

**LEMBAR PERSETUJUAN**

**APLIKASI KONTROL LOGIKA FUZZY UNTUK PENGATURAN  
POSISI SUDUT PUTAR MOTOR DC PADA MODEL SISTEM ROTARY  
PARKING**

**SKRIPSI  
KONSENTRASI TEKNIK KONTROL**

Diajukan untuk memenuhi persyaratan  
memperoleh gelar Sarjana Teknik



Disusun oleh:  
**RISA PRADITA CAMELIA**  
NIM. 0810630087

Telah diperiksa dan disetujui oleh :

**Dosen Pembimbing I**

**Dosen Pembimbing II**

**Fitriana Suhartati, ST., MT.,**

**NIP. 19741017 199802 2 001**

**Erni Yudaningtvas, ST., MT.,Dr**

**NIP. 19650913 199002 2 001**

**LEMBAR PENGESAHAN**  
**APLIKASI KONTROL LOGIKA FUZZY UNTUK PENGATURAN**  
**POSISI SUDUT PUTAR MOTOR DC PADA MODEL SISTEM ROTARY**  
**PARKING**

**SKRIPSI**  
**JURUSAN TEKNIK ELEKTRO**

Diajukan Untuk Memenuhi Persyaratan  
Memperoleh Gelar Sarjana Teknik

Disusun oleh:  
**RISA PRADITA CAMELIA**  
**NIM. 0810630087**

Skripsi ini telah diuji dan dinyatakan lulus pada  
Tanggal

**DOSEN PENGUJI**

**Purwanto, Ir., MT**  
**NIP. 19540424 198601 1 001**

**M. Aziz Muslim, ST., MT., PhD**  
**NIP. 19741203 200012 1 001**

**Goegoes Dwi N, ST., MT**  
**NIP. 19711013 200604 1 001**

Mengetahui

Ketua Jurusan Teknik Elektro

**Dr. Ir. Sholeh Hadi Pramono, MS.**  
**NIP. 19580728 198701 1 001**

## KATA PENGANTAR

Segala puji syukur penulis panjatkan hanya bagi Allah SWT, Pemelihara seluruh alam raya, yang atas limpahan rahmat, taufik dan hidayah-Nya, penulis mampu menyelesaikan Tugas Akhir ini. Tugas akhir ini dikerjakan demi memenuhi salah satu syarat guna memperoleh gelar sarjana Teknik di Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya.

Penulis menyadari bahwa tugas akhir ini bukanlah tujuan akhir dari belajar karena belajar adalah sesuatu yang tidak terbatas. terselesaikannya skripsi ini tentunya tak lepas dari dorongan dan uluran tangan berbagai pihak. Oleh karena itu, tak salah kiranya bila penulis mengungkapkan rasa terima kasih dan penghargaan kepada:

1. Bapak Sukamto dan Ibu Kudriyati selaku orang tua saya yang selalu menginspirasi dan tidak pernah lelah mendukung saya.
2. Ibu Fitriana Suhartati, ST., MT selaku pembimbing I yang telah membimbing dengan penuh kesabaran.
3. Ibu Erni Yudaningtyas, ST., MT., Dr, selaku pembimbing II sekaligus Kepala Laboratorium Sistem Kontrol, berkat bimbingan dan arahan yang beliau berikan sehingga skripsi ini dapat terselesaikan.
4. Adik-adiku, Riris, Firman, dan Lala yang selalu memberi semangat dan memberi keceriaan di rumah.
5. Gallant Hendrayana untuk perhatiannya yang selalu memberi semangat.
6. Teman-teman saya, Mas Aldo, Mas Bima, Kharis, Rahma, Wahyu, Maho, Ceri, Alvita, Riesta, Ninin, Rissa yang selalu membantu dalam pengerjaan Tugas Akhir ini.
7. Segenap anggota asisten Lab. Sistem Kontrol, Maho, Rio, Arif, Seif, dan adek-adek 2009 yang selalu memberi semangat dan dukungan untuk kelanacaran skripsi ini.
8. Segenap anggota tim robot, Bagus, Irfan dkk.

Semua pihak yang telah banyak membantu saya dan tidak saya sebutkan. Semoga Allah SWT membalas kebaikan dan ketulusan semua pihak yang telah membantu menyelesaikan skripsi ini dengan melimpahkan rahmat dan karunia-Nya



Semoga karya penelitian tugas akhir ini dapat memberikan manfaat dan kebaikan bagi banyak pihak demi kemaslahatan bersama serta bernilai ibadah di hadapan Allah SWT. Amien.

Malang, 31 Juli 2012

Penulis



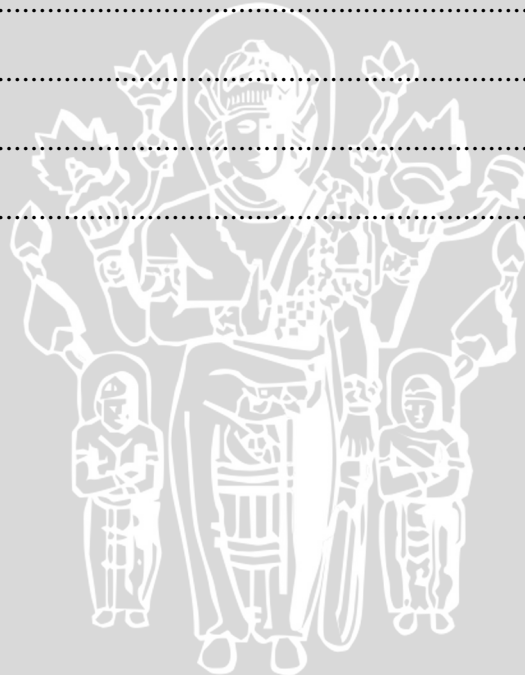
**DAFTAR ISI**

LEMBAR PERSETUJUAN.....	i
KATA PENGANTAR .....	iii
DAFTAR ISI.....	v
DAFTAR TABEL .....	viii
DAFTAR GAMBAR.....	ix
ABSTRAK.....	xii
<b>BAB I PENDAHULUAN</b>	
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Ruang Lingkup .....	2
1.4 Tujuan.....	2
1.5 Sistematika Pembahasan .....	3
<b>BAB II TINJAUAN PUSTAKA</b>	
2.1 Kontrol Logika <i>Fuzzy</i> .....	4
2.1.1 Struktur Dasar Kontrol Logika <i>Fuzzy</i> .....	5
2.1.2 Fungsi Keanggotaan.....	5
2.1.3 Kontroler Logika <i>Fuzzy</i> .....	6
2.1.3.1 Fuzzifikasi.....	7
2.1.3.2 Kaidah Aturan <i>Fuzzy</i> ( <i>Fuzzy Rule</i> ).....	7
2.1.3.3 Metode Inferensi MAX-MIN.....	8
2.1.3.4 Metode Defuzzifikasi <i>Center of Gravity</i> (COG) .....	9
2.2 Potensiometer Linier .....	10
2.3 Motor DC.....	10
2.4 <i>Pulse Width Modulation</i> (PWM) .....	12



2.5 Modul Pengendali Motor DC EMS 5A <i>H-Bridge</i> .....	12
2.6 Mikrokontroler Atmega 16.....	14
<b>BAB III METODOLOGI</b>	
3.1 Perancangan Sistem .....	16
3.4 Realisasi Pembuatan Sistem.....	16
3.5 Pengujian dan Analisa Data.....	17
3.6 Pengambilan Kesimpulan dan Saran.....	17
<b>BAB IV PERANCANGAN DAN PEMBUATAN ALAT</b>	
4.1 Perancangan Sistem .....	18
4.1.1 Blok Diagram Sistem.....	18
4.1.2 Prinsip Kerja Alat .....	19
4.2 Perancangan Perangkat Keras (hardware).....	20
4.2.1 Spesifikasi Alat.....	20
4.2.2 Potensiometer Linear .....	22
4.2.3 Modul Mikrokontroler AVR Atmega 16.....	22
4.2.4 Perancangan <i>Driver</i> Motor DC.....	24
4.3 Perancangan Sistem Kontrol Logika <i>Fuzzy</i> .....	25
4.3.1 Variabel Masukan dan Keluaran .....	25
4.3.2 Fungsi Keanggotaan Masukan .....	25
4.3.3 Fungsi Keanggotaan Keluaran .....	26
4.3.4 Perancangan Aturan <i>Fuzzy</i> .....	27
4.3.5 Metode Inferensi MAX-MIN .....	27
4.3.6 Defuzzifikasi .....	28
4.4 Perancangan Pergerakan Model <i>Rotary Parking</i> .....	29
4.5 Perancangan Perangkat Lunak (software).....	32
<b>BAB V PENGUJIAN DAN ANALISIS SISTEM</b>	

5.1 Pengujian Potensiometer .....	33
5.2 Pengujian Driver Motor DC .....	37
5.3 Pengujian Kecepatan .....	38
5.4 Pengukuran Letak Posisi Akhir .....	39
5.5 Pengujian Keseluruhan .....	40
5.5.1 Pengujian Tanpa Kontroler .....	40
5.5.2 Pengujian Satu <i>Set Point</i> .....	42
5.5.3 Pengujian Dengan <i>Set Point</i> Berbeda .....	48
<b>BAB IV PENUTUP</b>	
6.1 Kesimpulan .....	50
6.2 Saran .....	50
<b>DAFTAR PUSTAKA</b> .....	51
<b>LAMPIRAN</b> .....	52



**DAFTAR TABEL**

Tabel 4.1 Fungsi Pin Mikrokontroler.....23

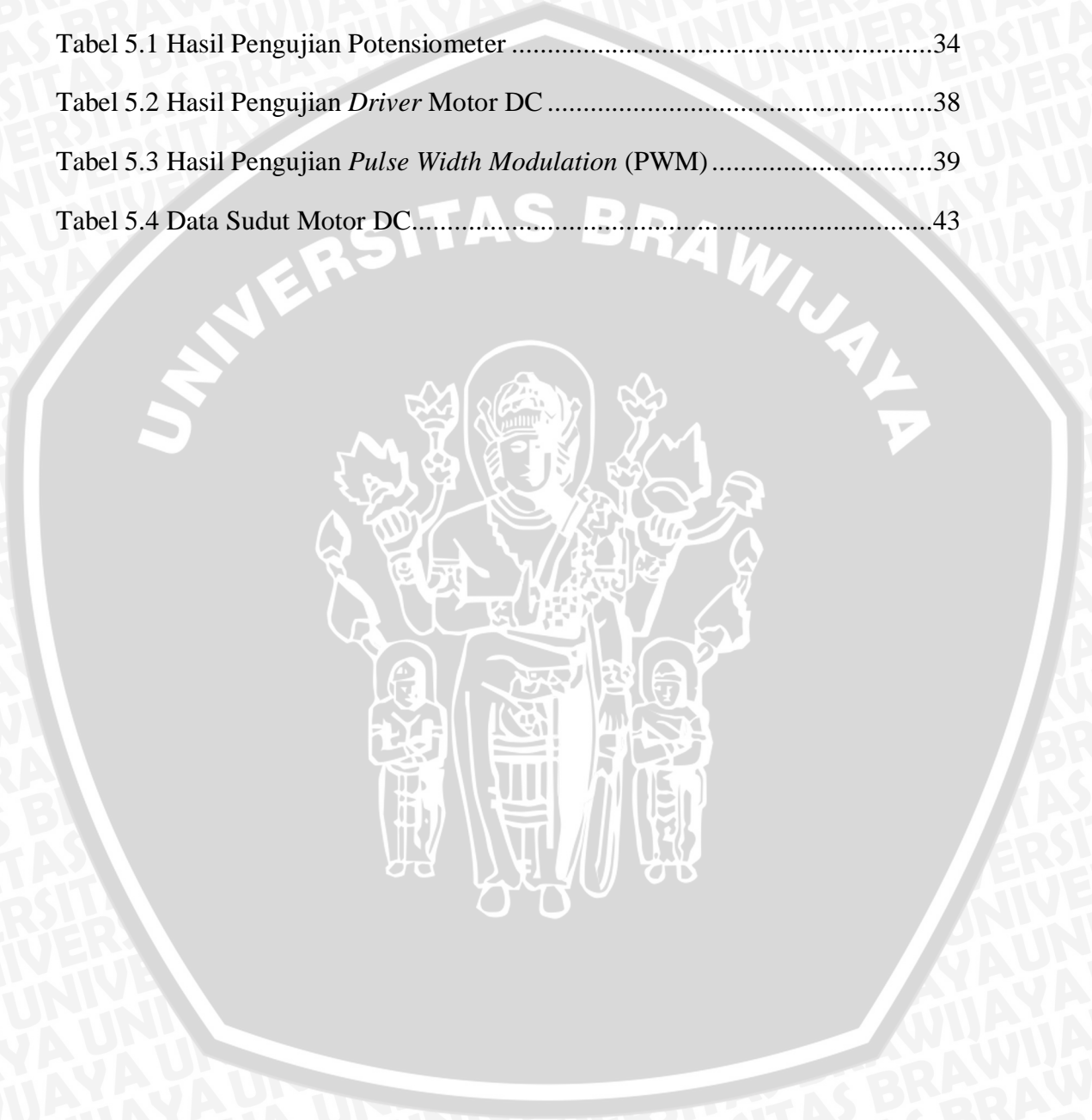
Tabel 4.2 Aturan *Fuzzy* .....27

Tabel 5.1 Hasil Pengujian Potensiometer .....34

Tabel 5.2 Hasil Pengujian *Driver* Motor DC .....38

Tabel 5.3 Hasil Pengujian *Pulse Width Modulation (PWM)* .....39

Tabel 5.4 Data Sudut Motor DC.....43





## DAFTAR GAMBAR

Gambar 2.1 Pengendali <i>Fuzzy</i> .....	5
Gambar 2.2 Fungsi Keanggotaan Bentuk <i>Triangular</i> .....	6
Gambar 2.3 Fungsi Keanggotaan Bentuk Trapesium.....	6
Gambar 2.4 Inferensi <i>Fuzzy</i> Dengan Metode MAX-MIN .....	9
Gambar 2.5 Ilustrasi Motor DC.....	11
Gambar 2.6 Sinyal PWM Secara Umum .....	12
Gambar 2.7 Modul Rangkaian <i>Driver</i> Motor EMS 5A H-Bridge.....	13
Gambar 2.8 Koneksi Modul EMS 5A H-Bridge .....	14
Gambar 2.9 Konfigurasi Pin ATmega 16.....	14
Gambar 2.10 Ilustrasi Motor DC <i>Brushed</i> .....	11
Gambar 2.11 Sinyal PWM Secara Umum .....	12
Gambar 2.12 Koneksi Modul EMS 5A H-Bridge .....	13
Gambar 2.13 Pengendali Motor DC Menggunakan EMS 5A H-Bridge .....	13
Gambar 2.14 Arsitektur AVR.....	15
Gambar 2.16 Konfigurasi PIN Atmega 16.....	16
Gambar 4.1 Diagram Balok Sistem.....	18
Gambar 4.2 Skema Sistem Rotary Parking Tampak Samping.....	21
Gambar 4.3 Skema Sistem Rotary Parking Tampak Depan .....	21
Gambar 4.4 Skema Sistem Rotary Parking Tampak Atas .....	21
Gambar 4.5 Rangkaian Potensiometer.....	22
Gambar 4.6 Potensiometer pada Sistem .....	22
Gambar 4.7 Minimum Sistem Mikrokontroler Atmega 16.....	23
Gambar 4.8 Modul Mikrokontroler Atmega 16 .....	24
Gambar 4.9 Koneksi Modul EMS 5A H-Bridge .....	24
Gambar 4.10 Modul Rangkaian Driver EMS 5A H-Bridge .....	25

Gambar 4.11 Fungsi Keanggotaan Masukan Error .....	26
Gambar 4.12 Fungsi Keanggotaan Masukan Delta Error .....	26
Gambar 4.13 Fungsi Keanggotaan Keluaran .....	27
Gambar 4.14 Proses Inferensi MAX-MIN .....	28
Gambar 4.15 Fungsi Keanggotaan Keluaran .....	28
Gambar 4.16 Pergerakan Ketika Push Button dan Limit Switch yang Dipilih Bernomor 2 Saat Garasi 1 Sebagai Acuan.....	30
Gambar 4.17 Pergerakan Ketika Push Button dan Limit Switch yang Dipilih Bernomor 1 Saat Garasi 2 Sebagai Acuan.....	31
Gambar 4.18 Pergerakan Ketika Push Button dan Limit Switch yang Dipilih Bernomor 1 Saat Garasi 1 Sebagai Acuan.....	31
Gambar 4.19 <i>Flowchart</i> Perangkat Lunak Sistem.....	32
Gambar 5.1 Pengujian Potensiometer.....	33
Gambar 5.2 Grafik Hubungan Antara Sudut dengan Resistansi .....	35
Gambar 5.3 Grafik Hubungan Antara Sudut dengan Tegangan.....	35
Gambar 5.4 Grafik Perbandingan Antara Tegangan Perhitungan dengan Tegangan Pengukuran terhadap Sudut .....	36
Gambar 5.5 Grafik Perbandingan Antara Tegangan Perhitungan dengan Tegangan Pengukuran terhadap Resistansi.....	36
Gambar 5.6 Diagram Blok Pengujian <i>Driver</i> Motor DC.....	38
Gambar 5.7 Pergerakan Garasi 2 Menuju Garasi Acuan Atau Garasi Dasar.....	40
Gambar 5.8 Respon Tegangan Masukan Motor dengan <i>Set Point</i> 60 (ADC 204) Tanpa Kontroler .....	41
Gambar 5.9 Respon Respon Tegangan Masukan Motor dengan <i>Set Point</i> 60 (ADC 204) Dengan Kontroler.....	42
Gambar 5.10 Respon Sistem dengan <i>Set Point</i> 60 (ADC 204).....	44

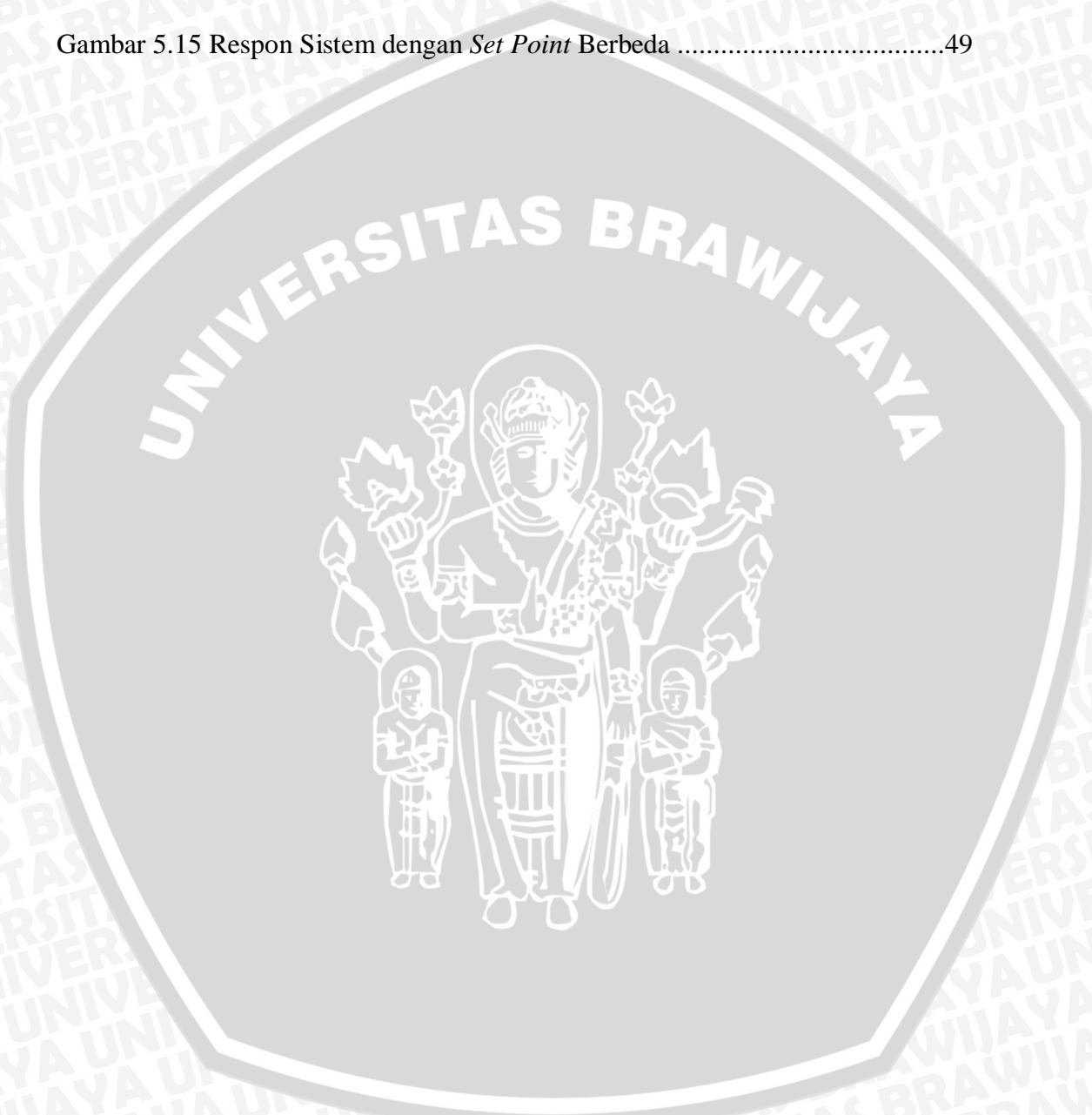
Gambar 5.11 Respon Sistem dengan *Set Point* 120 (ADC 306) .....45

Gambar 5.12 Respon Sistem dengan *Set Point* 180 (ADC 408) .....46

Gambar 5.13 Respon Sistem dengan *Set Point* 240 (ADC 510) .....47

Gambar 5.14 Respon Sistem dengan *Set Point* 300 (ADC 612) .....48

Gambar 5.15 Respon Sistem dengan *Set Point* Berbeda .....49



## ABSTRAK

**Risa Pradita Camelia**, Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya, Agustus 2012, Aplikasi Kontrol Logika *Fuzzy* untuk Pengaturan Posisi Sudut Putar Motor DC pada Model Sistem *Rotary Parking*, Dosen Pembimbing: Fitriana Suhartati, ST.,MT., Erni Yudaningtyas, Ir., MT.,Dr.

Saat ini, pengguna mobil semakin meningkat, sedangkan luas lahan parkir tidak bertambah. Salah satu solusi untuk mengatasi permasalahan tersebut adalah pembuatan sistem rotary parking, yaitu sistem parkir otomatis dengan menggunakan lahan yang relatif kecil namun dapat menampung mobil dengan jumlah yang lebih banyak. Salah satu sistem kontrol yang dapat digunakan pada model rotary parking ialah Kontrol Logika *Fuzzy* (KLF). Salah satu keuntungan sistem KLF adalah sistem yang sederhana sehingga lebih cepat dalam perancangan serta mudah dalam analisis. Pada penelitian ini digunakan metode Mamdani pada proses pengambilan keputusan. Dalam perencanaan pembuatannya digunakan mikrokontroler Atmega 16, potensiometer linier, motor DC, komunikasi serial RS-232 dan berat objek yang diibaratkan sebuah mobil. Hasil pengujian terhadap aplikasi kontroler fuzzy ini menunjukkan bahwa respon sistem untuk pergerakan garasi 1,2,3,4,5,6 mempunyai error *steady state* sebesar 0-0,833%. Sedangkan  $T_s$   $60^\circ = 1,665$  s, untuk  $T_s$   $120^\circ = 9,324$  s,  $T_s$   $180^\circ = 11,988$  s,  $T_s$   $240^\circ = 13,986$  s,  $T_s$   $300^\circ = 15,318$  s. Hasil pengujian ini menunjukkan bahwa kontroler *fuzzy* menghasilkan respon sesuai yang direncanakan dan dapat diaplikasikan pada model sistem *rotary parking* untuk mengontrol posisi sudut putar motor DC.

**Kata kunci** : *Rotary Parking*, Kontrol Logika *Fuzzy*, Motor DC.

## BAB I PENDAHULUAN

### 1.1 Latar Belakang

Seiring dengan perkembangan zaman, alat transportasi pribadi semakin berkembang pesat, salah satunya adalah mobil. Karena bentuk dan modelnya yang beraneka ragam, serta harga yang semakin terjangkau sehingga membuat alat transportasi ini semakin banyak digunakan. Akan tetapi saat ini penambahan jumlah pengguna mobil tidak disertai dengan penambahan luas lahan parkir yang tersedia terutama pada daerah kota besar, sehingga sering menimbulkan masalah bagi pengguna mobil.

Salah satu solusi untuk mengatasi permasalahan di atas adalah pembuatan sistem *rotary parking*, yaitu sistem parkir otomatis dengan menggunakan lahan yang relatif kecil namun dapat menampung mobil dengan jumlah yang lebih banyak. Sistem ini mampu mengatasi keterbatasan jumlah lahan parkir yang relatif sempit. Prinsip kerja dari sistem ini adalah mobil diparkir ke dalam garasi lewat sebuah gerbang dan digerakkan ke atas sementara garasi yang kosong diarahkan ke dasar agar pengguna parkir yang lain dapat menggunakan garasi yang kosong dengan cepat. Sedangkan untuk pengambilan mobil yang telah diparkir, telah disediakan *push-botton* untuk masing-masing garasi. Hal ini memudahkan pengguna karena pengguna dapat menekan tombol tersebut sesuai dengan nomor garasi parkir mobinya.

Pada perancangan sebelumnya telah dibuat sistem *rotary parking* dengan kontrol *on-off* dengan jumlah garasi empat buah, namun sistem *rotary parking* tersebut hanya bisa bergerak searah. Dalam penelitian ini digunakan kontrol logika *fuzzy* karena dalam perancangan tidak diperlukan model matematis sistem. Hal ini disebabkan karena kontrol logika *fuzzy* bekerja berdasarkan rule-rule yang diekstrak sesuai dengan pemikiran dan pengetahuan manusia baik sebagai operator atau ahli (Thiang, 2001).

Pada penelitian ini akan dibuat model *rotary* yang bisa menampung enam buah mobil dan bisa bergerak ke dua arah. Sehingga diharapkan membuat sistem ini bekerja lebih efektif daripada sistem yang sebelumnya telah dibuat, karena

selain dapat menampung mobil yang lebih banyak, pengguna juga tidak harus menunggu lama saat proses pengambilan mobil. Selain itu kecepatan dalam pergerakan garasi disesuaikan dengan posisi garasi tersebut menuju kecepatan yang diinginkan (*set point*), sehingga garasi bergerak dengan halus.

Sistem *rotary parking* ini membutuhkan alat yang dapat menggerakkan posisi penyimpanan mobil. Salah satu alat yang dapat digunakan sebagai penggerak adalah motor arus searah atau yang biasa disebut *Direct Current Motor* (Motor DC).

## 1.2 Rumusan Masalah

Mengacu pada permasalahan yang disebutkan pada latar belakang, maka ditetapkan rumusan masalah sebagai berikut :

- 1) Merancang sistem *rotary parking* dengan sistem elektriknya menggunakan modul mikrokontroler Atmega 16 dengan bahasa pemrograman CV-AVR.
- 2) Mengimplementasikan kontrol logika *fuzzy* untuk pergerakan motor DC dalam pengaturan posisi sudut putar motor DC pada model sistem *rotary parking*.

## 1.3 Ruang Lingkup

Pada skripsi ini dibatasi oleh hal-hal sebagai berikut:

- 1) Sistem parkir otomatis yang dibuat merupakan sebuah model yang dibuat seperti aslinya.
- 2) Pembahasan mekanik, model matematis, serta rangkaian elektronika tidak dibahas secara mendalam, pembahasan ditekankan pada penggunaan kontrol logika *fuzzy* pada sistem.

## 1.4 Tujuan

Tujuan dari skripsi ini adalah mengembangkan perancangan yang sebelumnya, yaitu merancang model sistem *rotary parking* yang dapat berputar dua arah dan mengimplementasikan kontrol logika *fuzzy* untuk pergerakan motor DC dalam pengaturan posisi sudut putar motor DC pada model sistem *rotary parking*.

## 1.5 Sistematika Pembahasan

Skripsi ini terdiri dari enam bagian dengan sistematika sebagai berikut:

### **BAB I      Pendahuluan**

Membahas latar belakang, rumusan masalah, ruang lingkup, tujuan dan sistematika pembahasan.

### **BAB II     Tinjauan Pustaka**

Membahas teori-teori yang mendukung dalam perancangan dan pembuatan alat.

### **BAB III    Metodologi Penulisan**

Membahas metode penelitian dan perencanaan alat.

### **BAB IV    Perancangan dan Pembuatan Alat**

Membahas perancangan kontroler menggunakan Kontrol Logika *Fuzzy* (KLF) dan menerapkannya ke dalam *software* di komputer, yang selanjutnya di tulis ke mikrokontroler sehingga sistem dapat bekerja dengan baik.

