

PERANCANGAN DAN PEMBUATAN SISTEM PENGENDALIAN POSISI BENDA TERHADAP ARAH GRAVITASI BUMI DENGAN FUZZY LOGIC CONTROLLER NLX 220

TUGAS AKHIR

Disusun oleh :

KUSNADI SUGIARTO
NRP. 2296.100.532

PERPUSTAKAAN ITS	
Tgl. Terima	27-7-2000
No. Jari	H
Agenda Prp.	21.1324

RSE
629.89
Sug
p-1
1999



JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
1999



**PERANCANGAN DAN PEMBUATAN SISTEM
PENGENDALIAN POSISI BENDA TERHADAP
ARAH GRAVITASI BUMI DENGAN
FUZZY LOGIC CONTROLLER NLX 220**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik Elektro
Pada**

**Bidang Studi Elektronika
Jurusan Teknik Elektro
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
S u r a b a y a**

**Mengetahui / Menyetujui
Dosen Pembimbing**



**PUJIONO, ST
NIP. 132 094 793**

S U R A B A Y A

Agustus, 1999

ABSTRAK

Suatu benda apabila ditempatkan di atas suatu tempat dimana tempat tersebut pada suatu posisi tertentu maka benda tersebut akan miring mengikuti arah posisi tempat benda tersebut. Dalam hal ini tidaklah menguntungkan benda tersebut karena dapat mengakibatkan benda jatuh dari tempatnya.

Pada tugas akhir ini akan dirancang suatu sistem pengendalian posisi benda terhadap arah gravitasi bumi, dimana posisi benda akan selalu tegak lurus terhadap gravitasi bumi walaupun posisi tempat tersebut berubah-ubah kemiringannya.

Fuzzy logic kontroller diterapkan dalam rancangan ini untuk mengendalikan posisi mekanik. Jadi dasarnya gerakan mekanis di deteksi oleh bandul yang kemudian diklasifikasikan oleh kontroller fuzzy logic. Dalam klasifikasinya setelah berapa derajat kemiringan bidang dianggap telah miring. Kemudian mekanis akan bergerak melawan gerak kemiringan bidang agar posisi bidang tetap tegak lurus dengan gravitasi. Gerakan mekanis terdiri dari 2 bagian yaitu gerakan mengitari sumbu X dan gerakan mengitari sumbu Y. Untuk gerakan mengitari sumbu X terbagi dalam 3 kecepatan, sedangkan untuk gerakan mengitari sumbu Y hanya 1 kecepatan karena harus menahan beban mekanis yang mengitari sumbu X.

KATA PENGANTAR

Ucapan rasa syukur senantiasa penyusun panjatkan kepada Allah swt, karena atas limpahan rahmat dan hidayah-Nya sehingga dapat menyelesaikan tugas akhir yang berjudul :

“PERANCANGAN DAN PEMBUATAN SISTEM PENGENDALIAN POSISI BENDA TERHADAP ARAH GRAVITASI BUMI DENGAN FUZZY LOGIC CONTROLLER NLX 220”

Tugas akhir ini dibuat untuk memenuhi satuan kredit semester sebagai persyaratan dalam menyelesaikan pendidikan Strata 1 di Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya.

Penyusun sangat berterima kasih kepada pihak-pihak yang banyak membantu dalam penyelesaian tugas akhir ini, terutama kepada :

- Ibu, Bapak serta saudaraku yang senantiasa memberikan dukungan moril dan material serta mendoakan hingga selesainya tugas akhir ini.
- Pujiono,ST, selaku dosen pembimbing yang telah banyak memberikan masukan.
- Ir. Teguh Yuwono, selaku Ketua Jurusan Teknik Elektro.
- Ir. Soetikno, selaku Koordinator Bidang Studi Elektronika dan dosen pembimbing yang telah banyak membantu memberikan sarana dan prasarana, masukan serta bimbingan.
- Paidi dan Tono yang selalu bersama dalam penderitaan dan kebahagiaan , serta Joki yang telah meminjamkan PIC_nya.
- Dwi, Ari, Amroe yang telah membantu menyelesaikan ADC_nya.
- Rekan-rekan seperjuangan Angkatan 94 dan LJ-96, atas bantuannya selama masa-masa sulit perkuliahan.

