motor dc, maka dapat dihasilkan data seperti yang terlihat pada tabel 4.2 dibawah. Ketika kaki *enable* diberikan kondisi 0 Volt maka segala kombinasi yang dihasilkan oleh kedua masukan (input 1 dan input 2) tetap akan membuat kondisi motor tidak bergerak. Sedangkan ketika *enable* diberikan tegangan dari adaptor sebesar 5 Volt, maka kombinasi input akan mempengaruhi pergerakan dari motor dc tersebut. Data secara lengkapnya dapat dilihat pada tabel dibawah ini.

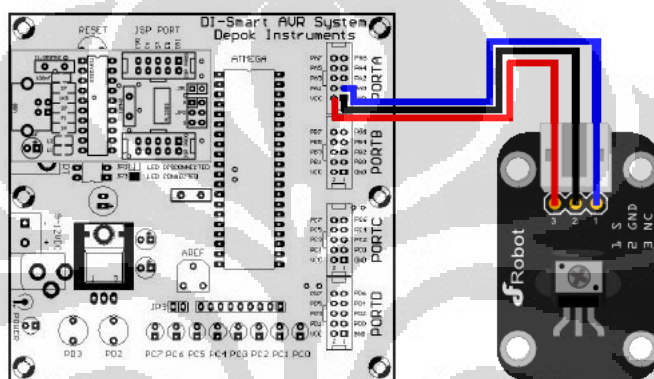
Tabel 4.2 Pengujian Driver Motor DC Dengan Beban Motor DC

Enable	INPUT 1	INPUT 2	Kondisi Motor
$V_{en} = 0 \text{ Volt}$	0 Volt	0 Volt	Diam
	0 Volt	5 Volt	Diam
	5 Volt	0 Volt	Diam
	5 Volt	5 Volt	Diam
$V_{en} = 5 \text{ Volt}$	0 Volt	0 Volt	Diam
	0 Volt	5 Volt	Putar Kiri
	5 Volt	0 Volt	Putar Kanan
	5 Volt	5 Volt	Diam

### 4.3 Pengujian Mekanisme Umpan Balik

Pada skripsi ini digunakan sebuah rotation sensor berbasis variable resistor sebagai mekanisme umpan-balik dari sistem. Sensor ini dapat berputar hingga  $300^\circ$  dan akan menghasilkan tegangan yang bervariasi ketika *resistor variabelnya*

diputar sesuai dengan besar sudut yang ingin diketahui besar tegangannya. Rotation sensor ini akan dihubungkan dengan PORT A pada mikrokontroler ATmega8535 yang dikoneksikan dengan komunikasi serial. Hasil dari pengukuran *rotation sensor* akan dilihat melalui aplikasi *hyperterminal*. Data analog yang keluar dari *rotation sensor* akan dikonversikan oleh mikrokontroler sebelum dikirim ke *hyperterminal* melalui PINA.0 pada mikrokontroler ATmega8535. Berikut ini adalah konfigurasi koneksi dari rotation sensor ke mikrokontroler:



Gambar 4.4 Konfigurasi Hubungan *Rotation Sensor* ke Mikrokontroler<sup>[5]</sup>

Seperti terlihat pada gambar 4.4 diatas bahwa keluaran data tegangan dari rotation sensor dihubungkan ke PINA.0 dari mikrokontroler yang akan mengkonversi nilai tersebut ke nilai digital agar dapat diolah. Tegangan referensi yang diberikan dikaki sensor sebesar 2,5 Volt. Gambar 4.5 berikut adalah potongan program untuk mengambil data ADC yang dikirimkan oleh rotation sensor:

```
$regfile = "m8535.dat"
$crystal = 11059200
$baud = 19200

Config ADC = Single , Prescaler = Auto
Start ADC

Dim Nilai_adc As Word

Nilai_adc = Getadc(0)
Print "Nilai ADC = " ; Nilai_adc

End
```

Gambar 4.5 Listing Program *Basic Compiler* Untuk Menampilkan Nilai ADC

Setelah program di download pada mikrokontroler maka setiap tegangan yang masuk pada PINA.0 akan ditampilkan ke *hyperterminal* dengan bentuk nilai digital karena telah dikonversi sebelumnya oleh mikrokontroler. Tetapi nilai tegangan awal yang keluar pada adc sangat tidak stabil, maka diberikan kapasitor sebesar 100  $\mu$ F sebagai kapasitor *bypass* untuk meredam *noise* sekaligus menstabilkan tegangan yang keluar. Besarnya ADC pada sudut aktuator akan ditampilkan pada tabel 4.3 berikut ini.

Tabel 4.3 Data ADC Pada Setiap Perubahan Sudut

Sudut	Nilai ADC	$V_{ref} = 2,5 \text{ Volt}$		% error	Sudut	Nilai ADC	$V_{ref} = 5,0 \text{ Volt}$		% error
		$V_{ukur}$	$V_{hitung}$				$V_{ukur}$	$V_{hitung}$	
10°	101	0,24 V	0,24 V	0	10°	159	0,78 V	0,77 V	1,29
9°	123	0,30 V	0,30 V	0	9°	166	0,82 V	0,81 V	1,23
8°	145	0,35 V	0,35 V	0	8°	174	0,86 V	0,85 V	1,17
7°	161	0,39 V	0,39 V	0	7°	188	0,92 V	0,91 V	1,09
6°	180	0,43 V	0,43 V	0	6°	198	0,98 V	0,96 V	2,08
5°	195	0,47 V	0,47 V	0	5°	207	1,02 V	1,01 V	0,99
4°	212	0,52 V	0,51 V	1,96	4°	218	1,07 V	1,06 V	0,94
3°	235	0,56 V	0,56 V	0	3°	228	1,12 V	1,11 V	0,90
2°	248	0,60 V	0,60 V	0	2°	239	1,18 V	1,16 V	1,72
1°	272	0,66 V	0,66 V	0	1°	250	1,23 V	1,22 V	0,81
0°	291	0,71 V	0,71 V	0	0°	255	1,25 V	1,24 V	0,80
-1°	303	0,73 V	0,73 V	0	-1°	263	1,30 V	1,28 V	1,56
-2°	320	0,78 V	0,77 V	1,29	-2°	276	1,36 V	1,34 V	1,49
-3°	340	0,83 V	0,83 V	0	-3°	285	1,41 V	1,39 V	1,43
-4°	364	0,88 V	0,88 V	0	-4°	294	1,45 V	1,43 V	1,39
-5°	380	0,93 V	0,93 V	0	-5°	301	1,48 V	1,47 V	0,68
-6°	400	0,97 V	0,97 V	0	-6°	310	1,53 V	1,51 V	1,32
-7°	422	1,02 V	1,02 V	0	-7°	321	1,59 V	1,56 V	1,92
-8°	444	1,08 V	1,08 V	0	-8°	339	1,67 V	1,65 V	1,21



-9°	466	1,13 V	1,13 V	0	-9°	348	1,72 V	1,70 V	1,17
-10°	478	1,16 V	1,16 V	0	-10°	354	1,75 V	1,73 V	1,15

Tegangan keluaran dari *rotation sensor* didapatkan dari pengukuran pada pin kaki 1 pada sensor dengan perbandingan tegangan yang diberikan oleh tegangan referensi.  $V_{ref}$  yang dipakai sebesar 2,5 Volt karena seperti terlihat pada tabel 4.3 bahwa ketika pengambilan data ADC pada rotation sensor dengan  $V_{ref}$  5,0 Volt, hasil persentase error dari tegangan keluaran yang dihasilkan cenderung besar. Ini menunjukkan bahwa ketika tegangan referensi yang diberikan sebesar 2,5 Volt, tegangan keluaran lebih stabil dan persentase error antara tegangan keluaran yang diukur langsung pada kaki sensor dengan tegangan keluaran yang didapat dari perhitungan lebih kecil. Data tegangan keluaran hasil perhitungan didapatkan dari persamaan 4.1 dibawah ini:

$$ADC = \frac{V_{in}}{V_{ref}} \times 1024 \quad (4.1)$$

Contoh perhitungan pada hasil ADC ketika aktuator sirip berada pada titik -2° dengan  $V_{in}$  yang didapat dari hasil pengukuran sebesar 0,78 Volt dan nilai ADC yang diterima mikrokontroler sebesar 364 maka tegangan yang didapat dari persamaan 4.1 diatas adalah:

$$320 = \frac{V_{in}}{2,5 \text{ Volt}} \times 1024; V_{in} = 0,77 \text{ Volt}$$

Perbedaan besar tegangan yang didapat dari hasil pengukuran dan perhitungan, maka akan didapatkan persentase error yang dapat dicari menggunakan persamaan berikut:

$$\% \text{ error} = \frac{|V_{pengukuran} - V_{perhitungan}|}{V_{perhitungan}} \times 100\% \quad (4.2)$$

$$\% \text{ error} = \frac{0,78 \text{ V} - 0,77 \text{ V}}{0,77 \text{ V}} \times 100\% ; \% \text{ error} = 1,29 \%$$

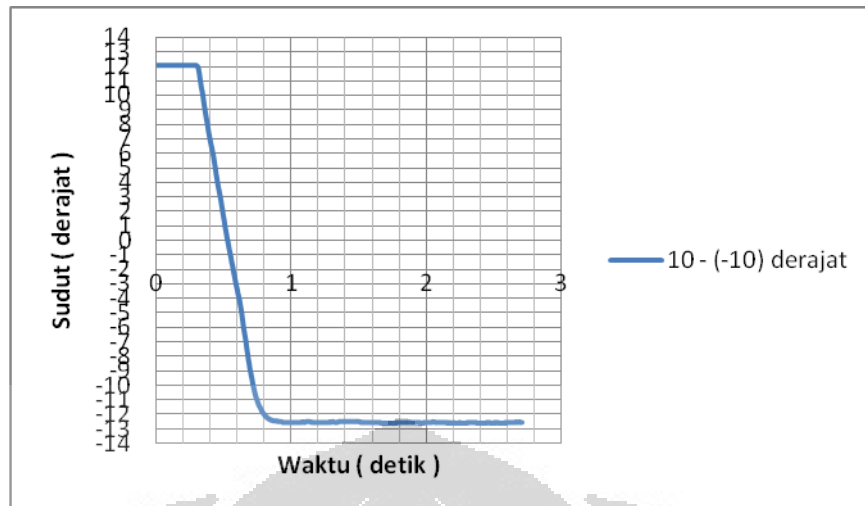
Karena pada saat pemrograman yang di inputkan adalah sudut yang diinginkan, maka data adc ini dikonversi terlebih dahulu menjadi data sudut dan didapatkan persamaan 4.3 :

$$y = -0,053x + 15,44 \quad (4.3)$$

Dimana nilai  $x$  adalah nilai ADC yang didapat dari sensor rotasi dan nilai  $y$  adalah sudut rata – rata dari aktuator.

