

**PENGENDALIAN KECEPATAN MOTOR DC PADA  
ALAT SABLON SATU WARNA OTOMATIS DENGAN  
KONTROL LOGIKA *FUZZY* MENGGUNAKAN  
MIKROKONTROLER AT89S52**

**SKRIPSI**

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Teknik



Disusun Oleh :  
**KIKI ANDRIANTO**  
0310632024 – 63

**DEPARTEMEN PENDIDIKAN NASIONAL  
UNIVERSITAS BRAWIJAYA  
FAKULTAS TEKNIK  
JURUSAN TEKNIK ELEKTRO  
MALANG  
2007**

**PENGENDALIAN KECEPATAN MOTOR DC PADA  
ALAT SABLON SATU WARNA OTOMATIS DENGAN  
KONTROL LOGIKA *FUZZY* MENGGUNAKAN  
MIKROKONTROLER AT89S52**

**SKRIPSI**

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Teknik



Disusun Oleh :  
**KIKI ANDRIANTO**  
0310632024 – 63

Telah diperiksa dan disetujui  
Dosen Pembimbing

**Pembimbing I**

**Pembimbing II**

**Moch. Rif'an, ST, MT**  
NIP. 132 283 659

**Ir. Erni Yudaningtyas, MT**  
NIP. 131 879 035



**PENGENDALIAN KECEPATAN MOTOR DC PADA  
ALAT SABLON SATU WARNA OTOMATIS DENGAN  
KONTROL LOGIKA *FUZZY* MENGGUNAKAN  
MIKROKONTROLER AT89S52**

Disusun oleh:

**KIKI ANDRIANTO**  
**0310632024 – 63**

Skripsi ini telah diuji dan dinyatakan lulus pada  
tanggal 7 Agustus 2007

**DOSEN PENGUJI**

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

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

**Fitriana Suhartati, ST, MT**  
**NIP. 132 206 527**

**Ir. Retnowati**  
**NIP. 131 124 656**

Mengetahui,  
Ketua Jurusan Teknik Elektro

**Ir. Heru Nurwarsito, M.Kom**  
**NIP. 131 879 033**

## PENGANTAR

Alhamdulillah Robbil ‘Alammin, segala puji bagi Allah SWT yang menguasai alam semesta dan segala ilmu pengetahuan. Hanya atas berkah rahmat dan hidayah-Nya penulis diberikan kesempatan dan kemampuan untuk bisa menyelesaikan penyusunan skripsi ini.

Penyusunan skripsi dengan judul ” **PENGENDALIAN KECEPATAN MOTOR DC PADA ALAT SABLON SATU WARNA OTOMATIS DENGAN KONTROL LOGIKA FUZZY MENGGUNAKAN MIKROKONTROLER AT89S52**” ini merupakan salah satu syarat untuk mendapatkan gelar Sarjana Teknik di Jurusan Teknik Elektro, Fakultas Teknik, Universitas Brawijaya.

Atas bimbingan, petunjuk, dan saran, serta kesempatan yang telah diberikan kepada penulis untuk menyelesaikan penyusunan skripsi ini, penulis dengan segala kerendahan hati menyampaikan rasa terima kasih yang sebesar-besarnya kepada :

1. Bapak Ir. Heru Nurwarsito, M.Kom selaku Ketua Jurusan Teknik Elektro.
2. Bapak Rudy Yuwono, ST, M.Sc selaku Sekretaris Jurusan Teknik Elektro.
3. Bapak Dipl. Ing. Ir. M. Rusli selaku KKDK Teknik Kontrol.
4. Bapak Moch. Rif’an, ST, MT selaku Dosen Pembimbing I.
5. Ibu Ir. Erni Yudaningtyas, MT selaku Dosen Pembing II.
6. Bapak Syamsul Arifin dan Ibu Aslikah kedua orang tuaku yang terkasih yang selalu mendukung dan memberikan semangat, segenap anggota keluargaku tersayang yang juga selalu memberikan dorongan.
7. Almarhum Kakekku M. Arsasi yang nasihatnya senantiasa ada di dalam hatiku.
8. Segenap Dosen, Staf, dan Karyawan di Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya.
9. Teman-teman di Jurusan Teknik Elektro Novian, Yunan, Novianto, Karel, Syahrul, Fathur, Asih, Ita, Marman dan lain-lain yang selalu meluangkan waktu untuk memberikan bantuan.

10. Teman-teman di Joyo Pranoto 601 D Hari, Arifin, Abid, Tejo, Udin, Didik, Aan dan lain-lain yang selalu memberikan keceriaan.

11. Serta semua pihak yang belum disebutkan. Teima kasih atas bantuannya baik yang secara langsung maupun tidak langsung.

Semoga skripsi ini bisa berguna dan bermanfaat bagi pembaca. Meskipun masih banyak kekurangan dalam penyusunan skripsi ini, yang untuk hal ini penulis mohon maaf yang sebesar-besarnya. Kritik dan saran yang membangun tetap penulis harapkan agar bisa memberikan hal yang lebih baik lagi dalam kesempatan lain di masa yang akan datang.

Malang, Agustus 2007

Penulis



## ABSTRAK

Kiki Andrianto, 0310632024, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Brawijaya Malang, *Pengendalian Kecepatan Motor DC Pada Alat Sablon Satu Warna Otomatis Dengan Kontrol Logika Fuzzy Menggunakan Mikrokontroler AT89S52*. Dosen Pembimbing : Moch. Rif'an, ST, MT dan Ir. Erni Yudaningtyas, MT.

---

Pakaian merupakan salah satu kebutuhan pokok manusia. Pakaian selain berfungsi untuk melindungi tubuh juga mempunyai fungsi estetika. Sablon merupakan salah satu cara untuk memberikan fungsi estetika dalam pakaian, sehingga sablon merupakan salah satu komoditas industri kecil dan menengah yang cukup menjanjikan. Industri kecil dan menengah membutuhkan alat sablon otomatis dengan harga yang bisa terjangkau. Dengan mengadaptasi teknologi printer, maka sablon otomatis dengan harga yang terjangkau memungkinkan untuk direalisasikan.

Sablon otomatis yang dirancang masih memiliki keterbatasan kemampuan, yaitu hanya bisa menyablon dalam satu warna. Sablon otomatis dirancang dengan menggunakan mikrokontroler AT89252 sebagai pengatur utama, motor DC dan motor stepper sebagai penggerak, pengatur kecepatan motor DC menggunakan PWM dan sensor obyek gambar menggunakan cahaya LED merah dan photodiode. Untuk memperoleh hasil sablon yang baik diperlukan kestabilan kecepatan pada motor DC penggerak, pengendalian kecepatan motor DC menggunakan kontrol logika *fuzzy*. Setelah perancangan perangkat keras dibangun sebuah perangkat lunak dengan bahasa *assembly* yang mengatur dan mengendalikan perangkat keras agar bisa berfungsi sebagai alat sablon satu warna otomatis dengan pengendalian kecepatan motor DC penggeraknya.

Dari hasil perancangan dilakukan pengujian untuk mengetahui respon transien sistem pengendalian kecepatan motor DC pada alat sablon satu warna otomatis. Dari hasil pengujian pengendalian kecepatan motor DC, diperoleh data-data sebagai berikut : Untuk motor DC tanpa beban dengan setpoint 1300 RPM diperoleh error keadaan mantap (ess) sebesar 1% dan waktu menetap (ts) sebesar  $\pm 2,6$  detik. Untuk motor DC berbeban dengan setpoint 1300 RPM diperoleh error keadaan mantap (ess) sebesar 1% dan waktu menetap (ts) sebesar  $\pm 2,7$  detik. Ketika motor DC diberi gangguan waktu pemulihan selama  $\pm 1,4$  detik. Untuk motor DC tanpa beban dengan perubahan setpoint dari 1300 RPM menjadi 1000 RPM diperoleh error keadaan mantap (ess) sebesar 1,6% dan waktu menetap (ts) sebesar 2,1 detik. Untuk motor DC berbeban dengan perubahan setpoint dari 1300 RPM menjadi 1000 RPM diperoleh error keadaan mantap (ess) sebesar 1,6% dan waktu menetap (ts) sebesar  $\pm 1,9$  detik.

**Kata Kunci** : Sablon, Mikrokontroler AT89S52, Kontrol Logika *Fuzzy*, Motor DC, Setpoint, Error Keadaan Mantap (ess), dan Waktu menetap (ts).

## DAFTAR ISI

<b>HALAMAN JUDUL</b> .....	i
<b>LEMBAR PERSETUJUAN</b> .....	ii
<b>KATA PENGANTAR</b> .....	iv
<b>ABSTRAK</b> .....	vi
<b>DAFTAR ISI</b> .....	vii
<b>DAFTAR GAMBAR</b> .....	ix
<b>DAFTAR TABEL</b> .....	xii
<b>BAB I PENDAHULUAN</b>	
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah .....	3
1.4 Tujuan .....	3
1.5 Sistematika Penulisan .....	4
<b>BAB II DASAR TEORI</b>	
2.1 Sablon .....	5
2.1.1 Sejarah Sablon .....	5
2.1.2 Teknik Sablon .....	5
2.1.3 Pembuatan Stensil pada <i>Screen Sablon</i> .....	6
2.2 Motor DC .....	6
2.2.1 Motor DC Daya Kecil .....	6
2.2.2 Torsi .....	7
2.2.3 Gaya Gerak Listrik (GGL) Induksi .....	8
2.2.4 Motor Servo .....	8
2.2.5 Motor <i>Stepper</i> .....	10
2.2.6 Pengendali Motor DC .....	11
2.3 Mikrokontroler AT89S52 .....	14
2.3.1 Penjelasan Fungsi Kaki-kaki AT89S52 .....	15
2.4 <i>Programmable Peripheral Interface (PPI) 8255</i> .....	18
2.4.1 Struktur PPI 8255 .....	20
2.4.2 Penjelasan Kaki-kaki PPI 8255 .....	20
2.4.3 Pemrograman PPI 8255 .....	22
2.5 <i>Pulse Width Modulation (PWM)</i> .....	23
2.6 Tachogenerator DC .....	24
2.7 ADC 0808/0809 .....	26
2.7.1 Konversi Rasiometri .....	28
2.8 Sistem Kontrol .....	29
2.8.1 Sistem Kontrol Logika <i>Fuzzy</i> .....	30
2.8.2 Logika <i>Fuzzy</i> .....	31
2.8.3 Himpunan <i>Fuzzy</i> .....	31
2.8.4 Operasi Himpunan <i>Fuzzy</i> .....	32
2.8.5 Langkah-langkah dalam Desain Kontrol Logika <i>Fuzzy</i> .....	34
2.8.6 <i>Fuzzification</i> .....	35
2.8.7 Basis Pengetahuan .....	36

2.8.8 Basis Aturan.....	38
2.8.9 Logika Pengambil Keputusan.....	42
2.8.10 <i>Defuzzification</i> .....	44
2.8.11 Respon Transien.....	46
<b>BAB III METODOLOGI</b>	
3.1 Studi Literatur.....	48
3.2 Perancangan.....	49
3.2.1 Pembuatan Alat.....	49
3.3 Pengujian Alat.....	50
3.4 Penarikan Kesimpulan dan Saran.....	50
<b>BAB IV PERANCANGAN</b>	
4.1 Blok Diagram Sistem.....	53
4.2 Perancangan Perangkat Keras.....	56
4.2.1 Perancangan Minimum Sistem AT89S52.....	56
4.2.2 Perancangan Antarmuka PPI 8255.....	57
4.2.3 Perancangan Rangkaian PWM.....	59
4.2.4 Perancangan <i>Driver</i> Motor DC.....	61
4.2.5 Perancangan Rangkaian ADC 0809.....	65
4.2.6 Perancangan <i>Driver</i> Motor <i>Stepper</i> .....	66
4.2.7 Perancangan Sensor Obyek Gambar.....	67
4.2.8 Perancangan Rangkaian <i>Limit Switch</i> .....	68
4.2.9 Perancangan Kontrol Logika <i>Fuzzy</i> .....	69
4.2.9.1 <i>Fuzzifikasi</i> .....	70
4.2.9.2 Basis Aturan.....	71
4.2.9.3 <i>Defuzzifikasi</i> .....	72
4.2.10 Perancangan Perangkat Lunak.....	73
4.2.10.1 Diagram Alir Perangkat Lunak Sistem.....	73
4.2.10.2 Diagram Alir Perangkat Lunak Kontrol Logika <i>Fuzzy</i> .....	75
<b>BAB V PENGUJIAN ALAT DAN ANALISIS</b>	
5.1 Pengujian Sensor Kecepatan Motor.....	77
5.2 Pengujian Mikrokontroler.....	79
5.3 Pengujian PPI 8255.....	80
5.4 Pengujian Rangkaian PWM dan <i>Driver</i> Motor DC.....	81
5.5 Pengujian ADC 0809.....	83
5.6 Pengujian Rangkaian <i>Driver</i> Motor <i>Stepper</i> .....	85
5.7 Pengujian Sensor Obyer Gambar.....	86
5.8 Pengujian <i>Limit Switch</i> .....	87
5.9 Pengujian Sistem Keseluruhan.....	88
<b>BAB VI KESIMPULAN DAN SARAN</b>	
6.1 Kesimpulan.....	99
6.2 Saran.....	100
<b>DAFTAR PUSTAKA</b> .....	102
<b>LAMPIRAN</b> .....	104

DAFTAR GAMBAR

Gambar 2.1	Rangkaian Ekuivalen Motor DC.....	8
Gambar 2.2	Motor Servo.....	9
Gambar 2.3	Prinsip Kerja Motor <i>Stepper</i> .....	11
Gambar 2.4	Jembatan-H.....	12
Gambar 2.5	Konfigurasi Kaki AT89S52.....	15
Gambar 2.6	Blok Diagram AT89S52.....	18
Gambar 2.7	Tiga Macam Mode Operasi PPI 8255.....	19
Gambar 2.8	Blok Diagram PPI 8255.....	20
Gambar 2.9	Konfigurasi Kaki PPI 8255.....	21
Gambar 2.10	Format Pengisian <i>Register Control Word</i> .....	23
Gambar 2.11	Bentuk Gelombang PWM dengan <i>Duty Cycle</i> Berbeda.....	24
Gambar 2.12	Blok Diagram ADC 0808/0809.....	27
Gambar 2.13	Konfigurasi Kaki ADC 0808/0809.....	27
Gambar 2.14	Sistem Kontrol Loop Tertutup.....	30
Gambar 2.15	Definisi Sistem Kontrol Logika <i>Fuzzy</i> .....	30
Gambar 2.16	Perbedaan Himpunan <i>Fuzzy</i> dengan Himpunan Klasik ( <i>Crisp</i> ).....	32
Gambar 2.17	Komplemen <i>Fuzzy</i> .....	32
Gambar 2.18	Irisan <i>Fuzzy</i> .....	33
Gambar 2.19	Gabungan <i>Fuzzy</i> .....	33
Gambar 2.20	Blok Diagram Kontrol Logika <i>Fuzzy</i> Sederhana.....	34
Gambar 2.21	Grafik Nilai Kebenaran.....	36
Gambar 2.22	Bagian Himpunan <i>Fuzzy</i> untuk Titik Prototipe Seimbang.....	37
Gambar 2.23		

(a) Titik Prototipe dan Rentang Jarak (b) Fungsi Keanggotaan ..... 37

Gambar 2.24  
Bentuk Baku dari Kumpulan Aturan..... 38

Gambar 2.25  
Tabel FAM untuk Masukan Tunggal Keluaran Tunggal ..... 42

Gambar 2.26  
Metode Keputusan Max-Min dengan Masukan Pasti ..... 43

Gambar 2.27  
Metode Keputusan Max-Prod dengan Masukan Pasti ..... 44

Gambar 2.28  
*Defuzzification* dengan Metode Max..... 45

Gambar 2.29  
*Defuzzification* dengan Metode Centroid ..... 45

Gambar 2.30  
*Defuzzification* dengan Metode Height ..... 46

Gambar 2.31  
Gambar 2.31 Kurva respon yang menunjukkan  $t_d$ ,  $t_r$ ,  $t_p$ ,  $M_p$  dan  $t_s$ . 47

Gambar 4.1  
Sistem Mekanik Alat ..... 51

Gambar 4.2  
Blok Diagram Sistem ..... 53

Gambar 4.3  
Rancangan Rangkaian Minimum Sistem AT89S52..... 57

Gambar 4.4  
Rancangan Rangkaian PPI 8255 ..... 59

Gambar 4.5  
Rancangan Rangkaian Pembangkit Sinyal PWM..... 45

Gambar 4.6  
Rancangan Rangkaian *Driver* Motor DC ..... 65

Gambar 4.7  
Rancangan Rangkaian ADC 0809 ..... 66

Gambar 4.8  
Rancangan *Driver* Motor *Stepper* ULN 2065..... 67

Gambar 4.9  
Rancangan Sensor Obyek Gambar..... 68

Gambar 4.10  
Rancangan Rangkaian *Limit Switch* ..... 69

Gambar 4.11  
Fungsi Keanggotaan Masukan Error..... 70

Gambar 4.12  
Fungsi Keanggotaan Masukan Perubahan Error..... 70

Gambar 4.13  
Fungsi Keanggotaan Keluaran ..... 71

Gambar 4.14  
Diagram Alir Perangkat Lunak Sistem..... 74

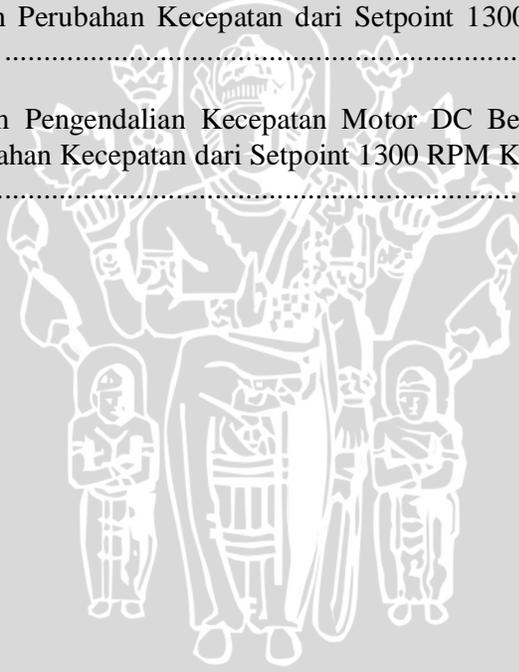
Gambar 4.15  
Diagram Alir Perangkat Lunak Kontrol Logika *Fuzzy* ..... 76

Gambar 5.1	Blok Pengujian Sensor Kecepatan Motor DC .....	78
Gambar 5.2	Blok Pengujian Mikrokontroler AT89S52 .....	79
Gambar 5.3	Blok Pengujian PPI 8255.....	80
Gambar 5.4	Pengujian Rangkaian PWM dan <i>Driver</i> Motor DC .....	82
Gambar 5.5	Blok Pengujian Rangkaian ADC 0809.....	84
Gambar 5.6	Blok Pengujian <i>Driver</i> Motor <i>Stepper</i> .....	85
Gambar 5.7	Blok Pengujian Sensor Obyek Gambar.....	86
Gambar 5.8	Blok Pengujian Rangkaian <i>Limit Switch</i> .....	87
Gambar 5.9	Grafik Keluaran Rata-rata Kecepatan Motor DC Tanpa Beban dengan Setpoint 1300 RPM .....	89
Gambar 5.10	Grafik Keluaran Rata-rata Kecepatan Motor DC Berbeban dengan Setpoint 1300 RPM.....	91
Gambar 5.11	Grafik Keluaran Rata-rata Kecepatan Motor DC dengan Setpoint 1300 RPM Diberi Beban saat $t = 10$ detik.....	92
Gambar 5.12	Grafik Keluaran Rata-rata Kecepatan Motor DC Tanpa Beban dengan Perubahan Kecepatan dari 1300 RPM ke 1000 RPM .....	94
Gambar 5.13	Grafik Keluaran Rata-rata Kecepatan Motor DC Berbeban dan Perubahan Kecepatan dari 1300 RPM ke 1000 RPM .....	97

DAFTAR TABEL

Tabel 2.1	Pemberian Level Logika Mode <i>Full Step</i> .....	13
Tabel 2.2	Pemberian Level Logika Mode <i>Full Step Double Bit</i> .....	13
Tabel 2.3	Pemberian Level Logika Mode <i>Half Step</i> .....	14
Tabel 2.4	Fungsi Alternatif Port 1 AT89S52.....	16
Tabel 2.5	Fungsi alternatif Port 3 AT89S52.....	17
Tabel 2.6	Sistem Operasi PPI 82552.....	2
Tabel 2.8	Jalur Alamat Masukan Sinyal Analog ADC 0808/0809.....	28
Tabel 4.1	Kebenaran 74LS138.....	58
Tabel 4.2	Basis Aturan Pengendalian Kecepatan Motor DC.....	72
Tabel 5.1	Hasil Pengujian Sensor Kecepatan Motor DC.....	78
Tabel 5.2	Hasil Pengujian Mikorkontroler AT89S52.....	79
Tabel 5.3	Hasil Pengujian PPI 8255.....	81
Tabel 5.4	Hasil Pengujian Rangkaian PWM dan Driver Motor DC.....	82
Tabel 5.5	Hasil Pengujian ADC 0809.....	84
Tabel 5.6	Hasil Pengujian Driver Motor <i>Stepper</i> .....	86
Tabel 5.7	Hasil Pengujian Sensor Obyek Gambar.....	87
Tabel 5.8	Hasil Pengujian Rangkaian <i>Limit switch</i> .....	88
Tabel 5.9	Hasil Pengujian Kecepatan Motor DC Tanpa Beban Dengan Setpoint 1300 RPM.....	89
Tabel 5.10	Hasil Pengujian Kecepatan Motor DC Berbeban dengan Setpoint 1300 RPM.....	90
Tabel 5.11	Hasil Pengujian Kecepatan Motor DC dengan Setpoint 1300 RPM Diberi Beban Pada Saat t = 10 detik.....	92

Tabel 5.12	Hasil Pengujian Perubahan Kecepatan Motor DC Tanpa Beban dari Setpoint 1300 RPM Ke 1000 RPM.....	94
Tabel 5.13	Hasil Pengujian Perubahan Kecepatan Motor DC Berbeban dari setpoint 1300 RPM ke 1000 RPM .....	96
Tabel 6.1	Respon Sisem Pengendalian Kecepatan Motor DC Tanpa Beban Dengan Setpoint 1300 RPM .....	99
Tabel 6.2	Respon Sistem Pengendalian Kecepatan Motor DC Berbeban dengan Setpoint 1300 RPM .....	99
Tabel 6.3	Respon Sistem Pengendalian Kecepatan Motor DC dengan Setpoint 1300 RPM Diberi Beban Pada Saat $t = 10$ detik.....	100
Tabel 6.4	Respon Sistem Pengendalian Kecepatan Motor DC Tanpa Beban Dengan Perubahan Kecepatan dari Setpoint 1300 RPM Ke 1000 RPM .....	100
Tabel 6.5	Respon Sistem Pengendalian Kecepatan Motor DC Berbeban Dengan Perubahan Kecepatan dari Setpoint 1300 RPM Ke 1000 RPM.....	100



## BAB I PENDAHULUAN

### 1.1. Latar Belakang

Manusia di kehidupan sehari-hari membutuhkan pakaian yang merupakan salah satu kebutuhan pokoknya. Sebagai kebutuhan pokok tidak hanya dibutuhkan pakaian yang berfungsi sebagai penutup dan pelindung tubuh saja, tetapi pada perkembangannya pakaian juga dibutuhkan dari segi keindahan. Keindahan pakaian bisa diperoleh dari model pakaian dan juga gambar atau tulisan yang terdapat dalam pakaian.

Untuk menambahkan gambar atau tulisan dalam pakaian biasanya digunakan teknik sablon. Sablon biasanya digunakan pada kain sebelum dijahit menjadi pakaian utuh. Dengan sablon maka gambar atau tulisan bisa diubah ubah sesuai dengan keinginan pembuat atau pemesan pakaian yang akan dibuat. Dari kelebihan tersebut maka sablon dalam industri kecil dan menengah merupakan salah satu komoditas yang menjanjikan.

Dalam industri kecil dan menengah biasanya sablon dilakukan secara manual. Langkah pertama yang dilakukan adalah setting desain, yang saat ini bisa dilakukan dengan komputer. Kemudian memindahkan setting desain tersebut ke dalam screen. Setelah itu baru sablon bisa dilakukan pada kain. Karena pada setiap warna membutuhkan setting atau bahkan screen tersendiri, maka terkadang proses sablon menjadi rumit dan lama.

Pengembangan peralatan sablon secara otomatis dengan harga yang terjangkau diperlukan untuk menunjang kelancaran sablon dalam industri kecil dan menengah. Penerapan teknologi memungkinkan otomatisasi sablon bisa diwujudkan. Dengan mengadaptasi teknologi printer yang saat ini umum digunakan sebagai alat pencetak dokumen pada komputer, otomatisasi sablon sangat mungkin untuk dilakukan dengan melakukan penyesuaian terhadap beberapa komponen yang akan digunakan.

Penggunaan motor DC dan motor *stepper* sebagai penggerak pada peralatan sablon otomatis diharapkan bisa mempercepat proses sablon yang akan dilakukan. Mikrokontroler AT89S52 memungkinkan untuk digunakan

sebagai CPU untuk mengkoordinasi dan mengendalikan kecepatan motor DC dalam alat sablon otomatis. Dengan digunakannya mikrokontroler AT89S52 maka koordinasi motor-motor penggerak diharapkan bisa terjalin dengan baik, sehingga proses sablon bisa dilakukan dengan cepat.

Penerapan kontrol logika *fuzzy* untuk pengendalian kecepatan motor DC sebagai penggeraknya diharapkan akan menambah kehandalan alat sablon otomatis. Pengendalian kecepatan motor DC dengan kontrol logika fuzzy dimaksudkan untuk menjaga kestabilan kecepatan motor DC meskipun ketika terjadi gangguan. Gangguan yang mempengaruhi kecepatan motor DC bisa ditimbulkan oleh beberapa hal, antara lain disebabkan oleh gesekan yang terjadi ketika proses penulisan tinta, dan gangguan yang disebabkan oleh komponen mekanik. Ketika proses penulisan tinta, gangguan yang terjadi diakibatkan oleh adanya gaya gesek antara penulis tinta dengan kain untuk media sablon. Besarnya gaya gesek yang terjadi berubah tergantung dari luas permukaan sentuh antara penulis tinta dengan media, sehingga pola gangguan yang terjadi berupa gangguan acak.

## 1.2. Rumusan Masalah

1. Bagaimana merancang dan membuat suatu alat yang bisa digunakan sebagai alat sablon satu warna otomatis dengan pengendalian kecepatan motor DC penggeraknya menggunakan kontrol logika *fuzzy* berbasis mikrokontroler AT89S52 ?
2. Bagaimana merancang dan membuat perangkat fuzzifikasi, basis aturan *fuzzy*, logika pengambilan keputusan, serta perangkat defuzzifikasi yang bisa diterapkan sebagai pengendali kecepatan motor DC pada alat sablon satu warna otomatis ?
3. Bagaimana respon sistem pengendalian kecepatan motor DC ketika tidak diberi beban ?
4. Bagaimana respon sistem pengendalian kecepatan motor DC ketika diberi beban ?
5. Bagaimana respon sistem pengendalian kecepatan motor DC ketika diberi gangguan ?

6. Bagaimana respon sistem pengendalian kecepatan motor DC dengan perubahan kecepatan ketika tidak diberi beban ?
7. Bagaimana respon sistem pengendalian kecepatan motor DC dengan perubahan kecepatan ketika diberi beban ?

### 1.3. Batasan masalah

1. Hanya membahas perancangan dan pembuatan alat sablon satu warna otomatis, kecuali pembahasan mekanik.
2. Hanya membahas perancangan dan pembuatan perangkat keras beserta komponen yang digunakan sebagai penggerak dan pengendali pada alat sablon satu warna otomatis.
3. Hanya membahas perancangan dan pembuatan perangkat lunak pada mikrokontroler AT89S52 sebagai CPU alat sablon satu warna otomatis.
4. Hanya membahas perangkat fuzzifikasi, basis aturan *fuzzy*, logika pengambilan keputusan, serta perangkat fuzzifikasi yang digunakan sebagai pengendali kecepatan motor DC pada alat sablon satu warna otomatis.

### 1.4. Tujuan

1. Merancang dan membuat suatu alat yang bisa digunakan sebagai alat sablon satu warna otomatis dengan pengendalian kecepatan motor DC penggeraknya menggunakan kontrol logika *fuzzy* berbasis mikrokontroler AT89S52.
2. Merancang dan membuat perangkat fuzzifikasi, basis aturan *fuzzy*, logika pengambilan keputusan, serta perangkat defuzzifikasi sebagai pengendali kecepatan motor DC penggerak pada alat sablon satu warna otomatis.
3. Mengetahui respon sistem pengendalian kecepatan motor DC ketika tidak diberi beban.
4. Mengetahui respon sistem pengendalian kecepatan motor DC ketika diberi beban.
5. Mengetahui respon sistem pengendalian kecepatan motor DC ketika diberi gangguan.

6. Mengetahui respon sistem pengendalian kecepatan motor DC dengan perubahan kecepatan ketika tidak diberi beban.
7. Mengetahui respon sistem pengendalian kecepatan motor DC dengan perubahan kecepatan ketika diberi beban

### 1.5. Sistematika Penulisan

Sistematika penulisan skripsi terdiri dari enam bab yang masing-masing bab berisikan :

#### **BAB I      Pendahuluan**

Berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan, dan sistematika penulisan.

#### **BAB II     Tinjauan Pustaka**

Berisi tentang teori-teori yang mendukung dalam perancangan dan pembuatan alat sablon satu warna otomatis.

#### **BAB III    Metodologi**

Berisi tentang metode penelitian, perancangan dan pembuatan alat sablon satu warna otomatis, serta pengujian.

#### **BAB IV    Perencanaan dan Pembuatan Alat**

Berisi tentang perancangan dan pembuatan alat sablon satu warna otomatis. Mulai dari perancangan dan pembuatan perangkat keras. Perancangan dan pembuatan perangkat lunak pada mikrokontroler AT89S52. Perancangan dan pembuatan perangkat fuzzifikasi, basis aturan *fuzzy*, logika pengambilan keputusan, serta perangkat defuzzifikasi sebagai pengendali kecepatan motor DC pada alat sablon satu warna otomatis.

#### **BAB V     Pengujian**

Berisi tentang hasil pengujian terhadap alat sablon satu warna otomatis yang telah direalisasikan.

#### **BAB VI    Kesimpulan dan Saran**

Memuat kesimpulan dan saran-saran.

## BAB II DASAR TEORI

### 2.1 Sablon

#### 2.1.1 Sejarah Sablon

Sablon tradisional merupakan teknik cetak dalam bentuk stensil sederhana, yang secara umum telah digunakan oleh masyarakat Jepang untuk mencetak gambar pada selembar kain yang akan digunakan untuk bahan pakaian. Teknik ini kemudian diadaptasi dan digunakan di Prancis. Sablon modern dipatenkan sekitar tahun 1900 oleh Samuel Simon di Inggris. Kemudian diadopsi oleh John Pilsworth di San Fransisco California pada tahun 1914, yang menggunakan sablon untuk membuat hasil cetakan dengan banyak warna. Seperti sebagaimana sablon yang ada sekarang ini (Meggs, 1998).

#### 2.1.2 Teknik Sablon

Dalam prosesnya sablon menggunakan *screen* yang terbuat dari lembaran berpori, dengan meregangkan selembar kain tipis pada sebuah bingkai. Kain yang digunakan pada awalnya adalah sutra, tapi sejak sekitar 1940 telah digantikan dengan polyester atau nylon. Sebagian area *screen* yang ditutup dengan bahan yang tidak tembus air (stensil) merupakan bagian positif dari gambar yang akan dicetak. Gambar akan dihasilkan dari tinta yang menembus pada bagian yang terbuka (Saunders dan Miles, 2006).

Proses mencetak dilakukan dengan meletakkan *screen* di atas selembar kertas atau kain kering. Tinta dituangkan di atas *screen*, kemudian ditekan dan diratakan pada seluruh permukaan *screen* yang terbuka dengan menggunakan rakel. Tinta akan menembus ruang terbuka dari *screen* menuju kertas atau kain yang ada di bawahnya. Setelah itu *screen* diangkat, tinta yang tidak menembus bagian *screen* yang terbuka akan “membanjiri” *screen* tersebut. *Screen* bisa digunakan kembali setelah dibersihkan. Bila tinta yang digunakan pada sebuah lembaran yang sama lebih dari satu warna, tinta dibiarkan mengering terlebih dahulu dan kemudian proses yang sama dilakukan dengan menggunakan *screen* dan tinta yang lain (Saunders dan Miles, 2006).

### 2.1.3 Pembuatan Stensil pada *Screen* Sablon

Ada beberapa cara yang digunakan untuk membuat stensil pada *screen* sablon. Cara paling sederhana adalah dengan membentuk bagian yang diinginkan, bisa dengan cara memotong lembaran kertas atau plastik dan menempelkannya pada *screen*, atau dengan menggambar secara langsung pada *screen* dengan bahan yang tidak tembus air ketika mengering. Untuk teknik yang bersifat lukisan secara langsung, seniman bisa memilih untuk menggambar dengan tinta gambar, setelah kering dilanjutkan dengan melapisi seluruh permukaan *screen* dengan pengisi *screen*. Ketika pengisi *screen* telah kering penyemprot digunakan untuk menyemprotkan air ke seluruh permukaan *screen*, sehingga area yang telah digambar dengan tinta air akan meluruh dan meninggalkan stensil di sekitarnya. Proses ini memungkinkan seniman untuk terlibat langsung dalam pembuatan *screen*, sehingga bisa menjaga keaslian karyanya (Saunders dan Miles, 2006).

Metode yang sekarang umum digunakan saat ini adalah dengan cara emulsi foto. Cara ini dilakukan dengan menempatkan gambar pada lapisan transparan. Gambar bisa dilakukan secara langsung pada lapisan, hasil *fotocopy*, atau hasil *printer* laser, asalkan hasilnya tidak tembus cahaya. Negatif hitam putih bisa digunakan, tetapi tidak seperti cara tradisional cara ini memerlukan metode bukaan menggunakan film positif. Lapisan yang telah dibuat diletakkan pada *screen* yang telah diberi lapisan pengemulsi, dan diberikan pencahayaan yang kuat. Area yang tertembus cahaya pada lapisan ini akan melewati cahaya menuju lapisan pengemulsi, yang kemudian mengeras dan lengket pada *screen*. Langkah selanjutnya adalah mencuci permukaan *screen*, area lapisan pengemulsi yang tidak terkena cahaya akan meluruh, sehingga membentuk stensil negatif yang menempel pada *screen* (Saunders dan Miles, 2006).

## 2.2 Motor DC

### 2.2.1 Motor DC Daya Kecil

Motor DC merupakan suatu mesin yang berfungsi sebagai pengubah tenaga listrik arus searah (DC) menjadi tenaga gerak atau mekanik. Tenaga gerak yang dihasilkan berupa putaran yang ada pada rotor (Zuhal, 1992:81).

Motor DC daya kecil menggunakan daya kurang dari satu tenaga kuda. Motor jenis ini kebanyakan digunakan sebagai motor servo dalam berbagai sistem kendali. Untuk motor servo dalam sistem-sistem pengaturan kecepatan, maupun sebagai penggerak macam-macam alat (Zuhul, 1992:81).

Motor DC daya kecil menggunakan magnet permanen untuk membangkitkan medan magnet statornya. Dengan demikian untuk membangkitkan medan magnet tidak perlu disediakan suatu daya. Selain itu, medan magnet yang ditimbulkan tidak terpengaruh oleh reaksi jangkar dari arus jangkar (Zuhul, 1992:81).

### 2.2.2 Torsi

Torsi disebabkan karena adanya arus ( $I$ ) bersama medan magnet ( $\phi$ ), yang sesuai dengan rumus (Zuhul, 1992:81).

$$T = C_a I_a \phi \quad (2-1)$$

Keterangan :

T = Torsi

$I_a$  = Arus jangkar(Ampere)

$\phi$  = Fluks medan magnet

$C_a$  = Konstanta, besar dari konstanta didapatkan dari (Zuhul, 1992:82).

$$C_a = \left(\frac{P}{a}\right) \times \left(\frac{Z}{60}\right) \quad (2-2)$$

Keterangan :

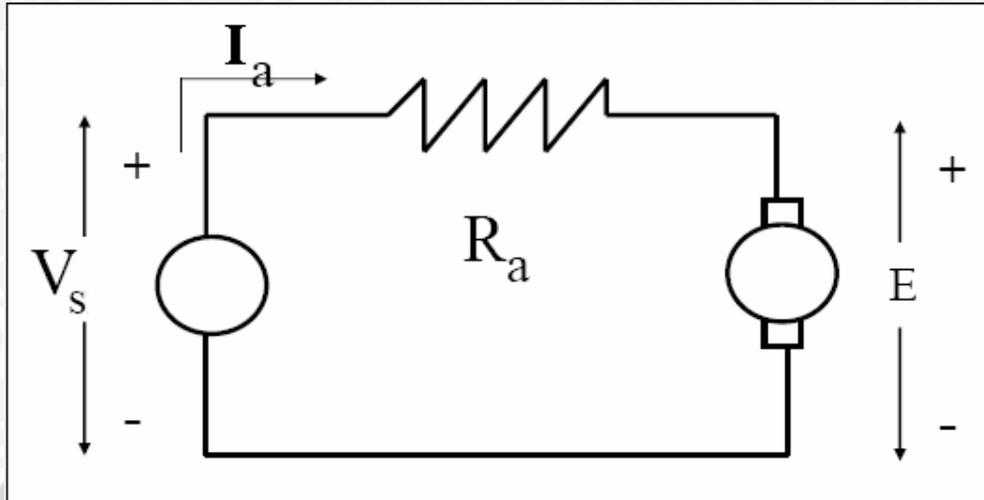
P = Jumlah kutub

a = Jalur paralel konduktor jangkar

Z = Jumlah konduktor jangkar

Karena menggunakan medan magnet permanen maka besarnya fluks medan magnet berharga konstan.

### 2.2.3 Gaya Gerak Listrik (GGL) Induksi



**Gambar 2.1 Rangkaian Ekuivalen Motor DC.**

(Sumber : Zuhail,1992:82)

Untuk sebuah motor DC daya kecil juga berlaku rumus (Zuhail, 1992:82)

$$V_s = E + R_a I_a \quad (2-3)$$

Keterangan :

$V_s$  = Tegangan sumber (Volt)

$E$  = Gaya gerak listrik induksi

$R_a$  = Tahanan dalam jangkar

$I_a$  = Arus jangkar

Dengan besar gaya gerak listrik induksi pada motor DC daya kecil dengan magnet permanen diperoleh dari rumus (Zuhail, 1992:83)

$$E = C_i n \phi \quad (2-4)$$

Keterangan :

$E$  = Gaya gerak listrik induksi (Volt).

$\phi$  = Fluks medan dengan harga konstan

$n$  = Putaran (rpm)

$C_i$  = Konstanta Induksi

### 2.2.4 Motor Servo

Motor servo pada awalnya didesain secara khusus untuk penguat servo.

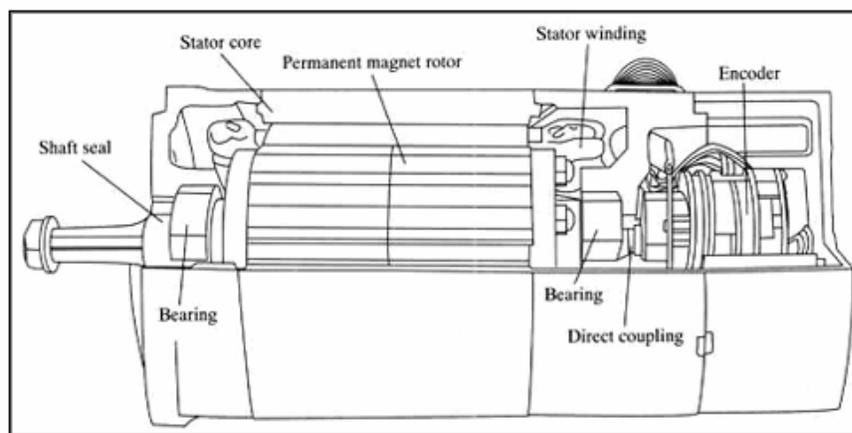
Tapi, sekarang motor servo didesain untuk aplikasi yang bisa menggunakan

penguat servo atau pengendali variabel frekuensi. Hal ini berarti motor servo bisa menggunakan sistem servo dalam suatu aplikasi, dan bisa menggunakan pengendali variabel frekuensi di aplikasi yang lain. Bahkan telah diterapkan sistem loop tertutup yang tidak perlu menggunakan motor stepper lagi dalam sebuah sistem servo, sehingga memungkinkan sebuah motor induksi AC yang dihubungkan dengan pengendali kecepatan juga dinamakan sebagai motor servo.

Beberapa perubahan perlu dilakukan agar bisa membuat sembarang motor untuk didesain sebagai motor servo. Beberapa hal di antaranya adalah (Kissel, 2000:1) :

1. Menjadikan kemampuan motor untuk beroperasi pada beberapa range kecepatan tanpa terjadi overheating.
2. Menjadikan kemampuan motor untuk beroperasi pada kecepatan nol dan menahan torsi yang mencukupi untuk menahan beban tetap pada posisi yang diinginkan.
3. Menjadikan kemampuan motor untuk beroperasi pada kecepatan yang sangat rendah untuk periode yang lama tanpa mengalami overheating.

Jenis motor yang paling sering digunakan dalam sebuah sistem servo adalah motor dengan magnet permanen. Tegangan yang digunakan dalam kumparannya bisa berupa tegangan AC atau tegangan DC. Sebuah motor servo dengan magnet permanen memiliki sebuah penyandi (encoder) dan tachometer yang disertakan dalam kerangkanya. Hal ini memungkinkan motor servo akan menunjukkan posisi atau kecepatan batang rotornya dengan akurasi yang tinggi (Kissel, 2000:2).



**Gambar 2.2 Motor Servo.**  
(Sumber : Kissel, 2000:2)

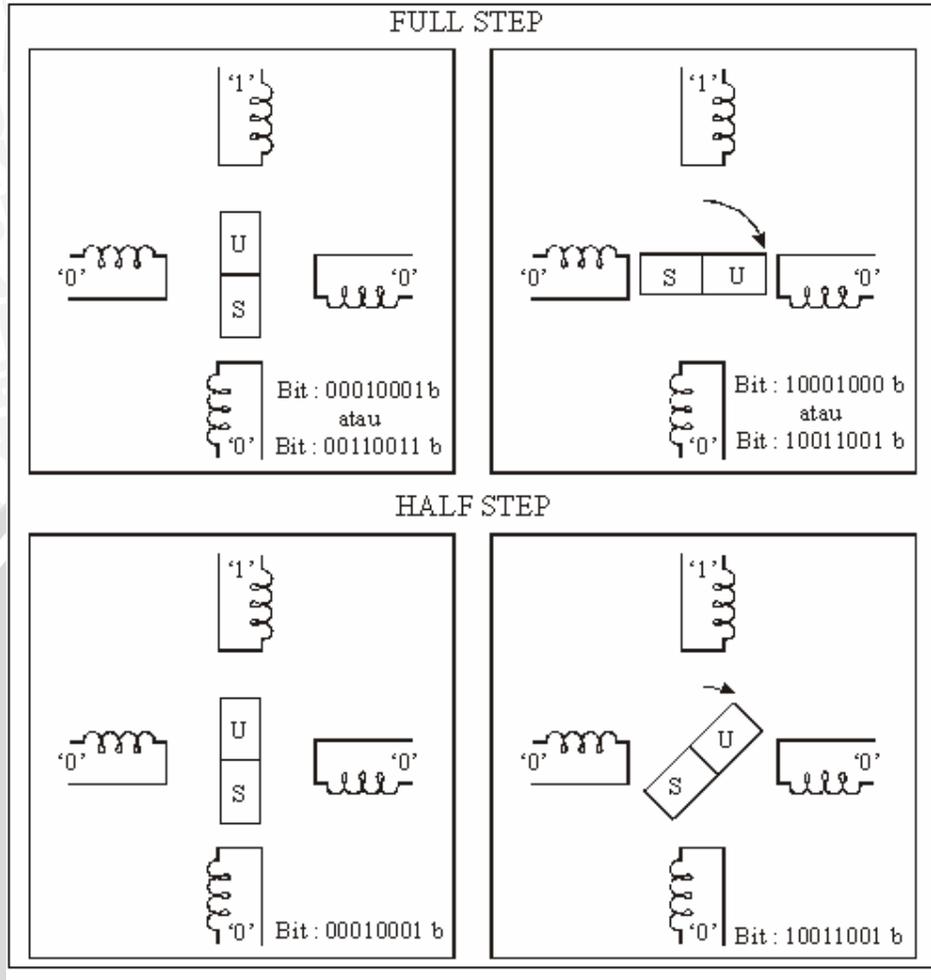
### 2.2.5 Motor Stepper

Putaran pada motor yang terjadi umumnya memanfaatkan adanya perbedaan kutub medan magnet antara rotor dengan stator. Perbedaan medan magnet yang terjadi memberikan gaya tarik maupun gaya tolak yang kemudian membuat rotor berputar terhadap stator. Karena besarnya medan magnet yang dihasilkan oleh lilitan pada stator terpengaruh oleh jumlah lilitan dan besar tegangan yang mengalir, menyebabkan medan magnet tersebut besarnya tidak pasti. Dengan demikian kecepatan, kekuatan, dan posisi putaran yang dihasilkan juga tidak terkendali (Pitowarno, 1992:26).

