

**IMPLEMENTASI KONTROLER LOGIKA FUZZY SEBAGAI KENDALI  
NUMERIS PROTOTIPE ALAT BUBUT  
BERBASIS MIKROKONTROLER AT89C51**

**SKRIPSI**

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Teknik



Disusun Oleh :

**WAHYU HIDAYAT**  
**NIM. 9901060408-63**

**FAKULTAS TEKNIK JURUSAN ELEKTRO  
UNIVERSITAS BRAWIJAYA  
MALANG  
2007**

**IMPLEMENTASI KONTROLER LOGIKA FUZZY SEBAGAI KENDALI  
NUMERIS PROTOTIPE ALAT BUBUT  
BERBASIS MIKROKONTROLER AT89C51**

**SKRIPSI**

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Teknik



Disusun Oleh :

**WAHYU HIDAYAT  
NIM. 9901060408-63**

Mengetahui dan menyetujui

**Dosen Pembimbing**

Ir. Erni Yudaningtyas,MT  
NIP. 131 879 035

Fitriana Suhartati,ST,MT  
NIP. 132 206 527

repository.ub.ac.id

**IMPLEMENTASI KONTROLER LOGIKA FUZZY SEBAGAI KENDALI  
NUMERIS PROTOTIPE ALAT BUBUT  
BERBASIS MIKROKONTROLER AT89C51**

Disusun Oleh :

**WAHYU HIDAYAT  
NIM. 9901060408-63**

Skripsi ini telah diuji dan dinyatakan lulus  
Pada tanggal 12 Agustus 2006

Dosen Penguji

**Ir. Chairuzzaini  
NIP. 130 682 589**

**Ir. Purwanto, MT.  
NIP. 131 574 847**

**Dipl.Ing. Ir. Moch. Rusli  
NIP. 131 653 473**

**Rusmi Ambarwati, ST, MT.  
NIP. 132 258 188**

Mengetahui

Ketua Jurusan Teknik Elektro

**Ir. Purwanto, MT.  
NIP. 131 574 847**

## KATA PENGANTAR

Alhamdulillah, dengan segala Rahmat serta Taufik yang begitu melimpah dari Allah SWT, *Pemilik segala jiwa* yang memberikan pengharapan serta keyakinan sehingga menghantarkan bantuan dari berbagai pihak akhirnya laporan skripsi dapat diselesaikan. Penulis berharap semoga skripsi ini dapat bermanfaat bagi para pembaca terutama mahasiswa jurusan Teknik Elektro Universitas Brawijaya Malang.

Penyusun mengucapkan terima kasih kepada :

1. Rasulullah Muhammad SAW, Sang pemberi suri tauladan terbaik bagi Umatnya. Semoga Shalawat dan Salam tetap tercurah kepada Beliau.
2. Ir. Erni Yudaningtyas, MT dan Fitriana Suhartati, ST, MT selaku dosen Pembimbing atas segala bimbingan dan arahannya kepada penulis.
3. Bapak dan Ibu Dosen serta segenap staf dan karyawan Jurusan Teknik Elektro.
4. Kepada kedua Orang Tua yang kuhormati dan kusayangi serta kedua kakakku, Mbak Yani & Fitri atas segala doa dan motivasinya.
5. Kepada teman-temanku *Crew C-334* yang selalu memberikan kemudahan dalam bentuk fasilitas plus semangat yang tiada habisnya
6. Kepada teman-teman Angkatan '99 seperjuangan yang selalu *support* semuanya.
7. Kepada teman-teman pencinta *Capoeira* atas suasana kekeluargaan dan kebersamaan yang hadir dalam melepas semua ketegangan yang ada
8. Kepada teman-teman LSP-RTD Malang, *saudara sejati* dalam berbagai situasi & kondisi, tempat berbagi, serta wadah perjuangan dinamis
9. Kepada teman-teman dan pihak-pihak yang tidak dapat penulis sebutkan satu persatu. Terima atas segalanya

Malang, April 2007

Penulis

## ABSTRAK

**Wahyu Hidayat, Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya Malang, April 2006, Implementasi Kontroler Logika Fuzzy Sebagai Kendali Numeris Prototipe Alat Bubut Berbasis Mikrokontroler AT89C51**  
**Pembimbing: Ir. Erni Yudaningtyas, MT dan Fitriana Suhartati, ST, MT**

*Di dalam industri perkakas rumah tangga, mesin bubut konvensional masih sering dipergunakan sebagai mesin yang berfungsi untuk membentuk, memotong, serta meratakan permukaan benda kerja. Hasil produksi dari mesin-mesin bubut konvensional masih sangat bergantung pada keterampilan dari operator yang menggunakannya sehingga menyebabkan keseragaman produk tidak optimal, laju produksi rendah, serta melibatkan operator yang lebih banyak. Untuk mengatasi permasalahan tersebut maka perlunya menggunakan mesin bubut yang dikendalikan secara numeris. Dalam hal ini Prototipe alat bubut yang dirancang menggunakan mikrokontroler AT89C51 sebagai kontroler logika fuzzy.*

*Kontrol logika fuzzy pada prototipe alat bubut ini memiliki 2 masukan yaitu Error dan  $\Delta$ Error dan 1 Keluaran yaitu Output. Data posisi masukan yang berasal dari sebuah komputer personal kemudian diolah menggunakan Mikrokontroler AT89C51 dan hasilnya dikeluarkan untuk menggerakkan alat untuk membubut benda kerja seberat 0,2 kg berdiameter 4,5 cm dan panjang 9,5 cm pada posisi  $x$  dan  $z$ . Dari hasil 4 pengujian dengan data koordinat  $x$  dan  $z$  yang berbeda, alat ini dapat bekerja pada sesuai dengan perencanaan dengan kesalahan berkisar antara 0,5-1,5% dan kesalahan rata-rata sistem sebesar 1,4335%, sehingga prototipe alat bubut dapat bergerak sesuai dengan koordinat  $x$  dan  $z$  yang diberikan menuju posisi yang diinginkan.*

**DAFTAR ISI**

	Halaman
KATA PENGANTAR .....	i
ABSTRAK .....	ii
DAFTAR ISI .....	iii
DAFTAR GAMBAR .....	vi
DAFTAR TABEL .....	viii

**BAB I. PENDAHULUAN**

1.1. Latar Belakang .....	1
1.2. Rumusan Masalah .....	3
1.3. Batasan Masalah .....	3
1.4. Tujuan .....	4
1.5. Sistematika Penulisan .....	5

**BAB II. TEORI ALAT BUBUT, MOTOR DC *STEPPER*, OPTOCOUPLER, MIKROKONTROLER DAN LOGIKA FUZZY**

2.1. Dasar Teori Alat Bubut .....	6
2.1.1. Gambaran Umum Alat bubut .....	6
2.1.2. Jenis Alat Bubut .....	6
2.1.3. Ukuran Alat Bubut .....	7
2.2. Motor DC .....	7
2.2.1. Jenis Motor DC <i>Stepper</i> .....	7
2.2.2. Prinsip kerja Motor DC <i>Stepper</i> .....	10
2.2.3. Penggerak Motor DC <i>Stepper</i> .....	12
2.2.4. Analisis DC .....	13
2.3. Optocoupler .....	14
2.3.1. Prinsip Kerja Optocoupler .....	14



2.4. Mikrokontroler AT89C51 .....	15
2.4.1. Struktur dan Operasi Port.....	16
2.4.2. Sistem Interupsi.....	18
2.5. DAC ( <i>Digital Analog Converter</i> ) .....	18
2.6. Dasar Teori Logika <i>Fuzzy</i> .....	19
2.6.1. Himpunan <i>Fuzzy</i> .....	20
2.6.2. Fungsi Keanggotaan <i>Fuzzy</i> .....	21
2.6.3. Operasi Himpunan <i>Fuzzy</i> .....	23
2.6.4. Variabel Linguistik ( <i>Linguistic Variable</i> ).....	25
2.6.5. Aturan <i>Fuzzy</i> “ <i>If-Then</i> ”.....	26
2.6.6. Kontrol Logika <i>Fuzzy</i> .....	28
2.6.6.1. <i>Fuzzifikasi</i> .....	28
2.6.6.2. <i>Defuzzifikasi</i> .....	29
2.6.7. Struktur Dasar Kontroler Logika <i>Fuzzy</i> .....	30

### **BAB III. METODOLOGI PENELITIAN**

3.1. Penentuan Spesifikasi Alat.....	33
3.2. Peralisasian alat.....	34
3.2.1. Perancangan Perangkat Keras dan Peralisasian Tiap Blok .....	34
3.2.2. Perancangan dan Peralisasian perangkat Lunak.....	34
3.3. Pengujian Rancangan .....	34
3.3.1. Pengujian Perangkat Keras .....	35
3.3.2. Pengujian Perangkat Lunak .....	35
3.3.3. Pengujian keseluruhan Sistem.....	35
3.4. Analisa Data dan Pengambilan Kesimpulan .....	35

### **BAB IV. PERANCANGAN DAN PEMBUATAN ALAT**

4.1. Perancangan Perangkat Keras.....	36
4.1.1. Cara Kerja Alat.....	36
4.1.2. Sensor Posisi .....	37
4.1.3. Mikrokontroler AT89C51 .....	39
4.1.4. Digital to Analog Converter.....	41

4.1.5. Rangkaian Driver Motor DC.....	42
4.1.6. Perancangan Kontroler Logika <i>Fuzzy</i> .....	42
4.2. Perancangan Perangkat Lunak.....	48

**BAB V. PENGUJIAN ALAT**

5.1. Pengujian Perangkat Keras.....	49
5.1.1. Pengujian Sensor Posisi.....	49
5.1.2. Pengujian DAC.....	50
5.1.3. Pengujian Driver Motor DC.....	52
5.2. Pengujian Perangkat Lunak.....	54
5.2.1. Pengujian Mikrokontroler.....	54
5.2.2. Pengujian <i>Fuzzy Inference System</i> .....	55
5.3. Pengujian Sistem Secara Keseluruhan.....	57

**BAB VI. KESIMPULAN DAN SARAN**

6.1. Kesimpulan.....	63
6.2. Saran.....	63

<b>DAFTAR PUSTAKA.....</b>	<b>65</b>
----------------------------	-----------

**LAMPIRAN**

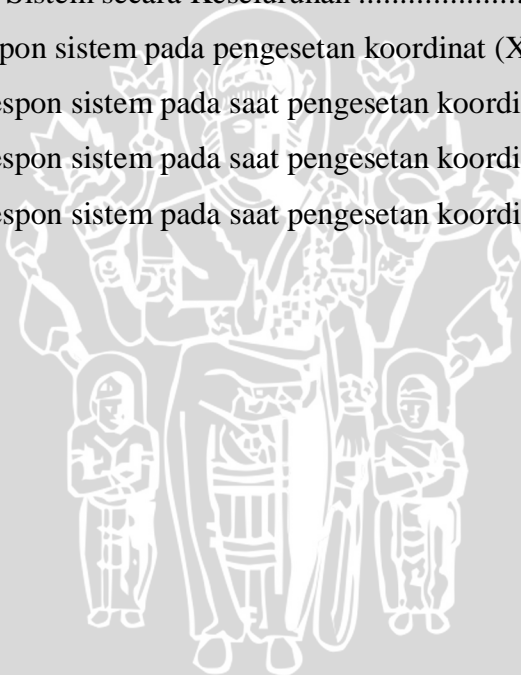




## DAFTAR GAMBAR

	Halaman
Gambar 2.1. Pembubut mesin konvensional seri Longem 32" & 40" .....	7
Gambar 2.2. Motor <i>Stepper</i> Unipolar .....	8
Gambar 2.3. Motor <i>Stepper</i> Bipolar .....	9
Gambar 2.4. Motor <i>Stepper</i> Unipolar 2-Phase .....	11
Gambar 2.5 Penguat Darlington .....	13
Gambar 2.6. <i>Optocoupler</i> .....	14
Gambar 2.7. Diagram Rangkaian <i>Optocoupler</i> .....	14
Gambar 2.8. Diagram Blok Mikrokontroler AT89C51 .....	16
Gambar 2.9. DAC R-2R delapan bit.....	19
Gambar 2.10. S-Function .....	22
Gambar 2.11. $\pi$ -Function .....	22
Gambar 2.12. T-Function .....	23
Gambar 2.13. Contoh operasi gabungan himpunan <i>fuzzy</i> A dan B .....	24
Gambar 2.14. Contoh operasi irisan himpunan <i>fuzzy</i> A dan B.....	24
Gambar 2.15. Contoh operasi komplemen himpunan <i>fuzzy</i> A .....	25
Gambar 2.16. Diagram representasi dari <i>fuzzy</i> kecepatan.....	26
Gambar 2.17. Metode Penalaran Fuzzy MAX-MIN .....	27
Gambar 2.18. Metode Penalaran fuzzy MAX-DOT .....	28
Gambar 2.19. Struktur kontroler logika <i>fuzzy</i> dasar .....	31
Gambar 4.1. Perancangan Perangkat Keras .....	36
Gambar 4.2. Rangkaian Sensor Posisi .....	37
Gambar 4.3. Pulsa keluaran yang menunjukkan posisi rotor.....	38
Gambar 4.4. Tempat pemasangan sensor posisi.....	38
Gambar 4.5. Mikrokontroler AT89C51 dengan Rangkaian Pendukungnya.....	40
Gambar 4.6. Rangkaian DAC.....	41
Gambar 4.7. Rangkaian Driver Motor DC.....	42
Gambar 4.8. Kontroler Logika Fuzzy .....	42
Gambar 4.9. Fungsi Keanggotaan <i>Crisp Input Error</i> .....	43

Gambar 4.10. Fungsi Keanggotaan <i>Crisp Input</i> $\Delta$ Error.....	40
Gambar 4.11. Fungsi Keanggotaan <i>Crisp Output</i> .....	40
Gambar 4.12. Grafik respon output terhadap waktu.....	45
Gambar 4.13. Diagram alir program utama .....	48
Gambar 5.1. Pengujian Sensor Posisi .....	49
Gambar 5.2. Pengujian DAC.....	50
Gambar 5.3. Grafik Keluaran DAC.....	52
Gambar 5.4. Pengujian <i>Driver</i> Motor DC.....	52
Gambar 5.5. Diagram Alir program motor.h.51.....	54
Gambar 5.6. Pengujian Mikrokontroler .....	55
Gambar 5.7. Pengujian <i>Fuzzy Inference System</i> .....	56
Gambar 5.8. Pengujian Sistem secara Keseluruhan .....	57
Gambar 5.9. Grafik respon sistem pada pengesetan koordinat (X0,Z5).....	58
Gambar 5.10. Grafik respon sistem pada saat pengesetan koordinat (X10,Z10)..	59
Gambar 5.11. Grafik respon sistem pada saat pengesetan koordinat (X15,Z15)..	60
Gambar 5.12. Grafik respon sistem pada saat pengesetan koordinat (X25,Z20)..	61



**DAFTAR TABEL**

	Halaman
Tabel 2.1. Pemberian pulsa langkah penuh ( <i>full step</i> ) .....	11
Tabel 2.2. Pemberian pulsa langkah penuh ( <i>full step</i> ) .....	11
Tabel 2.3. Pemberian pulsa langkah setengah ( <i>half step</i> ) .....	12
Tabel 2.4. Fungsi Tambahan <i>Port 3</i> .....	17
Tabel 2.5. Alamat Interupsi.....	18
Tabel 2.6. Kaidah atur dalam bentuk matrik.....	32
Tabel 4.1. Kaidah Atur.....	46
Tabel 5.1. Hasil Pengujian Sensor Posisi.....	49
Tabel 5.2. Hasil Pengujian DAC .....	51
Tabel 5.3. Data Hasil Pengujian <i>Driver Motor DC</i> .....	53
Tabel 5.4. Hasil Pengujian Sistem mikrokontroler.....	55
Tabel 5.5. Hasil Pengujian <i>Fuzzy Inference System</i> .....	56
Tabel 5.6. Hasil Pengujian Koordinat (X0,Z5) .....	57
Tabel 5.7. Hasil Pengujian Koordinat (X10,Z10) .....	59
Tabel 5.8. Hasil Pengujian Koordinat (X15,Z15) .....	60
Tabel 5.9. Hasil Pengujian Koordinat (X25,Z20) .....	61



## BAB I

### PENDAHULUAN

#### 1.1 Latar Belakang

Seiring dengan pesatnya teknologi yang berkembang di berbagai negara baik negara maju ataupun negara berkembang, masih banyak diantara masyarakat Indonesia yang bergantung dengan mesin-mesin yang bersifat konvensional dalam pembuatan berbagai peralatan perkakas rumah tangga. Salah satunya adalah mesin bubut yang biasanya digunakan untuk membentuk benda kerja dengan cara membuang sebagian material benda kerja sampai mencapai ukuran yang diinginkan.

Kualitas piranti mesin bubut konvensional sangat bervariasi. Mesin-mesin bubut tersebut dirancang dengan harga tertentu untuk memenuhi kebutuhan pasar. Secara umum, semakin mahal harga mesin, semakin tinggi kualitas kerjanya. Bagaimanapun, mesin bubut konvensional yang mahal tidak menjamin kualitas hasil kerja yang tinggi. Kunci keberhasilan mesin bubut konvensional bergantung pada keterampilan operator. Pekerja yang terampil dalam membuat keputusan-keputusan dalam pengoperasian mesin bubut konvensional masih menggunakan kemampuan intuisinya dalam menghasilkan keseragaman produk, mengurangi pemborosan komponen, jumlah (kuantitas) produk yang dihasilkan.

Walaupun keterampilan operator merupakan kunci keberhasilan dalam pengoperasian manual suatu mesin bubut konvensional, masih banyak faktor-faktor lain yang mempengaruhi produk hasil proses pembubutan dengan menggunakan mesin konvensional. Keterampilan ini akan nampak nyata dalam mempengaruhi akurasi suatu produk akhir, yaitu pada saat penentuan posisi (lewat pergeseran mesin) pahat dan benda kerja dengan tepat. Keterampilan pengoperasian mesin juga merupakan faktor kritis bila mesin dikontrol secara elektronik. [Gibbs,D & Crandell, T.M,1991: 7]

Untuk menyikapi masalah-masalah yang timbul dalam pengoperasian mesin bubut konvensional, diperlukan suatu pengaturan numerik (numerical control) agar produk akhir dalam suatu proses pembubutan dapat menghasilkan produk dengan bentuk yang seragam, mempunyai laju produksi yang tinggi (cepat), dan tidak memerlukan banyak operator dalam pengoperasiannya. Kontrol Numerik ( Numerical Control) adalah istilah yang digunakan untuk menjelaskan kontrol gerakan mesin dan berbagai fungsi lainnya yang menggunakan instruksi yang dinyatakan dalam seri bilangan dan dikendalikan oleh sistem kontrol elektronika. [Gibbs,D & Crandell, T.M,1991: 7]

Algoritma fuzzy merupakan algoritma yang mudah dan handal untuk diaplikasikan sebagai pengendali pada berbagai macam sistem dengan hanya menggunakan variabel-variabel masukan dan keluaran sistem tanpa harus mengetahui fungsi alih atau persamaan dinamik sistem tersebut. Keuntungan lain pemakaian algoritma fuzzy adalah kemudahan dalam memperbaiki unjuk kerja sistem yaitu dengan mengubah softwarena tanpa harus mengubah konfigurasi hardwarena.

Untuk mempelajari penggunaan alat bubut sebagai salah satu aplikasi kontrol logika fuzzy maka perlu dibuat suatu model sistem yang menggunakan suatu bentuk prototipe Alat bubut dengan menggunakan Mikrokontroler AT89C51 sebagai modul aplikasi fuzzy. Dalam hal ini kontrol logika fuzzy dibutuhkan dalam melakukan kontrol numerik dalam mengatur motor DC yang merupakan media pergeseran pada mesin bubut agar dapat mengkopel mesin bubut terhadap benda kerja sehingga benda kerja dapat di bentuk sesuai dengan sketsa yang diinginkan.

### 1.2 Rumusan Masalah

Dari latar belakang masalah diatas dapat dirumuskan permasalahan sebagai berikut:

1. Bagaimana merancang sistem kontrol logika fuzzy untuk sistem pergerakan prototipe alat bubut yang bergerak sesuai dengan arah koordinat  $x$  dan  $z$  menggunakan motor DC.
2. Bagaimana merancang alat bubut menggunakan modul Mikrokontroler AT89C51 agar dapat membentuk suatu benda kerja sesuai dengan performansi yang diinginkan.

### 1.3 Batasan Masalah

Masalah yang dirumuskan akan dibatasi beberapa hal sebagai berikut :

1. Permasalahan dititikberatkan pada masalah Logika Fuzzy sebagai pengendali.
2. Tidak membahas perangkat keras secara mendalam.

3. Jenis prototipe alat bubut yang digunakan adalah jenis alat bubut *kegunaan khusus*.
4. Motor DC diatur dengan kecepatan konstan.
5. Pembuatan perangkat lunak (*software*) yang digunakan berupa bahasa pemrograman Assembler mencakup *setpoint*, kontroler, dan gangguan.
6. Pembuatan perangkat lunak (*software*) sebagai media sketsa kontur bahan atau obyek menggunakan bahasa pemrograman Delphi 5.0.
7. Tidak membahas mengenai bahan baku yang digunakan.
8. Tidak membahas mengenai catu daya yang digunakan prototipe alat.

#### 1.4 Tujuan

Skripsi ini bertujuan untuk mendesain kontroler logika fuzzy berbasis Mikrokontroler AT89C51 untuk mengendalikan arah pergeseran suatu prototipe alat bubut terhadap panjang bahan berdasarkan koordinat  $x$  dan  $z$  dalam hal membentuk kontur suatu bahan atau benda kerja.

## 1.5 Sistematika Penulisan

Sistematika penulisan dalam skripsi ini sebagai berikut:

### BAB I Pendahuluan

Memuat latar belakang, rumusan masalah, tujuan, batasan masalah, tujuan dan sistematika pembahasan.

### BAB II Teori Penunjang

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

### BAB III Metodologi

Berisi tentang metode penelitian dan perencanaan alat serta pengujian.

### BAB IV Perencanaan dan Pembuatan Alat

Perancangan dan perealisasiian kontroler logika fuzzy untuk kendali arah pergeseran prototipe alat bubut terhadap benda kerja

### BAB V Pengujian

Memuat hasil pengujian terhadap alat yang telah direalisasikan.

### BAB VI Kesimpulan dan Saran

Memuat kesimpulan dan saran-saran.



## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Dasar Teori Alat Bubut

##### 2.1.1 Gambaran Umum Alat Bubut

Mesin bubut mempunyai gerak utama berputar dan berfungsi sebagai pengubah bentuk dan ukuran benda dengan jalan menyayat benda dengan suatu pahat penyayat, posisi benda kerja berputar sesuai dengan sumbu mesin dan pahat diam bergerak ke kanan atau ke kiri searah dengan sumbu mesin bubut menyayat benda kerja [Daryanto,2002: 1]

Alat bubut disesuaikan dengan untuk pekerjaan silindris dan mata pahat tunggal dalam membentuk suatu benda kerja menjadi bentuk yang silindris dengan cara bergerak sesuai dengan 3 koordinat yaitu,  $x$ ,  $y$ , dan  $z$ . Dalam skripsi ini prototipe alat bubut dititikberatkan pada pergerakan koordinat  $x$  dan  $z$  saja dalam membentuk suatu kontur bahan atau benda kerja.

##### 2.1.2 Jenis Alat Bubut

Penggolongan yang sesuai dari alat bubut adalah sulit karena terdapat keanekaragaman dalam ukuran, desain, metode penggerakan, dan kegunaan. Pada umumnya diberi nama sesuai dengan karakteristik disainnya yang menonjol. Penggolongan dari alat bubut :

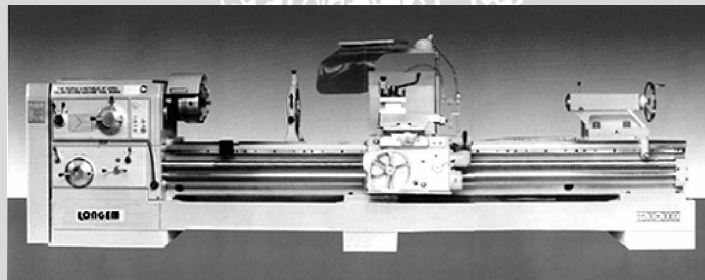
- a. Pembubut Kecepatan (*Speed Lathe*)
- b. Pembubut Mesin (*Machine Lathe*)
- c. Pembubut bangku
- d. Pembubut Ruang Perkakas
- e. Pembubut Kegunaan Khusus
- f. Pembubut Turet
- g. Pembubut Otomatis

Penggolongan ini dapat dibagi lagi menurut bentuk khususnya ( misalnya banyaknya atau pengaturan dari peluncur pahat ) dan penggunaan dari mesin (

misalnya untuk stok batangan dan pecekaman atau kombinasi dari keduanya ). Juga, beberapa dari alat bubut yang disebutkan dapat disesuaikan untuk kendali numeris menggunakan pita berlubang [Gibbs,D & Crandell T.M,1993: 7]

### 2.1.3 Ukuran Alat Bubut

Ukuran alat bubut dinyatakan dalam diameter benda kerja yang dapat diputar, sehingga sebuah alat bubut 400 mm adalah alat yang memiliki ruang bebas cukup diatas rel bangku untuk mengerjakan diameter 400 mm. Tetapi, ukuran kedua diperlukan untuk menentukan kapasitas ukuran selanjutnya dari alat, dalam pernyataan panjang benda kerja. Beberapa pabrik menyatakannya dalam panjang maksimum benda kerja diantara kedua pusat alat bubut, sedangkan pabrik lain menyatakannya dalam panjang angka.[Rahardjo, T,2005: 6]



**Gambar 2.1.** Pembubut mesin konvensional seri Longem 32" & 40"

Sumber : <http://www.mec-tools.com/en/product1show.asp>

## 2.2 Motor Stepper

### 2.2.1 Jenis Motor Stepper

Motor *stepper* merupakan motor DC yang tidak memiliki komutator. Pada umumnya motor *stepper* hanya mempunyai kumparan pada statornya sedangkan pada bagian rotornya merupakan permanen magnet. Dengan model motor seperti ini maka motor *stepper* dapat diatur posisinya pada posisi tertentu dan berputar ke arah yang diinginkan, searah jarum jam atau sebaliknya.