#### 4.4 Pengujian Sistem *Open-Loop*

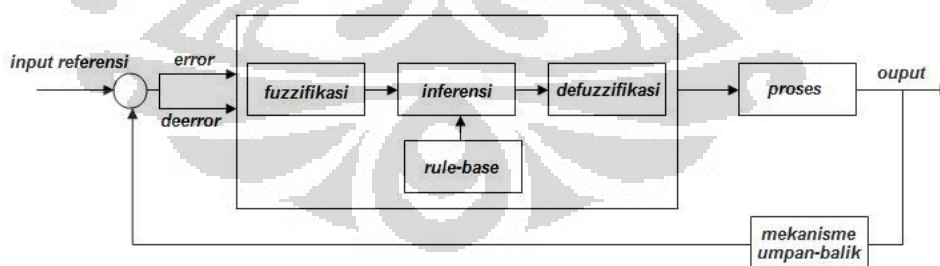
Pengujian sistem *open-loop* dilakukan dengan memberi *setpoint* pada motor tanpa adanya umpan-balik dari aktuator. Pengujian dilakukan dengan memberikan *setpoint* yang dalam pengambilan data ini diberikan pada titik  $-10^\circ$ . Aktuator mulai dari titik  $10^\circ$  menuju titik  $-10^\circ$  dengan tegangan yang diberikan pada motor sebesar 2,99 Volt. Grafik yang didapatkan pada gambar 4.6 adalah nilai dari keluaran adc yang dikirimkan oleh sensor rotasi yang kemudian dikonversi menjadi sudut berbanding dengan waktu. Terlihat pada grafik waktu yang dibutuhkan hingga keadaan tunak selama 0,4 detik, tetapi terjadi kesalahan tunak karena respon motor berhenti pada keadaan sudut  $-12,5^\circ$  dari yang seharusnya hanya  $-10^\circ$ .



Gambar 4.6 Grafik Respon *Brushed DC Motor* Sistem *Open-Loop*

#### 4.5 Pengujian Verifikasi Sistem *Fuzzy Logic*

Metode pengendalian posisi aktuator sirip roket yang dipakai pada skripsi ini adalah dengan menggunakan metode pengendalian fuzzy logic. Metode fuzzy logic berfungsi untuk mengatur keluaran sinyal pwm yang akan mengatur kecepatan motor dc tersebut sehingga pergerakan motor dapat dikendalikan. Input dari metode fuzzy didapatkan dari selisih nilai setpoint yang diberikan dengan nilai ADC yang diberikan oleh sensor rotasi pada alat. Diagram alir sistem yang akan dijalankan dapat dilihat pada gambar 4.7 berikut:

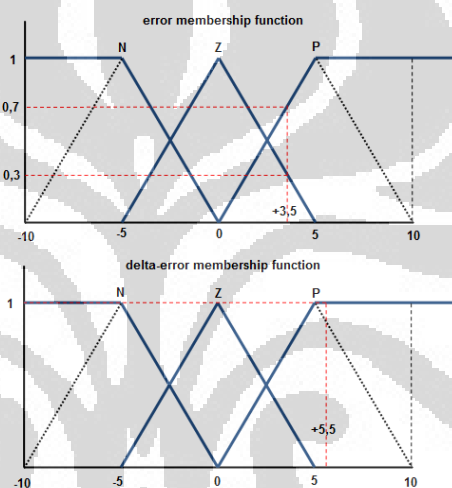


Gambar 4.7 Diagram Alir Metode *Fuzzy Logic* Pada Sistem.

Nilai *error* didapat dari selisih nilai masukan *setpoint* dengan nilai adc pada *rotation sensor*, nilai *deerror* adalah nilai *error* yang baru dikurangi dengan nilai *error* sebelumnya. Dari dua data nilai ini akan dihasilkan sebuah nilai keluaran yang akan mempengaruhi banyaknya pulsa PWM yang akan dikirimkan

ke motor. Kemampuan sistem dapat dilihat dari perbandingan antara error yang dihasilkan oleh alat dengan keluaran yang didapat pada akhir perhitungan metode. Semakin mendekati nilai error = 0 maka nilai keluaran dari sistem fuzzy logic juga akan semakin mengecil untuk mengurangi terjadinya overshoot pada grafik respon aktuator.

Berikut ini akan dilakukan pengujian melakukan simulasi perhitungan dengan memverifikasi hasil simulasi program dengan BASCOM. Sebagai contoh kasus, jika diketahui *error* sudut yang diterima sebesar 3,5 dan *delta error* sudut sebesar 5,5 maka akan didapatkan fungsi *membership* seperti gambar 4.8 berikut:



Gambar 4.8 Fungsi *Membership Function* pada Contoh Kasus.

Lalu dengan berdasarkan tabel rule base yang telah dirancang seperti yang telah diuraikan diatas, maka didapatkan derajat keanggotaan sebagai berikut:

- “error” = +3,5 ; “negative” = 0,0 ; “zero” = 0,3 ; “positive” = 0,7
- “delta-error” = +5,5 ; “negative” = 0,0 ; “zero” = 0,0 ; “positive” = 1,0

Dengan data derajat keanggotaan diatas maka rule yang dipergunakan adalah rule 5, rule 6, rule 8 dan rule 9 sebagai berikut:

5. If (e = 0) AND (er = 0) then Output Zero

- $\text{Mu}(5) = \text{Min}(0,3 ; 0,0) = 0,0$  ; Output Zero (0)

6. If ( $e > 0$ ) AND ( $er = 0$ ) then Output Positif

- $\text{Mu}(6) = \text{Min}(0,7 ; 0,0) = 0,0$  ; Ouput Positif (100)

8. If ( $e = 0$ ) AND ( $er > 0$ ) then Output Negatif

- $\text{Mu}(8) = \text{Min}(0,3 ; 1,0) = 0,3$  ; Ouput Negatif (-100)

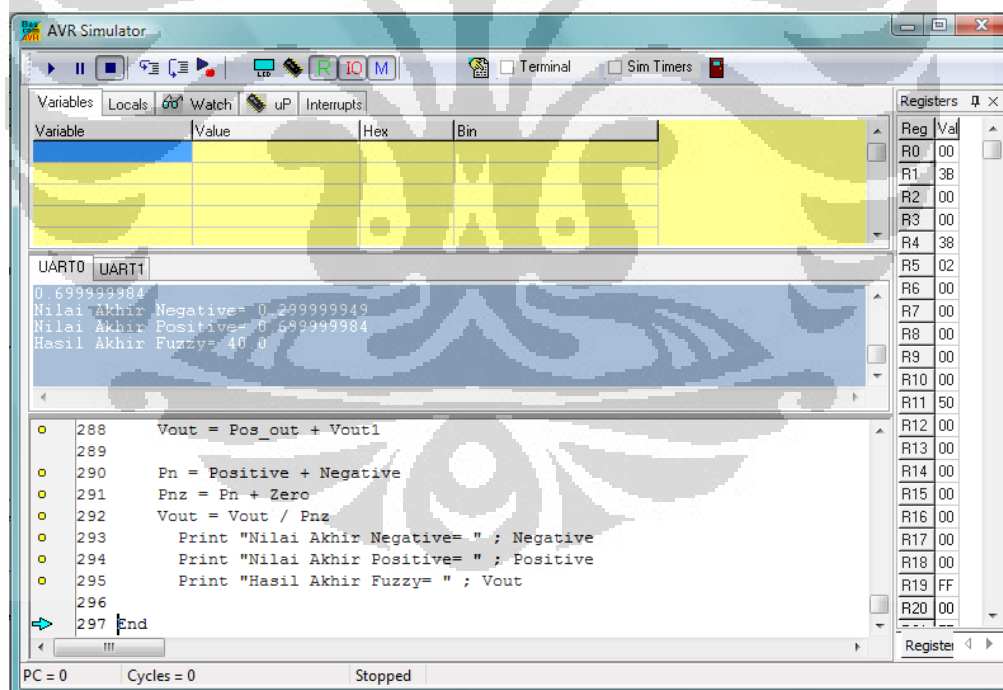
9. If ( $e > 0$ ) AND ( $er > 0$ ) then Output Positif

- $\text{Mu}(9) = \text{Min}(0,7 ; 1,0) = 0,7$  ; Ouput Positif (100)

Dengan nilai dari fungsi himpunan output dengan anggota ( $x$ ; -100; -50; 0; 50; 100) maka hasil dari perhitungan defuzzifikasi dengan metode COG sebagai berikut:

$$\text{fuzzy} = \frac{(0,0 \times 0 + 0,0 \times 100 + 0,3 \times -100 + 0,7 \times 100)}{0,0 + 0,0 + 0,3 + 0,7} = 40$$