### **BAB V     Pengujian dan Analisis**

Membahas hasil pengujian sistem yang sudah dibuat, serta analisis hasil yang diperoleh.

### **Bab VI    Kesimpulan dan Saran**

Membahas kesimpulan perancangan ini dan saran-saran yang diperlukan untuk pengembangan selanjutnya.

## BAB II

### TINJAUAN PUSTAKA

Untuk memudahkan dalam memahami cara kerja rangkaian maupun dasar-dasar perencanaan dari sistem yang dibuat, maka perlu adanya penjelasan dan uraian mengenai teori penunjang yang digunakan dalam penulisan tugas akhir ini. Teori penunjang yang akan dijelaskan dalam bab ini adalah:

- Kontrol logika *fuzzy*
- Sensor potensiometer linier
- Motor DC
- Modul mikrokontroler AVR ATmega 16
- Modul pengendali motor DC

#### 2.1 Kontrol Logika *Fuzzy*

*Fuzzy* secara harfiah berarti samar, sedangkan kebalikannya dalam hal ini adalah *Crisp* yang secara harfiah berarti tegas. Dalam kehidupan sehari-hari nilai samar lebih akrab daripada nilai tegas. Temperatur tertentu biasa dinyatakan sebagai panas, agak panas, atau sangat dingin daripada dinyatakan dalam nilai terukur tertentu.

Tahun 1965 L.A. Zadeh memodifikasi teori himpunan yang disebut himpunan kabur (*fuzzy Set*). Himpunan *fuzzy* di dasarkan pada gagasan untuk memperluas jangkauan fungsi karakteristik sehingga fungsi tersebut akan mencakup bilangan real pada interval  $[0,1]$ . Nilai keanggotaannya menunjukkan bahwa suatu nilai dalam semesta pembicaraan tidak hanya berada pada 0 atau 1, namun juga nilai yang terletak diantaranya. Dengan kata lain nilai kebenaran suatu hal tidak hanya bernilai benar atau salah. Nilai 0 menunjukkan salah, nilai 1 menunjukkan benar dan masih ada nilai-nilai yang terletak diantaranya.

Sejak tahun 1982 pengendalian berbasis logika *fuzzy* mengalami perkembangan pesat, terutama dalam hubungannya dengan penyelesaian masalah kendali yang bersifat tak linier, sulit dimodelkan, berubah karakteristiknya terhadap waktu (*time varying*) dan kompleks (Sivanandam, 2006).



### 2.1.1 Struktur Dasar Kontrol Logika Fuzzy

Dalam sistem pengendalian dengan logika *fuzzy* melibatkan suatu blok pengendali yang menerima satu atau lebih masukan dan mengumpangkan satu atau lebih keluaran ke plant atau blok lain sebagaimana ditunjukkan dalam Gambar 2.1.



Gambar 2.1 Pengendali Fuzzy

Sumber : Coughanowr,1991

Komponen utama penyusun kontrol logika *fuzzy* adalah unit fuzzifikasi, *fuzzy inference*, dan unit defuzzifikasi. Basis pengetahuan terdiri dari dua jenis (Yan, 1994).

### 2.1.2 Fungsi Keanggotaan

Fungsi keanggotaan menotasikan nilai kebenaran anggota-anggota himpunan *fuzzy*. Interval nilai yang digunakan untuk menentukan fungsi keanggotaan, yaitu nol dan satu. Tiap fungsi keanggotaan memetakan elemen himpunan *crisp* ke semesta himpunan *fuzzy*.

Suatu himpunan *fuzzy* A dalam semesta pembicaraan U dinyatakan dengan fungsi keanggotaan,  $\mu_A$  yang harganya berada dalam interval [0,1]. Secara matematika hal ini dinyatakan dengan :

$$\mu_A : U \rightarrow [0,1]$$

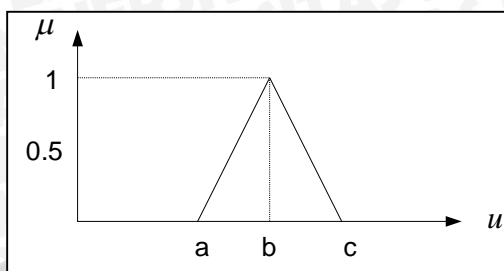
(Kuswadi, 2000)

- Fungsi keanggotaan bentuk Triangular

Definisi fungsi triangular sebagai berikut:

$$T(u;a,b,c) = \begin{cases} 0 & u < a \\ \frac{u-a}{b-a} & a \leq u \leq b \\ \frac{c-u}{c-b} & b \leq u \leq c \\ 0 & u > c \end{cases} \quad (2.1)$$

Fungsi keanggotaan bentuk Triangular ditunjukkan dalam Gambar 2.2



Gambar 2.2 Fungsi Keanggotaan Bentuk Triangular

Sumber : Yan, 1994

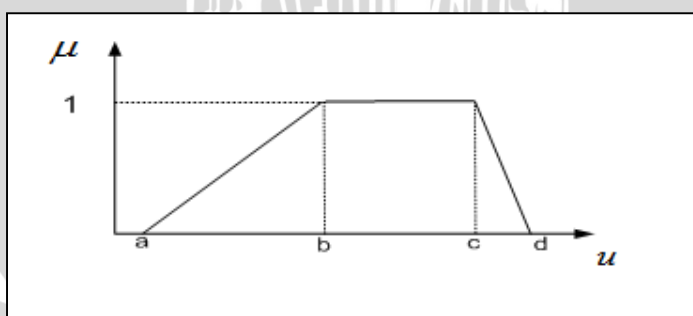
Fungsi keanggotaan bentuk triangular ini digunakan bila diinginkan himpunan *fuzzy* mempunyai nilai proporsional terhadap nol maupun satu.

➤ Fungsi keanggotaan bentuk Trapesium

Definisi fungsi trapesium sebagai berikut :

$$T(u; a, b, c, d) = \begin{cases} 0 & u < a \\ \frac{u - a}{b - a} & a \leq u \leq b \\ 1 & b \leq u \leq c \\ \frac{d - u}{d - c} & c \leq u \leq d \\ 0 & d \leq u \end{cases} \quad (2.2)$$

Fungsi keanggotaan bentuk Trapesium ditunjukkan pada Gambar 2.3



Gambar 2.3 Fungsi Keanggotaan Bentuk Trapesium

Sumber : Yan, 1994

### 2.1.3 Kontroler Logika *Fuzzy*

Kontroler logika *fuzzy* adalah sistem berbasis aturan (*rule based system*) yang didalamnya terdapat himpunan aturan *fuzzy* yang mempresentasikan



mekanisme pengambilan keputusan. Aturan yang dibuat digunakan untuk memetakan variabel input ke variabel output dengan pernyataan *If - Then*.

Kontroler ini akan menggunakan data tertentu (*crisp*) dari sejumlah sensor kemudian mengubahnya menjadi bentuk linguistik atau fungsi keanggotaan melalui proses fuzzifikasi. Lalu dengan aturan *fuzzy*, *inference engine* yang akan menentukan hasil keluaran *fuzzy*. Setelah itu hasil ini akan diubah kembali menjadi bentuk numerik melalui proses *defuzzifikasi*.

### 2.1.3.1 Fuzzifikasi

Proses fuzzifikasi merupakan proses untuk mengubah variabel non *fuzzy* (variabel numerik) menjadi variabel *fuzzy* (variabel linguistik). Nilai masukan-masukan yang masih dalam bentuk variabel numerik yang telah dikuantisasi sebelum diolah oleh pengendali logika *fuzzy* harus diubah terlebih dahulu ke dalam variabel *fuzzy*. Melalui fungsi keanggotaan yang telah disusun, maka dari nilai-nilai masukan tersebut menjadi informasi *fuzzy* yang berguna nantinya untuk proses pengolahan secara *fuzzy* pula. Proses ini disebut fuzzifikasi (Yan,1994). Proses fuzzifikasi diekspresikan sebagai berikut:

$$x = \text{fuzzifier}(x_0)$$

dengan:

$x_0$  = nilai *crisp* variabel masukan

$x$  = himpunan *fuzzy* variabel yang terdefinisi

*fuzzifier* = operator fuzzifikasi yang memetakan himpunan *crisp* ke himpunan *fuzzy*

Pedoman memilih fungsi keanggotaan untuk proses fuzzifikasi, menurut Jun Yan, menggunakan :

1. Himpunan *fuzzy* dengan distribusi simetris.
2. Gunakan himpunan *fuzzy* dengan jumlah ganjil, berkaitan erat dengan jumlah kaidah (*rules*).
3. Mengatur himpunan *fuzzy* agar saling menumpuk.
4. Menggunakan fungsi keanggotaan bentuk segitiga atau trapesium.

### 2.1.3.2 Kaidah Aturan Fuzzy (Fuzzy Rule)

*Fuzzy rule* adalah bagian yang menggambarkan dinamika suatu sistem terhadap masukan yang dikarakteristikan oleh sekumpulan variabel-variabel

linguistik dan berbasis pengetahuan seorang operator ahli. Pernyataan tersebut umumnya dinyatakan oleh suatu pernyataan bersyarat.

Dalam pengendali berbasis *fuzzy*, aturan pengendalian *fuzzy* berbentuk aturan “ IF – THEN”. Untuk sebuah sistem *Multi Input Single Output* (MISO) basis aturan pengendalian *fuzzy* berbentuk seperti berikut ini,

Rule 1 IF X is  $A_1$  AND Y is  $B_1$  THEN Z is  $C_1$

Rule 2 IF X is  $A_2$  AND Y is  $B_2$  THEN Z is  $C_2$

⋮  
⋮  
⋮  
⋮

Rule n IF X is  $A_n$  AND Y is  $B_n$  THEN Z is  $C_n$

Dengan X, Y, Z merupakan variabel linguistik, dimana X dan Y merupakan variabel masukan, dan Z merupakan variabel keluaran sistem.  $A_n$ ,  $B_n$ , dan  $C_n$  merupakan nilai linguistik dari X, Y, dan Z (Lee, 1990).

### 2.1.3.3 Metode Inferensi MAX-MIN

Metode inferensi merupakan proses untuk mendapatkan keluaran dari suatu kondisi masukan dengan mengikuti aturan-aturan yang telah ditetapkan. Keputusan yang didapatkan pada proses ini masih dalam bentuk *fuzzy* yaitu derajat keanggotaan keluaran.

Pada metode Max–Min aturan operasi minimum Mamdani digunakan untuk implikasi *fuzzy*. Persamaan aturan minimum adalah

$$\mu_{C_i} = \bigcup_1^n \alpha_i \wedge \mu_{C_i} \tag{2.3}$$

dengan  $\alpha_i = \mu_{A_i}(x_0) \wedge \mu_{B_i}(y_0)$

Sebagai contoh , terdapat dua basis kaidah atur *fuzzy*, yaitu :

$R_1$  : Jika x adalah  $A_1$  dan y adalah  $B_1$  maka z adalah  $C_1$

$R_2$  : Jika x adalah  $A_2$  dan y adalah  $B_2$  maka z adalah  $C_2$

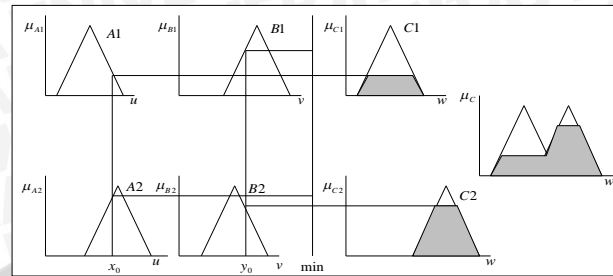
Pada metode penalaran MAX-MIN fungsi keanggotaan konsekuen dinyatakan dengan

$$\mu_{C_i}(W) = \mu_{C_i} \vee \mu_{C_i} = [\alpha_1 \wedge \mu_{C_1}(w)] \vee [\alpha_2 \wedge \mu_{C_2}(w)] \tag{2.4}$$

$$\text{dimana } \alpha_1 = \mu_{A_1}(x_0) \wedge \mu_{B_1}(y_0) \tag{2.5}$$

$$\alpha_2 = \mu_{A_2}(x_0) \wedge \mu_{B_2}(y_0) \tag{2.6}$$

Lebih jelas metode ini dideskripsikan dalam Gambar 2.4



Gambar 2.4 Inferensi *Fuzzy* dengan Metode MAX-MIN

Sumber : Yan, 1994

### 2.1.3.4 Metode Defuzzifikasi *Center Of Gravity* (COG)

Defuzzifikasi adalah proses untuk mendapatkan nilai numerik dari data *fuzzy* yang dihasilkan dari proses inferensi (Yan, 1994). Proses defuzzifikasi dinyatakan sebagai berikut :

$$y_0 = defuzzifier(y) \tag{2.7}$$

dengan:

- y : aksi kontrol *fuzzy*
- y<sub>0</sub> : aksi kontrol *crisp*
- defuzzifier : operator defuzzifikasi

#### ➤ Metode (*Center Of Gravity*)

Metode ini didefinisikan sebagai berikut:

$$U = \frac{\sum_{i=1}^n w_i u_i}{\sum_{i=1}^n w_i} \tag{2.8}$$

dengan:

- U = Keluaran
- w<sub>i</sub> = Bobot nilai benar w<sub>i</sub>
- u<sub>i</sub> = Nilai linguistik pada fungsi keanggotaan keluaran
- n = Banyak derajat keanggotaan



## 2.2 Potensiometer Linier

Potensiometer adalah resistor tiga terminal dengan sambungan geser yang membentuk pembagi tegangan dapat disetel. Jika hanya dua terminal yang digunakan (salah satu terminal tetap dan terminal geser), potensiometer berperan sebagai resistor variabel atau rheostat. Potensiometer biasanya digunakan untuk mengendalikan peranti elektronik seperti pengendali suara pada penguat.

Potensiometer linier sama halnya dengan potensiometer biasanya, di mana akan ada perubahan resistansi yang terjadi antara tiga kakinya apabila terjadi pergeseran pada poros potensiometer linier tersebut, namun potensiometer linier memiliki kepresisian dan perubahan resistansi yang linier yang terdapat diantara tiga kakinya apabila terjadi perderajat pergeseran pada poros potensiometer linier tersebut.

## 2.3 Motor DC

Prinsip kerja motor DC sesuai dengan hukum kemagnetan Lorenz, yaitu membangkitkan fluksi magnet pada suatu konduktor berarus dalam medan magnet sehingga timbul ggl induksi. Setiap arus yang mengalir melalui sebuah konduktor akan menimbulkan medan magnet. Arah medan magnet dapat ditentukan dengan kaidah tangan kiri.

Kaidah tangan kiri untuk motor menunjukkan arah arus yang mengalir didalam sebuah konduktor yang berada dalam medan magnet. Jari tengah menunjukkan arah arus yang mengalir pada konduktor, jari telunjuk menunjukkan arah medan magnet dan ibu jari menunjukkan arah gaya putar. Adapun besarnya gaya yang bekerja pada konduktor tersebut dapat dirumuskan dengan :

$$F = B \cdot I \cdot L \text{ (Newton)} \quad (2.9)$$

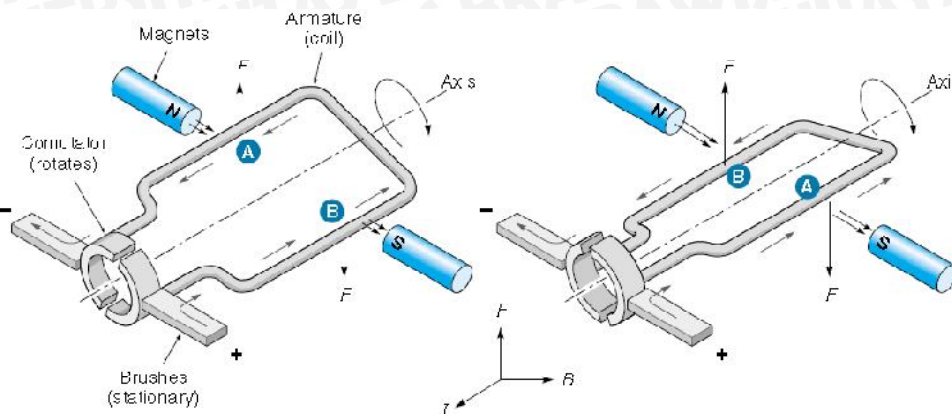
Dimana :

B = kerapatan fluks magnet (weber)

L = panjang konduktor (meter)

I = arus listrik (ampere)

(Soemarwanto, 1999)



**Gambar 2.5 Ilustrasi Motor DC**

Sumber : Kilian, 1996

Gambar 2.5 adalah ilustrasi cara kerja motor DC yang mempunyai satu lilit kawat a–b berada di dalam medan magnet. Lilitan ini dapat berputar dengan bebas, lilitan ini biasa disebut dengan jangkar (*armature*).

Pada jangkar diberikan arus yang berasal dari sumber yang terhubung dengan sikat (*brushes*). Sikat-sikat ini terpasang pada sebuah cincin yang terbelah dua, yang disebut cincin belah (*comutator*). Adapun tujuan dari konstruksi ini adalah agar lilitan kawat dapat berputar apabila ada arus listrik yang melewatinya.

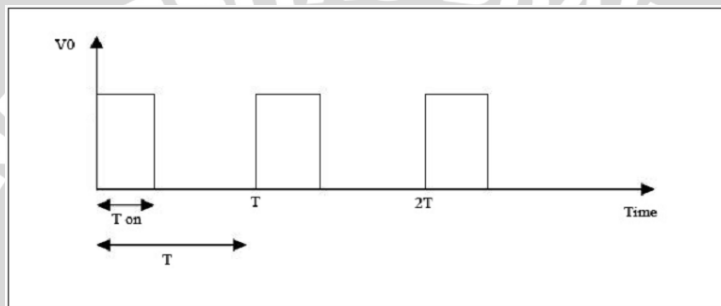
Pada kawat yang berada di kanan arus mengalir dari depan ke belakang. Pada kawat yang berada di bagian kiri, arus mengalir dari belakang ke depan kawat a dan b secara bergantian berada di kiri dan kanan. Karena itu arah arus di a dan arah arus di b selalu bersifat bolak-balik. Pembalikan arah arus itu terjadi pada saat lilitan kawat melintasi posisi vertikal.

Bagian *comutator* berfungsi sebagai penyearah mekanik. Fluksi magnet yang ditimbulkan magnet permanen disebut medan magnet motor. Dalam Gambar 2.5 arah fluk magnetik adalah dari kiri ke kanan. Adapun gaya yang bekerja pada penghantar b adalah ke atas, sementara gaya yang bekerja pada penghantar a adalah ke bawah . Gaya-gaya yang bekerja sama kuatnya, sehingga terdapat kopel yang bekerja pada kawat sehingga lilitan jangkar dapat berputar. Setelah berputar  $180^\circ$  arah arus berbalik, pada saat itu penghantar a dan penghantar b bertukar tempat. Akibatnya arah gerak putaran tidak berubah.

## 2.4 Pulse Width Modulation (PWM)

Pulse Width Modulation (PWM) digunakan untuk mengatur kecepatan dari motor DC. Kecepatan motor DC tergantung pada besarnya *duty cycle* yang diberikan pada motor DC tersebut. *Duty cycle* adalah rasio atau perbandingan lama waktu suatu sinyal berada dalam kondisi *high* dengan lama waktu suatu sinyal tersebut dalam kondisi (*high+low*).

Pada sinyal PWM, frekuensi sinyal konstan sedangkan *duty cycle* bervariasi dari 0%-100%. Dengan mengatur *duty cycle* akan diperoleh keluaran yang diinginkan. Sinyal PWM secara umum dapat dilihat dalam Gambar 2.6



Gambar 2.6 Sinyal PWM Secara Umum

Sumber: [www.electronics-scheme.com](http://www.electronics-scheme.com)

$$Duty\ cycle = \frac{T_{on}}{T} \times 100\% \dots (\%) \quad (2.9)$$

Dengan :

$$\begin{aligned} T_{on} &= \text{Periode logika tinggi} \\ T &= \text{Periode keseluruhan} \end{aligned}$$

$$V_{dc} = Duty\ cycle \times V_{cc} \dots (V) \quad (2.10)$$

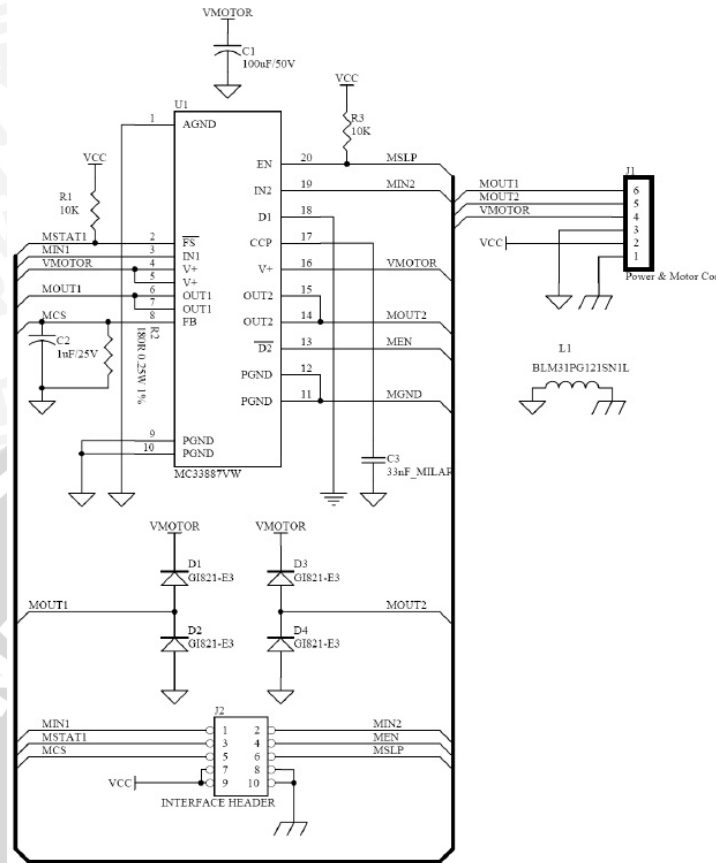
Sedangkan frekuensi sinyal dapat ditentukan dengan rumus berikut

$$f_{on} = \frac{f_{clk} / 0}{N \cdot 256} \dots (Hz) \quad (2.11)$$

## 2.5 Modul Pengendali Motor DC EMS 5A H-Bridge

*Embedded Module Series* (EMS) 5A H-Bridge merupakan *driver* H-Bridge yang didisain untuk menghasilkan *drive* 2 arah dengan arus kontinyu sampai dengan 5A pada tegangan 5 Volt sampai 40 Volt. Modul ini dilengkapi dengan rangkaian sensor arus beban yang dapat digunakan sebagai umpan balik ke pengendali. Modul ini mampu men-*drive* beban induktif seperti motor DC.

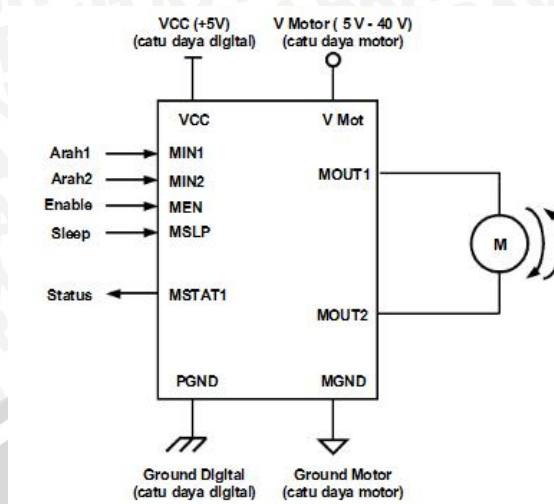




**Gambar 2.7 Modul Rangkaian Driver Motor EMS 5A H-Bridge**

Sumber: Datasheet EMS 5A H-Bridge

Sebuah modul H-Bridge 5A dapat digunakan untuk mengatur kerja 1 buah motor DC secara dua arah. Contoh koneksinya dapat dilihat dalam Gambar 2.8



Gambar 2.8 Koneksi Modul EMS 5A H-Bridge

Sumber: Datasheet EMS 5A H-Bridge

## 2.6 Mikrokontroler ATmega 16

Mikrokontroler ATmega 16 memiliki kapasitas 16 Kbyte Flash PEROM (*Flash Programmable and Erasable Read Only Memory*), 512 byte EEPROM, 1 Kbyte SRAM, serta 32 pin I/O. Tiap pin tersebut dapat diprogram secara paralel dan tersendiri, mempunyai dua buah *timer/counter* 8 bit dan satu buah *timer/counter* 16 bit, mempunyai 8 *channel* 10 bit ADC.

Masing-masing kaki dalam mikrokontroler ATmega 16 mempunyai fungsi tersendiri. Mikrokontroler ATmega8 mempunyai 28 pin, susunan masing-masing pin ditunjukkan dalam Gambar 2.9

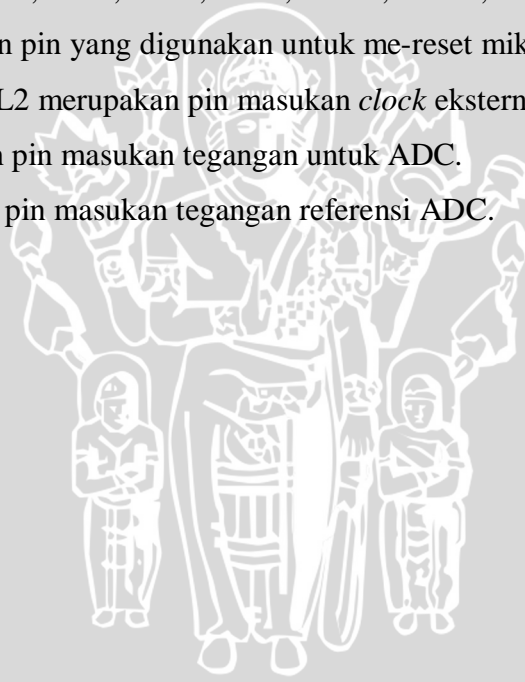
(XCK/T0) PB0	1	40	PAC (ADC0)
(T1) PB1	2	39	PA1 (ADC1)
(INT2/AIN0) PB2	3	38	PA2 (ADC2)
(OC0/AIN1) PB3	4	37	PA3 (ADC3)
(SS) PB4	5	36	PA4 (ADC4)
(MOSI) PB5	6	35	PA5 (ADC5)
(MISO) PB6	7	34	PA6 (ADC6)
(SCK) PB7	8	33	PA7 (ADC7)
RESET	9	32	AREF
VCC	10	31	GND
GND	11	30	AVCC
XTAL2	12	29	PC7 (TOSC2)
XTAL1	13	28	PC6 (TOSC1)
(RXD) PD0	14	27	PC5 (TDI)
(TXD) PD1	15	26	PC4 (TDO)
(NTD) PD2	16	25	PC3 (TMS)
(NTI) PD3	17	24	PC2 (TCK)
(CC13) PD4	18	23	PC <sup>+</sup> (SDA)
(CC1A) PD5	19	22	PG0 (SCL)
(CP1) PD6	20	21	PD7 (OC2)

Gambar 2.9 Konfigurasi Pin ATmega 16

Sumber: ATMEL, 2007

Fungsi kaki-kaki ATmega 16 adalah :

1. Vcc merupakan pin yang berfungsi sebagai masukan catu daya.
2. GND merupakan pin *Ground* .
3. Port A (PA0...7) merupakan pin *input/output* dua arah dan pin masukan ADC.
4. Port B (PB0...7) merupakan pin *input/output* dua arah dan pin dengan fungsi khusus seperti SPI, MISO, MOSI, SS, AIN1/OC0, AIN0/INT2, T1, T0 T1/XCK .
5. Port C (PC0...7) merupakan pin *input/output* dua arah dan pin dengan fungsi khusus, seperti TOSC2, TOSC1, TDI, TD0, TMS, TCK, SDA, SCL.
6. Port D (PD0...7) merupakan pin *input/output* dua arah dan pin dengan fungsi khusus, seperti RXD, TXD, INT0, INT1, OC1B, OC1A, ICP1.
7. RESET merupakan pin yang digunakan untuk me-reset mikrokontroler.
8. XTAL1 dan XTAL2 merupakan pin masukan *clock* eksternal.
9. AVCC merupakan pin masukan tegangan untuk ADC.
10. AREF merupakan pin masukan tegangan referensi ADC.



## BAB III METODOLOGI

Kajian dalam skripsi ini merupakan penelitian yang bersifat aplikatif, yaitu merancang suatu sistem pengendalian menggunakan kontroler *fuzzy* yang bertujuan agar dapat menampilkan performansi sistem sesuai dengan yang direncanakan.