Kecepatan motor *stepper* pada dasarnya ditentukan oleh kecepatan pemberian data pada statornya. Semakin cepat data yang diberikan maka motor *stepper* akan

semakin cepat pula berputarnya. Pada kebanyakan motor *stepper* kecepatannya dapat diatur dalam daerah frekuensi audio dan akan menghasilkan putaran yang cukup cepat.

Tipe motor *Stepper* berdasarkan sifat kemagnetan stator maupun rotornya dibedakan menjadi tiga macam yaitu :

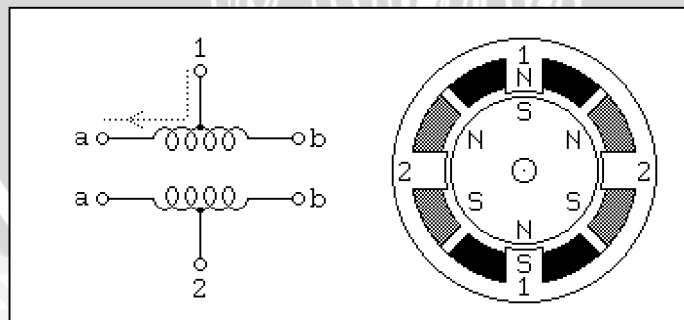
- Motor *stepper* PM (*Permanent Magnet*)
- Motor *stepper* VR (*Variable Reluctance*)
- Motor *stepper* Hibrid

Pada tipe permanent magnet rotor terbuat dari magnet permanen. Operasi motor tipe ini dilakukan dengan pensaklaran secara bergantian. Pada motor tipe *variable reluctance* mempunyai gigi-gigi yang berpasangan dan saling berhadapan. Jumlah gigi pada rotor dan stator tidak sama, sehingga dengan prinsip bahwa susunan magnet selalu berusaha meminimalkan jarak celah dalam bidang magnetik antara gigi rotor dengan stator, maka motor dapat dioperasikan.

Tipe motor *stepper* berdasarkan susunan lilitannya dibedakan menjadi dua yaitu :

- Tipe Unipolar

Motor *stepper* dengan tipe unipolar adalah motor *stepper* yang mempunyai 2 buah lilitan yang masing-masing lilitan ditengah-tengahnya diberikan sebuah tap seperti tampak pada Gambar 2.2.



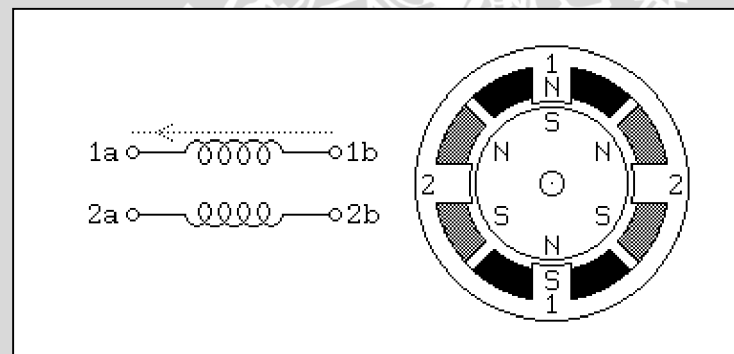
**Gambar 2.2** Motor *Stepper* Unipolar

Sumber : W. Susanto, 2002:06

Motor ini mempunyai step tiap  $30^{\circ}$  dan mempunyai dua buah lilitan yang didistribusikan berseberangan  $180^{\circ}$  di antara kutub pada stator. Sedangkan pada rotornya menggunakan magnet permanen yang berbentuk silinder dengan mempunyai 6 buah kutub, 3 kutub selatan dan 3 buah kutub utara. Sehingga dengan konstruksi seperti ini maka jika dibutuhkan kepresisian dari motor stepper yang lebih tinggi dibutuhkan pula kutub-kutub pada stator dan rotor yang semakin banyak pula. Pada Gambar 2.2, motor tersebut akan bergerak setiap step sebesar  $30^{\circ}$  dengan 4 bit urutan data (terdapat dua buah lilitan dengan tap, total lilitan menjadi 4 lilitan). Ketelitian dari magnet permanen di rotor dapat sampai  $1.8^{\circ}$  untuk tiap stepnya.

- Tipe Bipolar

Motor dengan tipe bipolar ini mempunyai konstruksi yang hampir sama dengan motor *stepper* tipe unipolar namun tidak terdapat tap pada lilitannya, seperti tampak pada Gambar 2.3.



**Gambar 2.3** Motor *Stepper* Bipolar

Sumber : Susanto,W, 2002:06

Penggunaan motor dengan tipe bipolar ini membutuhkan rangkaian yang sedikit lebih rumit untuk mengatur agar motor ini dapat berputar dalam dua arah. Biasanya untuk menggerakkan motor stepper jenis ini membutuhkan sebuah driver motor yang sering dikenal sebagai *H-Bridge*. Rangkaian ini akan mengontrol tiap-tiap lilitan secara *independent* termasuk dengan polaritasnya untuk tiap-tiap lilitan.

### 2.2.2 Prinsip Kerja Motor Stepper

Apabila kumparan stator diberikan pulsa maka akan menimbulkan medan magnet. Medan magnet yang terjadinya pada kumparan stator menyebabkan terjadinya tarik menarik antara kutub stator dengan kutub magnet permanen rotor yang sifatnya berlawanan (kutub utara stator menarik kutub selatan rotor). Untuk mengatur kecepatan motor *stepper* dilakukan dengan cara mengatur selang waktu pemberian pulsa pada kumparan stator.

Motor *stepper* dapat mempunyai langkah yang kecil sehingga resolusi terbesar dapat dicapai. Hubungan sudut langkah  $\theta_n$  dan jumlah langkah S diberikan pada persamaan :

$$S = \frac{360}{\theta_n} \dots\dots\dots(2.1)$$

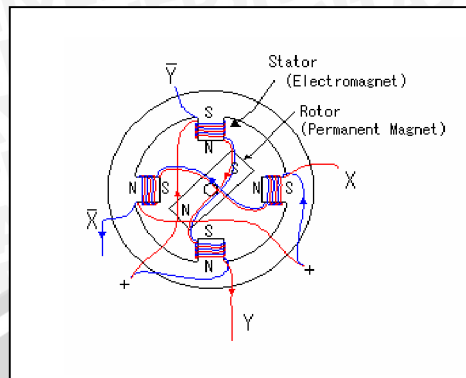
Dan kecepatan putaran motor langkah diberikan dalam jumlah perdetik dan laju langkah yaitu :

$$n = \frac{60 \cdot f}{S} \dots\dots\dots(2.2)$$

dimana :     n = kecepatan putar (rpm)  
               f = laju langkah (Hz)  
               S = jumlah langkah

Cara pemberian pulsa pada motor stepper unipolar 2- phase :

1. Memberikan arus kepada suatu kumparan satu persatu (X,Y,X',Y', dan seterusnya). Cara ini akan memberikan langkah penuh (*full step*).
2. Memberikan arus pada dua kumparan secara bergantian (X+Y,Y+X',X'+Y', dan seterusnya). Cara ini akan memberikan langkah penuh (*full step*)
3. Gabungan dari kedua cara akan memberikan langkah setengah (*half step*).



**Gambar 2.4.** Motor Stepper Unipolar 2-Phase

Sumber :.Jhon,D.W, 2002:08

**Tabel 2.1** Pemberian pulsa langkah penuh (*full step*)

Step	Lilitan				Desimal
	Y'	X'	Y	X	
1	0	0	0	1	1
2	0	0	1	0	2
3	0	1	0	0	4
4	1	0	0	0	8
5	0	0	0	1	1

Sumber :.Jhon, D.W, 2002:09

**Tabel 2.2** Pemberian pulsa langkah penuh (*full step*)

Step	Lilitan				Desimal
	Y'	X'	Y	X	
1	0	0	1	1	3
2	0	1	1	0	6
3	1	1	0	0	12
4	1	0	0	1	9
5	0	0	1	1	3

Sumber :.Jhon, D.W, 2002:10

**Tabel 2.3** Pemberian pulsa langkah setengah (*half step*)

Step	Lilitan				Desimal
	Y'	X'	Y	X	
1	0	0	0	1	1
2	0	0	1	1	3
3	0	0	1	0	2
4	0	1	1	0	6
5	0	1	0	0	4
6	1	1	0	0	12
7	1	0	0	0	8
8	1	0	0	1	9
9	0	0	0	1	1

Sumber : Jhon, D.W, 2002:10

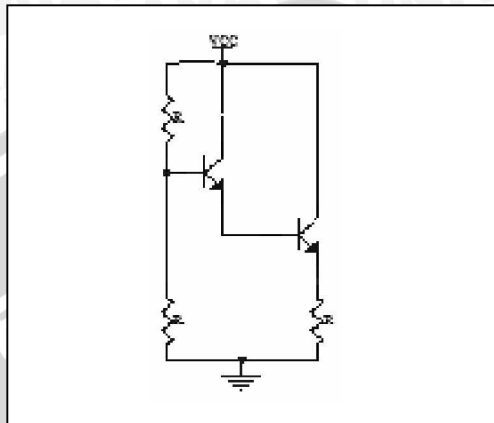
Arah putaran motor *stepper* ditentukan sesuai dengan arah pergantian pasangan kumparan yang diberikan arus. Bila urutan pergantian awal searah dengan arah perputaran jarum jam maka urutan sebaliknya akan menghasilkan arah putaran yang berlawanan dengan arah putaran jarum jam.

**2.2.3 Penggerak motor *stepper***

Motor listrik merupakan suatu mesin listrik yang merubah besaran listrik menjadi besaran mekanik. Hal ini disebabkan adanya arus yang melewati konduktor yang ada pada stator motor listrik. Untuk menggerakkan motor listrik maka diperlukan arus yang cukup sebagai input dari lilitan motor. Keluaran suatu rangkaian digital pada keadaan tinggi adalah 400  $\mu$ A sedangkan pada keadaan rendah TTL mampu membenamkan arus sebesar 16 mA. Dari karakteristik TTL yang seperti itu maka untuk menggerakkan motor *stepper* dengan memanfaatkan keluaran TTL maka diperlukan driver motor yang berfungsi untuk memperkuat arus.

Penguat Darlington merupakan rangkaian yang terdiri dari dua buah pengikut emitor yang terhubung kaskade, seperti yang ditunjukkan pada Gambar 2.5.

Penguatan tegangan keseluruhan mendekati 1. Akibatnya adalah penambahan impedansi input yang sangat tinggi dan penurunan impedansi output.



**Gambar 2.5** Penguat Darlington  
 Sumber : M.Barmawi, 1996 :207

**Analisis DC**

Pada transistor pertama dan kedua, akan ada tegangan jatuh pada basis dan emitornya, masing-masing sebesar  $V_{BE}$ . Pembagi tegangan akan menghasilkan tegangan Thevenin yang digunakan untuk arus basis. Karena adanya tegangan jatuh sebesar dua  $V_{BE}$ , maka arus emitor pada tingkat kedua adalah :

$$V_{TH} = \frac{R1}{R1 + R2} V_{CC} \dots\dots\dots(2.3)$$

$$I_{E2} = \frac{V_{TH} - 2V_{BE}}{R_E} \dots\dots\dots(2.4)$$

Arus emitor DC pada tingkat pertama sama dengan arus basis DC pada tingkat kedua, karena itu :

$$I_{B2} = I_{E1} = \frac{I_{E2}}{\beta_{dc2}} \dots\dots\dots(2.5)$$

$$I_{B1} = \frac{I_{E2}}{\beta_{dc1}} \dots\dots\dots(2.6)$$

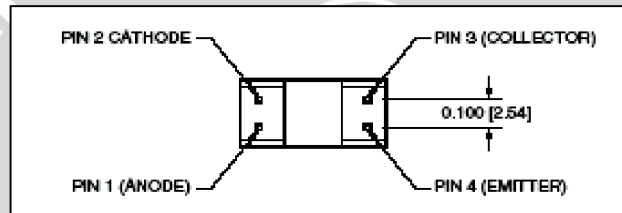


### 2.3. Optocoupler

*Optocoupler* adalah alat yang dipakai untuk mengkopel cahaya dari suatu sumber ke detektor tanpa adanya perantara. Oleh karena itu piranti ini sering disebut dengan *optoisolator/optocoupler*

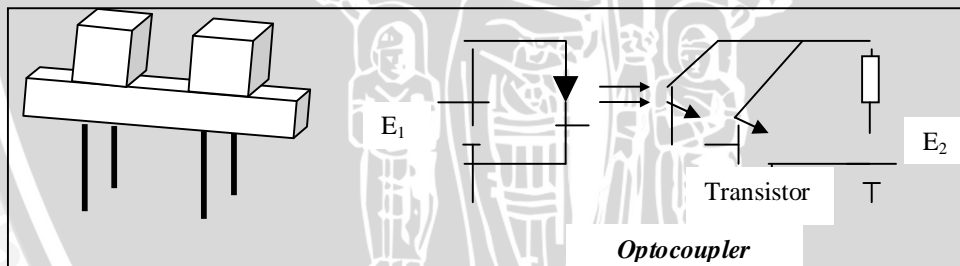
Sinyal listrik (arus) pada input diubah menjadi sinyal optik dengan menggunakan sumber cahaya (biasanya LED). Sinyal optik tersebut akan diterima oleh detektor untuk diubah kembali menjadi sinyal listrik (Malvino,1993: 136)

Umumnya *optocoupler* dipakai untuk mengisolasi sinyal listrik yang ada pada *input* dan *output* sehingga dapat digunakan transmisi sinyal antar rangkaian. Bentuk dan diagram rangkaian *optocoupler* dapat dilihat dalam Gambar 2.6 dan 2.7.



Gambar 2.6. *Optocoupler*

Sumber: Fairchild Semiconductor, 2001



Gambar 2.7. Diagram Rangkaian *Optocoupler*

Sumber : General Instrument, tanpa tahun

#### 2.3.1 Prinsip Kerja OptoCoupler

Optocoupler memiliki sebuah LED yang dihubungkan dengan masukan dan sebuah dioda yang dihubungkan dengan keluaran. Prinsip kerja dari optocoupler adalah:

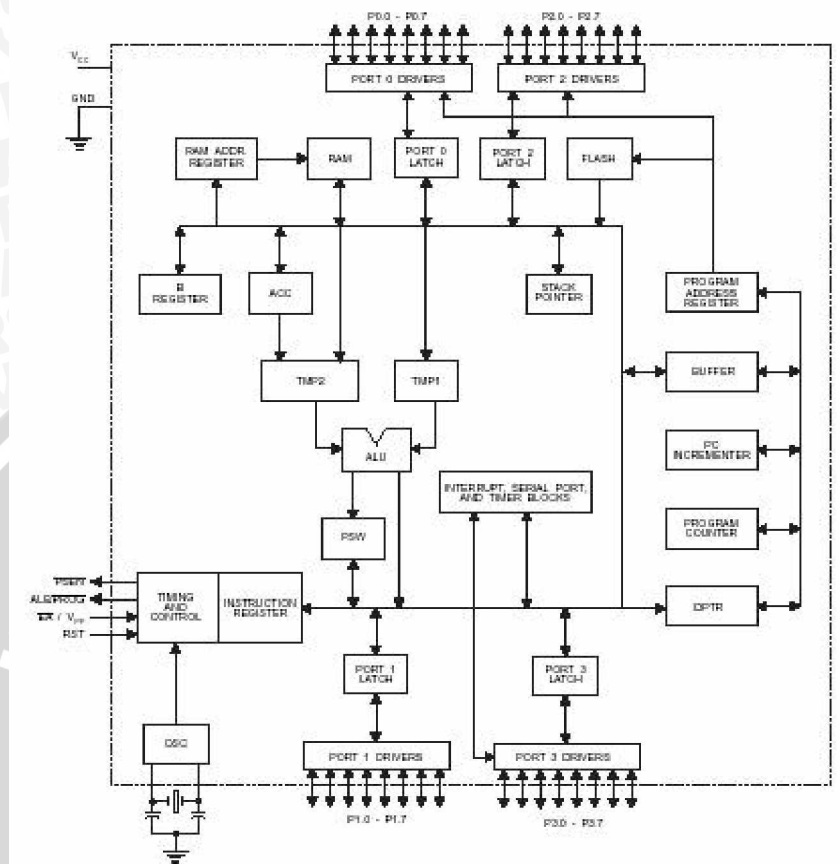
- Jika antara transistor dan LED dihalangi maka transistor akan off sehingga output dari kolektor akan berlogika high.
- Sebaliknya jika antara transistor dan LED tidak dihalangi maka transistor akan on sehingga outputnya akan berlogika low.

Dengan mengetahui prinsip kerja dari phototransistor ini maka harus dibuat penghalang antara transistor dan LED, pada penghalang itu diberi lubang sedikit. Penghalang itu harus dibuat seporos dengan roda tersebut. Sehingga ketika berputar output dari transistor akan mengalami high dan low. Penghalang yang biasanya digunakan adalah jeruji roda. Ketika roda berputar, maka ruji-ruji pada roda akan memotong cahaya yang dipancarkan oleh LED, sehingga akan mengalir arus yang melewati resistor output. Karena tegangan output sama dengan tegangan suplai dikurangi tegangan yang melewati resistor. Sehingga ketika masukan berubah, intensitas cahaya berubah, maka tegangan keluaran juga berubah.

#### 2.4 Mikrokontroler AT89C51

Mikrokontroler AT89C51 mempunyai karakteristik utama sebagai berikut :

- Ø CPU dengan lebar data 8 bit
- Ø Prosesor boolean untuk operasi logika 1 bit
- Ø Ruang memori program sebesar 64 Kbyte
- Ø Ruang memori data sebesar 64 Kbyte
- Ø EPROM sebesar 4 Kbyte untuk memori data pada *chip*
- Ø RAM sebesar 256 byte untuk memori data pada *chip*
- Ø 32 jalur *bidirectional* (dapat digunakan sebagai jalur masukan atau keluaran) dan setiap jalur bit dapat dialamati
- Ø tiga buah *counter/timer* 16 bit
- Ø UART (*Universal Asynchronous Receiver Transmitter*) *full duplex*
- Ø 5 jalur interupsi dengan 2 tingkat prioritas dapat diprogram
- Ø Osilator internal terdapat dalam *chip*



Gambar 2.8. Diagram Blok Mikrokontroler AT89C51

Sumber : Intel, 1994: 3-4

### 2.4.1 Struktur dan Operasi Port

Mikrokontroler MCS-51 memiliki 4 buah port. Setiap port memiliki 8 buah jalur I/O yang bersifat bidirectional.

Diagram blok mikrokontroler dalam Gambar 2.8 mewakili semua seri dalam keluarga mikrokontroler MCS-51. Perbedaan mendasar antara berbagai tipe mikrokontroler keluarga MCS-51 terletak pada kapasitas RAM dan ROM internal. Beberapa karakteristik *port* mikrokontroler MCS-51 dijelaskan secara singkat berikut ini :

- Port 0 merupakan *port* I/O 8 bit dua arah yang dapat berfungsi sebagai *input*, *output* atau keduanya. Port 0 berfungsi sebagai *input* dan *output* saat digunakan sebagai jalur alamat dan data (AD0 - AD7) yang terhubung dengan memori di luar mikrokontroler.
- Port 1 merupakan *port* I/O 8 bit dua arah yang dapat berfungsi sebagai *input* dan *output*. Fungsi *input* dan *output* dapat diprogram dengan mengontrol *latch Port 1*
- Port 2 merupakan *port* I/O dua arah yang dapat berfungsi sebagai *input* dan *output*. *Port* ini juga dapat digunakan untuk mengeluarkan 8 bit tinggi (A8 - A15) bersama-sama dengan *port 0* yang mengeluarkan alamat 8 bit rendah dalam berhubungan dengan memori eksternal.
- Port 3 merupakan *port* I/O dua arah seperti *port 1*. Fungsi *input* dan *output*nya dapat diprogram dengan mengontrol *latch P3* atau mengontrol beberapa *special function register (SFR)* tertentu. Pada *port* ini terdapat beberapa fungsi tambahan seperti yang terlihat dalam Tabel 2.4.

Tabel 2.4 Fungsi Tambahan *Port 3*

Jalur	Fungsi Alternatif	SFR
P3.0	<i>Input</i> data serial <i>RXD</i>	SBUF
P3.1	<i>Output</i> data serial <i>TXD</i>	SBUF
P3.2	<i>Input</i> Interupsi Eksternal <i>INT0</i>	TCON.1
P3.3	<i>Input</i> Interupsi Eksternal <i>INT1</i>	TCON.3
P3.4	<i>Input Counter T0</i>	TMOD
P3.5	<i>Input Counter T1</i>	TMOD
P3.6	<i>Output</i> Sinyal Kontrol <i>WR</i>	-
P3.7	<i>Output</i> Sinyal Kontrol <i>RD</i>	-

Sumber : Ayala , 1991 : 15

### 2.4.2 Sistem Interupsi

Mikrokontroler AT89C51 memiliki 5 buah sumber interupsi yang dapat dikendalikan oleh perangkat lunak, yaitu dengan mengatur kondisi logika bit-bit pada register IE (*Interrupt Enable*) (Ayala,1991:41).

Alamat yang melayani interupsi ditunjukkan dalam Tabel 2.5.

**Tabel 2.5** Alamat Interupsi

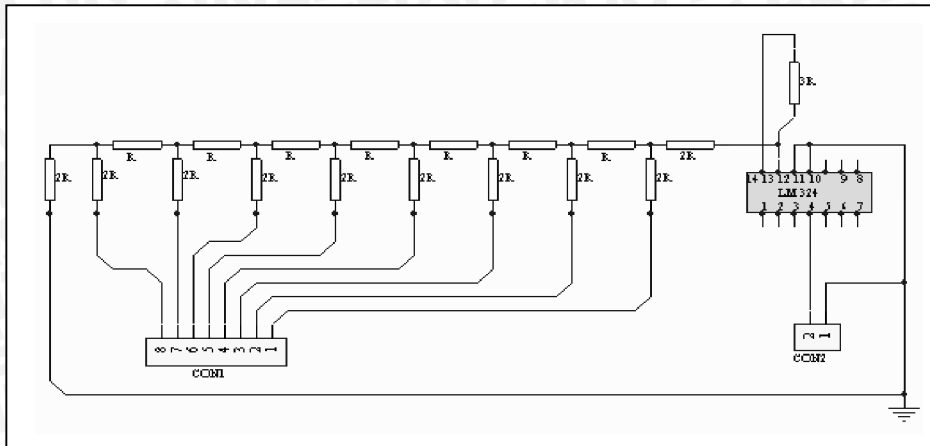
Interupsi	Alamat (Hex)
IE0	0003
TF0	000B
IE1	0013
TF1	001B
Serial	0023

Sumber : Ayala, 1991 : 41

### 2.5 DAC (Digital to Analog Converter)

Pengubah digital ke analog (digital to analog converter, DAC) menerima masukan digital paralel dan mengubahnya ke nilai tegangan atau arus yang disajikan masukan biner. Jika ini diulang untuk masukan digital yang berurutan akan terbentuk gelombang analog.

DAC R-2R Ladder adalah jenis DAC yang paling banyak digunakan. Jenis ini mengatasi persoalan ketelitian yang muncul pada resistor jenis terbebani (weighted-resistor) dengan mengurangi jangkauan nilai resistor sampai 2:1 (hanya dua nilai resistor yang digunakan) tidak bergantung pada resolusi yang digunakan. DAC R-2R Ladder 8 bit ditunjukkan pada Gambar 2.9



**Gambar 2.9.** DAC R-2R delapan bit.

Sumber : Malvino, 1991: 340

## 2.6 Dasar Teori Logika Fuzzy

Lotfi A. Zadeh pada tahun 1965 mengemukakan suatu himpunan yang disebut himpunan *fuzzy* (*fuzzy set*) untuk menerangkan suatu logika bertingkat. Logika ini kemudian dikenal dengan logika *fuzzy*. Logika ini menjadi dasar dari semua logika, dengan mengabaikan banyaknya tingkat kebenaran yang diasumsikan. Zadeh memilih kata “*fuzz*” untuk merepresentasikan suatu nilai logika kontinyu antara 0 (pasti salah) dan 1 (pasti benar). [Jamshidi, 1993: 12].