Pengujian selanjutnya adalah memverifikasi algoritma logika fuzzy yang telah dibuat dengan menjalankan program simulasinya dalam bahasa BASCOM, seperti pada gambar 4.9 berikut:



```

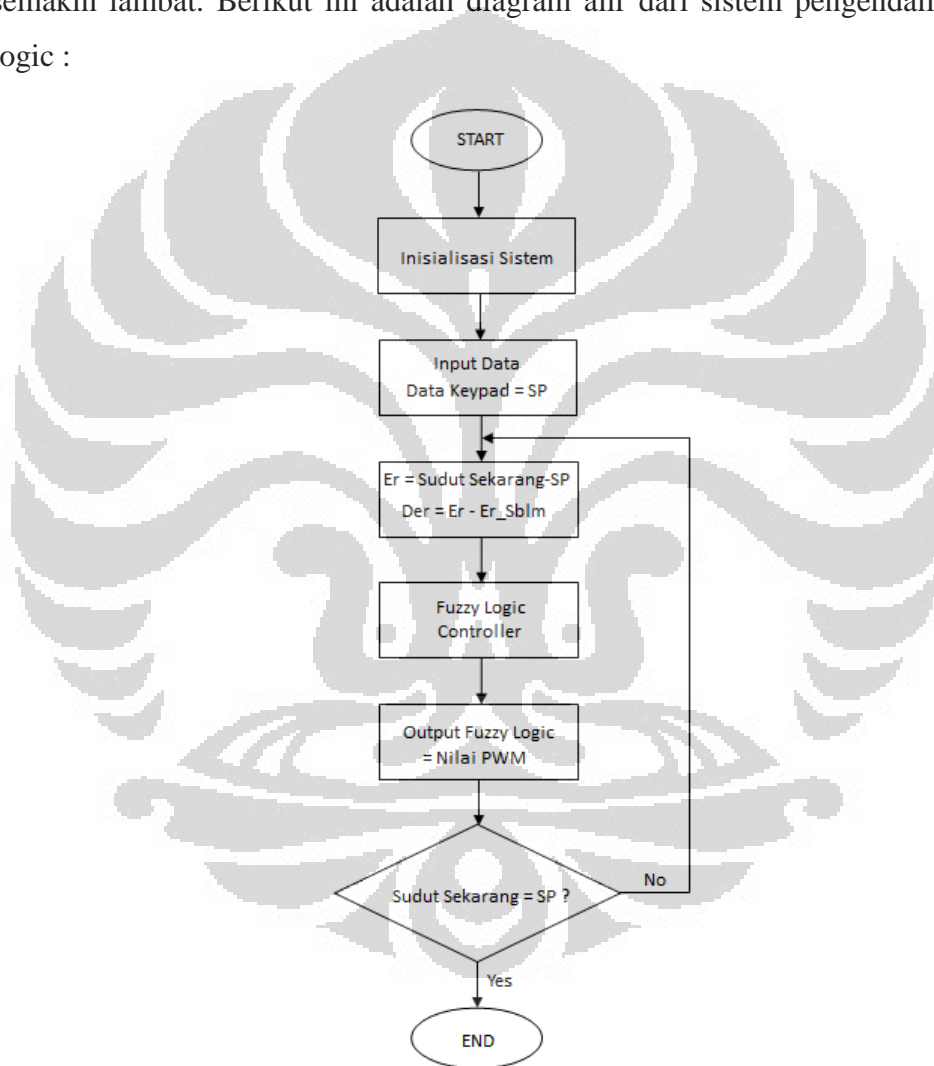
AVR Simulator
Variables Locals Watch uP Interrupts
Variable Value Hex Bin
Registers
R0 00
R1 3B
R2 00
R3 00
R4 3B
R5 02
R6 00
R7 00
R8 00
R9 00
R10 00
R11 50
R12 00
R13 00
R14 00
R15 00
R16 00
R17 00
R18 00
R19 FF
R20 00
UART0 UART1
0.699999984
Nilai Akhir Negative= 0.299999949
Nilai Akhir Positive= 0.699999984
Hasil Akhir Fuzzy= 40.0
288 Vout = Pos_out + Vout1
289
290 Pn = Positive + Negative
291 Pnz = Pn + Zero
292 Vout = Vout / Pnz
293 Print "Nilai Akhir Negative= " ; Negative
294 Print "Nilai Akhir Positive= " ; Positive
295 Print "Hasil Akhir Fuzzy= " ; Vout
296
297 End
PC = 0 Cycles = 0 Stopped
  
```

Gambar 4.9 Hasil Simulasi BASCOM

Setelah dilakukan percobaan, didapatkan data bahwa proses pengolahan data menggunakan metode *fuzzy* ini membutuhkan waktu selama 0,004 detik atau 4 ms dari saat *input error* diberikan hingga didapatkan keluaran *fuzzynya*.

#### 4.6 Pengujian Pengendali Fuzzy Pada Sirip

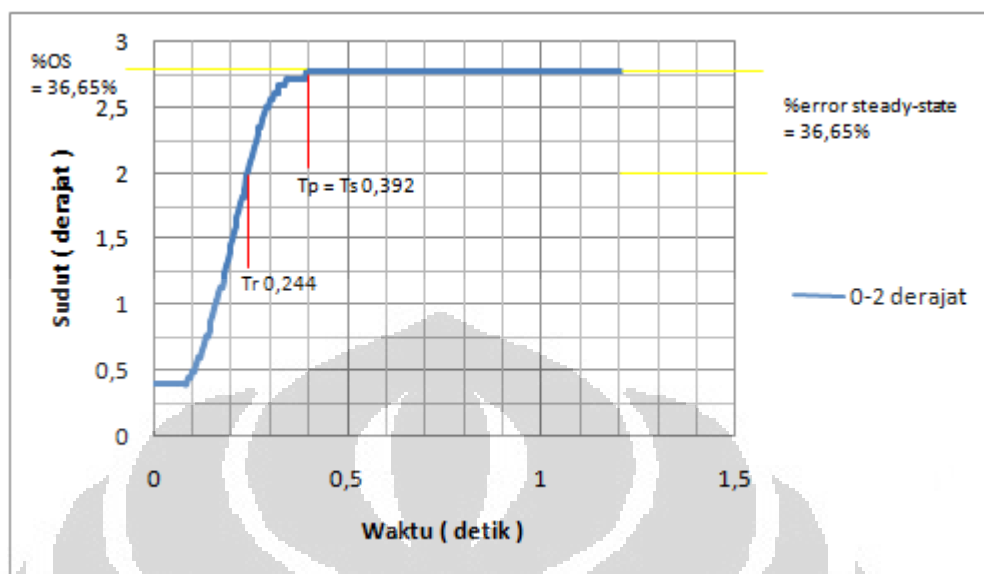
Pengujian dilakukan dengan sistem close-loop, dimana ada umpan-balik dari sistem yang akan memberikan input keadaan nyata dari sirip tersebut. Masukan setpoint diberikan oleh keypad dengan rentang masukan yang dapat dihasilkan antara  $-10^{\circ}$  hingga  $10^{\circ}$ . Sistem close-loop ini dipengaruhi oleh besar error dan perubahan error yang dihasilkan oleh aktuator. Semakin kecil error yang diterima oleh sistem fuzzy maka putaran motor yang dihasilkan oleh PWM akan semakin lambat. Berikut ini adalah diagram alir dari sistem pengendalian fuzzy logic :



Gambar 4.10 Flowchart Sistem Pengendalian *Fuzzy Logic*

Berikut ini beberapa grafik yang dihasilkan pada saat pengujian menggunakan sistem *close-loop* yang mengambil data dari pergerakan aktuator sirip dari posisi  $0^{\circ}$  menuju  $2^{\circ}$ ,  $5^{\circ}$ ,  $9^{\circ}$ , dan  $10^{\circ}$ .

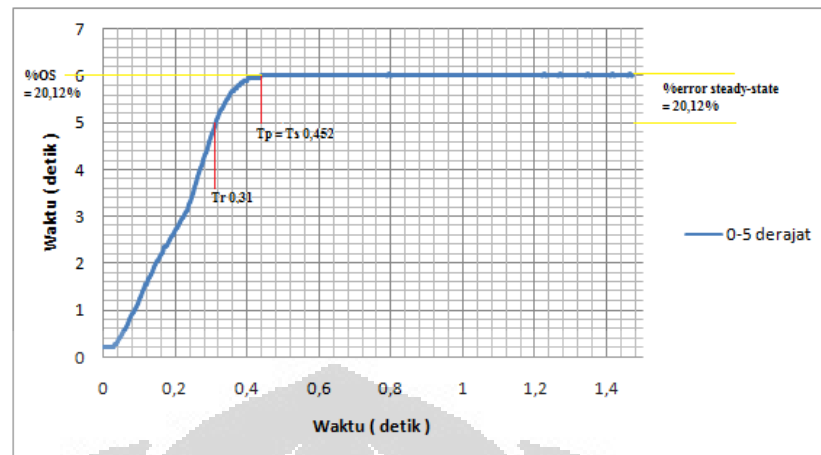
## 1. Pergerakan Aktuator Dari Posisi 0° Menuju 2°



Gambar 4.11 Grafik Respon Pergerakan Aktuator Dari 0° Menuju 2°

Pada gambar 4.11 diatas terlihat respon sistem yang didapatkan pada saat diberikan *input setpoint* pada sudut 2°, terlihat *rise time* pada grafik didapat saat  $t = 0,244$  detik. Respon sistem mengalami kesalahan tunak yang seharusnya berhenti pada nilai sudut 2° tetapi aktuator berhenti pada nilai 2,77° dengan waktu tunak dikisaran 0,392 detik sehingga didapatkan persentase *error* keadaan tunak dan persentase *overshoot* sebesar 36,65 %. Nilai kedua *error* ini sama dikarenakan nilai *peak time* dan *settling time* yang didapatkan juga sama yaitu pada  $t = 0,392$  detik, karena motor berhenti berputar walaupun belum mencapai nilai *setpoint* yang diinginkan. Hal ini disebabkan oleh nilai sudut rata – rata yang dihasilkan berbeda cukup jauh dengan nilai sudut sebenarnya dikarenakan kurang liniernya keluaran nilai ADC yang dihasilkan oleh sensor rotasi. Perbandingan roda gigi yang digunakan juga terlalu kecil sehingga mempersulit aktuator untuk mencapai sudut yang diinginkan, karena motor yang digunakan juga memiliki batas kemampuan untuk mencapai putaran pwm yang dihasilkan. Karena nilai pwmnya terlalu kecil, motor tidak dapat memutar lagi, dan keadaan tersebut dianggap keadaan tunak walaupun nilai sudut yang dihasilkan belum mencapai sudut yang diinginkan.