Motor *stepper* merupakan motor yang gerakan putarannya cukup terkendali, sehingga bisa digunakan untuk mendapatkan posisi serta jumlah putaran yang tepat. Mudahnya pengendalian putaran motor *stepper* karena putarannya hanya memanfaatkan daya tarik kutub magnet yang berlawanan. Kutub medan magnet yang ditimbulkan oleh lilitan stator berbeda dengan kutub pada rotor, sehingga rotor akan tertarik dan kemudian terhenti sedekat mungkin ke lilitan stator yang menghasilkan medan magnet (Pitowarno, 1992:27).

Pemberian logika 1 atau pemberian tegangan yang sesuai dengan kebutuhan motor *stepper* pada lilitan stator, akan menimbulkan medan magnet yang berlawanan dengan kutub medan magnet rotor. Perbedaan medan magnet antara lilitan stator dan rotor tersebut, akan menyebabkan rotor bergerak menuju lilitan stator. Kemudian bila pemberian tegangan dihentikan (diberi logika 0) dan pemberian logika 1 digantikan oleh lilitan di sebelahnya, maka rotor akan bergerak ke lilitan berikutnya. Pergerakan rotor yang demikian ini dinamakan *step*. Dengan adanya *step* yang berkelanjutan, putaran pada motor *stepper* bisa terjadi (Pitowarno, 1992:27).

Pengaturan logika 0 atau 1 yang diberikan pada semua lilitan stator yang ada secara serempak memudahkan pengaturan putaran motor *stepper*. Pengaturan logika dilakukan pada mikrokontroler dalam bentuk data. Kemudian data yang berisi informasi pemberian logika pengaturan putaran dikirim ke rangkaian *interface* motor *stepper*. Pada akhirnya motor *stepper* berputar sesuai dengan kendali yang diberikan dari logika pengaturan putaran (Pitowarno, 1992:27).



Gambar 2.3 Prinsip Kerja Motor Stepper.  
(Sumber : Pitowarno, 1992:28)

### 2.2.6 Pengendali Motor DC

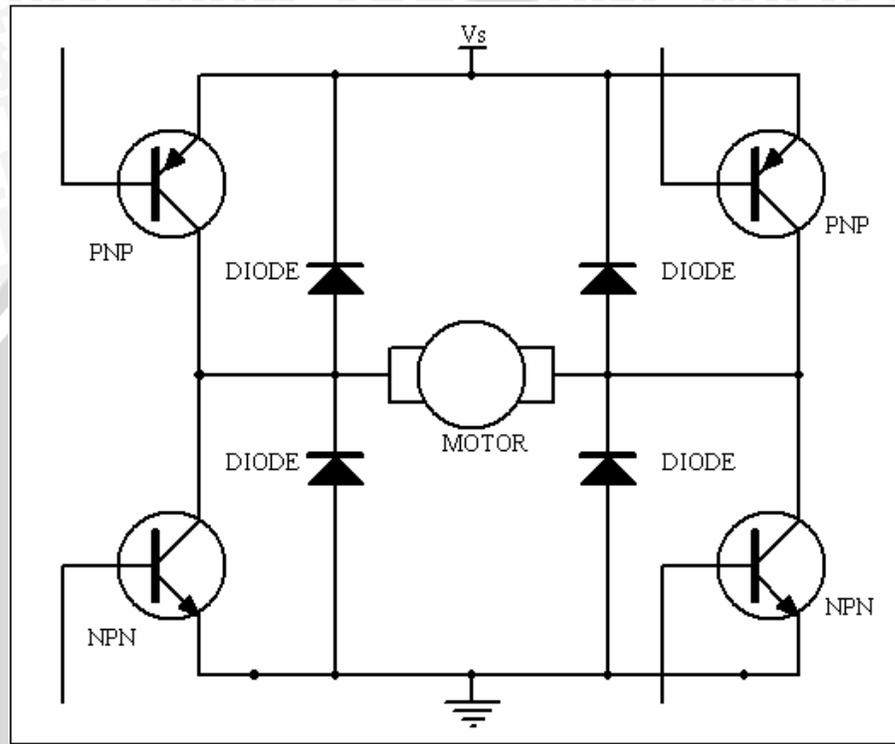
Pengendali motor bisa dibedakan berdasarkan jenis motor yang akan dikendalikan seperti magnet permanen, servo, seri, penguatan terpisah dan AC. Pengendali motor dihubungkan pada sebuah catu daya tersendiri dan rangkaian pengendalinya bisa berupa rangkaian analog atau rangkaian digital.

Beberapa jenis pengendali motor adalah sebagai berikut (Kamal, 2007:2):

#### 1. Jembatan-H

Motor DC umumnya dikendalikan dengan menggunakan konfigurasi transistor yang dinamakan “Jembatan-H”. Yang terdiri dari minimal 4 buah saklar mekanik atau saklar elektronis, seperti 2 buah transistor NPN dan 2 buah transistor PNP. Sebuah transistor NPN dan sebuah transistor PNP diaktifkan

secara bersamaan. Kedua transistor NPN atau transistor PNP bisa diaktifkan untuk membuat hubung singkat di antara terminal motor, yang bermanfaat untuk memperlambat motor dari medan magnet balik yang ditimbulkan (Kamal, 2007:5).



**Gambar 2.4 Jembatan-H.**  
(Sumber : Kamal, 2007:6)

## 2. Pengendali Servo

Umumnya pengendali servo menggunakan Pulse-Width Modulation (PWM). Lama waktu pulsa tinggi (umumnya antara 1 dan 2 mili detik) menentukan di mana motor akan memposisikan batang rotor atau mengatur kecepatan batang rotor (Adler dan Surtell, 2000:1).

Sebuah motor servo dengan magnet permanen memiliki sebuah penyandi (encoder) dan tachometer yang disertakan dalam kerangkanya. Hal ini memungkinkan motor servo akan menunjukkan posisi atau kecepatan batang rotornya dengan akurasi yang tinggi (Kissel, 2000:2).

## 3. Pengendali Motor Stepper

Pengendali motor stepper biasanya dilakukan dengan pengaturan pemberian logika pada motor *stepper* bisa dilakukan dengan 2 cara, yaitu mode

full step dan mode half step. Tabel 2.1, Tabel 2.2, dan Tabel 2.3 mengilustrasikan pemberian tegangan mode full step dan half step.

Pada prinsip kerja full step kutub magnet pada rotor akan mendapat tarikan dari lilitan stator yang sama setelah melalui 4 step. Pemberian logika 1 akan mulai kembali dari lilitan stator awal, mengulangi step 1. Besarnya perpindahan yang akan ditempuh putaran rotor setiap step adalah 1/4 dari 360° dibagi jumlah kutub magnet pada rotor. Untuk prinsip kerja half step, step 1 akan diulangi setelah melalui 8 step, sehingga besarnya perpindahan adalah 1/8 dari 360° dibagi jumlah kutub pada rotor. Jadi untuk mendapatkan posisi putaran motor stepper yang lebih akurat bisa didapatkan dengan penambahan jumlah kutub magnet pada rotor serta didukung dengan lilitan stator yang berpasangan sesuai dengan posisi kutub magnet rotor (Pitowarno, 1992:28).

**Tabel 2.1 Pemberian Level Logika Mode Full Step.**

FULL STEP								
Level Logika yang Diberikan Pada Lilitan								
Step	Arah putar searah jarum jam				Arah putar melawan jarum jam			
	L3	L2	L1	L0	L3	L2	L1	L0
1	1	0	0	0	0	0	0	1
2	0	1	0	0	0	0	1	0
3	0	0	1	0	0	1	0	0
4	0	0	0	1	1	0	0	0

Sumber : Pitowarno, 1992: 29.

**Tabel 2.2 Pemberian Level Logika Mode Full Step Double Bit.**

FULL STEP DOUBLE BIT								
Level Logika yang Diberikan Pada Lilitan								
Step	Arah putar searah jarum jam				Arah putar melawan jarum jam			
	L3	L2	L1	L0	L3	L2	L1	L0
1	1	1	0	0	0	0	1	1
2	0	1	1	0	0	1	1	0
3	0	0	1	1	1	1	0	0
4	1	0	0	1	1	0	0	1

Sumber : Pitowarno, 1992: 29.

Tabel 2.3 Pemberian Level Logika Mode *Half Step*.

HALF STEP								
Level Logika yang Diberikan Pada Lilitan								
Step	Arah putar searah jarum jam				Arah putar melawan jarum jam			
	L3	L2	L1	L0	L3	L2	L1	L0
1	1	0	0	0	0	0	0	1
2	1	1	0	0	0	0	1	1
3	0	1	0	0	0	0	1	0
4	0	1	1	0	0	1	1	0
5	0	0	1	0	0	1	0	0
6	0	0	1	1	1	1	0	0
7	0	0	0	1	1	0	0	0
8	1	0	0	1	1	0	0	1

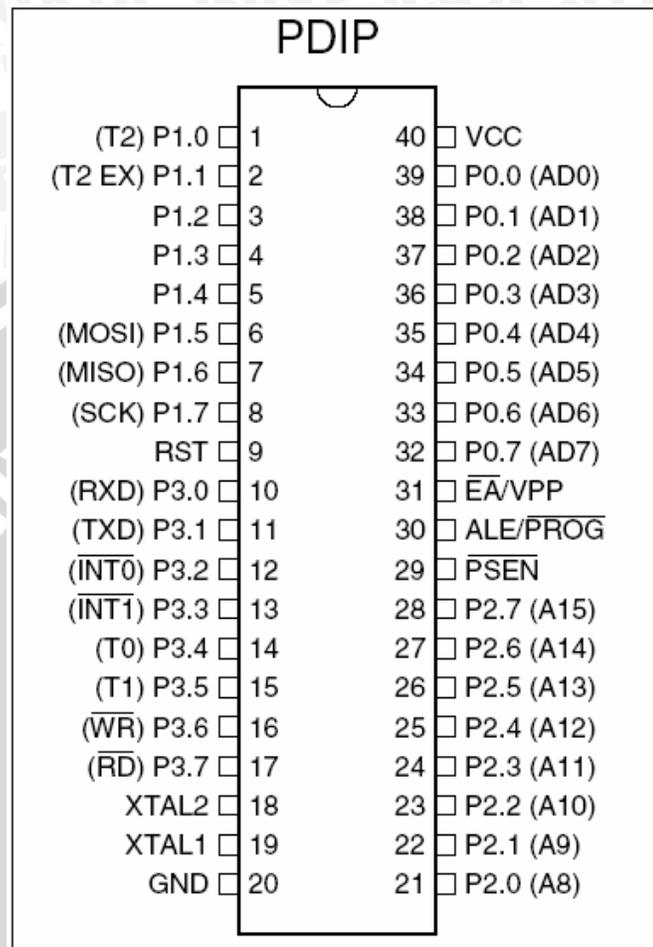
Sumber : Pitowarno, 1992:30.

### 2.3 Mikrokontroler AT89S52

AT89S52 merupakan mikrokontroler CMOS 8-bit berdaya rendah yang berkemampuan tinggi, dengan *flash memory* berkapasitas 8 Kbyte yang bisa diprogram secara *In System Programming* (ISP). Mikrokontroler ini dibuat menggunakan teknologi *high-density nonvolatile memory* yang cocok dengan industri standar 80C51 baik dari segi instruksi set maupun konfigurasi kaki. Kepingan flash yang terdapat di dalamnya memungkinkan program memori untuk diprogram ulang secara langsung melalui *In System Programming* atau diprogram secara biasa dengan programmer *nonvolatile memory*. Kombinasi dari CPU 8-bit serbaguna dengan *flash memory* yang bisa diprogram dalam sebuah kepingan tunggal, menjadikan AT89S52 sebagai mikrokontroler yang sangat kuat yang bisa memberikan kemudahan penyesuaian untuk berbagai kebutuhan serta efektifitas harga yang bisa dimanfaatkan sebagai solusi bagi banyak dan bermacam masalah aplikasi di bidang kontrol (Atmel Corporation, 2001:1).

AT89S52 mempunyai ciri-ciri standar berupa : 8 Kbyte flash, 256 byte RAM, 32 pin I/O (4 buah port I/O 8 bit), *watchdog timer*, 2 data pointer, 3 buah *timer/counter* 16 bit, dan sebuah 6 vektor 2 level arsitektur interupsi, sebuah serial port *full duplex*, kepingan oscillator, dan rangkaian clock. Sebagai tambahan, AT89S52 didesain dengan logika statis untuk operasi pada frekuensi turun sampai nol dan mendukung 2 perangkat lunak untuk mode *power saving* yang bisa dipilih. *Idle Mode* menghentikan kerja CPU sementara tetap mengijinkan RAM, timer/counter, port serial, dan sistem interupsi untuk tetap berfungsi. *Power-down Mode* menyimpan isi RAM tetapi menghentikan oscillator, dan menghentikan

fungsi kepingan lain sampai interupsi selanjutnya atau reset perangkat keras (Atmel Corporation, 2001:1).



**Gambar 2.5 Konfigurasi Kaki AT89S52.**  
(Sumber : Atmel Corporation, 2001:2)

### 2.3.1 Penjelasan Fungsi Kaki-kaki AT89S52

Fungsi dari kaki-kaki AT89S52 bisa dijelaskan sebagai berikut (Atmel Corporation, 2001:4-5) :

1. VCC (Kaki 40)

Berfungsi sebagai catu daya, sumber tegangan 5V.

2. GND (Kaki 20)

Berfungsi sebagai ground.

### 3. Port 0 (Kaki 32 - Kaki 39)

Port 0 merupakan port dua fungsi yang berada pada kaki 32-39. Port ini digunakan sebagai port I/O biasa serta dapat digunakan untuk memultiplek alamat dan data pada waktu mengakses memori eksternal.

### 4. Port 1 (Kaki 1 - Kaki 8)

Port 1 berfungsi sebagai port I/O biasa dan berada pada kaki 1-8. Port 1 juga berfungsi sebagai penerima byte alamat orde rendah ketika pemrograman flash dan pengujian.

**Tabel 2.4 Fungsi Alternatif Port 1 AT89S52.**

Port Pin	Alternate Functions
P1.0	T2 (external count input to Timer/Counter 2), clock-out
P1.1	T2EX (Timer/Counter 2 capture/reload trigger and direction control)
P1.5	MOSI (used for In-System Programming)
P1.6	MISO (used for In-System Programming)
P1.7	SCK (used for In-System Programming)

Sumber : Atmel Corporation, 2001:4

### 5. Port 2 (Kaki 21 – Kaki 28)

Port 2 merupakan port dua fungsi yaitu sebagai I/O serbaguna, atau sebagai bus alamat byte tinggi untuk rancangan yang melibatkan memori eksternal.

### 6. Port 3 (Kaki 10 – Kaki 17)

Port 3 adalah port dua fungsi yang berada pada kaki 10-17, port ini memiliki multi fungsi, seperti yang terlihat dalam Tabel 2.5.

Tabel 2.5 Fungsi alternatif Port 3 AT89S52.

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{\text{INT0}}$ (external interrupt 0)
P3.3	$\overline{\text{INT1}}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	$\overline{\text{WR}}$ (external data memory write strobe)
P3.7	$\overline{\text{RD}}$ (external data memory read strobe)

Sumber : Atmel Corporation, 2001:4

#### 7. RST (Kaki 9)

Ketika pulsa transisi dari rendah ke tinggi yang diumpankan ke kaki RST akan mereset AT89C51. Kaki ini dihubungkan dengan rangkaian *power on reset*.

#### 8. ALE/ $\overline{\text{PROG}}$ (Kaki 30)

ALE (*Address Latch Enable*) merupakan pulsa keluaran untuk menahan byte rendah pada alamat ketika mengakses memori eksternal.  $\overline{\text{PROG}}$  sebagai pulsa masukan program ketika memrogram flash.

#### 9. $\overline{\text{PSEN}}$ (Kaki 29)

$\overline{\text{PSEN}}$  (*Program Store Enable*), fungsinya adalah sebagai sinyal kontrol untuk memungkinkan mikrokontroler membaca program (*code*) dari memori eksternal.

#### 10. $\overline{\text{EA}}$ /VPP (Kaki 31)

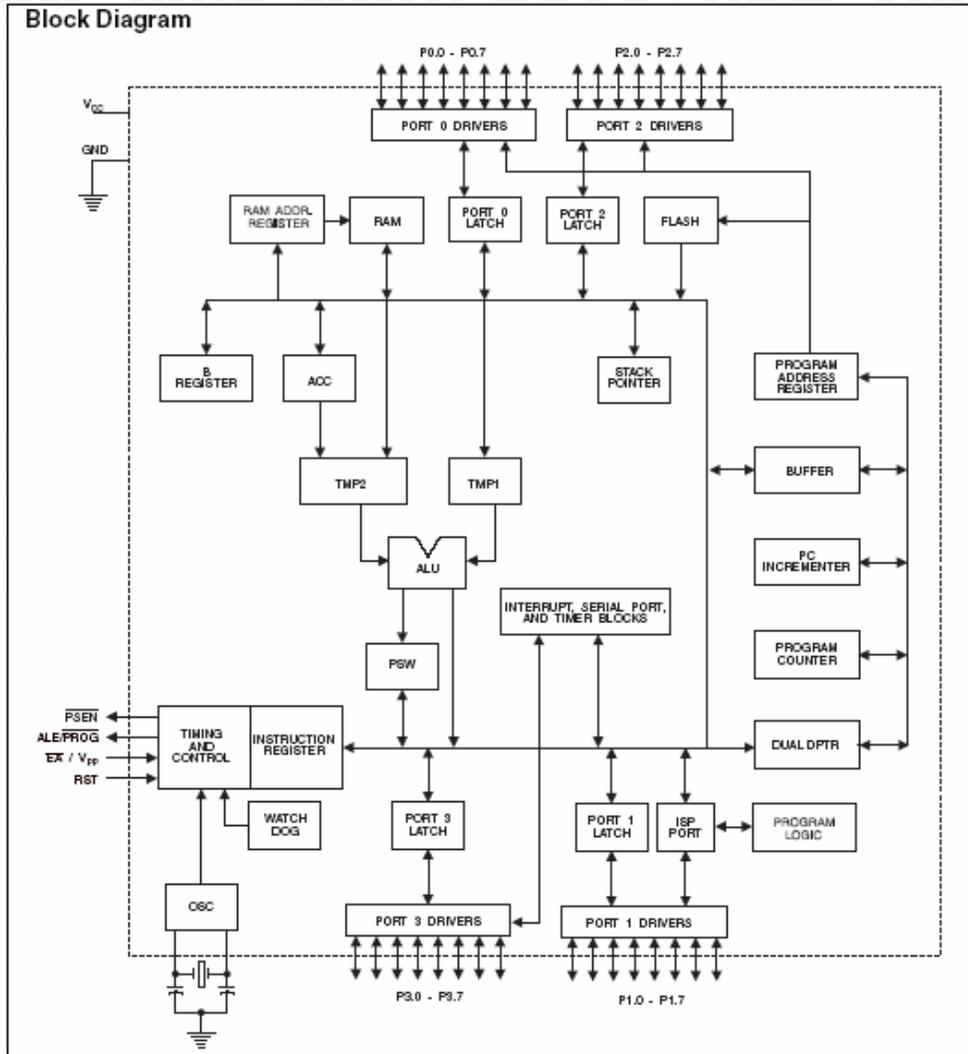
$\overline{\text{EA}}$  (*External Acces Enable*), bila diberikan logika tinggi, maka mikrokontroler akan melaksanakan instruksi dari ROM/EPROM. Bila diberikan logika rendah, mikrokontroler akan melaksanakan instruksi dari memori program luar.

#### 11. XTAL1 (Kaki 19)

Berfungsi sebagai masukan untuk oscillator dengan penguatan terbalik dan masukan untuk rangkaian operasi *clock internal*.

#### 12. XTAL2 (Kaki 18)

Keluaran dari oscillator dengan penguatan terbalik.



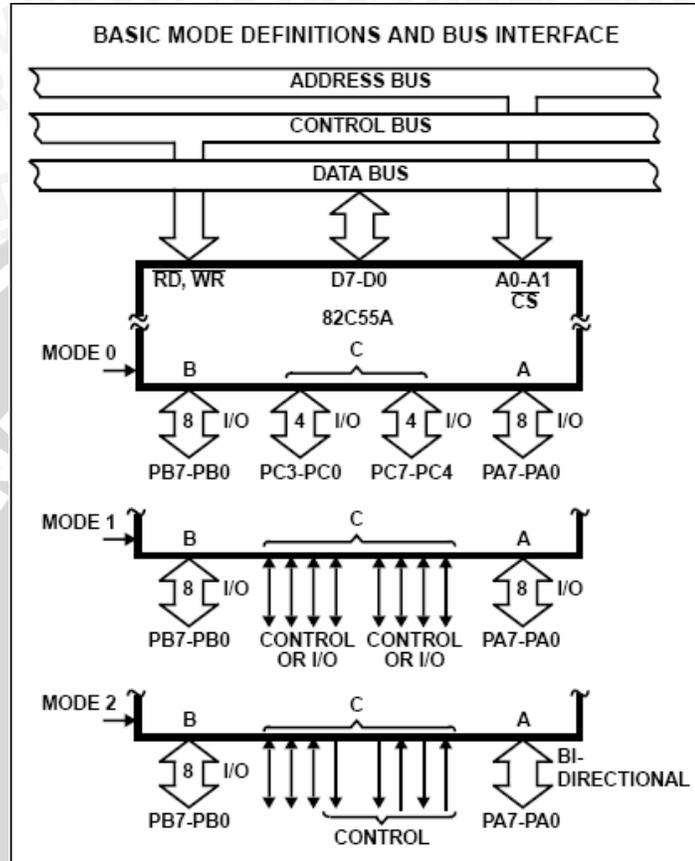
Gambar 2.6 Blok Diagram AT89S52.  
(Sumber : Atmel Corporation, 2003:3)

## 2.4 Programmable Peripheral Interface (PPI) 8255

Pelaksanaan pemrograman saluran I / O hanya bisa dilakukan dalam 3 mode, yaitu :

1. Mode 0 akan membentuk 3 port (A, B dan C) sebagai saluran I / O. Untuk port C terbagi dalam 2 kelompok saluran 4 bit, yaitu port C upper dan Port C lower.
2. Mode 1 akan membentuk 2 kelompok 12 bit saluran dengan 4 bit saluran control data dan 8 bit saluran I / O. Dengan demikian memungkinkan terjadinya pemindahan data I / O dari atau ke port tertentu dengan adanya sinyal-sinyal strobed (handshaking).

3. Mode 2 akan membentuk *bi-directional bus* dengan 8 bit saluran *handshake*. Keberadaan konfigurasi mode ini memungkinkan terjadinya komunikasi dua arah antara perangkat luar dengan prosesor (Putra, 2002:100).



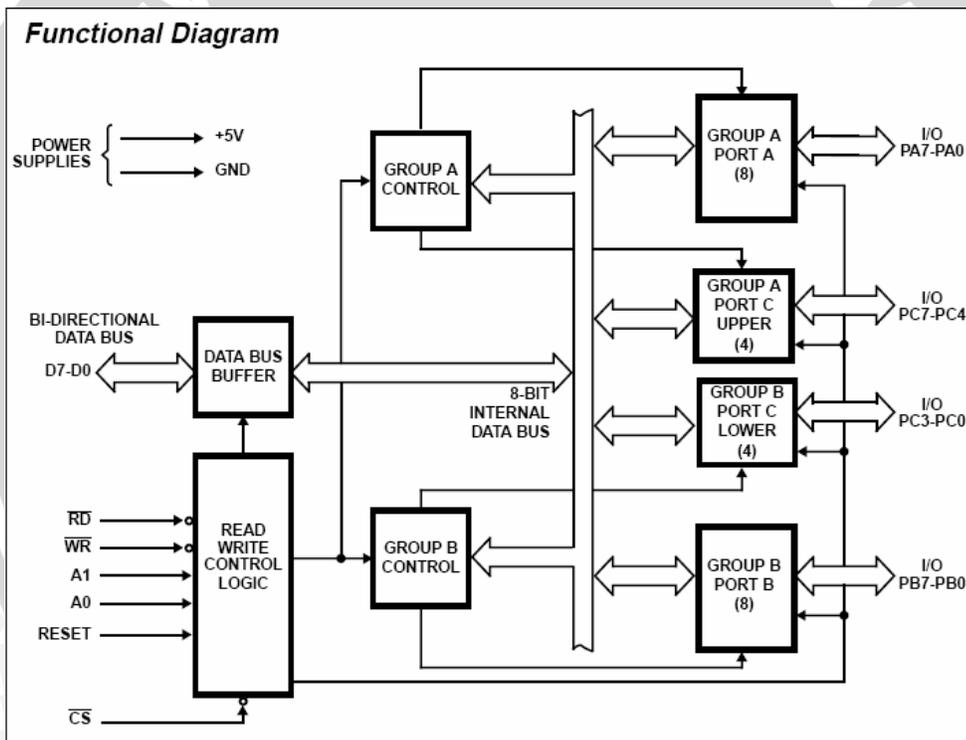
**Gambar 2.7 Tiga Macam Mode Operasi PPI 8255.**

(Sumber : Harris Semiconductor, 1998:4)

Gambar menjelaskan ketiga macam mode operasi PPI 8255. Dengan tiga mode yang ada, menjadikan PPI 8255 sebagai perangkat perantara yang fleksibel, dan mampu mengakomodasi semua sistem perantara paralel. *Port-port* yang ada bisa berubah fungsi dari *input* menjadi *output*, dari *strobed* menjadi biasa, dan dari *directional* menjadi *bi-directional* serta kebalikan dari ketiga perubahan fungsi tersebut. Perubahan mode dan status *port* dilaksanakan melalui pemrograman, dengan jalan memberikan data tertentu pada data *control* (CW) saat dijalankan (Putra, 2002:102).

### 2.4.1 Struktur PPI 8255

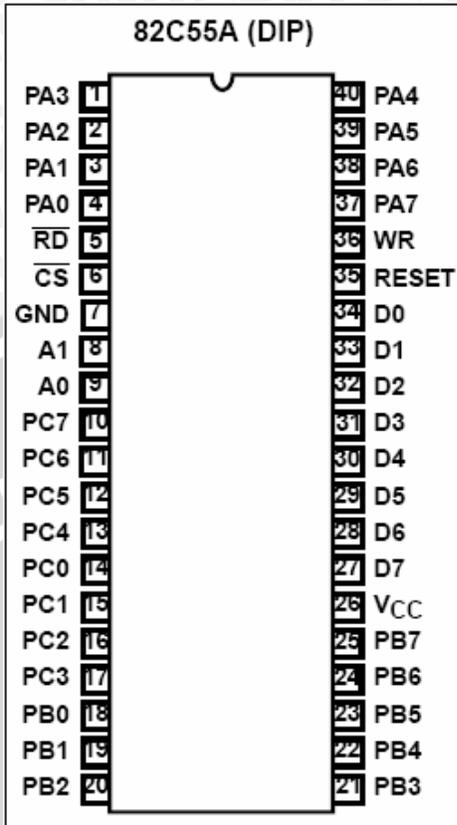
Struktur PPI 8255 tersusun atas beberapa komponen yang saling mendukung satu sama lain. Komponen tersebut adalah sebuah *buffer data bus*, sebuah blok *read / write control logic*, dua buah *group control* A dan B, serta empat buah blok yang menghubungkan *port input / output*. *Group control* A dan B mengendalikan 2 *group port input / output* yaitu *group A* dan *B*. *Group A* dan *B port input / output* sendiri tersusun dari 2 buah *channel*. Untuk *group A*, *channel* penyusunnya adalah *port A7-A0* dan *port C7-C4* (*port C upper*). Untuk *group B* terdiri dari *port B7-B0* dan *port C3-C0* (*port C lower*). PPI 8255 memiliki sebuah *control word* untuk memilih fungsi *port* yang ada sesuai dengan kebutuhan. Untuk lebih jelasnya perhatikan Gambar 2.8 (Harris Semiconductor, 1998:3).



Gambar 2.8 Blok Diagram PPI 8255.  
(Sumber : Harris Semiconductor, 1998:2)

### 2.4.2 Penjelasan Kaki-kaki PPI 8255

Penjelasan dari Gambar tentang kaki-kaki PPI 8255 adalah sebagai berikut (Harris Semiconductor, 1998:2) :



**Gambar 2.9 Konfigurasi Kaki PPI 8255.**  
(Sumber : Harris Semiconductor, 1998:1)

1.  $\overline{RD}$  (*Read*) berfungsi sebagai kontrol proses pembacaan data dari *port* ke *data bus* atau CPU.
2.  $\overline{WR}$  (*Write*) berfungsi sebagai kontrol proses penulisan data dari *data bus* atau CPU ke *port* yang dituju atau *control word*
3.  $\overline{CS}$  (*Chip Select*) merupakan pemilih dan pengunci PPI 8255. Apabila pin ini diaktifkan, maka PPI 8255 akan terpilih dan mulai bisa menerima dan mengirim data.
4. RESET mengembalikan semua *register* PPI 8255 ke kondisi awal, dan format baca dalam mode 0.
5. D0-D7 merupakan *data bus* 8 bit PPI 8255.
6. A0 dan A1 sebagai kendali pemilihan *port* A, B, dan C atau ke *control word*.
7. *Port* A, *Port* B, dan *Port* C merupakan sarana I / O data dari PPI 8255. *Port*-*port* ini memiliki dua *group port*, yaitu *group* A dan B, yang pengoperasiannya tergantung dari mode yang digunakan.

8. Vcc dan *ground* sebagai masukan catu daya untuk PPI 8255, dengan tegangan Vcc sebesar 5 Volt.

Keberadaan fasilitas kontrol RD, WR, CS, A0, A1, dan RESET pada pin-pin PPI 8255 memudahkan pelaksanaan operasi bersama sistem mikroprosesor atau sistem komputer. RD, WR, dan CS akan aktif bila diberi pulsa berlogika rendah. Kombinasi pemberian pulsa dari fasilitas kontrol untuk membentuk sistem operasi diberikan dalam Tabel 2.6 (Harris Semiconductor, 1998:3-4).

**Tabel 2.6 Sistem Operasi PPI 8255.**

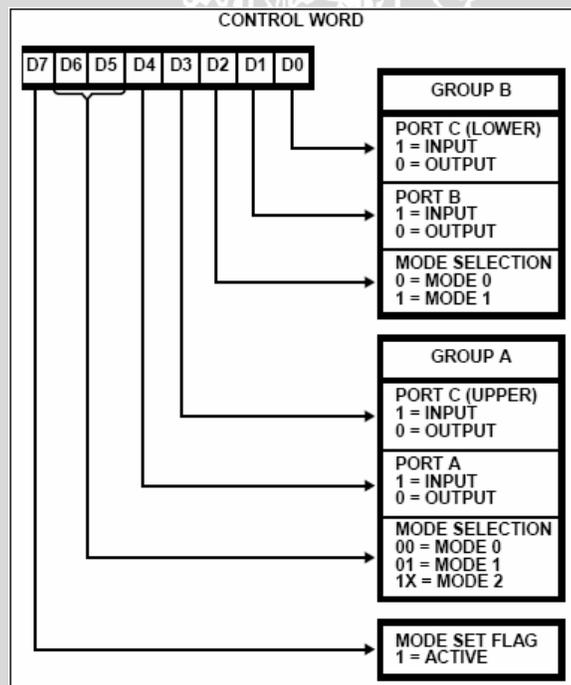
82C55A BASIC OPERATION					
A1	A0	RD	WR	CS	INPUT OPERATION (READ)
0	0	0	1	0	Port A → Data Bus
0	1	0	1	0	Port B → Data Bus
1	0	0	1	0	Port C → Data Bus
1	1	0	1	0	Control Word → Data Bus
OUTPUT OPERATION (WRITE)					
0	0	1	0	0	Data Bus → Port A
0	1	1	0	0	Data Bus → Port B
1	0	1	0	0	Data Bus → Port C
1	1	1	0	0	Data Bus → Control
DISABLE FUNCTION					
X	X	X	X	1	Data Bus → Three-State
X	X	1	1	0	Data Bus → Three-State

Sumber : Harris Semiconductor, 1998:3.

### 2.4.3 Pemrograman PPI 8255

Pemrograman PPI 8255 dilakukan dengan jalan mengeset *register control word* (CW). CW berisi tentang informasi mode operasi, bit set, bit reset, dan parameter-parameter lain. Dengan demikian pemrograman berfungsi untuk memilih mode operasi dan menentukan fungsi *input / output* suatu *port* (Harris Semiconductor, 1998:4)

*Register CW* terdiri dari 8 bit *register* dengan pin-pin penyusun D0-D7. Bit dari pin D0-D2 digunakan untuk *group B*. D0 digunakan untuk menginisialisasi *port C lower*, dengan logika 0 untuk *output* dan logika 1 untuk *input*. Bit dari D1 digunakan untuk *port B* dengan fungsi penggunaan logika sama dengan D0. Sedangkan D2 digunakan sebagai pemilih mode, logika 0 memilih mode 0 dan logika 1 memilih mode 1. Bit D3-D6 digunakan untuk *group A*, D3 untuk *port A* dan D4 untuk *port C upper*. Sedangkan bit pada D5 dan D6 sebagai pemilih mode. Dengan adanya 2 bit pemilih mode, memungkinkan memilih 1 mode dari 3 mode yang ada. Kombinasinya adalah 00 (D5 = 0 dan D6 =0) untuk mode 0, 01 untuk memilih mode 1, dan 1X (X bisa dianggap logika 0 atau 1) sebagai pemilih mode 2. Bit terakhir yakni bit D7, digunakan untuk mode *set flag*, dan harus mendapat logika 1. Untuk keterangan yang lebih jelas perhatikan Gambar 2.11 (Harris Semiconductor, 1992:5).

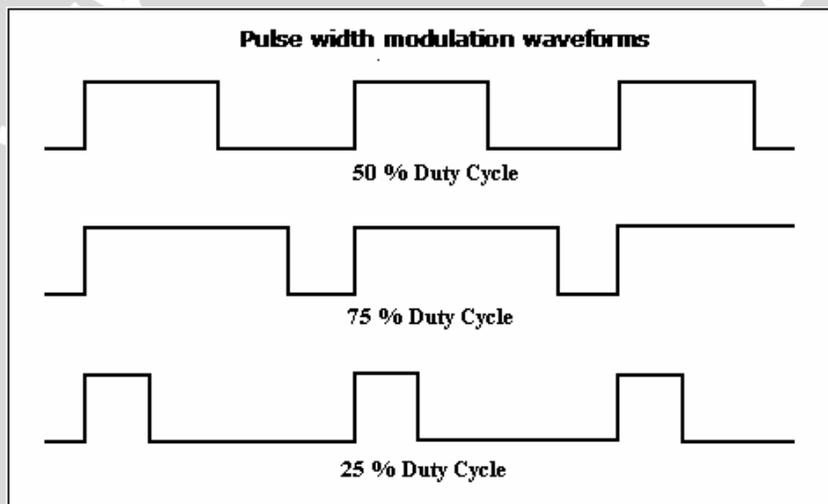


Gambar 2.10 Format Pengisian *Register Control Word*.  
(Sumber : Harris Semiconductor, 1998:4)

## 2.5 Pulse Width Modulation (PWM)

*Pulse Width Modulation* (PWM) merupakan teknik yang umum digunakan untuk pengendalian kecepatan motor yang bisa mengatasi permasalahan performa yang buruk pada saat permulaan (Adler dan Surtell,2000:1).

Analogi yang menggambarkan prinsip kerja PWM adalah seperti pada saat mengendarai *kendaraan* bermotor. Gas ditekan untuk menggerakkan kendaraan dan kemudian dilepas perlahan, memanfaatkan momentum disaat gas ditekan kendaraan akan bergerak maju. Suatu saat kendaraan bergerak melambat (bisa disebabkan hambatan angin, *gesekan* roda, bentuk jalan, dan lain-lain), dan untuk mempercepat laju kendaraan setelahnya gas ditekan kembali selama beberapa saat dan kemudian dilepas *lagi*. “*Duty cycle*” adalah rasio dari saat menggunakan gas (menekan dan melepas gas) dengan total waktu keseluruhan. *Duty cycle* 100% berarti menekan gas setiap saat, sedang *duty cycle* 50% berarti menekan gas setengah dari total waktu dan setengahnya lagi melepas gas. Untuk lebih jelasnya lihat Gambar 2.12 (Adler dan Surtell, 2000:1).



Gambar 2.11 Bentuk Gelombang PWM dengan *Duty Cycle* yang Berbeda.  
(Sumber : Adler dan Surtell, 2000:2)

PWM untuk mengendalikan kecepatan motor bekerja dengan cara yang hampir sama. Daripada *memberikan* tegangan yang berbeda kepada motor, lebih cenderung memberikan tegangan konstan yang bisa membuat motor segera berputar. Tegangan kemudian dilepaskan dan motor melambat. Dengan melanjutkan pemberian/pelepasan tegangan dengan *duty cycle* yang bervariasi, kecepatan motor bisa dikendalikan (Adler dan Suttler, 2000:2).

## 2.6 Tachogenerator DC

Tachogenerator DC merupakan alat yang bisa digunakan untuk mengukur kecepatan motor DC, dengan melakukan konversi putaran motor DC menjadi

tegangannya. Secara konstruksi tachogenerator hampir sama dengan konstruksi dari motor DC. Pada motor DC servo umumnya tachogenerator DC sudah termasuk di dalamnya. Tachogenerator DC berfungsi seperti ketika motor DC digunakan sebagai generator. Besarnya tegangan dari tachogenerator DC ketika digunakan untuk mengukur kecepatan motor DC dihitung dengan rumus 2-3 (Zuhul, 1992:82) :

$$V_s = E + R_a I_a$$

Keterangan :

$I_a$  = Arus jangkar

$R_a$  = Tahanan dalam jangkar

$V_s$  = Tegangan sumber

$E$  = Gaya gerak listrik induksi.

Karena berfungsi sebagai generator, maka tegangan sumber  $V_s$  tidak ada. Sehingga akan seperti motor DC dengan rangkaian terbuka. Dengan demikian besar nilai impedansi beban pada tahanan dalam jangkar  $R_a$  sangatlah besar dan menyebabkan nilai arus jangkar  $I_a = 0$ , sehingga rumus 2-3 menjadi (Zuhul, 1992:82):

$$I_a = \frac{1}{R_a} (V_s - E) = 0 \quad (2-5)$$

Kemudian dari rumus 2-4 diperoleh rumus (Zuhul, 1992:82):

$$E = C_i n \phi = V_s \quad (2-6)$$

Keterangan :

$E$  = Gaya gerak listrik induksi

$C_i$  = Konstanta induksi

$n$  = Putaran

$\phi$  = Fluks medan magnet

$V_s$  = Tegangan sumber

Karena fungsinya sebagai generator, maka tegangan sumber  $V_s$  tidak terhubung dengan catu daya. Tegangan yang mengalir di  $V_s$  berasal dari gaya gerak listrik  $E$ . Karena nilai  $C_i$  dan  $\phi$  yang konstan, maka besarnya tegangan sumber  $V_s$  sangat dipengaruhi oleh putaran  $n$ . Semakin besar (cepat) putaran  $n$  maka nilai  $V_s$  akan semakin besar juga.

Besarnya torsi yang terjadi dihitung dengan rumus 2-1 (Zuhail, 1992:81) :

$$T = C_a I_a \phi$$

Keterangan :

T = Torsi

$I_a$  = Arus jangkar(Ampere)

$\phi$  = Fluks medan magnet

Dari rumus 2-5 besarnya arus jangkar  $I_a = 0$ , maka besarnya kopel T = 0.

Sehingga rumus 2-1 menjadi (Zuhail, 1992:81) :

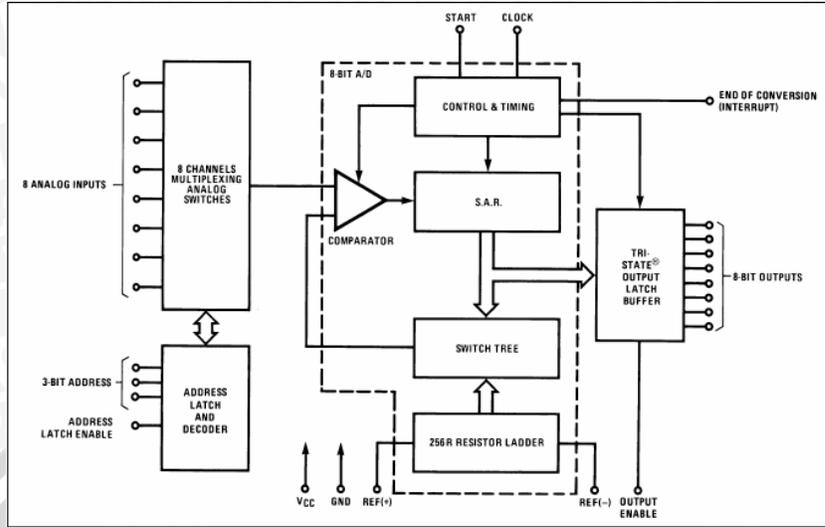
$$T = C_a I_a \phi = 0 \quad (2-7)$$

## 2.7 ADC 0808/0809

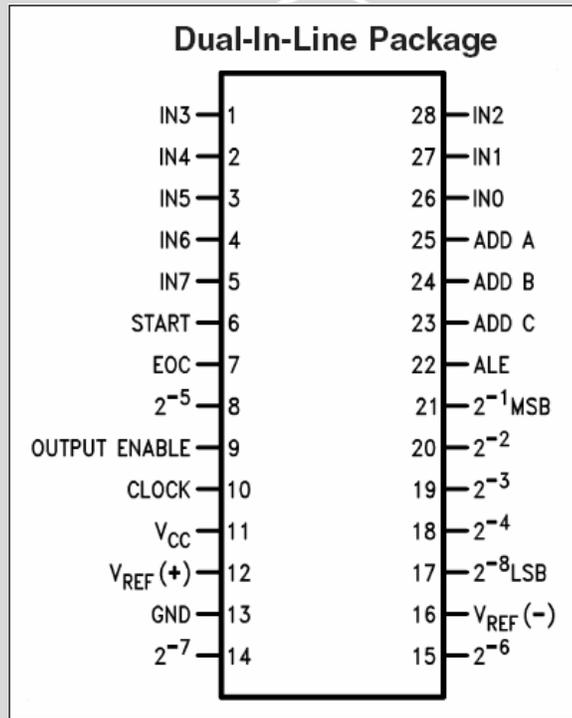
ADC 0808/ADC 0809 merupakan komponen konverter analog ke digital 8 bit dengan CMOS bersifat monolitik, 8 jalur multiplexer, dan mikroprosesor pengendali logika yang sesuai. A/D konverter menggunakan pendekatan berturut-turut dalam teknik konversi yang dilakukan. Konverter memiliki ciri-ciri pembandingan yang distabilkan oleh pemotong impedansi tinggi, 256R pembagi tegangan dengan susunan saklar analog, dan register pendekatan berturut-turut. 8 jalur multiplexer bisa mengakses secara langsung setiap jalur dari 8 jalur sinyal analog (National Semiconductor, 2002:1).

Komponen ini mengurangi kebutuhan nol dari luar, serta penyesuaian skala penuh. Antar muka yang mudah untuk mikroprosesor disediakan oleh *latched* dan pengkodean alamat masukan multiplexer serta *latched* keluaran oleh TTL TRI-STATE (National Semiconductor, 2002:1).

Desain dari ADC 0808/0809 telah dioptimasi melalui penggabungan dari berbagai aspek yang diinginkan pada beberapa teknik konversi analog ke digital. ADC 0808/0809 memberikan kecepatan tinggi, akurasi tinggi, ketergantungan temperatur minimal, akurasi jangka panjang yang baik serta berulang-ulang, dan kebutuhan catu daya yang sedikit. Kelebihan ini menjadikan komponen ini ideal digunakan untuk aplikasi mulai dari proses, kontrol mesin dan aplikasi otomotif (National Semiconductor, 2002:1).



Gambar 2.12 Blok Diagram ADC 0808/0809.  
(Sumber : National Semiconductor, 2002:1)



Gambar 2.13 Konfigurasi Kaki ADC 0808/0809.  
(Sumber : National Semiconductor, 2002:2)

Komponen ini memiliki multiplexer sinyal analog 8 jalur satu arah. Jalur input dipilih menggunakan penyandi alamat. Tabel 2.8 memperlihatkan kondisi masukan untuk memilih alamat dari jalur input yang diinginkan. Alamat di-latch ke dalam penyandi pada kondisi perubahan sinyal rendah ke tinggi untuk *Address Latch Enable* (National Semiconductor, 2002:5).

Tabel 2.8 Jalur Alamat Masukan Sinyal Analog ADC 0808/0809.

SELECTED ANALOG CHANNEL	ADDRESS LINE		
	C	B	A
IN0	L	L	L
IN1	L	L	H
IN2	L	H	L
IN3	L	H	H
IN4	H	L	L
IN5	H	L	H
IN6	H	H	L
IN7	H	H	H

Sumber : National Semiconductor, 2002:2.