Ciri utama dari logika *fuzzy* adalah adanya peluang untuk memanfaatkan toleransi ketidakpastian (*inexactness*) dan ketidaktepatan (*imprecision*). Perbedaan antara logika *fuzzy* dengan logika klasik adalah pada manipulasi simbolik dan numerik. Logika *fuzzy* dalam hal ini dapat menangani baik manipulasi simbolik maupun manipulasi numerik. Sedangkan logika klasik hanya dapat menangani manipulasi simbolik.

Zadeh mengatakan bahwa logika *fuzzy* diarahkan untuk menyediakan sebuah sistem formal berorientasi komputasi yang berisi konsep dan teknik untuk menghubungkan mode penalaran aproksimasi (*approximate reasoning*) dengan penalaran eksak (*exact reasoning*). Logika *fuzzy* menjadi alat matematik yang membuat komputer mampu memodelkan sistem-sistem yang tidak terdefinisi secara

tepat. Logika ini menjadikan komputer dapat membuat keputusan dari data-data yang tersedia walaupun tingkat keakuratannya sangat rendah.

Sebuah sistem berbasis logika *fuzzy* dapat mengurangi ketidakakuratan yang terjadi pada sistem klasik. Himpunan klasik (*crisp set*) memiliki persyaratan keanggotaan yang membatasi nilai anggota-anggota himpunannya hanya pada satu atau nol saja. Batasan dari logika ini terdefinisi dengan tajam pada angka nol dan satu, sehingga tingkat keakuratannya menjadi rendah. Sedangkan pada himpunan *fuzzy*, sebuah himpunan unsur-unsur atau anggota himpunan dapat menjadi anggota himpunan dengan tingkat keanggotaan antara nol dan satu. Ada transisi yang bertahap dari unsur yang menjadi anggota atau bukan anggota himpunan dalam sebuah himpunan *fuzzy*.

### 2.6.1. Himpunan Fuzzy

Himpunan didefinisikan sebagai kelompok objek-objek tertentu yang disebut elemen.[Pebles, 1993: 3]. Himpunan *fuzzy* didefinisikan dalam semesta himpunan X, himpunan *fuzzy* ditentukan oleh fungsi keanggotaan yang memetakan anggota X ke daerah keanggotaan yang selalu berada dalam interval [0,1].[Yan,J, 1994: 15].

Jika X merupakan kumpulan objek,  $X=\{x\}$  dengan X disebut semesta pembicaraan dan x merupakan anggota dari X, sebuah himpunan fuzzy A pada X digambarkan sebagai sekumpulan pasangan yang diurutkan :

$$A = \{(x, \mu_A(x)) | x \in X\} \dots \dots \dots (2.7)$$

dengan  $\mu_A$  adalah fungsi keanggotaan yang memetakan X ke interval [0,1], yaitu  $\mu_A : X \rightarrow [0,1]$

Jika penyokong himpunan *fuzzy* (x) hanya sebuah titik tunggal, maka  $A = \mu_1 / x$ , disebut sebuah *fuzzy* tunggal (*singleton fuzzy*).

Himpunan *fuzzy* yang dinyatakan dengan gabungan *fuzzy* tunggal diekspresikan sebagai

$$A = \mu_1 / x_1 + \mu_2 / x_2 + \dots + \mu_n / x_n \dots \dots \dots (2.8)$$

$$A = \sum \mu_A(x_i) / x_i$$



Bila X bersifat kontinu dan tak terbatas :

$$A = \int_x \mu_A(x) / x \dots\dots\dots(2.9)$$

**2.6.2 Fungsi Keanggotaan Fuzzy**

Fungsi keanggotaan menotasikan nilai kebenaran anggota himpunan. Interval nilai yang digunakan untuk menentukan fungsi keanggotaan, yaitu nol dan satu. Tiap fungsi keanggotaan memetakan elemen himpunan crisp ke semesta himpunan fuzzy. [G. Klir dan Bo Yuan, 1995: 11].

Ada dua cara untuk menentukan fungsi keanggotaan himpunan fuzzy, yaitu secara numerik dan secara fungsional.[Yan,J,1994: 17].

- Pendefinisian secara numerik, mengekspresikan fungsi keanggotaan himpunan fuzzy sebagai suatu vektor angka yang tergantung pada tingkat diskritisasi.
- Pendefinisian secara fungsional, mendefinisikan fungsi keanggotaan himpunan fuzzy dalam ekspresi analitik yang mengikuti tingkatan tiap elemen dalam pendefinisian semesta himpunan yang dihitung.

Pada himpunan *fuzzy*, semua informasi terkandung dalam fungsi keanggotaannya. Oleh karena itu penting untuk mengetahui bagian-bagian pada sebuah fungsi keanggotaan *fuzzy*.

Ada beberapa fungsi keanggotaan dalam logika fuzzy. Adapun yang sering digunakan adalah bentuk fungsi-S, fungsi- $\pi$ , fungsi T (triangular). Contoh berikut ini adalah tiga macam keanggotaan yang dinyatakan dengan fungsi keanggotaan S,  $\pi$  dan T (triangular).

**a. S-function**

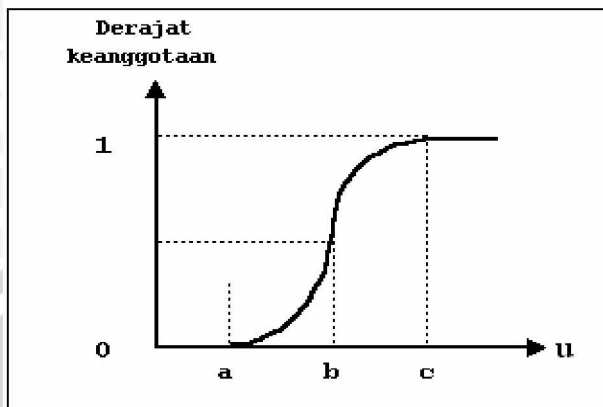
Definisi S-function adalah sebagai berikut :

$$S(u,a,b,c) = \left\{ \begin{array}{ll} 0 & u < a \\ 2 \left( \frac{u-a}{c-a} \right) & a \leq u \leq b \\ 1 - 2 \left( \frac{u-a}{c-a} \right) & b \leq u \leq c \\ 1 & u > c \end{array} \right\} \dots\dots\dots(2.10)$$





Bentuk diagramatik S-function ditunjukkan dalam Gambar 2.10.



Gambar 2.10. S-Function

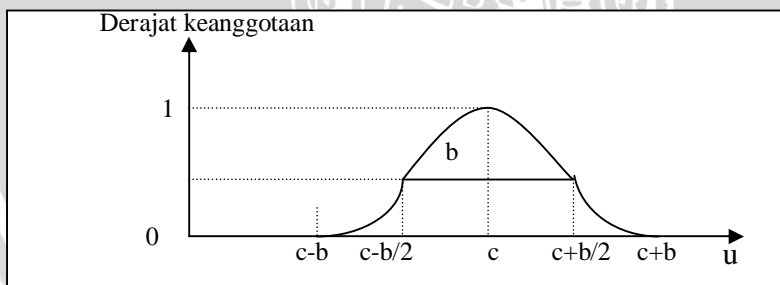
Sumber : Yan, J, 1994:29

**b.  $\pi$ -Function**

Definisi  $\pi$ -Function adalah sebagai berikut :

$$\pi(u; b, c) = \begin{cases} S(u; c - b, c - \frac{b}{2}, c) & u \leq c \\ 1 - S(u; c, c + \frac{b}{2}, c + b) & u \geq c \end{cases} \dots\dots\dots(2.11)$$

Bentuk diagramatik  $\pi$ -Function ditunjukkan dalam Gambar 2.11.



Gambar 2.11.  $\pi$ -Function

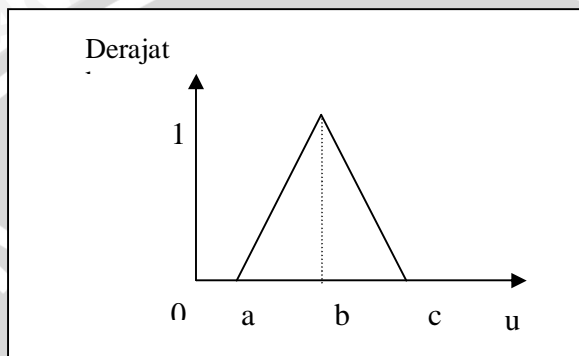
Sumber : Yan, J, 1994:29

**c. Triangular-function.**

Triangular-function didefinisikan sebagai berikut :

$$T(u; a, b, c) = \begin{cases} 0 & u < a \\ \frac{u-a}{b-a} & a \leq u \leq b \\ \frac{c-u}{c-b} & b \leq u \leq c \\ 0 & u > c \end{cases} \dots\dots\dots(2.12)$$

Bentuk diagramatik Triangular-function ditunjukkan dalam gambar 2.12.



**Gambar 2.12.** T-Function  
Sumber : Yan, J, 1994:30

**2.6.3 Operasi Himpunan Fuzzy**

Operasi dasar yang digunakan dalam himpunan *fuzzy* A dan B pada semesta himpunan X dengan fungsi keanggotaan  $\mu_A$  dan  $\mu_B$  terdiri dari gabungan, irisan, dan komplemen, operasi dasar yang berlaku didefinisikan sebagai berikut :

- a. Gabungan(*union*)

Gabungan himpunan *fuzzy* A dan B didefinisikan sebagai berikut :

$$\mu_{A \cup B} = \mu_A(x) \vee \mu_B(x) = maks\{\mu_A(x), \mu_B(x)\} \dots\dots\dots(2.13)$$

Operasi gabungan diperlihatkan dalam Gambar 2.11.

- b. Irisan(*intersection*)

Irisan dua himpunan *fuzzy* A dan B didefinisikan sebagai berikut:

$$\mu_{A \cap B} = \mu_A(x) \wedge \mu_B(x) = min\{\mu_A(x), \mu_B(x)\} \dots\dots\dots(2.14)$$

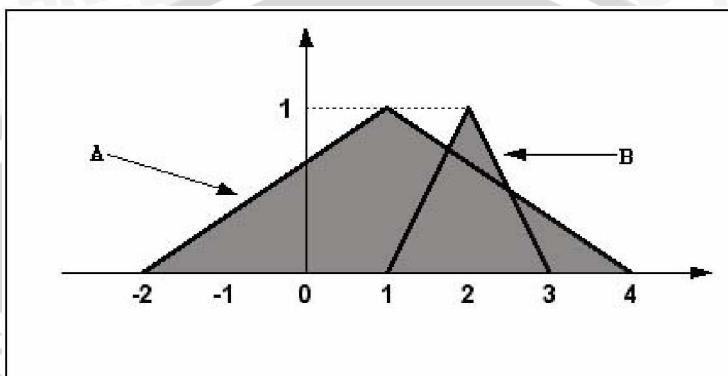
Operasi irisan diperlihatkan dalam Gambar 2.12.

- c. Komplemen (*complement*)

Komplemen himpunan fuzzy A dinotasikan sebagai  $\bar{A}$ . Operasi komplemen didefinisikan sebagai :

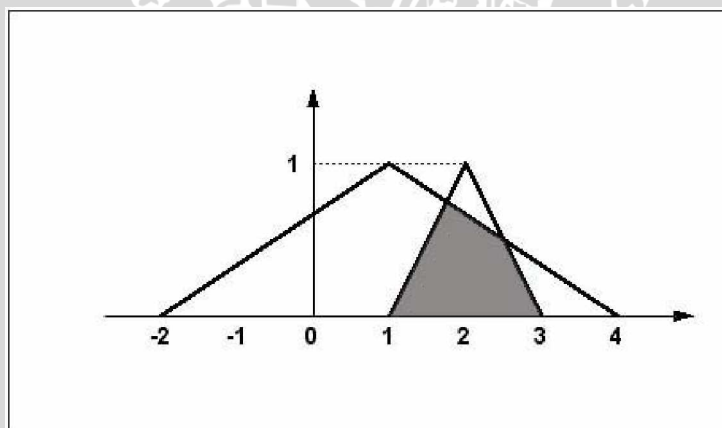
$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \dots \dots \dots (2.15)$$

Operasi komplemen himpunan fuzzy A diperlihatkan dalam Gambar 2.13.



**Gambar 2.13.** Contoh operasi gabungan himpunan fuzzy A dan B

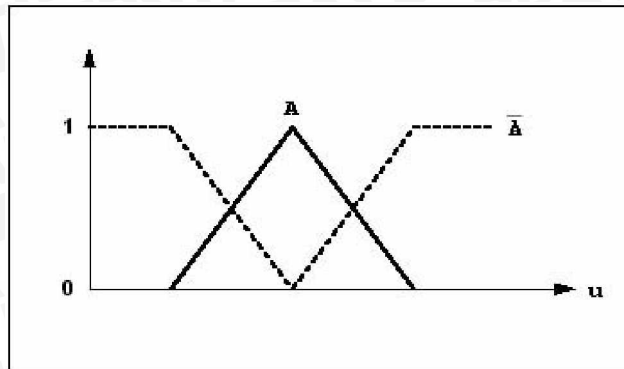
Sumber : Klir, G dan Yuan, B, 1995 : 111



**Gambar 2.14.** Contoh operasi irisan himpunan fuzzy A dan B

Sumber : Klir, G dan Yuan, B, 1995 : 111





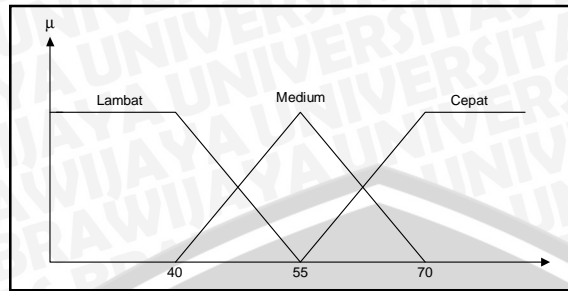
Gambar 2.15. Contoh operasi komplemen himpunan fuzzy  $A$

Sumber : Jamshidi, dkk., 1993 : 19

#### 2.6.4 Variabel Linguistik (*Linguistic Variable*)

Pendekatan dengan himpunan fuzzy untuk mempresentasikan cara berpikir manusia memerlukan variabel linguistik sebagai pengganti dari variabel yang biasa digunakan dalam pendekatan kuantitatif. Variabel linguistik ini berupa ungkapan-ungkapan yang bersifat kualitatif dan berfungsi untuk menyatakan himpunan fuzzy. Dalam berpikir manusia menggunakan ungkapan yang bersifat kualitatif terhadap informasi-informasi yang diinderanya.

Himpunan fuzzy memiliki nilai kebenaran yang dinyatakan dalam bentuk informasi linguistik, seperti sangat lambat, lambat, sedang, agak cepat, cepat dan sebagainya sebagai variabel, yang disebut variabel fuzzy. Variabel linguistik dapat dipandang sebagai variabel yang nilainya merupakan bilangan fuzzy atau sebagai variabel yang nilai-nilainya didefinisikan dalam bentuk linguistik. Sebagai contoh, lambat didefinisikan sebagai kecepatan dibawah 40 m/jam, medium sebagai kecepatan 55 m/jam dan cepat sebagai kecepatan diatas 70 m/jam, maka fungsi keanggotaan dengan kecepatan sebagai variabel linguistik dengan tiga variabel fuzzy: lambat, medium dan cepat yang ditunjukkan dalam Gambar 2.16.



Gambar 2.16. Diagram representasi dari fuzzy kecepatan

Sumber : Lee, 1990:406

### 2.6.5 Aturan Fuzzy “If-Then”

Kaidah atur yang digunakan dalam logika *fuzzy* berupa kaidah atur “If-Then”. Aturan “If” menunjukkan situasi sedangkan aturan “Then” menunjukkan aksi. Aturan “If” disebut antiseden sedangkan “Then” disebut konsekuen. Antiseden ini menunjukkan daerah masukan yang dikarakteristikan dalam semesta himpunan variabel masukan. Konsekuen mendefinisikan aksi kontrol yang diinginkan. [Smith dan Corner, 1991: 19]. Bentuk umum dari aturan logika *fuzzy* ini sebagai berikut:

$$\text{If (kondisi), Then (aksi) .....(2.16)}$$

Untuk menghubungkan suatu pernyataan dengan pernyataan lain digunakan operator AND, OR, dan ALSO. Operator AND dan OR digunakan pada bagian antiseden, sedangkan ALSO digunakan pada bagian konsekuen. Operator AND menyatakan operasi irisan, OR menyatakan operasi gabungan, dan ALSO digunakan untuk mengindikasikan adanya keluaran majemuk dalam aturan *fuzzy*. (Jun Yuan, 1994: 31)

Penggunaan operator ini dicontohkan sebagai berikut :

If  $x_1$  adalah  $A_{k1}$  OR  $x_2$  adalah  $A_{k2}$  AND  $x_3$  adalah  $A_{k3}$  Then  $y_1$  adalah  $B_{k1}$

ALSO  $y_2$  adalah  $B_{k2}$

secara sederhana ditulis

If  $A_{k1}$  OR  $A_{k2}$  AND  $A_{k3}$  Then  $B_{k1}$  ALSO  $B_{k2}$

Untuk sistem MISO (*multi input single output*) dengan  $n$  masukan, aturan *fuzzy* ke- $k$  ditunjukkan sebagai berikut :

$$\text{If } A_{k1} \text{ AND } \dots \text{ AND } A_{ki} \text{ AND } \dots \text{ AND } A_{kn} \text{ Then } B_k \dots \dots \dots (2.17)$$



dengan :  $i = 1, 2, \dots, n$

$A_{ki}$  = Himpunan *fuzzy* variabel masukan  $x_i$  dalam aturan *fuzzy* ke- $k$

$B_k$  = Himpunan *fuzzy* variabel keluaran  $y_j$  dalam aturan *fuzzy* ke- $k$

Metode-metode penalaran untuk memperoleh keputusan *fuzzy* ada dua, yaitu :

- a. Metode nilai point MAX-MIN
- b. Metode nilai point MAX-DOT

Diasumsikan terdapat dua basis kaidah atur untuk *fuzzy*, yaitu :

Kaidah atur 1 : *If*  $x$  adalah  $A_1$  AND  $y$  adalah  $B_1$  *Then*  $z$  adalah  $C_1$

Kaidah atur 2 : *If*  $x$  adalah  $A_2$  AND  $y$  adalah  $B_2$  *Then*  $z$  adalah  $C_2$

Jika kaidah atur ke- $i$  dinyatakan dengan  $\alpha_i$  maka dua kaidah atur di atas dapat dinyatakan kembali sebagai berikut :

$$\alpha_1 = \mu_{A_1}(x_0) \wedge \mu_{B_1}(y_0)$$

$$\alpha_2 = \mu_{A_2}(x_0) \wedge \mu_{B_2}(y_0) \dots\dots\dots(2.18)$$

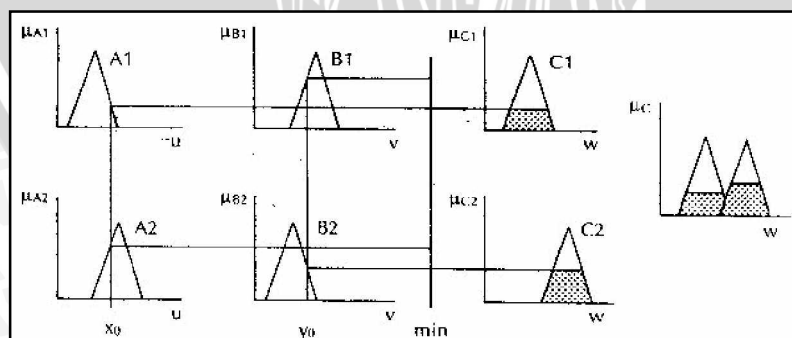
dengan  $x_0$  dan  $y_0$  menyatakan masukan *crisp*.

a. Metode penalaran MAX-MIN

Pada metode penalaran MAX-MIN fungsi keanggotaan konsekuen  $C$  dinyatakan dengan :

$$\mu_c(w) = (\alpha_1 \wedge \mu_{c_1}(w)) \vee (\alpha_2 \wedge \mu_{c_2}(w)) \dots\dots\dots(2.19)$$

metode penalaran MAX-MIN diperlihatkan dalam Gambar 2.17.



Gambar 2.17. Metode penalaran *fuzzy* MAX-MIN

Sumber : Yan, J, 1994 : 55

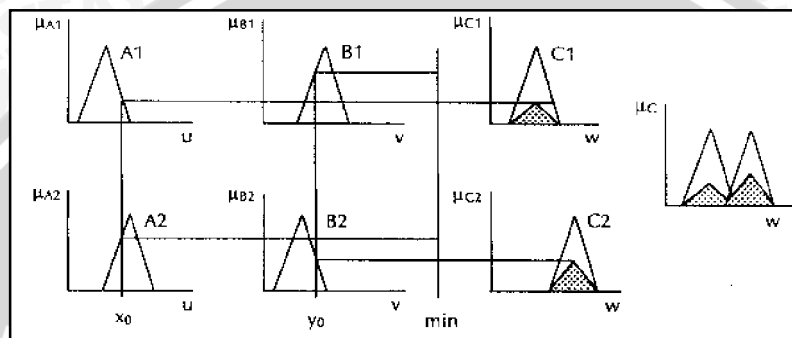


b. Metode penalaran MAX-DOT

Pada metode penalaran MAX-DOT fungsi keanggotaan konsekuen C dinyatakan dengan :

$$\mu_c(w) = (\alpha_1 \cdot \mu_{c1}(w)) \vee (\alpha_2 \cdot \mu_{c2}(w)) \dots\dots\dots(2.20)$$

metode penalaran MAX-DOT diperlihatkan dalam Gambar 2.16.



Gambar 2.16. Metode penalaran fuzzy MAX-DOT

Sumber : Yan, J, 1994 : 55

2.6.6 Kontrol Logika Fuzzy

Kontrol 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*.

Pengontrol 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, proses ini disebut *defuzzifikasi*.

2.6.6.1 Fuzzifikasi

*Fuzzifikasi* merupakan proses pemetaan dari himpunan *crisp* ke himpunan fuzzy. *Fuzzifikasi* berfungsi untuk mengubah hasil pengukuran ke dalam pernyataan linguistik. Operasi *fuzzifikasi* melibatkan transformasi sebuah himpunan nonfuzzy ke



dalam himpunan *fuzzy*. Proses *fuzzifikasi* dilakukan dengan menentukan fungsi keanggotaan.

Proses *fuzzifikasi* diekspresikan sebagai berikut :

$$x = \text{fuzzifier}(x_0) \dots\dots\dots(2.21)$$

- dengan :  $x_0$  = vektor nilai *crisp* variabel masukan
- $x$  = vektor himpunan *fuzzy* variabel yang terdefinisi
- $\text{fuzzifier}$  = 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, pemilihan sebaiknya 5 atau 7.
3. Mengatur himpunan *fuzzy* agar saling menumpuk.
4. Menggunakan fungsi keanggotaan bentuk segitiga atau trapesioda.

**2.6.6.2 Defuzzifikasi**

*Defuzzifikasi* merupakan proses pemetaan himpunan *fuzzy* ke himpunan *crisp*. Proses ini merupakan kebalikan dari proses *fuzzifikasi* berfungsi untuk mengubah keluaran linguistik ke dalam keluaran *crisp*.

Strategi *defuzzifikasi* diekspresikan sebagai berikut:

$$y_0 = \text{defuzzifier}(y) \dots\dots\dots(2.22)$$

- dengan :  $y$  = aksi kontrol *fuzzy*
- $y_0$  = aksi kontrol *crisp*
- $\text{defuzzifier}$  = operator *defuzzifikasi*

Metode *defuzzifikasi* ada dua, yaitu : Metode rata-rata maksimum dan metode pusat luasan. (Yan, J, dkk.,56).