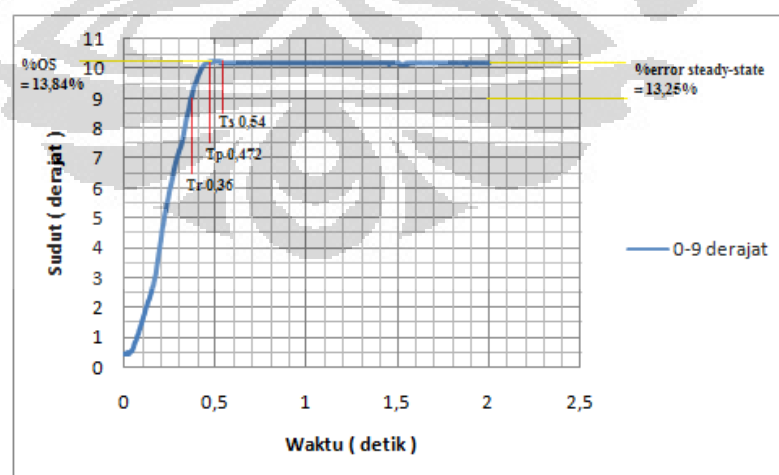
## 2. Pergerakan Aktuator Dari Posisi 0° Menuju 5°



Gambar 4.12 Grafik Respon Pergerakan Aktuator Dari  $0^\circ$  Menuju  $5^\circ$

Seperti terlihat pada gambar 4.12 diatas, kurang stabilnya tegangan yang dihasilkan oleh sensor rotasi menyebabkan pengendalian kurang memuaskan. Data respon motor yang didapatkan saat menggunakan metode pengendalian *fuzzy* pada contoh pergerakan yang sama yaitu dari  $0^\circ$  -  $5^\circ$  maka didapatkan nilai  $T_r = 0,31$  detik,  $T_p$  dan  $T_s = 0,452$  detik dengan *overshoot* maksimum pada titik  $6^\circ$  menghasilkan persentase *overshoot* dan kesalahan keadaan tunak sebesar 20,12%.

### 3. Pergerakan Aktuator Dari Posisi $0^\circ$ Menuju $9^\circ$



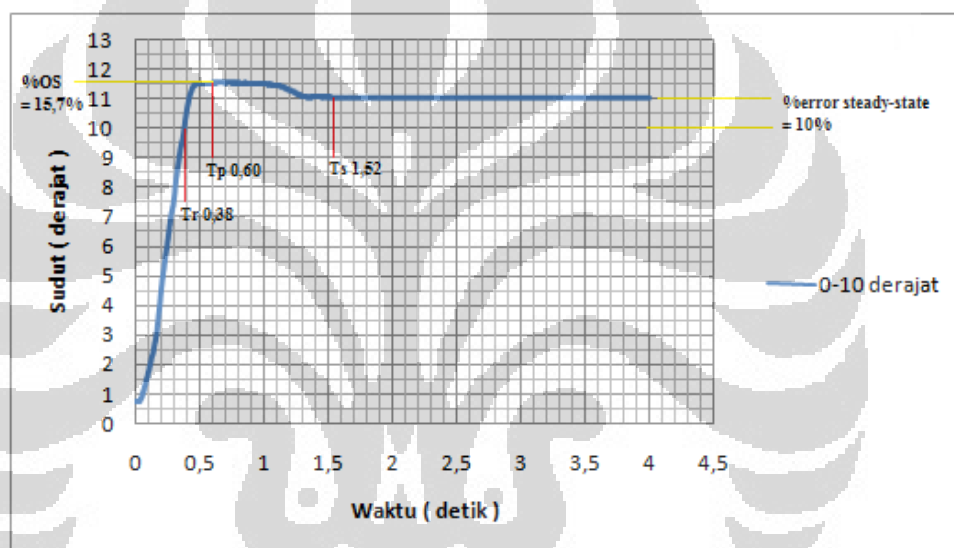
Gambar 4.13 Grafik Respon Pergerakan Aktuator Dari  $0^\circ$  Menuju  $9^\circ$

Grafik yang terdapat pada gambar 4.13 merupakan grafik respon pergerakan aktuator berbanding dengan waktu. Seperti terlihat pada gambar,



respon mencapai nilai *rise time* saat  $t = 0,36$  detik lalu respon mencapai *peak time* pada saat  $t = 0,472$  detik. Seperti dua contoh grafik sebelumnya, aktuator tidak mencapai *setpoint* yang diinginkan. Aktuator mencapai nilai *setpoint* pada saat  $t = 0,54$  detik diposisi sudut  $10,193^\circ$  dengan nilai persentase *error* keadaan tunak sebesar  $13,25\%$  setelah sebelumnya mencapai nilai *overshoot* pada posisi sudut  $10,246^\circ$  dengan waktu  $t = 0,472$  detik dengan persentase nilai *overshoot* sebesar  $13,84\%$ .

#### 4. Pergerakan Aktuator Dari Posisi $0^\circ$ Menuju $10^\circ$



Gambar 4.14 Grafik Respon Pergerakan Aktuator Dari  $0^\circ$  Menuju  $10^\circ$

Contoh respon yang keempat seperti yang terlihat pada gambar 4.14 adalah grafik respon pergerakan dari aktuator pada saat pengujian dari posisi  $0^\circ$  menuju  $10^\circ$ . Terlihat bahwa respon mencapai nilai *rise time* saat  $t = 0,38$  detik lalu respon mencapai nilai *peak* diposisi  $11,57^\circ$  saat  $t = 0,6$  detik dengan persentase *overshoot*  $15,7\%$ . Nilai tunak dicapai pada posisi aktuator  $11^\circ$  saat  $t = 1,52$  sehingga didapatkan persentase kesalahan tunak sebesar  $10\%$ .

Kelinieran dari sensor rotasi ini sangat mempengaruhi aktuator dikarenakan *setpoint* dan *error* yang diinginkan tergantung dari keluaran yang dihasilkan oleh sensor. Pada empat grafik yang ditampilkan diatas terlihat bahwa respon motor sempat mencapai nilai *setpoint* yang diinginkan, lalu karena

melewati *setpointnya* maka motor akan memberikan respon dengan membalik putarannya. Tetapi dikarenakan perbedaan *errornya* sudah sangat kecil, maka nilai keluaran *fuzzy* yang juga merupakan persentase nilai pwm yang dihasilkan akan kecil pula sehingga mengakibatkan motor tidak dapat memberikan respon putaran untuk mengembalikan aktuator menuju posisi yang diinginkan. Berikut ini adalah nilai rata – rata dari respon pergerakan motor yang didapatkan setelah melakukan pengujian seperti yang telah dibahas sebelumnya yang akan ditampilkan pada tabel 4.4 berikut ini:

Tabel 4.4 Respon Rata – Rata Aktuator Sirip Metode Fuzzy

Respon Transient	Posisi Aktuator ( dari 0° )				Nilai Rata - rata
	2°	5°	9°	10°	
Tr	0,24 detik	0,31 detik	0,36 detik	0,38 detik	0,32 detik
Tp	0,39 detik	0,45 detik	0,47 detik	0,60 detik	0,47 detik
Ts	0,39 detik	0,45 detik	0,54 detik	1,52 detik	0,72 detik
% OS	36,65 %	20,12 %	13,84 %	15,7 %	21,57 %
% SSE	36,65 %	20,12 %	13,25 %	10 %	20 %

#### 4.7 Perbandingan Kinerja Pengendali

Setelah metode pengendalian menggunakan fuzzy ini telah dicoba dan diimplementasikan, maka salah satu cara pengukuran kinerja dari pengendali adalah dengan membandingkan hasil respon aktuator dengan metode pengendalian lain. Metode yang digunakan sebagai pembanding adalah metode PID ( *Proportional, Integral, dan Derivative*) yang telah diuji coba dengan menggunakan aktuator yang sama. Pada tabel 4.5 dibawah merupakan grafik respon rata – rata aktuator dengan menggunakan metode pengendalian PID:

Tabel 4.5 Perbandingan Respon Transient Metode *Fuzzy* dan PID

<b>Respon</b>	<b>Tr</b>	<b>Tp</b>	<b>Ts</b>	<b>% OS</b>	<b>% SSE</b>
<b><i>Fuzzy</i></b>	0,32 s	0,47 s	0,72 s	21,57 %	20 %
<b>PID</b>	0,42 s	0,675 s	0,81 s	5,375 %	14,75 %

Dari tabel 4.5 diatas dapat disimpulkan bahwa respon dari pergerakan aktuator dengan menggunakan metode pengendalian PID lebih lambat dari pergerakan aktuator dengan metode pengendalian *fuzzy*, tetapi segi keakuratan pergerakan posisi aktuator sirip dengan menggunakan metode pengendalian PID lebih baik daripada pengendalian dengan metode *fuzzy*. Keakuratan posisi dari pengendalian PID lebih baik dari pengendalian *fuzzy* terlihat dari besarnya persentase kesalahan *overshoot* dan kesalahan keadaan tunak yang didapatkan dari kedua metode pengendalian tersebut. Pada pengendalian dengan metode PID didapatkan nilai rata – rata persentase *overshoot* sebesar 5,375 %, sementara persentase *overshoot* dari metode *fuzzy* didapatkan *error* sebesar 21, 57 %. Lalu persentase kesalahan tunak dari metode PID didapatkan nilai *error* sebesar 14, 75 %, sedangkan persentase kesalahan tunak pada pengendalian *fuzzy* sebesar 20 %. Karena dalam pengendalian posisi yang diutamakan adalah keakuratan dari pergerakan aktuatornya, maka dapat disimpulkan bahwa pengendalian posisi aktuator sirip dengan menggunakan metode pengendalian PID lebih baik dari pengendalian posisi aktuator dengan menggunakan metode pengendalian *fuzzy*.

## BAB 5

### KESIMPULAN DAN SARAN

Pada bab 5 ini berisi tentang rangkuman keseluruhan dari perancangan dan pengujian yang telah dilakukan, mengacu pada analisis dari sistem yang telah dirancang.

#### 4.1 Kesimpulan

Setelah melakukan perancangan dan analisa maka didapatkan kesimpulan mengenai sistem prototipe yang telah dibuat.