Masih banyak kekurangan di dalam tugas akhir ini yang membutuhkan pengembangan dan perbaikan, untuk itu penyusun menerima dengan terbuka segala kritik dan masukan demi kesempurnaannya.

Akhir kata semoga Tugas Akhir ini dapat bermanfaat dan menjadi sarana belajar serta informasi tambahan khususnya di Bidang Studi Elektronika ITS maupun bagi pemerhati dunia elektronika pada umumnya.

Surabaya, Juli 1999

Penyusun

DAFTAR ISI

	Halaman
LEMBAR PERSETUJUAN	ii
ABSTRAK	iii
KATA PENGANTAR	iv
DAFTAR ISI	vi
DAFTAR GAMBAR	x
DAFTAR TABEL	xii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Permasalahan	2
1.3 Tujuan	2
1.4 Pembatasan Masalah	3
1.5 Metodologi	3
1.6 Sistematika Penulisan	4
1.7 Relevansi	5
BAB II TEORI PENUNJANG	6
2.1 Pendahuluan	6
2.2 Pergerakan Benda	6
2.3 Logika Fuzzy	10
2.3.1 Prinsip Logika Fuzzy	10

2.3.2 Teori Logika Fuzzy	11
2.3.3 Proses Logika Fuzzy	14
2.3.4 Pengontrolan Dengan Logika Fuzzy	17
2.4 Fuzzy Microcontroller NLX 220	19
2.4.1 Deskripsi Umum	19
2.4.2 Features	21
2.4.3 Aplikasi	22
2.4.4 Fungsi Pin-Pin Dari NLX 220	22
2.4.5 Arsitektur Perangkat	25
2.4.5.1 Pengembangan Sistem Logika Fuzzy	26
2.4.5.2 Fungsi Keanggotaan	27
2.4.5.3 Variabel Fuzzy	30
2.4.5.4 Rule	31
2.4.5.5 Evaluasi Rule	32
2.4.5.6 Floating Membership Function	32
2.4.6 Operasional Perangkat	35
2.4.6.1 Fuzzifier	36
2.4.6.2 Pembaharuan Data Output Yang Dilatch	37
2.4.6.3 Defuzzier	37
2.4.7 Organisasi Memory	39
2.4.7.1 Rule Dan Penyimpan Variabel Fuzzy	39
2.4.8 Timming (Pewaktuan)	43
2.4.8.1 Pewaktuan Operasi	43
2.4.8.2 Pewaktuan Internal	44

2.4.8.3 Waktu Tunda Loopback Internal	44
2.4.8.4 Pewaktuan Output	45
2.4.8.5 Operasi Prescale	45
2.4.8.6 Mode Inaktif	46
2.5 Sensor	46
2.6 Op Amplifier Sebagai Positif Level Detektor	47
2.7 Motor Arus Searah (DC)	48
2.8 LPT1 IBM PC	50
2.8.1 Konfigurasi Pin	51
2.8.2 Sistem Memori	52
2.9 ADC MAX 555	54
BAB III PERANCANGAN ALAT	56
3.1 Perancangan Sistem	56
3.2 Perancangan Mekanik	58
3.3 Perancangan Perangkat Keras	58
3.3.1 Rangkaian Sensor Posisi	60
3.3.2 Rangkaian Positif Level Detektor	61
3.3.3 Rangkaian Pengatur Kecepatan Motor DC	62
3.3.4 ADC MAX 555	63
3.3.4.1 Prinsip Kerja	64
3.3.4.2 Referensi External	65
3.3.4.3 Bypassing	65
3.3.4.4 Amplifier Track/Hold	66

3.3.5 Interface Ke LPT1 IBM PC	66
3.4 Perancangan Perangkat Lunak	67
3.4.1 Input Dan Output	68
3.4.2 Fuzzy Variabel	69
3.4.3 Aturan	71
3.5 Perangkat Lunak Akusisi Data	73
BAB IV PENGUJIAN DAN PENGUKURAN	75
4.1 Pengujian Sensor Posisi	75
4.2 Pengujian Positif Level Detektor	76
4.3 Pengujian Rangkaian Kecepatan Motor DC	77
4.4 Pengujian Modul NLX220	79
4.5 Pengujian Sistem	80
BAB V PENUTUP	
5.1 Kesimpulan	81
5.2 Saran	82
DAFTAR PUSTAKA	83
LAMPIRAN	