- Metode Rata-Rata Maksimum (*Mean of Maximum*) MOM



Metode MOM disebut pula sebagai *defuzzifikasi* tinggi, aksi kontrol mewakili aksi kontrol yang fungsi keanggotaan mencapai maksimum. Nilai *crisp* yang dihasilkan oleh operasi MOM adalah :

$$W = \frac{\sum_{i=1}^n \alpha_i H_i W_i}{\sum_{i=1}^n \alpha_i H_i} \dots\dots\dots(2.23)$$

- dengan : n = jumlah aturan
- H<sub>i</sub> = tinggi maksimum fungsi keanggotaan himpunan *fuzzy*
- W<sub>i</sub> = kontrol keluaran dengan nilai *crisp*
- α<sub>i</sub> = *fire strength* aturan ke-i

- Metode Pusat Luasan (*Center of Area*) COA

Metode COA menghasilkan aksi kontrol pada pusat berat distribusi.

$$W = \frac{\sum_{i=1}^n \alpha_i M_i}{\sum_{i=1}^n \alpha_i A_i} \dots\dots\dots(2.24)$$

- dengan : n = jumlah aturan
- A<sub>i</sub> = luas himpunan *fuzzy*
- M<sub>i</sub> = momen fungsi keanggotaan himpunan *fuzzy* aturan keluaran kontrol ke-i
- α<sub>i</sub> = *fire strength* aturan ke-i

### 2.6.7 Struktur Kontroler Logika Fuzzy Dasar

Kontroler logika *fuzzy* menggunakan himpunan aturan kontrol dan mekanisme pengambil keputusan (*inference*) untuk menentukan aksi kontrol pada plant. Aturan kontrol merupakan ekspresi *fuzzy* yang merelasikan logika *fuzzy* masukan kontroler ke nilai keluaran kontroler. Mekanisme pengambil keputusan mengevaluasi kaidah atur kemudian menentukan aksi kontrol yang sesuai.

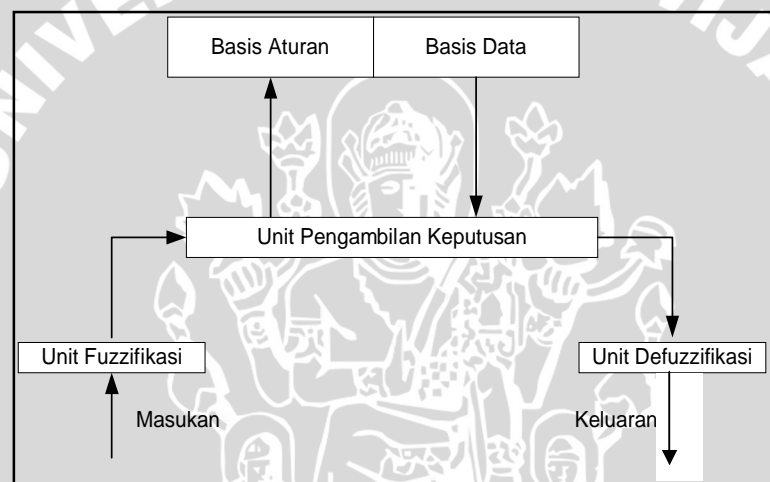
Kontroler logika *fuzzy* merupakan kontrol dengan lup tertutup. Elemen utama kontroler logika *fuzzy* terdiri dari unit *fuzzifikasi*, unit mekanisme pengambil keputusan, unit *defuzzifikasi*, dan unit basis pengetahuan. Basis pengetahuan terbagi dua yaitu basis data untuk menentukan fungsi keanggotaan himpunan *fuzzy* dan basis aturan untuk memetakan nilai *fuzzy* dari masukan ke nilai keluaran. Variabel sistem



yang digunakan ada dua yaitu variabel masukan (E dan CE) dan variabel keluaran(U). Struktur kontroler logika *fuzzy* dasar diperlihatkan dalam Gambar 2.17.

Jun Yan menentukan variabel masukan kontroler *fuzzy* akan diseleksi sebagai kesalahan (*error*, E) dan perubahan kesalahan (*change of error*, CE), keluaran merupakan variasi perintah yang disusun sebagai berikut:

- § Jika keluaran sama dengan nilai yang diinginkan dan perubahan kesalahan sama dengan nol, maka keluaran yang sebelumnya dipertahankan.
- § Jika keluaran tidak sesuai dengan yang diinginkan maka aksi kontrol tergantung pada nilai *error* dan perubahan *error*.



**Gambar 2.17.** Struktur kontroler logika *fuzzy* dasar

Sumber : Yan, J, 1994 : 47

Kaidah atur ditentukan dengan menggunakan kaidah atur pendekatan tabel.

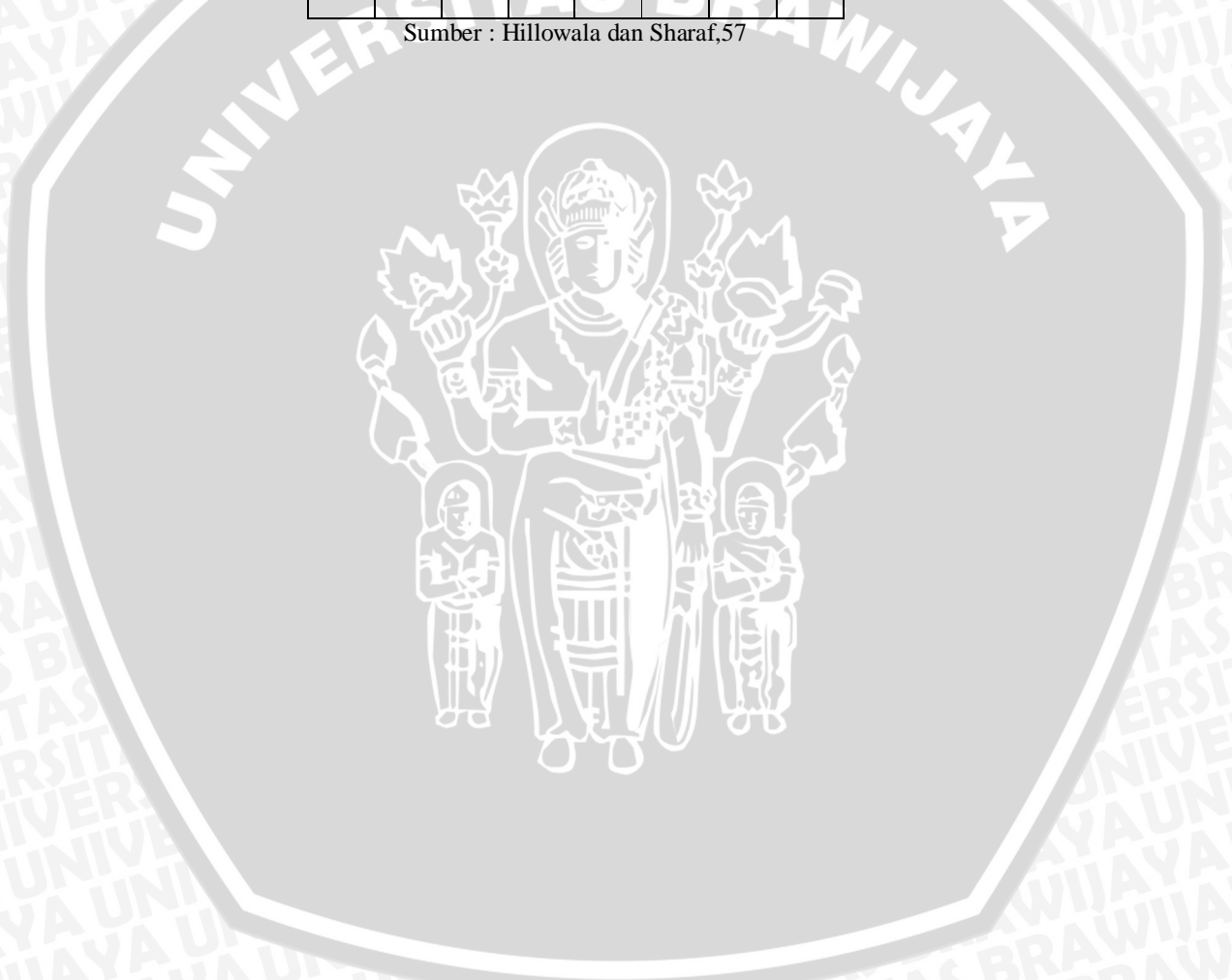
Suatu pengontrolan pada dasarnya bertujuan untuk mengontrol plant agar keluaran sesuai dengan nilai referensi atau titik acuan sehingga *error* antara referensi dan nilai keluaran plant sama dengan nol.

Jika variabel E dan CE dibagi menjadi tujuh varibel, yaitu Negatif Besar (NB), Negatif Sedang (NS), Negatif Kecil (NK), Sekitar Nol (SN), Positif Kecil (PK), Positif Sedang (PS) dan Positif Besar (PB), maka kaidah atur secara lengkap diperlihatkan pada Tabel 2.5.

**Tabel 2.5.** Kaidah atur dalam bentuk matrik

	NB	NS	NK	SN	PK	PS	PB
PB	SN	PK	PS	PB	PB	PB	PB
PS	NK	SN	PK	PS	PB	PB	PB
PK	NS	NK	SN	PK	PS	PB	PB
SN	NB	BS	NK	SN	PK	PS	PB
NK	NB	NB	NS	NK	SN	PK	PS
NS	NB	NB	NB	NS	NK	SN	PK
NB	NB	NB	NB	NB	NS	NK	SN

Sumber : Hillowala dan Sharaf,57



## BAB III

### METODE PENELITIAN

Penyusunan skripsi ini didasarkan pada masalah yang bersifat aplikatif yang diwujudkan dalam bentuk prototipe, yaitu perencanaan dan perealisasiian alat agar dapat menampilkan unjuk kerja sesuai dengan yang direncanakan dengan mengacu pada rumusan masalah. Data dan spesifikasi komponen yang digunakan dalam perencanaan merupakan data sekunder yang diambil dari buku data komponen elektronika. Pemilihan komponen berdasarkan perencanaan dan disesuaikan dengan komponen yang ada di pasaran.

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

- Penentuan spesifikasi alat
- Perealisasiian alat
- Pengujian alat
- Analisa data dan pengambilan kesimpulan

#### 3.1 Penentuan Spesifikasi Alat

Sebelum melakukan perencanaan dan perealisasiian alat, maka ditentukan spesifikasi alat yang akan dibuat. Adapun spesifikasi alat yang akan direalisasiikan sebagai berikut:

- Motor DC yang digunakan dalam perencanaan ini adalah Motor DC yang menggunakan permanen magnet. Alasan pemilihan motor DC tipe ini adalah kemudahan dalam pengontrolan dengan menggunakan pengaturan tegangan DC, dalam skripsi ini motor DC diatur dengan menggunakan tegangan yang konstan. Motor ini mempunyai fasilitas optical encoder yang menjadi satu dengan body motor dan ikut berputar pada saat motor berputar. Encoder ini berfungsi sebagai feedback untuk pengontrolan close loop.
- Prototipe mekanik alat bubut dengan ukuran 45 X 30 X 35 cm.
- *Fuzzy Logic Controller* menggunakan mikrokontroler AT89C51.



- Benda kerja yang dibubut memiliki bentuk silinder terbuat dari campuran bahan *parafin* dan *stearic acid* yang berdiameter 4,5 cm dan panjang 9,5 cm.
- Benda kerja yang dibubut adalah benda kerja yang memiliki bobot 0,2 Kg.
- Sensor limit bahan menggunakan *optocoupler* yang dipasang di samping poros statis pemutar benda kerja.

### 3.2 Perealisasian Alat

#### 3.2.1 Perancangan Perangkat Keras dan Perealisasian Tiap Blok

- Pembuatan blok diagram lengkap sistem.
- Penentuan dan perhitungan komponen yang akan digunakan
- Merakit perangkat keras masing masing blok.

#### 3.2.2 Perancangan dan Perealisasian Perangkat Lunak

Setelah kita mengetahui seperti apa perangkat keras yang kita rancang untuk aplikasi kontrol ini maka kita membutuhkan perangkat lunak untuk mengendalikan dan mengatur kerja dari alat ini. Program yang dibuat dalam bentuk bahasa assembler yang disimpan dalam *EEPROM* mikrokontroller. Dalam pembuatan program ini digunakan Software yang berisi program bahasa assembly untuk mengoperasikan peralatan, yang didalamnya juga terdapat rutin *fuzzy inference system*. Software yang digunakan untuk membuat sketsa benda kerja yang akan dibubut berupa program Delphi yang diinstal di dalam komputer sebagai piranti perangkat lunak tambahan.

### 3.3 Pengujian Rancangan

Untuk memastikan bahwa sistem ini berjalan sesuai yang direncanakan maka perlu dilakukan pengujian alat meliputi perangkat keras (*hardware*) maupun perangkat lunak (*software*) yang dilakukan baik secara per blok rangkaian maupun keseluruhan sistem.

### 3.3.1 Pengujian Perangkat Keras

Pengujian perangkat keras dilakukan dengan tujuan untuk menyesuaikan nilai tegangan dan arus pada alat dengan cara pengukuran yang dibandingkan dengan nilai tegangan dan arus yang diijinkan bekerja dalam komponen alat berdasarkan data sekunder komponen.

### 3.3.2 Pengujian Perangkat Lunak

Pengujian perangkat lunak dilakukan dengan melakukan pengujian pada subrutin-subrutin program dengan memberi masukan dan mengamati keluarannya, maupun pada rangkaian yang telah dirakit secara lengkap.

### 3.3.3 Pengujian Keseluruhan Sistem

Tujuan pengujian sistem secara keseluruhan adalah untuk mengetahui unjuk kerja alat setelah perangkat keras dan perangkat lunak diintegrasikan bersama.

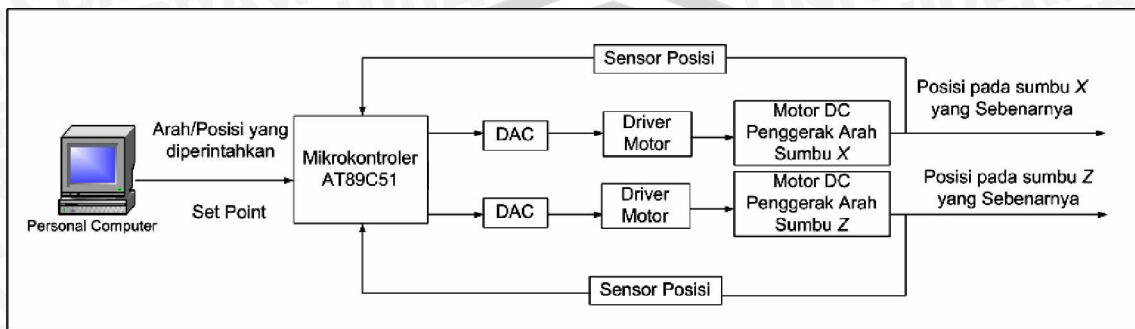
### 3.4 Analisa Data dan Pengambilan Kesimpulan

Kesimpulan didapat berdasarkan dari hasil perealisasiian dan pengujian pengendalian arah pergerakan mata pahat prototipe alat bubut dengan sesuai koordinat  $x$  dan  $z$  dan kesesuaian bentuk benda kerja yang di sketsa dengan benda kerja yang dibubut.

## BAB IV PERANCANGAN DAN PEMBUATAN ALAT

### 4.1 Perancangan Perangkat Keras

Blok diagram perancangan perangkat keras ditunjukkan dalam Gambar 4.1.



**Gambar 4.1.** Blok Diagram Perancangan Perangkat Keras

Sumber: Perancangan

#### 4.1.1 Cara Kerja Alat

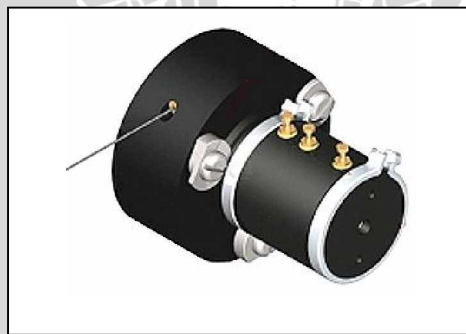
Berdasarkan blok diagram perancangan perangkat keras diatas maka cara kerja Prototipe alat Bubut adalah sebagai berikut :

- Dalam rancangan ini dipakai 2 acuan arah pergerakan yaitu koordinat  $x$  dan  $z$ .
- Dalam rancangan ini diasumsikan bahwa arah pergerakan  $x$  ke arah kanan titik *offset zero* (0,0) adalah bernilai positif (+) sedangkan arah pergerakan  $x$  ke arah kiri dari titik *offset zero* (0,0) adalah bernilai negatif (-)
- Dalam rancangan ini diasumsikan bahwa arah pergerakan  $z$  mendekati benda kerja adalah bernilai positif (+) sedangkan arah pergerakan  $z$  menjauhi benda kerja adalah bernilai negatif (-)
- Dalam rancangan ini arah pergerakan mata pahat pada alat bubut pada sumbu  $x$  dan  $z$  menggunakan acuan koordinat (X0,Z5), (X10,Z10), (X15,Z15),(X25,Z20)
- Benda kerja yang dipakai adalah plastisin berdiameter 50 mm
- Setelah memilih acuan posisi dan diameter benda kerja yang akan dipakai, Mikrokontroler akan menghitung titik koordinat  $x$  dan  $z$  dari transduser posisi.
- Mikrokontroler mulai menghitung besarnya Error dan  $\Delta$  Error dimana Error dan  $\Delta$  Error ini akan menjadi *crisp input* bagi *fuzzy logic controller*.
- Dilakukan Proses *Fuzzifikasi* sehingga dihasilkan derajat keanggotaan masing-masing nilai masukan.

- Proses selanjutnya adalah *rule evaluation*, dimana nilai masukan yang telah mempunyai derajat keanggotaan dimasukkan dalam kaidah atur yang direncanakan.
- Pada proses *rule evaluation* metode penalaran kontroler logika *fuzzy* yang digunakan adalah metode MIN-MAX.
- Setelah itu dilakukan proses *defuzzifikasi*, yaitu dengan menggunakan metode COG (*Center of Gravity*).
- Nilai yang dihasilkan dari proses *defuzzifikasi* dimasukkan dalam DAC.

#### 4.1.2 Sensor Posisi

Sensor posisi ini menggunakan transduser posisi seri 180-0803 yang berfungsi mengirimkan data perpindahan secara angular dalam bentuk tegangan. Transduser jenis putaran ini membuat pulsa listrik dan mengirimkannya kembali ke mikrokontroler AT89C51 dengan menggunakan rangkaian lilitan elektrik dalam *photo disk encoder*. Secara fisik, transduser ini dipasang pada salah satu roda gigi yang presisi. Didalam paket yang relatif kecil ini, terdapat satu seri lilitan listrik. salah satu lilitan ini disebut *rotor* yang berputar bersama dengan roda gigi yang presisi. Di sekitar sisi roda gigi yang presisi itu ada satu seri lilitan lagi yang saling berhubungan tetapi tidak ikut berputar yang disebut *stator*.

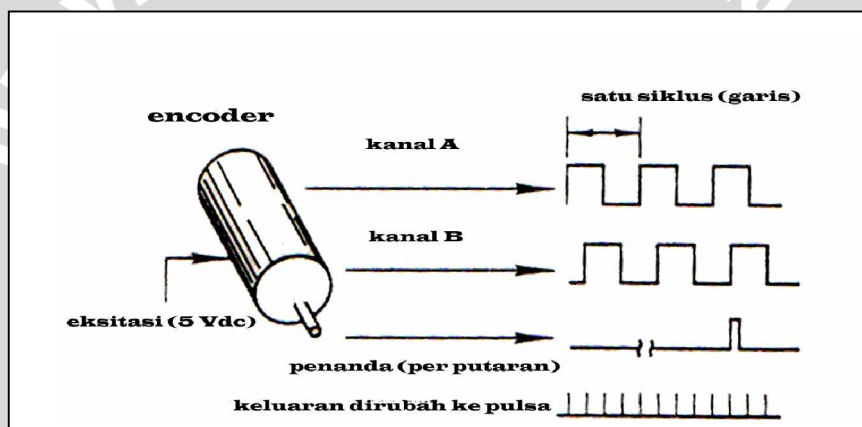


**Gambar 4.2.** Transduser Posisi seri 180-0803

Sumber : [www.spaceagecontrol.com](http://www.spaceagecontrol.com)

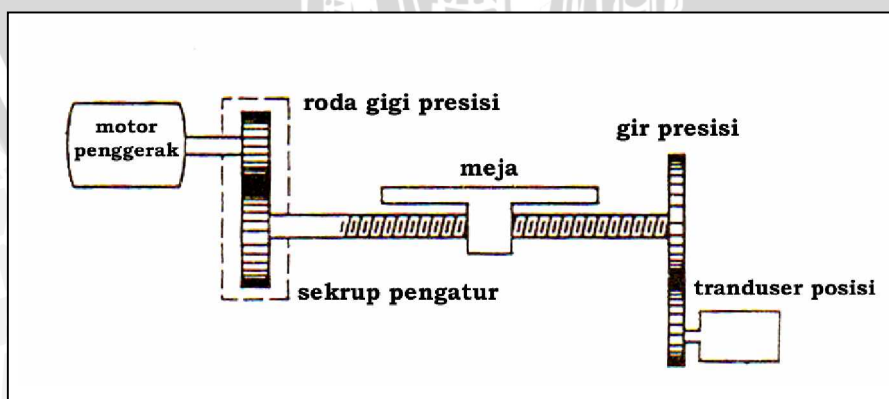


Lilitan stator atau fotosel diberi daya listrik pada tingkat tegangan yang telah ditetapkan oleh Mikrokontroler AT89C51. Dengan berputarnya roda gigi presisi oleh motor DC, tegangan diinduksikan ke fotosel *rotor*, dan tegangan ini bervariasi menurut posisi angular roda gigi presisi yang berhubungan dengan lilitan *stator* atau *photo disk encoder*. Informasi hasil tegangan induksi diberikan ke Mikrokontroler AT89C51, sehingga hitungan jumlah putaran roda gigi presisi didapatkan. Hitungan ini menginformasikan pergerakan atau posisi mata pahat yang digerakkan motor DC bila dibandingkan dengan instruksi atau posisi yang diperintahkan.