1. Prototipe ini dirancang dengan menggunakan brushed DC motor sebagai penggerak dari aktuator sirip roket karena memiliki torsi yang besar dan mudah untuk dikendalikan.
2. Digunakan mikrokontroler ATmega8535 dengan bahasa pemrograman *basic compiler* (BASCUM) sebagai pusat pemroses data.
3. Bahan konstruksi mekanik menggunakan plat besi setebal 1 mm agar lebih kokoh dan lebih presisi dalam pergerakannya, serta menggunakan sebuah alat bantu pengukuran berupa *pitch gauge* sebagai indikator besar sudut yang diperoleh oleh aktuator sirip roket.
4. Digunakan sebuah sensor rotasi berbasis variable resistor buatan DFrobot sebagai mekanisme umpan-balik yang akan mengirimkan data tegangan sesuai dengan besar sudut putaran dari tuas yang ada pada sensor tersebut.
5. Pergerakan tiap sudut dari aktuator sirip dapat dilihat pada *pitch gauge* yang memiliki indikator besar sudut dari  $-10^{\circ}$  hingga  $10^{\circ}$  dengan kenaikan minimal  $1^{\circ}$ .
6. Digunakan metode pengendalian logika fuzzy sebagai pengendali keluaran dari sistem dengan menggunakan metode inferensi Mamdani dan menggunakan metode centroid untuk defuzzifikasinya.
7. Hasil keluaran sistem dengan menggunakan metode fuzzy didapatkan nilai rata - rata  $T_r = 0,32$  detik,  $T_p = 0,47$  detik,  $T_s = 0,72$  detik

dengan nilai persentase overshoot sebesar 21,57 % dan juga kesalahan keadaan tunak sebesar 20 %.

8. Kemampuan motor dalam merespon pada keadaan pwm kecil menjadi kendala, karena ketika motor sudah tidak berputar maka sistem akan menganggapnya dalam keadaan tunak walalupun sudut yang dihasilkan belum merupakan sudut yang diinginkan.
9. Dibanding dengan metode pengendalian PID, respon yang didapatkan pada metode *fuzzy* masih lebih lambat dikarenakan masih kurang sempurnanya rancangan logika *fuzzy* maupun pemrogramannya.

#### 4.2 Saran

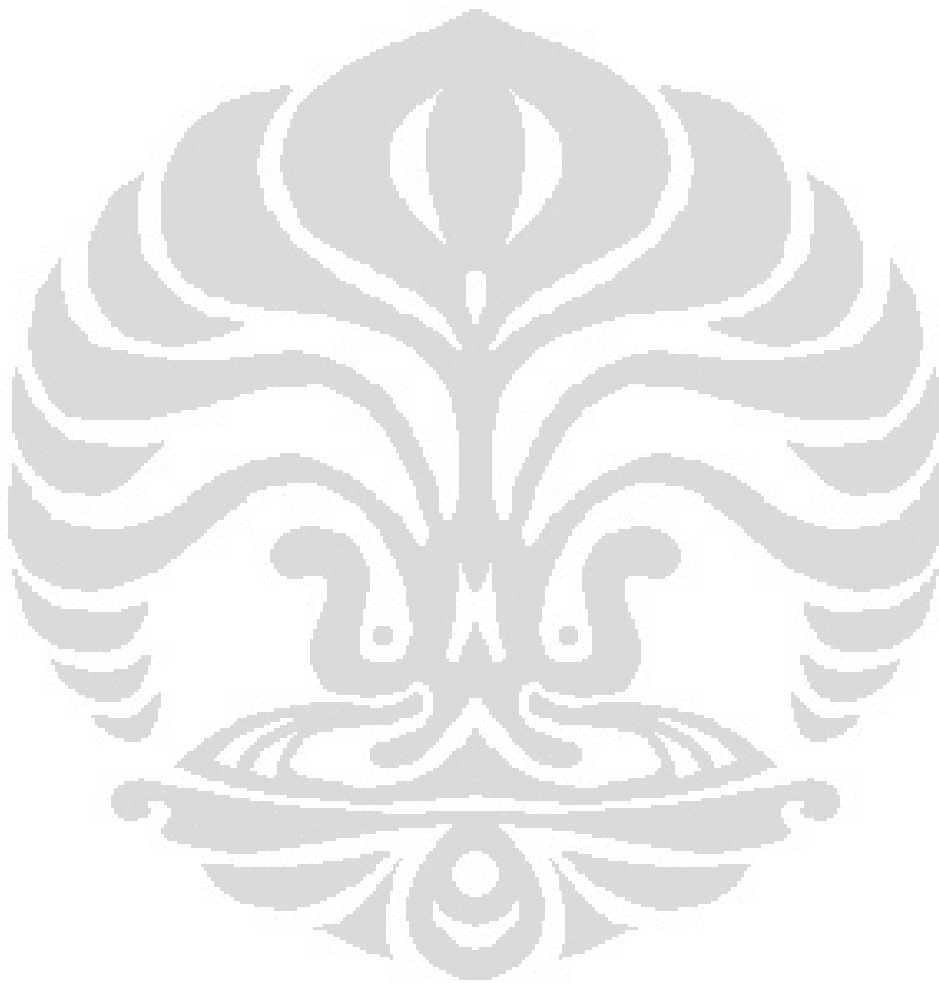
Setelah dilakukan analisa dari perancangan sistem aktuator pergerakan sirip roket maka ada beberapa hal yang perlu diperbaiki untuk pengembangan penelitian selanjutnya, diantaranya:

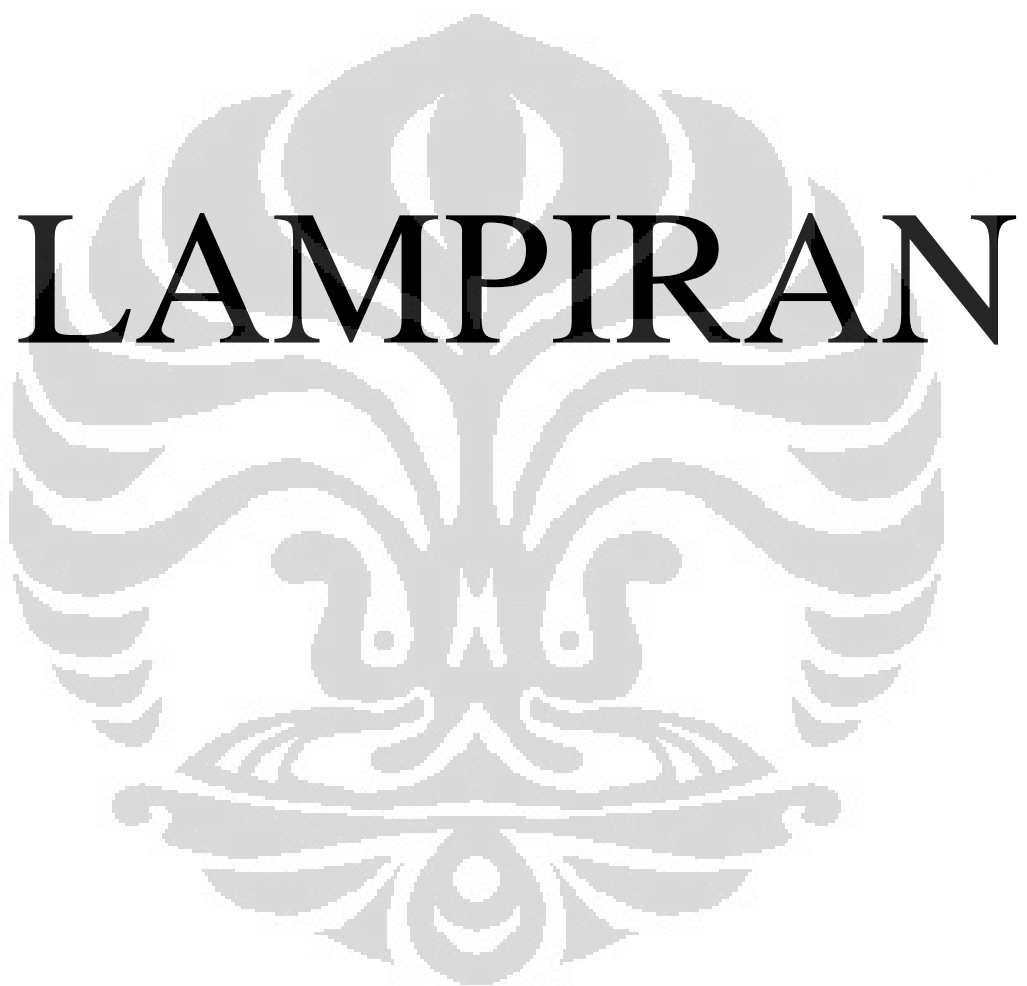
1. Konstruksi mekanik yang digunakan sebaiknya diperkokoh menggunakan material besi keseluruhannya, agar pergerakan alat lebih presisi.
2. Agar pengendalian pergerakan aktuator lebih baik, sensor yang digunakan sebagai umpan-balik sistem sebaiknya digunakan rotation encoder yang dapat menyimpan data pergerakan motor yang digunakan bukan menggunakan sensor dari pergerakan posisi aktuator siripnya.
3. Diperlukan perbaikan pada perancangan sistem kendali logika fuzzy dikarenakan masih belum stabilnya sistem dalam memenuhi permintaan keadaan yang diinginkan.
4. Motor yang digunakan sebaiknya yang memiliki torsi besar pada putaran yang kecil agar dapat mencapai keakuratan yang diinginkan.
5. Perbandingan gear pada transmisi antara shaft motor dengan shaft pada aktuator sebaiknya diperbesar perbandingannya, putaran motor harusnya direduksi terlebih dahulu agar putaran motor tidak sama dengan putaran pada aktuator agar pengendalian menjadi lebih mudah.