DAFTAR GAMBAR

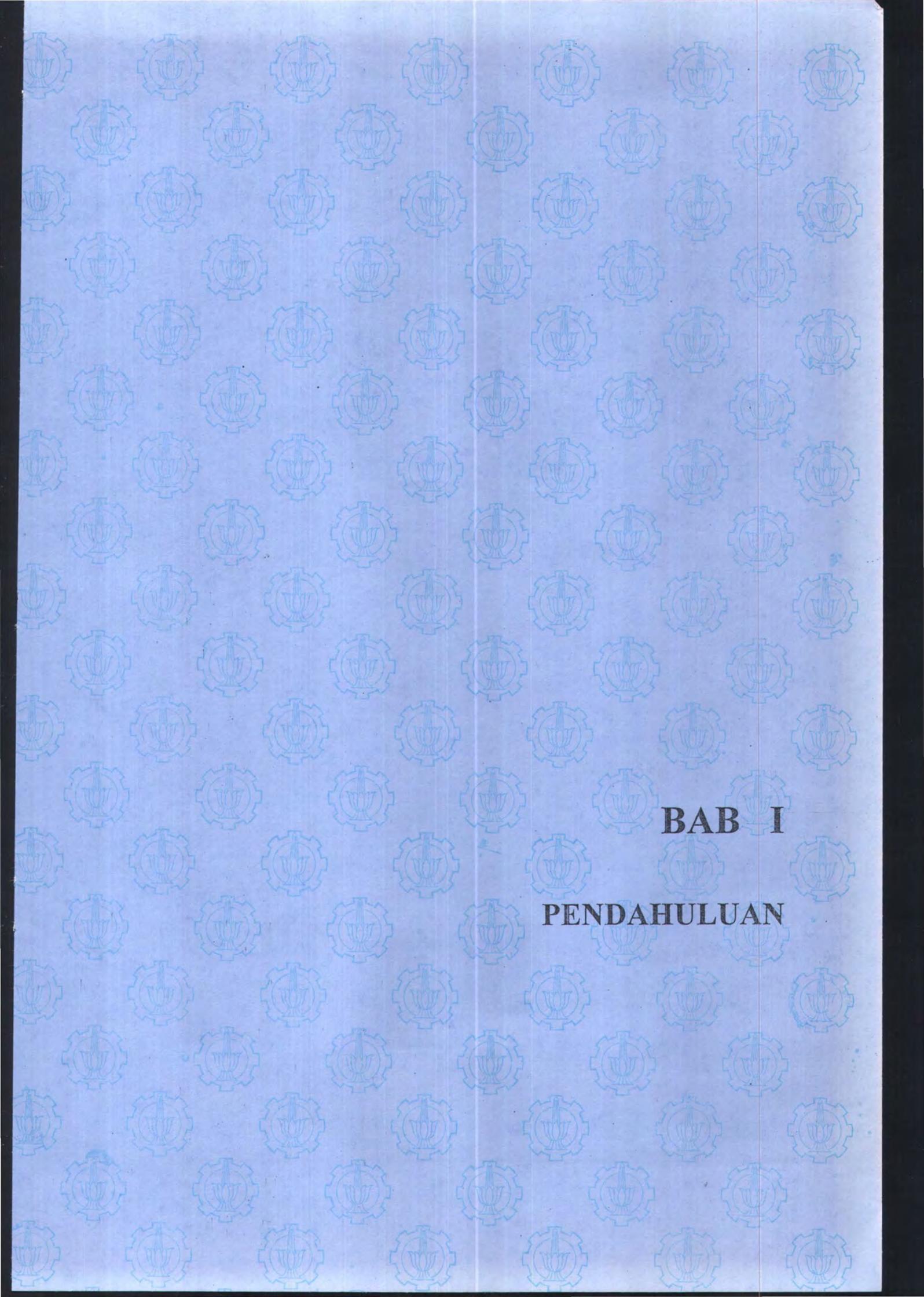
	Halaman
Gambar 2.1 Aksi dan reaksi sama besar dan berlawanan arah	8
Gambar 2.2 Gaya-gaya yang bekerja pada suatu benda	9
Gambar 2.3 Perbandingan antara logika konvensional dan logika fuzzy	12
Gambar 2.4 Istilah dalam logika fuzzy	13
Gambar 2.5 Proses fuzzifikasi	14
Gambar 2.6 Blok diagram sistem kontrol logika fuzzy	18
Gambar 2.7 NLX 220 dengan 28 pin	21
Gambar 2.8 Blok diagram NLX 220	26
Gambar 2.9 Fungsi keanggotaan temperatur	28
Gambar 2.10 Tipe fungsi keanggotaan	29
Gambar 2.11 Fungsi keanggotaan kecepatan	29
Gambar 2.12 Fungsi keanggotaan overlap	30
Gambar 2.13 Fuzzifikasi temperatur input	31
Gambar 2.14 Fungsi keanggotaan mengambang	33
Gambar 2.15 Defuzzifikasi immediate	38
Gambar 2.16 Defuzzifikasi accumulate	38
Gambar 2.17 Pewaktuan I/O	44
Gambar 2.18 Positif level detektor	47
Gambar 2.19 Prinsip kerja motor dc	48
Gambar 2.20 Persamaan motor dc	48
Gambar 2.21 Torsi motor dc	49