Langkah-langkah yang perlu dilakukan untuk merealisasikan alat yang akan dibuat adalah sebagai berikut:

### 3.1 Perancangan Sistem

Perancangan sistem dilakukan sebagai langkah awal sebelum terbentuknya suatu sistem beserta rangkaian elektronik pendukungnya, hal ini dimaksudkan agar sistem pada model *rotary parking* tersebut dapat berjalan sesuai dengan deskripsi awal yang telah direncanakan. Perancangan sistem yang dilakukan meliputi:

- Penentuan motor DC yang digunakan dalam model *rotary parking*.
- Penentuan spesifikasi sistem yang akan dibuat, meliputi :
  - a. Penentuan deskripsi kerja sistem secara keseluruhan.
  - b. Kontroler Logika *Fuzzy* dan rangkaian elektronik pendukung.
- Perancangan program pada *Code Vision AVR ATmega16* sebagai pusat pengontrol dengan kontroler *fuzzy*.

### 3.2 Realisasi Pembuatan Sistem

Untuk merealisasikan alat atau sistem yang telah dirancang sebelumnya maka perlu diperhatikan beberapa tahapan dalam pembuatan alat tersebut, diantaranya yaitu :

- Penyempurnaan mekanik model sistem *rotary parking* secara keseluruhan yang meliputi penentuan batasan sudut putar motor DC serta pemasangan sensor dan motor.
- Pembuatan rangkaian elektronika meliputi rangkaian potensiometer dan penguat tegangan.

- Pengisian program yang telah dirancang sebelumnya pada *Code Vision AVR ATmega16* dengan kendali kontroler *fuzzy*.

### 3.3 Pengujian dan Analisa Data

Setelah semua komponen pada alat sudah terhubung sesuai dengan diagram blok sistem yang telah dirancang dan perangkat lunak untuk mendukung sistem telah dibuat, maka diadakan pengujian dan analisa alat. Metode pengujian alat adalah sebagai berikut

- Pengujian pada sensor yang digunakan.
- Menguji pada tiap-tiap blok rangkaian.
- Mengadakan pengujian sistem secara keseluruhan.
- Mengevaluasi hasil pengujian sistem secara keseluruhan.

### 3.4 Pengambilan Kesimpulan dan Saran

Kesimpulan diambil berdasarkan data yang diperoleh dari pengujian sistem secara keseluruhan. Jika hasil yang didapatkan telah sesuai dengan yang direncanakan sebelumnya, maka sistem kendali tersebut telah berhasil memenuhi harapan dan tentunya memerlukan pengembangan lebih lanjut untuk penyempurnaan.

## BAB IV

### PERANCANGAN DAN PEMBUATAN ALAT

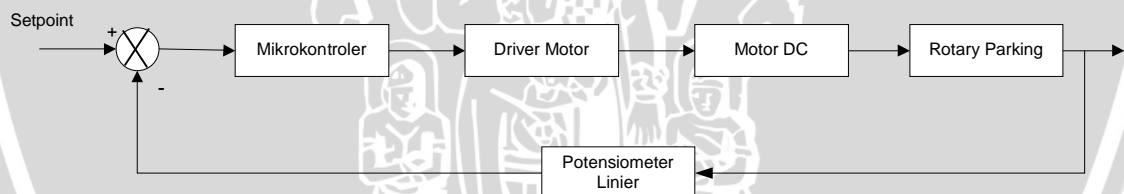
#### 4.1 Perancangan Sistem

Perancangan alat ini dilakukan secara bertahap dalam bentuk blok sehingga akan memudahkan dalam analisis pada setiap bloknya maupun secara keseluruhan. Perancangan ini terdiri dari:

- Cara kerja alat.
- Spesifikasi alat.
- Perancangan perangkat keras terdiri atas perancangan sensor potensiometer, modul pengendali motor DC, dan rangkaian mikrokontroler pengatur motor DC.
- Perancangan perangkat lunak terdiri atas perancangan algoritma kontrol logika *fuzzy* pada *Code Vision AVR* ATMega 16.

##### 4.1.1 Blok Diagram Sistem

Diagram blok sistem yang dirancang ditunjukkan dalam Gambar 4.1.



**Gambar 4.1 Diagram Balok Sistem**

Keterangan dari blok diagram diatas adalah sebagai berikut :

- *Set point* sistem berasal dari *push button* dan *limit switch*. *Set point* dari *push button* digunakan ketika proses pengambilan mobil, dan *set point* dari *limit switch* digunakan saat proses mengarahkan garasi kosong ke bagian dasar.
- Pusat pengendalian sistem menggunakan ATMega 16 yang memberikan keluaran berupa *duty cycle* PWM kepada *driver* motor.
- Motor yang digunakan adalah motor DC yang berfungsi sebagai aktuator. *Driver* motor menggunakan modul EMS 5A H-Bridge.

#### 4.1.2 Prinsip Kerja Alat

Cara kerja alat adalah sebagai berikut :

- Menggunakan catu daya sebesar 12 volt
- Potensiometer sebagai sensor posisi motor DC. Potensiometer akan memberikan keluaran berupa level tegangan yang berubah-ubah sesuai dengan posisi garasi pada *rotary parking*.
- Masukan sistem berasal dari *limit switch* dan *push button*. *Limit switch* dipasang di setiap garasi, berfungsi sebagai sensor pendeteksi ada tidaknya mobil pada suatu garasi, sehingga sebagai penentu *set point* pada saat proses memposisikan garasi kosong ke bawah. *Push button* berfungsi sebagai penentu *set point* pada proses pengambilan mobil. Seorang pengguna harus menekan *push button* yang bernomor sama dengan garasi tempat parkir mobilnya, sehingga garasi yang bernomor sesuai akan diposisikan di dasar.
- Ketika *limit switch* atau *push button* aktif maka mikrokontroler akan memproses *limit switch* mana atau *push button* yang aktif, kemudian menentukan *set point* dari sistem. Mikrokontroler akan membandingkan *set point* tersebut dengan besarnya tegangan yang diperoleh dari potensiometer linier, kemudian diproses dan diolah untuk menghasilkan data yang diinginkan. Selanjutnya mikrokontroler akan memerintahkan *driver* motor DC untuk menggerakkan motor ke kanan atau ke kiri dengan kecepatan sesuai dengan kondisi dan masukan dari *limit switch* ataupun *push button* dan potensiometer.
- Jika terdapat lebih dari satu *limit switch* yang mendeteksi bahwa tidak ada mobil di dalam garasi, maka *limit switch* yang dipilih adalah *limit switch* yang bernomor paling kecil.
- Proses tersebut berjalan terus sampai didapat posisi sudut putar motor DC yang diinginkan.
- Pada saat mobil sudah diparkir ke dalam garasi, maka pengguna harus keluar dari garasi dan menekan tombol *garage button* untuk memulai kerja sistem. Hal ini dilakukan untuk memastikan bahwa tidak ada orang

yang masih ada di dalam mobil, selain itu juga untuk memulai menggerakkan sistem megarahkan garasi yang kosong ke posisi dasar.

- Jika dalam penggunaannya terdapat 2 orang atau lebih secara bersamaan ingin mengambil mobilnya, maka kedua orang tersebut harus bergantian. Orang pertama terlebih dahulu menekan *push button* kemudian mengambil mobilnya, setelah itu orang berikutnya dapat melakukan langkah yang sama seperti orang pertama.

## 4.2 Perancangan Perangkat Keras (*hardware*)

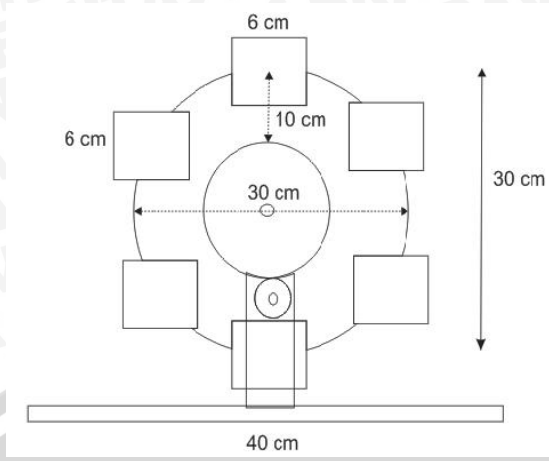
Berdasarkan diagram blok perancangan alat yang telah disusun, perancangan perangkat keras meliputi perancangan mekanik, sensor potensiometer dan *driver* motor serta penggunaan modul ATmega 16. Di bawah ini adalah penjelasan masing-masing rangkaian penyusun keseluruhan alat.

### 4.2.1 Spesifikasi Alat

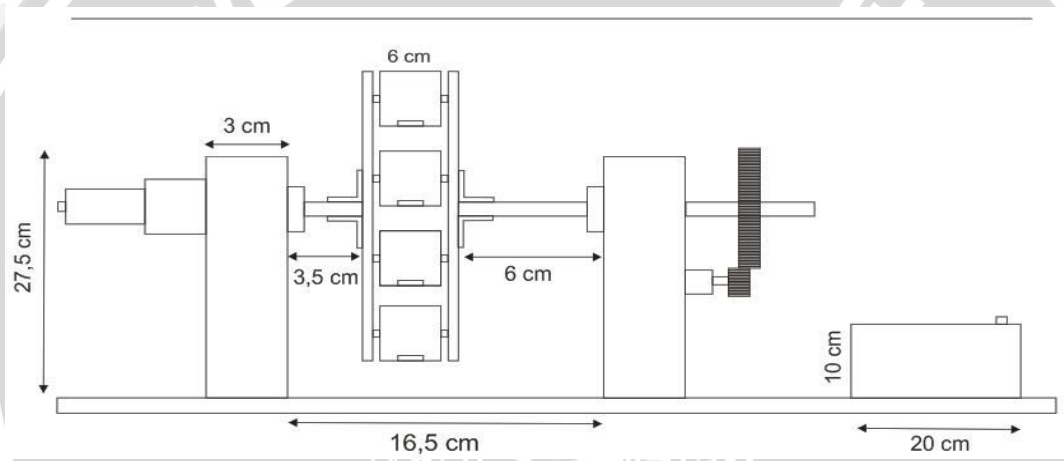
Spesifikasi model sistem *rotary parking* adalah sebagai berikut :

- Model sistem *rotary parking* yang dibuat terdiri 6 buah garasi yang dapat menampung 6 buah mobil.
- Pergerakan model *rotary parking* menggunakan motor DC dan menggunakan pipa aluminium sebagai poros penghubung antara model *rotary parking* dan motor DC.
- Potensiometer yang digunakan adalah potensiometer linier 10 K $\Omega$  yang dipasang sejajar dengan motor DC.
- Ukuran darimodel *rotary parking* dapat dilihat dalam Gambar 4.2, 4.3, dan 4.4.

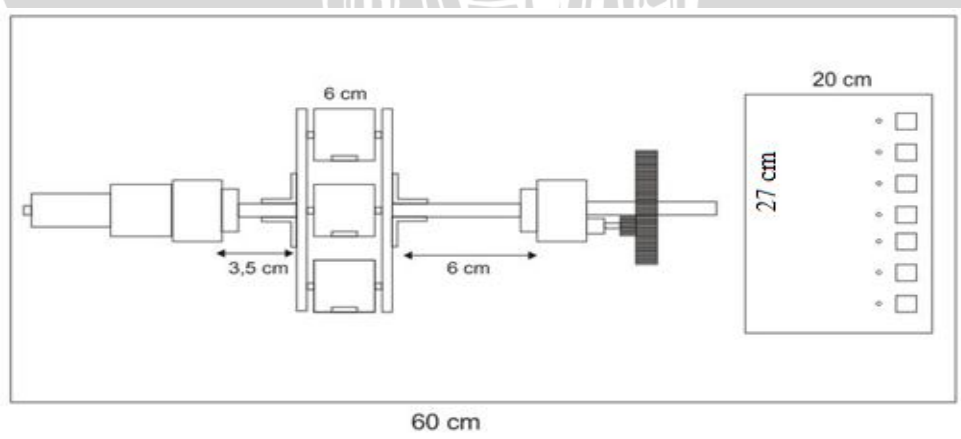




Gambar 4.2 Skema Model Rotary Parking Tampak Samping



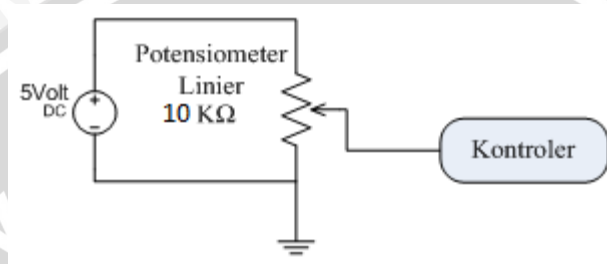
Gambar 4.3 Skema Model Rotary Parking Tampak Depan



Gambar 4.4 Skema Model Rotary Parking Tampak Atas

#### 4.2.2 Potensiometer Linier

Potensiometer linier digunakan sebagai pengukur posisi sudut putaran motor DC. Penempatan sensor ini tepat sejajar dengan motor DC, jadi setiap perubahan sudut putar dari motor DC akan ikut mempengaruhi perubahan resistansi pada kaki potensiometer linier. Sistem perancangan dan hasil perancangan potensiometer linier ditunjukkan pada gambar 4.5 dan 4.6 di bawah ini.



Gambar 4.5 Rangkaian Potensiometer

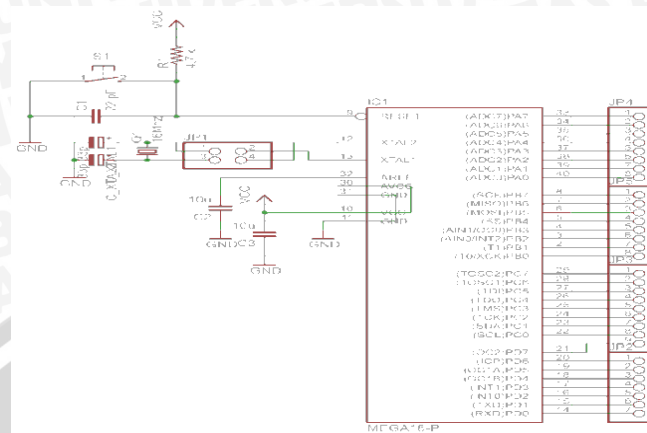


Gambar 4.6 Potensiometer Pada Sistem

#### 4.2.3 Modul Mikrokontroler AVR ATmega 16

Pada model *rotary parking* ini digunakan mikrokontroler ATmega 16 sebagai pengolah dalam proses pengaturan posisi motor DC dalam menggerakkan

model *rotary parking*. Konfigurasi kaki I/O dari mikrokontroler ATmega 16 ditunjukkan dalam Gambar 4.7

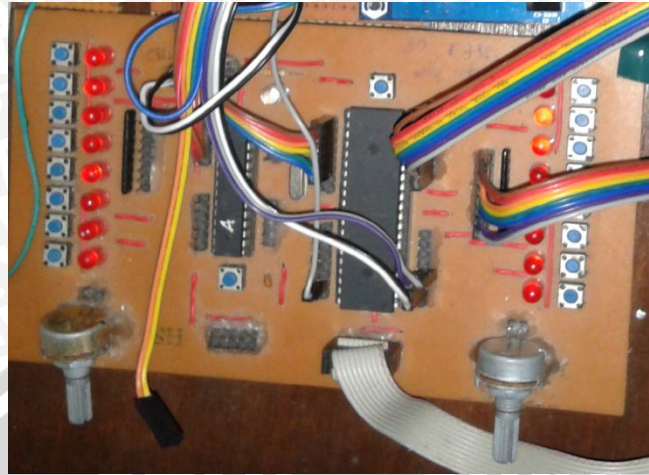


Gambar 4.7 Minimum Sistem Mikrokontroler ATmega 16

Mikrokontroler ATmega 16 mempunyai 4 port, 32 jalur yang dapat diprogram menjadi masukan atau keluaran. Pin masukan dan keluaran mikrokontroler pada perancangan ini akan difungsikan sesuai Tabel 4.1

Tabel 4.1 Fungsi Pin Mikrokontroler

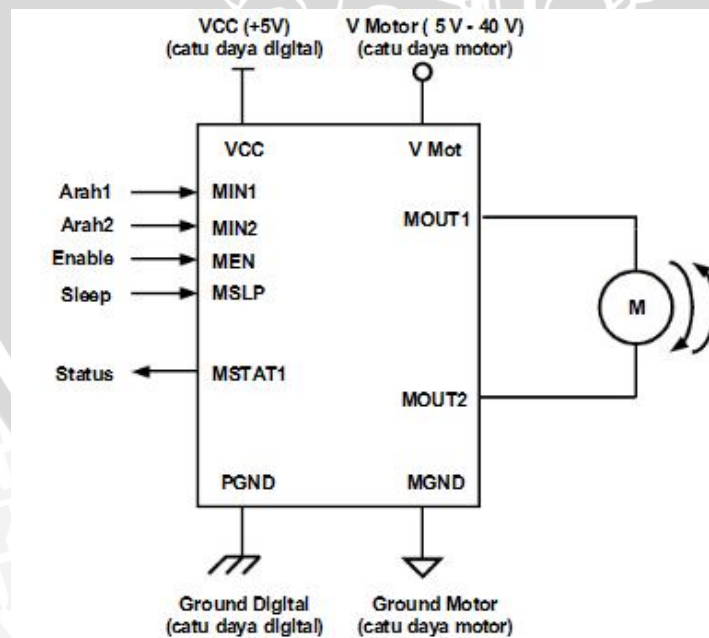
No	Pin	Fungsi
1	PA0	Jalur masukan potensiometer <i>rotary parking</i>
2	PB1	Jalur masukan pertama <i>driver motor</i>
3	PB2	Jalur masukan kedua <i>driver motor</i>
4	PC0	Jalur masukan <i>limit switch</i> pada garasi 1
5	PC1	Jalur masukan <i>limit switch</i> pada garasi 2
6	PC2	Jalur masukan <i>limit switch</i> pada garasi 3
7	PC3	Jalur masukan <i>limit switch</i> pada garasi 4
8	PC4	Jalur masukan <i>limit switch</i> pada garasi 5
9	PC5	Jalur masukan <i>limit switch</i> pada garasi 6
10	PD0	Jalur masukan <i>push button 1</i>
11	PD1	Jalur masukan <i>push button 2</i>
12	PD2	Jalur masukan <i>push button 3</i>
13	PD3	Jalur masukan <i>push button 4</i>
14	PD4	Jalur masukan <i>push button 5</i>
15	PD5	Jalur masukan <i>push button 6</i>
16	PD6	Jalur masukan <i>garage button</i>



Gambar 4.8 Modul Mikrokontroler AT Mega16

#### 4.2.4 Perancangan *Driver Motor DC*

Modul pengendali motor DC yang digunakan adalah modul EMS 5A H-Bridge. Secara garis besar, fungsi modul pengendali motor ini adalah untuk mengendalikan arah dan kecepatan putaran motor DC sesuai instruksi kendali dari mikrokontroler pengguna. Gambar koneksi modul pengendali motor DC EMS 5A H-Bridge ditunjukkan dalam Gambar 4.9



Gambar 4.9 Koneksi Modul EMS 5A H-Bridge

Rangkaian *driver* ini sanggup bekerja dengan tegangan maksimal 40 volt, serta kapasitas arus maksimum yang dapat dilewatkan pada modul ini sebesar 5 ampere.



Gambar 4.10 Modul Rangkaian Driver EMS 5A H-Bridge

### 4.3 Perancangan Sistem Kontrol Logika Fuzzy

#### 4.3.1 Variabel Masukan dan Keluaran

Sistem kontrol logika *fuzzy* yang dikembangkan dalam penelitian ini mempunyai dua *crisp input* yaitu *error* posisi dan *delta error* posisi serta satu *crisp output* yaitu perubahan tegangan. *Error* dan  $\Delta Error$ , didefinisikan dengan perumusan sebagai berikut:

$$Error(t) = SP - PV(t) \dots \dots \dots (1)$$

Dimana,

$SP$  = *Set point* (Nilai yang diinginkan)

$PV(t)$  = *Present Value* pada waktu  $t$  (Nilai aktual)

$$\Delta Error(t) = Error(t) - Error(t-1) \dots \dots \dots (2)$$

Dimana,

$Error(t)$  = *Error* pada waktu  $t$

$Error(t-1)$  = *Error* pada waktu  $(t-1)$

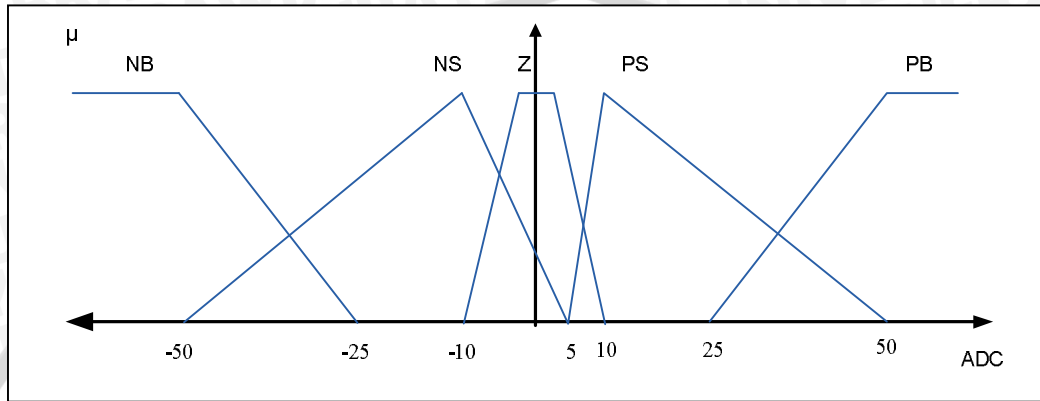
#### 4.3.2 Fungsi Keanggotaan Masukan

Fungsi keanggotaan masukan *error* posisi sudut putaran motor dan *delta error* posisi sudut putaran motor terdiri atas lima label yaitu *Negative Big* (NB), *Negative Small* (NS), *Zero* (Z), *Positive Small* (PS), *Positive Big* (PB). Data

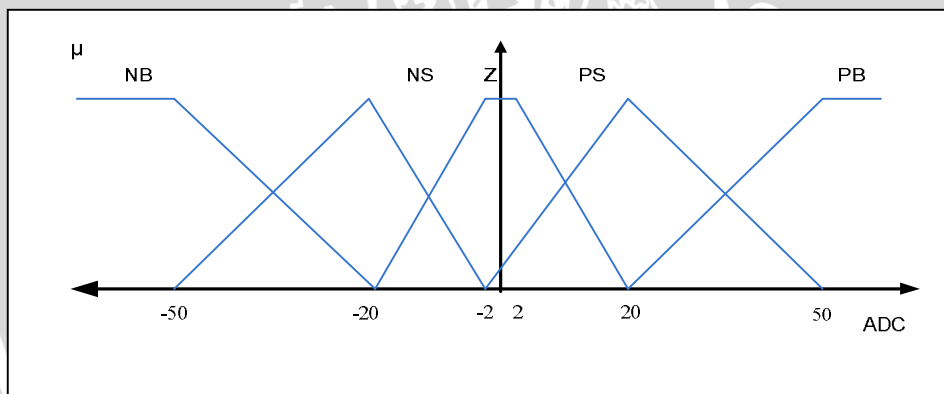


masukan dari fungsi keanggotaan didapat dari *limit switch* dan *push button* yang dibandingkan dengan hasil keluaran potensiometer linier.

Label masing-masing fungsi keanggotaan masukan berjumlah 5, agar responsnya tidak terlalu lambat tetapi tetap baik dalam pengendalian.



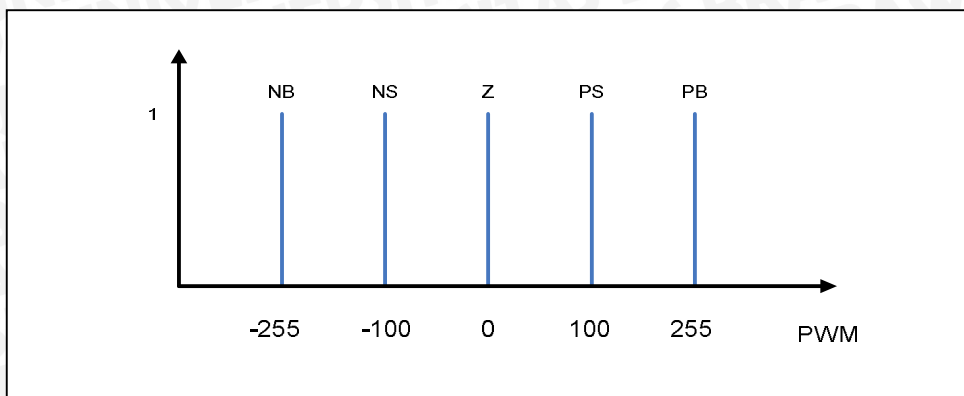
Gambar 4.11 Fungsi Keanggotaan Masukan Error



Gambar 4.12 Fungsi Keanggotaan Masukan Delta Error

### 4.3.3 Fungsi Keanggotaan Keluaran

Fungsi keanggotaan keluaran motor DC merupakan representasi arah putar dan besarnya nilai PWM yang mempengaruhi kecepatan putaran motor DC yang berbentuk *singleton*. Pemilihan bentuk fungsi keanggotaan berupa *singleton* dengan alasan agar mempercepat eksekusi program dan kapasitas program yang ditulis tidak melebihi kapasitas memori dari mikrokontroler.



Gambar 4.13 Fungsi Keanggotaan Keluaran Motor

**Keterangan:**

- NB : Motor berputar *Clock Wise* (CW) cepat
- NS : Motor berputar *Clock Wise* (CW) lambat
- Z : Motor Berhenti (*Stop*)
- PS : Motor berputar *Counter Clock Wise* (CCW) lambat
- PB : Motor berputar *Counter Clock Wise* (CCW) cepat

**4.3.4 Perancangan Aturan Fuzzy**

Tabel 4.2 Aturan Fuzzy Error

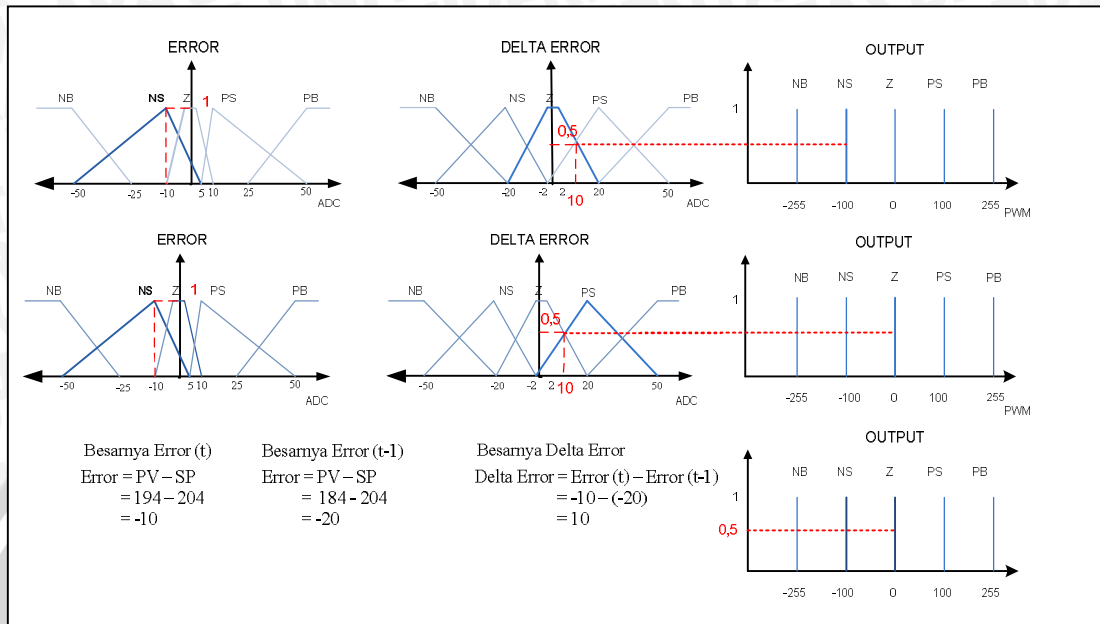
	NB	NS	Z	PS	PB
$\Delta$ Error	NB	NB	NS	NS	Z
NS	NB	NS	NS	Z	PS
Z	NS	NS	Z	PS	PS
PS	NS	Z	PS	PS	PB
PB	Z	PS	PS	PB	PB

**4.3.5 Metode Inferensi Max-Min**

Untuk mengetahui metode ini pada sistem, dimisalkan visualisasi proses pengolahan masukan ketika sensor potensiometer membaca posisi keluaran datanya 194 (data hasil dari ADC), kemudian data *set point* sebesar 204, dan



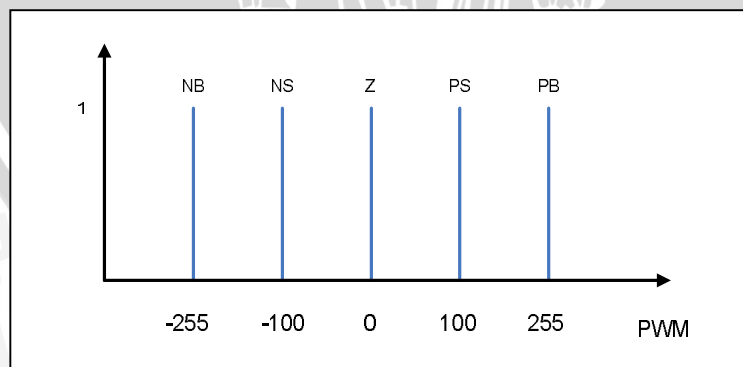
misalkan  $Error_{(t-1)}$  sebesar -20, maka dengan metode Max-Min ditetapkan ditunjukkan dalam Gambar 4.14



Gambar 4.14 Proses Inferensi Max-Min

### 4.3.6 Defuzzikasi

Defuzzifikasi adalah proses untuk mengubah keluaran *fuzzy* menjadi keluaran *crisp*. Hasil defuzzifikasi inilah yang digunakan untuk mengatur besarnya kecepatan pada masing-masing motor DC. Metode defuzzikasi yang digunakan adalah COG (*Center of Gravity*), dengan himpunan *fuzzy* keluaran *singleton* ditunjukkan dalam Gambar 4.15



Gambar 4.15 Fungsi Keanggotaan Keluaran

Keterangan:



- NB : Motor berputar CW (*Clock Wise*) cepat
- NS : Motor berputar CW (*Clock Wise*) lambat
- Z : Motor Berhenti (*Stop*)
- PS : Motor berputar CCW (*Counter Clock Wise*) lambat
- PB : Motor berputar CCW (*Counter Clock Wise*) cepat

Berdasarkan contoh hasil proses inferensi, maka keluaran motor didapatkan:

$$\begin{aligned}
 U &= \frac{\sum_{i=1}^n w_i u_i}{\sum_{i=1}^n w_i} \\
 &= \frac{0.5 \times NS + 0.5 \times Z}{0.5 + 0.5} \\
 &= \frac{0.5 \times (-100) + 0.5 \times (0)}{0.5 + 0.5} \\
 &= -50
 \end{aligned}$$

Jadi, pada saat model *rotary parking* bergerak, sensor potensiometer membaca data hasil ADC sebesar 194 yang menandakan *present value* (PV) dan dibandingkan terhadap *set point* (SP) sebesar 204, sehingga dari selisih PV dan SP ini diperoleh nilai *Error* ( $E_{(t)}$ ) sebesar (-10). Kemudian nilai *Error* aktual ( $E_{(t)}$ ) (-10) tersebut dibandingkan dengan nilai *Error* sebelumnya ( $E_{(t-1)}$ ) sebesar (-20), sehingga dari selisih ( $E_{(t)}$ ) dan ( $E_{(t-1)}$ ) diperoleh nilai *Delta Error* sebesar (10). Maka besarnya kecepatan putaran motor DC adalah (-50), artinya motor akan berputar berlawanan dengan arah jarum jam (*Counter Clock Wise*) dengan besarnya sinyal PWM 50.