## DAFTAR REFERENSI

- [1] Reston Condit. (2004). *Brushed DC Motor Fundamentals*. Microchip Technology Inc.
- [2] Chan, Yefri. (2010). *Teori dasar roda gigi*. Diambil tanggal 15 November 2011 dari <http://yefrichan.files.wordpress.com>
- [4] Depok Instrument. Teori Keypad Matriks 4x4. Diambil tanggal 20 April 2012 dari <http://depok-instrument.wordpress.com>
- [5] Depok Instrument. DI-Smart AVR. Diambil tanggal 3 Mei 2012 dari <http://depok-instrument.wordpress.com>
- [6] SGS-Thomson Microelectronic. (1995). *Dual Full-Bridge Driver*.
- [7] Dfrobot. (2010). *Rotation sensor*. Diambil tanggal 8 Oktober 2011 dari <http://www.robotshop.com/dfrobot-rotation-sensor-v1.html>
- [8] Depok Instrument. DI-MC8462 Motor Driver. Diambil tanggal 3 Mei 2012 dari <http://depok-instrument.wordpress.com>
- [9] Winarno, Totok. (2009). Aplikasi Fuzzy Logic Sebagai Kontrol Posisi Motor DC.
- [11] Motor, Maxon. (2010). *Planetary gearhead*. Diambil tanggal 17 Mei 2012 dari <http://uk.farnell.com/maxon-motors/166174/gearhead-planetary-246-1-ratio/dp/1761297>
- [12] Siradjuddin, Indrazno. Desain Algoritma dan Pemrograman Fuzzy pada Mikrokontroler AVR dengan menggunakan BASCOM Kompiler.
- [13] Lapan. (2012). Raket kendali RXX 200. Diambil tanggal 19 Mei 2012 dari [www.lapan.go.id](http://www.lapan.go.id)
- [14] Wikipedia. Epicylic Gearing. Diambil tanggal 12 Juni 2012 dari [http://en.wikipedia.org/wiki/Epicyclic\\_gearing](http://en.wikipedia.org/wiki/Epicyclic_gearing)
- [15] Wikipedia. Transient Response. Diambil tanggal 12 Juni 2012 dari [http://en.wikipedia.org/wiki/Transient\\_response](http://en.wikipedia.org/wiki/Transient_response)
- [16] Bousserhane, K., A. Hazzab, M. Rahli, B. Mazari, and M. Kamli. *Position Control of Linear Induction Motor using an Adaptive Fuzzy Integral-Backstepping Controller*. Diambil tanggal 23 Maret 2012 dari [http://www.journal.tfc.kg.ac.rs/Vol\\_3-1?01-Bousserhane.pdf](http://www.journal.tfc.kg.ac.rs/Vol_3-1?01-Bousserhane.pdf)

- [17] Rachman, Anwar Shiddiq Abdul. (2012). *Rancang Bangun Prototipe Sistem Aktuator Kendali Sirip menggunakan Brushless DC Motor*. Depok. Universitas Indonesia.
- [18] Kurniawan, Arief. (2012). *Rancang Bangun Prototipe Sistem Aktuator Sirip Roket Kendali Menggunakan Brushed Dc Motor dan Planetary Gear*. Depok. Universitas Indonesia.







```

$regfile = "m8535.dat"
$crystal = 11059200
$baud = 9600

$swstack = 64
$hstack = 64
$framesize = 64

'===== Lcd
Inisial
Config Lcd = 16 * 2
Config Lcdpin = Pin , Db4 = Portb.4 ,
Db5 = Portb.5 , , Db6 = Portb.6 , , Db7
= Portb.7 , E = Portb.2 , Rs = Portb.0
Cls
Cursor Off

'===== Lcd
Inisial
Const Trace = 0
'=====
keypad Inisial
Port_keypad Alias Portc
Pin_keypad Alias Pinc
Ddr_keypad Alias Ddrc

K0 Alias Pin_keypad.0
K1 Alias Pin_keypad.1
K2 Alias Pin_keypad.2
K3 Alias Pin_keypad.3

B0 Alias Port_keypad.4
B1 Alias Port_keypad.5
B2 Alias Port_keypad.6
B3 Alias Port_keypad.7

Declare Sub Keyup
Declare Function Getkey() As Byte
Declare Function Setingdigit(byval

'Inisialisasi ADC

Config Adc = Single , Prescaler = Auto
Start Adc

'=====
keypad Inisial

Config Timer1 = Pwm , Pwm = 10 ,
Compare A Pwm = Clear Down ,
Compare B Pwm = Clear Down ,
Prescale = 1

Config Portd.6 = Output
Config Portd.7 = Output

Dim S As String * 16
Dim Setpoint As Integer
Dim Vout As Single , Voutabs As Single
Dim Nilaipwm As Long

Dim Sudut As Byte
Dim Sp As Byte
Dim Sudut_skrng As Single
Dim Sudut_skrng_h As Single
Dim Sudut_skrng_l As Single
Dim X As Word
Dim O As Single

Dim Positive As Single
Dim Zero As Single
Dim Negative As Single

Dim Negative_er As Single
Dim Zero_er As Single
Dim Positive_er As Single

Dim Negative_der As Single
Dim Zero_der As Single
Dim Positive_der As Single

Dim Er As Single
Dim Der As Single
Dim Er_before As Single

'Dim Min05 As Single
'Dim Min02 As Single
'Dim Min50 As Single
'Dim Vout As Single
Dim Vout1 As Single
Dim Pn As Single
Dim Pnz As Single

Dim Output1 As Single
Dim Output2 As Single
Dim Output3 As Single
Dim Output4 As Single
Dim Output5 As Single
Dim Output6 As Single
Dim Output7 As Single
Dim Output8 As Single
Dim Output9 As Single

Dim Neg_out As Single
Dim Zero_out As Single
Dim Pos_out As Single

Dim Kecilr1 As Single
Dim Kecilr2 As Single
Dim Kecilr3 As Single
Dim Kecilr4 As Single
Dim Kecilr5 As Single
Dim Kecilr6 As Single
Dim Kecilr7 As Single
Dim Kecilr8 As Single
Dim Kecilr9 As Single
Dim Setpoints As Single

Portd.6 = 0
Portd.5 = 0

Do
  Cls
  Lcd " Input SP = "
  S = Setingdigit(2 , 0 , Angka)
  Setpoint = Val(s)
  Setpoints = Setpoint
  'Setpoint = 10
  Cls
  Lcd "fuzi start"
  Do
    X = Getadc(0)
    'X = 123
    Sudut_skrng = -0.0533 * X
    Sudut_skrng = Sudut_skrng + 15.445

    Er = Sudut_skrng - Setpoint
    'Print "sudut_skrng=" ; Sudut_skrng
    Der = Er - Er_before

'Fuzzifikasi]

If Er <= -10 Then
  Negative_er = 1
  Positive_er = 0
  Zero_er = 0

Elseif Er <= -5 Then
  Zero_er = 0
  Positive_er = 0
  Negative_er = Er + 10
  Negative_er = Negative_er / 5

Elseif Er <= 0 Then
  Negative_er = 0 - Er
  Negative_er = Negative_er / 5
  Zero_er = Er + 5
  Zero_er = Zero_er / 5
  Positive_er = 0

Elseif Er < 5 Then
  Negative_er = 0
  Zero_er = 5 - Er
  Zero_er = Zero_er / 5
  Positive_er = Er / 5

Elseif Er < 10 Then
  Negative_er = 0
  Zero_er = 0
  Positive_er = 10 - Er
  Positive_er = Positive_er / 5
Else
  Negative_er = 0
  Positive_er = 1
  Zero_er = 0
End If

'ambil nilai der
If Der <= -10 Then
  Negative_der = 1
  Positive_der = 0
  Zero_der = 0

Elseif Der <= -5 Then
  Zero_der = 0
  Positive_der = 0
  Negative_der = Der + 10
  Negative_der = Negative_der / 5

Elseif Der <= 0 Then
  Negative_der = 0 - Der
  Negative_der = Negative_der / 5
  Zero_der = Der + 5
  Zero_der = Zero_der / 5
  Positive_der = 0

Elseif Der < 5 Then
  Negative_der = 0
  Zero_der = 5 - Der
  Zero_der = Zero_der / 5
  Positive_der = Der / 5

Elseif Der < 10 Then
  Negative_der = 0
  Zero_der = 0
  Positive_der = 10 - Der
  Positive_der = Positive_der / 5
Else
  Negative_der = 1
  Positive_der = 0
  Zero_der = 0
End If