Gambar 2.22	Blok diagram port paralel	50
Gambar 2.23	Sinyal input dan output dari port paralel	52
Gambar 2.24	Diagram fungsi IC MAX 555	55
Gambar 3.1	Diagram blok pengendalian posisi benda terhadap arah gravitasi	56
Gambar 3.2	Perbedaan posisi benda terhadap arah gravitasi bumi	58
Gambar 3.3	Bentuk mekanik secara umum	59
Gambar 3.4	Rangkaian sensor posisi	60
Gambar 3.5	Rangkaian detektor	61
Gambar 3.6	Rangkaian kecepatan motor dc	63
Gambar 3.7	Diagram waktu tulis dan baca	65
Gambar 3.8	Referensi external	65
Gambar 3.9	Rangkaian ADC MAX555	67
Gambar 3.10	Input dan output pada NLX 220	69
Gambar 3.11	Variabel fuzzy untuk antecedent arah X, arahY dan sudutY	70
Gambar 3.12	Variabel antecedent sudut X	70
Gambar 3.13	Algoritma fuzzy	72
Gambar 3.14	Diagram alir akusisi data	73



DAFTAR TABEL

	Halaman
Tabel 2.1 Absolut maximum rating $T_a = 25\text{ }^\circ\text{C}$	24
Tabel 2.2 Analog conversion specification	24
Tabel 2.3 Specitifications and recommended operating conditions	25
Tabel 2.4 Organisai memory	39
Tabel 2.5 Commond byte/alamat ganjil	41
Tabel 2.6 Commond byte/alamat genap	42
Tabel 2.7 Alamat memory-pin pada port printer	53
Tabel 3.1 Pemilihan multiplexer pada hard wired	64
Tabel 3.2 Karakteristik timing	64
Tabel 4.1 Pengujian sensor untuk sumbu X	76
Tabel 4.2 Pengujian sensor untuk sumbu Y	76
Tabel 4.3 Pengujian switching	77
Tabel 4.4 Tegangan PWM dan pengendali motor DC sumbu X	77
Tabel 4.4 Tegangan PWM dan pengendali motor DC sumbu Y	78



BAB I

PENDAHULUAN

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi elektronika dewasa ini semakin pesat mulai dari sistem analog sampai sistem digital dengan banyaknya peralatan-peralatan yang sangat memberi kemudahan, keakuratan dan kenyamanan pekerjaan dalam kehidupan manusia. Salah satu teknologi elektronika yang mempunyai keterkaitan antara sistem analog dan sistem digital adalah kontroller menggunakan *logika fuzzy*.

Seiring dengan kemajuan teknologi kontrol maka kecenderungan manusia ingin melakukan pengontrolan terhadap suatu peralatan agar sesuai dengan keinginannya yaitu peralatan tersebut dapat menyesuaikan kondisi yang ada di sekitar peralatan tersebut dimana akan dapat memberikan suatu manfaat bagi manusia.

Perpaduan antara teknologi kontrol dengan teknologi elektronika yang menggunakan logika fuzzy akan memberikan solusi kemudahan suatu pengontrolan. Teknologi inilah yang dipakai pada tugas akhir ini untuk diterapkan pada sistem pengendalian posisi dalam rangka menjamin posisi benda pada suatu tempat agar selalu tegak lurus terhadap arah gravitasi bumi.