### 2.7.1 Konversi Rasiometri

ADC 0808/0809 didesain sebagai perangkat *Data Acquisition System* (DAS) pada sistem konversi rasiometri. Dalam sistem rasiometri, variabel fisik yang telah diukur diekspresikan sebagai prosentase skala penuh yang tidak berhubungan dengan standar mutlak. Tegangan masukan pada ADC 0808/0809 diberikan dalam persamaan (National Semiconductor, 2002:9) :

$$\frac{V_{IN}}{V_{fs} - V_Z} = \frac{D_X}{D_{MAX} - D_{MIN}} \quad (2-8)$$

Keterangan :

$V_{IN}$  = Tegangan masukan pada ADC 0808/0809

$V_{fs}$  = Tegangan skala penuh

$V_Z$  = Tegangan nol

$D_X$  = Data yang diukur

$D_{MAX}$  = Batas data maksimum

$D_{MIN}$  = Batas data minimum

Persamaan Konversi Analog ke Digital pada ADC 0808/0809 menggunakan persamaan peralihan antara kode yang berdekatan N dan N+1, dan dijelaskan sebagai berikut (National Semiconductor, 2002:12) :

$$V_{IN} = \left\{ (V_{REF(+)} - V_{REF(-)}) \left[ \frac{N}{256} + \frac{1}{512} \right] \pm V_{TUE} \right\} + V_{REF(-)} \quad (2-9)$$

Nilai tengah dari keluaran kode N diberikan dalam persamaan (National Semiconductor, 2002:12) :

$$V_{IN} \left\{ \left( V_{REF(+)} - V_{REF(-)} \right) \left[ \frac{N}{256} \right] \pm V_{TUE} \right\} + V_{REF(-)} \quad (2-10)$$

Keluaran kode N untuk masukan yang telah ditentukan berupa bilangan bulat dalam batas (National Semiconductor, 2002:12) :

$$N = \frac{V_{IN} - V_{REF(-)}}{V_{REF(+)} - V_{REF(-)}} \times 256 \pm \text{AkurasiMutlak} \quad (2-11)$$

Keterangan :

$V_{IN}$  = Tegangan pada masukan pembanding

$V_{REF(+)}$  = Tegangan pada REF(+)

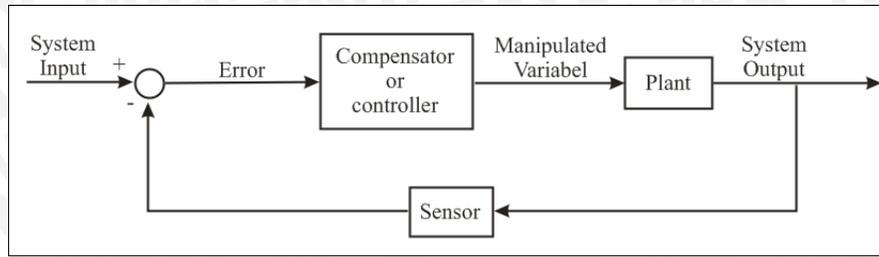
$V_{REF(-)}$  = Tegangan pada REF(-)

$V_{TUE}$  = Total tegangan error yang tidak dirubah (biasanya  $V_{REF(+)} : 512$ )

## 2.8 Sistem Kontrol

Sistem kontrol merupakan sebuah susunan yang terdiri dari komponen fisik yang didesain untuk mengubah, mengatur, atau mengendalikan sistem fisik lain melalui sebuah aksi kontrol, sehingga sistem fisik yang dikontrol mengeluarkan perilaku atau karakteristik sesuai dengan yang diinginkan. Sistem kontrol secara umum ada dua macam : sistem kontrol loop terbuka yang aksi kontrolnya tidak dipengaruhi oleh keluaran dari sistem fisik yang dikontrol, dan sistem kontrol loop tertutup (sistem kontrol umpan balik) yang aksi kontrolnya bergantung terhadap keluaran sistem fisik yang dikontrol (Ross,1997:469)

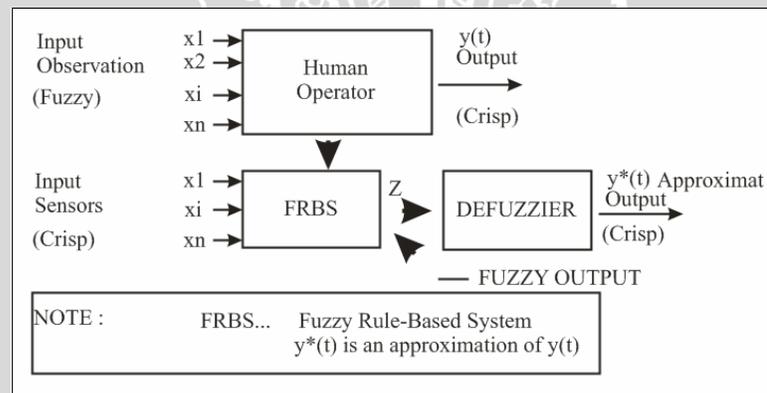
Untuk mengontrol suatu variabel fisik, diperlukan untuk suatu pengukuran. Sistem pengukuran untuk sinyal yang dikontrol dinamakan sensor. Sistem fisik yang dikontrol dinamakan plant. Dalam sebuah sistem kontrol loop tertutup, sinyal pemaksa yang pasti (masukan) sangat dipengaruhi oleh tanggapan sistem (keluaran). Untuk memperoleh tanggapan dan karakteristik yang memuaskan dari sebuah sistem kontrol loop tertutup, sangatlah penting untuk memberikan sebuah sistem tambahan (kompensator atau kontroler) ke dalam loop. Bentuk umum dari sistem kontrol loop tertutup bisa dilihat dalam Gambar (Ross,1997:469-470):



**Gambar 2.14 Sistem Kontrol Loop Tertutup.**  
(Sumber : Ross, 1997:471)

### 2.8.1 Sistem Kontrol Logika Fuzzy

Sistem kontrol logika *fuzzy* merupakan sistem kontrol berbasis aturan, di mana kumpulan aturan yang ada dinamakan aturan-aturan *fuzzy*. Aturan-aturan *fuzzy* memiliki peran sebagai mekanisme pengambilan keputusan untuk mengatur akibat dari beberapa sebab yang timbul dalam sebuah sistem. Tujuan dari sistem kontrol *fuzzy* biasanya adalah untuk menggantikan operator manusia dengan sistem berbasis aturan *fuzzy* (Jamshidi, 1993:5).



**Gambar 2.15 Definisi Sistem Kontrol Logika Fuzzy.**  
(Sumber : Jamshidi, 1993:5)

Ada beberapa asumsi mendasar yang mempengaruhi desain sistem kontrol logika *fuzzy*. 6 asumsi dasar yang umumnya muncul ketika memilih sistem kontrol berbasis logika *fuzzy* adalah (Ross, 1997:474-475) :

1. Plant teramati (observable) dan bisa dikontrol (controllable) : Variabel keadaan, variabel masukan, dan variabel keluaran tersedia untuk pengamatan, pengukuran, dan perhitungan.
2. Adanya basis pengetahuan yang bisa diperoleh dari kumpulan aturan linguistik yang dihasilkan oleh antara lain : pakar, keahlian teknis insinyur, intuisi,

- kumpulan data masukan / keluaran hasil pengukuran, atau analisis model yang bisa di-*fuzzy*-kan dan bisa menghasilkan aturan-aturan pengontrolan.
3. Adanya cara penyelesaian.
  4. Pembuat kontroler hanya membutuhkan cara penyelesaian yang “cukup bagus”. Tidak mementingkan penyelesaian yang optimal.
  5. Desain kontroler yang dibuat merupakan hasil pemikiran yang terbaik dari pengetahuan yang telah didapat dan dalam batasan kecermatan yang bisa diterima.
  6. Permasalahan kestabilan dan optimasi merupakan masalah yang ada dalam desain kontrol logika *fuzzy*.

### 2.8.2 Logika *Fuzzy*

Logika *fuzzy* berasal dari teori himpunan *fuzzy* yang dikenalkan pertama kali oleh Lotfi A. Zadeh pada tahun 1965. Himpunan *fuzzy* merupakan perluasan dari teori himpunan klasik. Dalam teori himpunan klasik hubungan keanggotaan dalam elemen dinilai berdasar lingkup biner merujuk pada kondisi yang tegas (*crisp*). Sebuah elemen dianggap anggota atau bukan anggota dalam himpunan. Sebagai perbandingan, teori himpunan *fuzzy* membolehkan penilaian yang bertahap terhadap keanggotaan sebuah elemen di dalam sebuah himpunan. Hal ini dijelaskan dengan bantuan dari fungsi keanggotaan yang dinilai dalam interval unit yang nyata  $[0,1]$ . Fungsi keanggotaan bisa berlaku sebagai fungsi penentu, memetakan semua elemen pada 1 atau 0 seperti dalam notasi klasik (Jamshidi,1993:2).

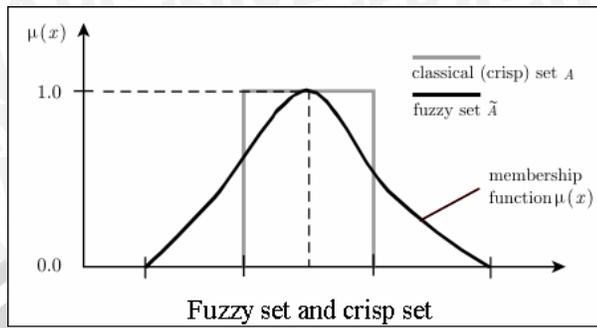
### 2.8.3 Himpunan *Fuzzy*

Himpunan *fuzzy* pada sebuah himpunan klasik  $X$  ditetapkan sebagai berikut (Jamshidi, 1993:17):

$$\tilde{A} = \langle (x, \mu_A(x)) \mid x \in X \rangle \quad (2-12)$$

Fungsi keanggotaan  $\mu_A(x)$  mengukur derajat keanggotaan dari elemen  $x$  terhadap himpunan pokok  $X$ . Sebuah elemen dipetakan pada nilai 0 yang berarti bukan anggota pada himpunan, 1 merupakan anggota yang sepenuhnya masuk

dalam himpunan. Nilai yang seksama antara 0 dan 1 merupakan anggota *fuzzy* (Jamshidi,1993:17).



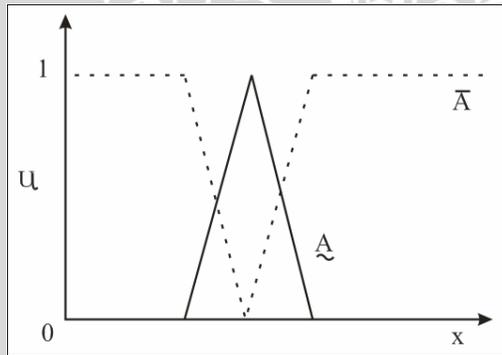
**Gambar 2.16 Perbedaan Himpunan Fuzzy dengan Himpunan Klasik (Crisp).**  
(Sumber : Jamshidi, 1993:17)

### 2.8.4 Operasi Himpunan Fuzzy

Operasi himpunan *fuzzy* merupakan operasi pada himpunan *fuzzy*. Operasi ini merupakan operasi yang umum digunakan pada operasi himpunan tegas. Ada lebih dari satu kemungkinan operasi yang bisa digunakan. Operasi yang sering digunakan disebut operasi himpunan *fuzzy* standar, yang terdiri dari (Jamshidi, 1993:18):

1. Komplemen *fuzzy* (Jamshidi,1993:18) :

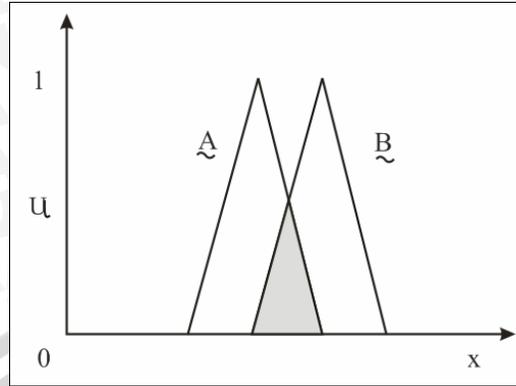
$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \tag{2-13}$$



**Gambar 2.17 Komplemen Fuzzy.**  
(Sumber : Jamshidi, 1993:19)

2. Irisan *fuzzy* (Jamshidi,1993:18) :

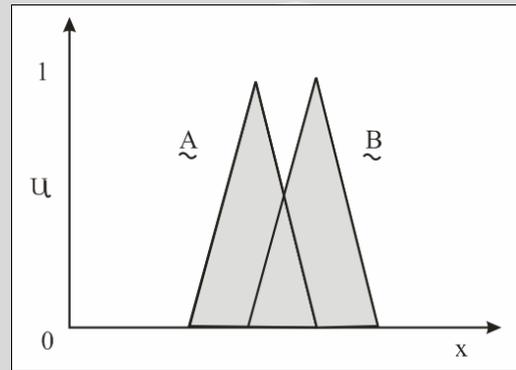
$$\mu_{(A \cap B)}(x) = \min[\mu_A(x), \mu_B(x)] \tag{2-14}$$



**Gambar 2.18 Irisan Fuzzy.**  
(Sumber : Jamshidi, 1993:19)

3. Gabungan *fuzzy* (Jamshidi, 1993:18) :

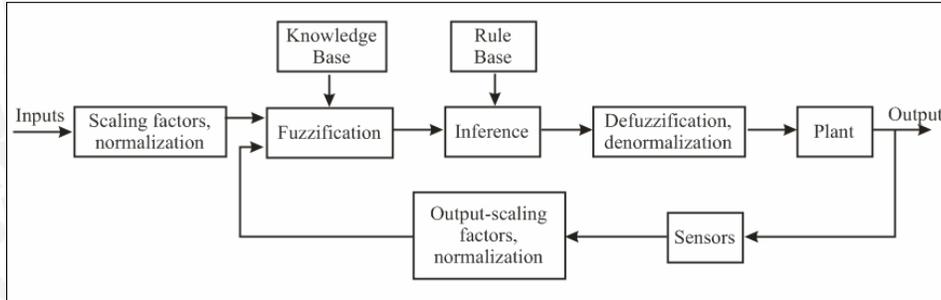
$$\mu(A \cup B)(x) = \max[\mu A(x), \mu B(x)] \quad (2-15)$$



**Gambar 2.19 Gabungan Fuzzy**  
(Sumber : Jamshidi, 1993:18)

### 2.8.5 Langkah-langkah dalam Desain Kontrol Logika Fuzzy

Kontrol logika *fuzzy* sederhana secara umum bisa dilihat dalam Gambar. Modul basis pengetahuan dalam Gambar berisi tentang semua bagian masukan dan keluaran dalam bentuk *fuzzy*. Hal itu termasuk kumpulan aturan dan hubungan fungsi keanggotaan dari variabel masukan dengan sistem berbasis aturan *fuzzy* terhadap variabel keluaran, atau dengan kata lain aksi kontrol terhadap plant yang dikontrol (Ross, 1997:475).



**Gambar 2.20 Blok Diagram Kontrol Logika Fuzzy Sederhana.**

(Sumber : Ross, 1997:476)

Langkah-langkah dalam desain sebuah kontrol logika *fuzzy* sederhana adalah sebagai berikut (Ross, 1997:475-476) :

1. Mengenali variabel-variabel dari plant (masukan, keadaan, dan keluaran).
2. Pembagian dari semesta pembicaraan atau rentang jarak masing-masing variabel ke dalam nilai dalam bagian himpunan *fuzzy*, memberi label linguistik kepada setiap bagian himpunan sebagai elemen dalam semesta pembicaraan.
3. Menentukan fungsi keanggotaan terhadap setiap bagian himpunan *fuzzy*.
4. Menentukan hubungan *fuzzy* antara bagian himpunan masukan atau keadaan pada satu sisi dan bagian himpunan keluaran pada sisi yang lain. Sehingga bisa membentuk basis aturan.
5. Memilih faktor skala yang sesuai untuk variabel masukan dan variabel keluaran agar masuk dalam rentang jarak  $[0,1]$  atau  $[-1,1]$ .
6. Memasukkan masukan *fuzzy* ke dalam kontroler.
7. Menggunakan pertimbangan pendekatan *fuzzy* untuk memutuskan keluaran yang dihasilkan dari masing-masing aturan.
8. Menjumlahkan setiap keluaran *fuzzy* yang direkomendasikan dari masing-masing aturan.
9. Menggunakan defuzzification untuk mendapatkan keluaran tegas.

Dari langkah-langkah yang biasanya dilakukan dalam mendesain sistem kontrol logika *fuzzy* sederhana, diperoleh beberapa prinsip yang umum untuk dilakukan ketika mendesain kontrol logika *fuzzy*. Prinsip-prinsip dalam mendesain sistem kontrol logika *fuzzy* adalah sebagai berikut (Ross, 1997:477) :

1. *Fuzzification*
2. *Knowledge Base* (Basis Pengetahuan) .
3. *Rule-Base* (Basis Aturan)

4. *Decision-masking Logic* (Logika Pengambil Keputusan)
5. *Defuzzification*

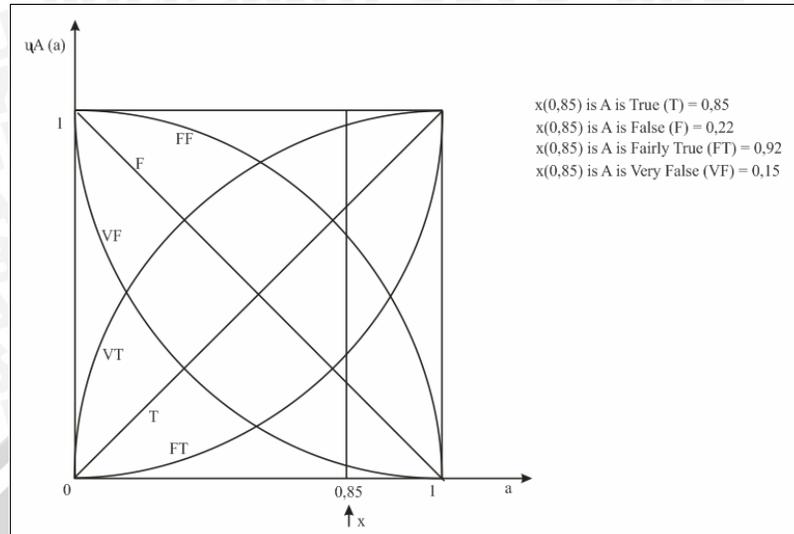
### 2.8.6 Fuzzification

Untuk memetakan variabel masukan, variabel keadaan dan variabel keluaran dari sistem kontrol logika *fuzzy* kedalam bagian himpunan *fuzzy* dalam semesta pembicaraan *fuzzy*. Proses mengubah nilai masukan tegas ke dalam nilai *fuzzy* biasanya dinamakan “*fuzzification*”. Nilai dalam himpunan *fuzzy* akan memiliki nilai kebenarannya masing-masing. Nilai kebenaran dalam himpunan *fuzzy* berfungsi hampir sama dengan himpunan tegas. Di mana dalam himpunan tegas anggotanya dipetakan dengan nilai kebenaran yang selalu berupa nilai 1 atau 0. Dalam himpunan *fuzzy* nilai tersebut tidak selalu mutlak 1 atau 0, misalnya nilai kebenaran bisa jadi hanya  $\frac{1}{4}$  bagian dari 1,  $\frac{1}{8}$  dari 1, dan lain-lain.

Bila  $\tau$  merupakan nilai kebenaran dalam *fuzzy*, dan dicontohkan bahwa “sangat benar (very true)”, ”benar (true)”, “sedikit benar (fairly true)”, “sedikit salah (fairly false)”, dan lain sebagainya merupakan nilai kebenaran. Nilai kebenaran tersebut bisa dianggap sebagai subyek *fuzzy* dari rentang jarak yang dikarakterisasi oleh fungsi keanggotaan  $x_\tau : [0,1] \rightarrow [0 \times 1]$ . Nilai kebenaran bisa digambarkan sebagai “x adalah A adalah  $\tau$ ”, atau dalam bentuk rumus (Jamshidi, 1993:68) :

$$x \text{ is } A \text{ is } \tau \rightarrow \mu = \mu x^+ \quad (2-16)$$

$$\mu x^+(x) = x_\tau[\mu x(x)] \quad (2-17)$$



**Gambar 2.21 Grafik Nilai Kebenaran.**

(Sumber : Jamshidi,1997:69)

### 2.8.7 Basis Pengetahuan.

Basis pengetahuan merupakan diperoleh dari aturan linguistik yang dihasilkan oleh antara lain : pakar, keahlian teknis insinyur, intuisi, kumpulan data masukan / keluaran hasil pengukuran, atau analisis model yang bisa di-fuzzy-kan dan bisa menghasilkan aturan-aturan pengontrolan (Ross, 1997:474).

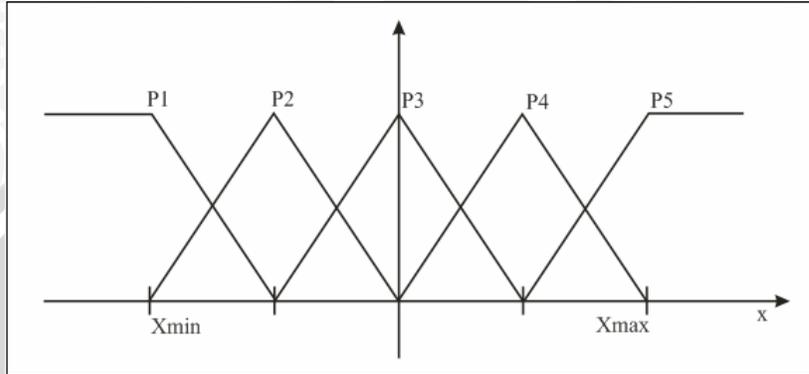
Basis pengetahuan menentukan fungsi keanggotaan dalam himpunan fuzzy. Ada tiga langkah yang sering digunakan berdasarkan intuisi untuk menentukan fungsi keanggotaan dari suatu kondisi masukan serta akibatnya terhadap keluaran. Tiga metode ini dijelaskan berdasarkan pada konsep kategori prototipe, derajat kesamaan, dan kesamaan dalam jarak (Jamshidi, 1993:108).

Untuk menentukan fungsi keanggotaan bila batasan variabel masukan dan variabel keluaran diketahui, maka langkah yang ditempuh adalah sebagai berikut (Jamshidi, 1993:108) :

1. Rentang jarak yang ada dibagi sama rata ke dalam n-jumlah bagian.
2. Diambil sebuah titik tengah dari bagian yang telah dibuat, titik ini mewakili titik prototipe.
3. Untuk dua bagian yang terluar (minimum dan maksimum) titik akhirnya dianggap saling terhubung. Sehingga derajat fungsi keanggotaannya diwakli oleh titik prototipe.

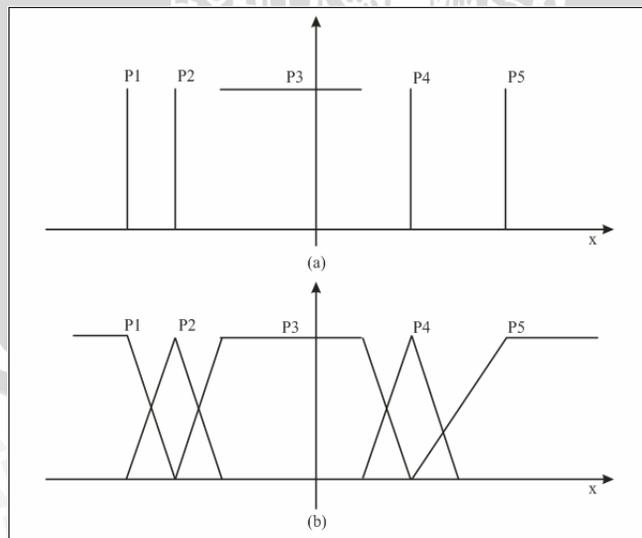
- Untuk setiap titik protipe yang berhubungan dengan titik prototipe lain dianggap sama dengan nol.

Dari 4 langkah yang ada, maka fungsi keanggotaan yang ditentukan akan berbentuk segitiga dalam jumlah sama dengan  $n$ . Prosedur ini dijelaskan dalam Gambar 2.23.



**Gambar 2.22 Bagian Himpunan fuzzy untuk Titik Prototipe Seimbang.**  
(Sumber : Jamshidi, 1993:109)

Cara lain yang mungkin dilakukan adalah dengan mengambil data berdasarkan keahlian seseorang. Data tersebut berupa nilai dari titik yang paling sering muncul, titik tunggal, atau rentang jarak (prototipe) yang berhubungan dengan kategori linguistik dalam sudut pandang ahli tersebut terhadap sistem. Dengan menggunakan cara yang hampir sama dengan cara yang sebelumnya dijelaskan, untuk lebih jelasnya lihat Gambar 2.24 (Jamshidi, 1993:109) :



**Gambar 2.23 (a) Titik Prototipe dan Rentang Jarak (b) Fungsi Keanggotaan.**  
(Sumber : Jamshidi, 1993:110)

### 2.8.8 Basis Aturan

Level *fuzzy* untuk memahami dan menggambarkan sebuah sistem yang kompleks pada dasarnya adalah dengan menempatkan bentuk kumpulan batasan terhadap dasar keluaran dari kondisi masukan yang pasti. Batasan secara umum dimodelkan oleh himpunan *fuzzy* dan relasi *fuzzy*. Pernyataan batasan baik bersyarat maupun tidak bersyarat biasanya dihubungkan dengan bahasa penghubung seperti “AND”, “OR”, atau “ELSE”. Bentuk baku dari kumpulan aturan diberikan dalam Gambar 2.25. Batasan (Restriction)  $R^1, R^2, \dots, R^r$  digunakan pada aksi keluaran atau keputusan yang dibuat untuk performa yang diinginkan (Jamshidi, 1993:59).

$R^1$	Restriction $R^1$
$R^2$	Restriction $R^2$
•	
•	
•	
$R^k$	Restriction $R^k$
$R^{k+1}$	IF condition $C^1$ , THEN restriction $R^{k+1}$
$R^{k+2}$	IF condition $C^2$ , THEN restriction $R^{k+2}$
•	
•	
•	
$R^r$	IF condition $C^{r-k}$ , THEN restriction $R^r$

**Gambar 2.24 Bentuk Baku dari Kumpulan Aturan.**

(Sumber : Jamshidi, 1993:60)

Pernyataan tak bersyarat dianggap sama dan bisa diperlakukan seperti pernyataan bersyarat dengan IF sebagai ketentuan kondisi dari kondisi masukan dalam semesta pembicaraan yang selalu bernilai benar dan memuaskan. Batasan tak bersyarat seperti “keluaran adalah rendah” bisa dianggap sama, dan ditulis sebagai (Jamshidi, 1993:61) :

IF suatu kondisi THEN keluaran adalah rendah.

Syarat dasar linguistik merupakan bentuk kumpulan batasan yang dimodelkan oleh himpunan *fuzzy* dan relasi *fuzzy* adalah sebagai berikut (Jamshidi, 1993:63) :

1. Syarat utama yang merupakan label dari bagian himpunan *fuzzy* pada semesta pembicaraan (seperti : panas, dingin, keras, bawah, dan lain sebagainya).
2. Negasi “NOT”, dan penghubung “AND” dan “OR”
3. Batasan seperti “sangat”, “lebih”, “sedikit”, dan lain sebagainya.
4. Penanda seperti tanda kurung.

Persamaan dari sistem relasi *fuzzy* bisa diperoleh melalui banyak teknik. Yang diawali dari bentuk umum pernyataan bersyarat *fuzzy* IF A THEN B (bentuk umum modus ponens) bisa diperoleh persamaan relasi *fuzzy* R yang mewakili bentuk umum modus ponens dengan “•” mewakili bentuk umum metode untuk komposisi relasi *fuzzy* (Jamshidi, 1993:73) :

$$B = A \bullet R \quad (2-18)$$

Untuk memperoleh keseluruhan keluaran dari himpunan *fuzzy* y dari beberapa keluaran tunggal yang dihasilkan oleh beberapa aturan tunggal atau persamaan relasi *fuzzy* digunakan aggregation of rule (penyatuan aturan). Penyatuan aturan ada 2 macam, yaitu (Jamshidi, 1997:75) :

1. Sistem aturan konjungsi : Dalam kasus di mana sistem aturan diinginkan untuk digabung secara memuaskan, aturan-aturan dihubungkan dengan penghubung “AND”. Keluaran penyatuan ditentukan oleh rumus irisan *fuzzy* dari masing-masing aturan keluaran. Rumus sistem aturan konjungsi (Jamshidi, 1993:75) :

$$y = (y^1) \text{AND} (y^2) \text{AND} \dots \text{AND} (y^r) \quad (2-19)$$

Atau (Jamshidi, 1993:75),

$$y = (y^1) \cap (y^2) \cap \dots \cap (y^r) \quad (2-20)$$

Yang ditentukan oleh fungsi keanggotaan (Jamshidi, 1993:75)

$$\mu_y(y) = \text{Min}[\mu_{y^1}(y), \mu_{y^2}(y), \dots, \mu_{y^r}(y)], \text{ untuk } : y \in Y \quad (2-21)$$

Sistem aturan konjungsi bisa digambarkan sebagai persamaan tunggal relasi *fuzzy* (Jamshidi, 1993:75) :

$$y = (x \bullet R^1) \text{AND} (x \bullet R^2) \text{AND} \dots \text{AND} (x \bullet R^r) \quad (2-22)$$

Yang sama dengan (Jamshidi, 1993:75) :

$$y = x \bullet (R^1 \text{AND} R^2 \text{AND} \dots \text{AND} R^r) \quad (2-23)$$

Akhirnya (Jamshidi, 1993:75) :

$$y = x \bullet R \quad (2-24)$$

Dengan R ditentukan sebagai (Jamshidi, 1993:75) :

$$R = R^1 \cap R^2 \cap \dots \cap R^{\Gamma} \quad (2-25)$$

2. Sistem aturan dijumpi : Dalam kasus di mana sistem aturan dibutuhkan hanya salah satu yang paling memuaskan, aturan-aturan dihubungkan dengan penghubung “OR”. Keluaran penyatuan ditentukan oleh rumus gabungan *fuzzy* dari masing-masing aturan keluaran. Rumus sistem aturan disjungsi (Jamshidi, 1993:76) :

$$y = (y^1)OR (y^2)OR \dots OR (y^{\Gamma}) \quad (2-26)$$

Atau (Jamshidi, 1993:76),

$$y = (y^1) \cup (y^2) \cup \dots \cup (y^{\Gamma}) \quad (2-27)$$

Yang ditentukan oleh fungsi keanggotaan (Jamshidi, 1993:76),

$$\mu_y(y) = \text{Max}[\mu_{y^1}(y), \mu_{y^2}(y), \dots, \mu_{y^{\Gamma}}(y)], \text{ untuk } : y \in Y \quad (2-28)$$

Sistem aturan disjungsi bisa digambarkan sebagai persamaan tunggal relasi *fuzzy* (Jamshidi, 1993:76) :

$$y = (x \bullet R^1)OR (x \bullet R^2)OR \dots OR (x \bullet R^{\Gamma}) \quad (2-29)$$

Yang sama dengan (Jamshidi, 1993:76) :

$$y = x \bullet (R^1 OR R^2 OR \dots OR R^{\Gamma}) \quad (2-30)$$

Akhirnya (Jamshidi, 1993:76) :

$$y = x \bullet R \quad (2-31)$$

Dengan R ditentukan sebagai (Jamshidi, 1993:76) :

$$R = R^1 \cup R^2 \cup \dots \cup R^{\Gamma} \quad (2-32)$$

Ada 2 cara yang umum digunakan untuk memperoleh relasi *fuzzy* R yang mewakili bentuk umum modus ponens IF A THEN B, yaitu (Jamshidi, 1993:79) :

1. Implikasi *Fuzzy* : Ada banyak cara berbeda untuk memperoleh relasi *fuzzy* R pada himpunan *fuzzy* dari bagian IF dan bagian THEN dalam pernyataan bersyarat IF A THEN B, yaitu (Jamshidi, 1993:79) :

$$\mu_R(x, y) = \text{Max}\{\text{Min}[\mu_A(x), \mu_B(y)], 1 - \mu_A(x)\} \quad (2-33)$$

$$\mu_R(x, y) = \text{Max}\{[\mu_B(y), 1 - \mu_A(x)]\} \quad (2-34)$$

$$\mu_R(x, y) = \text{Min}[\mu_A(x), \mu_B(y)] \quad (2-35)$$

$$\mu_R(x, y) = \text{Min}\{1, [1 - \mu_A(x) + \mu_B(y)]\} \quad (2-36)$$

$$\mu_R(x, y) = \text{Min}\{1, [\mu_A(x) + \mu_B(y)]\} \quad (2-37)$$

$$\mu_R(x, y) = \text{Min}\left\{1, \left[\frac{\mu_B(y)}{\mu_A(x)}\right]\right\} \quad (2-38)$$

$$\mu_R(x, y) = \text{Max}\{\mu_A(x) \cdot \mu_B(y), [1 - \mu_A(x)]\} \quad (2-39)$$

$$\mu_R(x, y) = \mu_A(x) \cdot \mu_B(y) \quad (2-40)$$

$$\mu_R(x, y) = [\mu_B(y)]^{\mu_A(x)} \quad (2-41)$$

$$\mu_R(x, y) = \begin{cases} \mu_B(y), & \text{for } \mu_B(y) < \mu_A(x) \\ 1 & \text{otherwise} \end{cases} \quad (2-42)$$

2. Prinsip Perluasan :  $f$  merupakan pemetaan dari  $x_1 \times x_2 \times \dots \times x_n$  pada semesta  $y$  seperti  $y = f(x_1, x_2, \dots, x_n)$ . Prinsip perluasan *fuzzy* membolehkan terjadinya induksi dari himpunan *fuzzy*  $n$   $A_1, A_2, \dots, A_n$ , oleh himpunan *fuzzy*  $B$  pada  $y$  melalui  $f$  sehingga (Jamshidi, 1993:80) :

$$\begin{aligned} \mu_B(y) &= \text{Sup}_{(x_1, x_2, \dots, x_n)} \\ &= f^{-1}(y) \{ \text{Min}[\mu_{A_1}(x_1), \mu_{A_2}(x_2), \dots, \mu_{A_n}(x_n)] \} \end{aligned} \quad (2-43)$$

$$\mu_B(y) = 0, \text{ if } f^{-1}(y) = \phi \quad (2-44)$$

Di mana  $f^{-1}(y)$  adalah kebalikan dari  $f$ , dan  $\mu_B(y)$  adalah yang terbesar diantara nilai keanggotaan  $\mu_{A_1} \times A_2 \times \dots \times A_n(x_1, x_2, \dots, x_n)$  dari realisasi  $y$  menggunakan  $n$ -tuples  $(x_1, x_2, \dots, x_n)$ .

Untuk memudahkan dalam menggunakan aturan-aturan yang telah dibuat dalam sistem kontrol *fuzzy*, biasanya dibuat *Fuzzy Associative Memory* (FAM). FAM memetakan aturan IF A THEN B yang merupakan bentuk umum modus ponens aturan *fuzzy* ke dalam sebuah tabel. Gambar model tabel yang menggambarkan hubungan dari variabel masukan *fuzzy* dengan variabel keluaran *fuzzy* dengan masing-masing relasinya diberikan dalam Gambar 2.26.

B5					X		
B4							
B3				X			
B2			X			X	
B1	X	X					X
Output	A1	A2	A3	A4	A5	A6	A7
Input							

Gambar 2.25 Tabel FAM untuk Masukan Tunggal Keluaran Tunggal.  
(Sumber : Jamshidi, 1993:111)

**2.8.9 Logika Pengambil Keputusan**

Logika pengambil keputusan merupakan

Komposisi relasi *fuzzy* yang digunakan untuk memperoleh hasil dari aturan-aturan yang ada adalah :

1. Komposisi Relasi Max-Min (Jamshidi, 1993:81)

$$y^k = x \bullet R^k$$

$$\mu_{y^k}(y) = \text{Max}_{x \in X} \{ \text{Min}[\mu_x(x), \mu_{R^k}(x, y)] \}$$
(2-45)

2. Komposisi Relasi Max-Prod (Jamshidi, 1993:81)

$$y^k = x * R^k$$

$$\mu_{y^k}(y) = \text{Max}_{x \in X} [\mu_x(x) \cdot \mu_{R^k}(x, y)]$$
(2-46)

3. Komposisi Relasi Min-Max (Jamshidi, 1993:82)

$$y^k = x + R^k$$

$$\mu_{y^k}(y) = \text{Min}_{x \in X} \{ \text{Max}[\mu_x(x), \mu_{R^k}(x, y)] \}$$
(2-47)

4. Komposisi Relasi Max-Max (Jamshidi, 1993:82)

$$y^k = x \circ R^k$$

$$\mu_{y^k}(y) = \text{Max}_{x \in X} \{ \text{Max}[\mu_x(x), \mu_{R^k}(x, y)] \}$$
(2-48)

5. Komposisi Relasi Min-Min (Jamshidi, 1993:82)

$$y^k = x \Delta R^k$$

$$\mu_{y^k}(y) = \text{Min}_{x \in X} \{ \text{Min}[\mu_x(x), \mu_{R^k}(x, y)] \}$$
(2-49)

6. Komposisi Relasi (p,q) (Jamshidi, 1993:82)

$$y^k = x \bullet p q R^k$$

$$\mu_{y^k}(y) = \text{Max}_{x \in X}(p) \{ \text{Min}(q) [\mu_x(x), \mu R^k(x, y)] \}$$
(2-50)

7. Komposisi Relasi Sum-Prod (Jamshidi, 1993:82)

$$y^k = x \times R^k$$

$$\mu_{y^k}(y) = f \left\{ \sum_{x \in X} [\mu_x(x) \cdot \mu R^k(x, y)] \right\}$$
(2-51)

8. Komposisi Relasi Max-Ave (Jamshidi, 1993:83)

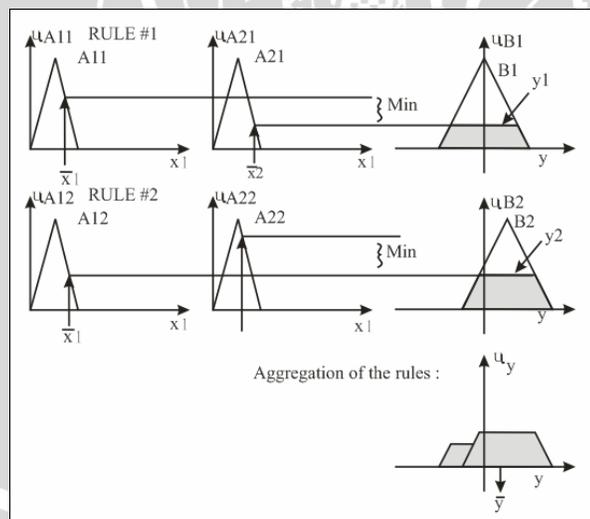
$$y^k = x \bullet av.R^k$$

$$\mu_{y^k}(y) = \frac{1}{2} \text{Max}_{x \in X} [\mu_x(x) + \mu R^k(x, y)]$$
(2-52)

Dari rumus-rumus komposisi relasi *fuzzy* bisa diperoleh logika pengambil keputusan. Metode untuk logika pengambil keputusan yang paling sering digunakan adalah metode Max-Min dan Max-Prod, dengan rumus sebagai berikut (Jamshidi, 1993:83) :

1. Metode Keputusan Max-Min (Jamshidi, 1993:83)

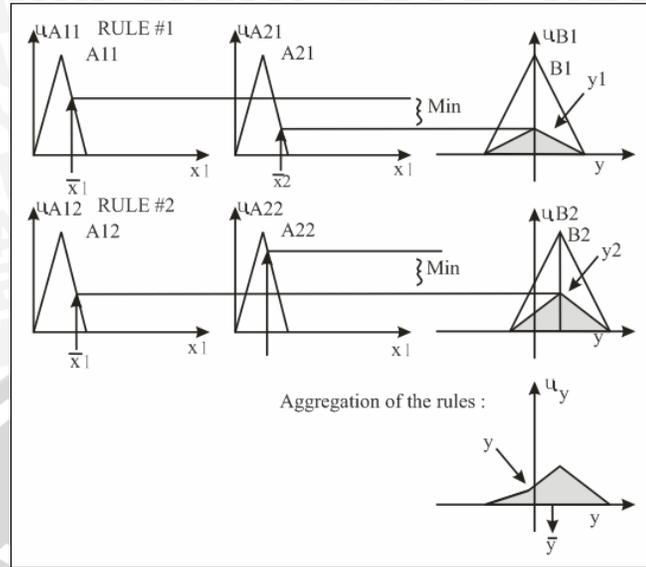
$$\mu_y(y) = \text{Max}_k \left\{ \text{Max}_{x \in X} \{ \text{Min} [\mu_x(x), \mu R^k(x, y)] \} \right\}$$
(2-53)



Gambar 2.26 Metode Keputusan Max-Min dengan Masukan Pasti.  
(Sumber : Jamshidi, 1993:99)

2. Metode Keputusan Max-Prod (Jamshidi, 1993:83)

$$\mu_y(y) = \text{Max}_k \left\{ \text{Max}_{x \in X} [\mu_x(x) \bullet \mu R^k(x, y)] \right\}$$
(2-54)



**Gambar 2.27 Metode Keputusan Max-Prod dengan Masukan Pasti.**  
(Sumber : Jamshidi, 1993:101)

### 2.8.10 Defuzzification.

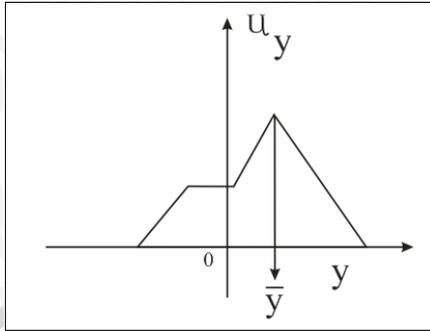
Bagian keluaran merupakan bagian untuk mengembalikan hasil dari pengambilan keputusan berdasarkan basis aturan yang ada ke dalam himpunan tegas. Hal ini dinamakan “defuzzification”. Rumus *defuzzification* adalah sebagai berikut (Jamshidi,1993:105) :

$$\bar{y} = DEFUZZ[\mu_y(y)] \quad (2-55)$$

*Defuzzification* memiliki banyak sekali metode, di antaranya adalah sebagai berikut (Jamshidi, 1993:105) :

1. Metode Max digunakan dalam kasus di mana fungsi keanggotaan berdasarkan nilainya memiliki titik puncak tunggal yang berhubungan dengan nilai himpunan tegas, dan nilai ini merupakan nilai puncak dari fungsi sehingga bisa dianggap perwakilan yang terbaik dari nilai *fuzzy* yang ada (Jamshidi, 1993:105).

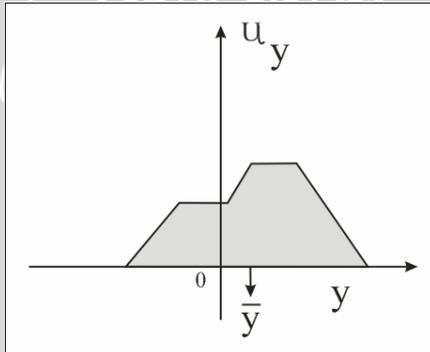
$$\text{Max}_{y \in Y} [\mu_y(y)] = \bar{y} \quad (2-56)$$



**Gambar 2.28 Defuzzification dengan Metode Max.**  
(Sumber : Jamshidi, 1993:106)

2. Metode Centroid merupakan metode yang mengambil bobot rata-rata dari fungsi keanggotaan atau Center of Gravity (pertengahan) dari area yang dilingkupi oleh kurva fungsi keanggotaan. Dengan menghitungnya berdasarkan rumus (Jamshidi, 1993:106) :

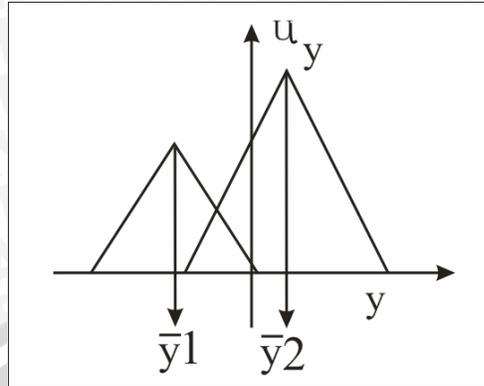
$$\bar{y} = \frac{\int uy(y) \cdot y dy}{\int uy(y) dy} \quad (2-57)$$



**Gambar 2.29 Defuzzification dengan Metode Centroid.**  
(Sumber : Jamshidi, 1993:107)

3. Metode Height hanya bisa digunakan dalam kasus di mana fungsi keanggotaan dari keluaran merupakan gabungan dari hasil penjumlahan dari fungsi yang simetris. Bisa dihitung dengan rumus (Jamshidi, 1993:107)

$$\bar{y} = \frac{\sum_{k=1}^{k=n} uy^k (\bar{y}^{-k}) \cdot \bar{y}^{-k}}{\sum_{k=1}^{k=n} uy^k (\bar{y}^{-k})} \quad (2-58)$$



**Gambar 2.30 Defuzzification dengan Metode Height.**  
(Sumber : Jamshidi, 1993:108)

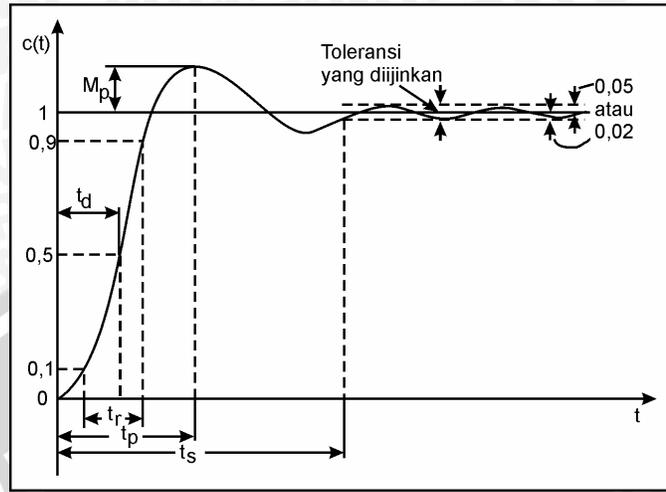
### 2.8.11 Respon Transien

Respon transien merupakan respon sistem yang berlangsung dari keadaan awal sampai keadaan akhir. Sedangkan respon keadaan tunak merupakan perilaku sistem jika  $t$  mendekati tak terhingga. Respon transien dan respon keadaan tunak merupakan bagian dari respon waktu sistem kontrol. Respon waktu sistem kontrol dapat digunakan sebagai metoda analisis untuk performansi sistem.