#### 4.4 Perancangan Pergerakan Model *Rotary Parking*

Pergerakan model sistem *rotary parking* ditentukan dari posisi garasi 1, 2, 3, 4, 5, dan 6 menuju posisi dasar pada model *rotary parking*. Penentuan *set point* atau besarnya sudut yang harus ditempuh berdasarkan pengukuran manual terhadap titik-titik tujuan. Pengukuran yang diambil berupa besar sudut putar motor DC sebagai posisi garasi yang akan dituju.

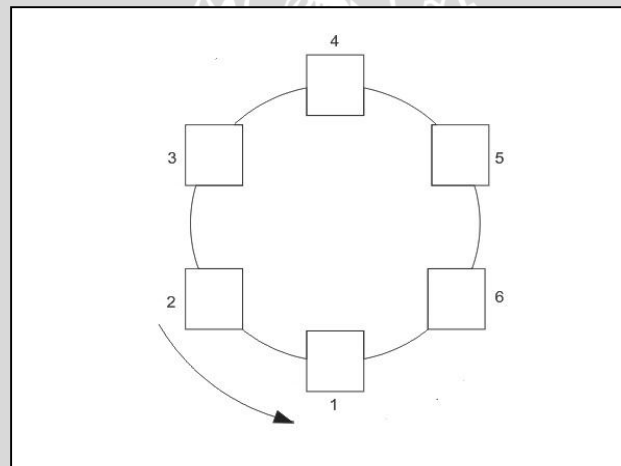
Ketika ingin mengarahkan garasi 1 ke bagian dasar maka *set point* sistem  $0^\circ$  atau dalam nilai ADC 102, jika ingin mengarahkan garasi 2 ke bagian dasar maka



repository.ub.ac.id

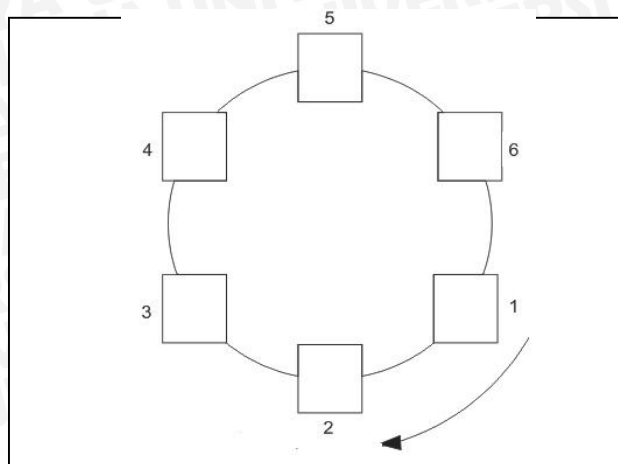
*set point* sistem  $60^\circ$  atau dalam nilai ADC 204, jika ingin mengarahkan garasi 3 ke bagian dasar maka *set point* sistem  $120^\circ$  atau dalam nilai ADC 306, jika ingin mengarahkan garasi 4 ke bagian dasar maka *set point* sistem  $180^\circ$  atau dalam nilai ADC 408, jika ingin mengarahkan garasi 5 ke bagian dasar maka *set point* sistem  $240^\circ$  atau dalam nilai ADC 510, jika ingin mengarahkan garasi 6 ke bagian dasar maka *set point* sistem  $300^\circ$  atau dalam nilai ADC 612.

Ketika nilai *present value* lebih kecil daripada *set point* maka motor DC akan bergerak *counter clock wise* atau berlawanan dengan arah jarum jam, sebaliknya ketika nilai *present value* lebih besar daripada *set point* maka motor DC bergerak *clock wise*, dan ketika nilai *present value* sama dengan nilai *set point* maka motor DC tidak bergerak. Pergerakan model rotary parking akan ditunjukkan dalam Gambar 4.16, Gambar 4.17, dan Gambar 4.18.



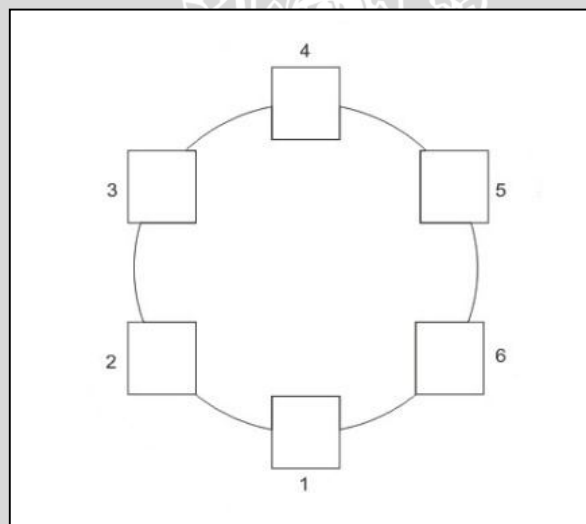
**Gambar 4.16 Pergerakan Ketika *Push button* dan *Limit switch* yang Dipilih Benomor 2 Saat Garasi 1 Sebagai Acuan**

Arah panah pada Gambar 4.16 menunjukkan bahwa ketika garasi yang di dasar adalah garasi 1 dan akan mengarahkan garasi 2 ke dasar maka *rotary parking* akan bergerak *counter clock wise*, karena nilai *present value* lebih kecil daripada nilai *set point*.



**Gambar 4.17 Pergerakan Ketika *Push button* dan *Limit switch* yang Dipilih Benomor 1 Saat Garasi 2 Sebagai Acuan**

Arah panah pada Gambar 4.17 menunjukkan bahwa ketika garasi yang di dasar adalah garasi 2 dan akan mengarahkan garasi 1 ke dasar maka *rotary parking* akan bergerak *clock wise*, karena nilai *present value* lebih besar daripada nilai *set point*.

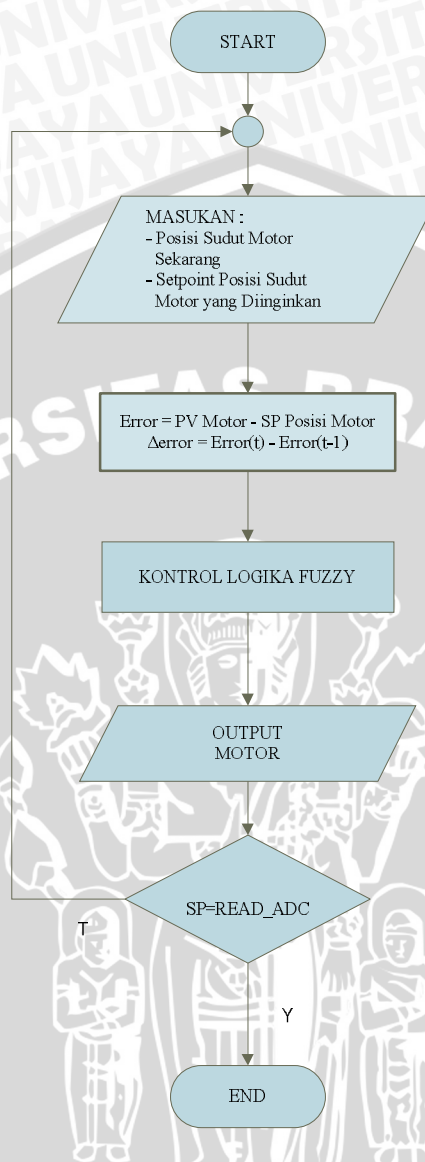


**Gambar 4.18 Pergerakan Ketika *Push button* dan *Limit switch* yang Dipilih Benomor 1 Saat Garasi 1 Sebagai Acuan**

Tidak adanya arah panah pada Gambar 4.18 menunjukkan bahwa ketika garasi yang di dasar adalah garasi 1 dan *push button* atau *limit switch* yang dipilih bernomor 1 maka *rotary parking* akan tidak bergerak, karena nilai *present value* lebih besar sama dengan nilai *set point*. Pergerakan *rotary parking* untuk keadaan yang lain dapat dilihat pada lembar lampiran.

#### 4.5 Perancangan Perangkat Lunak (*Software*)

Flowchart perancangan perangkat lunak ditunjukkan pada Gambar 4.19



Gambar 4.19 Flowchart Perangkat Lunak Sistem

## BAB V

### PENGUJIAN DAN ANALISIS SISTEM

Tujuan pengujian sistem ini adalah untuk menentukan apakah alat yang telah dibuat berfungsi dengan baik dan sesuai dengan perancangan. Pengujian pada sistem ini meliputi pengujian setiap blok maupun pengujian secara keseluruhan. Pengujian setiap blok ini dilakukan untuk menemukan letak kesalahan dan mempermudah analisis pada sistem apabila alat tidak bekerja sesuai dengan perancangan.

#### 5.1 Pengujian Potensiometer

##### a. Tujuan

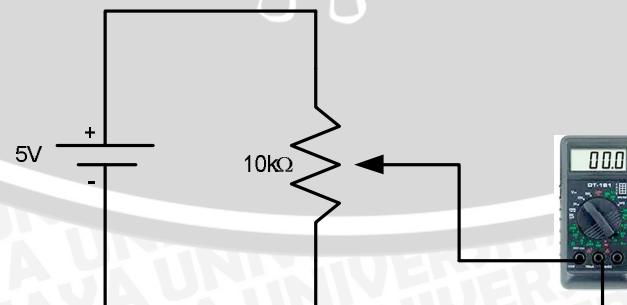
Mengetahui tingkat kelinieran dari potensiometer dalam membaca perubahan sudut putar motor DC.

##### b. Peralatan yang digunakan

- Potensiometer.
- Multimeter.
- Penggaris busur.
- Catu daya 5 volt.

##### c. Langkah pengujian

1. Merangkai peralatan seperti pada Gambar 5.1
2. Potensiometer diputar dari  $0^\circ$  sampai  $300^\circ$
3. Mencatat dan menggambar data tegangan keluaran tiap  $15^\circ$  dari potensiometer untuk melihat kelinierannya.



Gambar 5.1. Pengujian Potensiometer

d. Hasil Pengujian

Secara teori, tegangan keluaran dari potensiometer ditulis dalam persamaan

$$V_{output} = \frac{R_{potensiometer}}{R_{total}} \times V_{input} \dots\dots\dots(5.1)$$

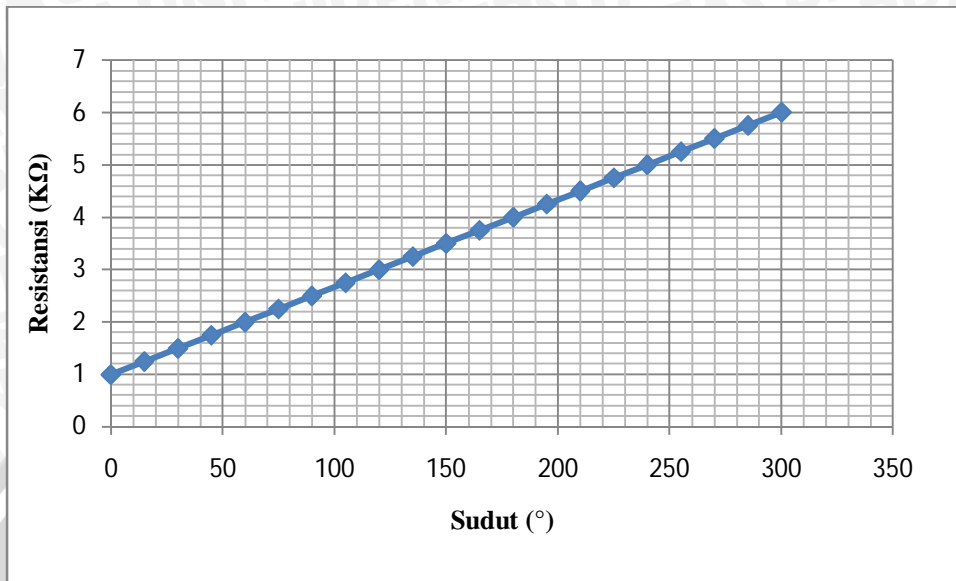
data hasil pengujian potensiometer ditunjukkan dalam Tabel 5.1 berikut

**Tabel 5.1 Hasil Pengujian Potensiometer**

No	Sudut (°)	Resistansi (K Ω)	Tegangan Perhitungan (V)	Tegangan Pengukuran (V)	Error (%)
1	0	1	0,498	0,50000	0,39900
2	15	1,25	0,626	0,62499	0,16100
3	30	1,5	0,754	0,74999	0,57874
4	45	1,75	0,883	0,87499	0,91529
5	60	2	0,999	0,99999	0,13230
6	75	2,25	1,145	1,12499	1,77880
7	90	2,5	1,267	1,24999	1,38765
8	105	2,75	1,391	1,37499	1,16465
9	120	3	1,502	1,49999	0,13433
10	135	3,25	1,651	1,62498	1,60102
11	150	3,5	1,782	1,74998	1,81056
12	165	3,75	1,967	1,87498	4,90772
13	180	4	2,002	1,99998	0,10100
14	195	4,25	2,189	2,12498	3,01279
15	210	4,5	2,304	2,24998	2,38622
16	225	4,75	2,436	2,37498	2,56945
17	240	5	2,524	2,49998	0,94769
18	255	5,25	2,712	2,62497	3,31532
19	270	5,5	2,837	2,74997	3,15256
20	285	5,5	2,837	2,87497	3,34016
21	300	5,75	2,971	2,99997	0,11010

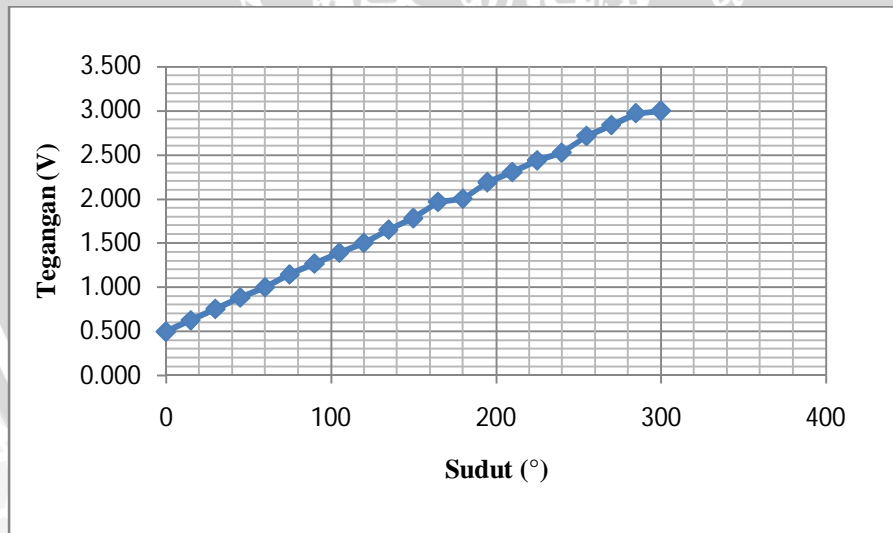


Grafik hubungan antara sudut dengan resistansi digambarkan pada Gambar 5.2



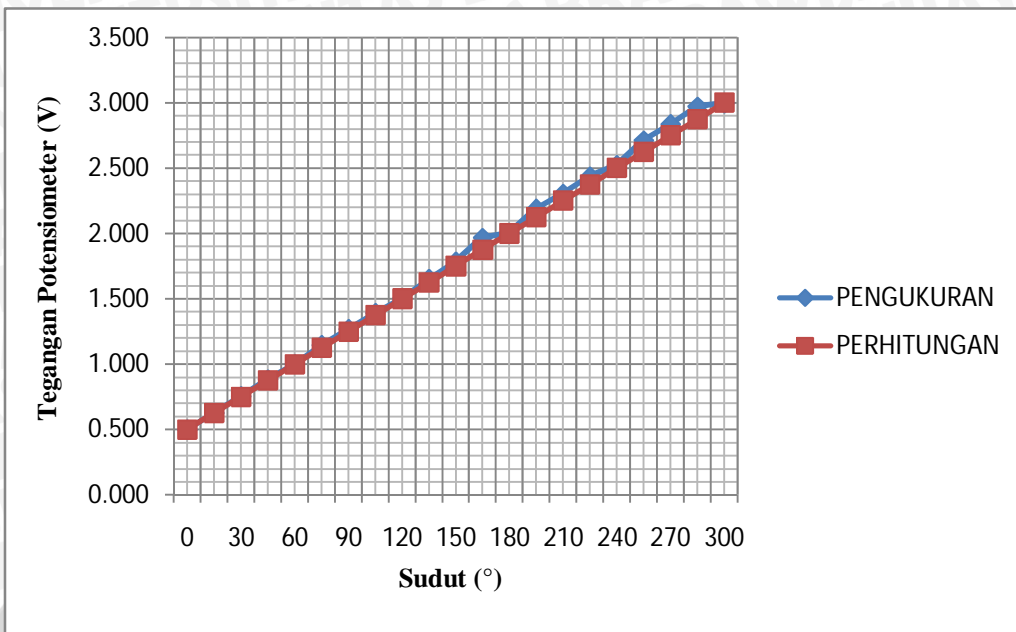
Gambar 5.2 Grafik Hubungan Antara Sudut dengan Resistansi

Hubungan antara sudut dengan tegangan potensiometer digambarkan pada Gambar 5.3



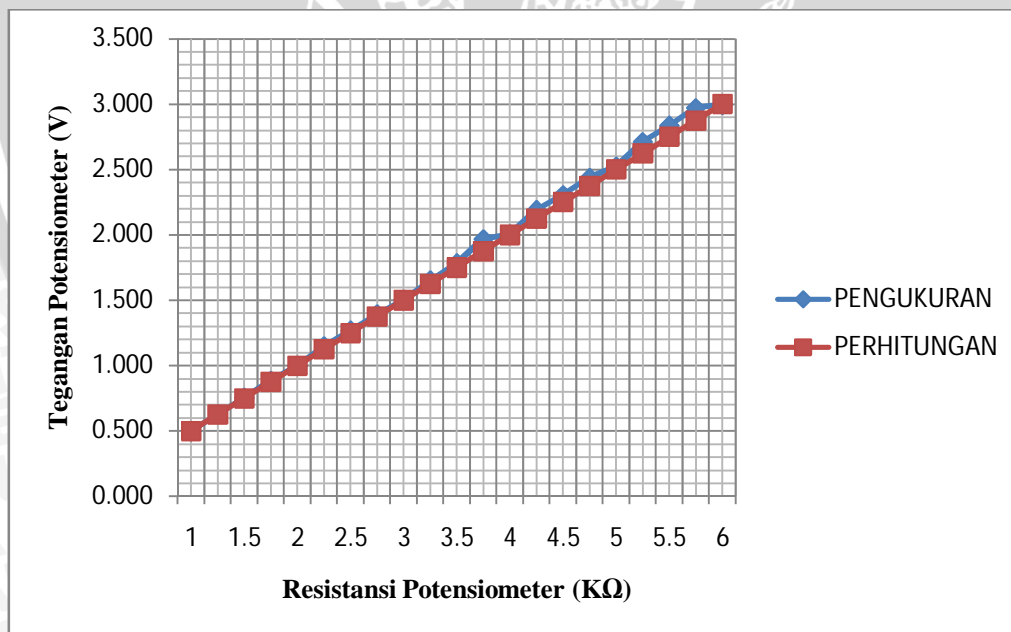
Gambar 5.3 Grafik Hubungan Antara Sudut dengan Tegangan

Grafik perbandingan antara tegangan perhitungan dan tegangan pengukuran pada potensiometer digambarkan pada Gambar 5.4



**Gambar 5.4 Grafik Perbandingan Antara Tegangan Perhitungan dengan Tegangan Pengukuran Terhadap Sudut**

Grafik perbandingan antara tegangan perhitungan dan tegangan pengukuran terhadap resistansi potensiometer digambarkan pada Gambar 5.5



**Gambar 5.5 Grafik Perbandingan Antara Tegangan Perhitungan dengan Tegangan Pengukuran Terhadap Resistansi**



Dari hasil pengujian yang dilakukan pada potensiometer dapat dilihat bahwa potensiometer yang digunakan mempunyai kelinieran yang baik karena hasil pengukuran perubahan tegangan terhadap perubahan resistansi dapat dibuat persamaan sebagai berikut:

$$V = 0,49999 R + 2 \dots \dots \dots (5.2)$$

Dimana,

V = Tegangan Keluaran Potensiometer

R = Resistansi Potensiometer

Sehingga ideal untuk digunakan sebagai pendeteksi posisi sudut putar motor DC dalam menggerakkan model *rotary parking* sampai pada posisi sudut putar yang sesuai dengan *set point*.

## 5.2 Pengujian Driver Motor DC

### a. Tujuan

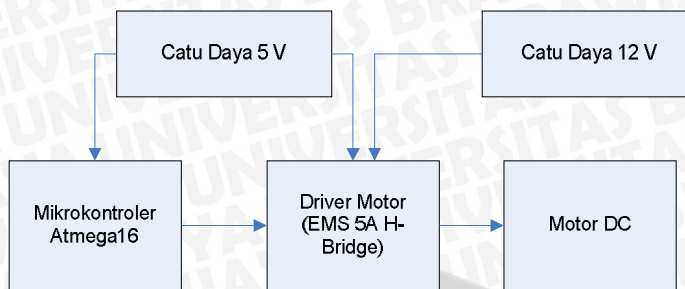
Mengetahui *output driver* motor apabila diberi *input* yang berbeda-beda.

### b. Peralatan yang digunakan

- Minimum sistem mikrokontroler ATmega8
- Driver motor
- Motor DC
- Catu daya 5V dan 12V

### c. Langkah pengujian

1. Merangkai peralatan seperti dalam Gambar 5.6
2. Mengisi mikrokontroler dengan program yang mengeluarkan data keluaran biner dengan keluaran 01 dan 10
3. Mengaktifkan catu daya 5 volt dan 12 volt
4. Mencatat pergerakan motor DC



Gambar 5.6 Diagram Blok Pengujian *Driver Motor DC*

d. Hasil pengujian

Tabel 5.2 Hasil Pengujian *Driver Motor DC*

No	Logika Keluaran		Arah Putaran Motor DC
	PORT B.1	PORT B.2	
1	1	0	Searah Jarum Jam
2	0	1	Berlawanan Arah Jarum jam

Berdasarkan hasil pengujian pada Tabel 5.2 dapat diketahui bahwa motor DC akan berputar searah jarum jam ketika diberi logika 1-0, dan apabila diberi logika 0-1 maka motor akan berputar berlawanan arah jam. Dengan demikian dapat disimpulkan bahwa rangkaian *driver* motor DC ini dapat berjalan dengan baik pada sistem yang direncanakan.

5.3 Pengujian Kecepatan

a. Tujuan

Untuk mengetahui letak posisi akhir garasi model *rotary parking* yang diinginkan yang telah digerakkan oleh motor DC.

b. Peralatan yang digunakan

- Modul mikrokontroler Atmega16
- *Tachometer*.

c. Langkah pengujian

Mengukur kecepatan motor dengan nilai *Pulse Width Modulation* (PWM) yang berbeda, sehingga didapatkan kecepatan yang diinginkan.



## d. Hasil pengujian

Hasil pengujian kecepatan motor dengan nilai *Pulse Width Modulation* (PWM) yang ditentukan ditunjukkan pada Tabel 5.3

**Tabel 5.3. Hasil Pengujian *Pulse Width Modulation* (PWM)**

PWM	V MK (Volt)	V Motor (Volt)	RPM
10	0.215	0.28	0.000
20	0.410	0.61	0.000
30	0.410	0.61	1.260
40	0.795	1.29	1.710
50	0.984	1.63	2.160
60	1.176	2.08	2.610
70	1.351	2.76	3.070
80	1.542	2.87	3.520
90	1.752	3.45	3.970
100	1.944	4.60	4.420
110	2.136	5.50	4.880
120	2.329	6.20	5.330
130	2.522	6.70	5.785
140	2.715	7.40	6.235
150	2.907	7.77	6.685
160	3.102	8.20	7.140
170	3.292	8.54	7.590
180	3.488	8.94	8.015
190	3.660	9.20	8.450
200	3.870	9.53	8.900
210	4.050	9.90	9.365
220	4.250	9.95	9.820
230	4.420	10.10	10.295
240	4.600	10.38	10.760
250	4.780	10.54	11.210
255	4.900	10.67	11.455

#### 5.4 Pengukuran Letak Posisi Akhir

##### a. Tujuan

Untuk mengetahui letak posisi akhir garasi model *rotary parking* yang diinginkan yang telah digerakkan oleh motor DC.

##### b. Peralatan yang digunakan

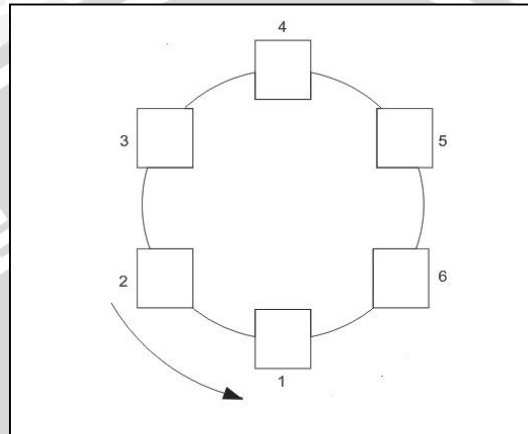
- Beban dengan berat 4 gram sebanyak 6 buah.
- Busur

c. Langkah pengujian

Mengukur posisi garasi 1,2,3,4,5,6 dari garasi acuan diukur menggunakan busur, sehingga didapatkan selisih tiap garasi memiliki beda sudut  $60^\circ$ .

d. Hasil pengujian

Hasil pengujian posisi garasi 2 dari titik acuan *rotary parking* dapat ditunjukkan dalam Gambar 5.7



**Gambar 5.7 Pergerakan Garasi 2 Menuju Garasi Acuan Atau Garasi Dasar**

Dari pengujian didapat bahwa ketika memposisikan garasi 2 menuju garasi acuan atau garasi dasar, maka motor DC bergerak sejauh  $60^\circ$ . Untuk pengukuran posisi garasi yang lain dapat dilihat pada lembar lampiran.