1.2 Permasalahan

Dari latar belakang yang telah diuraikan diatas timbul suatu permasalahan yaitu bagaimana merancang dan membuat suatu sistem pengendalian posisi benda yang cukup handal dalam merespon perubahan posisi tempat benda tersebut.

Pada tugas akhir ini diperlukan peralatan yang dapat mengatur posisi bidang agar selalu tegak lurus dengan arah gravitasi bumi dengan mengatur gerak lengannya. Lengan itu bergerak ke berputar mengitari arah salah satu sumbu koordinat. Dalam peralatan tersebut digunakan 2 lengan yang mengitari sumbu koordinat yang berbeda tetapi kedua lengan tersebut saling terkait. Diharapkan peralatan tersebut dapat menggunakan komponen yang minimal tetapi tidak menurunkan unjuk kerjanya.

1.3 Tujuan

Tujuan dalam tugas akhir ini :

1. Mempelajari logika fuzzy.
2. Mempelajari metoda perancangan rangkaian elektronika dengan menggunakan Fuzzy Logic.
3. Mempelajari sistem pengendalian posisi terhadap arah gravitasi bumi dengan memanfaatkan kontroller fuzzy..
4. Merancang dan membuat pengaturan posisi bidang dengan menggerakkan lengan-lenganya yang diatur oleh motor.

1.4 Pembatasan Masalah

Topik utama tugas akhir ini dibatasi pada perancangan dan pembuatan kontrol posisi yang menggunakan mikrokontroler fuzzy logic NLX220.

Kontrol posisi dilakukan dengan menerima input dari transducer untuk mengatur gerakan lengan. Transducer yang digunakan adalah jenis resistor variable sebanyak 4 transducer. 2 transducer digunakan untuk mengatur lengan yang mengitari sumbu X dan 2 transducer lainnya digunakan untuk mengatur lengan yang mengitari sumbu Y. Fungsi dari transducer tersebut untuk mengatur kecepatan gerak lengan dan mengatur posisi lengan dengan menggerakkan motor dc. Kecepatan lengan tersebut diatur dengan 3 kecepatan yaitu kecepatan rendah, sedang dan tinggi.

1.5 Metodologi

Langkah-langkah yang ditempuh dalam menyelesaikan tugas akhir ini adalah, sebagai berikut :

1. Dalam perencanaan alat ini terlebih dahulu dipelajari tentang logika fuzzy, pengaturan logika fuzzy dari literatur yang berkaitan.
2. Mendalami teori pendukung yang terkait dalam pembuatan alat tersebut seperti op-amp dan lain-lainnya.
3. Membuat secara blok diagram keseluruhan alat yang akan dibuat, dimana dengan blok tersebut dapat dipelajari dan diterapkan secara perblok untuk memudahkan pengujian.

4. Membuat program logika fuzzy yang telah disimulasikan untuk melakukan pengaturan kerja pada sistem perangkat kerasnya agar bekerja seperti yang diharapkan.
5. Melakukan pengujian peralatan tersebut dan melakukan pemecahan masalah selama melakukan pengujian agar peralatan tersebut sehingga dapat dianggap sebagai peralatan yang cukup handal.

1.6 Sistematika Penulisan

Sistematika penulisan buku tugas akhir ini dibagi dalam beberapa bab yaitu :

BAB I : Pendahuluan diuraikan mengenai latar belakang, permasalahan, metodologi dan sistematika penulisan.

BAB II : Teori penunjang membahas mengenai komponen dan sistem yang digunakan dalam tugas akhir ini.

BAB III : Perencanaan perangkat keras dan perangkat lunak yang dibahas di dalam perancangan alat yang meliputi rangkaian elektronika dan mekaniknya serta diagram alir dari perangkat lunak sampai pembuatan perangkat lunaknya.