Respon transien sistem kontrol praktis sering menunjukkan osilasi teredam sebelum mencapai keadaan tunak. Dalam menentukan karakteristik respon transien sistem kontrol terhadap masukan tangga satuan, biasanya dicari parameter sebagai berikut (Ogata, 1995:238) :

1. Waktu tunda (delay time),  $t_d$  : Waktu yang diperlukan untuk mencapai setengah harga akhir yang pertama kali.
2. Waktu naik (rise time),  $t_r$  : Waktu yang diperlukan untuk naik dari 10 sampai 90%, 5 sampai 95%, atau 0 sampai 100% dari harga akhir.
3. Waktu puncak (peak time),  $t_p$  : Waktu yang diperlukan respon untuk mencapai puncak lewatan yang pertama kali.
4. (Persen) Lewatan maksimum,  $M_p$  : Harga puncak maksimum dari kurva respon yang diukur dari satu.
5. Waktu penetapan (settling time),  $t_s$  : Waktu yang diperlukan kurva respon untuk mencapai dan menetap dalam daerah di sekitar harga akhir yang ditentukan dengan presentase mutlak dari harga akhir (biasanya 5% atau 2%).

Secara grafis kurva respon bisa dilihat dalam Gambar 2.31.



Gambar 2.31 Kurva respon yang menunjukkan  $t_d$ ,  $t_r$ ,  $t_p$ ,  $M_p$  dan  $t_s$ .  
(Sumber : Ogata, 1995:238)



## BAB III METODOLOGI

Metodologi yang dilakukan pada perancangan dan pembuatan pengendalian kecepatan motor DC pada alat sablon satu warna otomatis dengan kontrol logika *fuzzy* menggunakan mikrokontroler AT89S52 meliputi berbagai hal. Diantaranya, pengumpulan data yang dilakukan melalui studi literatur, perancangan alat secara keseluruhan baik perancangan perangkat keras dan perancangan perangkat lunak, pembuatan alat, pengujian alat yang telah dibuat serta analisis data dari hasil pengujian, dan terakhir pengambilan kesimpulan dan saran.

### 3.1 Studi Literatur

Studi literatur yang dilakukan berupa pengumpulan data yang bersangkutan dengan komponen-komponen yang akan digunakan dalam perancangan dan pembuatan pengendalian kecepatan motor DC pada alat sablon satu warna otomatis dengan kontrol logika *fuzzy* menggunakan mikrokontroler AT89S52. Perolehan data dalam studi literatur diambil dari buku-buku literatur, informasi dari *data sheet*, serta informasi dari makalah dan paper yang berkaitan dengan perancangan dan pembuatan alat yang diperoleh secara *online* dari internet.

Studi literatur yang dilakukan meliputi data-data tentang :

1. Gambaran sablon secara umum.
2. Teori motor DC.
3. Teori mikrokontroler AT89S52.
4. Teori interface menggunakan PPI 8255.
5. Teori tentang PWM.
6. Teori tentang Tachogenerator DC
7. Teori tentang ADC 0808/0809.
8. Teori tentang Kontrol Logika *Fuzzy*.

### 3.2 Perancangan

Perancangan pengendalian kecepatan motor DC pada alat sablon satu warna otomatis dengan kontrol logika *fuzzy* menggunakan mikrokontroler AT89S52 secara keseluruhan meliputi perancangan perangkat keras dan perancangan perangkat lunak. Mulai dari perancangan blok diagram perangkat keras secara keseluruhan. Sehingga bisa diperoleh gambaran umum dari alat sablon satu warna otomatis yang akan dibuat. Gambaran umum akan menjelaskan tentang prinsip kerja alat secara keseluruhan dan spesifikasi komponen penyusun yang mungkin bisa digunakan agar alat yang dirancang akan bisa berfungsi sebagai alat sablon satu warna otomatis beserta pengendalian kecepatan motor DC dengan kontrol logika *fuzzy* di dalamnya.

Melalui gambaran umum tentang prinsip kerja alat dan spesifikasi komponen yang mungkin digunakan dibuat blok diagram sistem secara keseluruhan. Blok diagram sistem keseluruhan disusun oleh blok-blok diagram perbagian yang sesuai dengan fungsi komponen penyusunnya. Perancangan blok diagram perbagian disesuaikan dengan karakteristik komponen penyusun, variabel dan parameter yang ada sebagai masukan maupun keluaran disesuaikan dengan blok diagram pada bagian lain yang terhubung secara langsung.

Perancangan perangkat lunak dimaksudkan agar semua komponen yang digunakan bisa terkoordinasi dengan baik, serta bisa berfungsi sebagai pengendali kecepatan motor DC dengan kontrol logika *fuzzy* pada alat sablon satu warna otomatis. Selain itu perangkat lunak juga dibangun agar bisa melakukan perhitungan variabel masukan (tegangan) dan error yang ditimbulkan akibat selisih dari sensor dengan set point yang diinginkan, sehingga bisa diperoleh aksi kontrol yang mengendalikan variabel keluaran (kecepatan) melalui algoritma kontrol logika *fuzzy*.

#### 3.2.1 Pembuatan alat

Pembuatan alat dilakukan sesuai dengan perancangan yang telah disusun, sehingga pembuatan alat merupakan realisasi dari rancangan yang telah disusun. Pembuatan meliputi pembuatan PCB per-blok, penyoderan komponen pada PCB, penyambungan antar blok dengan kabel konektor.

Pembuatan alat juga meliputi pembangunan perangkat lunak. Perangkat lunak dibangun dengan menggunakan Bahasa *Assembly* dari mikrokontroler AT89S52, dan dibangun sesuai dengan algoritma dan diagram alir yang ada di perancangan.

### 3.3 Pengujian Alat

Pengujian alat dilakukan per-blok, pengujian dimaksudkan untuk mengetahui kinerja per-blok dari alat yang telah dibuat. Selain dilakukan pengujian per-blok juga dilakukan pengujian alat secara keseluruhan. Pengujian alat secara keseluruhan dimaksudkan untuk mengetahui kinerja alat keseluruhan sebagai alat sablon satu warna otomatis dengan pengendalian kecepatan motor DC melalui kontrol logika *fuzzy*.

Setelah dilakukan pengujian, dilakukan analisis data hasil pengujian dengan membandingkannya dengan data dari hasil perhitungan teori yang ada. Sehingga bisa diketahui kelemahan dan kelebihan alat yang telah dibuat.

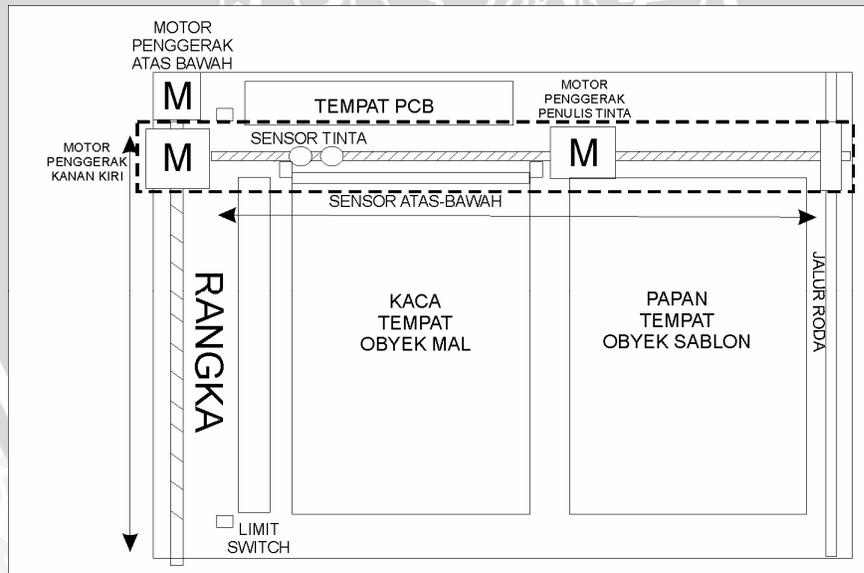
### 3.4 Penarikan Kesimpulan dan Saran

Penarikan kesimpulan dilakukan setelah melakukan analisis hasil pengujian per-blok maupun analisis hasil pengujian alat secara keseluruhan. Saran dilakukan setelah melihat adanya kekurangan dalam sistem yang telah dibuat, dengan harapan agar alat yang telah dibuat dapat dikembangkan lebih baik lagi.

## BAB IV PERANCANGAN

Perancangan yang akan dibahas meliputi perancangan perangkat keras dan perangkat lunak. Perancangan dimulai dengan merancang suatu alat yang berfungsi sebagai pengendalian kecepatan motor DC pada alat sablon satu warna otomatis dengan kontrol logika *fuzzy* menggunakan mikrokontroler AT89S52. Sehingga alat yang akan dirancang diharapkan memiliki spesifikasi dan kemampuan sebagai berikut :

1. Memiliki kemampuan untuk membuat obyek sablon yang sesuai dengan obyek yang diberikan pada mal.
2. Menggunakan motor DC sebagai penggerak.
3. Memiliki kemampuan untuk mengendalikan kecepatan motor DC dengan kontrol logika *fuzzy*.
4. Menggunakan mikrokontroler AT89S52 sebagai pengatur utama.



Gambar 4.1 Sistem Mekanik Alat.  
(Sumber : Perancangan)

Dari spesifikasi dan kemampuan alat yang telah ditentukan, dibuat sebuah sistem mekanik yang diharapkan bekerja sesuai dengan spesifikasi dan kemampuan. Rancangan sistem mekanik diperlihatkan dalam Gambar 4.1. Sistem mekanik yang akan dirancang memiliki prinsip kerja sebagai berikut :

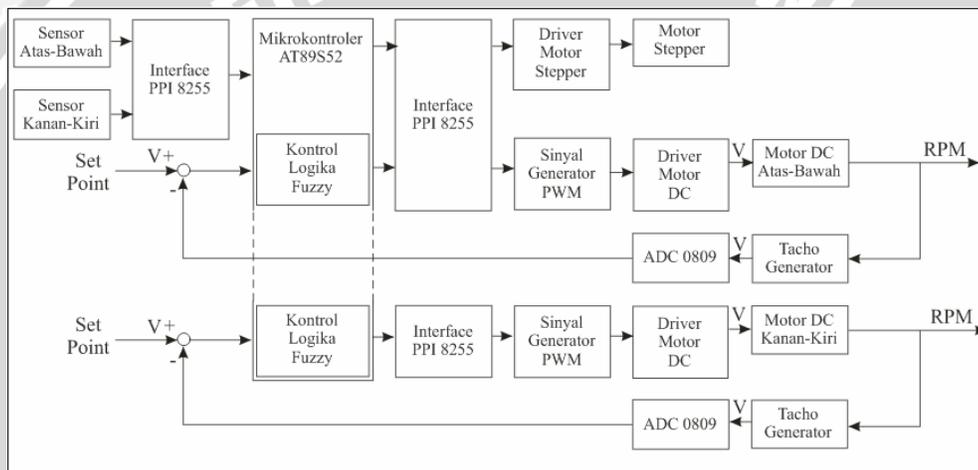
1. Obyek yang akan disablon diletakkan pada obyek mal. Media berupa kertas/plastik transparan dengan obyek gambar gelap. Tinta dengan warna yang diinginkan diisikan ke tempat penampung tinta.
2. Alat mulai bekerja dengan menjalankan motor DC penggerak atas bawah. Ketika bergerak sensor atas-bawah akan aktif. Sensor ini meliputi seluruh lebar obyek mal. Ketika ada obyek gambar berwarna gelap, sensor akan mendeteksinya. Masukan dari sensor ini akan menghentikan motor DC penggerak atas-bawah, dan menjalankan motor DC penggerak kanan-kiri.
3. Motor DC penggerak kanan-kiri bergerak sesuai dengan lebar mal. Kecepatan motor DC penggerak kanan-kiri akan melambat bila sensor pertama mendeteksi obyek gambar gelap, ketika sensor kedua mendeteksi obyek gambar gelap motor *stepper* penulis tinta akan bekerja.
4. Ketika sensor pertama tidak lagi mendeteksi adanya obyek gelap, kecepatan motor DC penggerak kanan-kiri akan konstan dengan kecepatan lambat. Ketika sensor kedua sudah tidak lagi mendeteksi obyek gelap, selain menghentikan motor *stepper* penulis tinta juga akan menjadikan motor DC penggerak kanan-kiri bergerak dengan kecepatan lebih tinggi.
5. Motor DC penggerak kanan-kiri akan berbalik arah jika *limit switch* di kanan obyek mal tersentuh. Dan akan berhenti bila *limit switch* di kiri tersentuh.
6. Ketika *limit switch* di kiri obyek mal tersentuh motor DC penggerak atas bawah akan bergerak, dan tidak merespon obyek gelap yang telah diterima. Sensor akan aktif lagi setelah beberapa saat. Bila merespon obyek gelap akan mengulangi langkah 2, bila tidak ada obyek gelap lagi maka akan terus bergerak sampai *limit switch* di bawah tersentuh.
7. Ketika *limit switch* di bawah tersentuh, motor DC penggerak atas bawah akan bergerak berbalik arah sampai menyentuh *limit switch* di atas dan

tidak menghiraukan obyek gambar gelap lagi. Motor DC penggerak atas-bawah akan berhenti bergerak ketika *limit switch* di atas tersentuh.

8. Kecepatan motor DC penggerak atas-bawah maupun motor DC penggerak kanan-kiri dikendalikan dengan kontrol logika *fuzzy*, sehingga kecepatan akan konstan sesuai dengan set point yang telah ditentukan sebelumnya.
9. Alat akan menunggu perintah selanjutnya.

#### 4.1 Blok Diagram Sistem

Berdasarkan prinsip kerja alat, perancangan dilanjutkan dengan merancang blok diagram sistem. Blok diagram sistem diberikan dalam Gambar 4.2.



**Gambar 4.2 Blok Diagram Sistem.**  
(Sumber : Perancangan)

Blok diagram sistem keseluruhan dibangun atas blok diagram perbagian yang meliputi blok diagram mikrokontroler AT89S52, blok diagram *interface* PPI 8255, blok diagram sinyal *generator pulse width modulation* (PWM), blok diagram *driver* motor *stepper* dan motor DC, blok diagram motor penggerak, blok diagram *tachogenerator*, blok diagram konverter ADC 0809, dan blok diagram sensor atas-bawah dan sensor kanan kiri.

Blok diagram mikroprosesor AT89S52 disusun oleh minimum sistem dari mikrokontroler AT89S52. Blok diagram ini terhubung langsung dengan blok diagram *interface* PPI 8255, dan blok diagram ADC 0809. Di dalam blok diagram terdapat perangkat lunak yang berfungsi sebagai kontrol logika *fuzzy*.

Blok diagram *interface* PPI 8255 merupakan blok diagram dari komponen antarmuka PPI 8255. Blok diagram ini berfungsi untuk menambah port I/O dari mikrokontroler AT89S52. Masukan dari blok diagram *interface* PPI adalah blok diagram sensor atas-bawah dan blok diagram kanan-kiri. Keluaran dari blok diagram PPI 8255 adalah blok diagram *driver* motor *stepper* dan blok diagram sinyal *generator* PWM.

Blok diagram *driver* motor *stepper* disusun oleh komponen IC ULN 2065. IC ini berfungsi untuk menguatkan arus dan menggerakkan motor *stepper* sesuai dengan data yang dikirim dari blok diagram mikrokontroler AT89S52 melalui blok diagram *interface* PPI 8255.

Blok diagram sinyal *generator* PWM berfungsi untuk menghasilkan sinyal dengan lebar pulsa sesuai dengan data yang dikirim dari blok diagram mikrokontroler AT89S52 melalui blok diagram *interface* PPI 8255. Sinyal yang dihasilkan merupakan keluaran dari blok diagram sinyal *generator* PWM dan dikirim ke blok diagram *driver* motor DC.

Blok diagram *driver* motor DC berfungsi untuk menggerakkan motor DC. Pengendalian kecepatan motor DC ditentukan dengan sinyal dari blok diagram sinyal *generator* PWM.

Blok diagram motor penggerak terdiri atas motor *stepper* sebagai penggerak penulis tinta, motor DC atas-bawah sebagai penggerak dengan arah ke atas dan ke bawah, dan motor DC kanan-kiri yang berfungsi sebagai penggerak dengan arah ke kanan dan ke kiri.

Blok diagram *tachogenerator* berfungsi sebagai sensor kecepatan motor DC. Masukan blok diagram ini berupa kecepatan putaran motor DC. Keluaran blok diagram ini berupa tegangan dengan nilai yang mewakili kecepatan motor DC.

Blok diagram ADC 0809 berfungsi sebagai konverter tegangan analog ke digital. Tegangan analog sebagai masukan berasal dari blok diagram *tachogenerator*. Tegangan digital sebagai keluaran dikirim ke blok diagram mikrokontroler AT89S52.

Blok diagram sensor atas-bawah terdiri dari 2 *limit switch* dan rangkaian saklar cahaya. *Limit switch* berfungsi untuk membalik arah putaran motor DC,

sedangkan saklar cahaya berfungsi untuk menghentikan motor DC penggerak atas bawah dan mengaktifkan motor DC penggerak kanan-kiri.

Blok diagram kanan-kiri terdiri dari 2 *limit switch* dan 2 rangkaian saklar cahaya inframerah. *Limit switch* berfungsi untuk membalik arah putaran motor DC. *Limit switch* di sebelah kiri juga berfungsi untuk menghentikan motor DC penggerak kanan-kiri dan mengaktifkan motor DC penggerak atas-bawah. 2 rangkaian saklar cahaya berfungsi untuk menentukan kecepatan motor DC penggerak kanan-kiri, dengan memilih data set point berupa tegangan dalam bentuk sinyal tegangan digital yang disimpan dalam perangkat lunak mikrokontroler AT89S52.

Blok diagram pengendalian kecepatan motor DC disusun atas beberapa blok diagram sehingga bisa membentuk sebuah sistem loop tertutup. Blok diagram tersebut antara lain: blok diagram kontrol logika *fuzzy* yang merupakan bagian perangkat lunak dari blok diagram mikrokontroler AT89S52. Blok diagram sinyal *generator* PWM sebagai penghasil sinyal dengan lebar pulsa yang ditentukan melalui kontrol logika *fuzzy*. Blok diagram *driver* motor DC sebagai penggerak motor DC. Blok diagram motor DC sebagai plant yang akan dikendalikan, masukan berupa tegangan dengan keluaran berupa kecepatan putar (*Rotation per minute* (RPM)). Blok diagram *tachogenerator* DC yang berfungsi mengubah variabel RPM ke variabel tegangan. Blok diagram ADC 0809 mengkonversi tegangan analog dari *tachogenerator* DC ke tegangan digital yang sesuai dengan mikrokontroler AT89S52. Set point merupakan data yang disimpan dalam perangkat lunak mikrokontroler AT89S52, data set point dipilih melalui rangkaian saklar cahaya dalam blok diagram sensor kanan-kiri. Error diperoleh dari hasil pengurangan set point dengan tegangan dari ADC 0809. Error dan perubahan error merupakan masukan kontrol logika *fuzzy*, nilai error dan perubahan error mempengaruhi aksi kontrol logika *fuzzy*. Aksi kontrol logika *fuzzy* diperoleh dari basis aturan yang ada dalam kontrol logika *fuzzy*. Aksi kontrol logika *fuzzy* berupa bilangan biner 4 bit yang merupakan penentu lebar pulsa dari sinyal yang dihasilkan oleh sinyal *generator* PWM.

## 4.2 Perancangan Perangkat Keras

### 4.2.1 Perancangan Minimum Sistem AT89S52

Agar mikrokontroler AT89S52 bisa berfungsi dengan baik, maka diperlukan beberapa rangkaian pendukung seperti yang terlihat dalam Gambar 4.3. Rangkaian *clock internal* terdiri dari rangkaian oscilator kristal dan kondensator. Mengacu pada Atmel Corporation, 2003:10, bahwa setiap satu *cycle* membutuhkan frekuensi  $\frac{1}{12}$  frekuensi oscilator. Maka untuk memudahkan perhitungan setiap *cycle*-nya digunakan oscilator kristal 12 MHz. Sehingga setiap *cycle*-nya akan memiliki frekuensi sebesar 1 MHz atau periode sebesar  $\frac{1}{1\text{MHz}} = 1 \mu\text{s}$ . Kemudian mengacu pada Atmel Corporation, 2003:18, jika menggunakan oscilator kristal, maka besarnya C1 dan C2 adalah  $30 \text{ pF} \pm 10 \text{ pF}$ . Sehingga kondensator 33 pF yang mudah didapatkan dipasaran bisa digunakan. Sedangkan untuk rangkaian power-on reset mengacu pada Atmel Corporation, 1997:62 reset akan terjadi bila logika tinggi diberikan pada kaki RST minimal selama 2 *cycle* atau  $2 \mu\text{s}$ . Serta mengacu pada Atmel Corporation, 1997:63 rangkaian RC seri dengan besar  $R = \pm 8,2 \text{ K}\Omega$  dan  $C 10 \mu\text{F}$  sudah memenuhi syarat minimal ini. Penggantian R dengan nilai sebesar  $10 \text{ K}\Omega$  masih memenuhi syarat minimal, hal ini sesuai dengan rumus (Mismail, 1997:10) :

$$v = V e^{-t/RC} \quad (4-1)$$

Dengan :

$$v = 0,7 \times V_{cc} = 0,35 \text{ V}$$

$$V = 5 \text{ V}$$

$$R = 10 \text{ K}\Omega$$

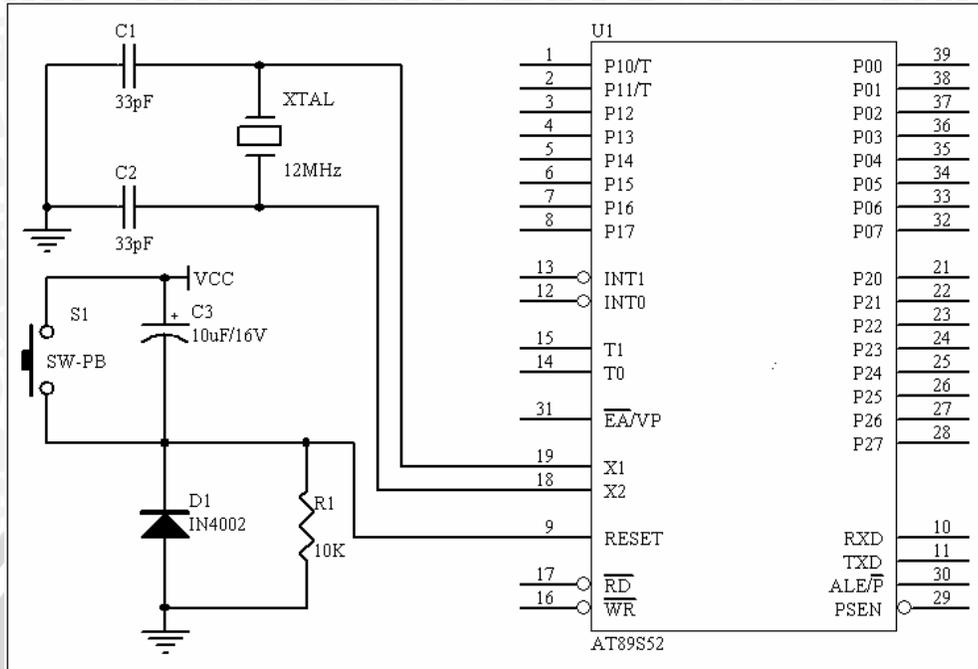
$$C = 10 \mu\text{F}$$

Sehingga rumus untuk mencari waktu (t) menjadi (Mismail, 1997:10) :

$$t = RC \ln \frac{V}{v} \quad (4-2)$$

$$t = 10 \text{ K}\Omega \times 10 \mu\text{F} \ln \frac{5\text{V}}{0,35\text{V}} \quad (4-3)$$

$$t = 0,035\text{s}$$



**Gambar 4.3 Rancangan Rangkaian Minimum Sistem AT89S52.**  
(Sumber : Perancangan)

#### 4.2.2 Perancangan Antar muka PPI 8255

Antar muka PPI 8255 digunakan untuk menambah port AT89S52, sebenarnya port AT89S52 yang berjumlah 4 port mencukupi untuk digunakan dalam kebutuhan port I/O rangkaian. Tapi karena rangkaian ADC 0809 membutuhkan rangkaian alamat pemilih masukan dan pengaktif komponen (*chip select* ( $\overline{CS}$ )), maka kaki  $\overline{RD}$  dan  $\overline{WR}$  (yang merupakan kaki pada bagian port 3) digunakan sebagai pengaktif komponen ADC 0809. Sehingga jumlah efektif port I/O AT89S52 menjadi berkurang, untuk mengatasi hal itu digunakan PPI 8255. Rangkaian antar muka PPI 8255 membutuhkan penahan alamat yang membedakan jalur alamat dengan jalur data dari port 0 AT89S52, penahan alamat menggunakan IC 74LS573. Dan untuk mengaktifkan komponen PPI 8255 menggunakan IC 74LS138 sebagai penyandi/demultiplexer.

Penahan 74LS573 ketika kaki LE mendapat masukan dari ALE (AT89S52) akan menahan data yang masuk dari port 0 AT89S52. Karena kaki  $\overline{OE}$  ditanahkan maka keluaran akan sesuai dengan data yang ditahan. Sehingga

bisa digunakan untuk menyandi alamat pada PPI 8255 sesuai dengan fungsinya (melalui *Control Word*).

Penyandi 74LS138 menggunakan masukan dari port 2 (kaki 2.5-kaki2.7) dan sesuai dengan Tabel 4.1 yang merupakan tabel kebenaran dari 74LS138, maka kaki keluaran dengan logika rendah digunakan sebagai pengaktif PPI 8255.

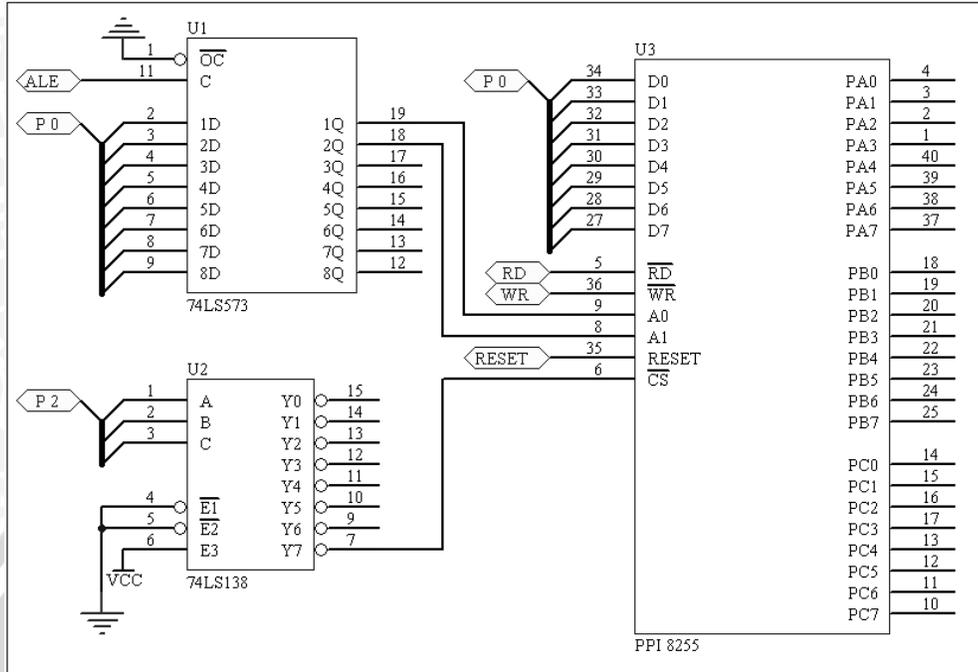
**Tabel 4.1 Kebenaran 74LS138.**

INPUTS						TRUTH TABLE							
E <sub>1</sub>	E <sub>2</sub>	E <sub>3</sub>	A <sub>0</sub>	A <sub>1</sub>	A <sub>2</sub>	0̄ <sub>0</sub>	0̄ <sub>1</sub>	0̄ <sub>2</sub>	0̄ <sub>3</sub>	0̄ <sub>4</sub>	0̄ <sub>5</sub>	0̄ <sub>6</sub>	0̄ <sub>7</sub>
H	X	X	X	X	X	H	H	H	H	H	H	H	H
X	H	X	X	X	X	H	H	H	H	H	H	H	H
X	X	L	X	X	X	H	H	H	H	H	H	H	H
L	L	H	L	L	L	L	H	H	H	H	H	H	H
L	L	H	H	L	L	H	L	H	H	H	H	H	H
L	L	H	L	H	L	H	H	L	H	H	H	H	H
L	L	H	H	H	L	H	H	H	L	H	H	H	H
L	L	H	L	L	H	H	H	H	H	L	H	H	H
L	L	H	H	L	H	H	H	H	H	H	L	H	H
L	L	H	L	H	H	H	H	H	H	H	H	L	H
L	L	H	H	H	H	H	H	H	H	H	H	L	L

H = HIGH Voltage Level  
L = LOW Voltage Level  
X = Don't Care

Sumber : ON Semiconductor, 2000:87.

PPI 8255 difungsikan sebagai perangkat antar muka dengan fungsi untuk port I/O tambahan. Dengan demikian mode yang dipilih adalah mode 0. Mode 0 menjadikan fungsi PPI 8255 sebagai port I/O biasa. Dengan fungsi port A sebagai jalur masukan dan port B, port C *upper* dan port C *lower* sebagai keluaran. Sehingga *control word* yang dimasukkan ke dalam program PPI adalah data 90H. Gambar rancangan rangkaian antar muka PPI 8255 diberikan dalam Gambar 4.4.



Gambar 4.4 Rancangan Rangkaian PPI 8255  
(Sumber : Perancangan)

### 4.2.3 Perancangan Rangkaian PWM

*Pulse Width Modulator* (PWM) merupakan pengendali kecepatan motor dengan mengatur persentase lama waktu logika tinggi dan logika rendah. Persentase 0% berarti lama logika rendah adalah 100% atau dengan kata lain motor DC tidak akan mendapat tegangan sumber sama sekali, sedangkan persentase 100% berarti lama logika tinggi adalah 100% atau dengan kata lain motor DC akan terhubung dengan sumber sepanjang waktu. Rangkaian PWM disusun atas pewaktu, *counter* / penghitung, penahan sinyal dan *comparator* / pembanding.

Rangkaian pewaktu menggunakan IC 74LS14 yang merupakan *Hex Inverter/Schmitt Triggers Dual Gate*. Besarnya frekuensi yang dibangkitkan dihitung berdasar rumus (Adler dan Surtell, 2003:5) :

$$f_{osc} = \frac{1}{2\pi RC} \tag{4-4}$$

Dengan menggunakan nilai  $R = 47K\Omega$  dan  $C = 1nF$ , maka besarnya frekuensi yang dibangkitkan adalah :

$$f_{osc} = \frac{1}{2 \times 3,14 \times 47 \cdot 10^3 \times 1 \cdot 10^{-9}}$$

$$f_{osc} = 3388 \text{ Hz}$$

Rangkaian *counter* / penghitung menggunakan IC 74LS161 yang merupakan penghitung 4 bit atau 16 bilangan. Karena untuk menyelesaikan satu putaran harus menyelesaikan 16 bilangan, maka besarnya frekuensi penghitung adalah  $\frac{1}{16}$  frekuensi oscilator. Sehingga besarnya frekuensi penghitung adalah :

$$f_c = \frac{f_{osc}}{16} \text{ Hz} \quad (4-5)$$

$$f_c = \frac{3388}{16} \text{ Hz}$$

$$f_c = 211,75 \text{ Hz}$$

Dan periode yang dibutuhkan untuk satu bilangan adalah :

$$T_c = \frac{1}{f_c} \quad (4-6)$$

$$T_c = \frac{1}{211,75 \text{ Hz}}$$

$$T_c = 4,72 \text{ ms}$$

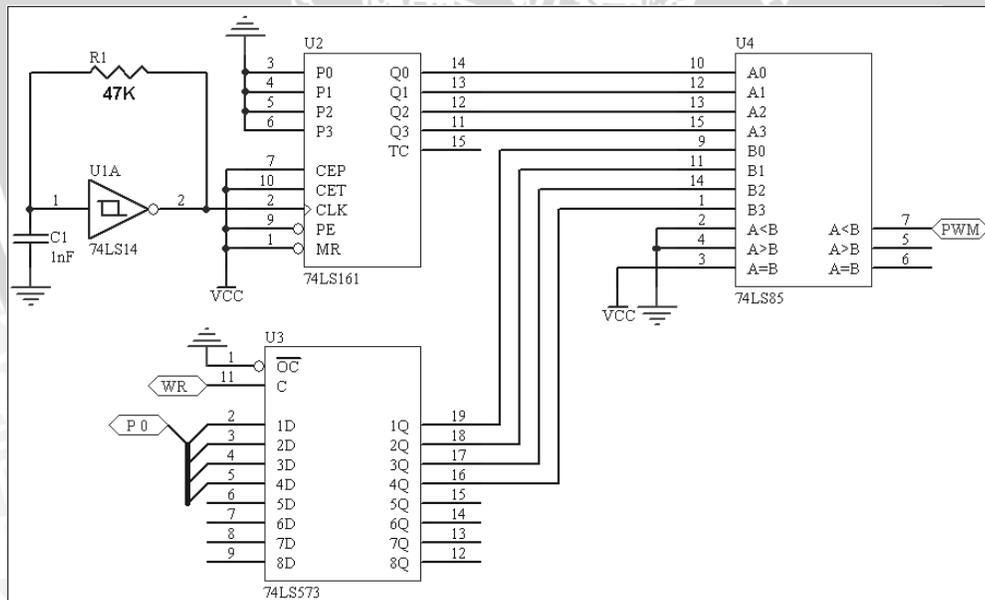
Data dari penghitung 74LS161 akan dibandingkan dengan data dari penahan 74LS573. Data penahan dari 74LS573 diperoleh dari data yang dikirim oleh mikrokontroler AT89S52. Data ini merupakan data hasil dari aksi kontrol yang dilakukan oleh kontrol logika *fuzzy*. Pembanding menggunakan IC 74LS85 yang merupakan *comparator* / pembeding 4 bit.

Besarnya sinyal PWM yang dikeluarkan bergantung dari sinyal yang dikeluarkan oleh pembeding 74LS85. Dengan menggunakan pembeding  $A < B$ , dimana A adalah data dari penghitung 74LS161 dan B adalah data dari penahan 74LS573.

Sehingga ketika data dari penahan 74LS573 (nilai dari B) adalah 0 (0000 dalam bentuk 4 bit) maka logika rendah yang akan selalu keluar dari kaki  $A < B$  pada pembeding 74LS85. Karena data dari penghitung 74LS161 (nilai dari A)

yang terkecil adalah 0 (0000 dalam bentuk 4 bit). Dengan demikian sinyal PWM yang akan keluar adalah  $\frac{0}{16} \times 100\% = 0\%$ .

Dan ketika data dari penahan 74LS573 (nilai dari B) adalah 16 (1111 dalam bentuk 4 bit) maka logika tinggi yang akan keluar lebih sering dari kaki  $A < B$  pada pembandingan 74LS85. Karena data dari penghitung 74LS161 (nilai dari A) yang terbesar adalah 16 (1111 dalam bentuk 4 bit). Dengan demikian sinyal PWM yang akan keluar adalah  $\frac{15}{16} \times 100\% = 93\%$ . Sinyal PWM yang keluar tidak bisa mencapai 100% karena ketika data dari penahan 74LS573 yang memiliki nilai terbesar 16 (1111 dalam bentuk 4 bit) tidak bisa melebihi nilai 16 (1111 dalam bentuk 4 bit) dari data dari penghitung 74LS161. Sehingga persamaan  $A < B$  hanya bisa dipenuhi ketika data dari penghitung berada di bawah nilai 16, atau  $\frac{15}{16}$  kali. Dengan demikian pengendalian kecepatan motor DC akan menggunakan set point untuk kecepatan maksimum berada pada 93% berdasarkan sinyal PWM maksimum. Rancangan rangkaian pembangkit sinyal PWM diberikan dalam Gambar 4.5.



Gambar 4.5 Rancangan Rangkaian Pembangkit Sinyal PWM.  
(Sumber : Perancangan)

#### 4.2.4 Perancangan *Driver Motor DC*

*Driver* motor DC menggunakan rangkaian Jembatan-H dengan memanfaatkan 2 transistor PNP TIP 127 dan 2 transistor NPN TIP 122. Keempat dibagi dalam 2 pasang, masing-masing pasangan terdiri dari transistor PNP TIP 127 dan transistor NPN TIP 122. Pasangan transistor tersebut difungsikan sebagai saklar yang bekerja bergantian sehingga bisa menggerakkan motor ke kanan dan ke kiri.

Pasangan transistor dikendalikan melalui sinyal DIR yang berasal dari mikrokontroler AT89S52, dan diaktifkan melalui rangkaian PWM. Sehingga rangkaian ini bisa berfungsi untuk mengatur arah putaran sekaligus mengendalikan kecepatan melalui sinyal dari rangkaian PWM. Agar rangkaian ini bisa bekerja melalui tegangan dari mikrokontroler dan rangkaian PWM, maka dibutuhkan rangkaian pendukung.

Rangkaian pendukung juga merupakan kombinasi dari 2 transistor PNP dan 4 transistor NPN yang berfungsi sebagai saklar bagi rangkaian jembatan-H. Transistor NPN yang digunakan adalah 2N2222, sedangkan transistor PNP yang digunakan adalah 2N2907. Selain transistor juga menggunakan IC 74LS04 dan 74LS00, kedua IC ini digunakan agar sinyal yang dikirimkan ke rangkaian jembatan-H tidak sampai mengaktifkan keempat transistor secara bersamaan.

Untuk memfungsikan transistor TIP 127 sebagai saklar, arus basis dengan data dari datasheet bahwa  $I_{c_{max}} = 8 \text{ A}$  dan  $h_{FE} = 1000$  bisa dihitung dengan (Kamal, 2007:4) :

$$I_b = I_c / h_{FE} \quad (4-7)$$

$$I_b = 8A / 1000$$

$$I_b = 8\text{mA}$$

Ketika kondisi saturasi  $V_{BE} = 2,5 \text{ V}$ , dengan  $V_{CC} = 12 \text{ V}$ , maka  $R_b$  (Kamal, 2007:4)

$$R_b = \frac{V_{CC} - V_{BE}}{I_b} \quad (4-8)$$

$$R_b = \frac{12V - 2,5V}{8\text{mA}}$$

$$R_b = 1187 \Omega$$

$$R_b \approx 1000 \Omega$$

Nilai  $R_b$  merupakan nilai  $R_c$  pada transistor 2N2907. Dengan  $V_{CC} = 5V$ ,  $V_{CE} = 12V$  dan data dari datasheet bahwa saat saturasi  $V_{CE} = 1,6V$  dan  $V_{BE} = 1,3V$ , maka besar  $I_c$  adalah (Kamal, 2007 : 5)

$$I_c = \frac{V_{CE} - V_{CE(SAT)}}{R_c} \quad (4-9)$$

$$I_c = \frac{12V - 1,6V}{1000\Omega}$$

$$I_c = 10,4 \text{ mA}$$

Dengan  $I_c \approx 10 \text{ mA}$  berdasar datasheet besar  $h_{FE} \text{ min} = 10$ , sehingga  $I_b$  bisa dihitung dengan Rumus 4-7

$$I_b = 10\text{mA} / 10$$

$$I_b = 1 \text{ mA}$$

Besar nilai  $R_b$  setelah dihitung dengan Rumus 4-8.

$$R_b = \frac{5V - 1,3V}{1\text{mA}}$$

$$R_b = 3700 \Omega$$

$$R_b \approx 3900 \Omega$$

Karena sumber  $V_{CC}$  sebesar  $12V$ , sedang besar  $V_{BE}$  maksimum =  $6V$  maka diperlukan rangkaian pembagi tegangan. Pembagi tegangan ini menggunakan 2 resistor yang sama besar yaitu  $3900 \Omega$ . Salah satu resistor sebagai  $R_b$  di transistor 2N2907 dan satunya lagi sebagai  $R_c$  di transistor 2N2222.

Untuk mencari nilai  $R_b$  pada 2N2222 dengan nilai  $R_c = 3900 \Omega$ ,  $V_{CC} = 5V$  dan  $V_{CE} = 12V$ , terlebih dahulu dihitung nilai  $I_c$ , dengan data kondisi saturasi dari datasheet bahwa  $V_{CE} = 1V$  dan  $V_{BE} = 2V$ . Besarnya nilai  $I_c$  bisa dihitung dengan Rumus 4-9.

$$I_c = \frac{12V - 1V}{3900\Omega}$$

$$I_c = 2,82 \text{ mA}$$

Dengan  $I_c = 2,82 \text{ mA}$ , maka besar  $h_{FE}$  minimum berdasar datasheet adalah 10, sehingga  $I_b$  bisa dihitung dengan Rumus 4-7.

$$I_b = 2,82\text{mA} / 10$$

$$I_b = 0,28 \text{ mA}$$

Sehingga  $R_b$  setelah dihitung dengan Rumus 4-8.

$$R_b = \frac{5V - 2V}{2,82mA}$$

$$R_b = 10638,29 \Omega$$

$$R_b \approx 10000 \Omega$$

Untuk memfungsikan transistor TIP 122 sebagai saklar, arus basis  $I_b$  dengan data dari datasheet bahwa  $I_{c_{max}} = 8 \text{ A}$  dan  $h_{FE} = 1000$  bisa dihitung dengan Rumus 4-7.

$$I_b = 8A/1000$$

$$I_b = 8mA$$

Ketika kondisi saturasi  $V_{BE} = 2,5 \text{ V}$ , dengan  $V_{CC} = 12 \text{ V}$ , maka  $R_b$  bisa dihitung dengan Rumus 4-8.

$$R_b = \frac{12V - 2,5V}{8mA}$$

$$R_b = 1187 \Omega$$

$$R_b \approx 1000 \Omega$$

Nilai  $R_b$  merupakan nilai  $R_c$  pada transistor 2N2222. Dengan  $V_{cc} = 5V$ ,  $V_{CE} = 12 \text{ V}$  dan data dari datasheet bahwa saat saturasi  $V_{CE} = 1,6 \text{ V}$  dan  $V_{BE} = 1,3 \text{ V}$ , maka besar  $I_c$  bisa dihitung dengan Rumus 4-9.

$$I_c = \frac{12V - 1,6V}{1000\Omega}$$

$$I_c = 10,4 \text{ mA}$$

Dengan  $I_c \approx 10 \text{ mA}$  berdasar datasheet besar  $h_{FE \text{ min}} = 10$ , sehingga  $I_b$  bisa dihitung dengan Rumus 4-7.

$$I_b = 10mA/10$$

$$I_b = 1 \text{ mA}$$

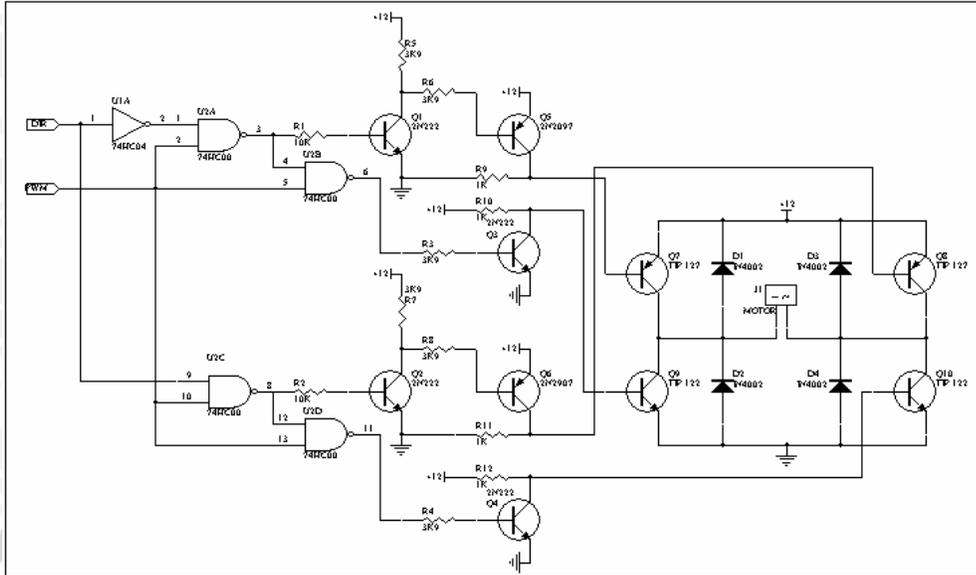
Besar nilai  $R_b$  setelah dihitung dengan Rumus 4-8.

$$R_b = \frac{5V - 1,3V}{1mA}$$

$$R_b = 3700 \Omega$$

$$R_b \approx 3900 \Omega$$

Gambar rancangan rangkaian *driver* motor DC diberikan dalam Gambar 4.6.