## 5.5 Pengujian Keseluruhan

### 5.5.1 Pengujian Tanpa Kontroler

a. Tujuan

Untuk mengetahui perbedaan respon sistem saat tidak ada kontroler dan saat ada kontroler

b. Peralatan yang digunakan

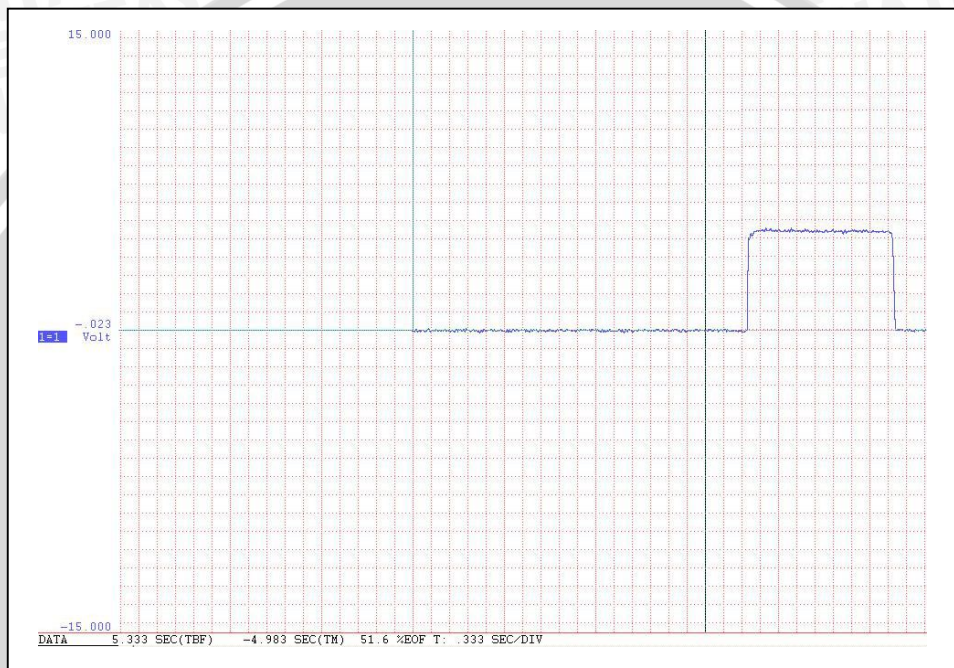
- Model *rotary parking* lengkap dengan potensiometer, motor DC, *limit switch*, serta *push button*.
- Catu daya 12 volt
- Sistem mikrokontroler ATmega 32
- *Driver* motor menggunakan modul EMS 5A H-Bridge

c. Langkah Pengujian

Pengujian alat ini dilakukan dengan cara menjalankan *rotary parking* dengan cara memasukkan nilai salah satu *set point* dengan tanpa kontroler yaitu 60° atau 204 ADC. Kemudian dengan *set point* yang sama *rotary parking* dijalankan dengan kontroler.

d. Hasil Pengujian

Grafik respon tegangan masukan motor tanpa kontroler dapat dilihat pada Gambar 5.8



**Gambar 5.8 Respon Tegangan Masukan Motor dengan *Set Point* 60° (ADC 204) Tanpa Kontroler**

$$T_s = 0,999 \text{ s} - 0,6993 \text{ s} = 0,2997 \text{ s}$$

$$T_r = 0,04995 \text{ s}$$

$$T_f = 0,04995 \text{ s}$$

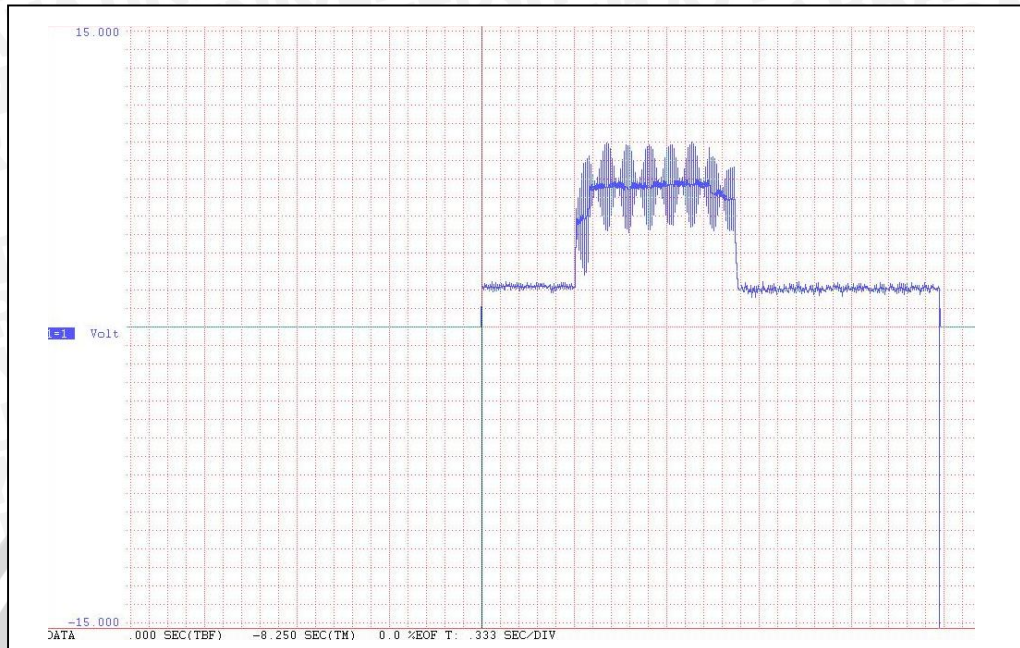
$$\tau = \frac{T_r + T_f}{2}$$

$$\tau = \frac{0,04995 + 0,04995}{2}$$

$$= 0,04995 \text{ s}$$

$$\frac{1}{\tau s + 1} = \frac{1}{0,04995s + 1}$$

Grafik respon sistem dengan kontroler dapat dilihat pada Gambar 5.9



**Gambar 5.9 Respon Tegangan Masukan Motor dengan *Set Point* 60° (ADC 204) Dengan Kontroler**

### 5.5.2 Pengujian Satu *Set point*

#### a. Tujuan

Untuk mengetahui kerja dari perangkat keras dan perangkat lunak setelah diintegrasikan dalam sebuah sistem terpadu

#### b. Peralatan yang digunakan

- Model *rotary parking* lengkap dengan potensiometer, motor DC, *limit switch*, serta *push button*.
- Catu daya 12 volt
- Sistem mikrokontroler ATmega 32
- *Driver* motor menggunakan modul EMS 5A H-Bridge
- Kabel serial

#### c. Langkah Pengujian

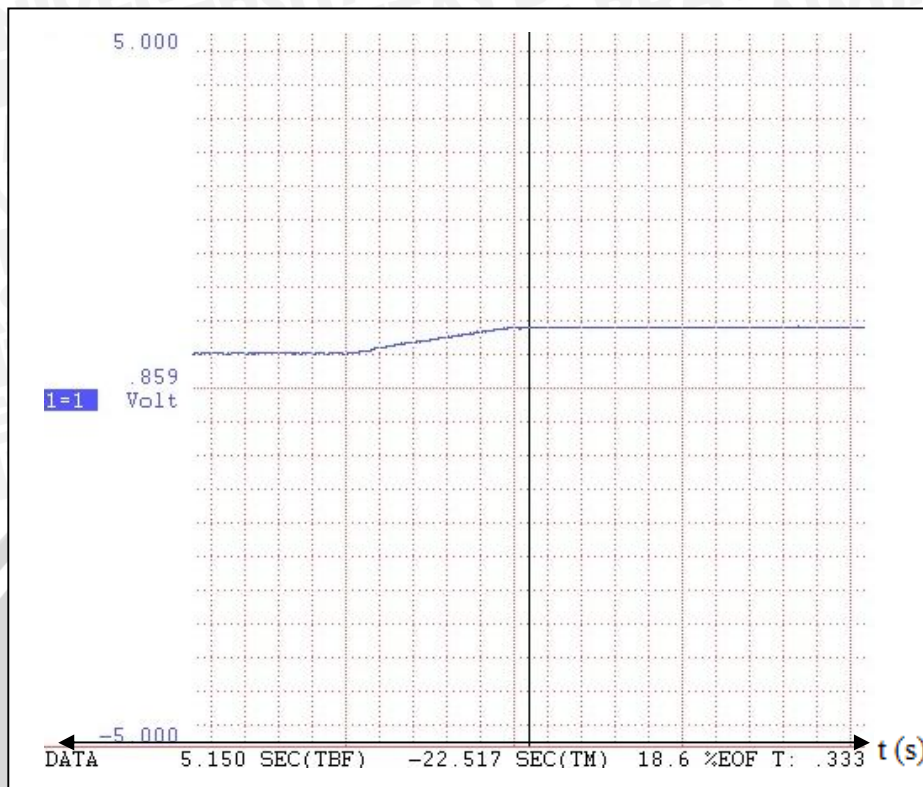
Pengujian alat ini dilakukan dengan cara menjalankan *rotary parking* dengan cara memasukkan nilai *set point* pada setiap garasi yang dituju. Nilai *set point* untuk masing-masing garasi berupa nilai ADC yang ditunjukkan dalam Tabel 5.4

Tabel 5.4 Data Sudut Motor DC

Garasi Dasar	Motor DC		
	ADC	Vin(V)	Sudut(°)
1	102	0.505	0
2	204	0.999	60
3	306	1.423	120
4	408	2.002	180
5	510	2.524	240
6	612	2.999	300

#### d. Hasil pengujian

Penyajian grafik respon sistem tiap sendi dibutuhkan data sudut dan waktu, sedangkan dalam proses pengujian data yang didapat berupa data ADC dan urutan data ADC. Untuk itu diperlukan melakukan pengkonversian data hasil ADC menjadi sudut dan pengkonversian urutan data ADC tersebut menjadi satuan waktu yang dapat dilihat pada Tabel 5.4. Karena penempatan dan data dari sensor potensiometer berbeda, maka pengkonversian dilakukan pada motor DC. Respon sistem dengan kontrol logika *fuzzy* ditunjukkan dalam Gambar 5.10, Gambar 5.11, Gambar 5.12, Gambar 5.13, dan 5.14. Dalam gambar tersebut digunakan  $time/div = 0,333 s$ , maksudnya adalah dalam arah horizontal satu kotak mewakili 0,333 *second*.



Gambar 5.10 Respon Sistem dengan Set Point 60° (ADC 204)

Setpoint 60° (ADC 204) :

- *Time Settling* atau  $T_s$  adalah waktu yang diperlukan kurva respon untuk mencapai dan menetap dalam daerah sekitar harga akhir yang nilainya ditentukan dengan presentase mutlak dari harga akhirnya (biasanya 5 % atau 2 %).  $t_s$  yang didapatkan dari pengujian ini adalah :

$T_s = \text{waktu awal di posisi tujuan} - \text{waktu awal posisi acuan}$

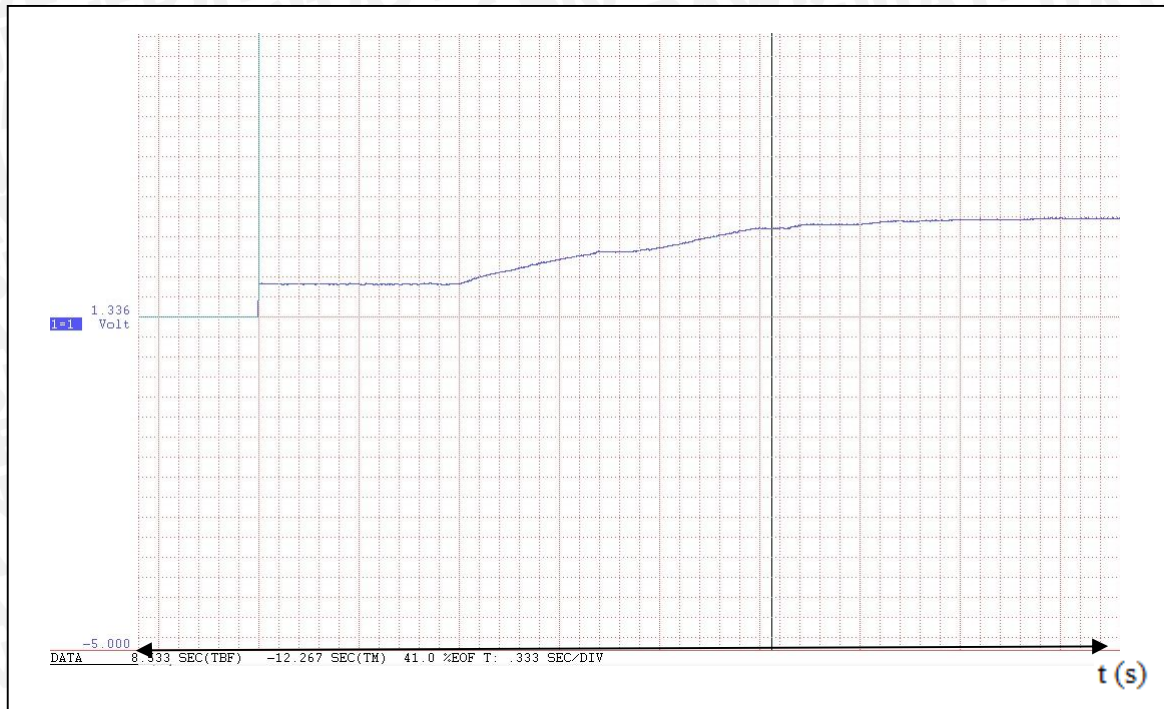
$$T_s = 2,997 - 1,332$$

$$T_s = 1,665 \text{ s}$$

- *Error steady state*,  $E_{ss}$  adalah selisih antara nilai keluaran dengan nilai masukan pada saat kondisi *steady state*.  $E_{ss}$  yang didapatkan dari pengujian ini adalah:

$$\cdot E_{ss} = \frac{60,5^\circ - 60^\circ}{60^\circ} \times 100 \% = 0,833 \%$$

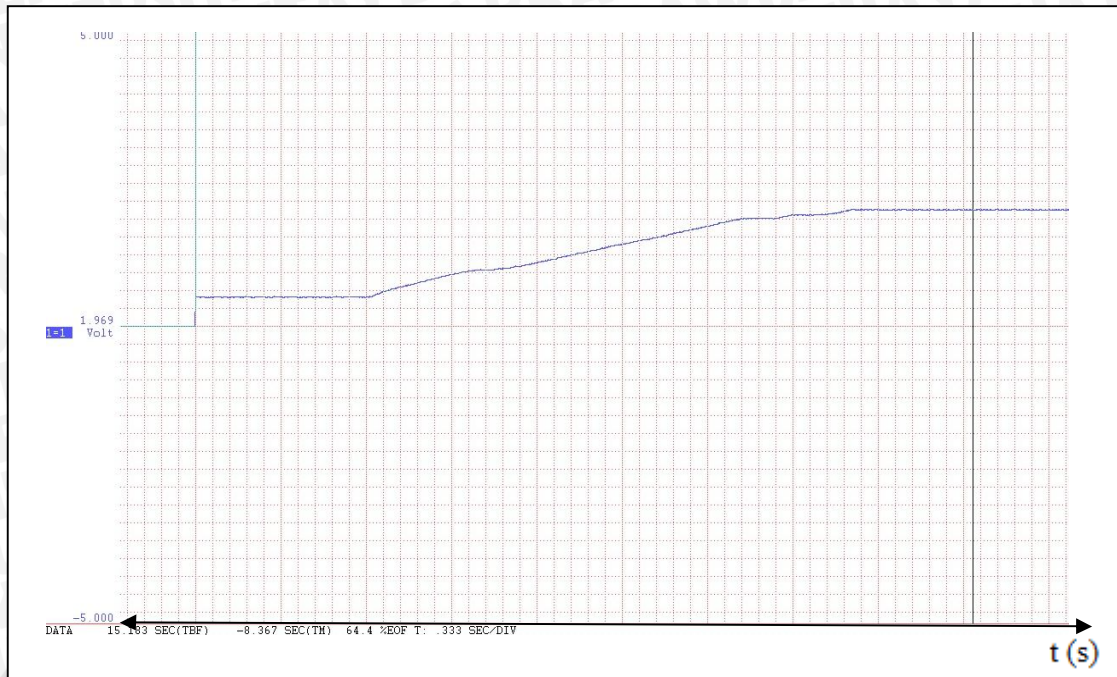




**Gambar 5.11 Respon Sistem dengan Set Point 120° (ADC 306)**

Set Point 120° (ADC 306) :

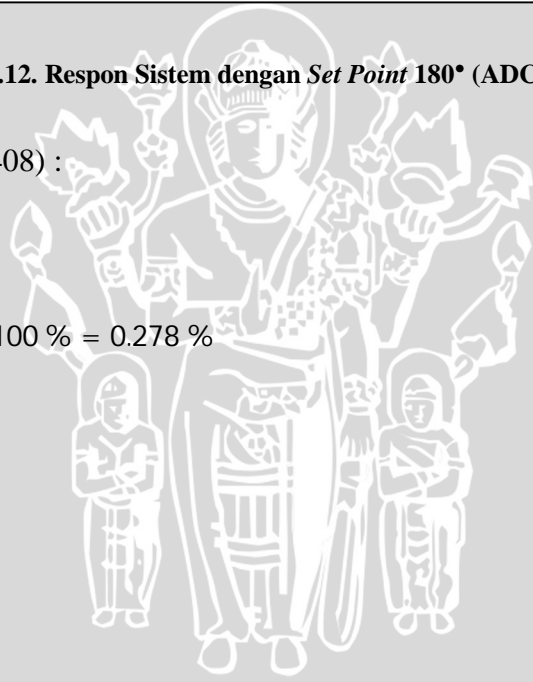
- $T_s = 12,654 - 3,33$
- $T_s = 9,324 \text{ s}$
- $Ess = \frac{120,5^\circ - 120^\circ}{120^\circ} \times 100 \% = 0.417\%$

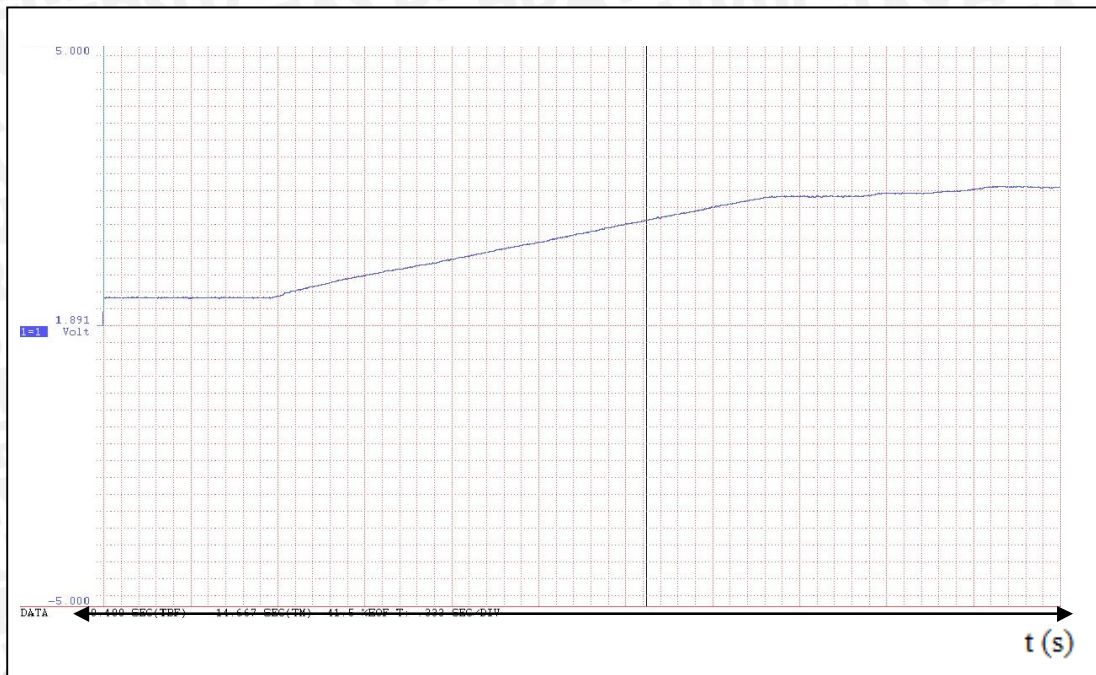


Gambar 5.12. Respon Sistem dengan Set Point 180° (ADC 408)

Set Point 180° (ADC 408) :

- $T_s = 15,318 - 3,33$
- $T_s = 11,988 \text{ s}$
- $Ess = \frac{180.5^\circ - 180^\circ}{180^\circ} \times 100 \% = 0.278 \%$

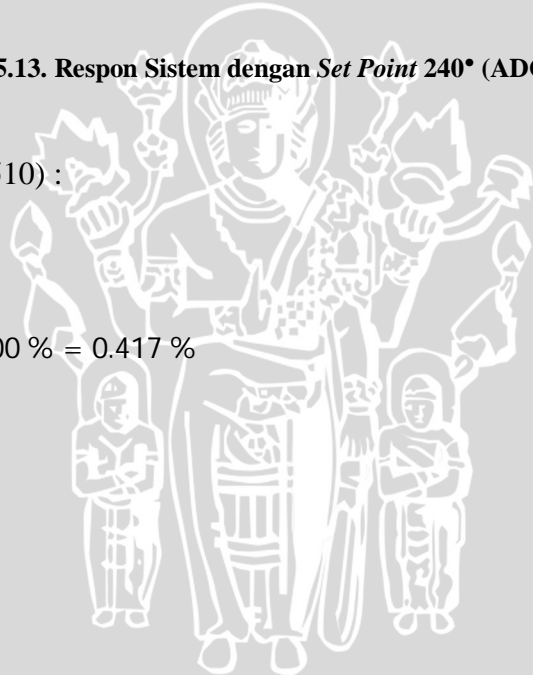


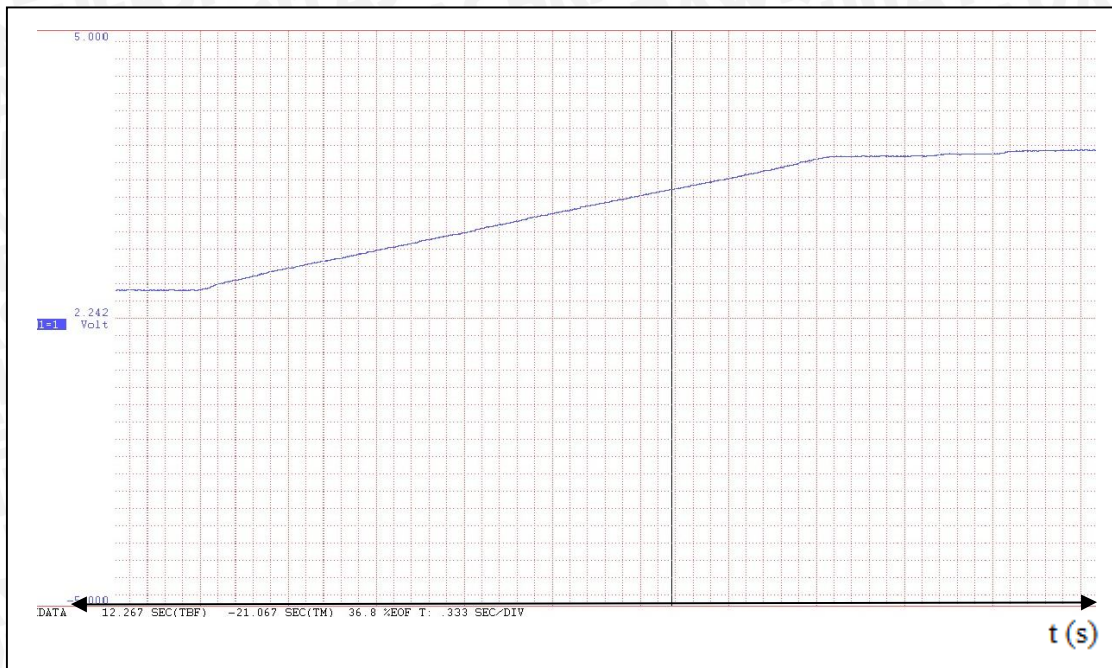


**Gambar 5.13. Respon Sistem dengan Set Point 240° (ADC 510)**

Set Point 240° (ADC 510) :

- $T_s = 16,983 - 2,997$
- $T_s = 13,986 \text{ s}$
- $Ess = \frac{241^\circ - 240^\circ}{240^\circ} \times 100 \% = 0.417 \%$





**Gambar 5.14. Respon Sistem dengan Set Point 300° (ADC 612)**

*Set Point 300° (ADC 612) :*

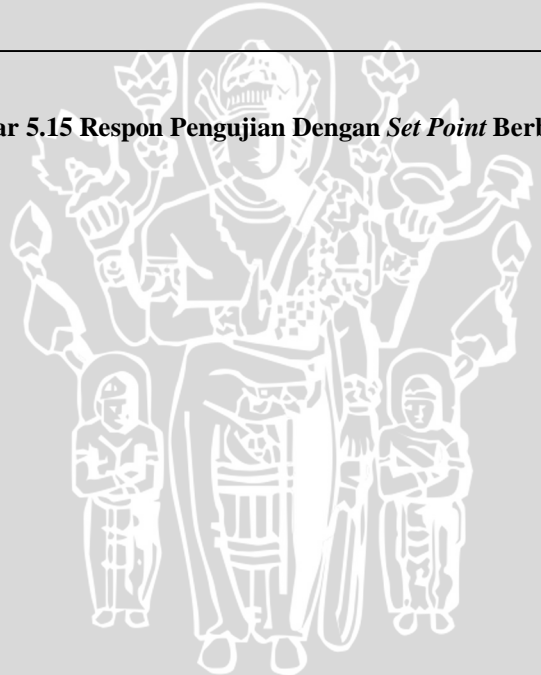
- $T_s = 16,983-1,665$
- $T_s = 15,318 \text{ s}$
- $Ess = \frac{301^\circ - 300^\circ}{300^\circ} \times 100 \% = 0.333\%$

### 5.5.3 Pengujian Dengan Set Point Berbeda

Pada pengujian ini, model *rotary parking* diberi *set point* berupa sudut yang didapat dari *push button* dan *limit switch*, yaitu *set point* pertama motor di posisi 120° (dengan data ADC 306) dan *set point* kedua di posisi 300° (dengan data ADC 612), *set point* ketiga di posisi 240° (dengan data ADC 510), *set point* keempat di posisi 60° (dengan data ADC 204), *set point* kelima di posisi 180° (dengan data ADC 408), dan *set point* terakhir di posisi 0° (dengan data ADC 102), maka respon sistem dengan kontroler *fuzzy* ditunjukkan dalam Gambar 5.15



Gambar 5.15 Respon Pengujian Dengan Set Point Berbeda



## BAB VI

### KESIMPULAN DAN SARAN

#### 6.1 Kesimpulan

Kesimpulan yang dapat diambil dalam pembuatan penelitian ini adalah sebagai berikut :

- 3) Dalam Pembuatan model sistem *rotary parking* digunakan sensor potensiometer linier sebagai sensor untuk mendeteksi posisi yang mempunyai error pembacaan 0-4,9 % dan sistem elektriknya menggunakan mikrokontroler Atmega 16 dengan bahasa pemrograman CV-AVR.
- 4) Hasil pengujian terhadap aplikasi kontrol logika *fuzzy* menunjukkan bahwa respon sistem untuk tiap *setpoint* mempunyai *error steady state* sebesar 0-0,833 %. Sedangkan  $T_s$  sudut  $60^\circ=1,665$  s,  $120^\circ=9,324$  s,  $180^\circ=11,988$  s,  $240^\circ=13,986$  s,  $300^\circ=15,318$  s. Hasil ini menunjukkan bahwa kontrol logika *fuzzy* menghasilkan respon sesuai dengan yang direncanakan dan dapat diaplikasikan pada model *rotary parking* untuk mengatur posisi sudut putar motor DC.

#### 6.2 Saran

Dalam Perancangan dan pembuatan alat ini masih terdapat kelemahan. Untuk memperbaiki kinerja alat dan pengembangan lebih lanjut disarankan :

- 1) Pergerakan *rotary parking* disarankan tidak ada batasan sudut putar motor DC untuk memaksimalkan gerak dari model *rotary parking*.
- 2) Dilakukan analisis tentang torsi beban sehingga kinerja *rotary parking* lebih baik.