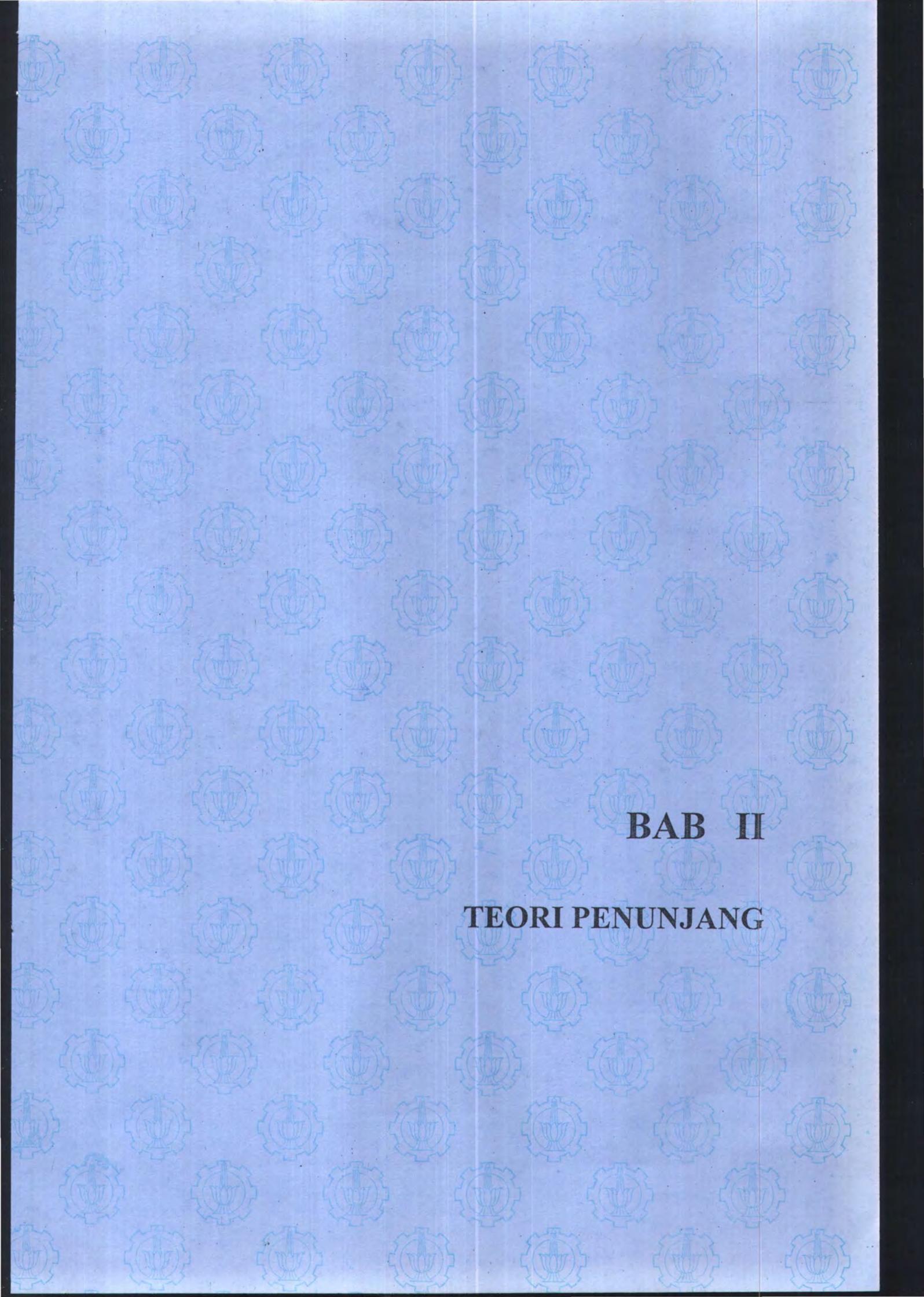
BAB IV : Pengujian dan pengukuran membahas tentang hasil-hasil yang dilakukan selama pengujian.

BAB V : Penutup, bab ini berisi kesimpulan dan saran-saran tugas akhir ini.

1.7 Relevansi

Dari tugas akhir ini diharapkan :

Menghasilkan referensi dan masukan bagi institusi pendidikan dan industri yang ingin melakukan perencanaan dan perancangan sistem pengendali posisi terhadap arah gravitasi yang sederhana tetapi cukup handal.



BAB II

TEORI PENUNJANG

BAB II

TEORI PENUNJANG

2.1 Pendahuluan

Dalam bab ini akan dibahas mengenai tentang dasar-dasar teori penunjang dalam perencanaan pengaturan posisi terhadap arah gravitasi bumi dengan menggunakan mikrokontroller logika fuzzy sebagai komponen pengendali utama.

Mula-mula dibahas terjadinya suatu pergerakan benda, kemudian teori tentang fuzzy yang merupakan dasar penting untuk merancang sistem kontrol yang dibuat. Dan akhirnya dibahas tentang rangkaian penunjang lainnya di dalam perencanaan dan pembuatan tugas akhir ini.