Gambar 4.6 Rancangan Rangkaian *Driver Motor DC*.  
(Sumber : Perancangan)

#### 4.2.5 Perancangan Rangkaian ADC 0809

Rangkaian ADC 0809 digunakan untuk mengkonversi tegangan analog dari *tachogenerator* DC menjadi tegangan digital 8 bit agar bisa digunakan pada mikrokontroler AT89S52. ADC 0809 memiliki kemampuan untuk mengubah tegangan analog 0- 5 Volt menjadi tegangan digital 8 bit. Karena sesuai dengan National Semiconductor, 2002:9, besarnya tegangan referensi ( $V_{ref}$ ) sama dengan besarnya tegangan masukan ( $V_{cc}$ ), sehingga  $V_{ref(+)} = V_{cc} = 5,12$  (max). Dengan demikian besarnya tiap langkah konversi atau 1 LSB bisa dihitung dengan rumus (National Semiconductor, 2002:9) :

$$1LSB = \frac{V_{IN}}{D_{MAX} - D_{MIN}} \quad (4-10)$$

Dengan  $D_{MAX} = 256$ ,  $D_{MIN} = 0$ , dan  $V_{IN} = 5$  V, maka besar tiap 1 LSB adalah :

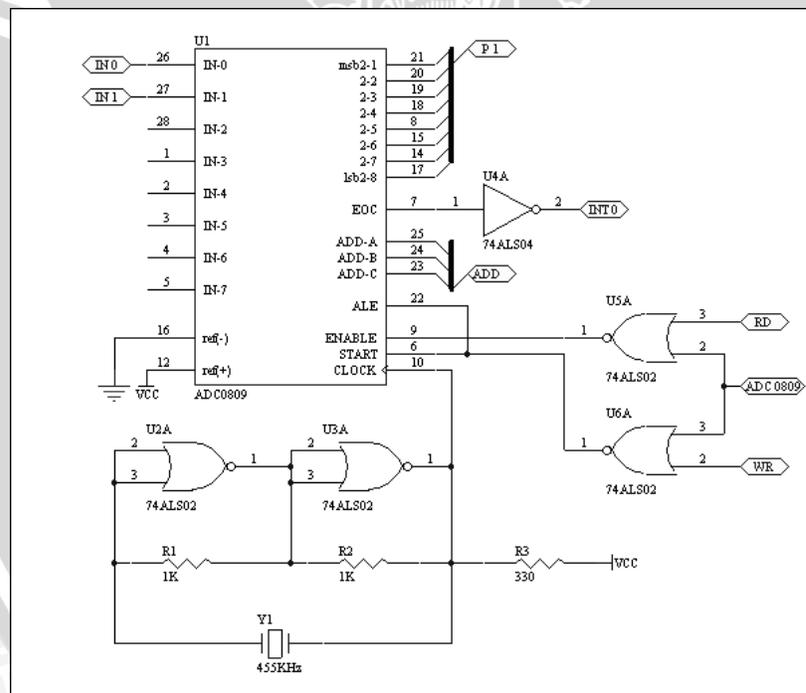
$$1LSB = \frac{5V}{256 - 0}$$

$$1LSB = 19,5mV$$

Besarnya 1 LSB membutuhkan tegangan sebesar 19,5 mV, sehingga setiap perubahan tegangan masukan sebesar 19,5 mV akan mengubah keluaran digital sebesar 1 bilangan.

Waktu yang diperlukan untuk melakukan konversi berkisar antara 1,6  $\mu$ s-1  $\mu$ s. Hal ini karena ADC bekerja efektif dengan rangkaian *clock* dengan frekuensi antara 640 KHz-1MHz (National Semiconductor, 2002:4). Frekuensi ini dibangkitkan melalui rangkaian oscillator.

ADC 0809 memiliki 8 jalur tegangan masukan, sehingga untuk memilih salah satu jalur yang akan digunakan memerlukan penyandi alamat. Karena tidak memiliki pengaktif komponen (*chip select*), maka memerlukan suatu rangkaian pendukung agar bisa berfungsi dengan baik. Rangkaian pendukung yang digunakan merupakan rangkaian gerbang NOR, sehingga akan aktif bila mendapat masukan  $\overline{RD}$  atau  $\overline{WR}$  dan satu masukan sebagai pengaktif dengan logika rendah. Pengaktif dengan logika rendah diperoleh dari IC 74LS138 penyandi/demultiplexer. Rancangan rangkaian ADC 0809 diberikan dalam Gambar 4.7.

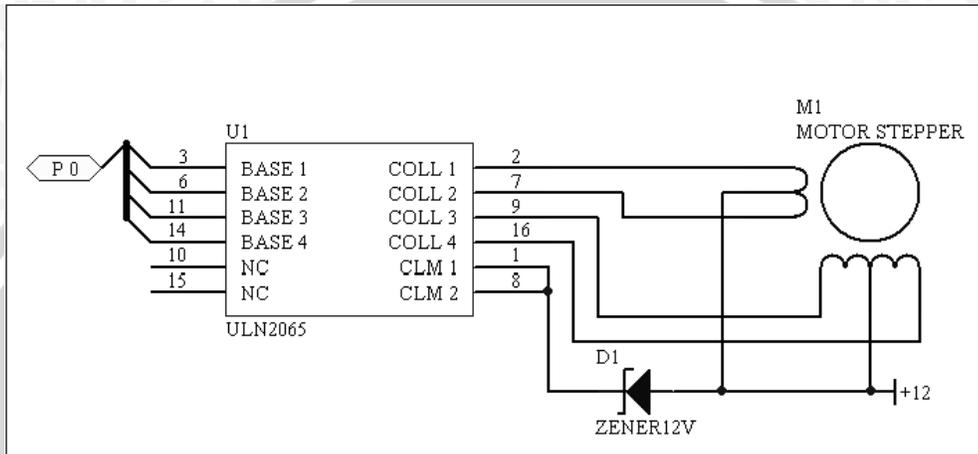


Gambar 4.7 Rancangan Rangkaian ADC 0809.  
(Sumber : Perancangan)

#### 4.2.6 Perancangan Driver Motor Stepper

Driver motor stepper menggunakan IC ULN 2065, masukan IC ini dari tegangan TTL 5V. Bisa digunakan untuk tegangan sampai dengan 50 V dan arus

sampai dengan 1,5 A. Sangat ideal digunakan untuk *Driver* motor *stepper* dengan arus sampai dengan 1 A. Pemasangan diode zener 12 V dimaksudkan untuk mencegah tegangan balik merusak IC ULN 2065. Tegangan balik diakibatkan oleh medan magnet kumparan motor *stepper* ketika tegangan sumber diputuskan. Rancangan rangkaian *Driver* motor *stepper* menggunakan IC ULN 2065 diberikan dalam Gambar 4.8.



Gambar 4.8 Rancangan *Driver* Motor *Stepper* ULN 2065.  
(Sumber : Perancangan)

#### 4.2.7 Perancangan Sensor Obyek Gambar

Sensor obyek gambar dimaksudkan untuk memberikan masukan logika rendah ke mikrokontroler ketika cahaya LED menuju sensor obyek gambar terhalang. Sensor cahaya menggunakan photodiode yang merupakan komponen peka cahaya. Ketika cahaya LED mengenai photodiode maka tegangan sebesar 5 V atau logika tinggi akan masuk ke mikrokontroler, sedangkan ketika cahaya LED terhalang maka tegangan keluaran dari photodiode sebagai sensor obyek gambar akan mendekati nol atau logika rendah.

Arus maksimum yang diperbolehkan untuk mengalir ke LED adalah sebesar 20 mA,  $V_{cc}$  maksimum 5 V, dan tegangan jatuh LED merah adalah 1,6 V. Sehingga besar nilai resistor yang akan digunakan adalah (Zheludev, 2007:189) :

$$R = \frac{V_{cc} - V_{LED}}{I_{LED}maks} \quad (4-11)$$

$$R = \frac{5V - 1,6V}{20mA}$$

$$R = 170 \Omega$$

Nilai resistor minimum adalah  $170 \Omega$ , dan dalam perancangan menggunakan resistor dengan nilai sebesar  $470 \Omega$ . Hal ini dimaksudkan agar arus yang mengalir ke LED tidak terlalu mendekati arus maksimumnya.

Untuk photodiode besarnya arus yang melewati merupakan arus yang digunakan untuk mikrokontroler sebesar  $0,01\text{mA}$  dan tegangan yang mendekati  $5\text{V}$  untuk logika tinggi. Sehingga dengan  $V_{cc}$  sebesar  $5\text{V}$ , maka besar  $R$  adalah :

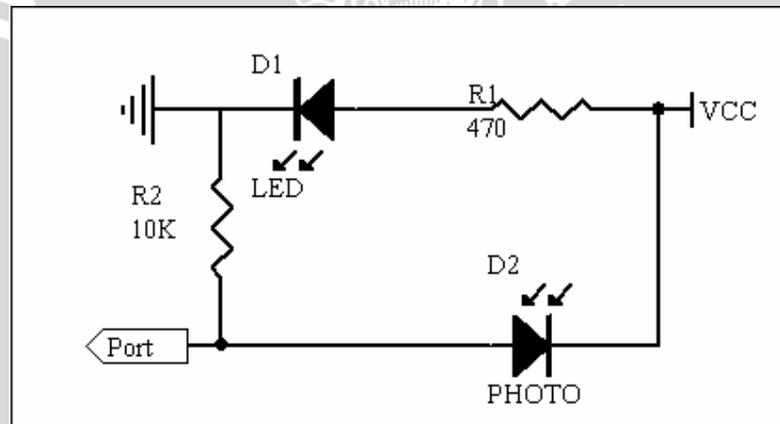
$$V_{cc} = R \cdot I_{MK} + V_{out} \quad (4-12)$$

$$5V = R \cdot 0,01\text{mA} + 4,9V$$

$$R = \frac{5V - 4,9V}{0,01\text{mA}}$$

$$R = 10000 \Omega$$

Gambar rancangan sensor obyek gambar diberikan dalam Gambar 4.9.



Gambar 4.9 Rancangan Sensor Obyek Gambar.  
(Sumber : Perancangan)

#### 4.2.8 Perancangan Rangkaian *Limit Switch*

Perancangan rangkaian *limit switch* menggunakan saklar tekan dan ketika ditekan akan mengirimkan logika tinggi ke mikrokontroler. *Limit switch* berfungsi sebagai pembatas gerakan motor DC sehingga bisa mengirimkan sinyal untuk membalik arah putaran atau menghentikan putaran motor DC.

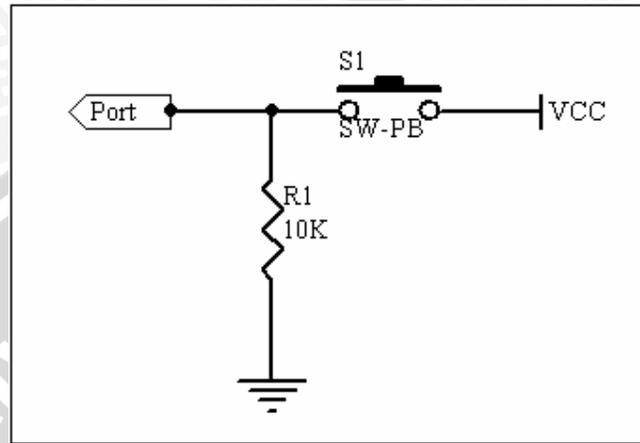
Besarnya arus yang digunakan untuk mikrokontroler sebesar  $0,01\text{mA}$  dan tegangan yang mendekati  $5\text{V}$  untuk logika tinggi. Sehingga dengan  $V_{cc}$  sebesar  $5\text{V}$ , maka besar  $R$  bisa dihitung menggunakan Rumus 4-12.

$$5V = R \cdot 0,01\text{mA} + 4,9V$$

$$R = \frac{5V - 4,9V}{0,01mA}$$

$$R = 10000 \Omega$$

Gambar rancangan *limit switch* diberikan dalam Gambar 4.10.



Gambar 4.10 Rancangan Rangkaian *Limit Switch*.  
(Sumber : Perancangan)

#### 4.2.9 Perancangan Kontrol Logika *Fuzzy*

Perancangan kontrol logika *fuzzy* dilakukan dengan mengacu pada langkah-langkah desain kontrol logika *fuzzy* (Ross, 1997:475-476) dan prinsip-prinsip dalam desain logika *fuzzy* (Ross, 1997:477).

Dalam perancangan variabel masukan yang pertama adalah besarnya tegangan yang menjadi set point dikurangi besarnya tegangan hasil pengukuran dari *tachogenerator* DC. Hasil pengurangan tersebut merupakan error ( $e$ ) (Ross, 1997:478).

$$\text{error} = \text{tegangan set point} - \text{tegangan tachogenerator DC} \quad (4-13)$$

Variabel masukan yang kedua adalah perubahan error ( $\dot{e}$ ), perubahan error diperoleh dari perbedaan error yang sekarang dengan error yang sebelumnya dibagi rentang waktu antara error yang sekarang dengan error waktu yang sebelumnya (Ross, 1997:478)

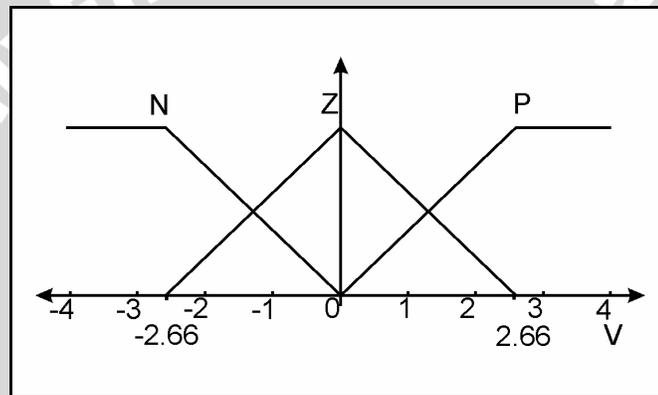
$$\dot{e} = \frac{e(t) - e(t - \Delta t)}{\Delta t} \quad (4-14)$$

Variabel keluaran adalah besarnya sinyal PWM yang diberikan pada *driver* motor. Sinyal PWM dibangkitkan melalui perbandingan bilangan biner 4

bit dari mikrokontroler AT89S52 melalui penahan 74LS573 dengan penghitung 4 bit 74LS161. Sehingga sinyal kontrol yang diberikan merupakan bilangan biner 4 bit, atau angka 1 sampai 16 dalam bilangan desimal.

#### 4.2.9.1 Fuzzifikasi

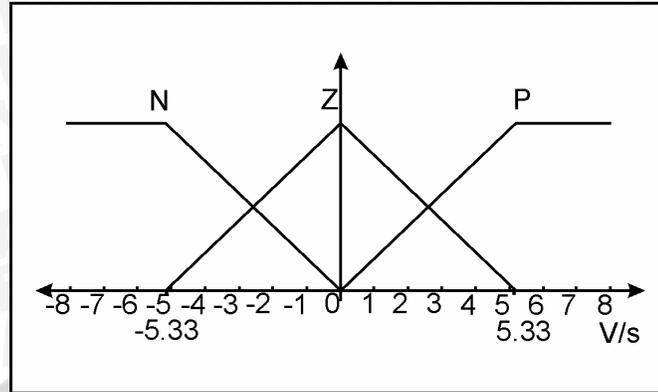
*Fuzzifikasi* masukan error menggunakan rentang jarak antara -4 V sampai 4 V. Dari rentang jarak yang ada dibagi ke dalam 3 bagian yang mewakili error negatif (N), error zero (Z), dan error positif (P). Fungsi keanggotaan dari 3 bagian yang ada menggunakan bentuk segitiga. 3 bagian ini masing-masing fungsi keanggotaannya diberikan dalam Gambar 4.11.



**Gambar 4.11 Fungsi Keanggotaan Masukan Error.**

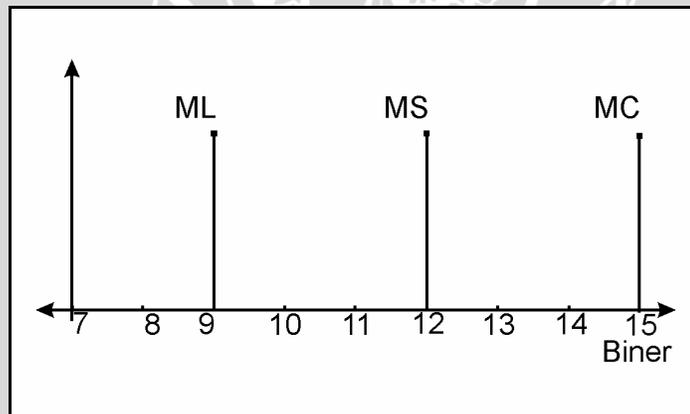
(Sumber: Perancangan)

*Fuzzifikasi* masukan perubahan error menggunakan rentang jarak antara -8 V/s sampai 8 V/s. Rentang jarak ini kemudian dibagi ke dalam 3 bagian yang mewakili perubahan error negatif (N), perubahan error zero (Z), dan perubahan error positif (P). Fungsi keanggotaan dari 3 bagian yang ada menggunakan bentuk segitiga. 3 bagian ini masing-masing fungsi keanggotaannya diberikan dalam Gambar 4.12.



Gambar 4.12 Fungsi Keanggotaan Masukan Perubahan Error.  
(Sumber: Perancangan)

*Fuzzifikasi* keluaran menggunakan rentang jarak antara 8 sampai 16 dalam bentuk bilangan biner. Rentang jarak ini dibagi kedalam 3 bagian yang mewakili kecepatan motor lambat (ML), kecepatan motor sedang (MS), dan kecepatan motor tinggi (MT). Fungsi keanggotaan dari 3 bagian yang ada menggunakan bentuk singleton, sehingga nilai bagian yang ada terletak pada bilangan tertentu. Nilai fungsi keanggotaan yang diwakili oleh bilangan biner tertentu dan dianggap nilai fungsi keanggotaan tersebut tidak bermassa. Gambar 3 bagian fungsi keanggotaan keluaran diberikan dalam Gambar 4.13.



Gambar 4.13 Fungsi Keanggotaan Keluaran.  
(Sumber: Perancangan)

#### 4.2.9.2 Basis Aturan

Setelah melakukan *fuzzifikasi* untuk variabel-variabel masukan dan keluaran, langkah berikut adalah perancangan basis aturan yang menghubungkan variabel-variabel masukan sebagai sebab dan variabel keluaran sebagai akibat.

Tabel 4.2 menjelaskan tentang basis aturan yang dirancang untuk digunakan dalam pengendalian kecepatan motor DC.

**Tabel 4.2 Basis Aturan Pengendalian Kecepatan Motor DC.**

$\Delta e$ \ e	N	Z	P
N	ML	ML	MS
Z	ML	MS	MC
P	MS	MC	MC

Sumber: Perancangan

Penjelasan dari basis aturan yang ada adalah sebagai berikut :

IF  $e = N$  AND  $e = N$  THEN Keluaran = ML

IF  $e = N$  AND  $e = Z$  THEN Keluaran = ML

IF  $e = N$  AND  $e = P$  THEN Keluaran = MS

IF  $e = Z$  AND  $e = N$  THEN Keluaran = ML

IF  $e = Z$  AND  $e = Z$  THEN Keluaran = MS

IF  $e = Z$  AND  $e = P$  THEN Keluaran = MC

IF  $e = P$  AND  $e = N$  THEN Keluaran = MS

IF  $e = P$  AND  $e = Z$  THEN Keluaran = MC

IF  $e = P$  AND  $e = P$  THEN Keluaran = MC

#### 4.2.9.3 Defuzzifikasi

Setelah proses *fuzzifikasi* dan *rule evaluation* selesai dirancang, langkah selanjutnya adalah proses *defuzzifikasi*, yaitu proses untuk mengubah *output fuzzy*

menjadi *crisp output*. Hasil *defuzzifikasi* inilah yang akan digunakan untuk mengatur besarnya tegangan *output* pada aktuator. Metode yang digunakan pada proses *defuzzifikasi* ini adalah menggunakan metode COG (*Center Of Gravity*), sesuai dengan rumus (Jamshidi, 1993:106).

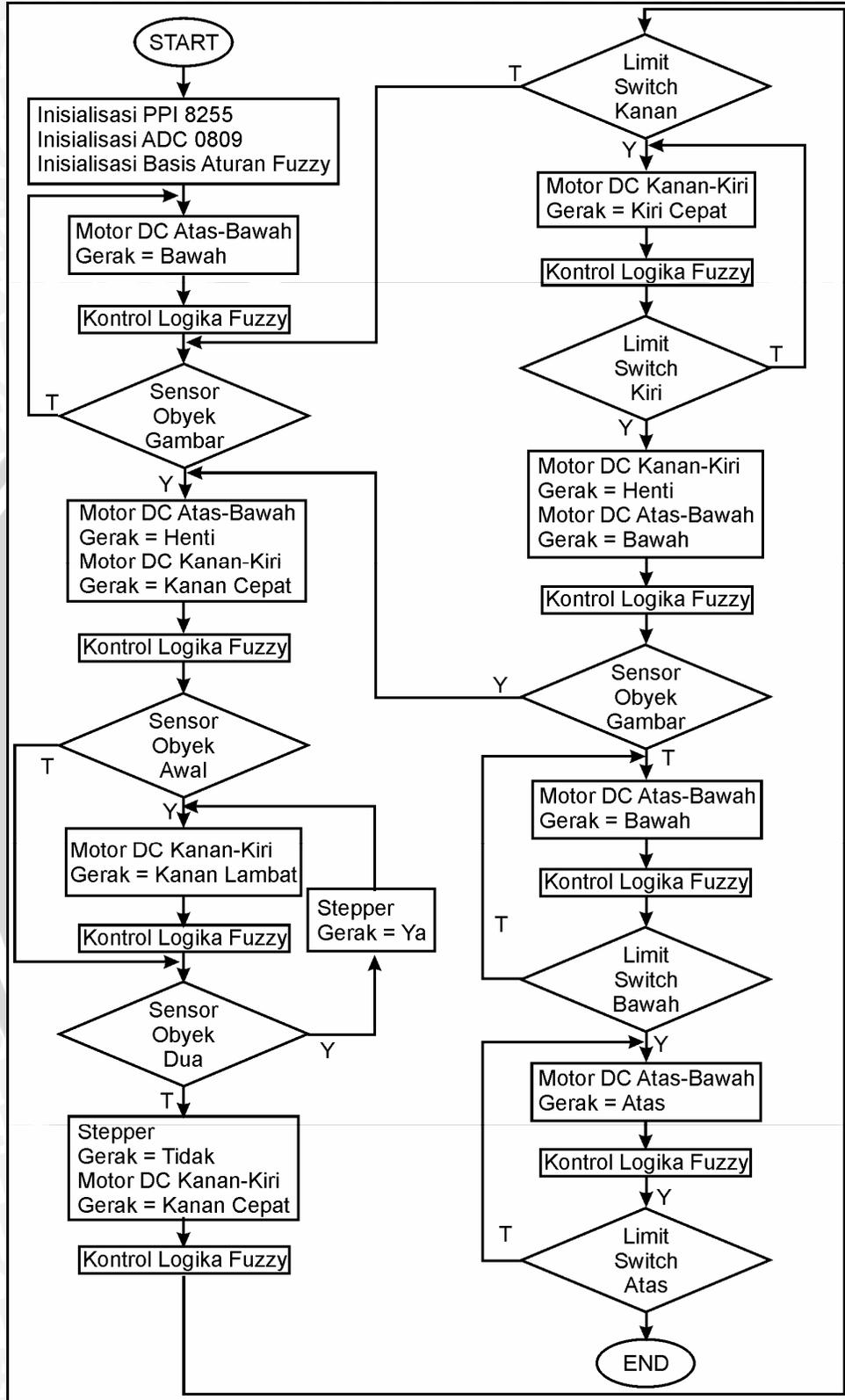
$$\text{Crisp Output } (y) = \frac{\sum_i \{(fuzzy\ output_i) \times (Singleton\ position\ on\ x\ axis)\}}{\sum_i (fuzzy\ output_i)} \quad (4-15)$$

#### 4.2.10 Perancangan Perangkat Lunak

Perangkat lunak dirancang untuk membuat sistem berjalan dengan baik sesuai dengan prinsip kerja alat dan memiliki kemampuan untuk mengendalikan kecepatan motor DC yang berfungsi sebagai penggerak. Perancangan perangkat lunak meliputi perancangan perangkat lunak sistem dan perancangan perangkat lunak kontrol logika *fuzzy*.

##### 4.2.10.1 Diagram Alir Perangkat Lunak Sistem

Diagram alir perangkat lunak sistem menjelaskan aliran program yang akan dirancang. Diagram alir setelah dimulai akan melakukan inisialisasi perangkat keras mulai dari antar muka PPI 8255 dan ADC 0809. Kemudian menggerakkan motor DC atas-bawah ke arah bawah. Ketika mendapat masukan keberadaan obyek gambar dari sensor saklar cahaya atas-bawah, motor DC atas-bawah berhenti dan motor DC kanan-kiri bergerak ke arah kanan. Masuknya sinyal obyek gambar dari sensor cahaya kanan-kiri awal membuat kecepatan motor DC kanan kiri. Sensor cahaya kanan kiri dua mengaktifkan motor *stepper* penulis tinta. Motor DC kanan-kiri akan berputar berbalik arah bila *limit switch* di kanan tersentuh, dan berhenti bila *limit switch* di kiri tersentuh. Ketika *limit switch* di kiri tersentuh motor DC atas-bawah akan kembali bergerak ke bawah. Bila tidak ada sinyal dari sensor cahaya atas-bawah, maka motor DC akan terus bergerak sampai menyentuh *limit switch* bawah dan arah putaran membalik. Proses selesai ketika *limit switch* di atas tersentuh. Semua proses pengendalian kecepatan motor DC penggerak dikendalikan melalui kontrol logika *fuzzy*. Untuk lebih jelasnya diagram alir perangkat lunak diberikan dalam Gambar 4.14.



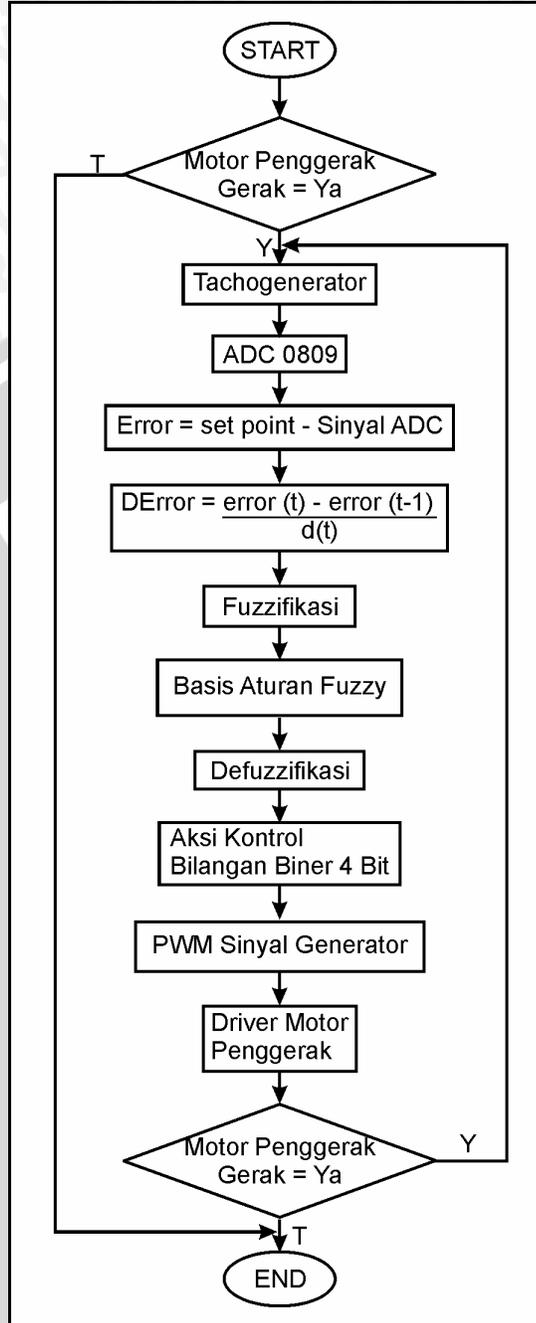
Gambar 4.14 Diagram Alir Perangkat Lunak Sistem  
(Sumber: Perancangan.)

#### 4.2.10.2 Diagram Alir Perangkat Lunak Kontrol Logika *Fuzzy*

Proses pengendalian kecepatan motor DC dilakukan melalui perangkat lunak dengan Kontrol Logika *Fuzzy* yang merupakan sistem pengendalian berbasis aturan. Perangkat lunak dimulai dengan melakukan *fuzzifikasi* variabel-variabel masukan dan keluaran yang ada. Variabel masukan berupa perhitungan error yang didapat dari selisih set point dengan sinyal pengukuran. Perubahan error diperoleh dari selisih error yang sekarang dikurangi error sebelumnya dibagi rentang waktu. Variabel keluaran merupakan bilangan biner 4 bit, yang merupakan masukan untuk sinyal *generator PWM*.

Basis aturan digunakan untuk menentukan keputusan aksi kontrol melalui informasi dari variabel-variabel masukan yang ada. Setelah diperoleh aksi kontrol dari basis aturan, dilakukan proses *defuzzifikasi* untuk mendapatkan aksi kontrol dalam bentuk himpunan tegas. Diagram alir perangkat lunak kontrol logika *fuzzy* diberikan dalam Gambar 4.15.





Gambar 4.15 Diagram Alir Perangkat Lunak Kontrol Logika Fuzzy.  
(Sumber: Perancangan)

## BAB V

### PENGUJIAN ALAT DAN ANALISIS

Tujuan dilakukannya pengujian alat adalah untuk mengetahui apakah alat yang telah dirancang dan dibuat telah bekerja dengan baik dan sesuai dengan perancangan. Pengujian ini dilakukan dengan pengujian per blok dan pengujian alat keseluruhan. Pengujian per blok dimaksudkan untuk mempermudah analisis dan perbaikan apabila alat yang telah dirancang dan dibuat kurang sesuai dengan perancangan.

Pengujian yang dilakukan meliputi:

1. Pengujian Sensor Kecepatan Motor DC.
2. Pengujian Mikrokontroler AT89S52.
3. Pengujian PPI 8255.
4. Pengujian Rangkaian PWM dan *Driver* Motor DC.
5. Pengujian Rangkaian ADC 0809
6. Pengujian Rangkaian *Driver* Motor *Stepper*
7. Pengujian Sensor Obyek Gambar.
8. Pengujian *Limit Switch*.
9. Pengujian Sistem Secara Keseluruhan.

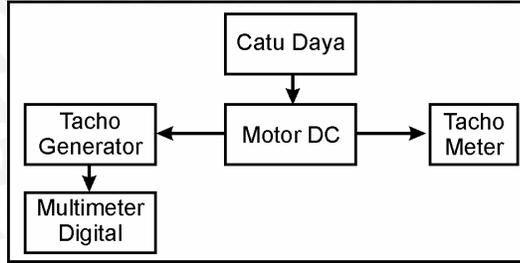
#### 5.1 Pengujian Sensor Kecepatan Motor

Pengujian sensor kecepatan motor dimaksudkan untuk mengetahui besar nilai tegangan keluaran dari *tachogenerator* yang mewakili besarnya nilai kecepatan putaran dari motor DC.

a. Peralatan yang digunakan :

1. Catu Daya
2. Motor DC
3. *Tachogenerator* DC
4. Multimeter Digital
5. *Tachometer*

b. Langkah-langkah Pengujian :



**Gambar 5.1 Blok Pengujian Sensor Kecepatan Motor DC.**  
(Sumber : Pengujian)

1. Merangkai blok pengujian kecepatan motor DC seperti dalam Gambar 5.1.
2. Mengaktifkan catu daya
3. Mengamati dan mencatat setiap perubahan level tegangan yang dihasilkan *tachogenerator* yang mewakili kecepatan motor dari pembacaan *tachometer* untuk setiap perubahan catu daya.

c. Hasil Pengujian dan Analisis Hasil Pengujian

**Tabel 5.1 Hasil Pengujian Sensor Kecepatan Motor DC.**

RPM Tachometer	Vout Tachogenerator			Rata-rata	Δ Vout Rata-rata
	1	2	3		
0	0	0	0	0	-
100	0,69	0,69	0,67	0,68	-
200	0,87	0,85	0,88	0,87	0,19
300	1,11	1,1	1,11	1,1	0,23
400	1,32	1,34	1,35	1,34	0,24
500	1,52	1,52	1,54	1,52	0,18
600	1,83	1,84	1,84	1,84	0,32
700	2,1	2,09	2,11	2,1	0,26
800	2,25	2,26	2,24	2,25	0,15
900	2,49	2,53	2,51	2,51	0,26
1000	2,7	2,72	2,72	2,71	0,2
1100	3,01	3,01	2,95	2,99	0,28
1200	3,17	3,14	3,17	3,16	0,17
1300	3,39	3,41	3,38	3,39	0,23
1400	3,65	3,64	3,66	3,65	0,26
1500	3,82	3,83	3,8	3,81	0,16
1600	4,12	4,1	4,09	4,1	0,29
1700	4,33	4,31	4,31	4,32	0,22
1800	4,51	4,52	4,53	4,52	0,2
1900	4,78	4,76	4,79	4,78	0,26
2000	5,01	5	4,99	5	0,22
				<b>Rata-rata</b>	<b>0,21</b>

Sumber : Pengujian

Dari Tabel 5.1 hasil pengujian untuk setiap perubahan RPM sebesar 100 RPM maka Vout tachogenerator mengalami perubahan rata-rata sebesar 0,21 V. Dengan perubahan Vout yang relatif linier terhadap perubahan RPM, maka Vout dari tachogenerator bisa dianggap mewakili besar RPM motor, sehingga bisa digunakan sebagai sensor kecepatan motor DC.

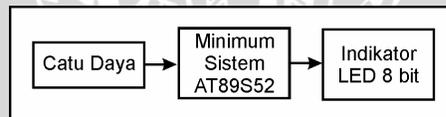
### 5.2 Pengujian Mikrokontroler

Tujuan pengujian mikrokontroler adalah untuk mengetahui kondisi awal dari sistem mikrokontroler agar sesuai dengan yang diharapkan.

a. Peralatan yang digunakan :

1. Minimum sistem AT89S52.
2. Catu daya.
3. Indikator LED 8 bit.

b. Langkah-langkah Pengujian :



Gambar 5.2 Blok Pengujian Mikrokontroler AT89S52  
(Sumber : Pengujian)

1. Merangkai blok pengujian mikrokontroler seperti dalam Gambar 5.2.
2. Mengisi mikrokontroler dengan program sederhana yakni meletakkan data 0FH dan F0H pada *accumulator* secara bergantian kemudian mengeluarkan data tersebut melalui port 1 yang telah dihubungkan dengan indikator LED 8 bit.
3. Mengaktifkan catu daya..
4. Mereset program untuk mengeluarkan data mikrokontroler.
5. Mencatat data keluaran dalam bentuk biner yang diwakili oleh aktifnya indikator LED 8 bit.

c. Hasil Pengujian dan Analisis Hasil Pengujian

Tabel 5.2 Hasil Pengujian Mikorkontroler AT89S52

Kondisi	Keluaran Indikator LED 8 bit							
	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

Sumber : Pengujian

Dalam Tabel 5.2 diasumsikan bahwa indikator LED aktif berlogika “1” dan tidak aktif berlogika “0”, maka dalam Tabel 5.2 dapat ditarik kesimpulan bahwa port 1 mikrokontroler menghasilkan logika 0Fh dan F0h secara bergantian sesuai dengan program.

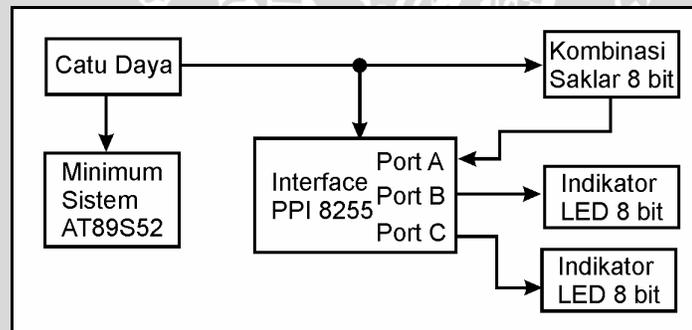
### 5.3 Pengujian PPI 8255

Tujuan pengujian PPI 8255 dimaksudkan untuk mengetahui apakah PPI 8255 sudah berfungsi dengan semestinya, yaitu *Port A*, *Port B* dan *Port C* sebagai masukan atau keluaran.

a. Peralatan yang digunakan :

1. Catu Daya
2. Minimum Sistem AT89S52
3. Rangkaian *Interface* PPI 8255
4. Indikator LED 8 bit
5. Kombinasi saklar 8 bit

b. Langkah-langkah Pengujian :



**Gambar 5.3 Blok Pengujian PPI 8255**  
(Sumber : Pengujian)

1. Merangkai blok pengujian PPI 8255 seperti dalam Gambar 5.3.
2. Mengisi program mikrokontroler dengan program yang berfungsi untuk mengaktifkan *port* PPI sebagai masukan dan keluaran. Alamat PPI 8255 pada 2000H. Mengisikan *Control Word* pada PPI 8255 agar beroperasi dalam mode 0 yang berfungsi untuk mengaktifkan *port A*, *port B*, dan *port C* sebagai saluran I/O biasa. *Port A* PPI 8255 sebagai masukan dan *port B* dan *port C* sebagai keluaran, *control word* yang diberikan pada PPI 8255 adalah 90H. Serta program yang mengirimkan keluaran logika “0” di port

B dan keluaran logika “1” di port C ketika port A sebagai masukan mendapatkan logika “0”, dan sebaliknya bila masukan mendapat logika “1”.

3. Mengaktifkan catu daya.
4. Mengaktifkan kombinasi saklar 8 bit sebagai data masukan di port A.
5. Mengamati dan mencatat perubahan data keluaran di port B dan port C melalui indikator LED 8 bit.

#### c. Hasil Pengujian dan Analisis Hasil Pengujian

**Tabel 5.3 Hasil Pengujian PPI 8255**

Kombinasi Saklar 8 bit	Indikator LED 8 bit	Kondisi Logika							
		Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
Tidak Ditekan		0	0	0	0	0	0	0	0
	Indikator LED 1	0	0	0	0	0	0	0	0
	Indikator LED 2	1	1	1	1	1	1	1	1
Ditekan		1	1	1	1	1	1	1	1
	Indikator LED 1	1	1	1	1	1	1	1	1
	Indikator LED 2	0	0	0	0	0	0	0	0

Sumber : Pengujian

Dalam Tabel 5.3 diasumsikan bahwa indikator LED aktif berlogika “1” dan tidak aktif berlogika “0”, dan saklar tidak ditekan (tidak aktif) berlogika “1” dan saklar ditekan (aktif) berlogika “0”. Dari Tabel, dapat ditarik kesimpulan bahwa PPI 8255 bisa berfungsi dengan baik sebagai *port I/O*.

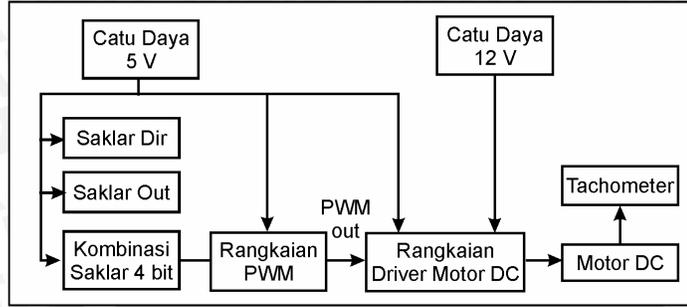
#### 5.4 Pengujian Rangkaian PWM dan Driver Motor DC

Pengujian rangkaian PWM dan *driver* motor DC dimaksudkan untuk mengetahui kinerja PWM dan *driver* motor DC, serta mengetahui kecepatan motor DC untuk berbagai perubahan sinyal masukan dari PWM.

##### a. Peralatan yang digunakan :

1. Catu Daya 5 V dan 12 V
2. Rangkaian PWM
3. Rangkaian *Driver* Motor DC
4. Kombinasi saklar 6 bit
5. *Tachometer*

b. Langkah-langkah Pengujian :



Gambar 5.4 Pengujian Rangkaian PWM dan Driver Motor DC.

(Sumber : Pengujian)

1. Merangkai blok pengujian seperti dalam Gambar 5.4.
2. Mengubah sinyal masukan untuk arah gerak dan pembangkit sinyal di rangkaian PWM.
3. Mengamati dan mencatat perubahan arah gerak dan kecepatan motor DC terhadap setiap perubahan sinyal masukan.

c. Hasil Pengujian dan Analisis Hasil Pengujian

Tabel 5.4 Hasil Pengujian Rangkaian PWM dan Driver Motor DC.

Saklar Kombinasi 6 Bit		Kecepatan Motor DC (RPM)			Rata-rata	d Rata-rata	Arah Gerak Motor				
		1	2	3							
Dir	0										
Out	0										
Kombinasi Saklar 4 bit	Bit 4	Bit 3	Bit 2	Bit 1							
	1	1	1	1	0	0	0	0	-	-	
	1	1	1	0	0	0	0	0	0	-	
	1	1	0	1	0	0	0	0	0	-	
	1	1	0	0	0	0	0	0	0	-	
	1	0	1	1	0	0	0	0	0	0	-
	1	0	1	0	0	0	0	0	0	0	-
	1	0	0	1	0	0	0	0	0	0	-
Dir	0										
Out	1										
Kombinasi Saklar 4 bit	Bit 4	Bit 3	Bit 2	Bit 1							
	1	1	1	1	1424	1423	1425	1424	-	Kanan	
	1	1	1	0	1319	1321	1313	1317	-83	Kanan	
	1	1	0	1	1267	1269	1264	1266	-73	Kanan	
	1	1	0	0	1182	1184	1186	1184	-82	Kanan	
	1	0	1	1	1091	1087	1092	1090	-94	Kanan	
	1	0	1	0	985	986	988	986	-85	Kanan	
	1	0	0	1	854	853	851	852	-133	Kanan	
1	0	0	0	685	686	686	685	-166	Kanan		
Rata-rata								-105			

Dir	1									
Out	1									
Kombinasi Saklar 4 bit	Bit 4	Bit 3	Bit 2	Bit 1						
1	1	1	1	1	1426	1425	1425	1425	-	Kiri
1	1	1	0	1	1340	1339	1339	1339	-86	Kiri
1	1	0	1	1	1278	1279	1278	1278	-61	Kiri
1	1	0	0	1	1184	1186	1186	1185	-93	Kiri
1	0	1	1	1	1071	1070	1069	1070	-115	Kiri
1	0	1	0	1	986	987	986	986	-83	Kiri
1	0	0	1	1	853	852	852	852	-134	Kiri
1	0	0	0	1	690	695	697	694	-158	Kiri
Rata-rata									-104	

Sumber : Pengujian

Dari Tabel 5.4 dapat disimpulkan bahwa ketika logika “0” diberikan pada saklar out, maka motor DC tidak akan berputar. Motor DC akan berputar bila logika “1” diberikan pada saklar out. Arah putaran motor DC sangat bergantung pada logika yang diberikan pada saklar DIR, logika “1” akan membuat motor DC berputar ke kiri sedangkan logika “0” akan membuat motor DC berputar ke kanan. Besarnya kecepatan putaran motor DC dipengaruhi oleh logika yang diberikan ke kombinasi saklar 4 bit. Besarnya penurunan kecepatan motor DC rata-rata untuk pengurangan tiap satu bit adalah 105 RPM.

Dengan kemampuan rangkaian PWM dan driver motor DC untuk mengatur arah putaran dan kecepatan sesuai dengan logika yang diberikan, maka rangkaian PWM dan driver motor DC telah berfungsi dengan baik.

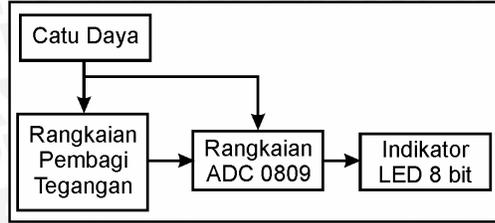
### 5.5 Pengujian ADC 0809

Tujuan pengujian rangkaian ADC 0809 ini adalah untuk mengetahui kepresisian ADC dalam mengkonversi besaran tegangan analog yang diberikan ke dalam nilai keluaran digital 8 bit.

a. Peralatan yang digunakan :

1. Rangkaian ADC 0809
2. Catu Daya
3. Rangkaian Pembagi Tegangan
4. Multimeter Digital
5. Indikator LED 8 bit

b. Langkah-langkah Pengujian :



Gambar 5.5 Blok Pengujian Rangkaian ADC 0809  
(Sumber : Pengujian)

1. Merangkai blok pengujian seperti dalam Gambar 5.5.
2. Menghubungkan keluaran rangkaian ADC 0809 dengan indikator LED 8 bit untuk mengetahui hasil konversi masukan tegangan analog ke dalam keluaran digital.
3. Mengatur perubahan tegangan masukan analog pada rangkaian pembagi tegangan dengan cara memutar resistor variabel.
4. Mengamati dan mencatat perubahan keluaran digital untuk setiap perubahan masukan tegangan analog.

c. Hasil Pengujian dan Analisis Hasil Pengujian

Tabel 5.5 Hasil Pengujian ADC 0809

Tegangan Masukan (mV)	Indikator LED 8 bit								Desimal	Perubahan Tiap bit (mV)
	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Bit 8		
0	0	0	0	0	0	0	0	0	0	0
500	0	0	0	1	1	0	1	0	26	19,23
1000	0	0	1	1	0	0	1	1	51	19,61
1500	0	1	0	0	1	1	1	1	79	18,99
2000	0	1	1	0	0	1	1	0	102	19,61
2500	1	0	0	0	0	0	1	1	131	19,08
3000	1	0	0	1	1	1	0	0	156	19,23
3500	1	0	1	1	0	1	0	0	180	19,44
4000	1	1	0	0	1	1	0	0	204	19,61
4500	1	1	1	0	0	1	0	1	229	19,65
5000	1	1	1	1	1	1	1	1	255	19,61
Rata-rata									19,41	

Sumber : Pengujian

Dari Tabel 5.5 bisa dilihat bahwa ADC 0809 dapat mengkonversi tegangan masukan analog 0 sampai 5 volt menjadi keluaran biner. Nilai kesalahan maksimum berdasarkan data yang diberikan ke ADC 0809 adalah sebesar 1 LSB.

Resolusi 1 LSB untuk sistem ini adalah :

$$\begin{aligned}\text{Resolusi 1 LSB} &= \frac{V_{ref}}{2^8 - 0} \\ &= \frac{5}{256} \\ &= 19,5 \text{ mV}\end{aligned}$$

Besarnya persentase kesalahan untuk perubahan tiap bit dapat diketahui dengan perhitungan sebagai berikut :

$$\begin{aligned}\text{Persentase kesalahan perubahan tiap bit} &= \frac{19,4 - 19,5}{19,5} \times 100\% \\ &= 0,51 \%\end{aligned}$$

Adanya kesalahan hasil konversi ADC 0809 disebabkan oleh tingkat akurasi komponen ADC 0809 serta kurang stabilnya tegangan referensi yang diberikan pada ADC.