Gambar 4.3. Pulsa keluaran yang menunjukkan posisi poros rotor

Sumber : perancangan



Gambar 4.4. Tempat pemasangan Transduser Posisi pada alat

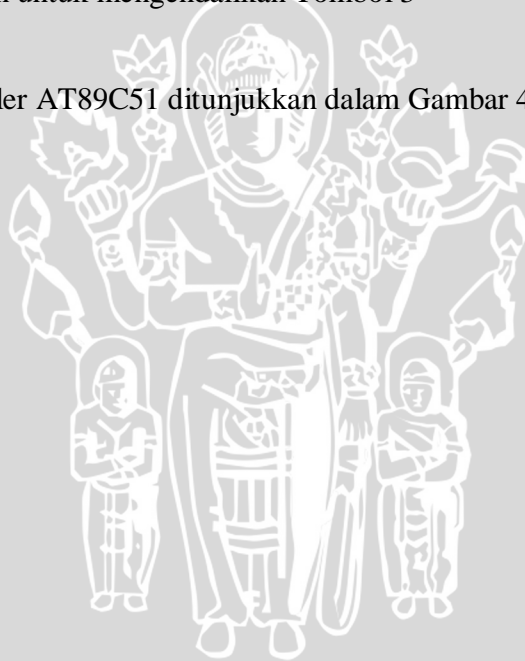
Sumber : perancangan

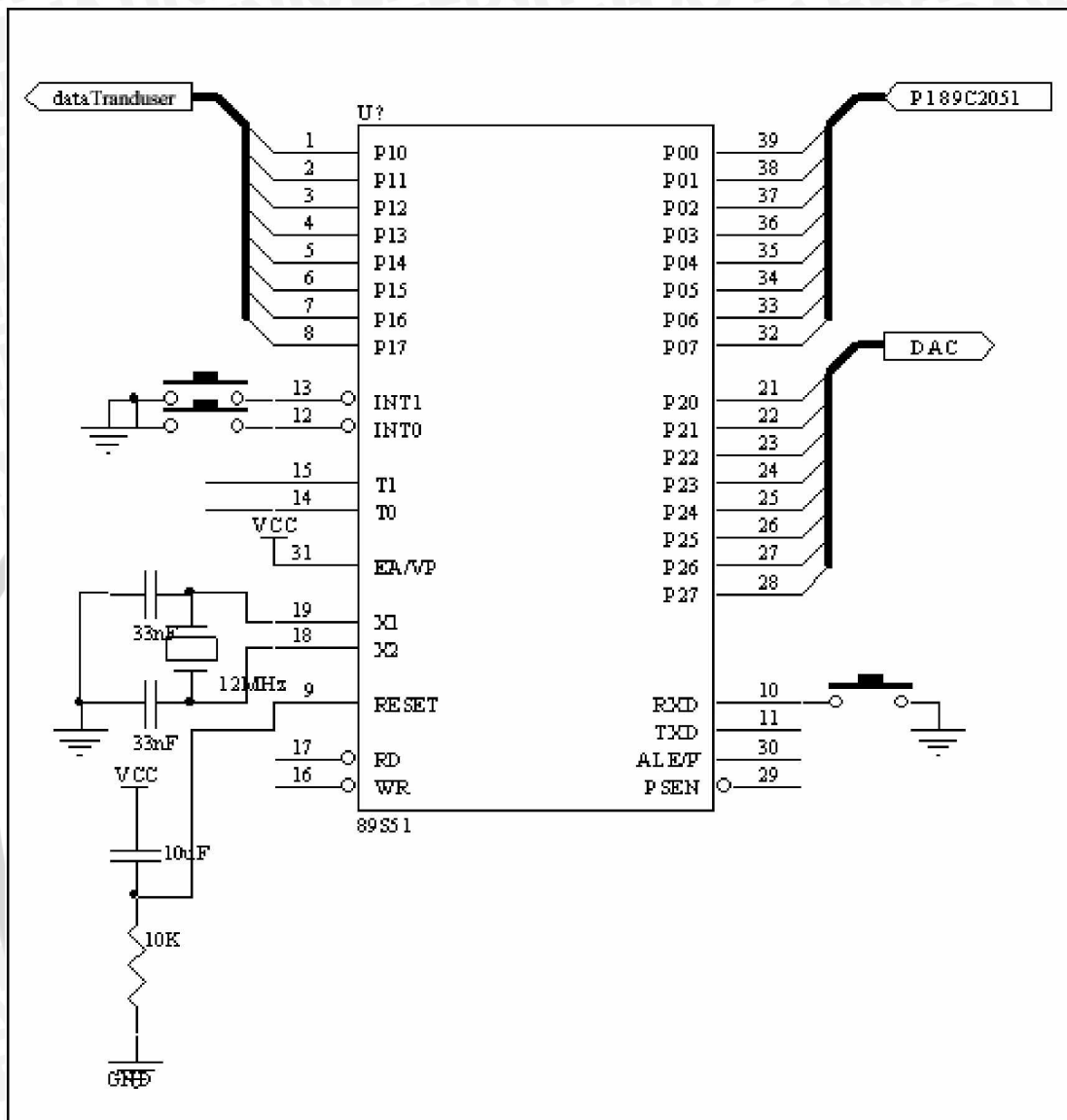
#### 4.1.3 Mikrokontroler AT89C51

Pin-pin pada keping mikrokontroler AT89C51 pada perancangan alat ini dialokasikan penggunaannya sebagai berikut :

- a. Port 0 ( pin 32 - 39 ) digunakan sebagai masukan dari AT89C2051.
- b. Port 1 ( pin 1 - 8 ) digunakan sebagai jalur data posisi dari sensor posisi
- c. Port 2 ( pin 21- 28 ) digunakan sebagai keluaran dan dihubungkan ke DAC.
- d. Pin RST (pin 9) digunakan untuk mereset mikrokontroler AT89C51.
- e. Pin 10 digunakan untuk mengendalikan Tombol 1
- f. Pin 12 digunakan untuk mengendalikan Tombol 2
- g. Pin 13 digunakan untuk mengendalikan Tombol 3

Rangkaian mikrokontroler AT89C51 ditunjukkan dalam Gambar 4.3





Gambar 4.5 Mikrokontroler AT89C51 dengan Rangkaian Pendukungnya  
Sumber: Perancangan

#### 4.1.4 Digital to Analog Converter DAC

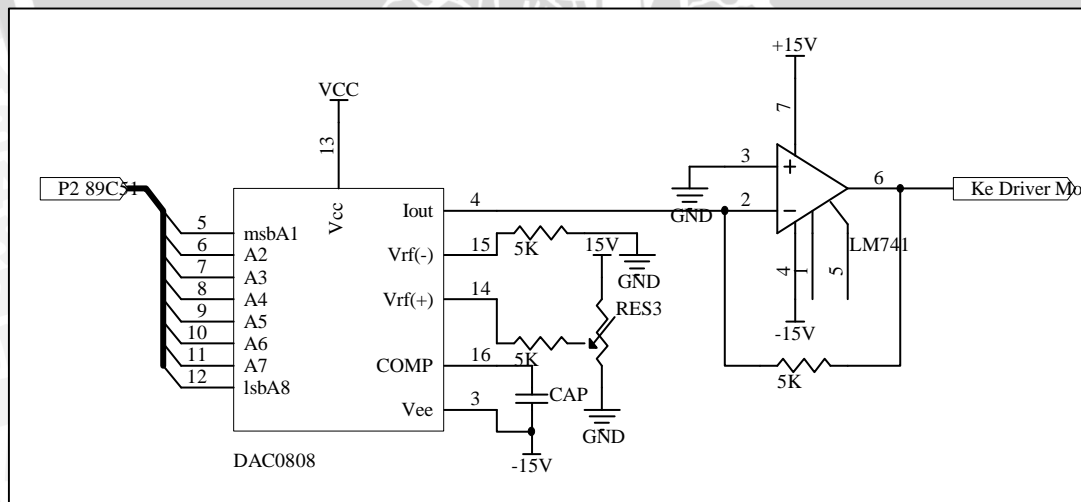
Digital to Analog Converter (DAC) 0808 adalah bagian utama yang mengkonversi sinyal digital yang berasal dari keluaran mikrokontroler menjadi sinyal analog yang berupa tegangan untuk menggerakkan motor DC.

Keluaran DAC dirancang untuk tegangan 0 volt sampai 6 volt. Tegangan  $V_{CC}$ , pada pin 13, sebesar +5 volt. Pin 14 yang merupakan tegangan referensi ditentukan sebesar +6 volt, sedangkan arus referensi berdasarkan datasheet ditentukan sebesar 2 mA sehingga  $R_{ref}$  dapat dihitung dengan persamaan:

$$R_{ref} = \frac{V_{ref}}{I_{ref}} \tag{4.3}$$

$$R_{ref} = \frac{6V}{2mA} = 6\ k\Omega$$

Pin 15 dihubungkan ke ground melalui sebuah hambatan yang bernilai 4,7 kΩ. Pin 2 dihubungkan langsung ke ground. Pin 3 ( $V_{EE}$ ) dihubungkan dengan tegangan sebesar -15 volt. dihubungkan dengan Kapasitor 0,1 uF yang dipasang antara pin 16 dan pin 3 berfungsi untuk memberi kompensasi frekuensi DAC 0808. Pin 4 merupakan keluaran dari DAC 0808 yang akan menjadi masukan rangkaian driver motor DC. Rangkaian DAC ditunjukkan pada Gambar 4.6

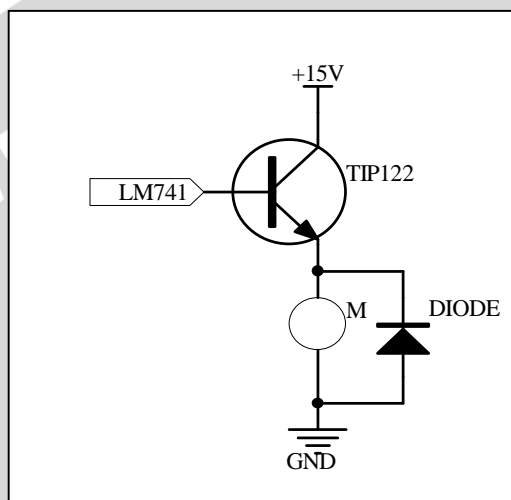


Gambar 4.6. Rangkaian DAC

Sumber: Perancangan

#### 4.1.5 Rangkaian Driver Motor DC

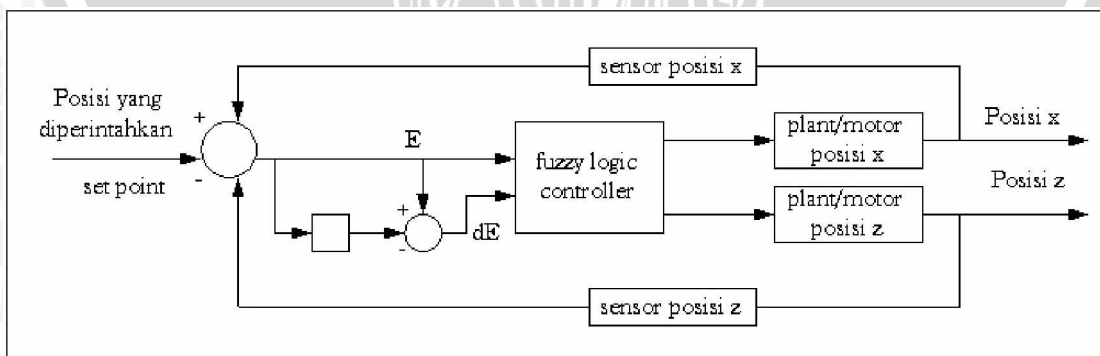
Rangkaian driver motor DC diperlukan untuk menggerakkan motor DC sesuai dengan perencanaan. Rangkaian driver motor DC ini terdiri dari op-amp, transistor jenis D400, dan transistor jenis TIP 41. Transistor ini diperlukan agar arus yang keluar dari op-amp yang kecil dapat dikuatkan untuk menggerakkan motor DC. Gambar 4.6 menunjukkan rangkaian driver motor DC.



Gambar 4.7. Rangkaian Driver Motor DC  
Sumber: Perancangan

#### 4.1.6 Perancangan Kontroler Logika Fuzzy

Blok diagram Kontroler Logika Fuzzy ditunjukkan dalam Gambar 4.8



Gambar 4.8. Blok Diagram Kontroler Logika Fuzzy  
Sumber: Perancangan

Tahapan-tahapan dalam kontrol logika fuzzy adalah sebagai berikut:

- a. Menentukan variabel masukan dan variabel keluaran .

Variabel masukan untuk kontroler logika fuzzy adalah Error dan  $\Delta$  Error, sedangkan variabel kelurannya adalah  $\Delta$ Output.

Dimana :

$$Err(n) = SP(n) - PV(n) \tag{4.4}$$

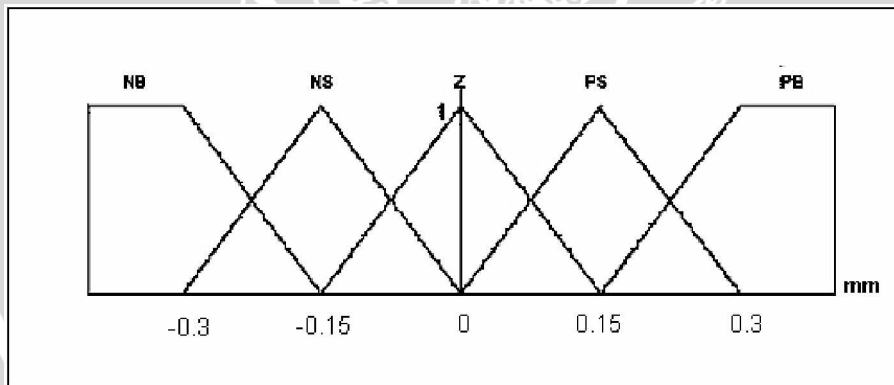
$$DErr(n) = Err(n) - Err(n-1) \tag{4.5}$$

$$Output(n) = Output(n-1) + Doutput \tag{4.6}$$

- b. menentukan jenis fungsi keanggotaan (*membership function*) masukan dan keluaran serta nilai label.

Fungsi keanggotaan untuk masukan berbentuk segitiga sedangkan untuk fungsi keanggotaan keluaran berbentuk *singleton*.

Fungsi keanggotaan untuk Error untul arah pergerakan atau posisi  $x$  terdiri dari 5 label yaitu NB (negative big), NS (negative small), ZE (zero), PS (positive small), dan PB (positive big).. Fungsi keanggotaan Error ditunjukkan dalam Gambar 4.9.

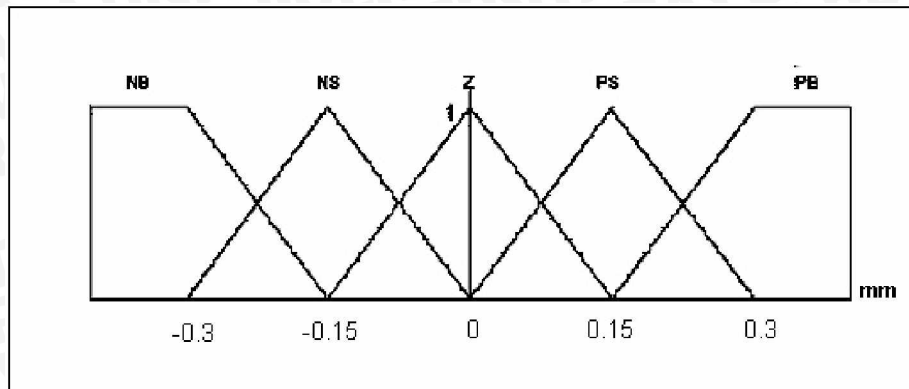


Gambar 4.9. Fungsi Keanggotaan Crisp Input Error

Sumber: Perancangan

Fungsi keanggotaan untuk  $\Delta$  Error mempunyai 5 label, yaitu NB (negative big), NS (negative small), ZE (zero), PS (positive small), dan PB (positive big). Fungsi keanggotaan  $\Delta$  Error ditunjukkan dalam Gambar 4.10.

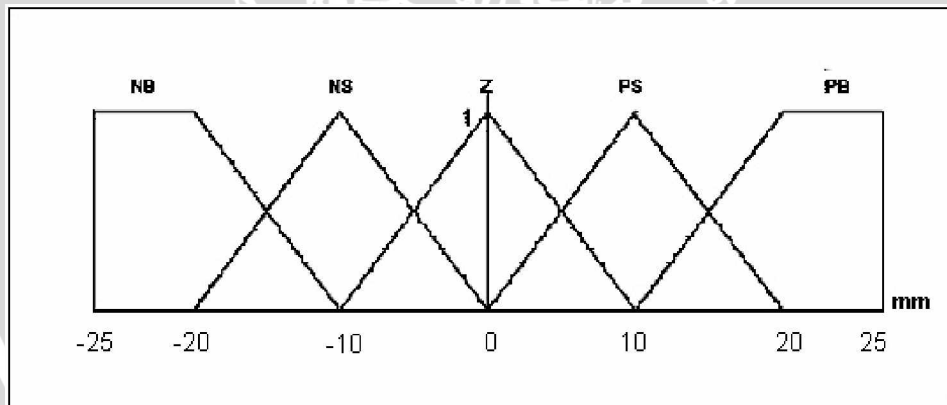




Gambar 4.10. Fungsi Keanggotaan *Crisp Input  $\Delta$  Error*

Sumber: Perancangan

fungsi keanggotaan untuk  $\Delta$ Output mempunyai 5 label yaitu posisi pahat terlalu mundur, posisi pahat sedikit mundur, posisi pahat relatif diam, posisi pahat sedikit maju, posisi pahat terlalu maju dari titik benda kerja. Fungsi keanggotaan  $\Delta$ Output ditunjukkan dalam Gambar 4.11.



Gambar 4.11. Fungsi Keanggotaan *Crisp Output*

Sumber: Perancangan

c. *Fuzzifikasi*

*fuzzifikasi* adalah proses mengubah nilai masukan (*crisp input*) menjadi input fuzzy. Pada masing-masing label dibagi menjadi empat bagian yaitu *Point 1*, *Slope 1*, *Point 2*, dan *Slope 2* seperti ditunjukkan dalam Gambar. Apabila *crisp*

*input* terletak pada segment 1 dari suatu label ( $Point 1 \leq Input < Point 2$ ), maka derajat keanggotaan *crisp input* pada label tersebut dinyatakan sebagai berikut:

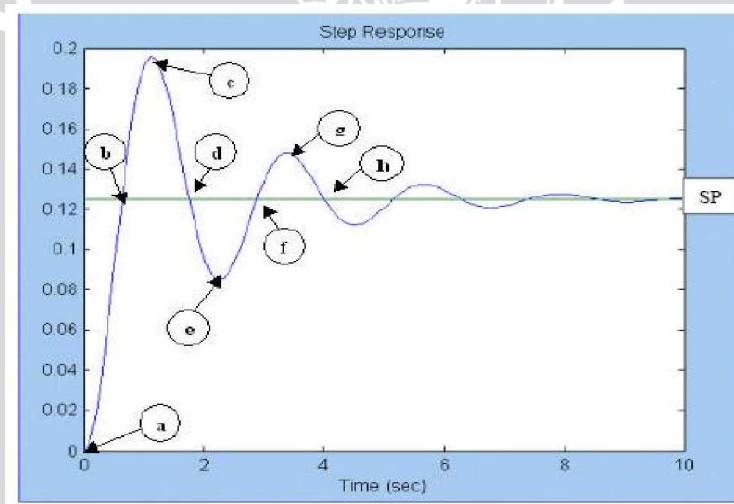
$$\mu = (Input - Point1) \times Slope1 \quad (4.7)$$

Sedangkan apabila *crisp input* terletak pada segment 2 dari suatu label ( $Input \geq Point 2$ ), maka derajat keanggotaan *crisp input* pada label tersebut dinyatakan sebagai berikut:

$$\mu = SFF - (Input - Point2) \times Slope2 \quad (4.8)$$

d. Kaidah atur kontroler logika *fuzzy*

Proses pembuatan *rules* dilakukan dengan menerapkan kemampuan manusia dalam mengendalikan suatu sistem kendali. Secara umum respon fungsi step pada suatu sistem kendali akan memberikan output seperti yang terlihat pada gambar 4.12



**Gambar 4.12.** Grafik Respon Output terhadap waktu  
Sumber : [www.uk.petra.ac.id/inkubator/thiang/](http://www.uk.petra.ac.id/inkubator/thiang/)

Pada saat sistem diaktifkan ( $output=0$  dan  $waktu =0$ ) output akan naik dengan cepat pada *setting point* (SP), bahkan jauh melebihi SP. Output ini oleh kontroler akan diturunkan menuju SP, tetapi proses penurunan ini mengalami kelebihan sehingga outputnya akan berosilasi di sekitar SP, hingga akhirnya outputnya mencapai SP. Apa yang dilakukan kontroler harus sesuai dengan harga output saat itu



dan sebelumnya (tergantung dari *error* dan  $\Delta error$ ) sehingga kontroler dapat mengambil tindakan yang tepat untuk menyesuaikan outputnya.

Dari gambar 4.12 terlihat jika *error* adalah positif besar dan  $\Delta error$  adalah nol maka output yang harus dikeluarkan adalah positif dan besar (point **a**), atau bila dituliskan dalam bentuk rule:

**If** error  $e(n)$  is positif big **and**  $\Delta error$  is zero **then** output  $u(n)$  is positive big.

Kaidah atur kontroler logika *fuzzy* pada skripsi ini didasarkan pada pengalaman dan dalam bentuk Jika-Maka (*if-then*). Setelah *crisp input* diubah menjadi *input fuzzy*, selanjutnya diolah sesuai dengan kaidah aturnya. Karena terdapat dua pengendalian, maka juga terdapat dua kaidah atur kontroler logika *fuzzy*.

Metode penalaran *fuzzy* yang dipergunakan oleh kontroler logika *fuzzy* adalah metode MIN-MAX.

Sebagai contoh untuk kondisi

If Error is NB and  $\Delta$  Error is NB then Posisi Pahat terhadap benda kerja is PB. Karena Error dan  $\Delta$  Error sangat besar, ini menandakan posisi pahat kurang maju mendekati benda kerja maka posisi pahat memerlukan posisi sangat maju atau kondisi PB (Positive Big). Kaidah atur selengkapnya terdapat dalam Tabel 4.4

**Tabel 4.1** Tabel Kaidah Atur Pengaturan

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

Sumber: Perancangan

Keterangan :

PB : Kondisi Pahat terlalu maju dari titik acuan benda kerja

PS : Kondisi Pahat sedikit maju dari titik acuan benda kerja

Z : Kondisi Pahat relatif diam

NS : Kondisi Pahat kurang mundur dari titik acuan benda kerja

NB : Kondisi Pahat terlalu mundur dari titik acuan benda kerja

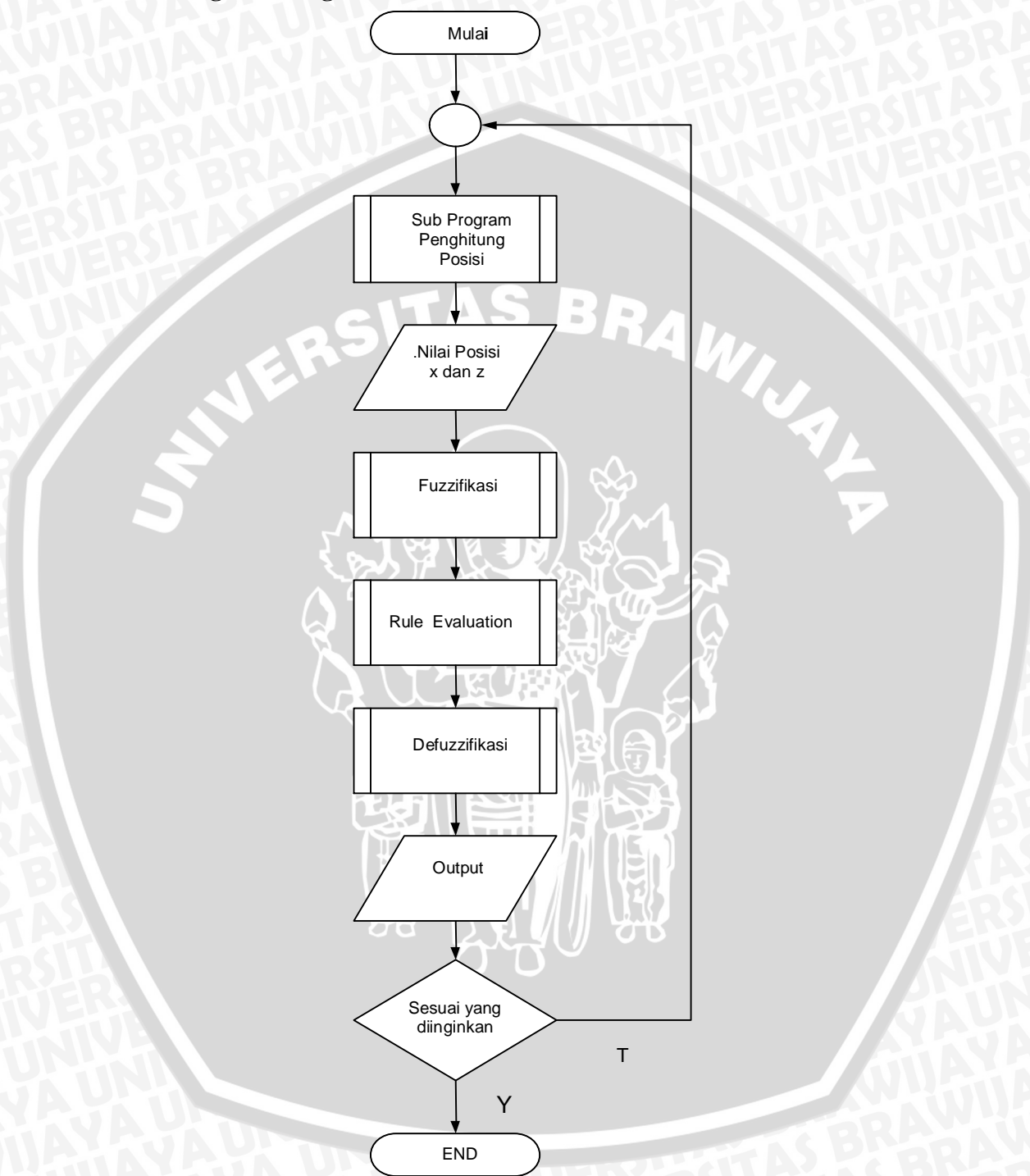
e. *Defuzzifikasi*

*Defuzzifikasi* adalah proses untuk merubah *output fuzzy* menjadi *crisp output*.

Hasil *defuzzifikasi* inilah yang digunakan untuk mengatur besarnya tegangan pada motor DC. Metode *defuzzifikasi* yang digunakan adalah COG (*center of Gravity*).



### 4.2 Perancangan Perangkat Lunak



Gambar 4.13. Diagram alir program utama

Sumber: Perancangan

## BAB V

### PENGUJIAN ALAT

#### 5.1 Pengujian Perangkat Keras

##### 5.1.1 Pengujian Sensor Posisi

###### a. Tujuan

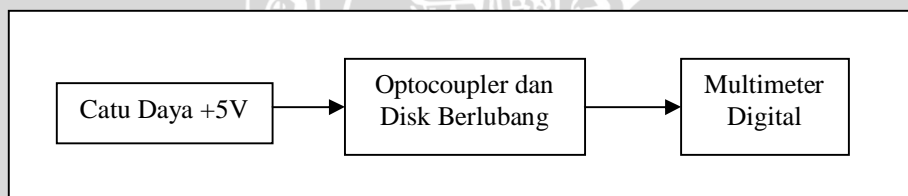
Mengetahui level tegangan sinyal listrik oleh rangkaian *optocoupler* pada saat cahaya yang dipancarkan LED terhalang atau tidak terhalang oleh bagian disk.

###### b. Peralatan

- rangkaian *optocoupler* dan disk yang berlubang
- catu daya + 5 volt
- multimeter digital

###### c. Langkah Pengujian

- merangkai peralatan seperti dalam Gambar 5.1. Keluaran rangkaian terhubung dengan multimeter digital.
- mengaktifkan catu daya.
- menggerakkan prototipe alat bubut pada koordinat z
- mencatat tegangan yang tampak pada multimeter digital saat cahaya yang dipancarkan LED terhalang atau tidak terhalang.