2.2 Pergerakan Benda

Gerak ialah konsep relatif dan harus selalu mengacu kepada suatu kerangka acuan tertentu yang dipilih oleh pengamat. Gerakan suatu benda merupakan hasil langsung dari interaksinya dengan benda-benda yang disekitarnya. Gerakan benda yang jatuh ke bawah merupakan hasil interaksi antara benda dengan bumi. Bila suatu benda dan bumi merupakan partikel-partikel, maka hasil kecepatan 2 partikel tersebut tidaklah konstan tetapi berubah terhadap waktu dan biasanya lintasannya melengkung, seperti ditunjukkan gambar 2.1 oleh kurva 1 dan 2. Pada suatu saat t , partikel (1) berada di A dengan kecepatan v_1 dan partikel (2) berada di B dengan kecepatan v_2 . Kemudian pada

saat t' , partikel-partikel itu berada di A' dan B' dengan kecepatan v_1' dan v_2' .

Perubahan kecepatan partikel 1 dan 2 selang waktu $\Delta t = t' - t$ adalah :

$$\Delta v_1 = v_1' - v_1 \quad \text{dan} \quad \Delta v_2 = v_2' - v_2.$$

Karena dalam eksperimen didapatkan bahwa kedua perubahan dalam kecepatan selama waktu tertentu mempunyai arah yang berlawanan maka dapat ditulis :

$$\Delta v_1 = -\Delta v_2$$

Dan apabila partikel tersebut mempunyai suatu massa maka perubahan momentumnya adalah :

$$\Delta p_1 = -\Delta p_2$$

$$\Delta p_1 = \Delta(m \cdot v_1) = m (\Delta \cdot v_1).$$

$$\Delta p_2 = \Delta(m \cdot v_2) = m (\Delta \cdot v_2).$$

Laju perubahan momentum sebuah partikel yang disebabkan interaksi partikel satu dengan yang lainnya disebut gaya yang besarnya :

$$F_1 = \frac{dp_1}{dt}$$

sehingga :

$$F_1 = -F_2.$$

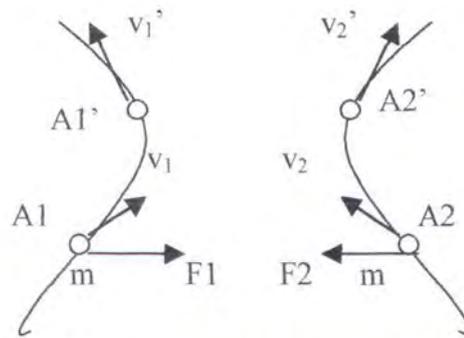
Dimana F_1 adalah gaya pada partikel (1) karena interaksinya dengan partikel (2) dan F_2 adalah gaya pada partikel (2) karena interaksinya dengan partikel (1).

Dari persamaan diatas, dapat dikatakan secara umum bahwa:

$$F = \frac{d(mv)}{dt}.$$

Jika massanya konstan maka akan didapat :

$$F = m \frac{dv}{dt} \quad \text{atau} \quad F = m \cdot a$$



Gambar 2.1. Aksi dan reaksi adalah sama besar dan berlawanan arah.

Menurut hukum Newton tentang gravitasi universal¹ dapat dinyatakan dengan mengatakan bahwa :

Interaksi gravitasi antara dua benda dapat dinyatakan dengan sebuah gaya tarikan sentral yang sebanding dengan massa benda-benda dan berbanding terbalik terhadap kuadrat jarak yang memisahkannya.

$$F = \gamma \frac{mM}{R^2}$$

γ = Konstanta perbandingan ($6,67 \times 10^{-11} \text{ N m}^2 \text{ kg}^{-2}$)

m = Massa benda.

M = Massa bumi ($5,97 \times 10^{24} \text{ kg}$).

R = Jarak benda dengan pusat bumi (jari-jari bumi : $6,37 \times 10^6 \text{ m}$).