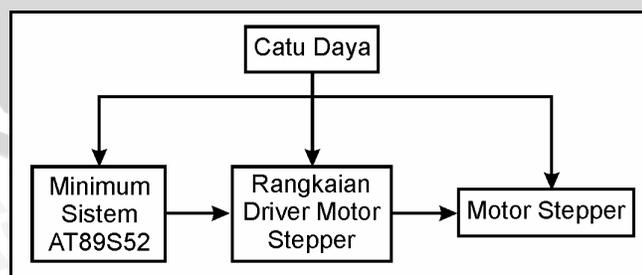
### 5.6 Pengujian Rangkaian *Driver Motor Stepper*.

Pengujian rangkaian *driver motor stepper* dimaksudkan untuk mengetahui kinerja *driver motor stepper* dalam menggerakkan motor *stepper* dengan menggunakan keluaran dari mikrokontroler AT89S52.

a. Peralatan yang digunakan :

1. Catu Daya 5V dan 12 V
2. Minimum sistem AT89S52
3. Rangkaian *driver motor stepper*
4. Motor *stepper*

b. Langkah-langkah Pengujian :



Gambar 5.6 Blok Pengujian Driver Motor Stepper.

(Sumber : Pengujian)

1. Merangkai blok pengujian seperti dalam Gambar 5.6.

2. Memasukkan program ke dalam mikrokontroler AT89S52 yang berfungsi mengaktifkan kumparan motor *stepper* secara bergantian dengan arah gerak ke kanan dan ke kiri dan *delay* sebesar  $\pm 5$ ms.
3. Menghidupkan rangkaian blok pengujian
4. Mengamati dan mencatat gerakan motor *stepper*.

c. Hasil Pengujian dan Analisis Hasil Pengujian

**Tabel 5.6 Hasil Pengujian Driver Motor Stepper.**

Kondisi	Arah Putaran Motor Stepper
I	Kanan
II	Kiri

Sumber : Pengujian

Dari Tabel 5.6 bisa disimpulkan bahwa *driver* motor *stepper* bekerja dengan baik karena telah sesuai dengan program yang diisikan.

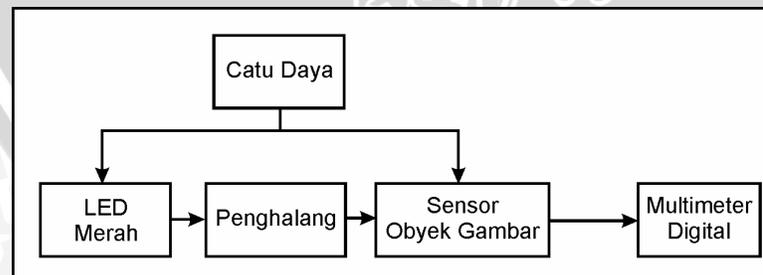
### 5.7 Pengujian Sensor Obyek Gambar

Pengujian sensor obyek gambar dimaksudkan untuk mengetahui besar nilai tegangan keluaran yang dihasilkan oleh sensor obyek gambar ketika cahaya LED merah terhalang oleh obyek gambar.

a. Peralatan yang digunakan :

1. Catu daya
2. Rangkaian sensor obyek gambar
3. Multimeter Digital
4. Penghalang

b. Langkah-langkah Pengujian :



**Gambar 5.7 Blok Pengujian Sensor Obyek Gambar.**

(Sumber : Pengujian)

1. Merangkai blok pengujian sensor obyek gambar seperti dalam Gambar 5.7.

2. Mengaktifkan catu daya
3. Memberikan penghalang pada LED merah.
4. Mengamati dan mencatat level tegangan yang keluar ketika LED merah terhalang dan ketika LED merah tidak terhalang.

c. Hasil Pengujian dan Analisis Hasil Pengujian

**Tabel 5.7 Hasil Pengujian Sensor Obyek Gambar**

Cahaya LED	Vout Sensor Obyek Gambar (Volt)			Rata-rata (Volt)	Logika
	1	2	3		
Tidak terhalang	4,56	4,57	4,56	4,56	1
Terhalang	0,14	0,13	0,13	0,13	0

Sumber : Pengujian

Dari Tabel 5.7 dapat ditarik kesimpulan bahwa ketika cahaya LED merah tidak terhalang, maka rangkaian sensor obyek gambar akan berlogika “1”. Sedangkan pada saat cahaya LED merah terhalang, maka rangkaian sensor obyek gambar akan berlogika “0”. Sehingga rangkaian sensor obyek gambar sensor keberadaan obyek gambar bekerja dengan baik.

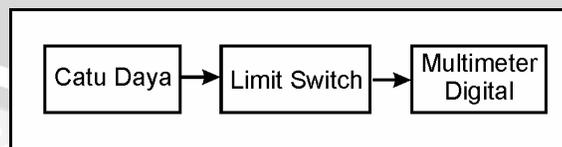
### 5.8 Pengujian *Limit Switch*.

Pengujian *limit switch* dimaksudkan untuk mengetahui tegangan keluaran yang dihasilkan oleh rangkaian *limit switch* ketika aktif dan tidak aktif.

a. Peralatan yang digunakan :

1. Catu daya
2. Rangkaian *limit switch*
3. Multimeter digital

b. Langkah-langkah Pengujian :



**Gambar 5.8 Blok Pengujian Rangkaian *Limit switch*.**

(Sumber : Pengujian)

1. Merangkai blok pengujian *limit switch* seperti dalam Gambar 5.8.
2. Mengaktifkan catu daya.

3. Menekan *limit switch*.
4. Mengamati dan mencatat level tegangan yang dihasilkan ketika *limit switch* tidak ditekan (tidak aktif) dan ketika *limit switch* ditekan (aktif).

c. Hasil Pengujian dan Analisis Hasil Pengujian

**Tabel 5.8 Hasil Pengujian Rangkaian *Limit switch*.**

Limit Switch	Vout Limit Switch (Volt)			Rata-rata (Volt)	Logika
	1	2	3		
Tidak ditekan	0	0	0	0	0
Ditekan	4,91	4,92	4,92	4,91	1

Sumber : Pengujian

Dari Tabel 5.8 dapat ditarik kesimpulan bahwa pada saat *limit switch* tidak ditekan (tidak aktif), maka rangkaian *limit switch* akan berlogika “0”. Sedangkan pada saat *limit switch* ditekan (aktif), maka rangkaian *limit switch* akan berlogika “1”. Sehingga rangkaian *limit switch* yang sebagai sensor batas bekerja dengan baik.

### 5.9 Pengujian Sistem Keseluruhan.

Pengujian sistem keseluruhan dimaksudkan untuk mengetahui kemampuan sistem dalam mengendalikan kecepatan motor DC.

a. Peralatan yang digunakan :

1. Catu Daya 5 V dan 12 V
2. Rangkaian Alat Pengendalian Kecepatan Motor DC
3. *Stopwatch*
4. Beban Berupa Karet *Konveyor*.
5. *Tachometer*

b. Langkah-langkah Pengujian :

1. Menjalankan Alat Pengendalian Kecepatan Motor DC
2. Memberikan beban ketika motor DC bergerak.
3. Mengamati dan mencatat perubahan kecepatan motor DC setiap 1 detik selama 20 detik.

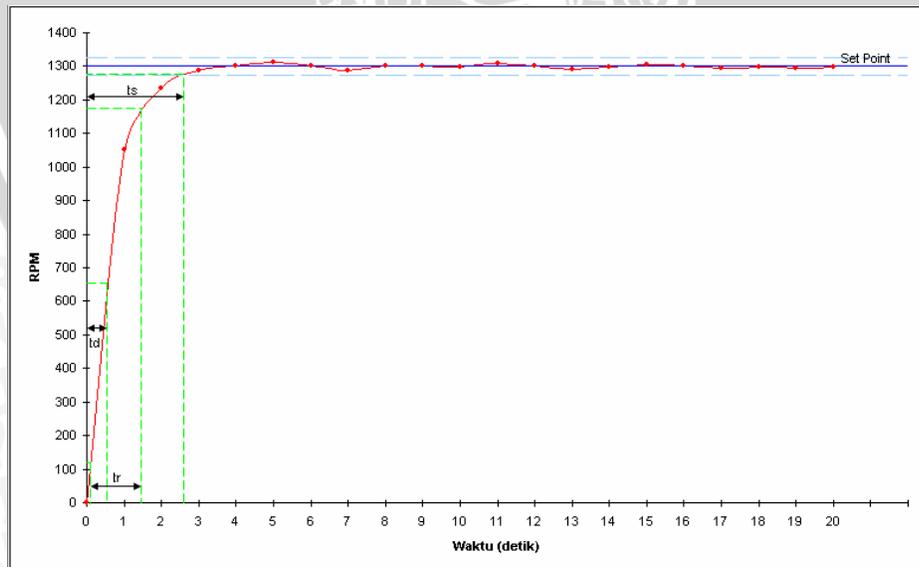
c. Hasil Pengujian dan Analisis Hasil Pengujian

**Tabel 5.9 Hasil Pengujian Kecepatan Motor DC Tanpa Beban Dengan Setpoint 1300 RPM.**

Waktu (detik)	Kecepatan Motor DC (RPM)			Rata-rata	error	d error
	1	2	3			
0	0	0	0	0	1300	-
1	1051	1048	1056	1051	249	-1051
2	1237	1240	1230	1235	65	-184
3	1286	1289	1287	1287	13	-52
4	1299	1300	1301	1300	0	-13
5	1313	1310	1314	1312	-12	-12
6	1303	1302	1304	1303	-3	9
7	1291	1287	1284	1287	13	16
8	1306	1301	1297	1301	-1	-14
9	1293	1313	1304	1303	-3	-2
10	1302	1300	1289	1297	3	6
11	1308	1315	1298	1307	-7	-10
12	1299	1304	1307	1303	-3	4
13	1286	1291	1299	1292	8	11
14	1301	1305	1288	1298	2	-6
15	1305	1311	1299	1305	-5	-7
16	1294	1302	1306	1300	0	5
17	1288	1286	1310	1294	6	6
18	1300	1297	1301	1299	1	-5
19	1286	1306	1288	1293	7	6
20	1299	1299	1294	1297	3	-4

Sumber : Pengujian

Dari Tabel 5.9 dibuat gambar grafik keluaran seperti dalam Gambar 5.9.



**Gambar 5.9 Grafik Keluaran Rata-rata Kecepatan Motor DC Tanpa Beban dengan Sepoint 1300 RPM.**  
(Sumber : Pengujian)

Dalam Gambar 5.9 *set point* berada pada kecepatan 1300 RPM. Toleransi error keadaan mantap ditentukan sebesar 2% dari *set point* atau sebesar 26 RPM. Nilai 10% dari *set point* adalah 130 RPM, untuk 50% adalah 650 RPM, dan 90% adalah 1170 RPM.

Berdasarkan Gambar 5.9 :

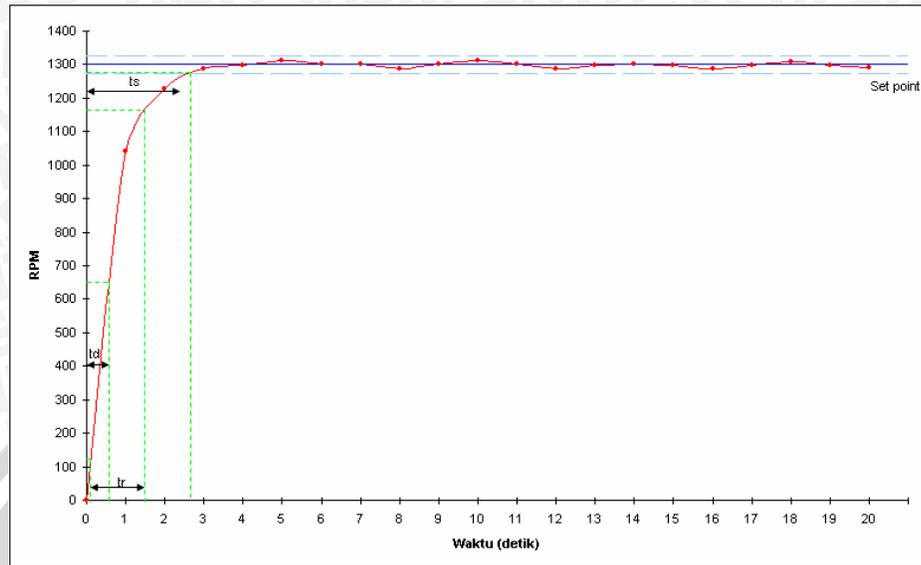
1. Waktu untuk pertama kali mencapai nilai 50% dari *set point* atau waktu tunda (td) adalah sebesar  $\pm 0,6$  detik.
2. Waktu untuk pertama kali mencapai nilai 10% dari *set point* adalah  $\pm 0,1$  detik dan waktu untuk pertama kali mencapai nilai 90% dari *set point* adalah  $\pm 1,4$  detik. Sehingga waktu naik (tr) dari nilai 10% sampai dengan 90% adalah  $\pm 1,3$  detik.
3. Waktu untuk pertama kali mencapai keadaan mantap atau waktu menetap (ts) dalam kisaran 2 % dari *set point* adalah  $\pm 2,6$  detik.
4. Nilai error keadaan mantap yang terbesar adalah 13 RPM atau 1 % dari set point.

**Tabel 5.10 Hasil Pengujian Kecepatan Motor DC Berbeban dengan Setpoint 1300 RPM.**

Waktu (detik)	Kecepatan Motor DC (RPM)			Rata-rata	error	d error
	1	2	3			
0	0	0	0	0	1300	-
1	1043	1041	1044	1042	258	-1042
2	1226	1229	1225	1226	74	-184
3	1289	1287	1285	1287	13	-61
4	1297	1301	1300	1299	1	-12
5	1314	1310	1312	1312	-12	-13
6	1299	1302	1301	1300	0	12
7	1304	1305	1302	1303	-3	-3
8	1286	1290	1288	1288	12	15
9	1302	1301	1300	1301	-1	-13
10	1310	1313	1315	1312	-12	-11
11	1299	1303	1301	1301	-1	11
12	1287	1286	1290	1287	13	14
13	1296	1299	1302	1299	1	-12
14	1310	1312	1289	1303	-3	-4
15	1299	1300	1299	1299	1	4
16	1287	1289	1292	1289	11	10
17	1299	1298	1301	1299	1	-10
18	1311	1308	1312	1310	-10	-11
19	1301	1296	1302	1299	1	11
20	1291	1286	1293	1290	10	9

Sumber : Pengujian

Dari Tabel 5.10 dibuat gambar grafik keluaran seperti dalam Gambar 5.10.



**Gambar 5.10 Grafik Keluaran Rata-rata Kecepatan Motor DC Berbeban dengan Setpoint 1300 RPM.**  
(Sumber : Pengujian)

Dalam Gambar 5.10 *set point* berada pada kecepatan 1300 RPM. Toleransi error keadaan mantap ditentukan sebesar 2% dari *set point* atau sebesar 26 RPM. Nilai 10% dari *set point* adalah 130 RPM, untuk 50% adalah 650 RPM, dan 90% adalah 1170 RPM.

Berdasarkan Gambar 5.10 :

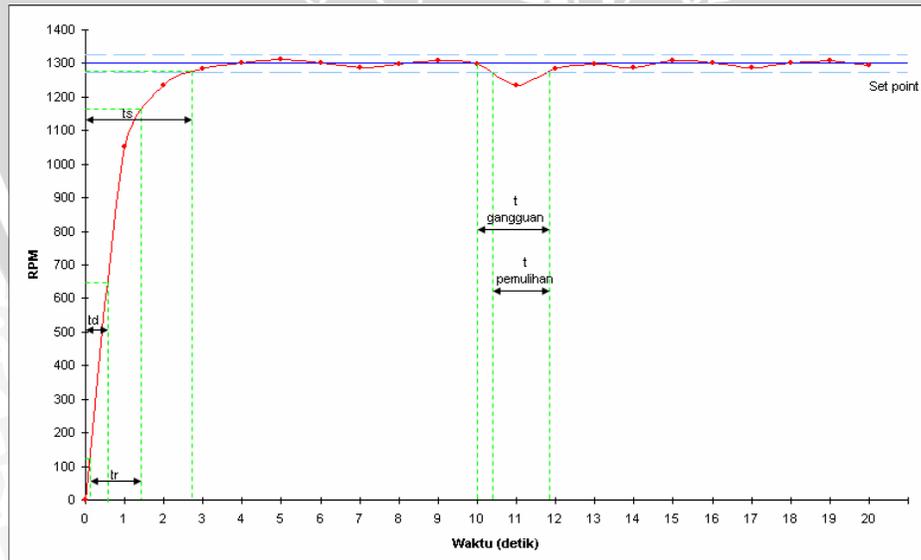
1. Waktu untuk pertama kali mencapai nilai 50% dari *set point* atau waktu tunda ( $t_d$ ) adalah sebesar  $\pm 0,7$  detik.
2. Waktu untuk pertama kali mencapai nilai 10% dari *set point* adalah  $\pm 0,1$  detik dan waktu untuk pertama kali mencapai nilai 90% dari *set point* adalah  $\pm 1,5$  detik. Sehingga waktu naik ( $t_r$ ) dari nilai 10% sampai dengan 90% adalah  $\pm 1,4$  detik.
3. Waktu untuk pertama kali mencapai keadaan mantap atau waktu menetap ( $t_s$ ) dalam kisaran 2 % dari *set point* adalah  $\pm 2,7$  detik.
4. Nilai error keadaan mantap yang terbesar adalah 13 RPM atau 1 % dari *set point*.

**Tabel 5.11 Hasil Pengujian Kecepatan Motor DC dengan Setpoint 1300 RPM Diberi Beban Pada Saat  $t = 10$  detik.**

Waktu (detik)	Kecepatan Motor DC (RPM)			Rata-rata	error	d error
	1	2	3			
0	0	0	0	0	1300	-
1	1052	1047	1052	1050	250	-1050
2	1238	1242	1235	1238	62	-188
3	1287	1286	1285	1286	14	-48
4	1301	1302	1301	1301	-1	-15
5	1312	1311	1313	1312	-12	-11
6	1301	1300	1303	1301	-1	11
7	1289	1288	1287	1288	12	13
8	1301	1301	1297	1299	1	-11
9	1309	1313	1307	1309	-9	-10
10	1301	1305	1299	1301	-1	8
11	1237	1235	1236	1236	64	65
12	1281	1283	1285	1283	17	-47
13	1299	1301	1299	1299	1	-16
14	1286	1291	1289	1288	12	11
15	1312	1311	1310	1311	-11	-23
16	1301	1302	1300	1301	-1	10
17	1288	1286	1289	1287	13	14
18	1300	1297	1301	1299	1	-12
19	1306	1308	1309	1307	-7	-8
20	1299	1299	1294	1297	3	10

Sumber : Pengujian

Dari Tabel 5.11 dibuat gambar grafik keluaran seperti dalam Gambar 5.11

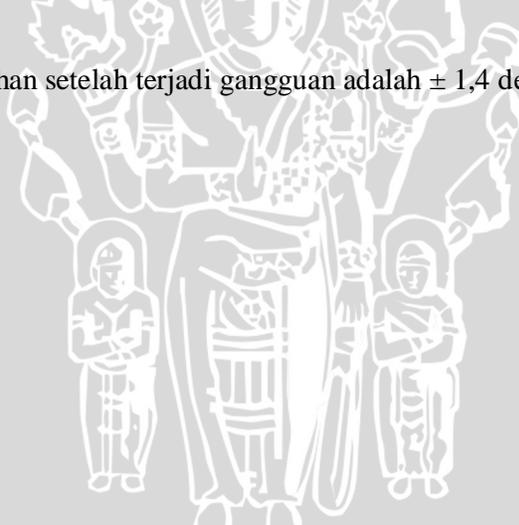


**Gambar 5.11 Grafik Keluaran Rata-rata Kecepatan Motor DC dengan Setpoint 1300 RPM Diberi Beban saat  $t = 10$  detik.**  
(Sumber : Pengujian)

Dalam Gambar 5.11 *set point* berada pada kecepatan 1300 RPM. Toleransi error keadaan mantap ditentukan sebesar 2% dari *set point* atau sebesar 26 RPM. Nilai 10% dari *set point* adalah 130 RPM, untuk 50% adalah 650 RPM, dan 90% adalah 1170 RPM.

Berdasarkan Gambar 5.11 :

1. Waktu untuk pertama kali mencapai nilai 50% dari *set point* atau waktu tunda (td) adalah sebesar  $\pm 0,6$  detik.
2. Waktu untuk pertama kali mencapai nilai 10% dari *set point* adalah  $\pm 0,1$  detik dan waktu untuk pertama kali mencapai nilai 90% dari *set point* adalah  $\pm 1,4$  detik. Sehingga waktu naik (tr) dari nilai 10% sampai dengan 90% adalah  $\pm 1,3$  detik.
3. Waktu untuk pertama kali mencapai keadaan mantap atau waktu menetap (ts) dalam kisaran 2 % dari *set point* adalah  $\pm 2,6$  detik.
4. Nilai *error* terbesar ketika terjadi gangguan adalah 64 RPM atau 4,92% dari *set point*.
5. Waktu pemulihan setelah terjadi gangguan adalah  $\pm 1,4$  detik.

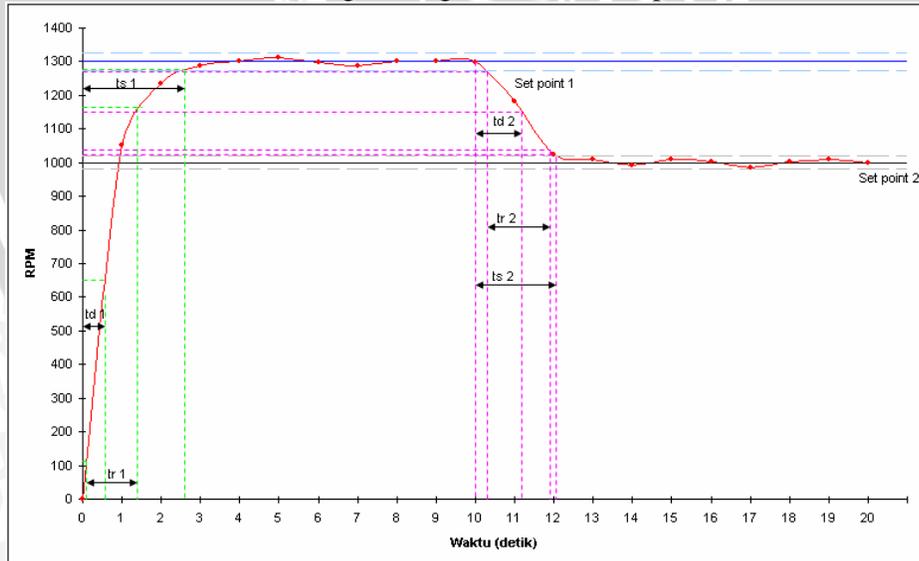


**Tabel 5.12 Hasil Pengujian Perubahan Kecepatan Motor DC Tanpa Beban dari Setpoint 1300 RPM Ke 1000 RPM.**

Waktu (detik)	Kecepatan Motor DC (RPM)			Rata-rata	error	d error
	1	2	3			
0	0	0	0	0	1300	-
1	1053	1050	1054	1052	248	-1052
2	1235	1239	1235	1236	64	-184
3	1287	1287	1289	1287	13	-51
4	1301	1302	1301	1301	-1	-14
5	1310	1311	1314	1311	-11	-10
6	1301	1299	1298	1299	1	12
7	1287	1288	1286	1287	13	12
8	1303	1303	1299	1301	-1	-14
9	1289	1313	1304	1302	-2	-1
10	1302	1300	1294	1298	-298	-296
11	1178	1183	1186	1182	-182	116
12	1021	1024	1026	1023	-23	159
13	1009	1013	1004	1008	-8	15
14	991	995	990	992	8	16
15	1009	1008	1011	1009	-9	-17
16	1001	1002	1003	1002	-2	7
17	981	985	986	984	16	18
18	999	1001	1003	1001	-1	-17
19	1012	1014	1009	1011	-11	-10
20	1003	999	1000	1000	0	11

Sumber : Pengujian

Dari Tabel 5.12 dibuat gambar grafik keluaran seperti dalam Gambar 5.12.



**Gambar 5.12 Grafik Keluaran Rata-rata Kecepatan Motor DC Tanpa Beban dengan Perubahan Kecepatan dari 1300 RPM ke 1000 RPM.**  
(Sumber : Pengujian)

Dalam Gambar 5.12 *set point* pertama berada pada kecepatan 1300 RPM. Toleransi error keadaan mantap ditentukan sebesar 2% dari *set point* atau sebesar 26 RPM. Nilai 10% dari *set point* adalah 130 RPM, untuk 50% adalah 650 RPM, dan 90% adalah 1170 RPM. Sedangkan *set point* kedua berada pada kecepatan 1000 RPM. Toleransi error keadaan mantap ditentukan sebesar 2% dari *set point* atau sebesar 20 RPM. Nilai awal adalah ketika berada pada *set point* pertama atau sebesar 1300 RPM, sehingga rentang kecepatan dari nilai awal menuju *set point* kedua adalah sebesar -300 RPM. Nilai 10% dari *set point* kedua merupakan nilai 10% dari -300 RPM sehingga berada pada kecepatan 1270 RPM. Nilai 90% dari *set point* kedua merupakan nilai dari 90% dari -300 RPM sehingga berada pada kecepatan 1030 RPM. Nilai 50% dari -300 RPM berada pada kecepatan 1150 RPM. Waktu perubahan kecepatan atau perubahan *set point* dari 1300 RPM menjadi 1000 RPM adalah pada saat  $t = 10$  detik.

Berdasarkan Gambar 5.12 :

1. Waktu untuk pertama kali mencapai nilai 50% dari *set point* pertama atau waktu tunda 1 ( $td_1$ ) adalah sebesar  $\pm 0,6$  detik.
2. Waktu untuk pertama kali mencapai nilai 10% dari *set point* pertama adalah  $\pm 0,1$  detik dan waktu untuk pertama kali mencapai nilai 90% dari *set point* pertama adalah  $\pm 1,4$  detik. Sehingga waktu naik 1 ( $tr_1$ ) dari nilai 10% sampai dengan 90% adalah  $\pm 1,3$  detik.
3. Waktu untuk pertama kali mencapai keadaan mantap atau waktu menetap 1 ( $ts_1$ ) dalam kisaran 2 % dari *set point* pertama adalah  $\pm 2,6$  detik.
4. Nilai error keadaan mantap pada *set point* pertama yang terbesar adalah 13 RPM atau 1 % dari *set point*.
5. Waktu pertama kali mencapai 50% dari *set point* pertama menuju *set point* kedua atau waktu tunda 2 ( $td_2$ ) adalah  $\pm 1,2$  detik.
6. Waktu untuk pertama kali mencapai nilai 10% dari *set point* kedua terhadap nilai awal 1300 RPM adalah  $\pm 0,3$  detik dan waktu untuk pertama kali mencapai nilai 90% dari *set point* kedua terhadap nilai awal 1300 RPM adalah  $\pm 1,9$  detik. Sehingga waktu naik 2 ( $tr_2$ ) dari nilai 10% sampai dengan 90% untuk *set point* kedua terhadap nilai awal 1300 RPM adalah  $\pm 1,6$  detik.

7. Waktu untuk pertama kali mencapai keadaan mantap atau waktu menetap 2 (ts 2) dalam kisaran 2 % dari *set point* kedua adalah  $\pm 2,1$  detik.
8. Nilai error keadaan mantap pada *set point* kedua yang terbesar adalah 16 RPM atau 1,6 % dari set point.

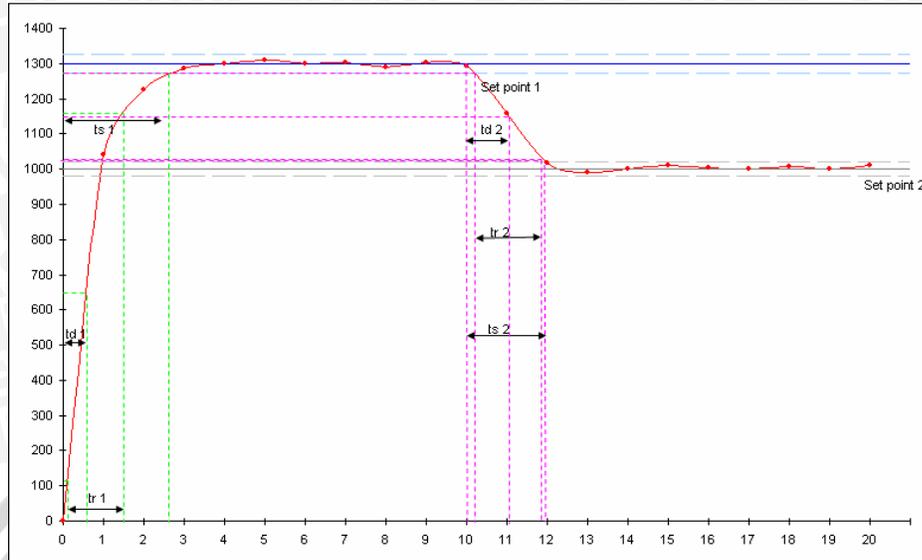
**Tabel 5.13 Hasil Pengujian Perubahan Kecepatan Motor DC Berbeban dari setpoint 1300 RPM ke 1000 RPM.**

Waktu (detik)	Kecepatan Motor DC (RPM)			Rata-rata	error	d error
	1	2	3			
0	0	0	0	0	1300	--
1	1042	1040	1043	1041	259	-1041
2	1224	1227	1223	1224	76	-183
3	1286	1288	1284	1286	14	-62
4	1297	1301	1300	1299	1	-13
5	1308	1307	1309	1308	-8	-9
6	1299	1302	1301	1300	0	8
7	1304	1305	1302	1303	-3	-3
8	1286	1290	1288	1288	12	15
9	1302	1301	1300	1301	-1	-13
10	1299	1287	1288	1291	-291	-290
11	1150	1158	1163	1157	-157	134
12	1012	1015	1021	1016	-16	141
13	998	987	986	990	10	26
14	999	1001	1003	1001	-1	-11
15	1009	1008	1011	1009	-9	-8
16	1005	1002	1003	1003	-3	6
17	998	999	1001	999	1	4
18	1009	1008	1009	1008	-8	-9
19	1001	999	1002	1000	0	8
20	1012	1011	1010	1011	-11	-11

Sumber : Pengujian

Dari Tabel 5.13 dibuat grafik keluaran seperti yang terlihat dalam Gambar

5.13.



**Gambar 5.13 Grafik Keluaran Rata-rata Kecepatan Motor DC Berbeban dan Perubahan Kecepatan dari 1300 RPM ke 1000 RPM.**  
(Sumber : Pengujian)

Dalam Gambar 5.13 *set point* pertama berada pada kecepatan 1300 RPM. Toleransi error keadaan mantap ditentukan sebesar 2% dari *set point* atau sebesar 26 RPM. Nilai 10% dari *set point* adalah 130 RPM, untuk 50% adalah 650 RPM, dan 90% adalah 1170 RPM. Sedangkan *set point* kedua berada pada kecepatan 1000 RPM. Toleransi error keadaan mantap ditentukan sebesar 2% dari *set point* atau sebesar 20 RPM. Nilai awal adalah ketika berada pada *set point* pertama atau sebesar 1300 RPM, sehingga rentang kecepatan dari nilai awal menuju *set point* kedua adalah sebesar -300 RPM. Nilai 10% dari *set point* kedua merupakan nilai 10% dari -300 RPM sehingga berada pada kecepatan 1270 RPM. Nilai 90% dari *set point* kedua merupakan nilai dari 90% dari -300 RPM sehingga berada pada kecepatan 1030 RPM. Nilai 50% dari -300 RPM berada pada kecepatan 1150 RPM. Waktu perubahan kecepatan atau perubahan *set point* dari 1300 RPM menjadi 1000 RPM adalah pada saat  $t = 10$  detik.

Berdasarkan Gambar 5.13 :

1. Waktu untuk pertama kali mencapai nilai 50% dari *set point* pertama atau waktu tunda 1 ( $td_1$ ) adalah sebesar  $\pm 0,6$  detik.
2. Waktu untuk pertama kali mencapai nilai 10% dari *set point* pertama adalah  $\pm 0,1$  detik dan waktu untuk pertama kali mencapai nilai 90% dari

*set point* pertama adalah  $\pm 1,6$  detik. Sehingga waktu naik 1 (*tr 1*) dari nilai 10% sampai dengan 90% adalah  $\pm 1,5$  detik.

3. Waktu untuk pertama kali mencapai keadaan mantap atau waktu menetap 1 (*ts 1*) dalam kisaran 2 % dari *set point* pertama adalah  $\pm 2,6$  detik.
4. Nilai error keadaan mantap pada *set point* pertama yang terbesar adalah 14 RPM atau 1,07 % dari *set point*.
5. Waktu pertama kali mencapai 50% dari *set point* pertama menuju *set point* kedua atau waktu tunda 2 (*td 2*) adalah  $\pm 1,1$  detik.
6. Waktu untuk pertama kali mencapai nilai 10% dari *set point* kedua terhadap nilai awal 1300 RPM adalah  $\pm 0,3$  detik dan waktu untuk pertama kali mencapai nilai 90% dari *set point* kedua terhadap nilai awal 1300 RPM adalah  $\pm 1,8$  detik. Sehingga waktu naik 2 (*tr 2*) dari nilai 10% sampai dengan 90% untuk *set point* kedua terhadap nilai awal 1300 RPM adalah  $\pm 1,5$  detik.
7. Waktu untuk pertama kali mencapai keadaan mantap atau waktu menetap 2 (*ts 2*) dalam kisaran 2 % dari *set point* kedua adalah  $\pm 1,9$  detik.
8. Nilai *error* keadaan mantap pada *set point* kedua yang terbesar adalah 16 RPM atau 1,6 % dari *set point*.

## BAB VI

### KESIMPULAN DAN SARAN

#### 6.1 Kesimpulan

Setelah melakukan perancangan, pembuatan, dan pengujian per blok serta pengujian sistem keseluruhan untuk pengendalian kecepatan motor DC pada alat sablon satu warna otomatis dengan kontrol logika *fuzzy* menggunakan mikrokontroler AT89S52 bisa disimpulkan bahwa :

1. Respon sistem pengendalian kecepatan motor DC tanpa beban dari hasil pengujian sistem adalah sebagai berikut : waktu tunda ( $t_d$ )  $\pm 0,6$  detik, waktu naik ( $t_r$ )  $\pm 1,3$  detik, waktu menetap ( $t_s$ )  $\pm 2,6$  detik, dan error keadaan mantap pada kisaran 1%, seperti yang terlihat dalam Tabel 6.1.

**Tabel 6.1 Respon Sistem Pengendalian Kecepatan Motor DC Tanpa Beban dengan Setpoint 1300 RPM.**

Waktu Tunda (td)	Waktu Naik (tr)	Waktu Menetap (ts)	Error Keadaan Mantap (ess)
$\pm 0,6$ detik	$\pm 1,3$ detik	$\pm 2,6$ detik	1%

Sumber : Pengujian

2. Respon sistem pengendalian kecepatan motor DC berbeban adalah sebagai berikut : waktu tunda ( $t_d$ )  $\pm 0,7$  detik, waktu naik ( $t_r$ )  $\pm 1,4$  detik, waktu menetap ( $t_s$ )  $\pm 2,7$  detik, dan error keadaan mantap pada kisaran 1%, seperti yang terlihat dalam Tabel 6.2.

**Tabel 6.2 Respon Sistem Pengendalian Kecepatan Motor DC Berbeban dengan Setpoint 1300 RPM.**

Waktu Tunda (td)	Waktu Naik (tr)	Waktu Menetap (ts)	Error Keadaan Mantap (ess)
$\pm 0,7$ detik	$\pm 1,4$ detik	$\pm 2,7$ detik	1%

Sumber : Pengujian

3. Respon sistem pengendalian kecepatan motor DC ketika diberi gangguan saat  $t = 10$  detik adalah sebagai berikut : waktu tunda ( $t_d$ )  $\pm 0,6$  detik, waktu naik ( $t_r$ )  $\pm 1,3$  detik, waktu menetap ( $t_s$ )  $\pm 2,6$  detik, waktu pemulihan  $\pm 1,4$  detik dan error keadaan mantap pada kisaran 1%, seperti yang terlihat dalam Tabel 6.3.

**Tabel 6.3 Respon Sistem Pengendalian Kecepatan Motor DC dengan Setpoint 1300 RPM Diberi Beban Pada Saat  $t = 10$  detik.**

Waktu Tunda (td)	Waktu Naik (tr)	Waktu Menetap (ts)	Error Keadaan Mantap (ess)	Waktu Pemulihan
$\pm 0,6$ detik	$\pm 1,3$ detik	$\pm 2,6$ detik	1%	$\pm 1,4$ detik

Sumber : Pengujian

4. Respon sistem pengendalian kecepatan motor DC tanpa beban dengan perubahan kecepatan dari setpoint 1300 RPM ke 1000 RPM adalah sebagai berikut : waktu tunda ( $t_d$ )  $\pm 1,2$  detik, waktu naik ( $t_r$ )  $\pm 1,6$  detik, waktu menetap ( $t_s$ )  $\pm 2,1$  detik, dan error keadaan mantap pada kisaran 1,6%, seperti yang terlihat dalam Tabel 6.4.

**Tabel 6.4 Respon Sistem Pengendalian Kecepatan Motor DC Tanpa Beban Dengan Perubahan Kecepatan dari Setpoint 1300 RPM Ke 1000 RPM.**

Set Point	Waktu Tunda (td)	Waktu Naik (tr)	Waktu Menetap (ts)	Error Keadaan Mantap (ess)
1300 RPM	$\pm 0,6$ detik	$\pm 1,3$ detik	$\pm 2,6$ detik	1%
1000 RPM	$\pm 1,2$ detik	$\pm 1,6$ detik	$\pm 2,1$ detik	1,6%

Sumber : Pengujian

5. Respon sistem pengendalian kecepatan motor DC berbeban dengan perubahan kecepatan dari setpoint 1300 RPM ke 1000 RPM adalah sebagai berikut : waktu tunda ( $t_d$ )  $\pm 1,1$  detik, waktu naik ( $t_r$ )  $\pm 1,5$  detik, waktu menetap ( $t_s$ )  $\pm 1,9$  detik, dan error keadaan mantap pada kisaran 1,6%, seperti yang terlihat dalam Tabel 6.5.

**Tabel 6.5 Respon Sistem Pengendalian Kecepatan Motor DC Berbeban Dengan Perubahan Kecepatan dari Setpoint 1300 RPM Ke 1000 RPM.**

Set Point	Waktu Tunda (td)	Waktu Naik (tr)	Waktu Menetap (ts)	Error Keadaan Mantap (ess)
1300 RPM	$\pm 0,6$ detik	$\pm 1,5$ detik	$\pm 2,6$ detik	1%
1000 RPM	$\pm 1,1$ detik	$\pm 1,5$ detik	$\pm 1,9$ detik	1,6%

Sumber : Pengujian

## 6.2 Saran

Masih banyak kekurangan dan kelemahan alat sablon satu warna yang telah dirancang dan dibuat. Untuk perbaikan dan pengembangan lebih lanjut dari alat ini disarankan agar :

1. Disarankan untuk menggunakan komponen sensor obyek gambar dengan dimensi yang lebih kecil, sehingga akurasi gambar sablon yang akan dihasilkan menjadi lebih baik.

2. Disarankan untuk menggunakan komponen penyemprot tinta yang lebih baik dan bisa lebih dari satu warna, sehingga gambar yang dihasilkan bisa benar-benar sesuai dengan obyek gambar yang diinginkan.
3. Disarankan untuk melakukan pengembangan agar alat bisa diaplikasikan menggunakan program printer dari komputer, sehingga obyek gambar yang akan dihasilkan berdasarkan perintah print yang berasal dari kebanyakan program aplikasi di komputer.

UNIVERSITAS BRAWIJAYA



## DAFTAR PUSTAKA

- Adler, M., dan Surtell, T., 2000. *Pulse Width Modulation*, (Online) (<http://www.eleinmec.com>) diakses tanggal 10 Juni 2007.
- Atmel Corporation, 2001. *8-bit Microcontroller with 8K Bytes In-System Programming Flash AT98S52*, (Online) (<http://www.atmel.com>) diakses tanggal 6 Februari 2007.
- Harris Semiconductor, 1998. *82C55A CMOS Programable Peripheral Interface*, (Online) (<http://www.harris.com>) diakses tanggal 6 Februari 2007.
- Jamshidi, M., Vadiie, N., dan Ross, T. J., 1993. *Fuzzy Logic and Control Software and Hardware Applications*, New Jersey: Prentice-Hall, Inc.
- Kamal, I., 2007. *Build a 5A H-Bridge Motor Controller!*, (Online) (<http://www.ikalologic.com>) diakses tanggal 16 Juni 2007.
- Kissel, T. E., 2000. *Industrial Electronic*, Prentice-Hall, Inc, (Online) (<http://zone.ni.com>) dikases tanggal 16 Juni 2007.
- Meggs, P. B.A., 1998. *A History of Graphic Design*, Jhon Wiley and Sons, Inc, (Online) (<http://en.wikipedia.org>) diakses tanggal 10 Juni 2007.
- Mismail, B., 1997. *Rangkaian Listrik Jilid II*, Bandung : Penerbit ITB.
- National Semiconductor, 2002. *ADC 0808/ADC 0809 8-Bit  $\mu$ P Compatible A/D Converters with 8-Channel Multiplexer*, (Online) (<http://www.national.com>) diakses tanggal 2 Februari 2007.
- Ogata, K., 1995. *Teknik Kontrol Automatik*, Jakarta : Penerbit Erlangga.
- Pitowarno, E. 1992. *Sistem Interfacing*, Surabaya : Politeknik Elektronika Surabaya.
- Putra, E. A., 2002. *Teknik Antarmuka Komputer : Konsep dan Aplikasi*, Yogyakarta : Graha Ilmu.
- Ross, T. J. 1997. *Fuzzy Logic with Engineering Applications, International Edition*, Singapura: McGraw-Hill, Inc.
- Saunders, G., dan Miles, R., 2006. *Prints Now: Directions and Definitions*, Victoria and Abert Museum, (Online) (<http://en.wikipedia.org>) diakses tanggal 10 Juni 2007.

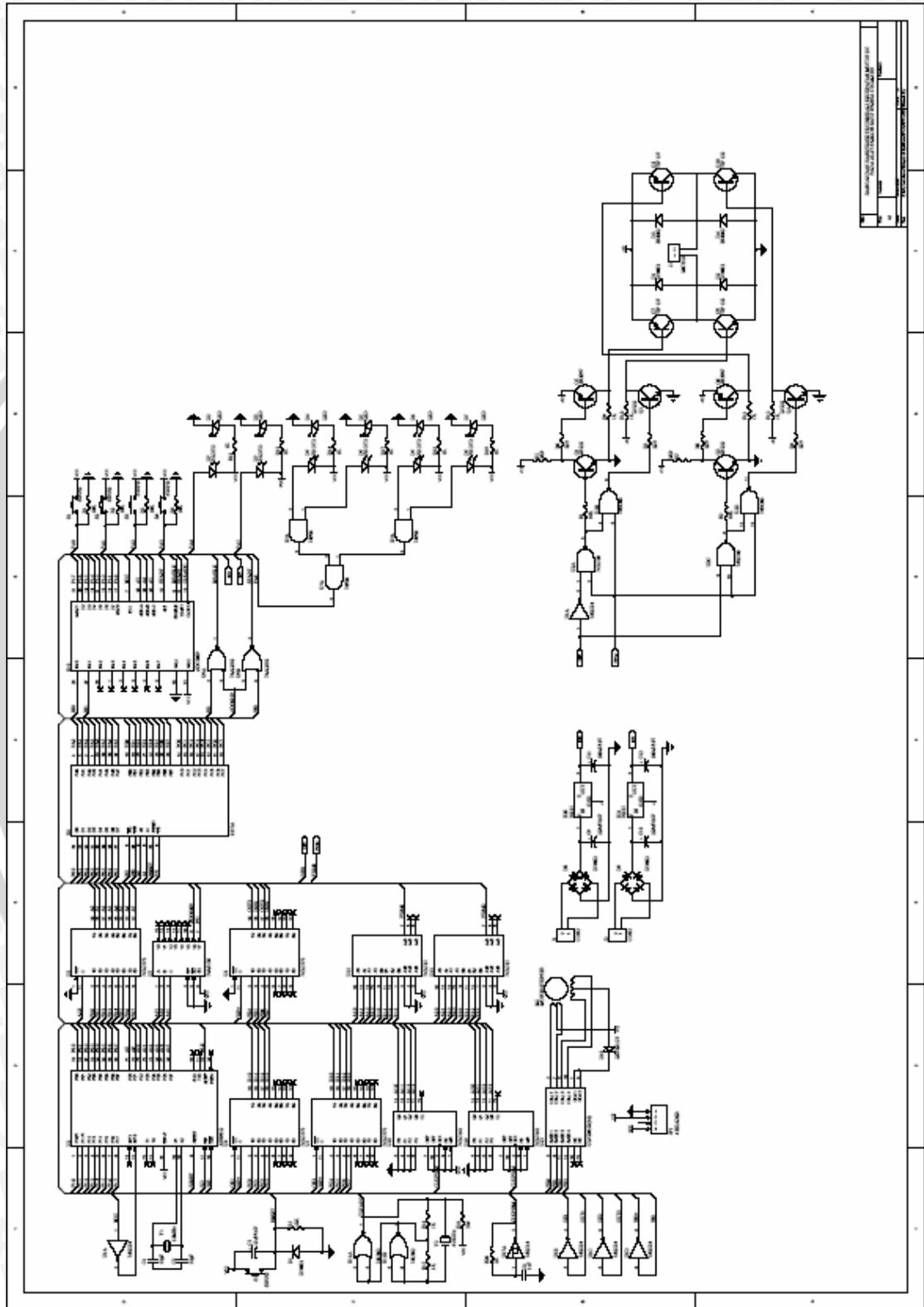
- Zheludev, N., 2007. *The Life Times of The LED –a 100- Year History*, Nature Photonics, (Online) (<http://en.wikipedia.org>) diakses tanggal 2 Juni 2007.
- Zuhal, 1992. *Dasar Teknik Tenaga Listrik dan Elektronika Daya*, Jakarta: PT. Gramedia Pustaka Utama.