**Gambar 5.1** Pengujian Sensor Posisi

Sumber : Perancangan

###### d. Hasil Pengujian dan Analisis

**Tabel 5.1** Hasil Pengujian Sensor Posisi

Cahaya LED	Tegangan pada multimeter digital (volt)	Logika
Terhalang	4,32	1
Tidak Terhalang	0,15	0

Dari Tabel 5.1 terlihat bahwa pada saat terhalang akan sensor posisi akan memberi logika “1” sedangkan pada saat tidak terhalang akan memberi logika “0”.

Sensor posisi ini dapat digunakan dengan baik sampai pada posisi pergerakan 40 mm. Sehingga sensor posisi ini bisa dikatakan berjalan dengan baik pada aplikasi yang direncanakan.

### 5.1.2 Pengujian DAC

#### a. Tujuan

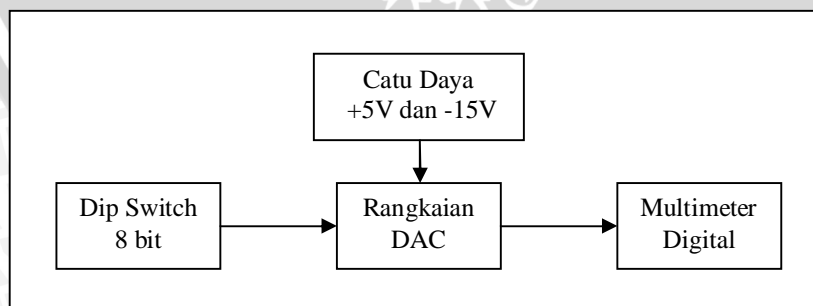
Mengetahui keluaran analog yang dihasilkan oleh DAC berdasarkan masukan digital yang diberikan.

#### b. Peralatan

- rangkaian DAC 0808
- catu daya +5 volt dan -15 volt
- dip switch 8 bit
- multimeter digital

#### c. Langkah Pengujian

- merangkai peralatan seperti dalam Gambar 5.2. Keluaran DAC terhubung dengan multimeter digital.
- mengaktifkan catu daya
- mengubah kontak dip switch 8 bit sesuai dengan data masukan yang diinginkan.
- mencatat tegangan yang tampak pada multimeter digital berdasarkan data masukan yang diinginkan. Pengujian dilakukan mulai data masukan 00 H sampai FF H dengan kenaikan sebesar 25 heksa.



Gambar 5.2 Pengujian DAC

Sumber : Perancangan

d. Hasil Pengujian dan Analisis

Hasil pengujian DAC yang ada dalam tabel 5.2. terlihat kesalahan data berkisar dari 0,351% sampai dengan 2,417% dengan rata-rata kesalahan 1.495%. Dengan rata-rata kesalahan kurang dari 5% maka DAC ini bisa dikatakan berjalan dengan baik pada aplikasi yang direncanakan.

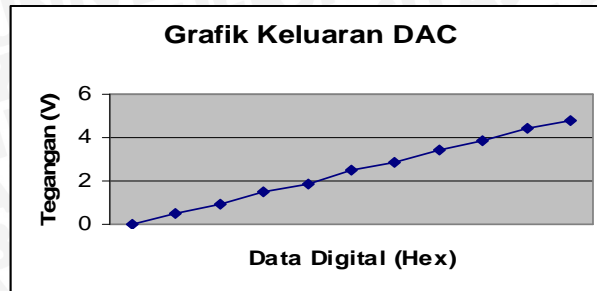
Prosentase kesalahan dari hasil pengujian terhadap nilai yang diinginkan adalah sebagai berikut:

$$\text{Prosentase penyimpangan} = \frac{|V_0 \text{ perhitungan} - V_0 \text{ pengukuran}|}{V_0 \text{ perhitungan}} \times 100\%$$

$$\text{Penyimpangan rata-rata} = \frac{\sum \text{Prosentase Penyimpangan}}{\sum \text{pengujian}} = \frac{16,455}{11} = 1,495\%$$

Tabel 5.2 Hasil Pengujian DAC

Data Yang Dimasukkan (Hex)	Vo DAC dari Pengukuran (Volt)	Vo DAC dari Perhitungan (Volt)	Kesalahan (%)
00	0	0	0
19	0.47	0.488	1.639
32	0.94	0.977	1.740
4B	1.50	1.465	2.389
64	1.89	1.953	2.202
7D	2.50	2.441	2.417
96	2.88	2.930	1.706
AF	3.43	3.418	0.351
C8	3.85	3.906	1.434
E1	4.42	4.395	0.569
FF	4.80	4.98	2.008



Gambar 5.3 Grafik Keluaran DAC

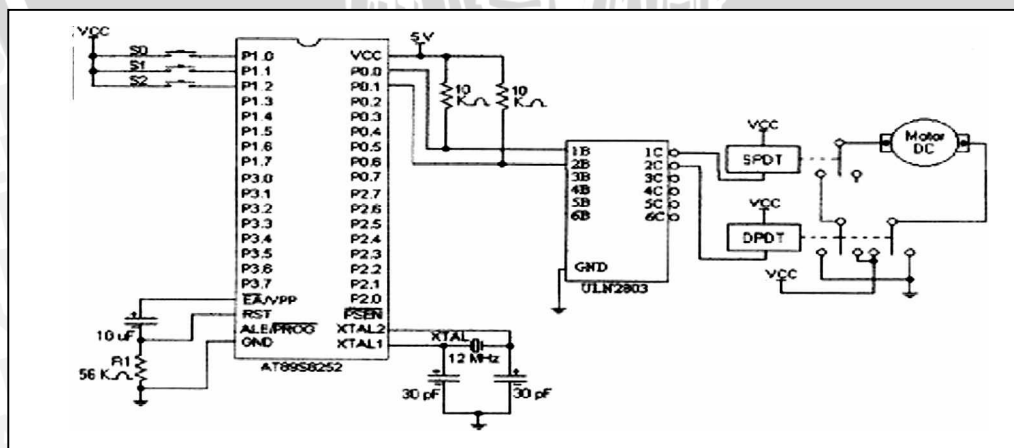
### 5.1.3 Pengujian Driver Motor DC

#### a. Tujuan

Pengujian ini dilakukan untuk mengetahui kinerja mikrokontroler melakukan pengendalian pergerakan motor DC dengan baik. Dalam hal ini motor DC harus bisa berputar dua arah yang berlawanan.

#### b. Peralatan

- Catu daya 5 volt
- Sistem minimum mikrokontroler AT89C51
- 3 buah sakelar
- 1 buah IC ULN2803
- 1 buah relay SPDT
- 1 buah DPDT
- 1 buah motor DC



Gambar 5.4. Pengujian Driver Motor DC

c. Prosedur Pengujian

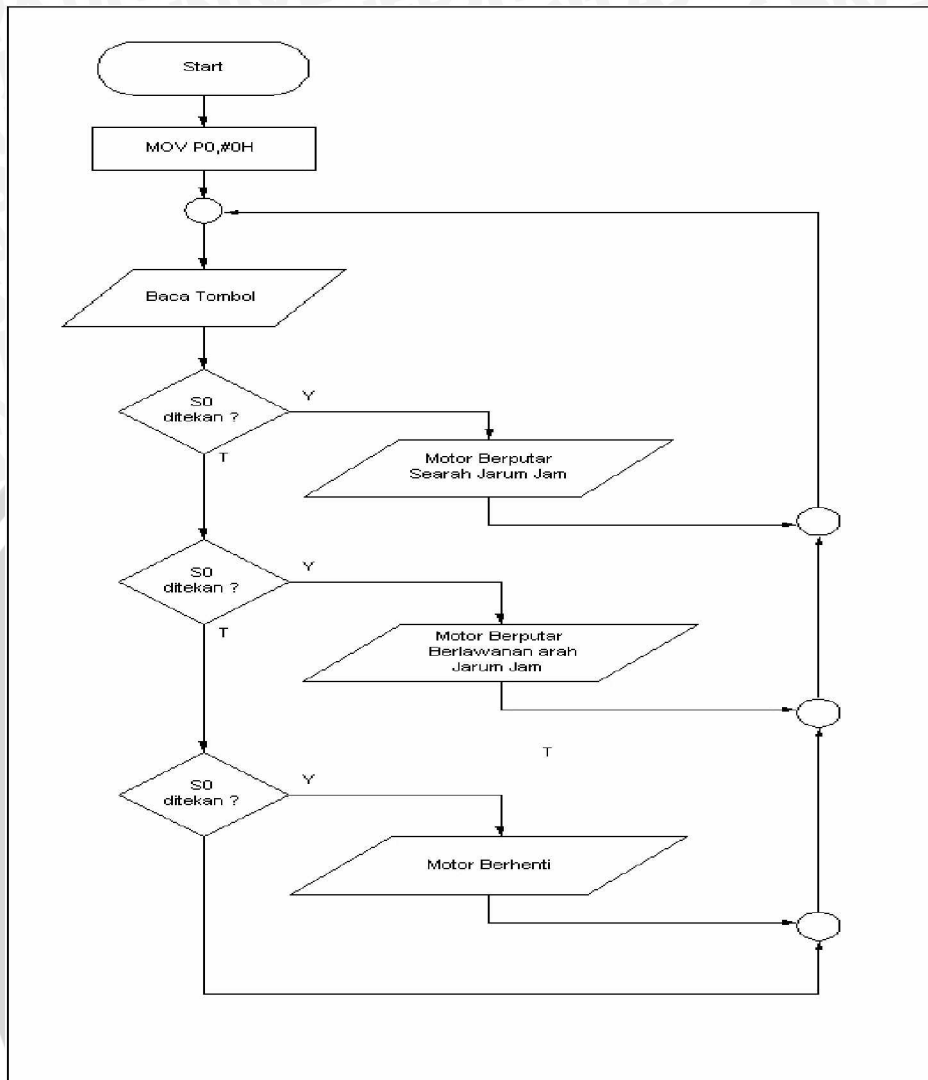
- a. Menuliskan program motor.h51 ke dalam mikrokontroler. Pada program ini, penekanan tombol S0 akan menyebabkan motor DC berputar searah jarum jam, penekanan tombol S1 akan menyebabkan motor DC berputar berlawanan arah jarum jam dan penekanan tombol S2 akan menyebabkan motor berhenti.
- b. Merangkai alat-alat seperti dalam Gambar 5.4.
- c. Memberikan masukan dengan menekan tombol sakelar S0 dan S1 secara bergantian.
- d. Mengamati arah putaran motor DC.

**Tabel.5.3.** Data Hasil Pengujian *Driver* Motor DC

<b>Tombol Yang Ditekan</b>	<b>Kondisi Motor DC</b>
Tidak Ada	Tidak Berputar
S0	Berputar Searah Jarum Jam
S1	Berputar Berlawanan arah Jarum Jam
S2	Berhenti

Dari data hasil pengujian bisa diambil kesimpulan bahwa *driver* motor DC bisa bekerja dengan baik.





Gambar 5.5. Diagram Alir program

## 5.2 Pengujian Perangkat Lunak

### 5.2.1 Pengujian Sistem Mikrokontroler

#### a. Tujuan

Mengetahui kondisi awal dari sistem mikrokontroler agar sesuai dengan yang diharapkan.

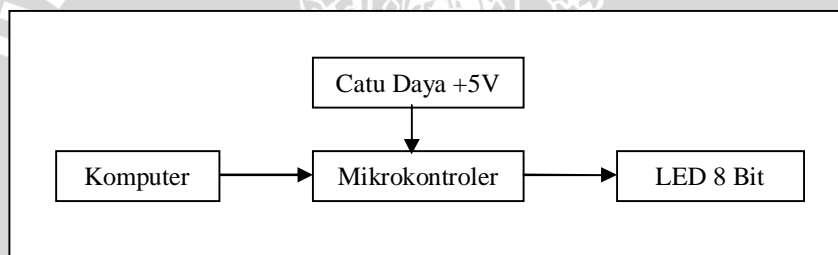
#### b. Peralatan

- komputer dengan kabel penghubungnya
- minimum sistem mikrokontroler AT89C51

- lampu LED mewakili keluaran 8 bit
- catu daya +5volt

c. Langkah Pengujian

- merangkai peralatan seperti dalam Gambar 5.6. Keluaran terhubung dengan lampu LED yang mewakili keluaran 8 bit.
- mengisi mikrokontroler dengan program sederhana yang meletakkan 0FH dan FOH pada Accumulator secara bergantian kemudian memindahkannya ke port 1. kemudian download pada mikrokontroler AT89C51.
- mengaktifkan catu daya.
- mereset program untuk mengeluarkan data mikrokontroler.
- mencatat data keluaran yang diwakili oleh lampu LED 8 bit ke dalam bentuk biner.



Gambar 5.6 Pengujian Mikrokontroler

e. Hasil Pengujian dan Analisis

Tabel 5.4. Hasil Pengujian Sistem mikrokontroler

Kondisi	Keluaran pada LED Display							
	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
I	1	1	1	1	0	0	0	0
II	0	0	0	0	1	1	1	1

Dari tabel 5.4 terlihat bahwa port 1 memberikan logika 0FH dan FOH secara bergantian sesuai dengan isi program

5.2.2 Pengujian *Fuzzy inference System*

a. Tujuan

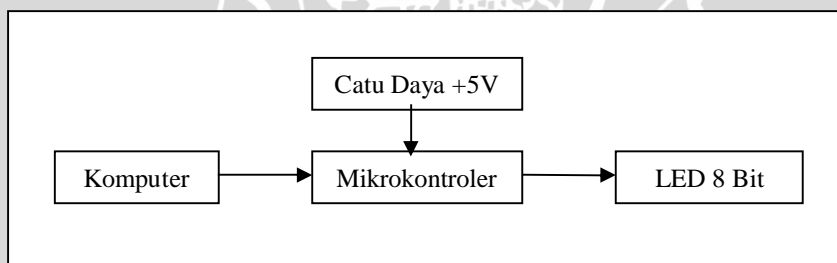
Mengetahui jalannya program *fuzzy inference system* dengan adanya dua masukan, Error dan  $\Delta$  Error

b. Peralatan

- komputer dengan kabel penghubungnya
- minimum sistem mikrokontroler AT89C51
- lampu LED mewakili keluaran 8 bit
- catu daya +5volt

c. Langkah Pengujian

- merangkai peralatan seperti dalam Gambar 5.7. Keluaran terhubung dengan lampu LED yang mewakili keluaran 8 bit.
- mengisi mikrokontroler dengan program dengan *kernel fuzzy inference system*. Untuk pegujian ini data masukan yang diberikan untuk Error sebesar 78H dan untuk  $\Delta$  Error sebesar 56H. Kemudian download pada mikrokontroler AT89C51.
- mengaktifkan catu daya.
- mereset program untuk mengeluarkan data mikrokontroler.
- mencatat data keluaran yang diwakili oleh lampu LED 8 bit ke dalam bentuk biner.



d. Hasil Pengujian dan Analisis

**Tabel 5.5** Hasil Pengujian *Fuzzy Inference System*

Keluaran pada LED Display							
Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
1	0	0	0	1	1	0	1

Dari Tabel 5.5 terlihat nilai *crisp output* yang dihasilkan sesuai dengan perencanaan, dengan ini bisa disimpulkan *kernel fuzzy inference system* berjalan sebagaimana mestinya.

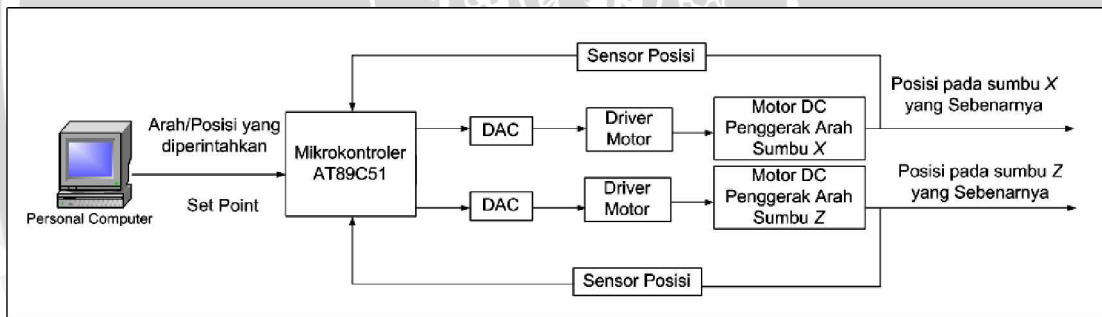
### 5.3 Pengujian Sistem Secara Keseluruhan

#### a. Tujuan

Pengujian ini bertujuan untuk mengetahui unjuk kerja perangkat lunak dengan perangkat keras saat diintegrasikan bersama-sama. Pengujian dilakukan dengan merangkaikan alat-alat seperti pada Gambar 5.8, kemudian mengaktifkan semua alat yang telah dirakit setelah itu mengamati kerja dari sistem. Hasil keluaran dari *crisp output* diukur dan dicatat.

#### b. Peralatan

- sensor posisi
- minimum sistem mikrokontroler AT89C51
- rangkaian DAC
- rangkaian driver motor DC
- motor DC



Gambar 5.8 Pengujian Sistem secara Keseluruhan

#### d. Hasil Pengujian dan Analisis

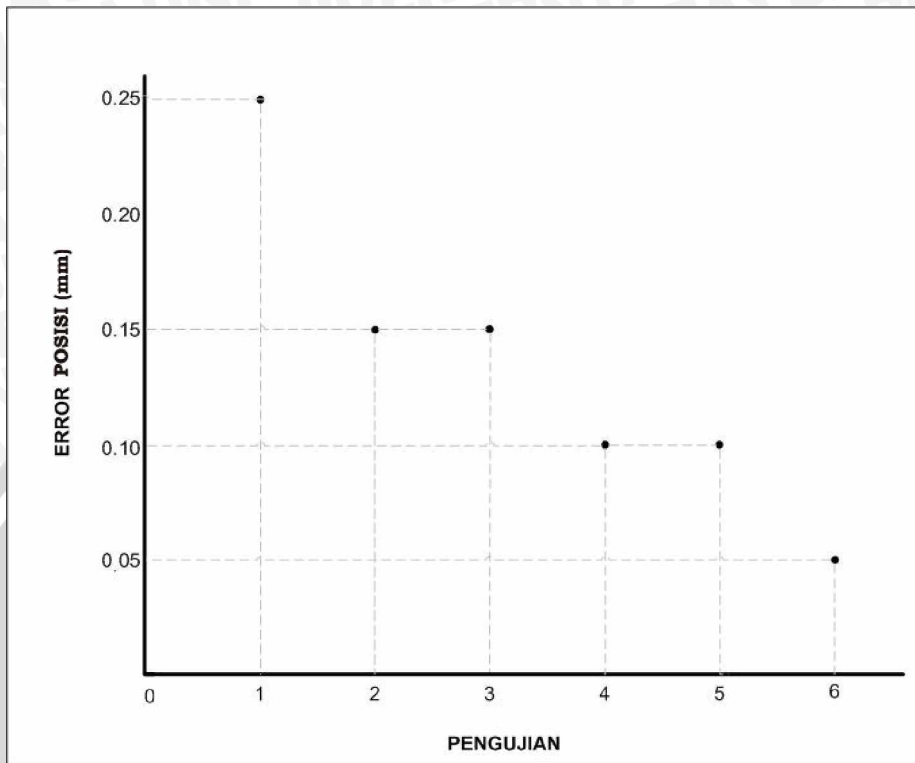
Disini dipakai 4 acuan posisi yaitu  $(X_0, Z_5)$ ,  $(X_{10}, Z_{10})$ ,  $(X_{15}, Z_{15})$ ,  $(X_{25}, Z_{20})$ .

Benda kerja yang digunakan berdiameter 4,5 mm

Tabel 5.6 Hasil Pengujian Koordinat  $(X_0, Z_5)$

Pengujian	Posisi (X,Z)	Error (mm)	Error (%)
1	$(X_0, Z_{4.75})$	0.25	5
2	$(X_0, Z_{4.85})$	0.15	3
3	$(X_0, Z_{4.85})$	0.15	3
4	$(X_0, Z_{4.90})$	0.10	2
5	$(X_0, Z_{4.90})$	0.10	2
6	$(X_0, Z_{4.95})$	0.05	1





**Gambar 5.9.** Grafik error posisi pada pengujian dengan pengesetan koordinat (X0,Z5) Prosentase kesalahan dari hasil pengujian terhadap nilai yang diinginkan adalah sebagai berikut:

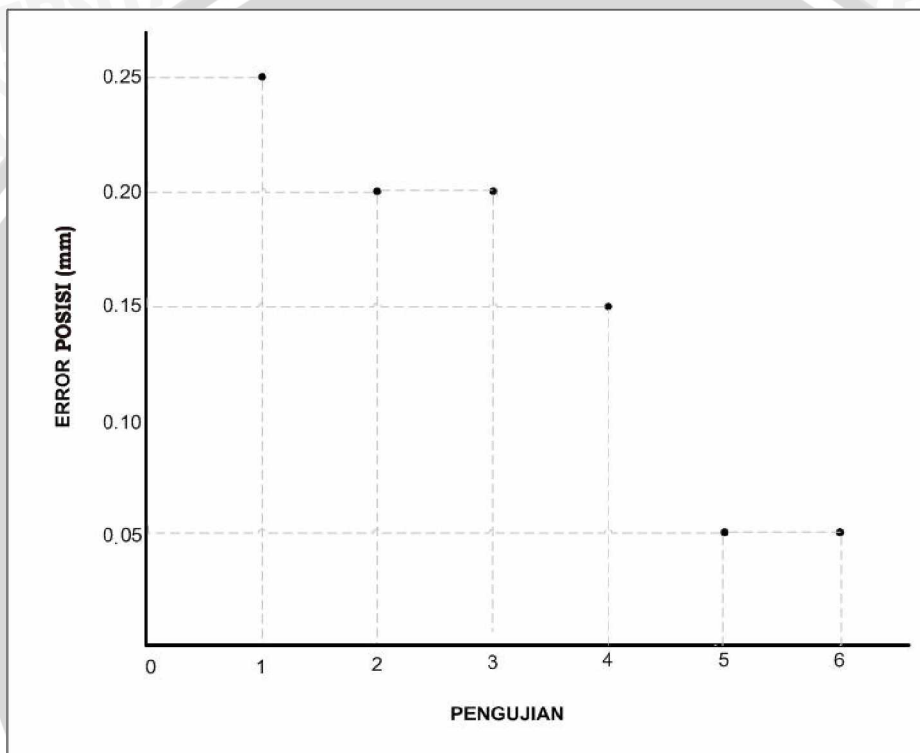
$$\text{Prosentase penyimpangan} = \frac{|V_0 \text{ perhitungan} - V_0 \text{ pengukuran}|}{V_0 \text{ perhitungan}} \times 100\%$$

$$\text{Penyimpangan rata-rata} = \frac{\sum \text{Prosentase Penyimpangan}}{\sum \text{pengujian}} = \frac{16}{6} = 2,667\%$$

Dari Tabel 5.6. pada hasil pengujian pergerakan mata pahat pada prototipe alat bubut pada koordinat (X0,Z5) didapatkan kesalahan data berkisar 1% sampai 5% sedangkan kesalahan rata-rata 2,667%

**Tabel 5.7** Hasil Pengujian Koordinat (X10,Z10)

Pengujian	Posisi (X,Z)	Error (mm)	Error (%)
1	(X9.75,Z10)	0.25	2.5
2	(X9.80,Z10)	0.20	2
3	(X9.80,Z10)	0.20	2
4	(X9.85,Z10)	0.15	1.5
5	(X9.95,Z10)	0.05	0.5
6	(X9.95,Z10)	0.05	0.5



**Gambar.5.10.** Grafik error posisi pada pengujian dengan pengesetan koordinat (X10,Z10)

Prosentase kesalahan dari hasil pengujian terhadap nilai yang diinginkan adalah sebagai berikut:

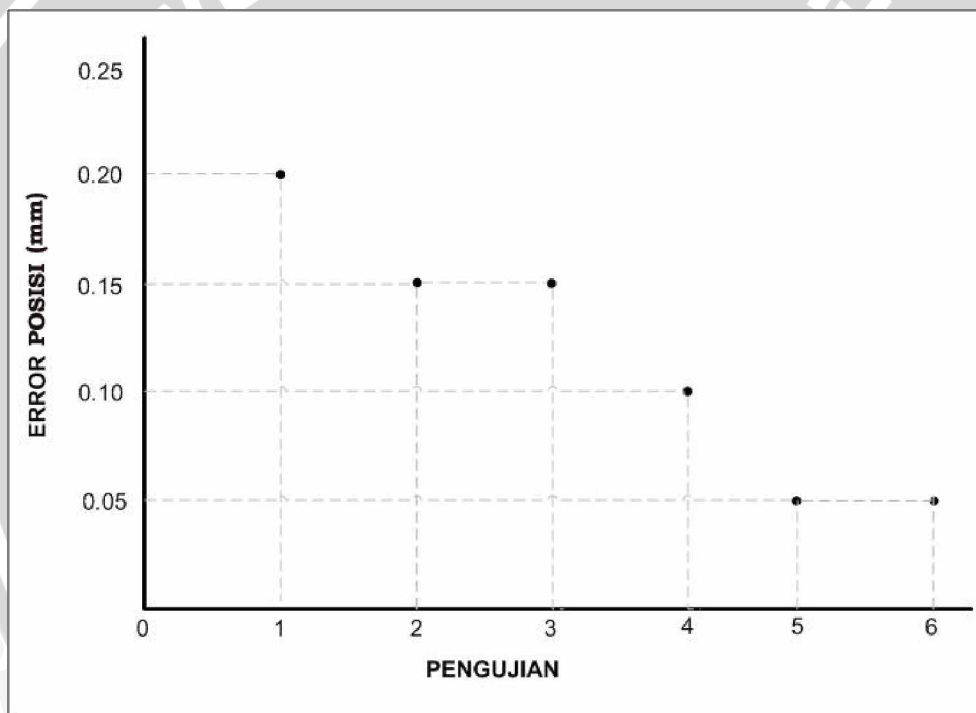
$$\text{Prosentase penyimpangan} = \frac{|V_0 \text{ perhitungan} - V_0 \text{ pengukuran}|}{V_0 \text{ perhitungan}} \times 100\%$$

$$\text{Penyimpangan rata-rata} = \frac{\sum \text{Pr osentase Penyimpangan}}{\sum \text{pengujian}} = \frac{9}{6} = 1,5\%$$