## DAFTAR PUSTAKA

- ATMEL. 2007. *ATMEGA16/ATMEGA16L, 8-bit AVR Microcontroller with 16 Kbytes in System Programmable Flash*.
- Coughanowr. 1991. *Process System Analysis And Control Second Edition*. Mc Graw-Hill International Edition.
- Innovative Electronics. 2007. *Data Sheet EMS 5A H-Bridge*.
- Kilian, C. 1996. *Modern Control Technology : Component And System*. West Publishing Company.
- Kuswadi, S. 2000. *Kendali Cerdas*. Yogyakarta : ANDI.
- Lee, C. 1990. *Fuzzy Logic in Control System : Fuzzy Logic Controller Part I*. IEEE
- Prima A. 2011. *Implementasi Kontrol Logika Fuzzy untuk Pergerakan Robot Lengan ROB2R pada Pemindahan Objek dengan Variasi Jarak*. Skripsi tidak dipublikasikan. Malang : Universitas Brawijaya.
- Sivanandam. 2006. *Introduction to Fuzzy Logic Using Matlab*. India: Springer.
- Soemarwanto. 1999. *Diktat Kuliah Dasar Konversi Energi Elektrik*. Malang : Universitas Brawijaya.
- Thiang, R. 2001. *Aplikasi Kendali Fuzzy Logic untuk Pengaturan Kecepatan Motor Universal*. Surabaya : Universitas Kristen Petra.
- Yan, J. 1994. *Using Fuzzy Logic*. London: Prentice Hall.

# LAMPIRAN



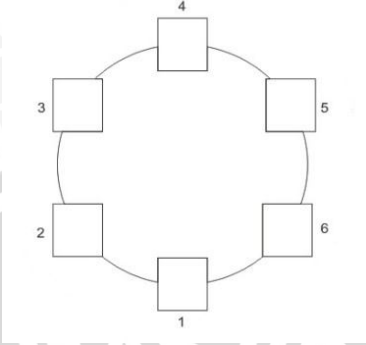
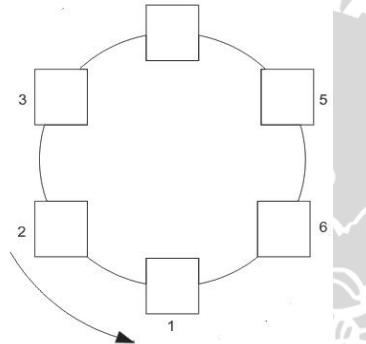
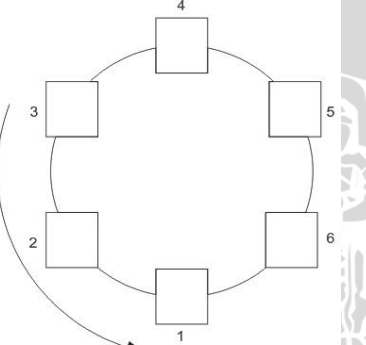
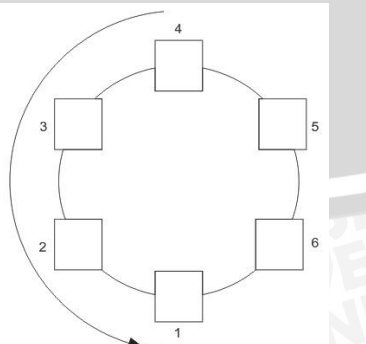


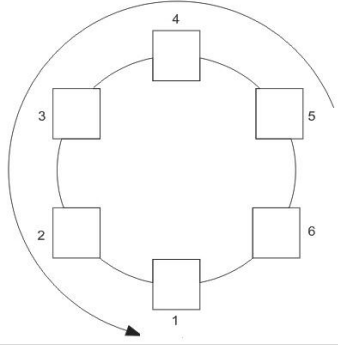
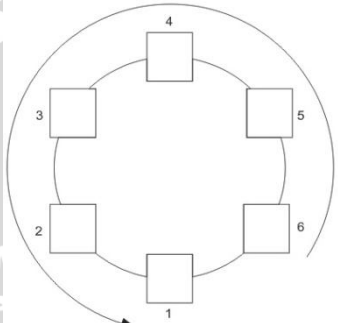
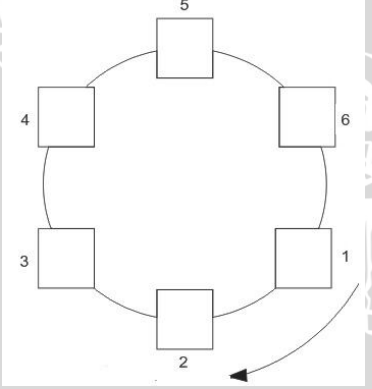
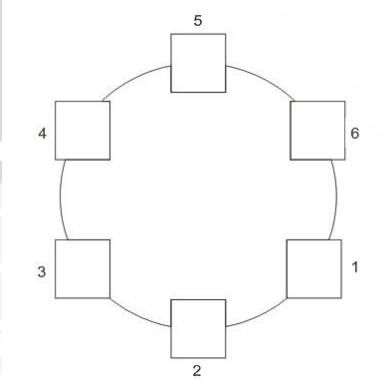
# LAMPIRAN I

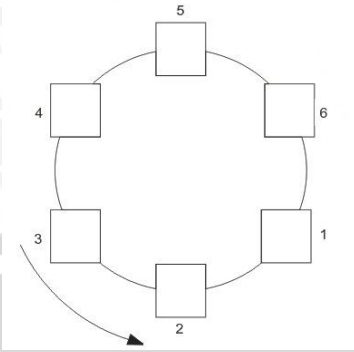
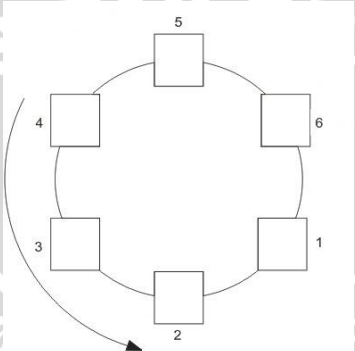
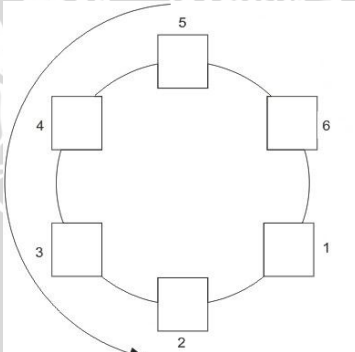
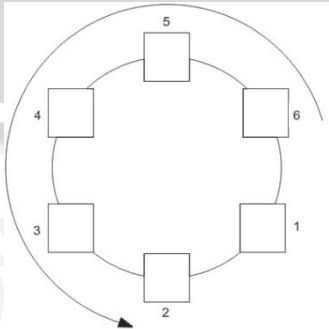
## PERANCANGAN PERGERAKAN

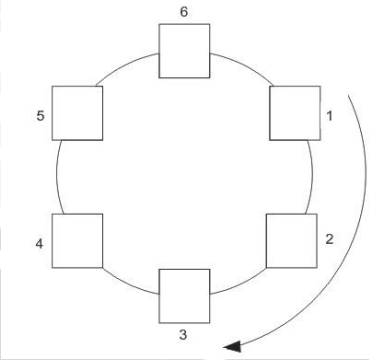
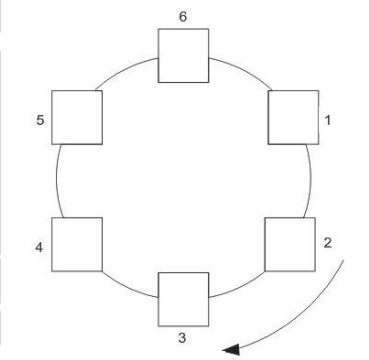
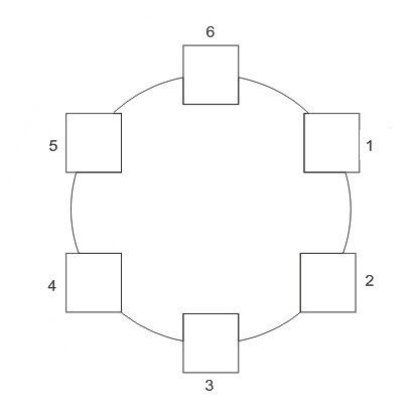
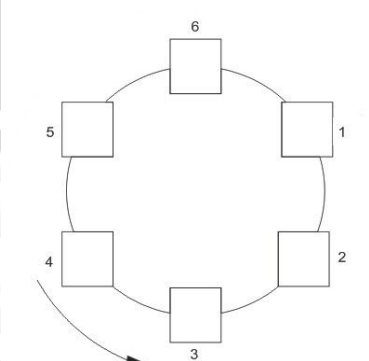


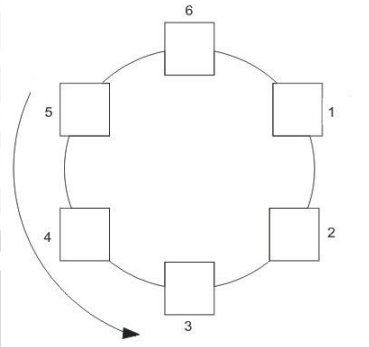
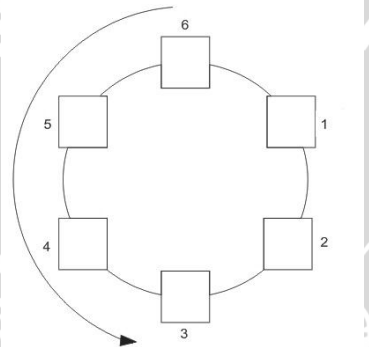
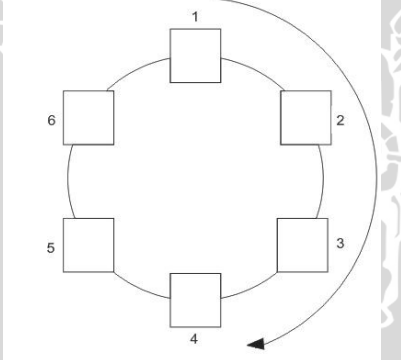
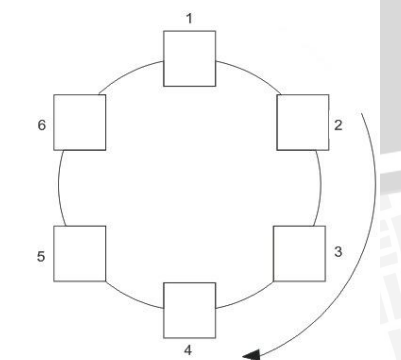
PERANCANGAN PERGERAKAN MODEL *ROTARY PARKING*

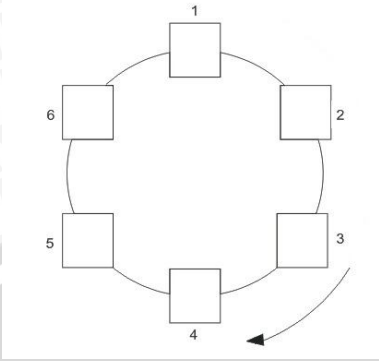
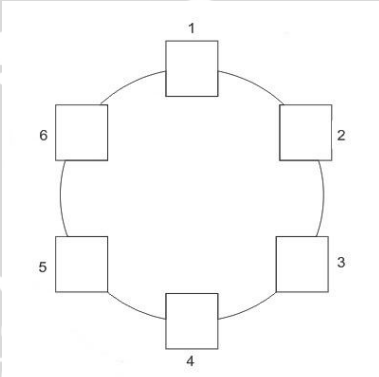
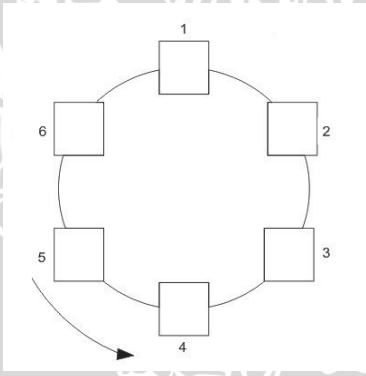
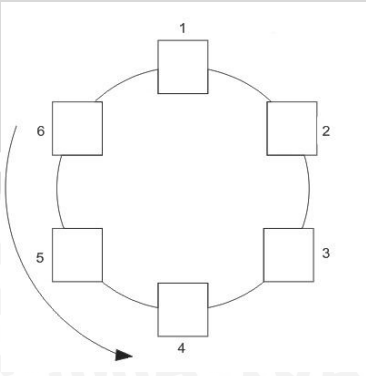
Garasi Acuan	Garasi Tujuan	Gambar Pergerakan	Keterangan Pergerakan
1	1		Stop
	2		Counter Clock Wise (CCW)
	3		Counter Clock Wise (CCW)
	4		Counter Clock Wise (CCW)

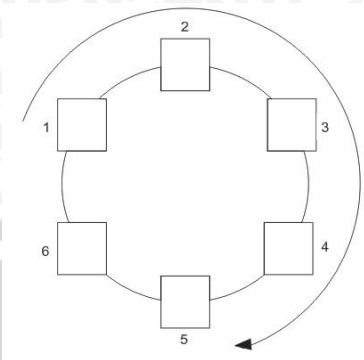
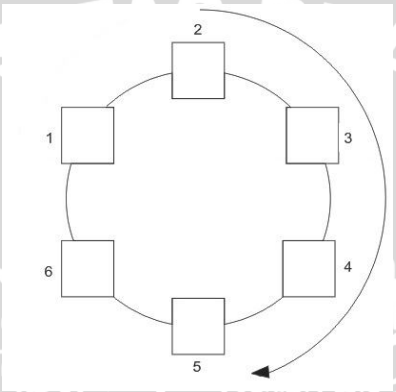
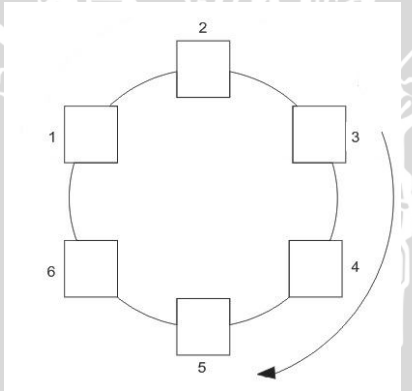
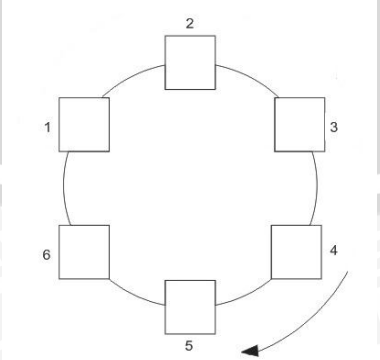
Garasi Acuan	Garasi Tujuan	Gambar Pergerakan	Keterangan Pergerakan
1	5		Counter Clock Wise (CCW)
2	6		Counter Clock Wise (CCW)
2	1	 	Clock Wise (CW)  Stop

Garasi Acuan	Garasi Tujuan	Gambar Pergerakan	Keterangan Pergerakan
2	3		Counter Clock Wise (CCW)
	4		Counter Clock Wise (CCW)
	5		Counter Clock Wise (CCW)
	6		Counter Clock Wise (CCW)

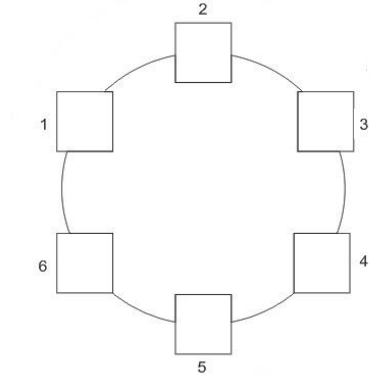
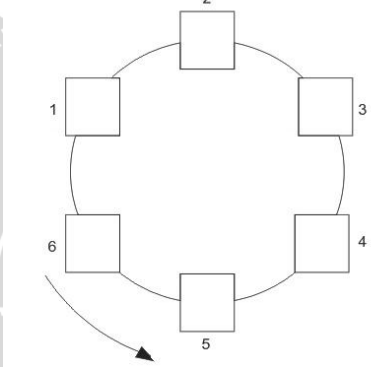
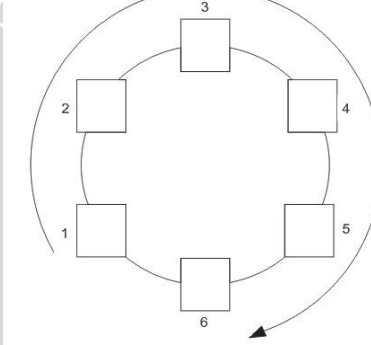
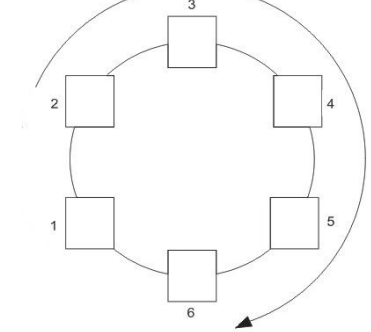
Garasi Acuan	Garasi Tujuan	Gambar Pergerakan	Keterangan Pergerakan
3	1		Clock Wise (CW)
	2		Clock Wise (CCW)
	3		Stop
	4		Counter Clock Wise (CCW)

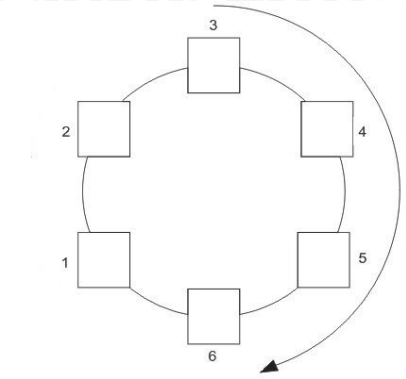
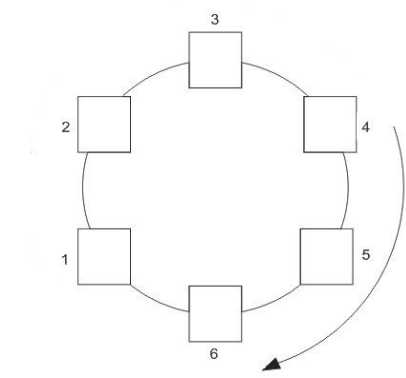
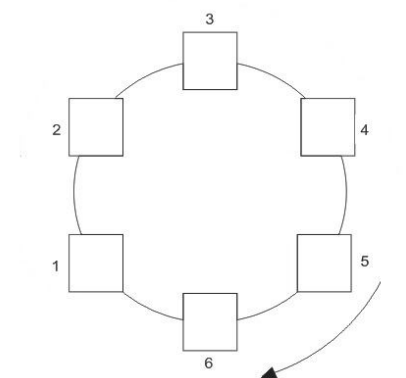
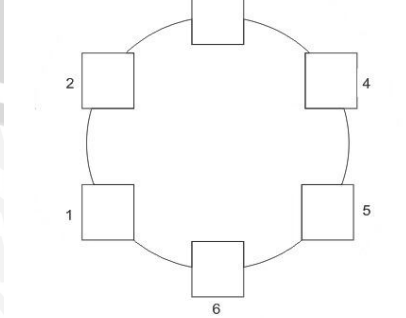
Garasi Acuan	Garasi Tujuan	Gambar Pergerakan	Keterangan Pergerakan
3	5		Counter Clock Wise (CCW)
	6		Counter Clock Wise (CCW)
4	1		Clock Wise (CW)
	2		Clock Wise (CW)

Garasi Acuan	Garasi Tujuan	Gambar Pergerakan	Keterangan Pergerakan
4	3		Clock Wise (CW)
	4		Stop
	5		Counter Clock Wise (CCW)
	6		Counter Clock Wise (CCW)

Garasi Acuan	Garasi Tujuan	Gambar Pergerakan	Keterangan Pergerakan
5	1		Clock Wise (CW)
	2		Clock Wise (CW)
	3		Clock Wise (CW)
	4		Clock Wise (CW)



Garasi Acuan	Garasi Tujuan	Gambar Pergerakan	Keterangan Pergerakan
5	5		Stop
	6		Counter Clock Wise (CCW)
6	1		Clock Wise (CW)
	2		Clock Wise (CW)

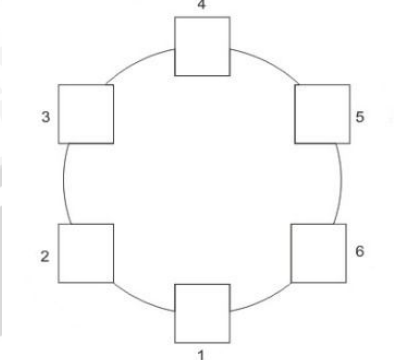
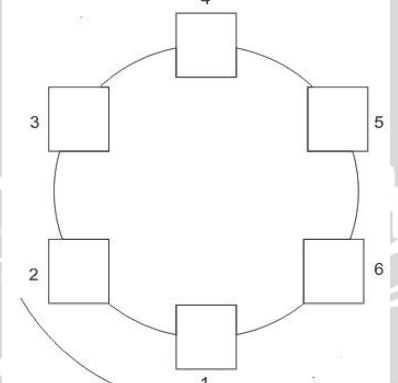
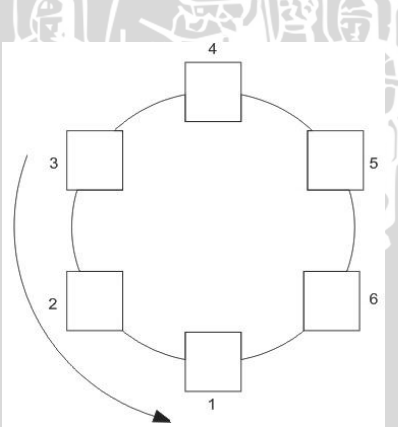
Garasi Acuan	Garasi Tujuan	Gambar Pergerakan	Keterangan Pergerakan
6	3		Clock Wise (CW)
	4		Clock Wise (CW)
	5		Clock Wise (CW)
	6		Stop

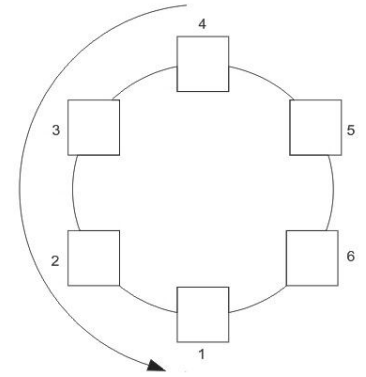
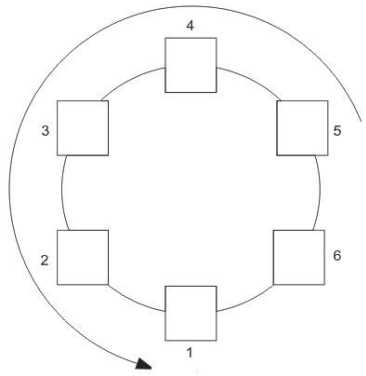
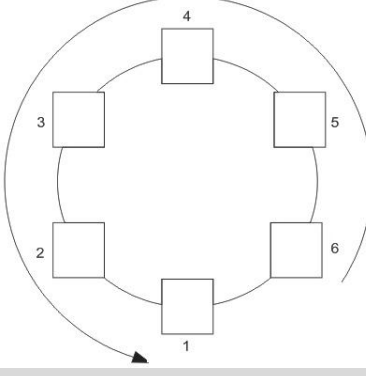
## LAMPIRAN II

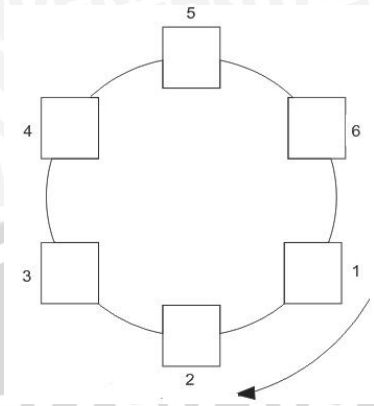
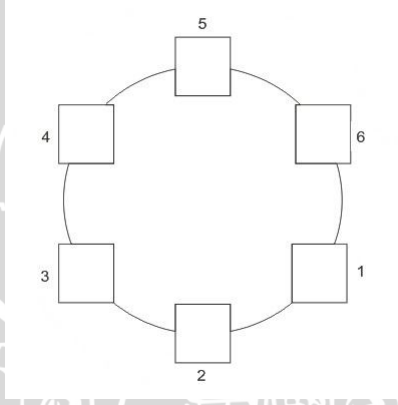
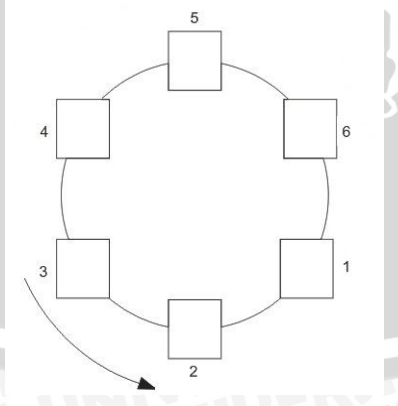
### PENGUJIAN POSISI AKHIR

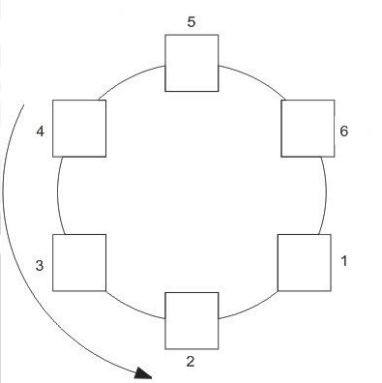
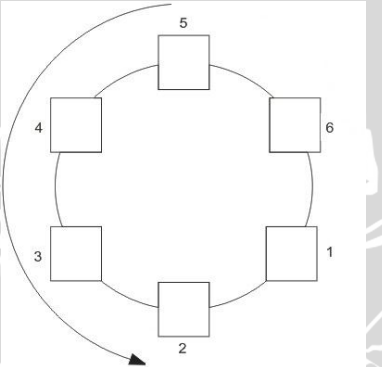
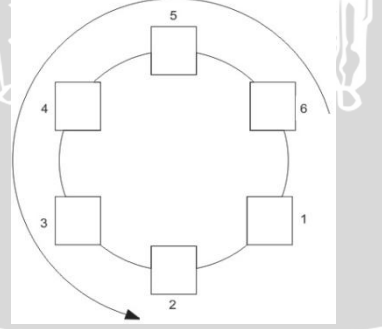


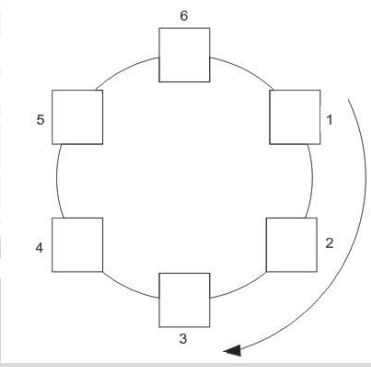
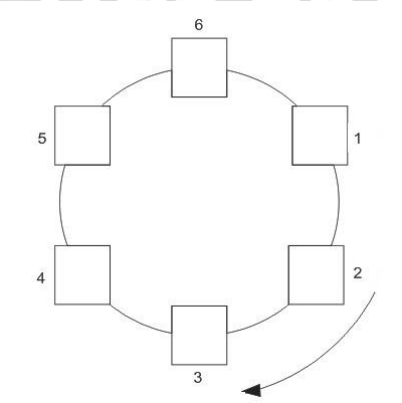
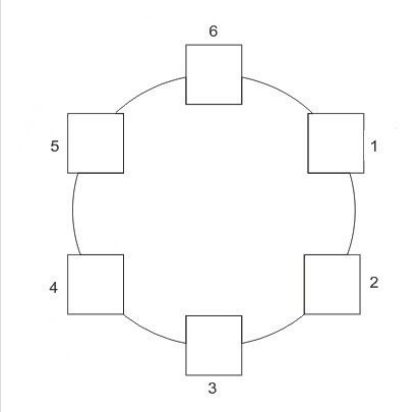
PENGUJIAN LETAK POSISI AKHIR

Garasi Acuan	Garasi Tujuan	Gambar Pergerakan	Sudut (°)
1	1		0
	2		60
	3		120

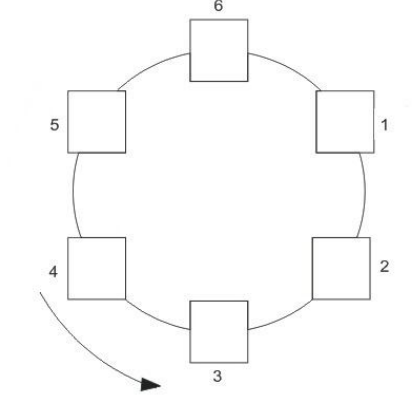
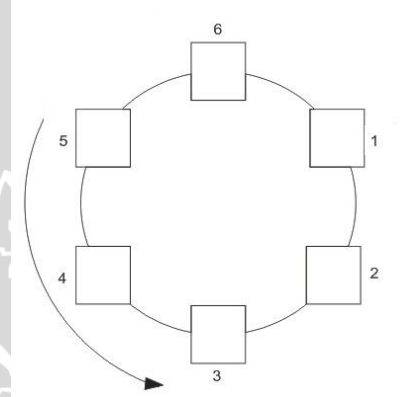
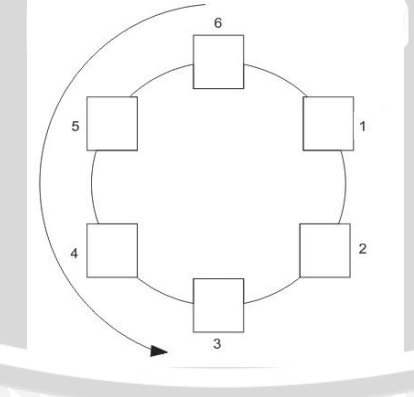
Garasi Acuan	Garasi Tujuan	Gambar Pergerakan	Sudut (°)
1	4		180
	5		240
	6		300