FOTO ALAT



SCHEMATIK ALAT KESELURUHAN



### Features

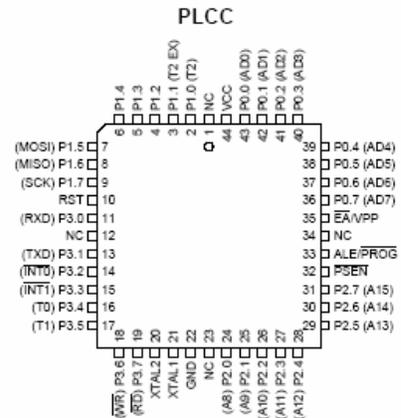
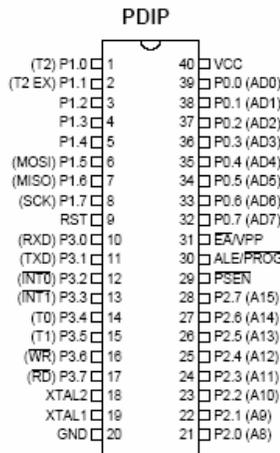
- Compatible with MCS-51® Products
- 8K Bytes of In-System Programmable (ISP) Flash Memory
  - Endurance: 1000 Write/Erase Cycles
- 4.0V to 5.5V Operating Range
- Fully Static Operation: 0 Hz to 33 MHz
- Three-level Program Memory Lock
- 256 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Three 16-bit Timer/Counters
- Eight Interrupt Sources
- Full Duplex UART Serial Channel
- Low-power Idle and Power-down Modes
- Interrupt Recovery from Power-down Mode
- Watchdog Timer
- Dual Data Pointer
- Power-off Flag
- Fast Programming Time
- Flexible ISP Programming (Byte and Page Mode)

### Description

The AT89S52 is a low-power, high-performance CMOS 8-bit microcontroller with 8K bytes of in-system programmable Flash memory. The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard 80C51 instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with in-system programmable Flash on a monolithic chip, the Atmel AT89S52 is a powerful microcontroller which provides a highly-flexible and cost-effective solution to many embedded control applications.

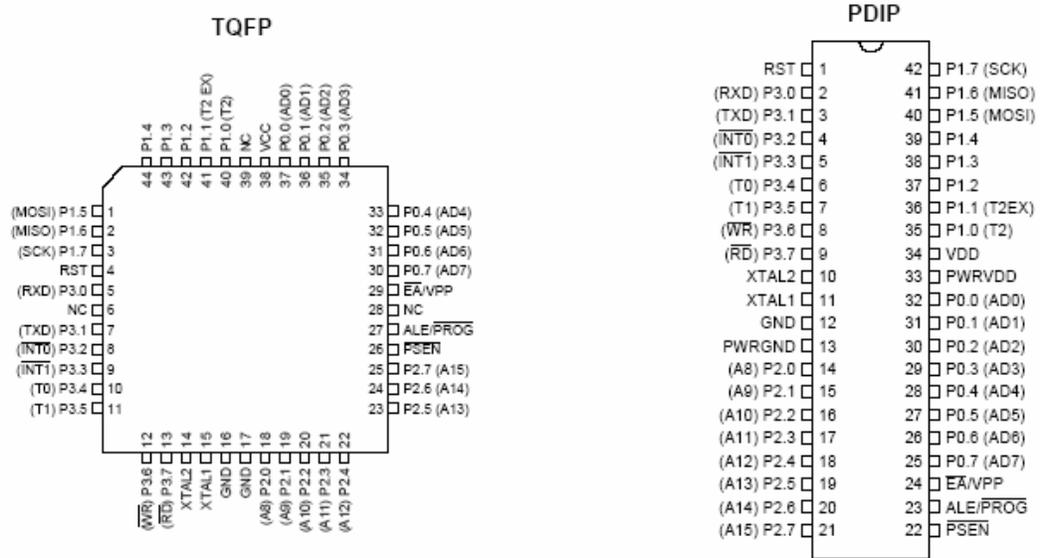
The AT89S52 provides the following standard features: 8K bytes of Flash, 256 bytes of RAM, 32 I/O lines, Watchdog timer, two data pointers, three 16-bit timer/counters, a six-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89S52 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. The Power-down mode saves the RAM contents but freezes the oscillator, disabling all other chip functions until the next interrupt or hardware reset.

### Pin Configurations

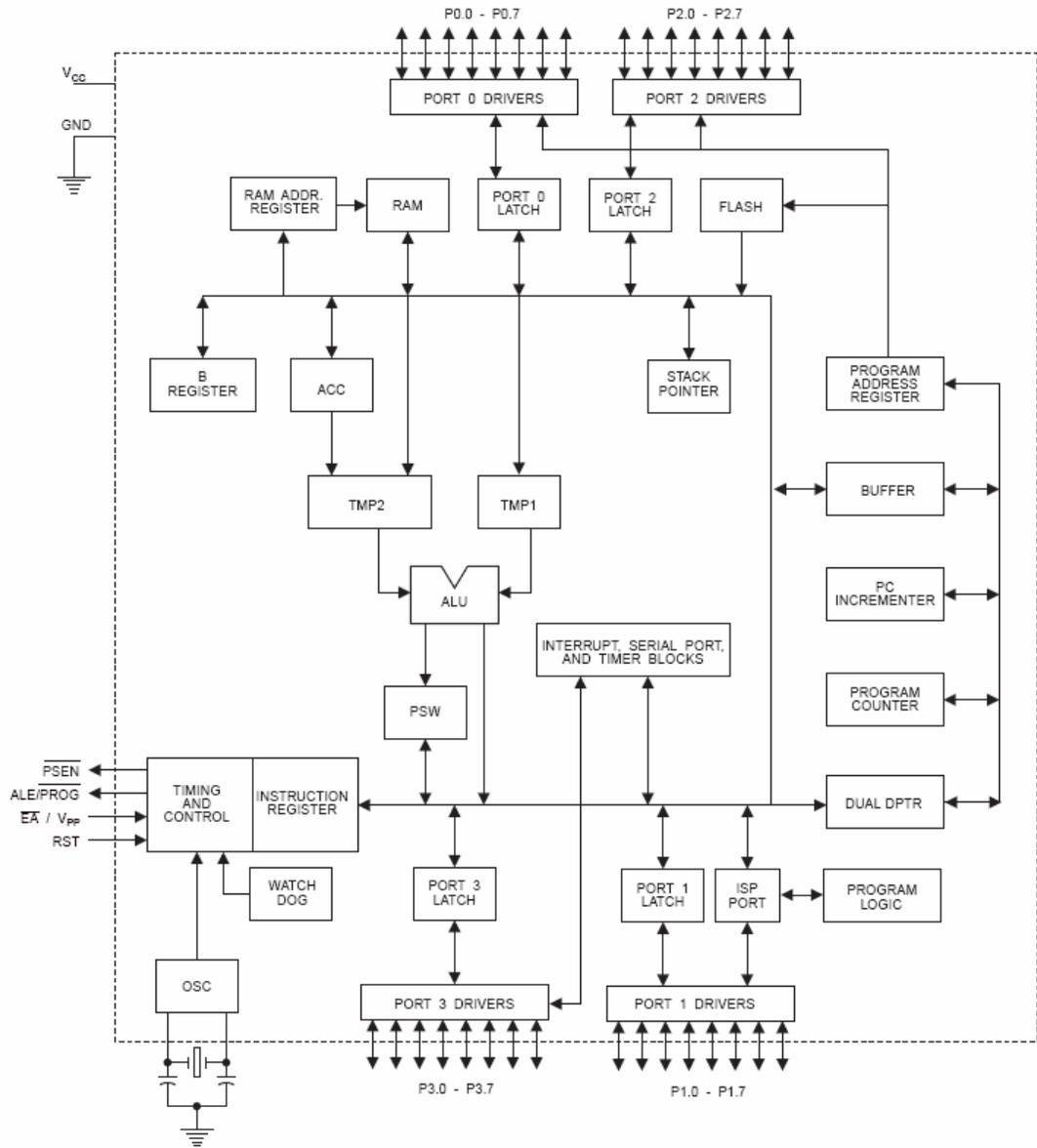


**8-bit  
Microcontroller  
with 8K Bytes  
In-System  
Programmable  
Flash**

**AT89S52**



Block Diagram



### Pin Description

VCC	Supply voltage.
GND	Ground.
Port 0	<p>Port 0 is an 8-bit open drain bidirectional I/O port. As an output port, each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs.</p> <p>Port 0 can also be configured to be the multiplexed low-order address/data bus during accesses to external program and data memory. In this mode, P0 has internal pull-ups.</p> <p>Port 0 also receives the code bytes during Flash programming and outputs the code bytes during program verification. <b>External pull-ups are required during program verification.</b></p>
Port 1	<p>Port 1 is an 8-bit bidirectional I/O port with internal pull-ups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins, they are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (<math>I_{IL}</math>) because of the internal pull-ups.</p> <p>In addition, P1.0 and P1.1 can be configured to be the timer/counter 2 external count input (P1.0/T2) and the timer/counter 2 trigger input (P1.1/T2EX), respectively, as shown in the following table.</p> <p>Port 1 also receives the low-order address bytes during Flash programming and verification.</p>

Port Pin	Alternate Functions
P1.0	T2 (external count input to Timer/Counter 2), clock-out
P1.1	T2EX (Timer/Counter 2 capture/reload trigger and direction control)
P1.5	MOSI (used for In-System Programming)
P1.6	MISO (used for In-System Programming)
P1.7	SCK (used for In-System Programming)

Port 2	<p>Port 2 is an 8-bit bidirectional I/O port with internal pull-ups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins, they are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (<math>I_{IL}</math>) because of the internal pull-ups.</p> <p>Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ DPTR). In this application, Port 2 uses strong internal pull-ups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register.</p> <p>Port 2 also receives the high-order address bits and some control signals during Flash programming and verification.</p>
Port 3	<p>Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins, they are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (<math>I_{IL}</math>) because of the pull-ups.</p> <p>Port 3 receives some control signals for Flash programming and verification.</p> <p>Port 3 also serves the functions of various special features of the AT89S52, as shown in the following table.</p>

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{INT0}$ (external interrupt 0)
P3.3	$\overline{INT1}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	$\overline{WR}$ (external data memory write strobe)
P3.7	$\overline{RD}$ (external data memory read strobe)

RST

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device. This pin drives high for 98 oscillator periods after the Watchdog times out. The DISRTO bit in SFR AUXR (address 8EH) can be used to disable this feature. In the default state of bit DISRTO, the RESET HIGH out feature is enabled.

ALE/ $\overline{\text{PROG}}$

Address Latch Enable (ALE) is an output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input ( $\overline{\text{PROG}}$ ) during Flash programming.

In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external data memory.

If desired, ALE operation can be disabled by setting bit 0 of SFR location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is weakly pulled high. Setting the ALE-disable bit has no effect if the microcontroller is in external execution mode.

$\overline{\text{PSEN}}$

Program Store Enable ( $\overline{\text{PSEN}}$ ) is the read strobe to external program memory.

When the AT89S52 is executing code from external program memory,  $\overline{\text{PSEN}}$  is activated twice each machine cycle, except that two  $\overline{\text{PSEN}}$  activations are skipped during each access to external data memory.

$\overline{\text{EA}}/\text{VPP}$

External Access Enable.  $\overline{\text{EA}}$  must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed,  $\overline{\text{EA}}$  will be internally latched on reset.

$\overline{\text{EA}}$  should be strapped to  $V_{\text{CC}}$  for internal program executions.

This pin also receives the 12-volt programming enable voltage ( $V_{\text{PP}}$ ) during Flash programming.

XTAL1

Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

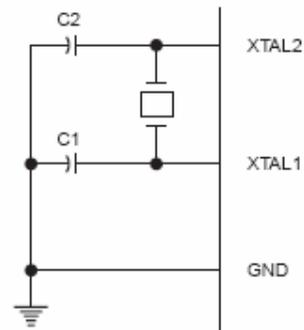
XTAL2

Output from the inverting oscillator amplifier.

**Oscillator Characteristics**

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier that can be configured for use as an on-chip oscillator, as shown in Figure 7. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven, as shown in Figure 8. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

Figure 7. Oscillator Connections



Note: 1. C1, C2 = 30 pF ± 10 pF for Crystals  
 = 40 pF ± 10 pF for Ceramic Resonators

**Absolute Maximum Ratings\***

Operating Temperature.....	-55°C to +125°C
Storage Temperature.....	-65°C to +150°C
Voltage on Any Pin with Respect to Ground.....	-1.0V to +7.0V
Maximum Operating Voltage.....	6.6V
DC Output Current.....	15.0 mA

\*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

### DC Characteristics

The values shown in this table are valid for  $T_A = -40^\circ\text{C}$  to  $85^\circ\text{C}$  and  $V_{CC} = 4.0\text{V}$  to  $5.5\text{V}$ , unless otherwise noted.

Symbol	Parameter	Condition	Min	Max	Units
$V_{IL}$	Input Low Voltage	(Except EA)	-0.5	$0.2 V_{CC} - 0.1$	V
$V_{IL1}$	Input Low Voltage ( $\overline{\text{EA}}$ )		-0.5	$0.2 V_{CC} - 0.3$	V
$V_{IH}$	Input High Voltage	(Except XTAL1, RST)	$0.2 V_{CC} + 0.9$	$V_{CC} + 0.5$	V
$V_{IH1}$	Input High Voltage	(XTAL1, RST)	$0.7 V_{CC}$	$V_{CC} + 0.5$	V
$V_{OL}$	Output Low Voltage <sup>(1)</sup> (Ports 1,2,3)	$I_{OL} = 1.6 \text{ mA}$		0.45	V
$V_{OL1}$	Output Low Voltage <sup>(1)</sup> (Port 0, ALE, PSEN)	$I_{OL} = 3.2 \text{ mA}$		0.45	V
$V_{OH}$	Output High Voltage (Ports 1,2,3, ALE, PSEN)	$I_{OH} = -60 \mu\text{A}$ , $V_{CC} = 5\text{V} \pm 10\%$	2.4		V
		$I_{OH} = -25 \mu\text{A}$	$0.75 V_{CC}$		V
		$I_{OH} = -10 \mu\text{A}$	$0.9 V_{CC}$		V
$V_{OH1}$	Output High Voltage (Port 0 in External Bus Mode)	$I_{OH} = -800 \mu\text{A}$ , $V_{CC} = 5\text{V} \pm 10\%$	2.4		V
		$I_{OH} = -300 \mu\text{A}$	$0.75 V_{CC}$		V
		$I_{OH} = -80 \mu\text{A}$	$0.9 V_{CC}$		V
$I_L$	Logical 0 Input Current (Ports 1,2,3)	$V_{IN} = 0.45\text{V}$		-50	$\mu\text{A}$
$I_{TL}$	Logical 1 to 0 Transition Current (Ports 1,2,3)	$V_{IN} = 2\text{V}$ , $V_{CC} = 5\text{V} \pm 10\%$		-650	$\mu\text{A}$
$I_{LI}$	Input Leakage Current (Port 0, $\overline{\text{EA}}$ )	$0.45 < V_{IN} < V_{CC}$		$\pm 10$	$\mu\text{A}$
RRST	Reset Pulldown Resistor		50	300	K $\Omega$
$C_{IO}$	Pin Capacitance	Test Freq. = 1 MHz, $T_A = 25^\circ\text{C}$		10	pF
$I_{CC}$	Power Supply Current	Active Mode, 12 MHz		25	mA
		Idle Mode, 12 MHz		6.5	mA
	Power-down Mode <sup>(1)</sup>	$V_{CC} = 5.5\text{V}$		50	$\mu\text{A}$

Notes: 1. Under steady state (non-transient) conditions,  $I_{OL}$  must be externally limited as follows:

Maximum  $I_{OL}$  per port pin: 10 mA

Maximum  $I_{OL}$  per 8-bit port:

Port 0: 26 mA      Ports 1, 2, 3: 15 mA

Maximum total  $I_{OL}$  for all output pins: 71 mA

If  $I_{OL}$  exceeds the test condition,  $V_{OL}$  may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.

2. Minimum  $V_{CC}$  for Power-down is 2V.





# 82C55A

## CMOS Programmable Peripheral Interface

June 1998

### Features

- Pin Compatible with NMOS 8255A
- 24 Programmable I/O Pins
- Fully TTL Compatible
- High Speed, No "Wait State" Operation with 5MHz and 8MHz 80C86 and 80C88
- Direct Bit Set/Reset Capability
- Enhanced Control Word Read Capability
- L7 Process
- 2.5mA Drive Capability on All I/O Ports
- Low Standby Power (ICCSB) .....10µA

### Description

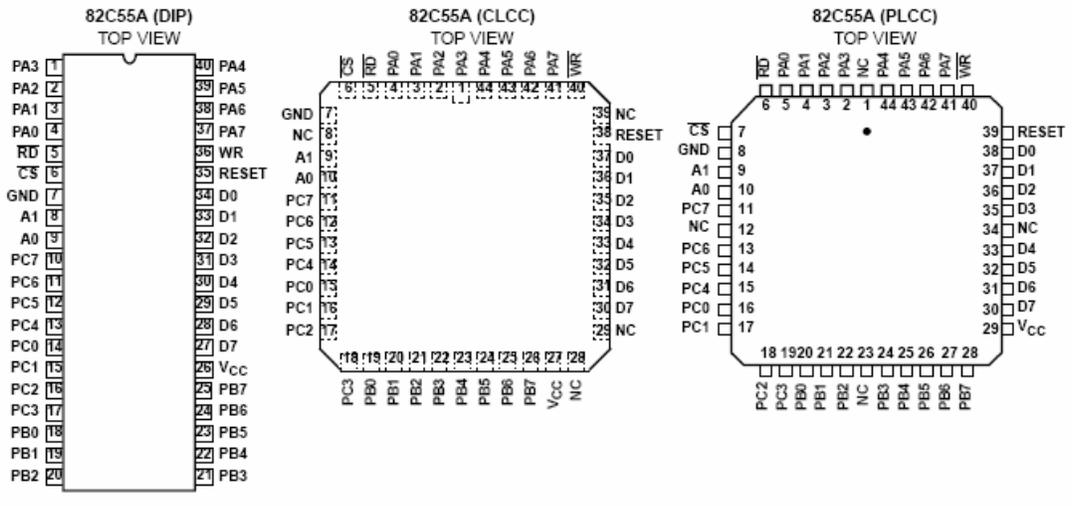
The Harris 82C55A is a high performance CMOS version of the industry standard 8255A and is manufactured using a self-aligned silicon gate CMOS process (Scaled SAJI IV). It is a general purpose programmable I/O device which may be used with many different microprocessors. There are 24 I/O pins which may be individually programmed in 2 groups of 12 and used in 3 major modes of operation. The high performance and industry standard configuration of the 82C55A make it compatible with the 80C86, 80C88 and other microprocessors.

Static CMOS circuit design insures low operating power. TTL compatibility over the full military temperature range and bus hold circuitry eliminate the need for pull-up resistors. The Harris advanced SAJI process results in performance equal to or greater than existing functionally equivalent products at a fraction of the power.

### Ordering Information

PART NUMBERS		PACKAGE	TEMPERATURE RANGE	PKG. NO.
5MHz	8MHz			
CP82C55A-5	CP82C55A	40 Ld PDIP	0°C to 70°C	E40.6
IP82C55A-5	IP82C55A		-40°C to 85°C	E40.6
CS82C55A-5	CS82C55A	44 Ld PLCC	0°C to 70°C	N44.85
IS82C55A-5	IS82C55A		-40°C to 85°C	N44.85
CD82C55A-5	CD82C55A	40 Ld CERDIP	0°C to 70°C	F40.6
ID82C55A-5	ID82C55A		-40°C to 85°C	F40.6
MD82C55A-6/B	MD82C55A/B	SMD#	-55°C to 125°C	F40.6
8406801QA	8406802QA			F40.6
MR82C55A-6/B	MR82C55A/B	44 Pad CLCC	-55°C to 125°C	J44.A
8406801XA	8406802XA		SMD#	J44.A

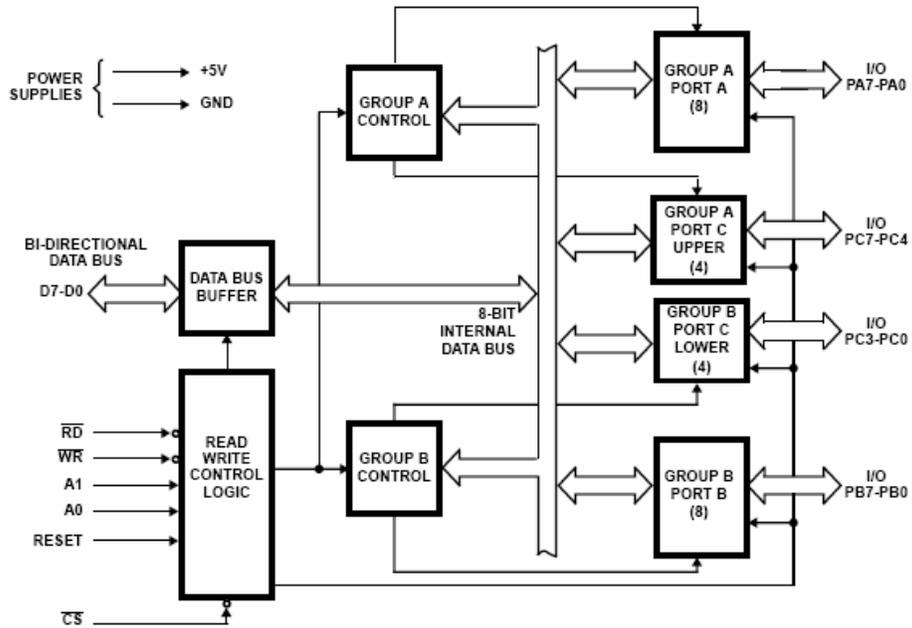
### Pinouts



**Pin Description**

SYMBOL	PIN NUMBER	TYPE	DESCRIPTION
V <sub>CC</sub>	26		V <sub>CC</sub> : The +5V power supply pin. A 0.1µF capacitor between pins 26 and 7 is recommended for decoupling.
GND	7		GROUND
D0-D7	27-34	I/O	DATA BUS: The Data Bus lines are bidirectional three-state pins connected to the system data bus.
RESET	35	I	RESET: A high on this input clears the control register and all ports (A, B, C) are set to the input mode with the "Bus Hold" circuitry turned on.
$\overline{CS}$	6	I	CHIP SELECT: Chip select is an active low input used to enable the 82C55A onto the Data Bus for CPU communications.
$\overline{RD}$	5	I	READ: Read is an active low input control signal used by the CPU to read status information or data via the data bus.
$\overline{WR}$	36	I	WRITE: Write is an active low input control signal used by the CPU to load control words and data into the 82C55A.
A0-A1	8, 9	I	ADDRESS: These input signals, in conjunction with the $\overline{RD}$ and $\overline{WR}$ inputs, control the selection of one of the three ports or the control word register. A0 and A1 are normally connected to the least significant bits of the Address Bus A0, A1.
PA0-PA7	1-4, 37-40	I/O	PORT A: 8-bit input and output port. Both bus hold high and bus hold low circuitry are present on this port.
PB0-PB7	18-25	I/O	PORT B: 8-bit input and output port. Bus hold high circuitry is present on this port.
PC0-PC7	10-17	I/O	PORT C: 8-bit input and output port. Bus hold circuitry is present on this port.

**Functional Diagram**





**Functional Description**

**Data Bus Buffer**

This three-state bi-directional 8-bit buffer is used to interface the 82C55A to the system data bus. Data is transmitted or received by the buffer upon execution of input or output instructions by the CPU. Control words and status information are also transferred through the data bus buffer.

**Read/Write and Control Logic**

The function of this block is to manage all of the internal and external transfers of both Data and Control or Status words. It accepts inputs from the CPU Address and Control busses and in turn, issues commands to both of the Control Groups.

**(CS)** Chip Select. A "low" on this input pin enables the communication between the 82C55A and the CPU.

**(RD)** Read. A "low" on this input pin enables 82C55A to send the data or status information to the CPU on the data bus. In essence, it allows the CPU to "read from" the 82C55A.

**(WR)** Write. A "low" on this input pin enables the CPU to write data or control words into the 82C55A.

**(A0 and A1)** Port Select 0 and Port Select 1. These input signals, in conjunction with the RD and WR inputs, control the selection of one of the three ports or the control word register. They are normally connected to the least significant bits of the address bus (A0 and A1).

82C55A BASIC OPERATION

A1	A0	RD	WR	CS	INPUT OPERATION (READ)
0	0	0	1	0	Port A → Data Bus
0	1	0	1	0	Port B → Data Bus
1	0	0	1	0	Port C → Data Bus
1	1	0	1	0	Control Word → Data Bus
OUTPUT OPERATION (WRITE)					
0	0	1	0	0	Data Bus → Port A
0	1	1	0	0	Data Bus → Port B
1	0	1	0	0	Data Bus → Port C
1	1	1	0	0	Data Bus → Control
DISABLE FUNCTION					
X	X	X	X	1	Data Bus → Three-State
X	X	1	1	0	Data Bus → Three-State

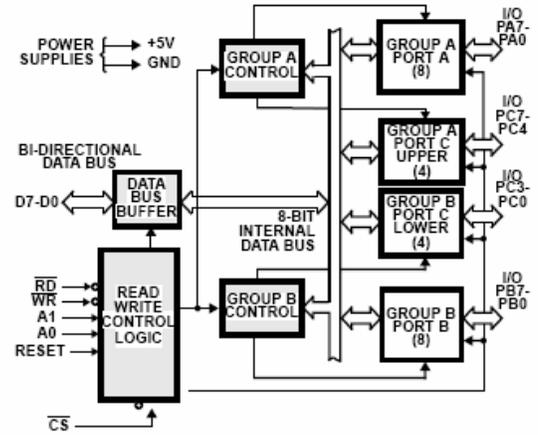


FIGURE 1. 82C55A BLOCK DIAGRAM. DATA BUS BUFFER, READ/WRITE, GROUP A & B CONTROL LOGIC FUNCTIONS

**(RESET)** Reset. A "high" on this input initializes the control register to 9Bh and all ports (A, B, C) are set to the input mode. "Bus hold" devices internal to the 82C55A will hold the I/O port inputs to a logic "1" state with a maximum hold current of 400µA.

**Group A and Group B Controls**

The functional configuration of each port is programmed by the systems software. In essence, the CPU "outputs" a control word to the 82C55A. The control word contains information such as "mode", "bit set", "bit reset", etc., that initializes the functional configuration of the 82C55A.

Each of the Control blocks (Group A and Group B) accepts "commands" from the Read/Write Control logic, receives "control words" from the internal data bus and issues the proper commands to its associated ports.

Control Group A - Port A and Port C upper (C7 - C4)

Control Group B - Port B and Port C lower (C3 - C0)

The control word register can be both written and read as shown in the "Basic Operation" table. Figure 4 shows the control word format for both Read and Write operations. When the control word is read, bit D7 will always be a logic "1", as this implies control word mode information.

The modes for Port A and Port B can be separately defined, while Port C is divided into two portions as required by the Port A and Port B definitions. All of the output registers, including the status flip-flops, will be reset whenever the mode is changed. Modes may be combined so that their functional definition can be "tailored" to almost any I/O structure. For instance: Group B can be programmed in Mode 0 to monitor simple switch closings or display computational results, Group A could be programmed in Mode 1 to monitor a keyboard or tape reader on an interrupt-driven basis.

The mode definitions and possible mode combinations may seem confusing at first, but after a cursory review of the complete device operation a simple, logical I/O approach will surface. The design of the 82C55A has taken into account things such as efficient PC board layout, control signal definition vs. PC layout and complete functional flexibility to support almost any peripheral device with no external logic. Such design represents the maximum use of the available pins.

**Single Bit Set/Reset Feature (Figure 5)**

Any of the eight bits of Port C can be Set or Reset using a single Output instruction. This feature reduces software requirements in control-based applications.

When Port C is being used as status/control for Port A or B, these bits can be set or reset by using the Bit Set/Reset operation just as if they were output ports.

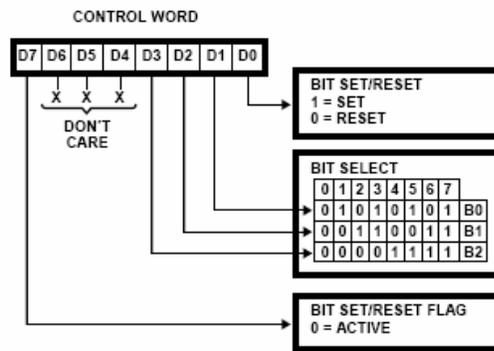


FIGURE 5. BIT SET/RESET FORMAT

**Interrupt Control Functions**

When the 82C55A is programmed to operate in mode 1 or mode 2, control signals are provided that can be used as interrupt request inputs to the CPU. The interrupt request signals, generated from port C, can be inhibited or enabled by setting or resetting the associated INTE flip-flop, using the bit set/reset function of port C.

This function allows the programmer to enable or disable a CPU interrupt by a specific I/O device without affecting any other device in the interrupt structure.

**INTE Flip-Flop Definition**

(BIT-SET)-INTE is SET - Interrupt Enable

(BIT-RESET)-INTE is Reset - Interrupt Disable

NOTE: All Mask flip-flops are automatically reset during mode selection and device Reset.

**Operating Modes**

**Mode 0 (Basic Input/Output).** This functional configuration provides simple input and output operations for each of the three ports. No handshaking is required, data is simply written to or read from a specific port.

Mode 0 Basic Functional Definitions:

- Two 8-bit ports and two 4-bit ports
- Any Port can be input or output
- Outputs are latched
- Input are not latched
- 16 different Input/Output configurations possible

MODE 0 PORT DEFINITION

A		B		GROUP A		GROUP B		
D4	D3	D1	D0	PORT A	PORTC (Upper)	#	PORT B	PORTC (Lower)
0	0	0	0	Output	Output	0	Output	Output
0	0	0	1	Output	Output	1	Output	Input
0	0	1	0	Output	Output	2	Input	Output
0	0	1	1	Output	Output	3	Input	Input
0	1	0	0	Output	Input	4	Output	Output
0	1	0	1	Output	Input	5	Output	Input
0	1	1	0	Output	Input	6	Input	Output
0	1	1	1	Output	Input	7	Input	Input
1	0	0	0	Input	Output	8	Output	Output
1	0	0	1	Input	Output	9	Output	Input
1	0	1	0	Input	Output	10	Input	Output
1	0	1	1	Input	Output	11	Input	Input
1	1	0	0	Input	Input	12	Output	Output
1	1	0	1	Input	Input	13	Output	Input
1	1	1	0	Input	Input	14	Input	Output
1	1	1	1	Input	Input	15	Input	Input

**Absolute Maximum Ratings**  $T_A = 25^\circ\text{C}$

Supply Voltage .....	+8.0V
Input, Output or I/O Voltage .....	GND-0.5V to $V_{CC}+0.5V$
ESD Classification .....	Class 1

**Operating Conditions**

Voltage Range .....	+4.5V to 5.5V
Operating Temperature Range	
C82C55A .....	0°C to 70°C
I82C55A .....	-40°C to 85°C
M82C55A .....	-55°C to 125°C

**Thermal Information**

Thermal Resistance (Typical, Note 1)	$\theta_{JA}$	$\theta_{JC}$
CERDIP Package .....	50°C/W	10°C/W
CLCC Package .....	65°C/W	14°C/W
PDIP Package .....	50°C/W	N/A
PLCC Package .....	46°C/W	N/A
Maximum Storage Temperature Range .....	-65°C to 150°C	
Maximum Junction Temperature		
CDIP Package .....	175°C	
PDIP Package .....	150°C	
Maximum Lead Temperature (Soldering 10s) .....	300°C (PLCC Lead Tips Only)	

**Die Characteristics**

Gate Count .....	1000 Gates
------------------	------------

*CAUTION: Stresses above those listed in "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress only rating and operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.*

**NOTE:**

- $\theta_{JA}$  is measured with the component mounted on an evaluation PC board in free air.

**Electrical Specifications**  $V_{CC} = 5.0V \pm 10\%$ ;  $T_A = 0^\circ\text{C}$  to  $+70^\circ\text{C}$  (C82C55A);  
 $T_A = -40^\circ\text{C}$  to  $+85^\circ\text{C}$  (I82C55A);  
 $T_A = -55^\circ\text{C}$  to  $+125^\circ\text{C}$  (M82C55A)

SYMBOL	PARAMETER	LIMITS		UNITS	TEST CONDITIONS
		MIN	MAX		
$V_{IH}$	Logical One Input Voltage	2.0 2.2	-	V	I82C55A, C82C55A, M82C55A
$V_{IL}$	Logical Zero Input Voltage	-	0.8	V	
$V_{OH}$	Logical One Output Voltage	3.0 $V_{CC}-0.4$	-	V	$I_{OH} = -2.5\text{mA}$ , $I_{OH} = -100\mu\text{A}$
$V_{OL}$	Logical Zero Output Voltage	-	0.4	V	$I_{OL} +2.5\text{mA}$
$I_I$	Input Leakage Current	-1.0	+1.0	$\mu\text{A}$	$V_{IN} = V_{CC}$ or GND, DIP Pins: 5, 6, 8, 9, 35, 36
$I_O$	I/O Pin Leakage Current	-10	+10	$\mu\text{A}$	$V_O = V_{CC}$ or GND DIP Pins: 27 - 34
IBHH	Bus Hold High Current	-50	-400	$\mu\text{A}$	$V_O = 3.0V$ . Ports A, B, C
IBHL	Bus Hold Low Current	50	400	$\mu\text{A}$	$V_O = 1.0V$ . Port A ONLY
IDAR	Darlington Drive Current	-2.5	Note 2, 4	mA	Ports A, B, C. Test Condition 3
ICCSB	Standby Power Supply Current	-	10	$\mu\text{A}$	$V_{CC} = 5.5V$ , $V_{IN} = V_{CC}$ or GND. Output Open
ICCP	Operating Power Supply Current	-	1	mA/MHz	$T_A = +25^\circ\text{C}$ , $V_{CC} = 5.0V$ , Typical (See Note 3)

**NOTES:**

- No internal current limiting exists on Port Outputs. A resistor must be added externally to limit the current.
- ICCP = 1mA/MHz of Peripheral Read/Write cycle time. (Example: 1.0 $\mu\text{s}$  I/O Read/Write cycle time = 1mA).
- Tested as  $V_{OH}$  at -2.5mA.

**Capacitance**  $T_A = 25^\circ\text{C}$

SYMBOL	PARAMETER	TYPICAL	UNITS	TEST CONDITIONS
CIN	Input Capacitance	10	pF	FREQ = 1MHz, All Measurements are referenced to device GND
CI/O	I/O Capacitance	20	pF	



## ADC0808/ADC0809

### 8-Bit $\mu$ P Compatible A/D Converters with 8-Channel Multiplexer

#### General Description

The ADC0808, ADC0809 data acquisition component is a monolithic CMOS device with an 8-bit analog-to-digital converter, 8-channel multiplexer and microprocessor compatible control logic. The 8-bit A/D converter uses successive approximation as the conversion technique. The converter features a high impedance chopper stabilized comparator, a 256R voltage divider with analog switch tree and a successive approximation register. The 8-channel multiplexer can directly access any of 8-single-ended analog signals.

The device eliminates the need for external zero and full-scale adjustments. Easy interfacing to microprocessors is provided by the latched and decoded multiplexer address inputs and latched TTL TRI-STATE outputs.

The design of the ADC0808, ADC0809 has been optimized by incorporating the most desirable aspects of several A/D conversion techniques. The ADC0808, ADC0809 offers high speed, high accuracy, minimal temperature dependence, excellent long-term accuracy and repeatability, and consumes minimal power. These features make this device ideally suited to applications from process and machine control to consumer and automotive applications. For 16-channel multiplexer with common output (sample/hold port) see ADC0816 data sheet. (See AN-247 for more information.)

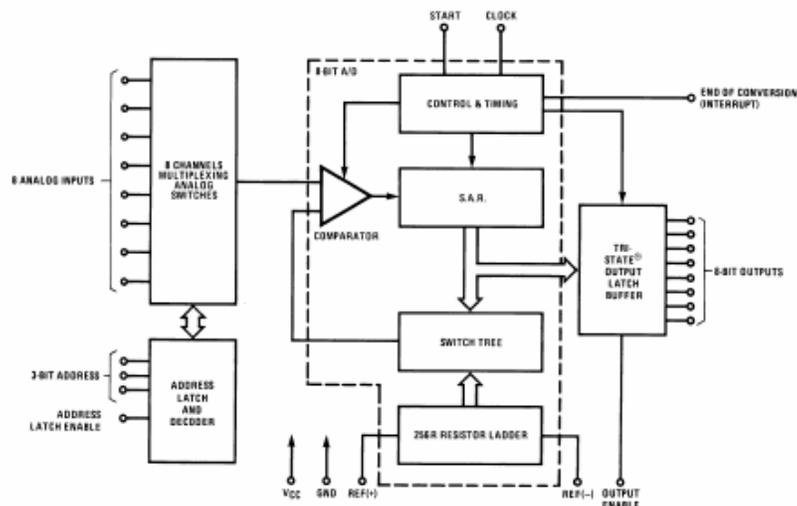
#### Features

- Easy interface to all microprocessors
- Operates ratiometrically or with 5 V<sub>DC</sub> or analog span adjusted voltage reference
- No zero or full-scale adjust required
- 8-channel multiplexer with address logic
- 0V to 5V input range with single 5V power supply
- Outputs meet TTL voltage level specifications
- ADC0808 equivalent to MM74C949
- ADC0809 equivalent to MM74C949-1

#### Key Specifications

- |                          |                                       |
|--------------------------|---------------------------------------|
| ■ Resolution             | 8 Bits                                |
| ■ Total Unadjusted Error | $\pm \frac{1}{2}$ LSB and $\pm 1$ LSB |
| ■ Single Supply          | 5 V <sub>DC</sub>                     |
| ■ Low Power              | 15 mW                                 |
| ■ Conversion Time        | 100 $\mu$ s                           |

#### Block Diagram

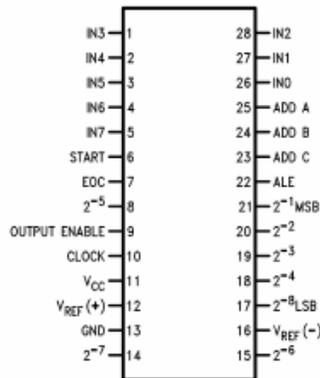


See Ordering  
Information

00667201

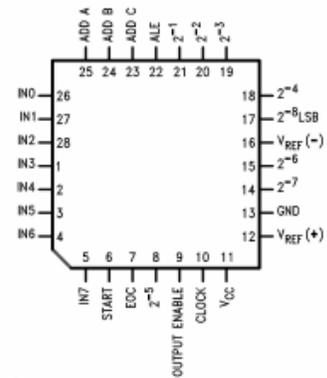
Connection Diagrams

Dual-In-Line Package



00567211  
Order Number ADC0808CCN or ADC0809CCN  
See NS Package J28A or N28A

Molded Chip Carrier Package



00567212  
Order Number ADC0808CCV or ADC0809CCV  
See NS Package V28A

Ordering Information

TEMPERATURE RANGE		-40°C to +85°C	
Error	±½ LSB Unadjusted	ADC0808CCN	ADC0808CCV
	±1 LSB Unadjusted	ADC0809CCN	ADC0809CCV
Package Outline		N28A Molded DIP	V28A Molded Chip Carrier

Absolute Maximum Ratings (Notes 2, 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage (V <sub>CC</sub> ) (Note 3)	6.5V
Voltage at Any Pin	-0.3V to (V <sub>CC</sub> +0.3V)
Except Control Inputs	
Voltage at Control Inputs (START, OE, CLOCK, ALE, ADD A, ADD B, ADD C)	-0.3V to +15V
Storage Temperature Range	-65°C to +150°C
Package Dissipation at T <sub>A</sub> =25°C	875 mW
Lead Temp. (Soldering, 10 seconds)	

Dual-In-Line Package (plastic)	260°C
Molded Chip Carrier Package	
Vapor Phase (60 seconds)	215°C
Infrared (15 seconds)	220°C
ESD Susceptibility (Note 8)	400V

Operating Conditions (Notes 1, 2)

Temperature Range (Note 1)	T <sub>MIN</sub> ≤ T <sub>A</sub> ≤ T <sub>MAX</sub>
ADC0808CCN, ADC0809CCN	-40°C ≤ T <sub>A</sub> ≤ +85°C
ADC0808CCV, ADC0809CCV	-40°C ≤ T <sub>A</sub> ≤ +85°C
Range of V <sub>CC</sub> (Note 1)	4.5 V <sub>DC</sub> to 6.0 V <sub>DC</sub>

Electrical Characteristics

Converter Specifications: V<sub>CC</sub>=5 V<sub>DC</sub>=V<sub>REF+</sub>, V<sub>REF(-)</sub>=GND, T<sub>MIN</sub> ≤ T<sub>A</sub> ≤ T<sub>MAX</sub> and f<sub>CLK</sub>=640 kHz unless otherwise stated.

Symbol	Parameter	Conditions	Min	Typ	Max	Units
	ADC0808 Total Unadjusted Error (Note 5)	25°C			±½	LSB
		T <sub>MIN</sub> to T <sub>MAX</sub>			±¾	LSB
	ADC0809 Total Unadjusted Error (Note 5)	0°C to 70°C			±1	LSB
		T <sub>MIN</sub> to T <sub>MAX</sub>			±1¼	LSB
	Input Resistance	From Ref(+) to Ref(-)	1.0	2.5		kΩ
	Analog Input Voltage Range	(Note 4) V(+) or V(-)	GND-0.10		V <sub>CC</sub> +0.10	V <sub>DC</sub>
V <sub>REF(+)</sub>	Voltage, Top of Ladder	Measured at Ref(+)		V <sub>CC</sub>	V <sub>CC</sub> +0.1	V
$\frac{V_{REF(+)} + V_{REF(-)}}{2}$	Voltage, Center of Ladder		V <sub>CC</sub> /2-0.1	V <sub>CC</sub> /2	V <sub>CC</sub> /2+0.1	V
V <sub>REF(-)</sub>	Voltage, Bottom of Ladder	Measured at Ref(-)	-0.1	0		V
I <sub>IN</sub>	Comparator Input Current	f <sub>c</sub> =640 kHz, (Note 6)	-2	±0.5	2	µA

## Functional Description

**Multiplexer.** The device contains an 8-channel single-ended analog signal multiplexer. A particular input channel is selected by using the address decoder. *Table 1* shows the input states for the address lines to select any channel. The address is latched into the decoder on the low-to-high transition of the address latch enable signal.

TABLE 1.

SELECTED ANALOG CHANNEL	ADDRESS LINE		
	C	B	A
IN0	L	L	L
IN1	L	L	H
IN2	L	H	L
IN3	L	H	H
IN4	H	L	L
IN5	H	L	H
IN6	H	H	L
IN7	H	H	H

## CONVERTER CHARACTERISTICS

### The Converter

The heart of this single chip data acquisition system is its 8-bit analog-to-digital converter. The converter is designed to give fast, accurate, and repeatable conversions over a wide range of temperatures. The converter is partitioned into 3 major sections: the 256R ladder network, the successive approximation register, and the comparator. The converter's digital outputs are positive true.

The 256R ladder network approach (*Figure 1*) was chosen over the conventional R/2R ladder because of its inherent monotonicity, which guarantees no missing digital codes. Monotonicity is particularly important in closed loop feedback control systems. A non-monotonic relationship can cause oscillations that will be catastrophic for the system. Additionally, the 256R network does not cause load variations on the reference voltage.

The bottom resistor and the top resistor of the ladder network in *Figure 1* are not the same value as the remainder of the network. The difference in these resistors causes the output characteristic to be symmetrical with the zero and full-scale points of the transfer curve. The first output transition occurs when the analog signal has reached  $+\frac{1}{2}$  LSB and succeeding output transitions occur every 1 LSB later up to full-scale.

The successive approximation register (SAR) performs 8 iterations to approximate the input voltage. For any SAR type converter, n-iterations are required for an n-bit converter. *Figure 2* shows a typical example of a 3-bit converter. In the ADC0808, ADC0809, the approximation technique is extended to 8 bits using the 256R network.

The A/D converter's successive approximation register (SAR) is reset on the positive edge of the start conversion pulse. The conversion is begun on the falling edge of the start conversion pulse. A conversion in process will be interrupted by receipt of a new start conversion pulse. Continuous conversion may be accomplished by tying the end-of-conversion (EOC) output to the SC input. If used in this mode, an external start conversion pulse should be applied after power up. End-of-conversion will go low between 0 and 8 clock pulses after the rising edge of start conversion.

The most important section of the A/D converter is the comparator. It is this section which is responsible for the ultimate accuracy of the entire converter. It is also the comparator drift which has the greatest influence on the repeatability of the device. A chopper-stabilized comparator provides the most effective method of satisfying all the converter requirements.

The chopper-stabilized comparator converts the DC input signal into an AC signal. This signal is then fed through a high gain AC amplifier and has the DC level restored. This technique limits the drift component of the amplifier since the drift is a DC component which is not passed by the AC amplifier. This makes the entire A/D converter extremely insensitive to temperature, long term drift and input offset errors.

*Figure 4* shows a typical error curve for the ADC0808 as measured using the procedures outlined in AN-179.