Dari Tabel 5.7 pada hasil pengujian pergerakan mata pahat pada prototipe alat bubut pada koordinat (X10,Z10) didapatkan kesalahan data berkisar 0,5% sampai 2,5% sedangkan kesalahan rata-rata 1,5%

**Tabel 5.8** Hasil Pengujian Koordinat (X15,Z15)

Pengujian	Posisi (X,Z)	Error (mm)	Error (%)
1	(X14.80,Z15)	0.20	1.33
2	(X14.85,Z15)	0.15	1
3	(X14.85,Z15)	0.15	1
4	(X14.90,Z15)	0.10	0.667
5	(X14.95,Z15)	0.05	0.333
6	(X14.95,Z15)	0.05	0.333



**5.11.** Grafik Grafik error posisi pada pengujian dengan pengesetan koordinat (X15,Z15)

Prosentase kesalahan dari hasil pengujian terhadap nilai yang diinginkan adalah sebagai berikut:

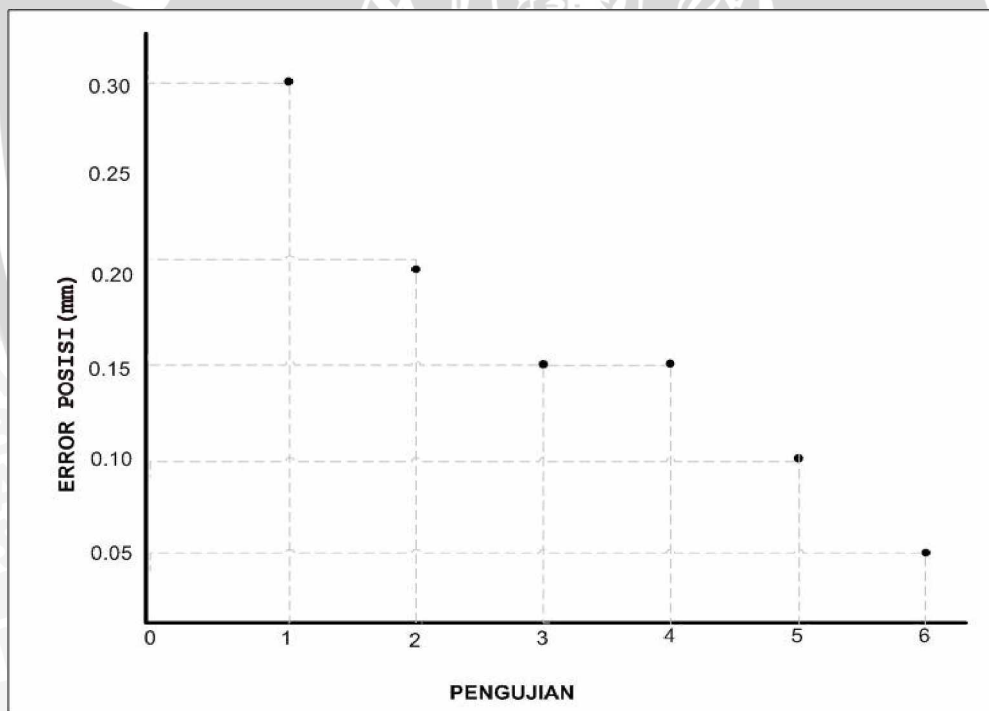
$$\text{Prosentase penyimpangan} = \frac{|V_0 \text{ perhitungan} - V_0 \text{ pengukuran}|}{V_0 \text{ perhitungan}} \times 100\%$$

$$\text{Penyimpangan rata-rata} = \frac{\sum \text{Prosentase Penyimpangan}}{\sum \text{pengujian}} = \frac{4,663}{6} = 0,777\%$$

Dari Tabel 5.8. pada hasil pengujian pergerakan mata pahat pada prototipe alat bubut pada koordinat (X15,Z15) didapatkan kesalahan data berkisar 0,33% sampai 1,33% sedangkan kesalahan rata-rata 0,777%

**Tabel 5.9** Hasil Pengujian Koordinat (X25,Z20)

Pengujian	Posisi (X,Z)	Error (mm)	Error (%)
1	(X25,Z19.70)	0.30	1,5
2	(X25,Z19.80)	0.20	1
3	(X25,Z19.85)	0.15	0.75
4	(X25,Z19.85)	0.15	0.75
5	(X25,Z19.90)	0.10	0.5
6	(X25,Z19.95)	0.05	0.25



**5.12.** Grafik error posisi pada pengujian dengan pengesetan koordinat (X25,Z20)

Prosentase kesalahan dari hasil pengujian terhadap nilai yang diinginkan adalah sebagai berikut:

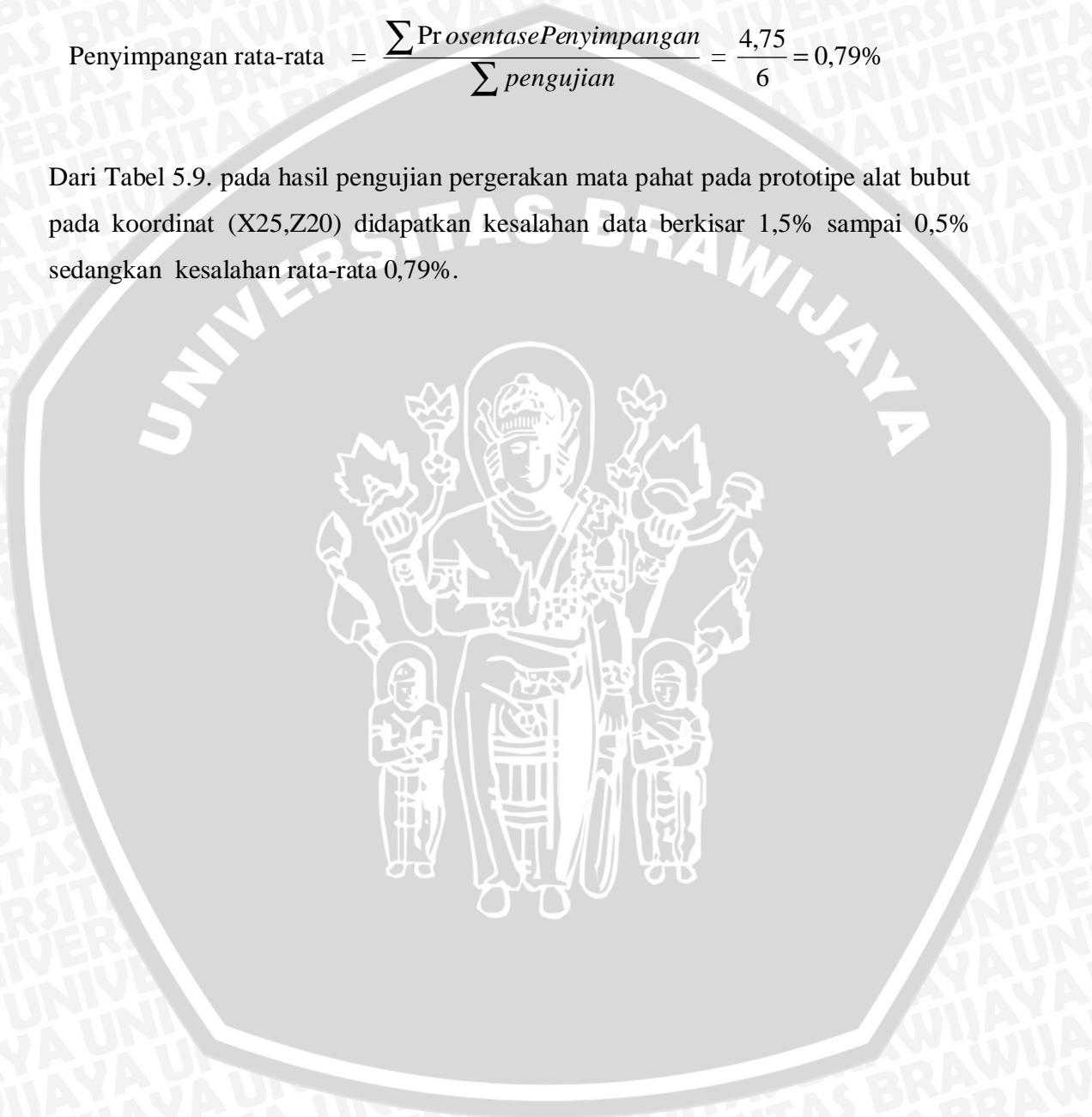




$$\text{Prosentase penyimpangan} = \left| \frac{V_0 \text{ perhitungan} - V_0 \text{ pengukuran}}{V_0 \text{ perhitungan}} \right| \times 100\%$$

$$\text{Penyimpangan rata-rata} = \frac{\sum \text{Prosentase Penyimpangan}}{\sum \text{pengujian}} = \frac{4,75}{6} = 0,79\%$$

Dari Tabel 5.9. pada hasil pengujian pergerakan mata pahat pada prototipe alat bubut pada koordinat (X25,Z20) didapatkan kesalahan data berkisar 1,5% sampai 0,5% sedangkan kesalahan rata-rata 0,79%.



## BAB VI

### KESIMPULAN DAN SARAN

#### 6.1 Kesimpulan

Dari hasil perancangan dan pengujian alat dapat disimpulkan bahwa:

- a. Data *crisp output* yang berasal dari mikrokontroler dapat dikonversi oleh rangkaian DAC 0808 dengan kesalahan rata-rata 1,495 %..
- b. Sistem kontrol logika fuzzy yang dirancang untuk sistem pergerakan prototipe alat bubut yang bergerak sesuai dengan arah koordinat  $x$  dan  $z$  menggunakan motor DC dapat bekerja sesuai dengan koordinat yang diinginkan.
- c. Dalam membentuk sketsa suatu benda kerja dapat digunakan dengan menggunakan pemograman Delphi 5.0. dari Komputer Personal yang dipasang secara antarmuka dengan modul mikrokontroler AT89C51 untuk menggerakkan prototipe alat bubut.
- d. Kesalahan rata-rata sistem didapatkan dengan perhitungan :

$$\text{Kesalahan rata-rata sistem} = \frac{\sum \text{Peyimpangan rata-rata}}{\sum \text{Pengujian sistem}} = \frac{5,4734}{4}$$

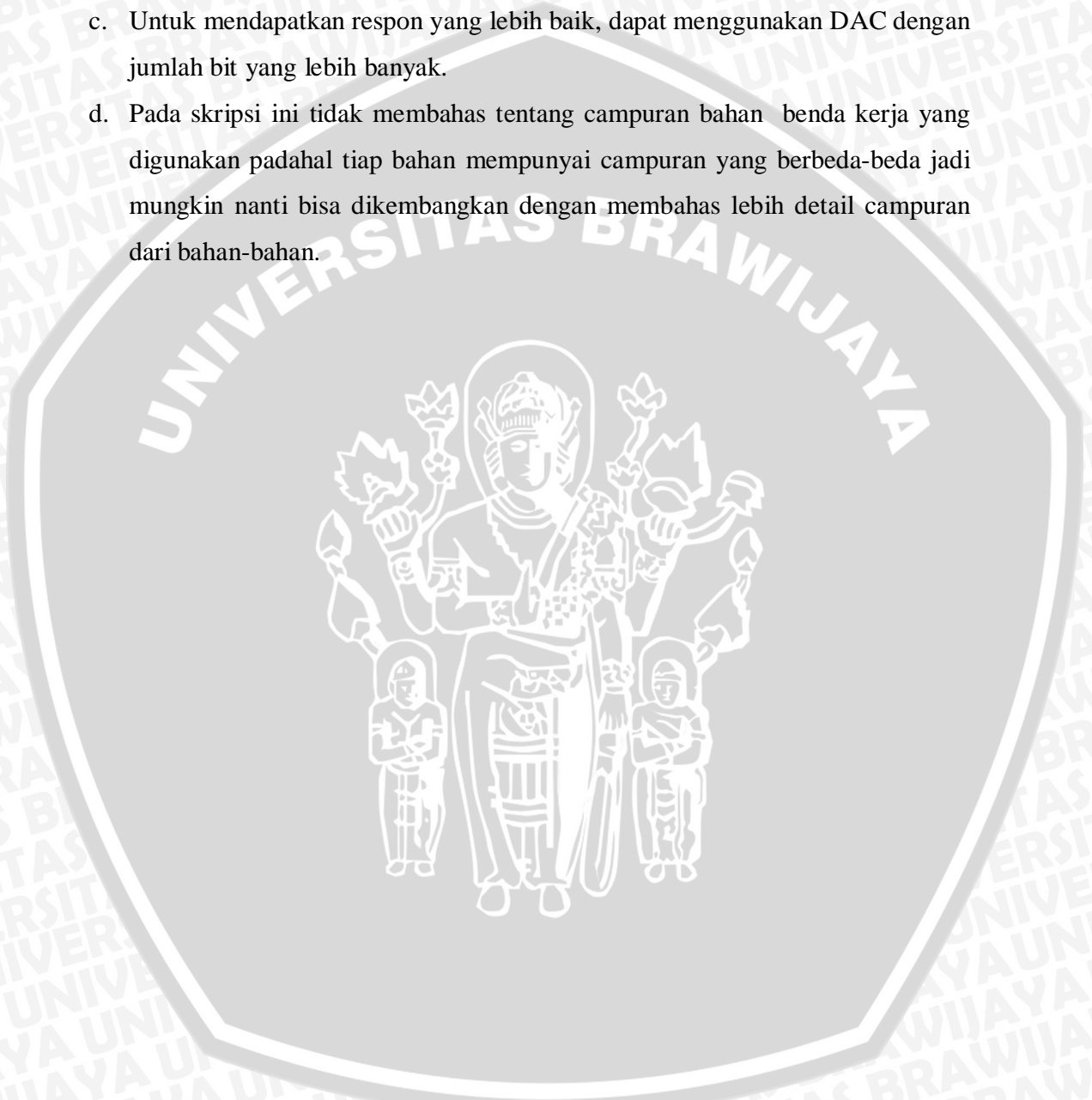
sebesar 1,4335%

#### 6.2 Saran

Dalam perancangan dan pembuatan alat ini masih terdapat kelemahan dan kekurangan sistem. Beberapa hal yang dapat dijadikan saran dalam alat ini adalah:

- a. Untuk meningkatkan kemampuan *fuzzy logic controller* dapat dilakukan dengan menambah jumlah fungsi keanggotaan dan memodifikasi aturan-aturan *fuzzy* pada masing-masing variabel masukan dan keluarannya..

- b. Untuk meningkatkan kemampuan kinerja dari prototipe alat bubut, koordinat pergerakan mata pahat alat bubut yang digunakan lebih dari 2 koordinat yaitu menggunakan koordinat  $x,y,z$
- c. Untuk mendapatkan respon yang lebih baik, dapat menggunakan DAC dengan jumlah bit yang lebih banyak.
- d. Pada skripsi ini tidak membahas tentang campuran bahan benda kerja yang digunakan padahal tiap bahan mempunyai campuran yang berbeda-beda jadi mungkin nanti bisa dikembangkan dengan membahas lebih detail campuran dari bahan-bahan.



## DAFTAR PUSTAKA

- Ayala, K.J, 1991, *The 8051 Microcontroller Architecture, Programming and Application*. USA: West Publishing Company.
- Klir, G.J, dan Yuan, B, 1995, *Fuzzy Sets and Fuzzy Logic Theory and Applications*. Prentice Hall International Inc.
- Hilloowala, R. M and Sharaf, A. M., 1996, *A Rule-Based Fuzzy Logic Controller for PWM Inverter a Stand Alone Energy Conversion Scheme*, IEEE Trans. On Industrila Applications, Vol 32. No. 1 Page 57-65.
- Jamshidi, M., Vadiie, N., Ross, Timothy J.,1993, *Fuzzy Logic and Control Software and Hardware Application*, New Jersey, Prentice-Hall Inc.
- Kuswandi, S, *Kendali Cerdas (Intelligent Control)*, EEPIS Press, Surabaya.
- Malvino, A.P, 1995, *Prinsip-Prinsip Elektronika*, Edisi Kedua, Terjemahan Hanafi Gunawan, Jakarta : Erlangga.
- Malvino, A.P, 1996, *Prinsip-Prinsip Elektronika*, Jilid Pertama, Terjemahan Prof. M. Barnawi, Ph.D., Jakarta : Erlangga.
- National Semiconductor. 1995, *National Application Specific Analog Products*,, Santa Clara : National Semiconductor Corporation.
- Yan, Jun, 1994, *Using Fuzzy Logic*, New York, Prentice Hall.
- Daryanto, 2002, *Mesin Perkakas bengkel*, Rineka Putra, Jakarta.
- Gibbs, D, dan Crandell T.M, 2001, *Dasar-Dasar Teknik dan Pemograman CNC*, Rosda Jayaputra, Jakarta

UNIVERSITAS BRAWIJAYA

LAMPIRAN 1  
LISTING PROGRAM  
ASSEMBLER



```
LOC  OBJ          LINE      SOURCE
0090          1      motor      bit p1.0
00B2          2      sensor_x   bit p3.2
00B3          3      sensor_y   bit p3.3
00B4          4      sensor_bahan bit p3.4
0030          5      step_y     equ 30h
0031          6      step_x     equ 31h
0032          7      setdelay   equ 32h
0080          8      port_stepper equ p0
          9
0000         10      org 0
0000 02018E      11          jmp mulai
0023         12      org 23h
0023 C2AC         13          clr     es
0025 3098FD      14          jnb    ri,$
0028 C298         15          clr    ri
002A E599         16          mov    A,sbuf ; data serial
dimasukkan accumulator
          17          ;      mov    p1,a
002C B4FE05      18          cjne   a,#254,ujiakhir
002F 7850         19          mov    r0,#50H
0031 020095      20          jmp    slsint
0034 B4FF5C      21      ujiakhir: cjne   a,#255,ujiisi
0037 7454         22          MOV    A,#'T'
0039 12019C      23          CALL   KIRIMSERIAL
003C A950         24          mov    r1,50h
003E B97906      25          cjne   r1,#'y',ujiyB
0041 120098      26          CALL   GESER_MAJU
0044 020095      27          jmp    slsint
0047 B95906      28      ujiyB:  cjne   r1,#'Y',ujix
004A 1200BA      29          CALL   GESER_mundur
004D 020095      30          jmp    slsint
0050 B97806      31      ujix:   cjne   r1,#'x',ujixb
0053 1200DD      32          CALL   GESER_KANAN
0056 020095      33          jmp    slsint
0059 B95806      34      ujixB:  cjne   r1,#'X',ujil
005C 12011A      35          CALL   GESER_Kiri
005F 020095      36          jmp    slsint
0062 B96C06      37      ujil:   cjne   r1,#'l',ujim
0065 120158      38          CALL   KIRI_PENUH
0068 020095      39          jmp    slsint
006B B96D06      40      ujim:   cjne   r1,#'m',Onmotor
006E 120169      41          CALL   MUNDUR_PENUH
0071 020095      42          jmp    slsint
0074 B95205      43      Onmotor:cjne   r1,#'R',OFFmotor
0077 D290         44          setb   motor
0079 020095      45          jmp    slsint
007C B97205      46      OFFmotor:cjne   r1,#'r',UJISD
007F C290         47          clr    motor
0081 020095      48          jmp    slsint
0084 B9440E      49      UJISD:  cjne   r1,#'D',slsint
0087 E551         50          MOV    A,51H
```

LOC	OBJ	LINE	SOURCE
0089	F532	51	MOV SETDELAY,A
008B	744F	52	MOV A,#'O'
008D	12019C	53	CALL KIRIMSERIAL
0090	020095	54	JMP SLSINT
0093	F6	55	ujiisi: mov @r0,a
0094	08	56	inc r0
0095	D2AC	57	slsint: setb es
0097	32	58	reti
		59	
		60	GESER_MAJU:
0098	A951	61	MOV R1,51H
		62	ULANG_MAJU:
009A	0530	63	INC step_y
009C	E530	64	mov a,step_y
009E	75F008	65	mov b,#8
00A1	84	66	div ab
00A2	E5F0	67	mov a,b
00A4	F530	68	mov step_y,a
00A6	9001C1	69	Mov dptr,#pola_y
00A9	93	70	movc a,@+dptr
00AA	F580	71	mov port_stepper,a
00AC	1201B8	72	call delay
00AF	D9E9	73	DJNZ R1,ULANG_MAJU
00B1	758000	74	mov port_stepper,#0
00B4	744F	75	MOV A,#'O'
00B6	12019C	76	CALL KIRIMSERIAL
00B9	22	77	RET
		78	GESER_MUNDUR:
00BA	A951	79	MOV R1,51H
		80	ULANG_Mundur:
00BC	E530	81	mov a,step_y
00BE	2408	82	add a,#8
00C0	14	83	dec a
00C1	75F008	84	mov b,#8
00C4	84	85	div ab
00C5	E5F0	86	mov a,b
00C7	F530	87	mov step_y,a
00C9	9001C1	88	Mov dptr,#pola_y
00CC	93	89	movc a,@+dptr
00CD	F580	90	mov port_stepper,a
00CF	1201B8	91	call delay
00D2	D9E8	92	DJNZ R1,ULANG_Mundur
00D4	758000	93	mov port_stepper,#0
00D7	744F	94	MOV A,#'O'
00D9	12019C	95	CALL KIRIMSERIAL
00DC	22	96	RET
		97	GESER_KANAN:
00DD	A951	98	MOV R1,51H
00DF	B90003	99	cjne r1,#0,ul_kanan
00E2	0200EE	100	jmp sisa_kanan
		101	UL_kanan:
00E5	7A64	102	MOV R2,#100
		103	ULANG_KANAN:
00E7	120104	104	CALL step_kanan
00EA	DAFB	105	DJNZ R2,ULANG_KANAN

```
LOC  OBJ          LINE      SOURCE
00EC  D9F7          106          DJNZ R1,UL_kanan
                                107      sisa_kanan:
00EE  A952          108          MOV R1,52H
00F0  B90003        109          cjne r1,#0,ul_kanan1
00F3  0200FB        110          jmp HABISKANAN
                                111      ul_kanan1:
00F6  120104        112          CALL step_kanan
00F9  D9FB          113          DJNZ R1,UL_kanan1
                                114      HABISKANAN:
00FB  758000        115          mov     port_stepper,#0
00FE  744F          116          MOV A,#'O'
0100  12019C        117          CALL KIRIMSERIAL
0103  22            118          RET
                                119
                                120      step_kanan:
0104  0531          121          INC step_x
0106  E531          122          mov a,step_x
0108  75F008        123          mov b,#8
010B  84            124          div ab
010C  E5F0          125          mov a,b
010E  F531          126          mov step_x,a
0110  9001C9        127          Mov dptr,#pola_x
0113  93            128          movc a,@+dptr
0114  F580          129          mov port_stepper,a
0116  1201B8        130          call delay
0119  22            131          RET
                                132      GESER_KIRI:
011A  A951          133          MOV R1,51H
011C  B90003        134          cjne r1,#0,ul_kIRI
011F  02012B        135          jmp sisa_kIRI
0122  7A64          136          ul_kIRI:MOV R2,#100
                                137      ULANG_KIRI:
0124  120141        138          CALL STEP_KIRI
0127  DAFB          139          DJNZ R2,ULANG_KIRI
0129  D9F7          140          DJNZ R1,UL_KIRI
                                141      sisa_kIRI:
012B  A952          142          MOV R1,52H
012D  B90003        143          cjne r1,#0,ul_kIRI1
0130  020138        144          jmp HABIS_kIRI
                                145      ul_kIRI1:
0133  120141        146          CALL STEP_KIRI
0136  D9FB          147          DJNZ R1,UL_KIRI1
                                148      HABIS_kIRI:
0138  758000        149          mov port_stepper,#0
013B  744F          150          MOV A,#'O'
013D  12019C        151          CALL KIRIMSERIAL
0140  22            152          RET
                                153
                                154      STEP_KIRI:
0141  E531          155          mov a,step_x
0143  2408          156          add a,#8
0145  14            157          dec a
0146  75F008        158          mov b,#8
0149  84            159          div ab
014A  E5F0          160          mov a,b
```



```

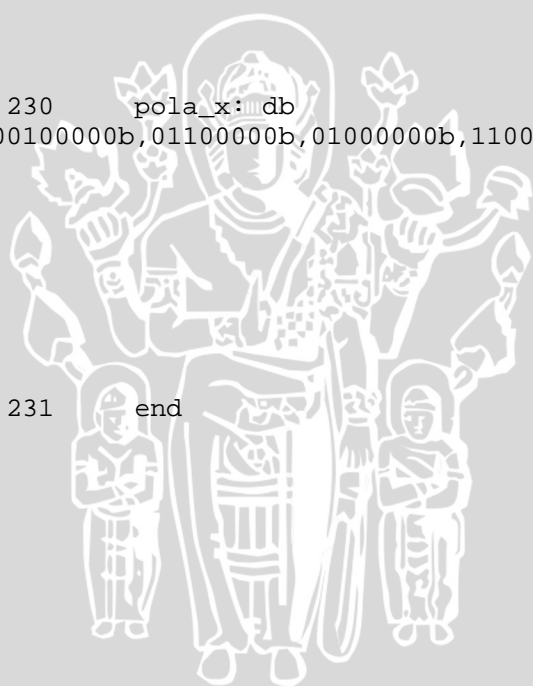
LOC  OBJ          LINE      SOURCE
014C F531          161      mov  step_x,a
014E 9001C9       162      Mov  dptr,#pola_x
0151 93           163      movc  a,@a+dptr
0152 F580          164      mov  port_stepper,a
0154 1201B8       165      call delay
0157 22           166      RET
                                167
                                168      KIRI_PENUH:
0158 20B208       169      JB   SENSOR_X,slskr
015B 3141         170      CALL STEP_KIRI
015D 30B2F8       171      JNB  SENSOR_X,KIRI_PENUH
0160 758000       172      mov  port_stepper,#0
0163 744F         173      slskr: MOV  A,'#0'
0165 12019C       174      CALL KIRIMSERIAL
0168 22           175      RET
                                176
                                177      MUNDUR_PENUH:
0169 20B31C       178      JB   SENSOR_Y,MUNDURsls
016C E530         179      mov  a,step_y
016E 2408         180      add  a,#8
0170 14           181      dec  a
0171 75F008       182      mov  b,#8
0174 84           183      div  ab
0175 E5F0         184      mov  a,b
0177 F530         185      mov  step_y,a
0179 9001C1       186      Mov  dptr,#pola_Y
017C 93           187      movc  a,@a+dptr
017D F580         188      mov  port_stepper,a
017F 1201B8       189      call delay
0182 30B3E4       190      JNB  SENSOR_Y,MUNDUR_PENUH
0185 758000       191      mov  port_stepper,#0
                                192      MUNDURsls:
0188 744F         193      MOV  A,'#0'
018A 12019C       194      CALL KIRIMSERIAL
018D 22           195      ret
018E 1201A8       196      mulai: call  inisialisasi_2400
0191 758000       197      mov  port_stepper,#0
0194 755114       198      mov  51h,#20
0197 753232       199      MOV  SETDELAY,#50
                                200      ; call  GESER_KANAN
                                201      ; call  geser_maju
                                202      ; CALL  MUNDUR_PENUH
                                203      ; CALL  KIRI_PENUH
019A 80FE         204      JMP  $
                                205
                                206      kirimserial:
019C C2AC         207      clr  es
019E F599         208      mov  sbuf,a ; data
serial dimasukkan accumulator
01A0 3099FD       209      jnb  ti,$
01A3 C299         210      clr  ti
01A5 D2AC         211      setb es
01A7 22           212      ret
                                213      ; proses pengaturan komunikasi serial
dengan komputer
                                214      inisialisasi_2400:
01A8 D2AF         215      setb EA