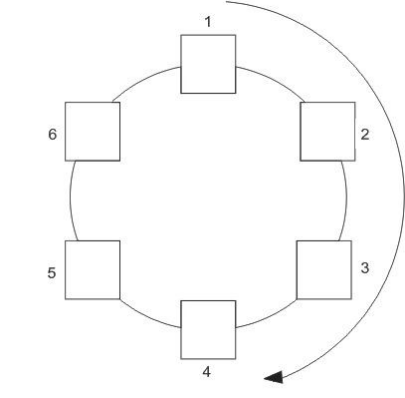
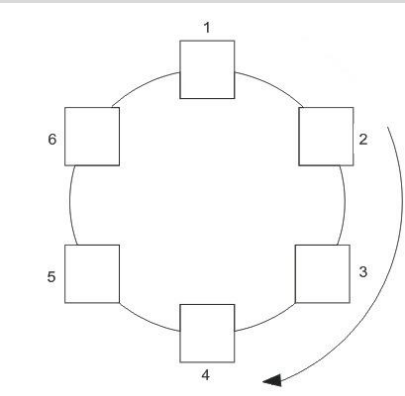
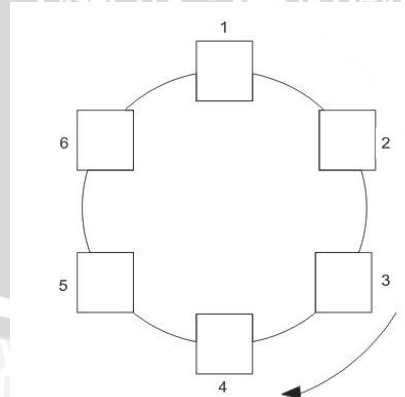
Garasi Acuan	Garasi Tujuan	Gambar Pergerakan	Sudut (°)
2	1		- 60
	2		0
	3		60

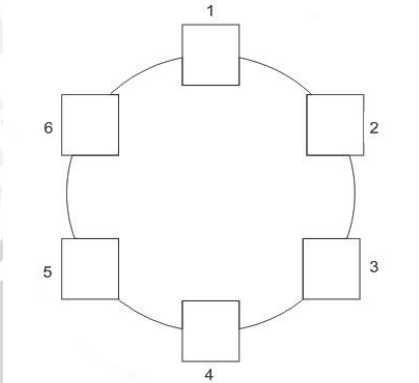
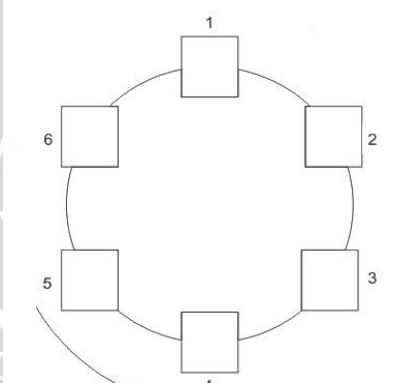
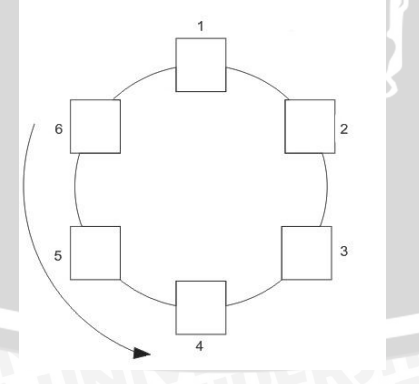
Garasi Acuan	Garasi Tujuan	Gambar Pergerakan	Sudut (°)
2	4		120
	5		180
	6		240

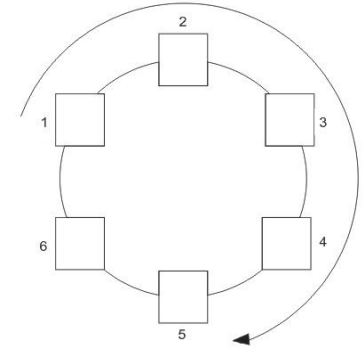
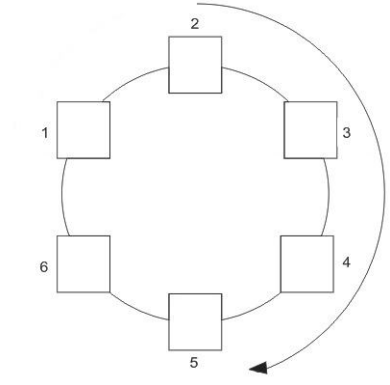
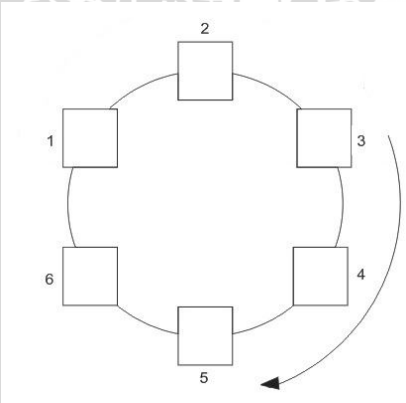
Garasi Acuan	Garasi Tujuan	Gambar Pergerakan	Sudut (°)
3	1		- 120
	2		- 60
	3		0

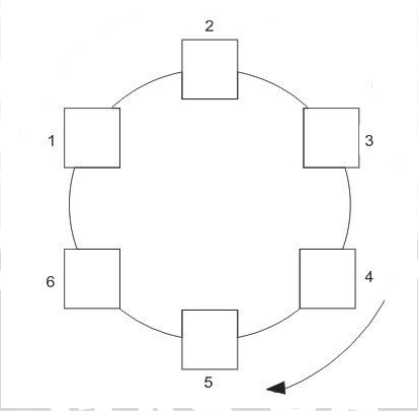
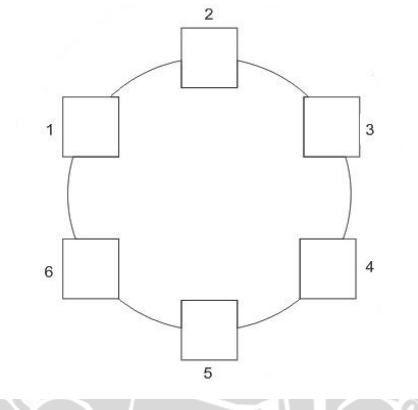
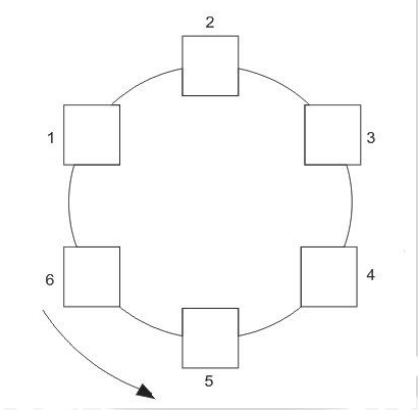


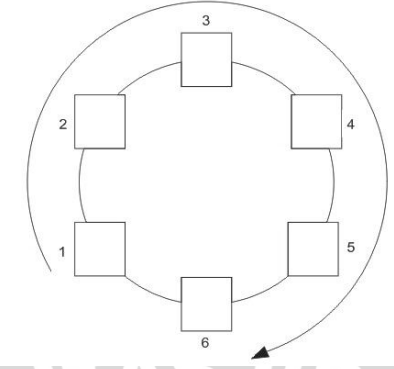
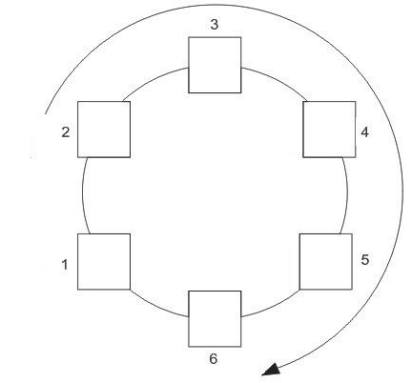
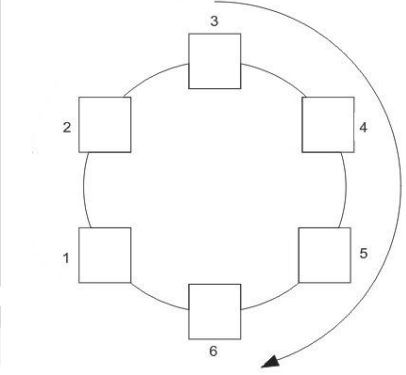
Garasi Acuan	Garasi Tujuan	Gambar Pergerakan	Sudut (°)
3	4		60
	5		120
	6		180

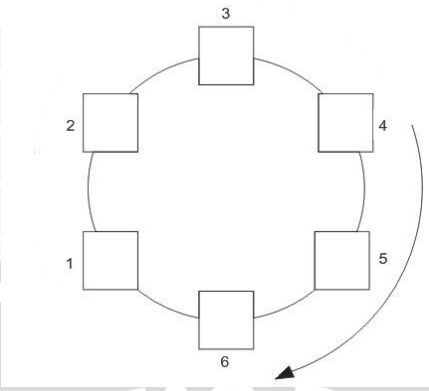
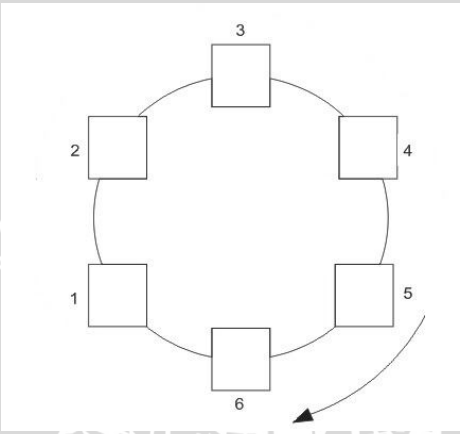
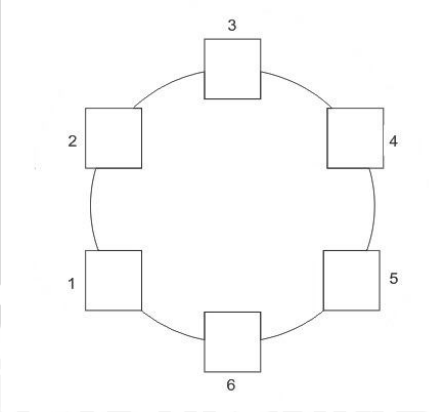
Garasi Acuan	Garasi Tujuan	Gambar Pergerakan	Sudut (°)
4	1		- 180
	2		- 120
	3		- 60

Garasi Acuan	Garasi Tujuan	Gambar Pergerakan	Sudut (°)
4	4		0
	5		60
	6		120

Garasi Acuan	Garasi Tujuan	Gambar Pergerakan	Sudut (°)
5	1		- 240
	2		- 180
	3		- 120

Garasi Acuan	Garasi Tujuan	Gambar Pergerakan	Sudut (°)
5	4		- 60
	5		0
	6		60

Garasi Acuan	Garasi Tujuan	Gambar Pergerakan	Sudut (°)
6	1		- 300
	2		- 240
	3		- 180

Garasi Acuan	Garasi Tujuan	Gambar Pergerakan	Sudut (°)
6	4		- 120
	5		- 60
	6		0

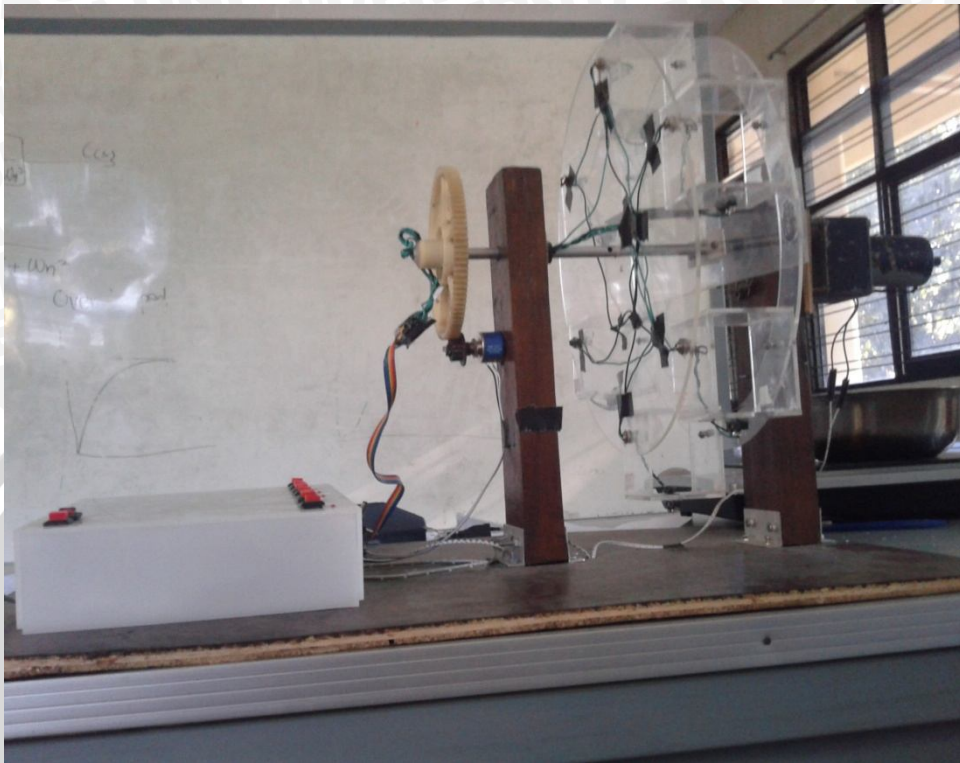
# LAMPIRAN III

## FOTO ALAT





## FOTO ALAT



Gambar 1 Model *Rotary Parking* Tampak Depan



Gambar 2 Model *Rotary Parking* Tampak Samping



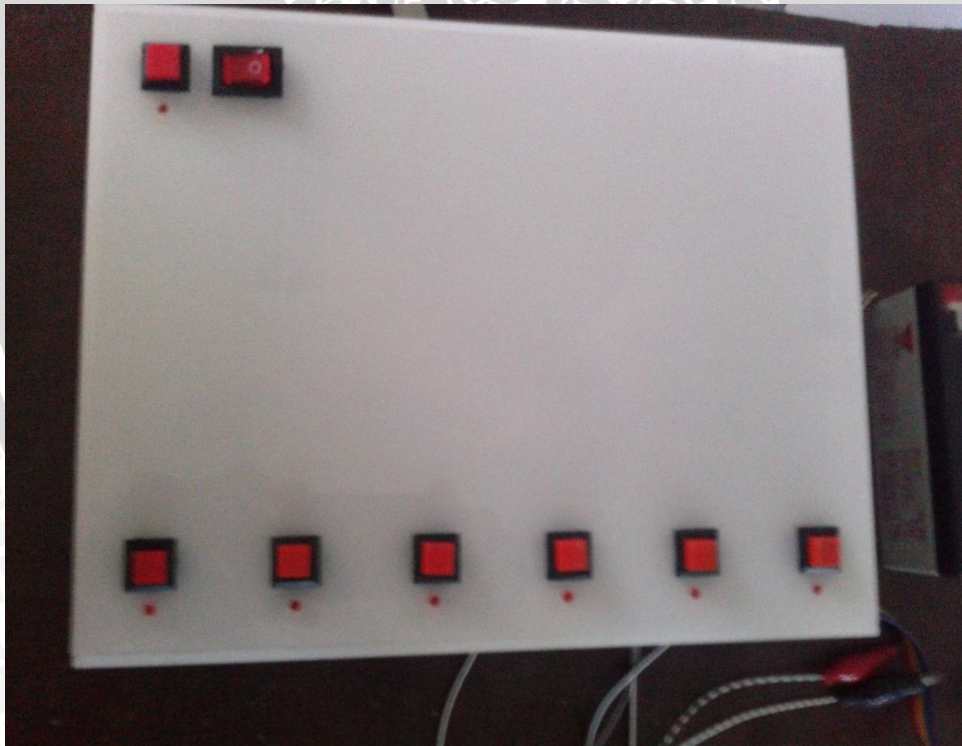
Gambar 3 Model *Rotary Parking* Tampak Atas



Gambar 4 Motor DC Pada Model *Rotary Parking*



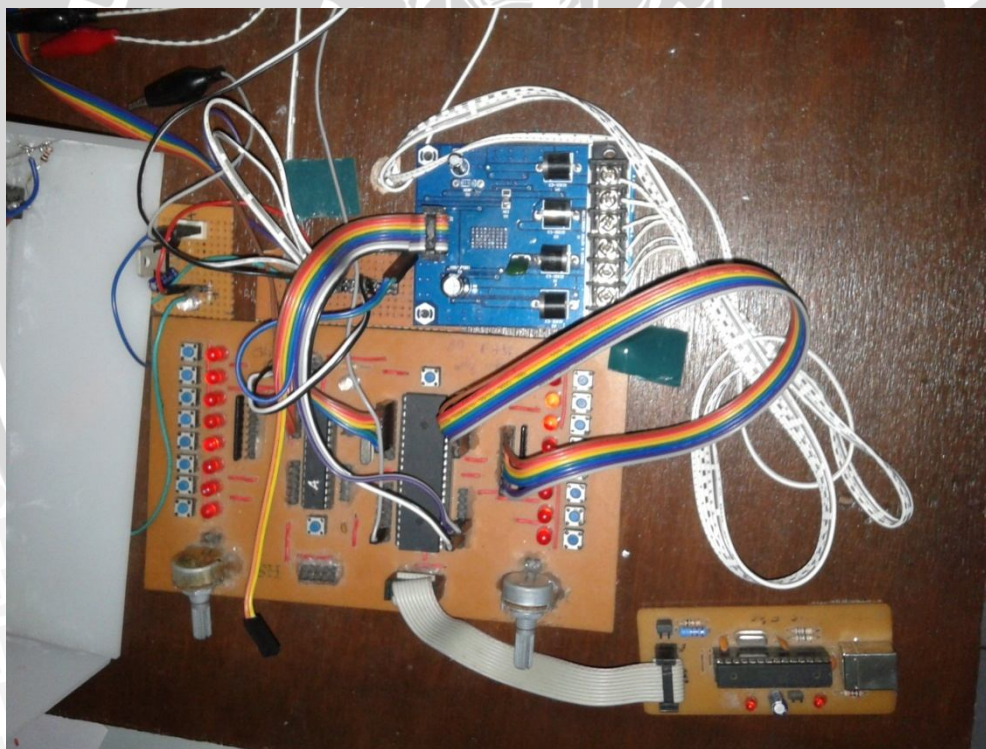
**Gambar 5 Potensiometer Linier Pada Model Rotary Parking**



**Gambar 6 Push Button Pada Model Rotary Parking**



Gambar 7 Limit Switch Pada Garasi Model Rotary Parking



Gambar 8 Sistem Elektrik Model Rotary Parking

# LAMPIRAN IV

## LISTING PROGRAM



/\*  
 \*\*\*\*\*/

This program was produced by the

CodeWizardAVR V1.24.8d Professional

Automatic Program Generator

© Copyright 1998-2006 Pavel Haiduc, HP InfoTech s.r.l.

<http://www.hpinfotech.com>

Project :

Version :

Date : 03/07/2012

Author : F4CG

Company : F4CG

Comments:

Chip type : Atmega16

Program type : Application

Clock frequency : 16,000000 MHz

Memory model : Small

External SRAM size : 0

Data Stack size : 512

\*\*\*\*\*/

```
#include <mega16.h>
```

```
#include <math.h>
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```



```
#include <delay.h>
```

```
#define ADC_VREF_TYPE 0x00
```

```
#define maju PORTB.1=0;PORTB.2=1;
```

```
#define mundur PORTB.1=1;PORTB.2=0;
```

```
#define stop PORTB.1=1;PORTB.2=1;OCR0=0;
```

```
#define max_speed 255
```

```
#define min_speed 60
```

```
//deklarasi PIN untuk LS
```

```
#define LS0 PINC.0
```

```
#define LS1 PINC.1
```

```
#define LS2 PINC.2
```

```
#define LS3 PINC.3
```

```
#define LS4 PINC.4
```

```
#define LS5 PINC.5
```

```
//deklarasi PIN untuk PB
```

```
#define PB0 PIND.0
```

```
#define PB1 PIND.1
```

```
#define PB2 PIND.2
```

```
#define PB3 PIND.3
```

```
#define PB4 PIND.4
```

```
#define PB5 PIND.5
```



```
#define GB PIND.6
```

```
void LS();
```

```
void CEKPB();
```

```
void MOTOR_MOVE();
```

```
void KONTROLER(int i);
```

```
void fuzzy(int ERROR,int DERROR);
```

```
void fuzzyfikasi(int ERR,int DERR);
```

```
void fuzzy_rule_follow();
```

```
void defuzzy();
```

```
float triangle(int value, float x0, float x1, float x2);
```

```
float trapezoid(int value, float x0, float x1, float x2, float x3);
```

```
float grade(int value, float x0, float x1);
```

```
float reverse_grade(int value, float x0, float x1);
```

```
int SP,er=0,ers=0,der=0;
```

```
struct side{
```

```
    float NB;
```

```
    float NS;
```

```
    float Z;
```

```
    float PS;
```

```
    float PB;
```

```
};
```

```
struct side e;
```



```
struct side de;
struct side def;
struct side K;

// Read the AD conversion result
unsigned int read_adc(unsigned char adc_input)
{
    ADMUX=adc_input|ADC_VREF_TYPE;

    // Start the AD conversion
    ADCSRA|=0x40;

    // Wait for the AD conversion to complete
    while ((ADCSRA & 0x10)==0);

    ADCSRA|=0x10;
    return ADCW;
}

// Declare your global variables here

void main(void)
{
    // Declare your local variables here

    // Input/Output Ports initialization
    // Port A initialization
```

```
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
```

```
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
```

```
PORTA=0x00;
```

```
DDRA=0x00;
```

```
// Port B initialization
```

```
// Func7=In Func6=In Func5=In Func4=In Func3=Out Func2=Out Func1=Out
```

```
Func0=In
```

```
// State7=T State6=T State5=T State4=T State3=0 State2=0 State1=0 State0=T
```

```
PORTB=0x00;
```

```
DDRB=0x0E;
```

```
// Port C initialization
```

```
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
```

```
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
```

```
PORTC=0x00;
```

```
DDRC=0x00;
```

```
// Port D initialization
```

```
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
```

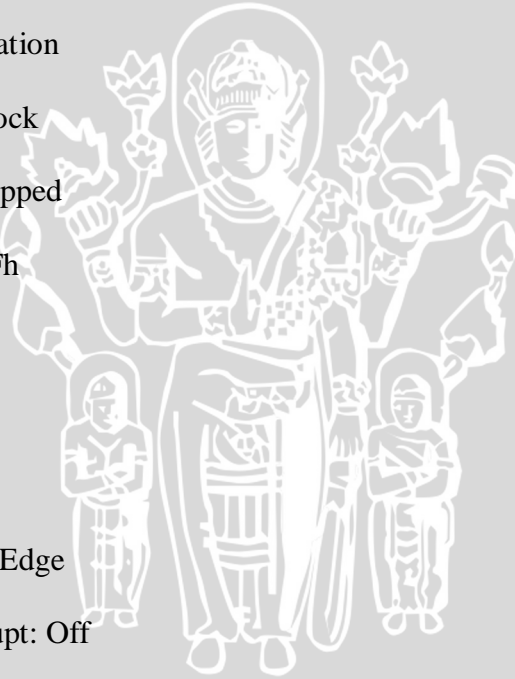
```
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
```

```
PORTD=0x00;
```

```
DDRD=0x00;
```

```
// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: 15,625 kHz
// Mode: Fast PWM top=FFh
// OC0 output: Non-Inverted PWM
TCCR0=0x6D;
TCNT0=0x00;
OCR0=0x00;
```

```
// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer 1 Stopped
// Mode: Normal top=FFFFh
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer 1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x00;
```



```
TCNT1H=0x00;
```

```
TCNT1L=0x00;
```

```
ICR1H=0x00;
```

```
ICR1L=0x00;
```

```
OCR1AH=0x00;
```

```
OCR1AL=0x00;
```

```
OCR1BH=0x00;
```

```
OCR1BL=0x00;
```

```
// Timer/Counter 2 initialization
```

```
// Clock source: System Clock
```

```
// Clock value: Timer 2 Stopped
```

```
// Mode: Normal top=FFh
```

```
// OC2 output: Disconnected
```

```
ASSR=0x00;
```

```
TCCR2=0x00;
```

```
TCNT2=0x00;
```

```
OCR2=0x00;
```

```
// External Interrupt(s) initialization
```

```
// INT0: Off
```

```
// INT1: Off
```

```
// INT2: Off
```

```
MCUCR=0x00;
```

```
MCUCSR=0x00;
```

```
// Timer(s)/Counter(s) Interrupt(s) initialization
```

```
TIMSK=0x00;
```

```
// USART initialization
```

```
// Communication Parameters: 8 Data, 1 Stop, No Parity
```

```
// USART Receiver: Off
```

```
// USART Transmitter: On
```

```
// USART Mode: Asynchronous
```

```
// USART Baud rate: 9600
```

```
UCSRA=0x00;
```

```
UCSRB=0x08;
```

```
UCSRC=0x86;
```

```
UBRRH=0x00;
```

```
UBRRL=0x67;
```

```
// Analog Comparator initialization
```

```
// Analog Comparator: Off
```

```
// Analog Comparator Input Capture by Timer/Counter 1: Off
```

```
ACSR=0x80;
```

```
SFIOR=0x00;
```

```
// ADC initialization
```



```
// ADC Clock frequency: 1000,000 kHz
```

```
// ADC Voltage Reference: AREF pin
```

```
ADMUX=ADC_VREF_TYPE;
```

```
ADCSRA=0x84;
```

```
while (1)
```

```
{
```

```
LS();
```

```
CEKPB();
```

```
}
```

```
//cek LS (room kosong)
```

```
void LS()
```

```
{
```

```
int flag=0;
```

```
while(flag==0)
```

```
{
```

```
if(LS0==0)
```

```
{
```

```
if(LS1==0)
```

```
{
```



```
if(LS2==0)
```

```
{
```

```
if(LS3==0)
```

```
{
```

```
if(LS4==0)
```

```
{
```

```
if(LS5==0)
```

```
{
```

```
stop;
```

```
break;
```

```
}
```

```
while(LS5==1)
```

```
{
```

```
flag++;
```

```
SP=612;
```

```
MOTOR_MOVE();
```

```
if(read_adc(1)==SP)
```

```
{
```

```
stop;
```

```
break;
```

```
}
```

```
}
```

```
else
```

```
while(LS4==1)
```

```
{
    flag++;
    SP=510;
    MOTOR_MOVE();

    If(read_adc(1)==SP)
    {
        stop;
        break;
    }

}

else
while(LS3==1)
{
    flag++;
    SP=408;
    MOTOR_MOVE();

    If(read_adc(1)==SP)
    {
        stop;
        break;
    }

}
```



```
else
while(LS2==1)
{
flag++;
SP=306;
MOTOR_MOVE();

If(read_adc(1)==SP)
{
stop;
break;
}
}
else
while(LS1==1)
{
flag++;
SP=204;
MOTOR_MOVE();

If(read_adc(1)==SP)
{
stop;
break;
}
}
```



```
}  
else  
while(LS0==1)  
{  
flag++;  
SP=102;  
MOTOR_MOVE();  
If(read_adc(1)==SP)  
{  
stop;  
break;  
}  
}  
while(flag==1)  
{  
MOTOR_MOVE();  
while  
(((SP==102)&&(LS0==0))||((SP==204)&&(LS1==0))||((SP==306)&&(LS2==0)  
)||((SP==408)&&(LS3==0))||((SP==510)&&(LS4==0))||((SP==612)&&(LS5==0)  
))  
{  
stop;  
flag=0;  
break;  
}
```

```
};  
}  
  
//cek PB  
void CEKPb()  
{  
    if(PB0==0)  
    {  
        PB0=0;  
        if (LS0==0)  
        {SP=102;}  
        MOTOR_MOVE ();  
        while(read_adc(1)==SP)  
        {  
            stop;  
            PB0=1;  
            break;  
        };  
    };  
};
```

```
if(PB1==0)  
{  
    PB1=0;
```

```
if(LS1==0)
    {SP=204;}
    MOTOR_MOVE ();
    while(read_adc(1)==SP)
    {
        stop;
        PB1=1;
        break;
    }
}
```

```
if(PB2==0)
    {
        PB2=0;
        if(LS2==0)
            {SP=306;}
            MOTOR_MOVE ();
            while(read_adc(1)==SP)
            {
                stop;
                PB2=1;
                break;
            }
    }
```