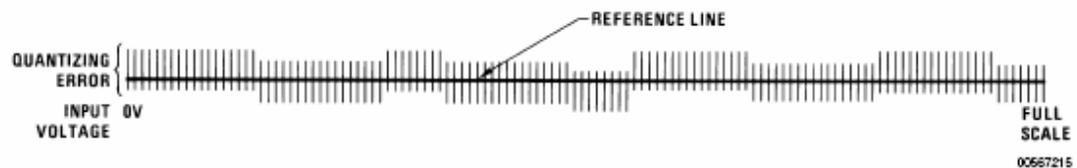


FIGURE 4. Typical Error Curve

## Applications Information

### OPERATION

#### 1.0 RATIOMETRIC CONVERSION

The ADC0808, ADC0809 is designed as a complete Data Acquisition System (DAS) for ratiometric conversion systems. In ratiometric systems, the physical variable being measured is expressed as a percentage of full-scale which is not necessarily related to an absolute standard. The voltage input to the ADC0808 is expressed by the equation

$$\frac{V_{IN}}{V_{FS} - V_Z} = \frac{D_X}{D_{MAX} - D_{MIN}} \quad (1)$$

$V_{IN}$ =Input voltage into the ADC0808

$V_{FS}$ =Full-scale voltage

$V_Z$ =Zero voltage

$D_X$ =Data point being measured

$D_{MAX}$ =Maximum data limit

$D_{MIN}$ =Minimum data limit

A good example of a ratiometric transducer is a potentiometer used as a position sensor. The position of the wiper is directly proportional to the output voltage which is a ratio of the full-scale voltage across it. Since the data is represented as a proportion of full-scale, reference requirements are greatly reduced, eliminating a large source of error and cost for many applications. A major advantage of the ADC0808, ADC0809 is that the input voltage range is equal to the supply range so the transducers can be connected directly across the supply and their outputs connected directly into the multiplexer inputs, (*Figure 9*).

Ratiometric transducers such as potentiometers, strain gauges, thermistor bridges, pressure transducers, etc., are suitable for measuring proportional relationships; however, many types of measurements must be referred to an absolute standard such as voltage or current. This means a system reference must be used which relates the full-scale voltage to the standard volt. For example, if  $V_{CC}=V_{REF}=5.12V$ , then the full-scale range is divided into 256 standard steps. The smallest standard step is 1 LSB which is then 20 mV.

## Applications Information (Continued)

### 2.0 RESISTOR LADDER LIMITATIONS

The voltages from the resistor ladder are compared to the selected into 8 times in a conversion. These voltages are coupled to the comparator via an analog switch tree which is referenced to the supply. The voltages at the top, center and bottom of the ladder must be controlled to maintain proper operation.

The top of the ladder, Ref(+), should not be more positive than the supply, and the bottom of the ladder, Ref(-), should

not be more negative than ground. The center of the ladder voltage must also be near the center of the supply because the analog switch tree changes from N-channel switches to P-channel switches. These limitations are automatically satisfied in ratiometric systems and can be easily met in ground referenced systems.

Figure 10 shows a ground referenced system with a separate supply and reference. In this system, the supply must be trimmed to match the reference voltage. For instance, if a 5.12V is used, the supply should be adjusted to the same voltage within 0.1V.

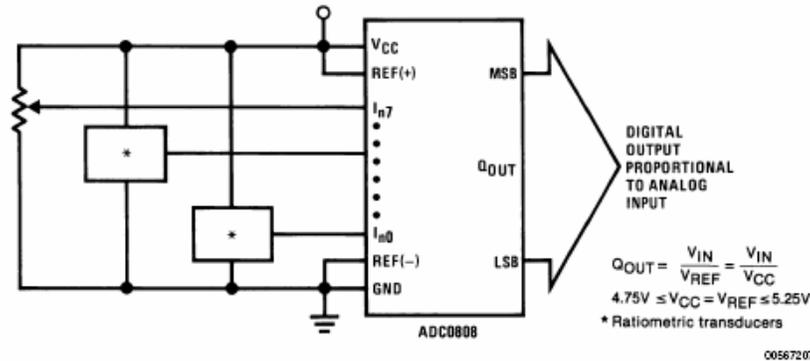
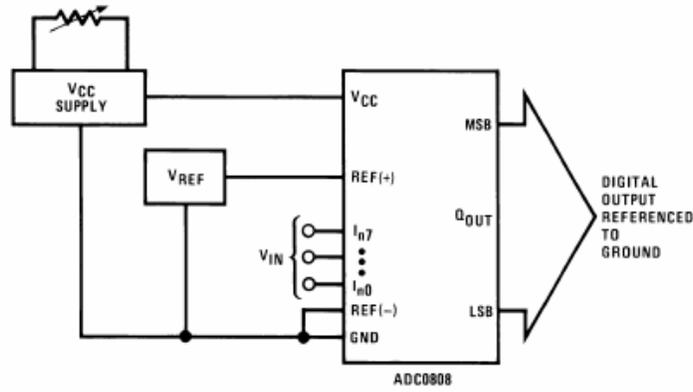


FIGURE 9. Ratiometric Conversion System

The ADC0808 needs less than a milliamp of supply current so developing the supply from the reference is readily accomplished. In Figure 11 a ground referenced system is shown which generates the supply from the reference. The buffer shown can be an op amp of sufficient drive to supply the milliamp of supply current and the desired bus drive, or if a capacitive bus is driven by the outputs a large capacitor will supply the transient supply current as seen in Figure 12. The LM301 is overcompensated to insure stability when loaded by the 10  $\mu$ F output capacitor.

The top and bottom ladder voltages cannot exceed  $V_{CC}$  and ground, respectively, but they can be symmetrically less than  $V_{CC}$  and greater than ground. The center of the ladder voltage should always be near the center of the supply. The sensitivity of the converter can be increased, (i.e., size of the LSB steps decreased) by using a symmetrical reference system. In Figure 13, a 2.5V reference is symmetrically centered about  $V_{CC}/2$  since the same current flows in identical resistors. This system with a 2.5V reference allows the LSB bit to be half the size of a 5V reference system.



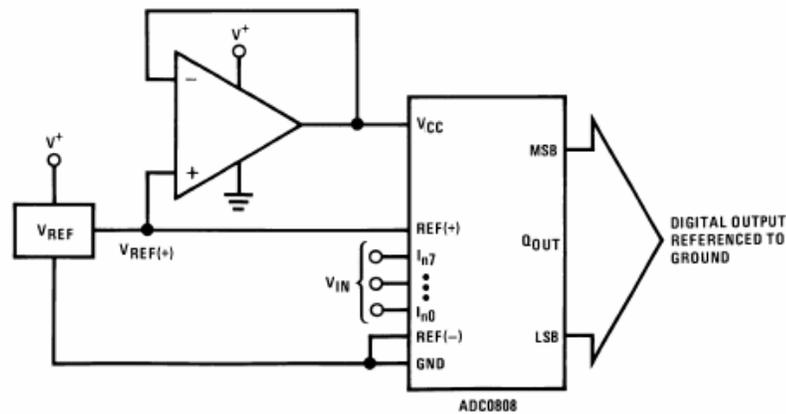


00567234

$$Q_{OUT} = \frac{V_{IN}}{V_{REF}}$$

$$4.75V \leq V_{CC} = V_{REF} \leq 5.25V$$

FIGURE 10. Ground Referenced Conversion System Using Trimmed Supply



00567235

$$Q_{OUT} = \frac{V_{IN}}{V_{REF}}$$

$$4.75V \leq V_{CC} = V_{REF} \leq 5.25V$$

FIGURE 11. Ground Referenced Conversion System with Reference Generating V<sub>CC</sub> Supply

### 3.0 CONVERTER EQUATIONS

The transition between adjacent codes N and N+1 is given by:

$$V_{IN} = \left( (V_{REF(+)} - V_{REF(-)}) \left[ \frac{N}{256} + \frac{1}{512} \right] \pm V_{TUE} \right) + V_{REF(-)} \quad (2)$$

The center of an output code N is given by:

$$V_{IN} \left( (V_{REF(+)} - V_{REF(-)}) \left[ \frac{N}{256} \right] \pm V_{TUE} \right) + V_{REF(-)} \quad (3)$$

The output code N for an arbitrary input are the integers within the range:

$$N = \frac{V_{IN} - V_{REF(-)}}{V_{REF(+)} - V_{REF(-)}} \times 256 \pm \text{Absolute Accuracy} \quad (4)$$

Where: V<sub>IN</sub>=Voltage at comparator input  
V<sub>REF(+)</sub>=Voltage at Ref(+)  
V<sub>REF(-)</sub>=Voltage at Ref(-)  
V<sub>TUE</sub>=Total unadjusted error voltage (typically V<sub>REF(+)</sub>±512)

**Applications Information** (Continued)

**4.0 ANALOG COMPARATOR INPUTS**

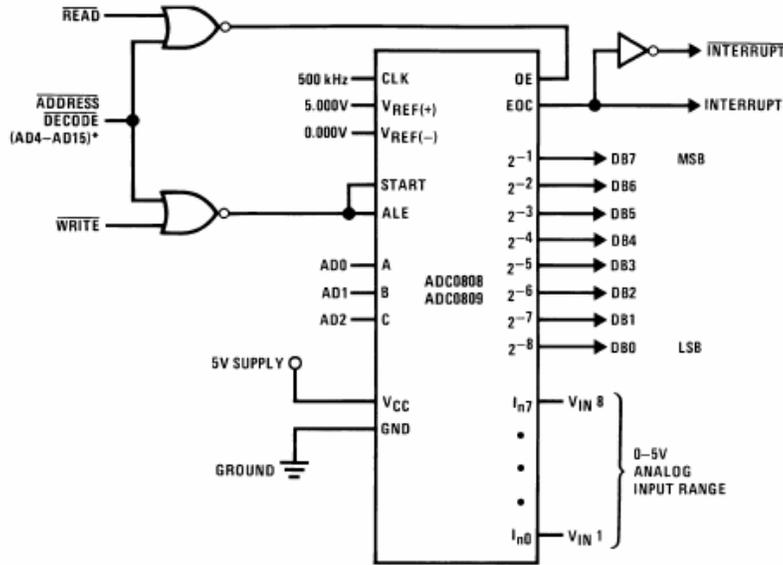
The dynamic comparator input current is caused by the periodic switching of on-chip stray capacitances. These are connected alternately to the output of the resistor ladder/switch tree network and to the comparator input as part of the operation of the chopper stabilized comparator.

The average value of the comparator input current varies directly with clock frequency and with  $V_{IN}$  as shown in Figure 6.

If no filter capacitors are used at the analog inputs and the signal source impedances are low, the comparator input current should not introduce converter errors, as the transient created by the capacitance discharge will die out before the comparator output is strobed.

If input filter capacitors are desired for noise reduction and signal conditioning they will tend to average out the dynamic comparator input current. It will then take on the characteristics of a DC bias current whose effect can be predicted conventionally.

**Typical Application**



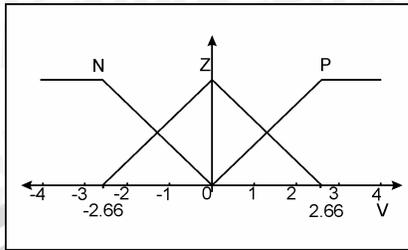
\*Address latches needed for 8085 and SC/MP interfacing the ADC0808 to a microprocessor

00567210

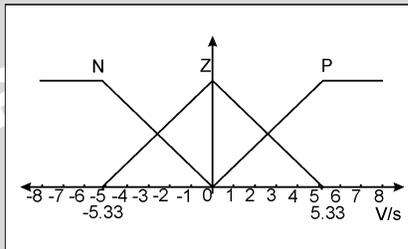
**TABLE 2. Microprocessor Interface Table**

PROCESSOR	READ	WRITE	INTERRUPT (COMMENT)
8080	MEMR	MEMW	INTR (Thru RST Circuit)
8085	RD	WR	INTR (Thru RST Circuit)
Z-80	RD	WR	INT (Thru RST Circuit, Mode 0)
SC/MP	NRDS	NWDS	SA (Thru Sense A)
6800	VMA $\cdot\bar{\phi}$ 2 $\cdot$ R/W	VMA $\cdot\bar{\phi}$ 1 $\cdot$ R/W	TRQA or TRQB (Thru PIA)

PERHITUNGAN KELUARAN FUZZY DARI ASUMSI BEBERAPA  
DERAJAT KEANGGOTAAN MASUKAN error DAN derror



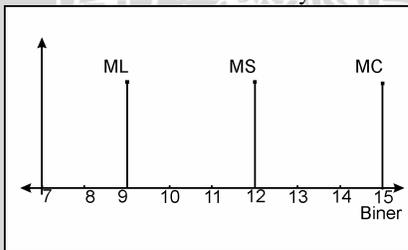
Fungsi Keanggotaan Masukan error



Fungsi Keanggotaan Masukan derror

	e	N	Z	P
$\Delta e$				
N		ML	ML	MS
Z		ML	MS	MC
P		MS	MC	MC

Basis Aturan Fuzzy



Fungsi Keanggotaan Keluaran

Diasumsikan 5 kondisi derajat keanggotaan error dan 5 kondisi derajat keanggotaan derror yang mewakili derajat keanggotaan dari masing masing label pada fungsi keanggotaan.

- Asumsi 1a error adalah  $N=1, Z=0$
- Asumsi 2a error adalah  $N=0,5, Z=0,5$
- Asumsi 3a error adalah  $N=0, Z=1, P=0$
- Asumsi 4a error adalah  $Z=0,5, P=0,5$
- Asumsi 5a error adalah  $Z=0, P=1$

- Asumsi 1b derror adalah  $N=1, Z=0$
- Asumsi 2b derror adalah  $N=0,5, Z=0,5$
- Asumsi 3b derror adalah  $N=0, Z=1, P=0$
- Asumsi 4b derror adalah  $Z=0,5, P=0,5$
- Asumsi 5a derror adalah  $Z=0, P=1$

Dilakukan perhitungan dari kemungkinan pasangan 5 asumsi error dan 5 asumsi derror

@ **Perhitungan 1** :

Asumsi 1a dengan asumsi 1b

Fungsi Keanggotaan error :

error N=1

Fungsi Keanggotaan derror :

derror N=1

Basis Aturan Fuzzy

If error N=1 and derror N=1, then ML=1

Defuzzifikasi

$$\text{Keluaran Singleton} = \frac{(1 \times 9)}{1}$$

$$\text{Crisp Keluaran Singleton} = 9$$

@ **Perhitungan 2** :

Asumsi 1a dengan asumsi 2b

Fungsi Keanggotaan error :

error N=1

Fungsi Keanggotaan derror :

derror N=0,5, Z=0,5

Basis Aturan Fuzzy

If error N=1 and derror N=0,5, then ML=0,5

If error N=1 and derror Z=0,5, then ML=0,5

Defuzzifikasi

$$\text{Keluaran Singleton} = \frac{(0,5 \times 9) + (0,5 \times 9)}{(0,5 + 0,5)}$$

$$\text{Keluaran Singleton} = \frac{9}{1}$$

$$\text{Crisp Keluaran Singleton} = 9$$

@ **Perhitungan 3** :

Asumsi 1a dengan asumsi 3b

Fungsi Keanggotaan error :

error N=1

Fungsi Keanggotaan derror :

derror N=0, Z=1, P=0

Basis Aturan Fuzzy

If error N=1 and derror Z=1, then ML=1

Defuzzifikasi

$$\text{Keluaran Singleton} = \frac{(1 \times 9)}{1}$$

$$\text{Crisp Keluaran Singleton} = 9$$

@ **Perhitungan 4** :

Asumsi 1a dengan asumsi 4b

Fungsi Keanggotaan error :

error N=1

Fungsi Keanggotaan derror :

derror Z=0,5, P=0,5

Basis Aturan Fuzzy

If error N=1 and derror Z=0,5, then ML=0,5

If error N=1 and derror P=0,5, then MS=0,5

Defuzzifikasi

Keluaran Singleton =

$$(0,5 \times 9) + (0,5 \times 12)$$

$$(0,5 + 0,5)$$

$$\text{Keluaran Singleton} = \frac{10,5}{1}$$

$$\text{Crisp Keluaran Singleton} = 10,5$$

@ **Perhitungan 5** :

Asumsi 1a dengan asumsi 5b

Fungsi Keanggotaan error :

error N=1

Fungsi Keanggotaan derror :

derror Z=0, P=1

Basis Aturan Fuzzy

If error N=1 and derror P=1, then MS=1

Defuzzifikasi

$$\text{Keluaran Singleton} = \frac{12}{1}$$

$$\text{Crisp Keluaran Singleton} = 12$$

@ **Perhitungan 6** :

Asumsi 2a dengan asumsi 1b

Fungsi Keanggotaan error :

error N=0,5, Z=0,5

Fungsi Keanggotaan derror :

derror N=1

Basis Aturan Fuzzy

If error N=0,5 and derror N=1, then ML=0,5

If error Z=0,5 and derror N=1, then MS=0,5

Defuzzifikasi

$$\text{Keluaran Singleton} =$$

$$(0,5 \times 9) + (0,5 \times 9)$$

$$(0,5 + 0,5)$$

$$\text{Keluaran Singleton} = \frac{9}{1}$$

$$\text{Crisp Keluaran Singleton} = 9$$

@ **Perhitungan 7** :

Asumsi 2a dengan asumsi 2b

Fungsi Keanggotaan error :

error N=0,5, Z=0,5

Fungsi Keanggotaan derror :

derror N=0,5, Z=0,5

Basis Aturan Fuzzy

If error N=0,5 and derror N=0,5, then ML=0,5

If error N=0,5 and derror Z=0,5, then ML=0,5

If error Z=0,5 and derror N=0,5, then ML=0,5

If error Z=0,5 and derror Z=0,5, then MS=0,5

Defuzzifikasi

$$\text{Keluaran Singleton} = \frac{(0,5 \times 9) + (0,5 \times 9) + (0,5 \times 9) + (0,5 \times 12)}{(0,5 + 0,5 + 0,5 + 0,5)}$$

$$(0,5 + 0,5 + 0,5 + 0,5)$$

$$\text{Keluaran Singleton} = \frac{19,5}{2}$$

$$\text{Crisp Keluaran Singleton} = 9,75$$

**@ Perhitungan 8 :**

Asumsi 2a dengan asumsi 3b

Fungsi Keanggotaan error :

error N=0,5, Z=0,5

Fungsi Keanggotaan derror :

derror N=0, Z=1, P=0

Basis Aturan Fuzzy

If error N=0,5 and derror Z=1, then ML=0,5

If error Z=0,5 and derror Z=1, then ML=0,5

Defuzzifikasi

$$\begin{aligned} \text{Keluaran Singleton} &= \\ \frac{(0,5 \times 9) + (0,5 \times 9)}{(0,5 + 0,5)} \end{aligned}$$

$$\text{Keluaran Singleton} = \frac{9}{1}$$

$$\text{Crisp Keluaran Singleton} = 9$$

**@ Perhitungan 9 :**

Asumsi 2a dengan asumsi 4b

Fungsi Keanggotaan error :

error N=0,5, Z=0,5

Fungsi Keanggotaan derror :

derror Z=0,5, P=0,5

Basis Aturan Fuzzy

If error N=0,5 and derror Z=0,5, then ML=0,5

If error N=0,5 and derror P=0,5, then ML=0,5

If error Z=0,5 and derror Z=0,5, then ML=0,5

If error Z=0,5 and derror P=0,5, then MS=0,5

Defuzzifikasi

$$\begin{aligned} \text{Keluaran Singleton} &= \\ \frac{(0,5 \times 9) + (0,5 \times 9) + (0,5 \times 9) + (0,5 \times 12)}{(0,5 + 0,5 + 0,5 + 0,5)} \end{aligned}$$

$$\text{Keluaran Singleton} = \frac{19,5}{2}$$

$$\text{Crisp Keluaran Singleton} = 9,75$$

**@ Perhitungan 10 :**

Asumsi 2a dengan asumsi 5b

Fungsi Keanggotaan error :

error N=0,5, Z=0,5

Fungsi Keanggotaan derror :

derror Z=0, P=1

Basis Aturan Fuzzy

If error N=0,5 and derror P=1, then MS=0,5

If error Z=0,5 and derror P=1, then MC=0,5

Defuzzifikasi

$$\begin{aligned} \text{Keluaran Singleton} &= \\ \frac{(0,5 \times 12) + (0,5 \times 15)}{(0,5 + 0,5)} \end{aligned}$$

$$\text{Keluaran Singleton} = \frac{13,5}{1}$$

$$\text{Crisp Crisp Keluaran} = 13,5$$

**@ Perhitungan 11 :**

Asumsi 3a dengan asumsi 1b

Fungsi Keanggotaan error :

error N=0, Z=1, P=0

Fungsi Keanggotaan derror :

derror N=1

Basis Aturan Fuzzy

If error Z=1 and derror N=1, then

ML=1

Defuzzifikasi

$$\text{Keluaran Singleton} = \frac{(1 \times 9)}{1}$$

$$\text{Keluaran Singleton} = \frac{9}{1}$$

$$\text{Crisp Keluaran Singleton} = 9$$

**@ Perhitungan 12 :**

Asumsi 3a dengan asumsi 2b

Fungsi Keanggotaan error :

error N=0, Z=1, P=0

Fungsi Keanggotaan derror :

derror N=0,5, Z=0,5

Basis Aturan Fuzzy

If error Z=1 and derror N=0,5, then ML=0,5

If error Z=1 and derror Z=0,5, then MS=0,5

Defuzzifikasi

$$\begin{aligned} \text{Keluaran Singleton} &= \\ \frac{(0,5 \times 9) + (0,5 \times 12)}{(0,5 + 0,5)} \end{aligned}$$

$$\text{Keluaran Singleton} = \frac{10,5}{1}$$

$$\text{Crisp Keluaran Singleton} = 10,5$$

**@ Perhitungan 13 :**

Asumsi 3a dengan asumsi 3b

Fungsi Keanggotaan error :

error N=0, Z=1, P=0

Fungsi Keanggotaan derror :

derror N=0, Z=1, P=0

Basis Aturan Fuzzy

If error Z=1 and derror Z=1, then MS=1

Defuzzifikasi

$$\text{Keluaran Singleton} = \frac{(1 \times 12)}{1}$$

$$\text{Crisp Keluaran Singleton} = 12$$

**@ Perhitungan 14 :**

Asumsi 3a dengan asumsi 4b

Fungsi Keanggotaan error :

error N=0, Z=1, P=0

Fungsi Keanggotaan derror :  
 derror Z=0,5, P=0,5  
 Basis Aturan Fuzzy  
 If error Z=1 and derror Z=0,5, then MS=0,5  
 If error Z=1 and derror P=0,5, then MC=0,5  
 Defuzzifikasi  
 Keluaran Singleton =  

$$\frac{(0,5 \times 12) + (0,5 \times 15)}{(0,5 + 0,5)}$$

Keluaran Singleton =  $\frac{13,5}{1}$   
 Crisp Keluaran Singleton = 13,5

**@ Perhitungan 15 :**

Asumsi 3a dengan asumsi 5b  
 Fungsi Keanggotaan error :  
 error N=0, Z=1, P=0  
 Fungsi Keanggotaan derror :  
 derror Z=0, P=1  
 Basis Aturan Fuzzy  
 If error Z=1 and derror P=1, then MC=1  
 Defuzzifikasi

Keluaran Singleton =  $\frac{(1 \times 15)}{1}$   
 Crisp Keluaran Singleton = 15

**@ Perhitungan 16 :**

Asumsi 4a dengan asumsi 1b  
 Fungsi Keanggotaan error :  
 error Z=0,5, P=0,5  
 Fungsi Keanggotaan derror :  
 derror N=1  
 Basis Aturan Fuzzy  
 If error Z=0,5 and derror N=1, then ML=0,5  
 If error P=0,5 and derror N=1, then MS=0,5  
 Defuzzifikasi  
 Keluaran Singleton =  

$$\frac{(0,5 \times 9) + (0,5 \times 12)}{(0,5 + 0,5)}$$

Keluaran Singleton =  $\frac{10,5}{1}$   
 Crisp Keluaran Singleton = 10,5

**@ Perhitungan 17 :**

Asumsi 4a dengan asumsi 2b  
 Fungsi Keanggotaan error :  
 error Z=0,5, P=0,5  
 Fungsi Keanggotaan derror :  
 derror N=0,5, Z=0,5  
 Basis Aturan Fuzzy  
 If error Z=0,5 and derror N=0,5, then  
 ML=0,5  
 If error Z=0,5 and derror Z=0,5, then  
 MS=0,5

If error P=0,5 and derror N=0,5, then  
 MS=0,5

If error P=0,5 and derror Z=0,5, then  
 MC=0,5

Defuzzifikasi  
 Keluaran Singleton =  

$$\frac{(0,5 \times 9) + (0,5 \times 12) + (0,5 \times 12) + (0,5 \times 15)}{(0,5 + 0,5 + 0,5 + 0,5)}$$

Keluaran Singleton =  $\frac{24}{2}$   
 Crisp Keluaran Singleton = 12

**@ Perhitungan 18 :**

Asumsi 4a dengan asumsi 3b  
 Fungsi Keanggotaan error :  
 error Z=0,5, P=0,5  
 Fungsi Keanggotaan derror :  
 derror N=0, Z=1, P=0  
 Basis Aturan Fuzzy  
 If error Z=0,5 and derror Z=1, then  
 MS=0,5  
 If error P=0,5 and derror Z=1, then  
 MC=0,5

Defuzzifikasi  
 Keluaran Singleton =  

$$\frac{(0,5 \times 12) + (0,5 \times 15)}{(0,5 + 0,5)}$$

Keluaran Singleton =  $\frac{13,5}{1}$   
 Crisp Keluaran Singleton = 9

**@ Perhitungan 19 :**

Asumsi 4a dengan asumsi 4b  
 Fungsi Keanggotaan error :  
 error Z=0,5, P=0,5  
 Fungsi Keanggotaan derror :  
 derror Z=0,5, P=0,5  
 Basis Aturan Fuzzy  
 If error Z=0,5 and derror Z=0,5, then MS=0,5  
 If error Z=0,5 and derror P=0,5, then MC=0,5  
 If error P=0,5 and derror Z=0,5, then MC=0,5  
 If error P=0,5 and derror P=0,5, then MC=0,5  
 Defuzzifikasi

Keluaran Singleton =  

$$\frac{(0,5 \times 12) + (0,5 \times 15) + (0,5 \times 15) + (0,5 \times 15)}{(0,5 + 0,5 + 0,5 + 0,5)}$$

Keluaran Singleton =  $\frac{28,5}{2}$   
 Crisp Keluaran Singleton = 14,25

**@ Perhitungan 20 :**

Asumsi 4a dengan asumsi 5b

Fungsi Keanggotaan error :  
 error  $Z=0,5$ ,  $P=0,5$   
 Fungsi Keanggotaan derror :  
 derror  $Z=0$ ,  $P=1$   
 Basis Aturan Fuzzy  
 If error  $Z=0,5$  and derror  $P=1$ , then  $MC=0,5$   
 If error  $P=0,5$  and derror  $P=1$ , then  $MC=0,5$   
 Defuzzifikasi  
 Keluaran Singleton =  

$$\frac{(0,5 \times 15) + (0,5 \times 15)}{(0,5 + 0,5)}$$

$$\text{Keluaran Singleton} = \frac{15}{1}$$

$$\text{Crisp Keluaran Singleton} = 15$$

**@ Perhitungan 21 :**

Asumsi 5a dengan asumsi 1b  
 Fungsi Keanggotaan error :  
 error  $Z=0$ ,  $P=1$   
 Fungsi Keanggotaan derror :  
 derror  $N=1$   
 Basis Aturan Fuzzy  
 If error  $P=1$  and derror  $N=1$ , then  $MS=1$   
 Defuzzifikasi

$$\text{Keluaran Singleton} = \frac{(1 \times 12)}{1}$$

$$\text{Keluaran Singleton} = \frac{12}{1}$$

$$\text{Crisp Keluaran Singleton} = 12$$

**@ Perhitungan 22 :**

Asumsi 5a dengan asumsi 2b  
 Fungsi Keanggotaan error :  
 error  $Z=0$ ,  $P=1$   
 Fungsi Keanggotaan derror :  
 derror  $N=0,5$ ,  $Z=0,5$   
 Basis Aturan Fuzzy  
 If error  $P=1$  and derror  $N=0,5$ , then  $MS=0,5$   
 If error  $P=1$  and derror  $Z=0,5$ , then  $MC=0,5$   
 Defuzzifikasi

$$\text{Keluaran Singleton} = \frac{(0,5 \times 12) + (0,5 \times 15)}{(0,5 + 0,5)}$$

$$\text{Keluaran Singleton} = \frac{13,5}{1}$$

$$\text{Crisp Keluaran Singleton} = 13,5$$

**@ Perhitungan 23 :**

Asumsi 5a dengan asumsi 3b  
 Fungsi Keanggotaan error :  
 error  $Z=0$ ,  $P=1$   
 Fungsi Keanggotaan derror :  
 derror  $N=0$ ,  $Z=1$ ,  $P=0$   
 Basis Aturan Fuzzy  
 If error  $P=1$  and derror  $Z=1$ , then  $MC=1$   
 Defuzzifikasi

$$\text{Keluaran Singleton} = \frac{(1 \times 15)}{1}$$

$$\text{Keluaran Singleton} = \frac{15}{1}$$

$$\text{Crisp Keluaran Singleton} = 15$$

**@ Perhitungan 24 :**

Asumsi 5a dengan asumsi 4b  
 Fungsi Keanggotaan error :  
 error  $Z=0$ ,  $P=1$   
 Fungsi Keanggotaan derror :  
 derror  $Z=0,5$ ,  $P=0,5$   
 Basis Aturan Fuzzy  
 If error  $P=1$  and derror  $Z=0,5$ , then  $MC=0,5$   
 If error  $P=1$  and derror  $P=0,5$ , then  $MC=0,5$   
 Defuzzifikasi

$$\text{Keluaran Singleton} = \frac{(0,5 \times 15) + (0,5 \times 15)}{(0,5 + 0,5)}$$

$$\text{Keluaran Singleton} = \frac{15}{1}$$

$$\text{Crisp Keluaran Singleton} = 15$$

**@ Perhitungan 25 :**

Asumsi 5a dengan asumsi 5b  
 Fungsi Keanggotaan error :  
 error  $Z=0$ ,  $P=1$   
 Fungsi Keanggotaan derror :  
 derror  $Z=0$ ,  $P=1$   
 Basis Aturan Fuzzy  
 If error  $P=1$  and derror  $P=1$ , then  $MC=1$   
 Defuzzifikasi

$$\text{Keluaran Singleton} = \frac{(1 \times 15)}{1}$$

$$\text{Keluaran Singleton} = \frac{15}{1}$$

$$\text{Crisp Keluaran Singleton} = 15$$

```

=====
;PENGENDALIAN KECEPATAN MOTOR DC PADA ALAT SABLON SATU
WARNA OTOMATIS
;DENGAN KONTROL LOGIKA FUZZY MENGGUNAKAN
MIKROKONTROLER AT89S52
=====

```

```

;KIKI ANDRIANTO
;0310632024-63
=====

```

```

;JURUSAN TEKNIK ELEKTRO
;FAKULTAS TEKNIK
;UNIVERSITAS BRAWIJAYA MALANG
=====

```

```

;LISTING PROGRAM
=====

```

```

;DATA-DATA
=====

```

```

ORG          00H

```

```

;INISIALISASI
=====

```

```

;===PPI===

```

```

CW PPI      EQU 2003H
PA PPI      EQU 2000H
PB PPI      EQU 2001H
PC PPI      EQU 2002H
PROG PPI    EQU 90H

```

```

;===ADC 0809===

```

```

ALMADC      EQU 4000H
IN0ADC      EQU 00H
IN1ADC      EQU 38H
DATADC0     EQU 30H
DATADC1     EQU 31H

```

```

;===KONTROL FUZZY===

```

```

SETP1       EQU 0AAH
SETP2       EQU 87H
FUZZY1      EQU 32H
FUZZY2      EQU 33H
FUZZY3      EQU 34H
ERROR1      EQU 35H
ERROR2      EQU 36H
DERROR1     EQU 37H
DERROR2     EQU 38H
KELUARAN    EQU 39H
MFER_N      EQU 40H
MFER_Z      EQU 41H
MFER_P      EQU 42H
MFDE_N      EQU 43H

```

```
MFDE_Z EQU 44H
MFDE_P EQU 45H
SETP EQU 46H
SETADC EQU 47H
HIT5MS EQU 48H
DASEN EQU 21H
```

```
=====
```

```
====INISIALISASI PPI====
```

```
MULAI:
```

```
MOV DPTR,#CWPP1
```

```
MOV A,#PROGPPI
```

```
MOVX @DPTR,A
```

```
AJMP JLN_AWL
```

```
====SR BACA SENSOR====
```

```
BCSEN:
```

```
MOV DPTR,#PAPPI
```

```
MOVX A,@DPTR
```

```
MOV DASEN,A
```

```
RET
```

```
====MOTOR ATAS-BAWAH JALAN BAWAH AKTIF====
```

```
MAB_JL_BW:
```

```
MOV DPTR,#PBPP1
```

```
MOV A,#05H
```

```
MOVX @DPTR,A
```

```
RET
```

```
====MOTOR ATAS BAWAH JALAN ATAS AKTIF====
```

```
MAB_JL_AT:
```

```
MOV DPTR,#PBPP1
```

```
MOV A,#09H
```

```
MOVX @DPTR,A
```

```
RET
```

```
====MOTOR ATAS BAWAH STOP====
```

```
MAB_ST:
```

```
MOV DPTR,#PBPP1
```

```
MOV A,#0CH
```

```
MOVX @DPTR,A
```

```
RET
```

```
====MOTOR KANAN-KIRI JALAN KANAN AKTIF====
```

```
MKK_JL_KN:
```

```
MOV DPTR,#PBPP1
```

```
MOV A,#0AH
```

```
MOVX @DPTR,A
```

```
RET
```

```
====MOTOR KANAN-KIRI JALAN KIRI AKTIF====
```

```
MKK_JL_KR:
```

```
MOV DPTR,#PBPP1
```

```
MOV A,#06H
```

```
MOVX @DPTR,A
```

```

RET
;===MOTOR KANAN-KIRI STOP===
MKK_ST:
MOV DPTR,#PBPPI
MOV A,#0CH
MOVX @DPTR,A
RET
;===MOTOR STEPPER JALAN===
MST:
MOV DPTR,#PBPPI
MOVX @DPTR,A
RET
;===CEK POSISI MOTOR ATAS-BAWAH===
JLN_AWL:
ACALL BCSN
;JBC A.0,MTRAB_BW
JBC DASEN.0,MTRAB_BW
AJMP MTRAB_AT1
;===MOTOR ATAS BAWAH GERAK ATAS 1===
MTRAB_AT1:
ACALL FUZZ1
MOV DPTR,#PCPPI
MOV A,KELUARAN
MOVX @DPTR,A
ACALL MAB_JL_AT
ACALL BCSN
JNB DASEN.0,MTRAB_AT1
AJMP JLN_AWL
;===MOTOR ATAS BAWAH GERAK BAWAH===
MTRAB_BW:
ACALL MKK_ST
ACALL FUZZ1
MOV DPTR,#PCPPI
MOV A,KELUARAN
MOVX @DPTR,A
ACALL MAB_JL_BW
ACALL BCSN
JB DASEN.6,MTRKK_KN
AJMP MTRAB_BW
;===MOTOR KANAN-KIRI GERAK KANAN CEPAT===
MTRKK_KN:
ACALL MAB_ST
ACALL BCSN
JBC DASEN.3,MTRKK_KR
ACALL BCSN
JBC DASEN.4,MTRKK_KN2
ACALL FUZZ2
MOV DPTR,#PCPPI

```

```

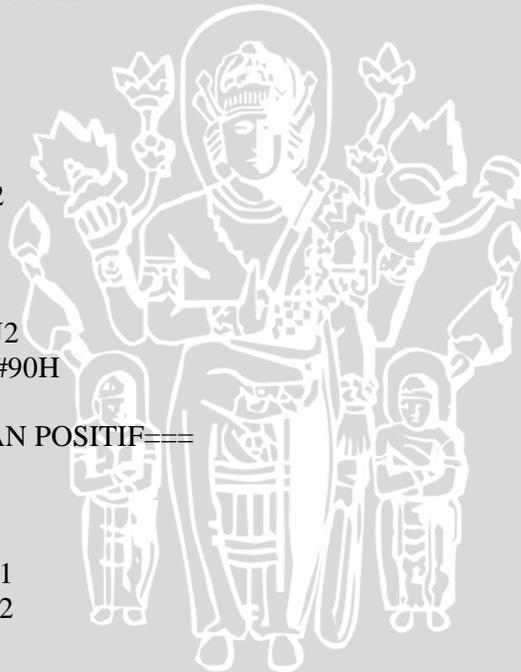
MOV A,KELUARAN
MOVX @DPTR,A
ACALL MKK_JL_KN
AJMP MTRKK_KN
;===MOTOR KANAN-KIRI GERAK KANAN LAMBAT===
MTRKK_KN2:
ACALL FUZZ3
MOV DPTR,#PCPPI
MOV A,KELUARAN
MOVX @DPTR,A
ACALL MKK_JL_KN
ACALL BCSEN
JNB DASEN.5,MTRKK_KN2
ACALL BCSEN
JBC DASEN.5,STPR_JL_KN
AJMP MTRKK_KN
;===MOTOR KANAN-KIRI GERAK KANAN LAMBAT+STEPPER===
STPR_JL_KN:
ACALL STPR_JL1
ACALL BCSEN
JNB DASEN.5,STPR_JL_KR
AJMP MTRKK_KN2
STPR_JL_KR:
ACALL STPR_JL2
AJMP MTRKK_KN
;===MOTOR KANAN-KIRI GERAK KIRI===
MTRKK_KR:
ACALL FUZZ2
MOV DPTR,#PCPPI
MOV A,KELUARAN
ACALL MKK_JL_KR
ACALL BCSEN
JBC DASEN.2,MTRAB_BW
AJMP MTRKK_KR
;===HITUNG KELUARAN FUZZY 1===
FUZZ1:
MOV SETADC,#IN0ADC
MOV SETP,#SETP1
ACALL ERROR1A
ACALL KONTROL1
RET
KONTROL1:
ACALL CEKMF1
RET
;===CEK FUNGSI KEANGGOTAAN ERROR===
CEKMF1:
ACALL CEKMF1A
ACALL CEKMF1B

```

```

RET
;===KEANGGOTAAN NEGATIF===
CEKMFEN:
MOV A,MFER_N
JZ CEKMFEP
ACALL CEKDRJEN1
ACALL CEKDRJEN2
CEKDRJEN1:
PUSH B
MOV B,#55H
CARIDRJ1:
DJNZ B,BANDING1
RET
BANDING1:
MOV A,MFER_N
CJNE A,B,CARIDRJ1
MOV KELUARAN,#0C0H
RET
CEKDRJEN2:
PUSH B
MOV B,#0AAH
CARIDRJ2:
DJNZ B,BANDING2
RET
BANDING2:
MOV A,MFER_N
CJNE A,B,CARIDRJ2
MOV KELUARAN,#90H
RET
;===KEANGGOTAAN POSITIF===
CEKMFEP:
MOV A,MFER_P
JZ CEKMFEN
ACALL CEKDRJEP1
ACALL CEKDRJEP2
CEKDRJEP1:
PUSH B
MOV B,#55H
CARIDRJP1:
DJNZ B,BANDINGP1
RET
BANDINGP1:
MOV A,MFER_P
CJNE A,B,CARIDRJP1
MOV KELUARAN,#0C0H
RET
CEKDRJEP2:
PUSH B

```



```

MOV B,#0AAH
CARIDRJP2:
DJNZ B,BANDINGP2
RET
BANDINGP2:
MOV A,MFER_P
CJNE A,B,CARIDRJP2
MOV KELUARAN,#0F0H
RET
;===HITUNG KELUARAN FUZZY 2===
FUZZ2:
MOV SETP,#SETP1
ACALL ERROR1A
ACALL KONTROL2
RET
KONTROL2:
ACALL CEKMFE1A
RET
;===CEK FUNGSI KEANGGOTAAN ERROR===
CEKMFE1A:
ACALL CEKMFEN1A
ACALL CEKMFEP1A
RET
;===KEANGGOTAAN NEGATIF===
CEKMFEN1A:
MOV A,MFER_N
JZ CEKMFEP1A
ACALL CEKDRJEN1A
ACALL CEKDRJEN1B
CEKDRJEN1A:
PUSH B
MOV B,#43H
CARIDRJ1A:
DJNZ B,BANDING1A
RET
BANDING1A:
MOV A,MFER_N
CJNE A,B,CARIDRJ1A
MOV KELUARAN,#0AH
RET
CEKDRJEN1B:
PUSH B
MOV B,#87H
CARIDRJ2B:
DJNZ B,BANDING2B
RET
BANDING2B:
MOV A,MFER_N

```

```

CJNE A,B,CARIDRJ2B
MOV KELUARAN,#07H
RET
;===KEANGGOTAAN POSITIF===
CEKMFEP1A:
MOV A,MFER_P
JZ CEKMFEN1A
ACALL CEKDRJEP1A
ACALL CEKDRJEP1B
CEKDRJEP1A:
PUSH B
MOV B,#43H
CARIDRJP1A:
DJNZ B,BANDINGP1A
RET
BANDINGP1A:
MOV A,MFER_P
CJNE A,B,CARIDRJP1A
MOV KELUARAN,#0AH
RET
CEKDRJEP1B:
PUSH B
MOV B,#87H
CARIDRJP1B:
DJNZ B,BANDINGP1B
RET
BANDINGP1B:
MOV A,MFER_P
CJNE A,B,CARIDRJP1B
MOV KELUARAN,#0DH
RET
;===HITUNG KELUARAN FUZZY 3===
FUZZ3:
MOV SETP,#SETP2
MOV SETADC,#IN1ADC
ACALL ERROR1A
ACALL KONTROL3
RET
KONTROL3:
ACALL CEKMFEA
RET
;===CEK FUNGSI KEANGGOTAAN ERROR===
CEKMFEA:
ACALL CEKMFENA
ACALL CEKMFEP1A
RET
;===KEANGGOTAAN NEGATIF===
CEKMFENA:

```

```
MOV A,MFER_N
JZ CEKMFEP A
ACALL CEKDRJENA
ACALL CEKDRJENB
CEKDRJENA:
  PUSH B
  MOV B,#43H
CARIDRJA:
  DJNZ B,BANDINGA
  RET
BANDINGA:
  MOV A,MFER_N
  CJNE A,B,CARIDRJA
  MOV KELUARAN,#0AH
  RET
CEKDRJENB:
  PUSH B
  MOV B,#87H
CARIDRJB:
  DJNZ B,BANDINGB
  RET
BANDINGB:
  MOV A,MFER_N
  CJNE A,B,CARIDRJB
  MOV KELUARAN,#07H
  RET
;===KEANGGOTAAN POSITIF===
CEKMFEP A:
  MOV A,MFER_P
  JZ CEKMFENA
  ACALL CEKDRJEP A
  ACALL CEKDRJEP B
CEKDRJEP A:
  PUSH B
  MOV B,#43H
CARIDRJPA:
  DJNZ B,BANDINGPA
  RET
BANDINGPA:
  MOV A,MFER_P
  CJNE A,B,CARIDRJPA
  MOV KELUARAN,#0AH
  RET
CEKDRJEP B:
  PUSH B
  MOV B,#87H
CARIDRJPB:
  DJNZ B,BANDINGPB
```



```

RET
BANDINGPB:
MOV A,MFER_P
CJNE A,B,CARIDRJPB
MOV KELUARAN,#0DH
RET

;===ERROR1===
ERROR1A:
ACALL BACAADC
MOV A,DATADC0
SUBB A,#SETP
MOV ERROR1,A
ERROR1B:
ACALL BACAADC
MOV A,DATADC0
SUBB A,#SETP
MOV ERROR2,A
JC MFEN
AJMP MFEP
DERROR:
MOV A,ERROR2
SUBB A,ERROR1
;MOV DERROR1,A
;JC DERROR1,MFDEN
JC MFDEN
AJMP MFDEP
MFEN:
MOV MFER_N,ERROR2
AJMP DERROR
MFEP:
MOV MFER_P,ERROR2
AJMP DERROR
MFDEN:
MOV MFDE_N,DERROR1
RET
MFDEP:
MOV MFDE_P,DERROR1
RET
DMFP:
MOV MFDE_P,DERROR1
MOV B,#85H
DIV AB
MOV MFDE_Z,A
RET
;===BACA DATA ADC===
BACAADC:
MOV DPTR,#ALMADC

```



```

MOV A,SETADC
MOVX @DPTR,A
;===TUNGGU EOC===
TUNGEOC:
JB INT0,TUNGEOC
ACALL TUNGGUADC
MOV A,#00H
MOV DATADC0,P1
RET
;===TUNGGU ADC===
TUNGGUADC:
PUSH B
MOV B,#0FFH
LOOPTNGADC:
DJNZ B,LOOPTNGADC
POP B
RET
;===TIMER MODE 0,5 DETIK===
HIT500MS:
MOV HIT5MS,#100
TNG500MS:
ACALL TNG5MS
DJNZ HIT5MS,TNG500MS
RET
TNG5MS:
PUSH TMOD
MOV TMOD,#12H
MOV TH0,#0EDH
MOV TL0,#0FFH
SETB TR0
NTNG5MS:
JBC TF0,SDH5MS
AJMP TNG5MS
SDH5MS:
CLR TR0
POP TMOD
RET
;===GERAK MOTOR STEPPER===
STPR_JL1:
MOV A,00000111B
ACALL MST
ACALL TNG5MS
MOV A,00001011B
ACALL MST
ACALL TNG5MS
MOV A,00001101B
ACALL MST
ACALL TNG5MS

```



MOV A,00001110B  
ACALL MST  
RET  
STPR\_JL2:  
MOV A,00001110B  
ACALL MST  
ACALL TNG5MS  
MOV A,00001101B  
ACALL MST  
ACALL TNG5MS  
MOV A,00001011B  
ACALL MST  
ACALL TNG5MS  
MOV A,00000111B  
ACALL MST  
RET  
END