```

```

LOC  OBJ          LINE      SOURCE
01AA  758920        216      mov      TMOD,#20h
01AD  758DF3        217      mov      TH1,#0F3H      ;
01B0  D28E           218      setb    TR1
01B2  759850        219      mov      SCON,#50h
01B5  D2AC           220      setb    es
01B7  22             221      RET
                222      delay:
01B8  AB32           223      MOV     R3,setdelay
01BA  7DFE           224      delay1: MOV     R5,#255
01BC  DDFE           225      DELAY2: djnz   R5,DELAY2
01BE  DBFA           226      djnz   R3,delay1
01C0  22             227      ret
                228
01C1  01             229      pola_y: db
00000001b,00000011b,00000010b,00000110b,00000100b,00001100b,00001000b,
00001001b
01C2  03
01C3  02
01C4  06
01C5  04
01C6  0C
01C7  08
01C8  09
01C9  10             230      pola_x: db
00010000b,00110000b,00100000b,01100000b,01000000b,11000000b,10000000b,
10010000b
01CA  30
01CB  20
01CC  60
01CD  40
01CE  C0
01CF  80
01D0  90
                231      end

```



SYMBOL TABLE LISTING

N A M E	T Y P E	V A L U E	A T T R I B U T E S
B . . . . .	D ADDR	00F0H	A
DELAY . . . . .	C ADDR	01B8H	A
DELAY1 . . . . .	C ADDR	01BAH	A
DELAY2 . . . . .	C ADDR	01BCH	A
EA . . . . .	B ADDR	00A8H.7	A
ES . . . . .	B ADDR	00A8H.4	A
GESER_KANAN . . . . .	C ADDR	00DDH	A
GESER_KIRI . . . . .	C ADDR	011AH	A
GESER_MAJU . . . . .	C ADDR	0098H	A
GESER_MUNDUR . . . . .	C ADDR	00BAH	A
HABIS_KIRI . . . . .	C ADDR	0138H	A
HABISKANAN . . . . .	C ADDR	00FBH	A
INISIALISASI_2400	C ADDR	01A8H	A
KIRI_PENUH . . . . .	C ADDR	0158H	A
KIRIMSERIAL . . . . .	C ADDR	019CH	A
MOTOR . . . . .	B ADDR	0090H.0	A
MULAI . . . . .	C ADDR	018EH	A
MUNDUR_PENUH . . . . .	C ADDR	0169H	A
MUNDURSLS . . . . .	C ADDR	0188H	A
OFFMOTOR . . . . .	C ADDR	007CH	A
ONMOTOR . . . . .	C ADDR	0074H	A
P0 . . . . .	D ADDR	0080H	A
P1 . . . . .	D ADDR	0090H	A
P3 . . . . .	D ADDR	00B0H	A
POLA_X . . . . .	C ADDR	01C9H	A
POLA_Y . . . . .	C ADDR	01C1H	A
PORT_STEPPER . . . . .	D ADDR	0080H	A
RI . . . . .	B ADDR	0098H.0	A
SBUF . . . . .	D ADDR	0099H	A
SCON . . . . .	D ADDR	0098H	A
SENSOR_BAHAN . . . . .	B ADDR	00B0H.4	A
SENSOR_X . . . . .	B ADDR	00B0H.2	A
SENSOR_Y . . . . .	B ADDR	00B0H.3	A
SETDELAY . . . . .	NUMB	0032H	A
SISA_KANAN . . . . .	C ADDR	00EEH	A
SISA_KIRI . . . . .	C ADDR	012BH	A
SLSINT . . . . .	C ADDR	0095H	A
SLSKR . . . . .	C ADDR	0163H	A
STEP_KANAN . . . . .	C ADDR	0104H	A
STEP_KIRI . . . . .	C ADDR	0141H	A
STEP_X . . . . .	NUMB	0031H	A
STEP_Y . . . . .	NUMB	0030H	A
TH1 . . . . .	D ADDR	008DH	A
TI . . . . .	B ADDR	0098H.1	A
TMOD . . . . .	D ADDR	0089H	A
TR1 . . . . .	B ADDR	0088H.6	A
UJIAKHIR . . . . .	C ADDR	0034H	A
UJIISI . . . . .	C ADDR	0093H	A
UJIL . . . . .	C ADDR	0062H	A
UJIM . . . . .	C ADDR	006BH	A
UJISD . . . . .	C ADDR	0084H	A



N A M E	T Y P E	V A L U E	A T T R I B U T E S
UJIX. . . . .	C ADDR	0050H	A
UJIXB . . . . .	C ADDR	0059H	A
UJIYB . . . . .	C ADDR	0047H	A
UL_KANAN. . . . .	C ADDR	00E5H	A
UL_KANAN1 . . . . .	C ADDR	00F6H	A
UL_KIRI . . . . .	C ADDR	0122H	A
UL_KIRI1. . . . .	C ADDR	0133H	A
ULANG_KANAN . . . . .	C ADDR	00E7H	A
ULANG_KIRI. . . . .	C ADDR	0124H	A
ULANG_MAJU. . . . .	C ADDR	009AH	A
ULANG_MUNDUR. . . . .	C ADDR	00BCH	A

```

; Petra Fuzzy Development System File
; Developed by Control System Laboratory's Team
; Copyright Control System Laboratory 1998
; File Name = POSFUZZ.ASM
; Created for PetraFuz FIS Kernel Engine Only
; This code is generated for MCS51 Family
; FIS Kernel source code is available for USD 50
; Please contact : petrafuz@peter.petra.ac.id

```

```

;VERSION EQU 1.0
NUMINP EQU 03H
NUMOUT EQU 01H

```

```

                CSEG
                ORG 4100H
INPUT_MFS:     ;Input Membership Functions

```

```

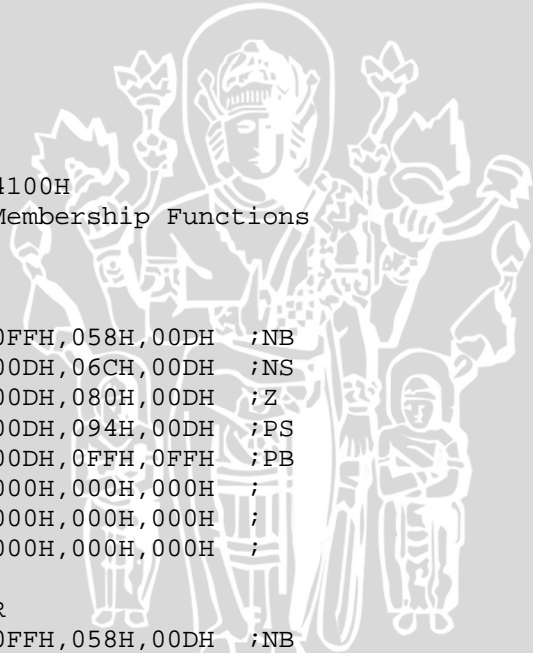
IN1MF:         ;ERROR
                DB 000H,0FFH,058H,00DH ;NB
                DB 058H,00DH,06CH,00DH ;NS
                DB 06CH,00DH,080H,00DH ;Z
                DB 080H,00DH,094H,00DH ;PS
                DB 094H,00DH,0FFH,0FFH ;PB
                DB 000H,000H,000H,000H ;
                DB 000H,000H,000H,000H ;
                DB 000H,000H,000H,000H ;

```

```

IN2MF:         ;d_ERROR
                DB 000H,0FFH,058H,00DH ;NB
                DB 058H,00DH,06CH,00DH ;NS
                DB 06CH,00DH,080H,00DH ;Z
                DB 080H,00DH,094H,00DH ;PS
                DB 094H,00DH,0FFH,0FFH ;PB
                DB 000H,000H,000H,000H ;
                DB 000H,000H,000H,000H ;
                DB 000H,000H,000H,000H ;

```



```
                ORG 41A0H
OUTPUT_MFS:    ;Output Membership Functions
OUT0MF:       ;OUTPUT
              DB 000H                ;NB
              DB 040H                ;NS
              DB 080H                ;Z
              DB 0C0H                ;PS
              DB 0FFH                ;PB
              DB 000H                ;
              DB 000H                ;

                ORG 41B8H
RULE_START:
              DB 000H                ;IF ERROR IS NB
              DB 010H                ;AND d_ERROR IS NB
              DB 080H                ;THEN OUTPUT IS PB

              DB 008H                ;IF ERROR IS NB
              DB 011H                ;AND d_ERROR IS NS
              DB 081H                ;THEN OUTPUT IS PB

              DB 008H                ;IF ERROR IS NB
              DB 012H                ;AND d_ERROR IS Z
              DB 082H                ;THEN OUTPUT IS PB

              DB 008H                ;IF ERROR IS NB
              DB 013H                ;AND d_ERROR IS PS
              DB 082H                ;THEN OUTPUT IS PS

              DB 008H                ;IF ERROR IS NB
              DB 014H                ;AND d_ERROR IS PB
              DB 083H                ;THEN OUTPUT IS Z

              DB 009H                ;IF ERROR IS NS
              DB 010H                ;AND d_ERROR IS NB
              DB 080H                ;THEN OUTPUT IS PB

              DB 009H                ;IF ERROR IS NS
              DB 011H                ;AND d_ERROR IS NS
              DB 080H                ;THEN OUTPUT IS PB

              DB 009H                ;IF ERROR IS NS
              DB 012H                ;AND d_ERROR IS Z
              DB 081H                ;THEN OUTPUT IS PS

              DB 009H                ;IF ERROR IS NS
              DB 013H                ;AND d_ERROR IS PS
              DB 082H                ;THEN OUTPUT IS Z

              DB 009H                ;IF ERROR IS NS
              DB 014H                ;AND d_ERROR IS PB
              DB 082H                ;THEN OUTPUT IS NS

              DB 00AH                ;IF ERROR IS Z
              DB 010H                ;AND d_ERROR IS NB
              DB 080H                ;THEN OUTPUT IS PB
```

```
DB 00AH ;IF ERROR IS Z
DB 011H ;AND d_ERROR IS NS
DB 080H ;THEN OUTPUT IS PS

DB 00AH ;IF ERROR IS Z
DB 012H ;AND d_ERROR IS Z
DB 080H ;THEN OUTPUT IS Z

DB 00AH ;IF ERROR IS Z
DB 013H ;AND d_ERROR IS PS
DB 081H ;THEN OUTPUT IS NS

DB 00AH ;IF ERROR IS Z
DB 014H ;AND d_ERROR IS PB
DB 082H ;THEN OUTPUT IS NB

DB 00BH ;IF ERROR IS PS
DB 010H ;AND d_ERROR IS NB
DB 080H ;THEN OUTPUT IS PS

DB 00BH ;IF ERROR IS PS
DB 011H ;AND d_ERROR IS NS
DB 080H ;THEN OUTPUT IS Z

DB 00BH ;IF ERROR IS PS
DB 012H ;AND d_ERROR IS Z
DB 080H ;THEN OUTPUT IS NS

DB 00BH ;IF ERROR IS PS
DB 013H ;AND d_ERROR IS PS
DB 080H ;THEN OUTPUT IS NB

DB 00BH ;IF ERROR IS PS
DB 014H ;AND d_ERROR IS PB
DB 081H ;THEN OUTPUT IS NB

DB 00CH ;IF ERROR IS PB
DB 010H ;AND d_ERROR IS NB
DB 080H ;THEN OUTPUT IS Z

DB 00CH ;IF ERROR IS PB
DB 011H ;AND d_ERROR IS NS
DB 080H ;THEN OUTPUT IS NS

DB 00CH ;IF ERROR IS PB
DB 012H ;AND d_ERROR IS Z
DB 080H ;THEN OUTPUT IS NB

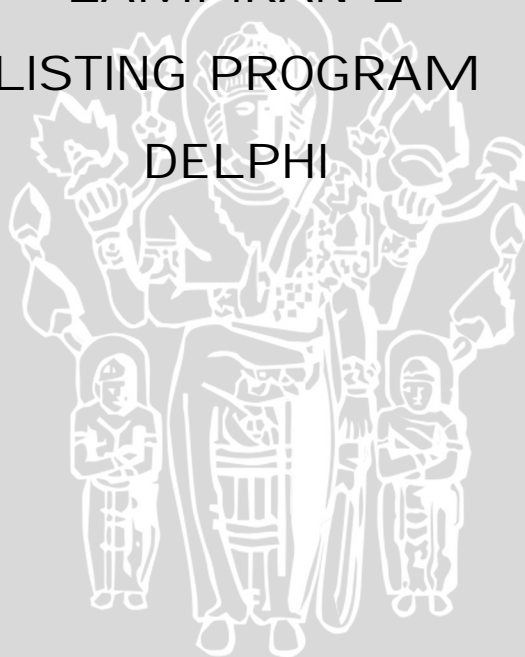
DB 00CH ;IF ERROR IS PB
DB 013H ;AND d_ERROR IS PS
DB 080H ;THEN OUTPUT IS NB

DB 00CH ;IF ERROR IS PB
DB 014H ;AND d_ERROR IS PB
DB 080H ;THEN OUTPUT IS NB
```

```
END_OF_RULE: DB 0FFH
```

UNIVERSITAS BRAWIJAYA

LAMPIRAN 2  
LISTING PROGRAM  
DELPHI



```

Label1: TLabel;
Button1: TButton;
Edit1: TEdit;
Edit2: TEdit;
Button2: TButton;
Button3: TButton;
Button4: TButton;
Button5: TButton;
Edit3: TEdit;
Button6: TButton;
Button7: TButton;
Label2: TLabel;
Button9: TButton;
OpenDialog1: TOpenDialog;
GroupBox1: TGroupBox;
Image1: TImage;
procedure Button1Click(Sender: TObject);
procedure ComPort1RxChar(Sender: TObject; Count: Integer);
procedure Button2Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure Edit2Change(Sender: TObject);
procedure Button5Click(Sender: TObject);
procedure Button6Click(Sender: TObject);
procedure Button7Click(Sender: TObject);
procedure Button8Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure Button9Click(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  XX,YY,i,geserkanan:INTEGER;
  Form1: TForm1;
  warnasbl: tcolor;
  pahat: boolean;
  dataa:string;

implementation

{$R *.DFM}

procedure TForm1.Button1Click(Sender: TObject);
begin
  ComPort1.WriteStr(chr(254)+'x'+chr(strtoint(edit1.text) DIV
100)+chr(strtoint(edit1.text) MOD 100)+chr(255));

```



```
end;

procedure TForm1.ComPort1RxChar(Sender: TObject; Count: Integer);
var dataku:string;
begin
  ComPort1.ReadStr(dataku,count);
  edit2.text:=edit2.text+dataku;
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
  ComPort1.WriteStr(chr(254)+'X'+chr(strtoint(edit1.text) DIV
100)+chr(strtoint(edit1.text) MOD 100)+chr(255));
end;

procedure TForm1.Button3Click(Sender: TObject);
begin
  ComPort1.WriteStr(chr(254)+'y'+chr(strtoint(edit1.text))+chr(255));
end;

procedure TForm1.Button4Click(Sender: TObject);
begin
  ComPort1.WriteStr(chr(254)+'Y'+chr(strtoint(edit1.text))+chr(255));
end;

procedure TForm1.Edit2Change(Sender: TObject);
begin
  if edit2.text='T' then label1.caption:='Perintah Diterima';
  if edit2.text='O' then
  if CheckBox1.Checked
  then timer1.Enabled:=true
  else
  label1.caption:='Perintah Tlah Dilaksanakan';
  edit2.text:='';
end;

procedure TForm1.Button5Click(Sender: TObject);
begin
  ComPort1.WriteStr(chr(254)+'m'+chr(255));
end;

procedure TForm1.Button6Click(Sender: TObject);
begin
  ComPort1.WriteStr(chr(254)+'D'+chr(strtoint(edit3.text))+chr(255));
end;

procedure TForm1.Button7Click(Sender: TObject);
begin
  ComPort1.WriteStr(chr(254)+'l'+chr(255));
```

```
end;

procedure TForm1.Button8Click(Sender: TObject);

begin
  CheckBox1.Checked:=true;
  memo1.text:="";
  for yy:=1 to IMAGE1.height do
  begin
    warnasbl:=IMAGE1.Canvas.Pixels[0,yy];
    for xx:=1 to IMAGE1.Width do
    begin
      if IMAGE1.Canvas.Pixels[xx,yy]<>warnasbl then
      begin
        warnasbl:=IMAGE1.Canvas.Pixels[xx,yy];
        memo1.Lines.Append(inttostr(xx));
      end;
    end;
  end;
  xx:=0;
  yy:=1;
  pahat:=true;
  i:=0;
  timer1.Enabled:=true;
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
  XX:=1;
  YY:=1;
end;

procedure TForm1.Button9Click(Sender: TObject);
begin
  if OpenFileDialog1.Execute then
  begin
    if OpenFileDialog1.FileName<>" then
      image1.Picture.LoadFromFile(OpenDialog1.FileName);
    GroupBox1.Caption:=OpenDialog1.FileName;
  end;
end;

procedure TForm1.Timer1Timer(Sender: TObject);
begin
  Timer1.Enabled:=false;
  ComPort1.WriteStr(chr(254)+'D'+chr(5)+chr(255));
  repeat
```



```

ComPort1.ReadStr(dataa,1);
until dataa='O';
ComPort1.WriteStr(chr(254)+'m'+chr(255));
repeat
  ComPort1.ReadStr(dataa,1);
until dataa='O';
if xx=0 then
begin
  ComPort1.WriteStr(chr(254)+'l'+chr(255));
  repeat
    ComPort1.ReadStr(dataa,1);
until dataa='O';
end;
ComPort1.WriteStr(chr(254)+'D'+chr(20)+chr(255));
repeat
  ComPort1.ReadStr(dataa,1);
until dataa='O';
if pahat then
begin
  ComPort1.WriteStr(chr(254)+'y'+chr(yy)+chr(255));
  repeat
    ComPort1.ReadStr(dataa,1);
until dataa='O';
end
else
begin
  ComPort1.WriteStr(chr(254)+'D'+chr(5)+chr(255));
  repeat
    ComPort1.ReadStr(dataa,1);
until dataa='O';
end;
geserkanan:=strtoint(memo1.Lines.Strings[i])-xx;
xx:=strtoint(memo1.Lines.Strings[i]);
ComPort1.WriteStr(chr(254)+'x'+chr(geserkanan DIV 100)+chr(geserkanan MOD
100)+chr(255));
pahat:=not(pahat);
if xx=image1.Width then
begin
  xx:=0;
  pahat:=true;
  yy:=yy+1;
end;
i:=i+1;
if i=memo1.Lines.Count then CheckBox1.Checked:=false;
end;
end.

```

UNIVERSITAS BRAWIJAYA

LAMPIRAN 3

DATASHEET



# UNIVERSITAS BRAWIJAYA

## LAMPIRAN 4 FOTO ALAT



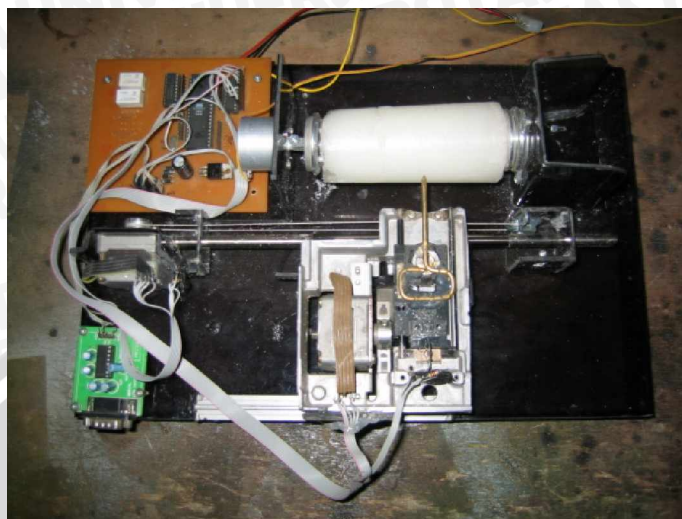


Foto Prototipe Alat Bubut Tampak Atas

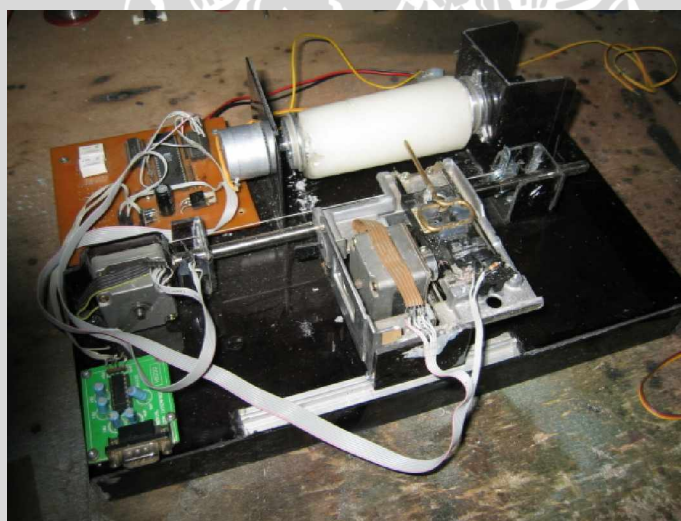


Foto Prototipe Alat Bubut Tampak Samping