```
if(PB3==0)
{
PB3=0;
if(LS3==0)
{SP=408;}
MOTOR_MOVE ();
while(read_adc(1)==SP)
{
stop;
PB3=1;
break;
}
}

if(PB4==0)
{
PB4=0;
if(LS4==0)
{SP=510;}
MOTOR_MOVE ();
while(read_adc(1)==SP)
{
stop;
```



```
PB4=1;
```

```
break;
```

```
}
```

```
}
```

```
if(PB5==0)
```

```
{
```

```
PB5=0;
```

```
if(LS5==0)
```

```
{SP=612;}
```

```
MOTOR_MOVE ();
```

```
while(read_adc(1)==SP)
```

```
{
```

```
stop;
```

```
PB5=1;
```

```
break;
```

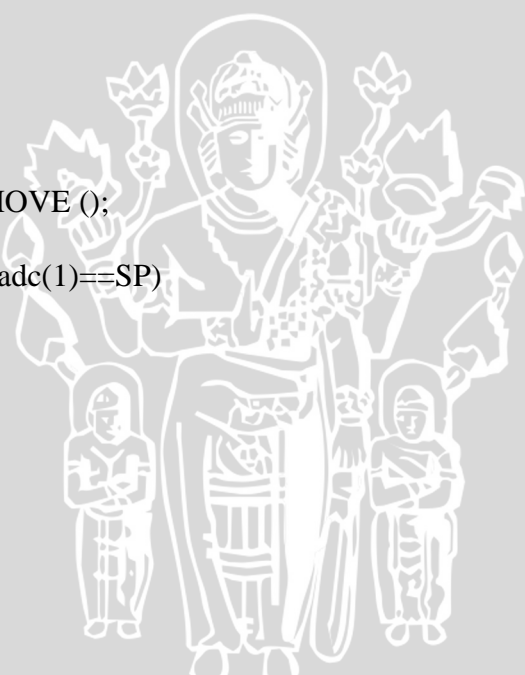
```
}
```

```
}
```

```
}
```

```
void MOTOR_MOVE()
```

```
{
```



```
int x;
x=(int) read_adc(1);
er=(SP-x);
der=er-ers;
//Fuzzy
fuzzy(er,der);
ers=er;
}
void KONTROLER(int i)
{
int x=0;
x=i;
if (x>255)
{
x=255;
};
if (x<-255)
{
x=-255;
};
if(x==0)
{
stop;
};
};
```



```

if(x>0)
{
mundur;

OCR0= (unsigned char)x;
};

if(x<0)
{
maju;

OCR0= (unsigned char) abs(x);
};
}

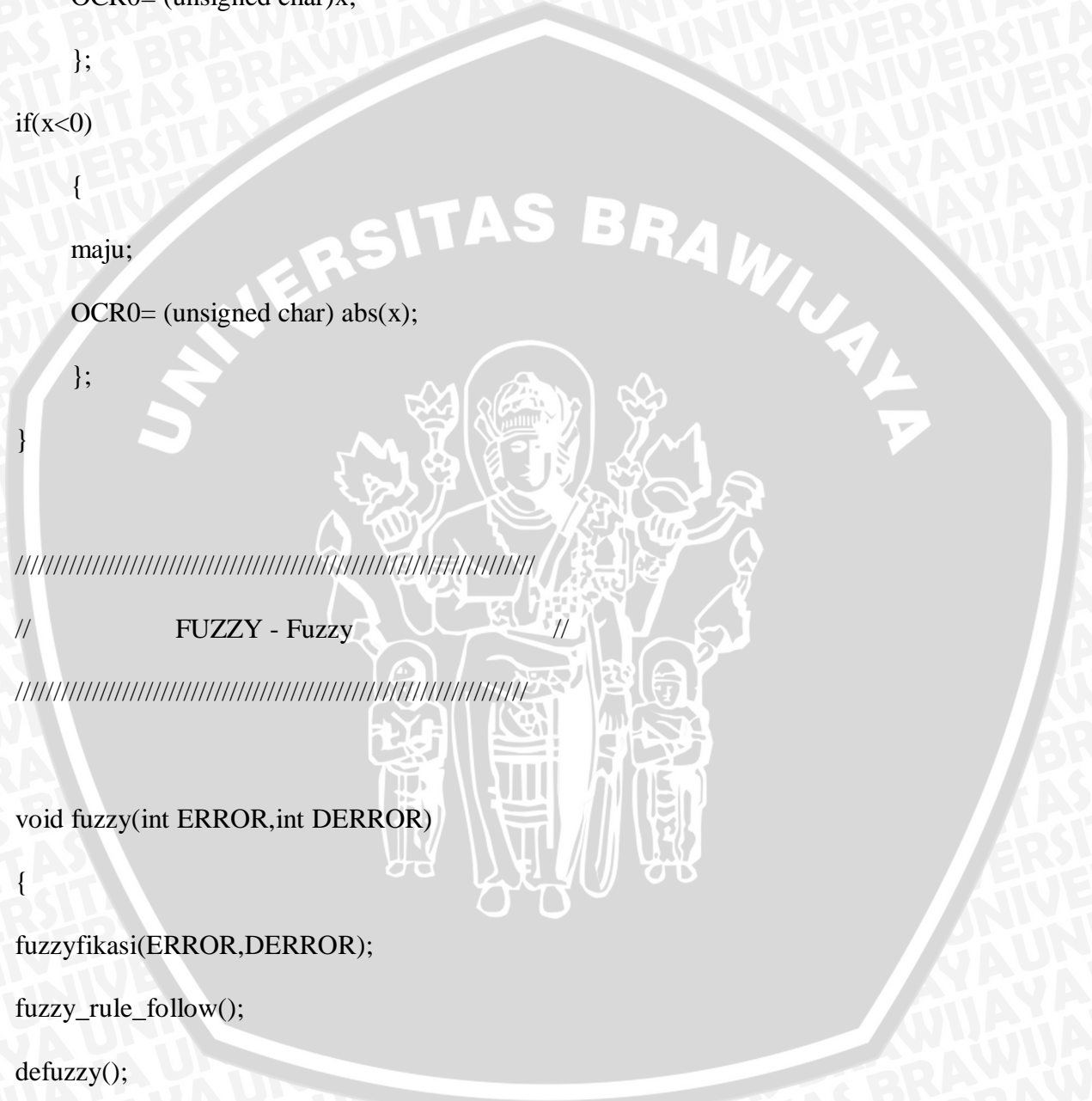
////////////////////////////////////
//          FUZZY - Fuzzy          //
////////////////////////////////////

void fuzzy(int ERROR,int DERROR)
{
fuzzyfikasi(ERROR,DERROR);

fuzzy_rule_follow();

defuzzy();
}

```





```

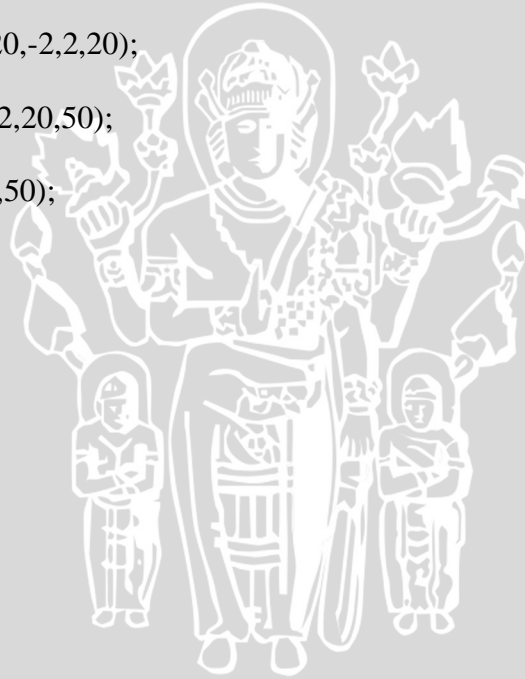
void fuzzyfikasi(int ERR,int DERR)
{
    e.NB = reverse_grade(ERR,-50,-25);
    e.NS = triangle(ERR,-50,-10,5);
    e.Z = trapezoid(ERR,-10,-5,5,10);
    e.PS = triangle(ERR,5,10,50);
    e.PB = grade(ERR,25,50);
    de.NB = reverse_grade(DERR,-50,-20);
    de.NS = triangle(DERR,-50,-20,-2);
    de.Z = trapezoid(ERR,-20,-2,2,20);
    de.PS = triangle(DERR,-2,20,50);
    de.PB = grade(DERR,20,50);
}

```

```

void fuzzy_rule_follow()
{
    float x;
    def.NB=0;
    def.NS=0;
    def.Z=0;
    def.PS=0;
    def.PB=0;
}

```



```

if(e.NB>0)
{
  if(de.NB>0)
  {
    x=fmin(e.NB,de.NB);
    def.NB =fmax(x,def.NB);
  }
  if(de.NS>0)
  {
    x=fmin(e.NB,de.NS);
    def.NB =fmax(x,def.NB);
  }
  if(de.Z>0)
  {
    x=fmin(e.NB,de.Z);
    def.NS =fmax(x,def.NS);
  }
  if(de.PS>0)
  {
    x=fmin(e.NB,de.PS);
    def.NS =fmax(x,def.NS);
  }
  if(de.PB>0)
  {

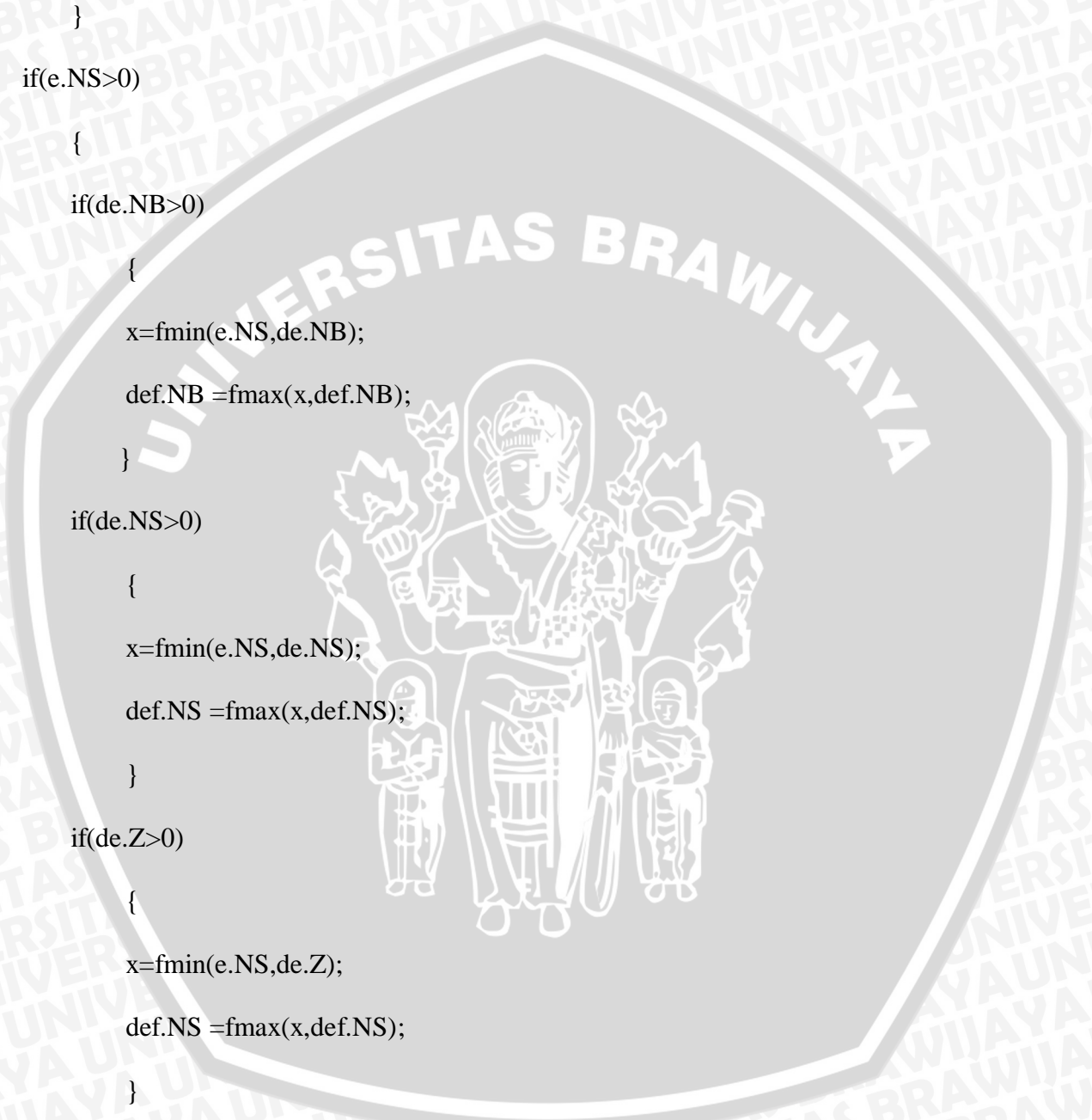
```



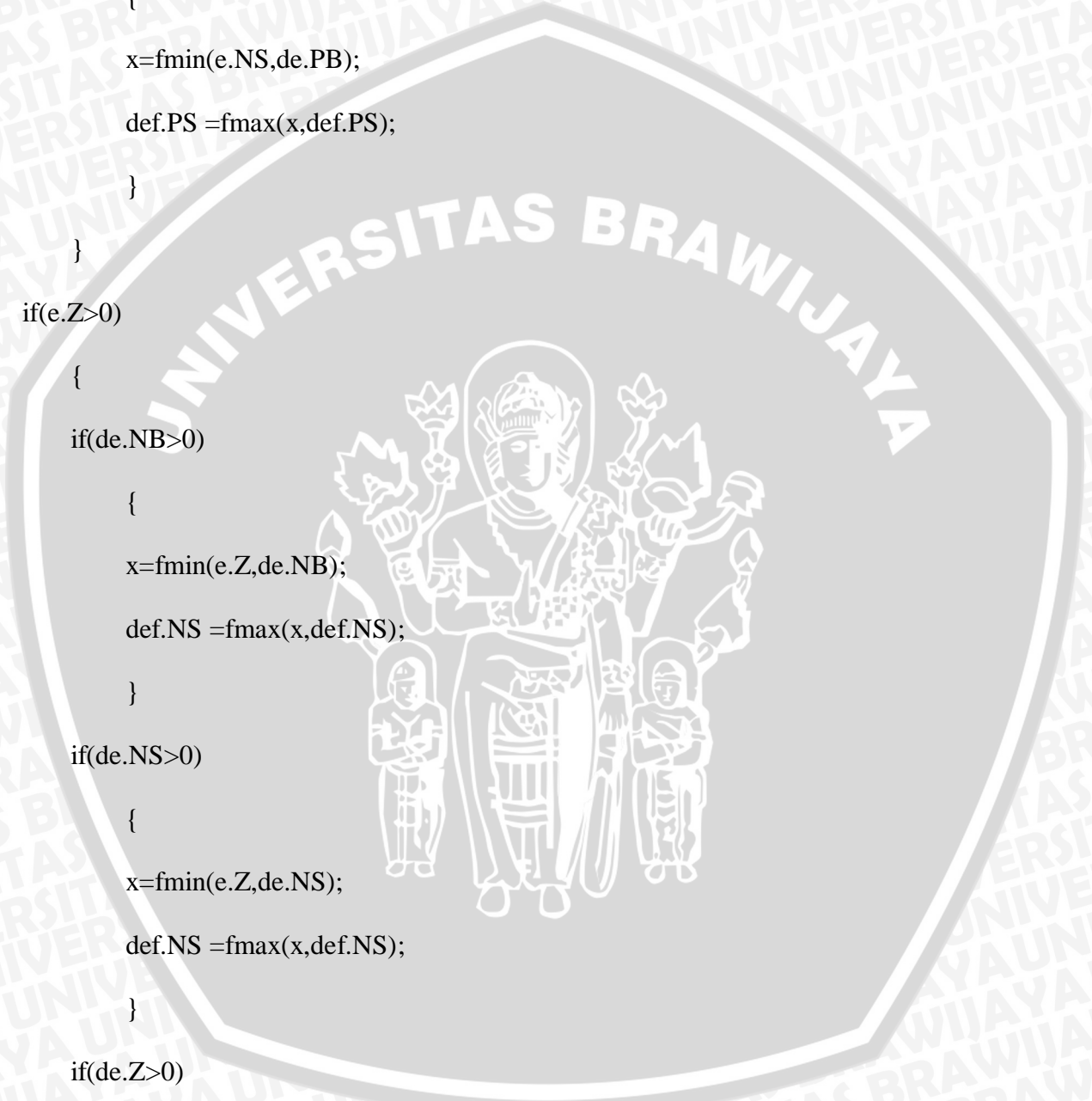
```

x=fmin(e.NB,de.PB);
def.Z =fmax(x,def.Z);
}
}
if(e.NS>0)
{
if(de.NB>0)
{
x=fmin(e.NS,de.NB);
def.NB =fmax(x,def.NB);
}
if(de.NS>0)
{
x=fmin(e.NS,de.NS);
def.NS =fmax(x,def.NS);
}
if(de.Z>0)
{
x=fmin(e.NS,de.Z);
def.NS =fmax(x,def.NS);
}
if(de.PS>0)
{
x=fmin(e.NS,de.PS);

```



```
def.Z =fmax(x,def.Z);
}
if(de.PB>0)
{
x=fmin(e.NS,de.PB);
def.PS =fmax(x,def.PS);
}
}
if(e.Z>0)
{
if(de.NB>0)
{
x=fmin(e.Z,de.NB);
def.NS =fmax(x,def.NS);
}
if(de.NS>0)
{
x=fmin(e.Z,de.NS);
def.NS =fmax(x,def.NS);
}
}
if(de.Z>0)
{
x=fmin(e.Z,de.Z);
def.Z =fmax(x,def.Z);
```



```

    }
    if(de.PS>0)
    {
        x=fmin(e.Z,de.PS);
        def.PS =fmax(x,def.PS);
    }

```

```

if(de.PB>0)
{
    x=fmin(e.Z,de.PB);
    def.PS =fmax(x,def.PS);
}

```

```

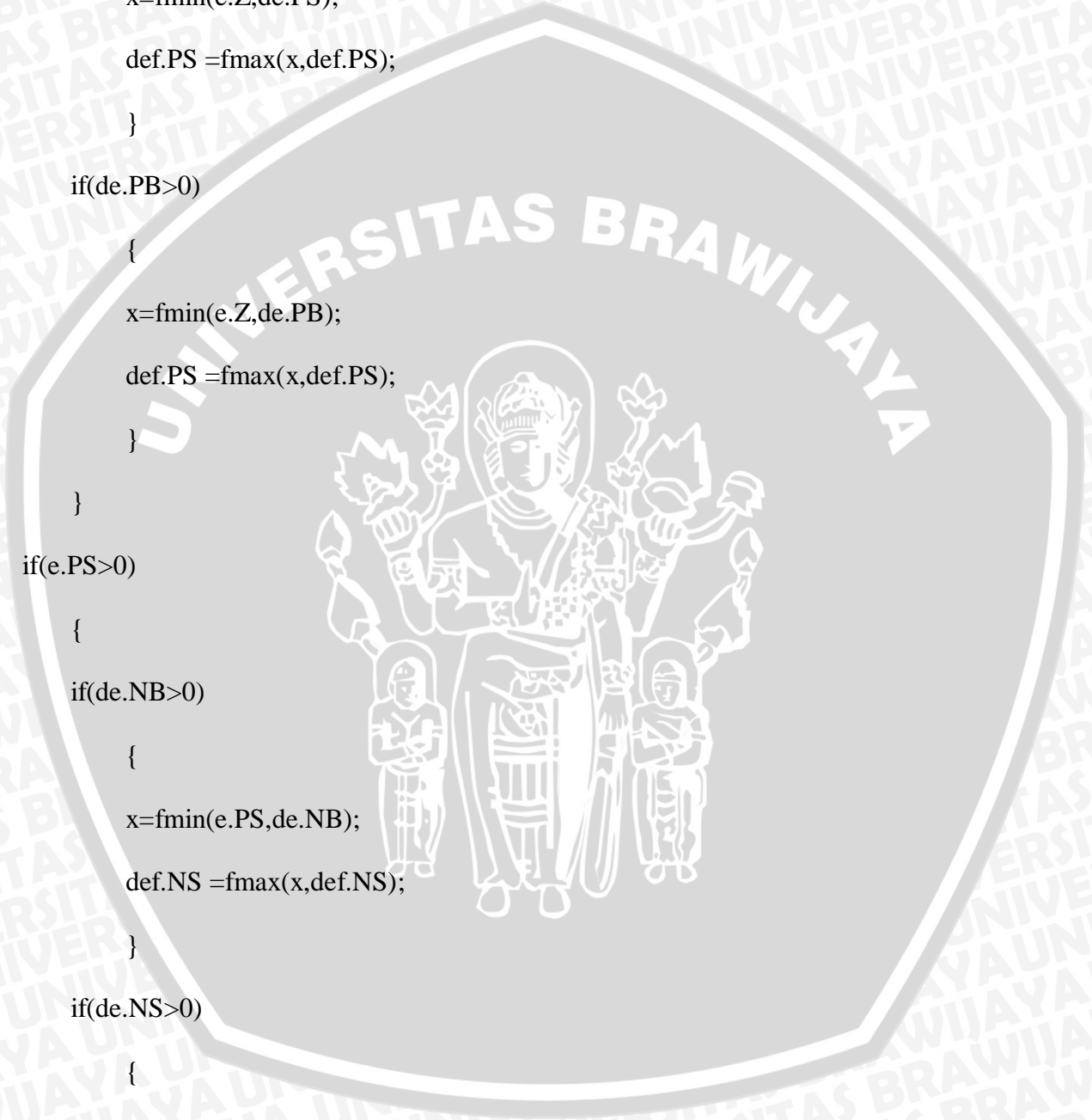
}
if(e.PS>0)
{
    if(de.NB>0)
    {
        x=fmin(e.PS,de.NB);
        def.NS =fmax(x,def.NS);
    }

```

```

    if(de.NS>0)
    {
        x=fmin(e.PS,de.NS);
        def.Z =fmax(x,def.Z);
    }
}

```



```

if(de.Z>0)
{
x=fmin(e.PS,de.Z);
def.PS =fmax(x,def.PS);
}

```

```

if(de.PS>0)
{
x=fmin(e.PS,de.PS);
def.PS =fmax(x,def.PS);
}

```

```

if(de.PB>0)
{
x=fmin(e.PS,de.PB);
def.PB =fmax(x,def.PB);
}
}

```

```

if(e.PB>0)
{

```

```

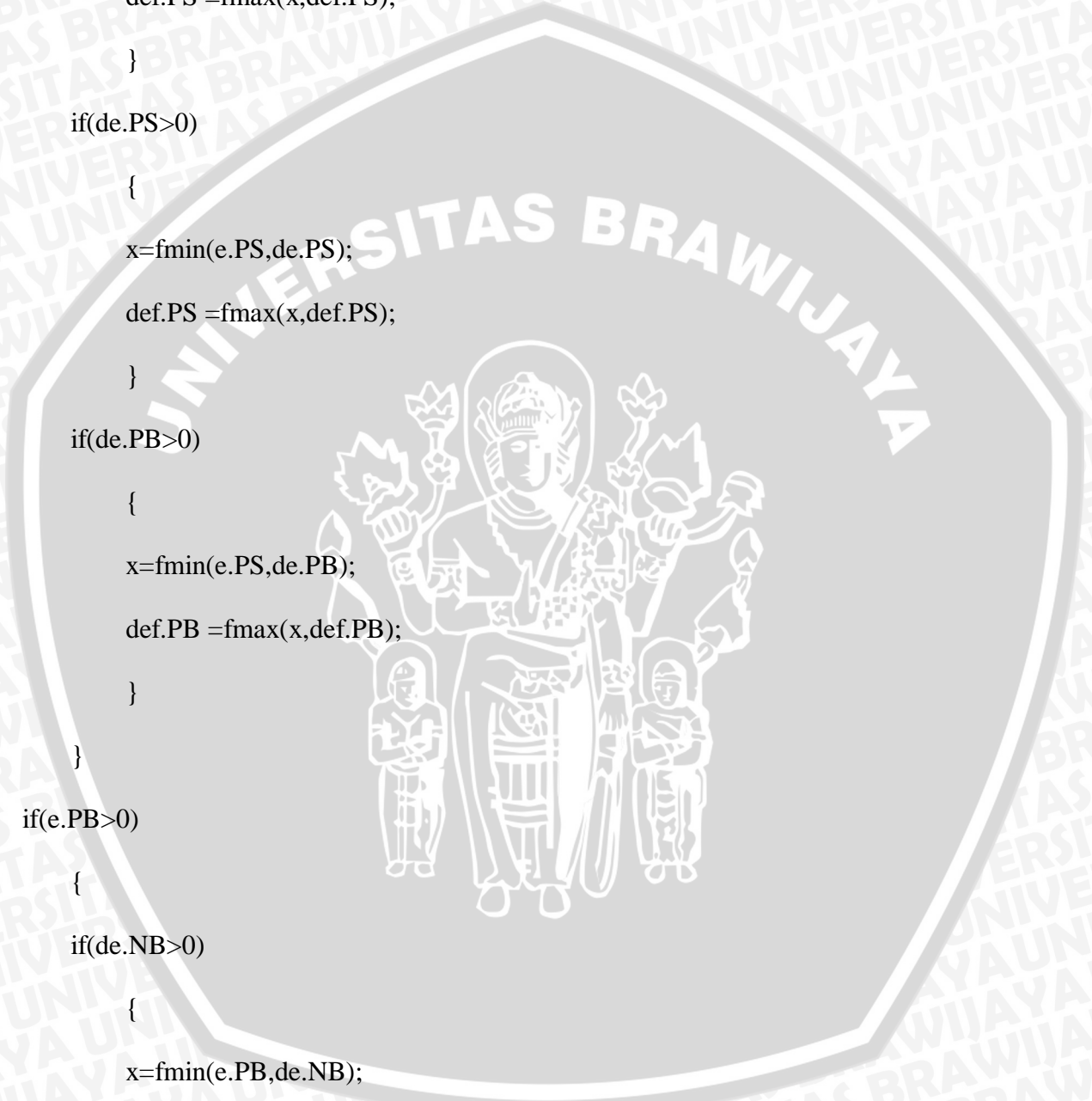
if(de.NB>0)
{
x=fmin(e.PB,de.NB);
def.Z =fmax(x,def.Z);
}
}

```

```

if(de.NS>0)

```



```

    {
        x=fmin(e.PB,de.NS);
        def.PS =fmax(x,def.PS);
    }
    if(de.Z>0)
    {
        x=fmin(e.PB,de.Z);
        def.PS =fmax(x,def.NS);
    }
    if(de.PS>0)
    {
        x=fmin(e.PB,de.PS);
        def.PB =fmax(x,def.PB);
    }
    if(de.PB>0)
    {
        x=fmin(e.PB,de.PB);
        def.PB =fmax(x,def.PB);
    }
}
}

```

```

void defuzzy()
{

```



```

int fuz;
float x=0,y=0;
//konstanta defuzzifikasi
K.NB =-255; K.NS =-100; K.Z=0; K.PS=100; K.PB=255;
//nilai defuzzifikasi
y = def.NB + def.NS + def.Z + def.PS + def.PB;
x = K.NB*def.NB + K.NS*def.NS + K.Z*def.Z + K.PS*def.PS + K.PB*def.PB;
fuz = (int) (x/y);
KONTROLER(fuz);
}

////////////////////////////////////
float triangle(int value, float x0, float x1, float x2) //x0 minimum, x1 medium, x2
maximum
{
float result = 0;
float x;
x = value;
if((x <= x0) || (x >= x2))
{result = 0;}
else if(x == x1)
{result = 10;}
else if((x>x0) && (x<x1))

```





```

{result = 10*((x-x0)/(x1-x0));}
else
{result =10*((x2-x)/(x2-x1));}
return result;
}

```

```

float trapezoid(int value,float x0,float x1,float x2,float x3) //x0 minimum, x1
medium min, x2 medium max, x3 maximum

```

```

{
float result = 0;
float x;
x=value;
if((x <= x0) || (x >= x3)){result = 0;}
else if((x>=x1) && (x<=x2)){result = 10;}
else if((x>x0) && (x<x1)){result = 10*((x-x0)/(x1-x0));}
else {result =10* ((x3-x)/(x3-x2));}
return result;
}

```

```

float grade(int value,float x0, float x1) //x0 minimum, x1 maximum for forward
membership

```

```


{
float result = 0;
float x;
x = value;

```



```
if(x <= x0){result = 0;}
    else if(x >= x1){result = 10;}
    else {result = 10*((x-x0)/(x1-x0));}
return result;
}
```

```
float reverse_grade(int value,float x0,float x1) //x0 minimum, x1 maximum
for reverse membership
{
float result = 0;
float x;
x = value;
if(x <= x0){result = 10;}
    else if(x >= x1)
    {result = 0;}
    else {result = 10*((x1-x)/(x1-x0));}
return result;
}
```



# LAMPIRAN V

## *DATA SHEET*