```

```

'Inferensi]
'cari nilai R1
  If Negative_er <= Negative_der Then
    Kecilr1 = Negative_er
  Elseif Negative_er >= Negative_der
  Then
    Kecilr1 = Negative_der
  End If

'cari nilai R2
  If Zero_er < Negative_der Then
    Kecilr2 = Zero_er
  Elseif Zero_er >= Negative_der Then
    Kecilr2 = Negative_der
  End If

'cari nilai R3
  If Positive_er <= Negative_der Then
    Kecilr3 = Positive_er
  Elseif Positive_er >= Negative_der
  Then
    Kecilr3 = Negative_der
  End If

'cari nilai R4
  If Negative_er <= Zero_der Then
    Kecilr4 = Negative_er
  Elseif Negative_er >= Zero_der Then
    Kecilr4 = Zero_der
  End If

'cari nilai R5
  If Zero_er <= Zero_der Then
    Kecilr5 = Zero_er
  Elseif Zero_er >= Zero_der Then
    Kecilr5 = Zero_der
  End If
  'Print "R5=" ; Kecilr5

'cari nilai R6
  If Positive_er <= Zero_der Then
    Kecilr6 = Positive_er
  Elseif Positive_er >= Zero_der Then
    Kecilr6 = Zero_der
  End If

'cari nilai R7
  If Negative_er <= Positive_der Then
    Kecilr7 = Negative_er
  Elseif Negative_er >= Positive_der
  Then
    Kecilr7 = Zero_der
  End If

'cari nilai R8
  If Zero_er <= Positive_der Then
    Kecilr8 = Zero_er
  Elseif Zero_er >= Positive_der Then
    Kecilr8 = Positive_der
  End If

'cari nilai R9
  If Positive_er <= Positive_der Then
    Kecilr9 = Positive_er
  Elseif Positive_er >= Positive_der
  Then
    Kecilr9 = Positive_der
  End If

'[Defuzzifikasi] menentukan nilai crisp

Output1 = Kecilr1 * -100
Output2 = Kecilr2 * 100
Output3 = Kecilr3 * 100
Output4 = Kecilr4 * -100
Output5 = Kecilr5 * 0
Output6 = Kecilr6 * 100
Output7 = Kecilr7 * -100
Output8 = Kecilr8 * -100
Output9 = Kecilr9 * 100

Neg_out = Output1 + Output4
Neg_out = Neg_out + Output7
Neg_out = Neg_out + Output8

Zero_out = Output5
Pos_out = Output2 + Output3
Pos_out = Pos_out + Output6
Pos_out = Pos_out + Output9

Negative = Kecilr1 + Kecilr4
Negative = Negative + Kecilr7
Negative = Negative + Kecilr8
' Print "Negative=" ; Negative

Zero = Kecilr5
' Print "Zero=" ; Zero

Positive = Kecilr2 + Kecilr3
Positive = Positive + Kecilr6
Positive = Positive + Kecilr9
' Print "Positive=" ; Positive

Vout1 = Neg_out + Zero_out
Vout = Pos_out + Vout1

Pn = Positive + Negative
Pnz = Pn + Zero
Vout = Vout / Pnz
' Print "Nilai Akhir NegOut=" ;
Neg_out
' Print "Nilai Akhir ZeroOut=" ;
Zero_out
' Print "Nilai Akhir PosOut=" ;
Pos_out
'Print "Hasil Akhir Fuzzy=" ; Vout

'masukkan er menjadi nilai er_before
Er = Er_before

'perhitungan fuzzy
Voutabs = Abs(vout)
Nilaipwm = Voutabs * 1023
Nilaipwm = Nilaipwm / 100

If Vout > 0 Then
  'Portd.7 = 0
  Portd.6 = 1
  Nilaipwm = 1023 - Nilaipwm
  'Print "Kanan"

Else
  'Portd.7 = 1
  Portd.6 = 0
  'Print "Kiri"
End If

Print X
Pwm1a = Nilaipwm
Pwm1b = Nilaipwm
Waitms 1

Loop
Cls
S = Fusing(sudut_skrng , "###.###")
Lcd "Sudut=" ; S
S = Fusing(setpoints , "###.###")
Lowerline
Lcd "SP=" ; S
Keyup
Do
  Sp = Getkey()
Loop Until Sp <> "$"
Keyup
Loop

Sub Keyup
  Ddr_keypad = &B11110000
  B0 = 0
  B1 = 0
  B2 = 0
  B3 = 0
  Do
    Waitms 10
    If K0 = 1 Then
      If K1 = 1 Then
        If K2 = 1 Then
          If K3 = 1 Then
            Exit Do
          End If
        End If
      End If
    End If
  End If
Loop
End Sub

Function Getkey() As Byte
  Ddr_keypad = &B11110000
  B0 = 1 : B1 = 1 : B2 = 1 : B3 = 1

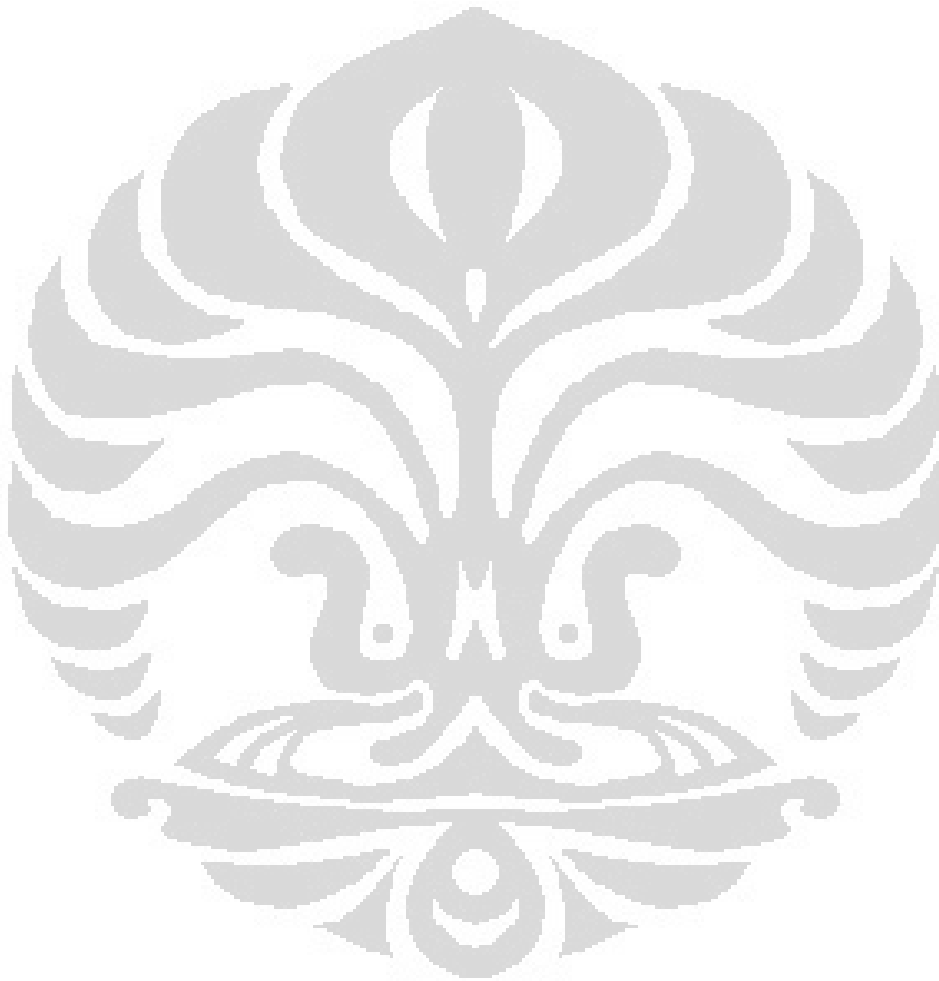
  Getkey = "$"
  B0 = 0
  If K0 = 0 Then
    Getkey = "1"
  Elseif K1 = 0 Then
    Getkey = "2"
  Elseif K2 = 0 Then
    Getkey = "3"
  Elseif K3 = 0 Then
    Getkey = "A"
  End If
  B0 = 1

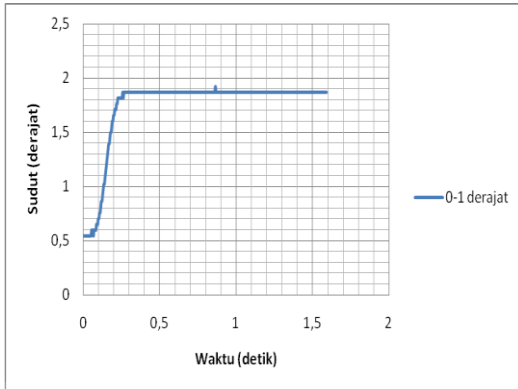
  B1 = 0
  If K0 = 0 Then
    Getkey = "4"
  Elseif K1 = 0 Then
    Getkey = "5"
  Elseif K2 = 0 Then
    Getkey = "6"
  Elseif K3 = 0 Then
    Getkey = "B"
  End If
  B1 = 1

  B2 = 0
  If K0 = 0 Then
    Getkey = "7"
  Elseif K1 = 0 Then
    Getkey = "8"
  Elseif K2 = 0 Then
    Getkey = "8"
  Elseif K3 = 0 Then
    Getkey = "8"
  End If

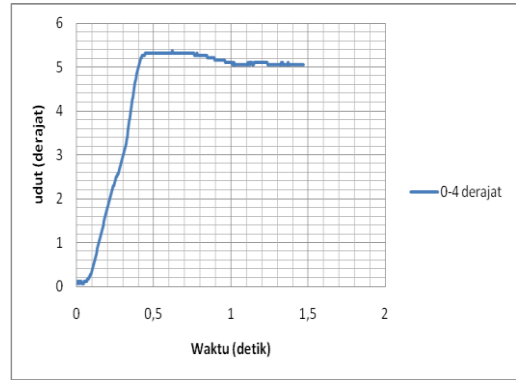
```

```
    Getkey = "9"  
Elseif K3 = 0 Then  
    Getkey = "C"  
End If  
B2 = 1  
  
B3 = 0  
If K0 = 0 Then  
    Getkey = "*"   
Elseif K1 = 0 Then  
    Getkey = "0"  
Elseif K2 = 0 Then  
    Getkey = "#"  
Elseif K3 = 0 Then  
    Getkey = "D"  
End If  
B3 = 1  
End Function
```

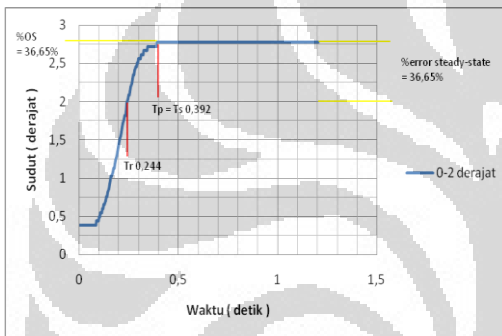




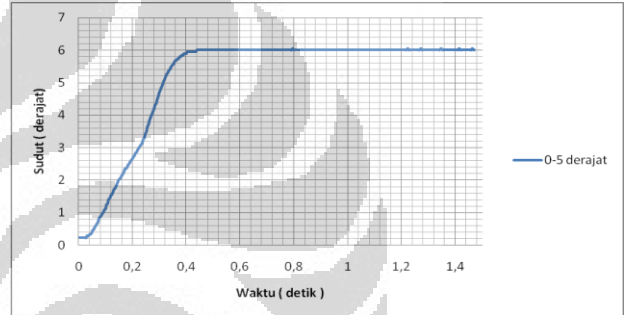
**Pergerakan Aktuator Dari Posisi 0° Menuju 1°**



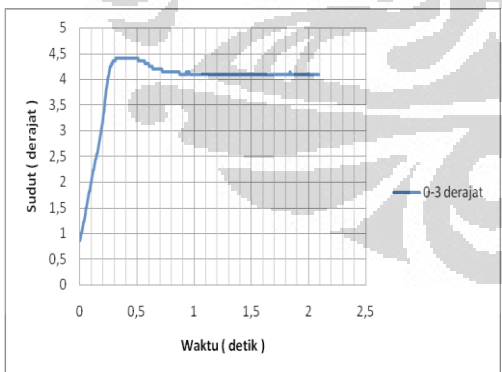
**Pergerakan Aktuator Dari Posisi 0° Menuju 4°**



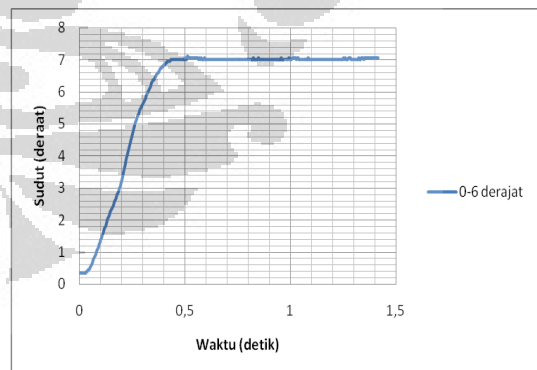
**Pergerakan Aktuator Dari Posisi 0° Menuju 2°**



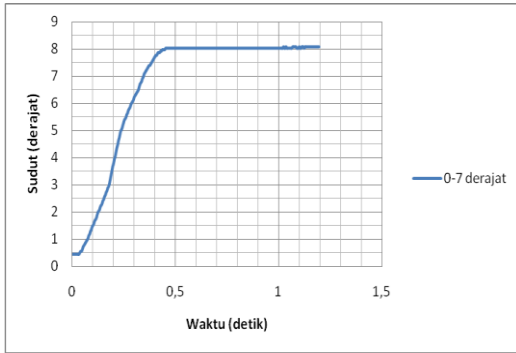
**Pergerakan Aktuator Dari Posisi 0° Menuju 5°**



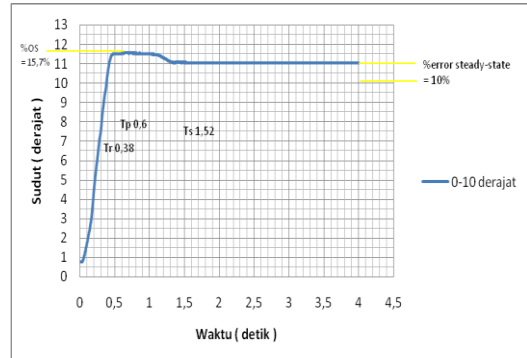
**Pergerakan Aktuator Dari Posisi 0° Menuju 3°**



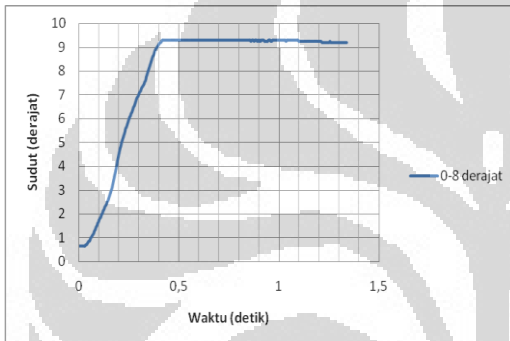
**Pergerakan Aktuator Dari Posisi 0° Menuju 6°**



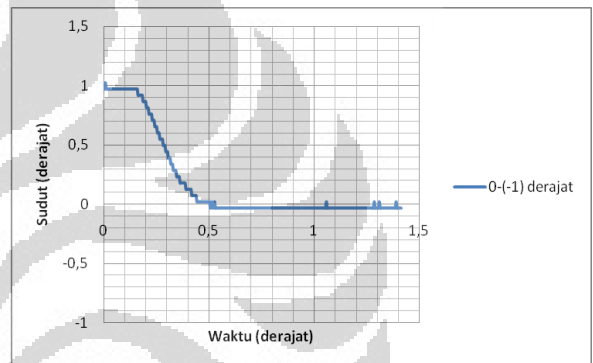
**Pergerakan Aktuator Dari Posisi 0° Menuju 7°**



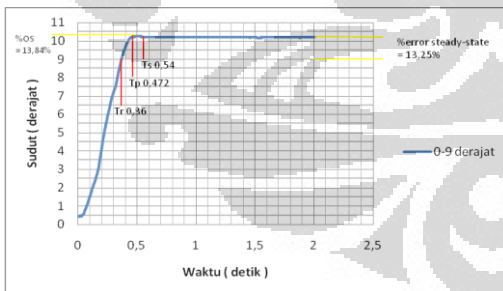
**Pergerakan Aktuator Dari Posisi 0° Menuju 10°**



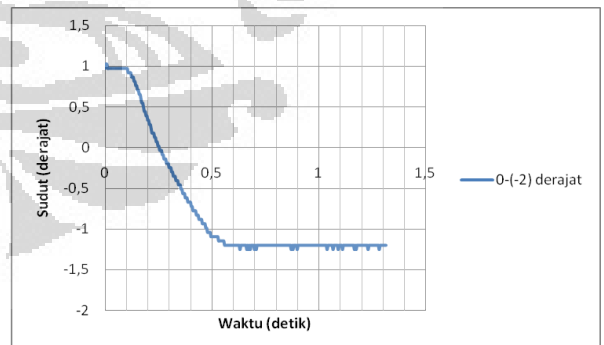
**Pergerakan Aktuator Dari Posisi 0° Menuju 8°**



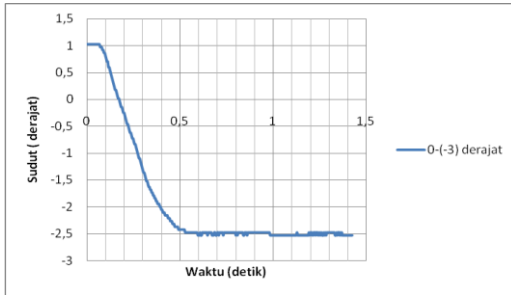
**Pergerakan Aktuator Dari Posisi 0° Menuju -1°**



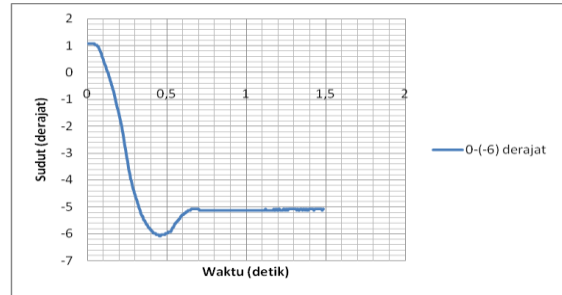
**Pergerakan Aktuator Dari Posisi 0° Menuju 9°**



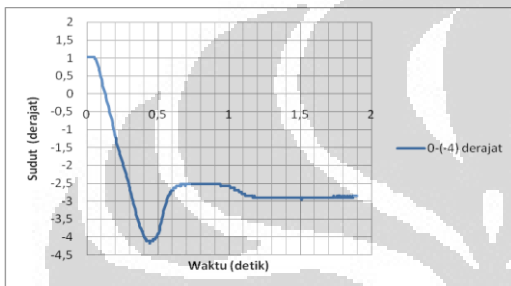
**Pergerakan Aktuator Dari Posisi 0° Menuju -2°**



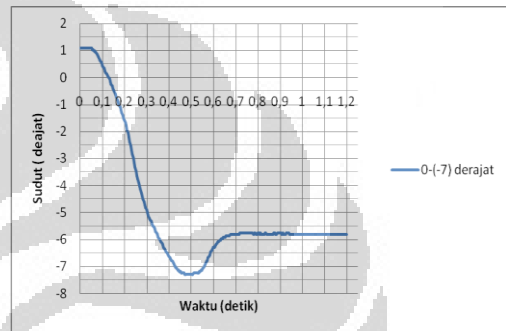
**Pergerakan Aktuator Dari Posisi 0° Menuju -3°**



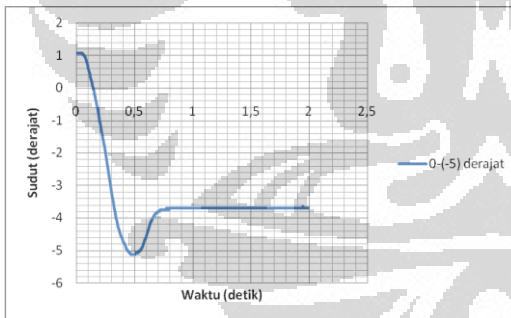
**Pergerakan Aktuator Dari Posisi 0° Menuju -6°**



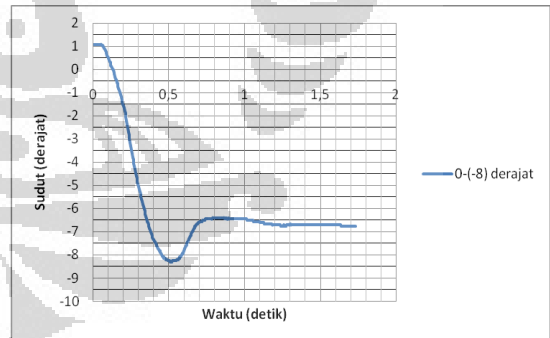
**Pergerakan Aktuator Dari Posisi 0° Menuju -4°**



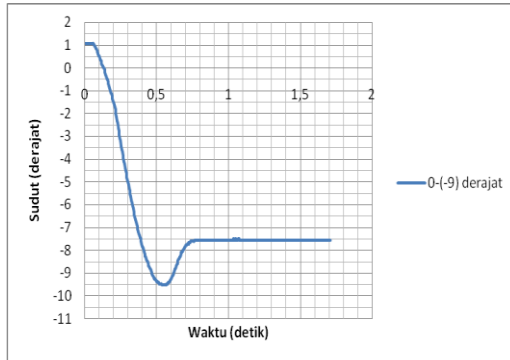
**Pergerakan Aktuator Dari Posisi 0° Menuju -7°**



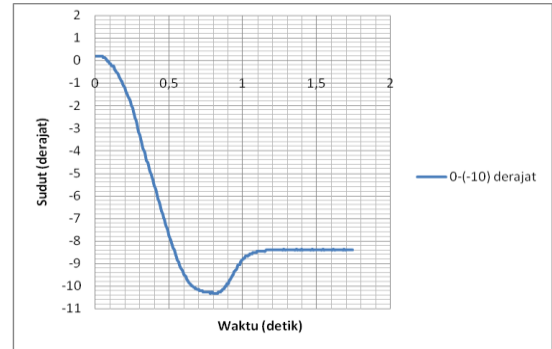
**Pergerakan Aktuator Dari Posisi 0° Menuju -5°**



**Pergerakan Aktuator Dari Posisi 0° Menuju -8°**



**Pergerakan Aktuator Dari Posisi 0° Menuju -9°**



**Pergerakan Aktuator Dari Posisi 0° Menuju -10°**