Hubungan antara gaya gravitasi tersebut dengan persamaan gaya secara umum dapat disimpulkan bahwa percepatan benda jatuh ke bumi yang kemudian disebut dengan percepatan gravitasi adalah :

$$g = a \quad \text{dimana} \quad a = \gamma \frac{M}{R^2}$$

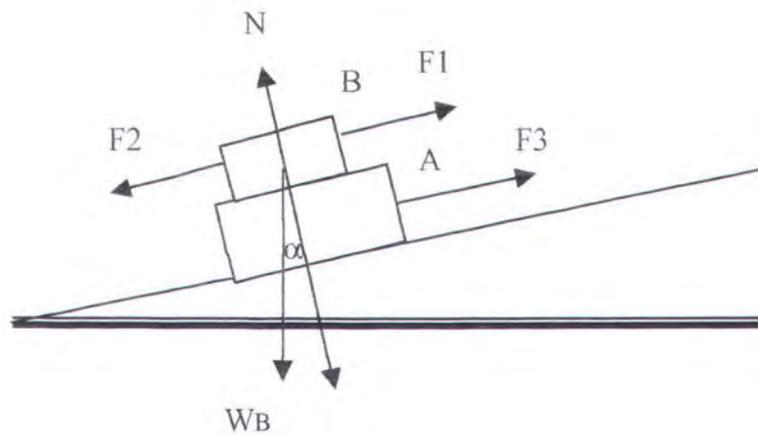
g = percepatan gravitasi bumi.

¹Marcelo Alonso, Edward J. Finn, DASAR-DASAR FISIKA UNIVERSITAS, (Erlangga), 1990, p.289

Jika suatu benda berinteraksi dengan benda yang lainnya maka gaya yang bekerja pada benda tersebut maka total gaya tersebut adalah :

$$F_1 + F_2 + \dots + F_n = F.$$

Hal ini dapat dilihat dari gambar berikut ini :



Gambar 2.2. Gaya-gaya yang bekerja pada suatu benda.

Maka gaya total yang bekerja pada benda B adalah :

$$F = F_1 + F_2$$

Dimana :

$$F_2 = W_B \sin \alpha$$

$$W_B = m_B \cdot g. \text{ (gaya berat benda B)}$$

Apabila gaya $F_1 = 0$, maka :

$$m \cdot a = 0 - F_2 \text{ sehingga } a = -g \sin \alpha.$$

Dilihat dari kejadian di atas maka dapat diketahui bahwa semakin besar sudut lintasan permukaan maka semakin besar percepatan benda tersebut untuk bergeser ke bawah.

2.3 Logika Fuzzy

2.3.1 Prinsip Logika Fuzzy

Fuzzy secara leksikal mengandung arti tak jelas, samar atau kabur. Dalam teori sistem pengaturan, kata fuzzy dihubungkan dengan kata logika, sehingga didapat kata logika fuzzy yang berarti suatu logika yang samar. Dengan kata lain menentukan fenomena-fenomena di alam nyata yang mengandung sifat serba tidak tepat atau samar kita tentukan aturan-aturan yang samar juga.

Pada tahun 1965 Profesor Lotfi Zadeh dari University of California menyatakan bahwa suatu hal tidak dapat dinyatakan dengan pasti namun dapat memiliki kondisi diantara keduanya. Kondisi *true* atau *false* tidak dapat menerangkan banyak kejadian di alam semesta, dimana kebanyakan kejadian tersebut merupakan perubahan dari benar ke salah (atau sebaliknya) dan bukan hanya bernilai benar atau salah.

Untuk menerangkan perubahan yang tidak tertentu dari benar ke salah tersebut, Zadeh mengemukakan teori yang memperluas himpunan klasik menjadi apa yang dinamakan himpunan fuzzy. Logika fuzzy merupakan logika dengan nilai banyak (multi valued logic). Sehingga logika fuzzy mengizinkan adanya tingkat keanggotaan (degree of Membership) dari suatu anggota himpunan bernilai antara 0 dan 1, sehingga sesuatu dapat bernilai sebagian benar dan sebagian salah dalam suatu waktu. Bart Kosko telah membuktikan bahwa logika boole adalah suatu kasus khusus dari logika fuzzy.

Contoh sederhana tentang prinsip fuzzy adalah pada pembagian usia seseorang, misalnya sebagai berikut²:



² Jun Yan, Michael Ryan, James Power, USING FUZZY LOGIC, (Prentice Hall), 1994, p. 3

Usia muda < 35 tahun

usia menengah 35 tahun sampai 55 tahun

lanjut usia \geq 56 tahun

Jika menggunakan prinsip Boole maka terlihat bahwa usia 55 dan 56 memiliki klasifikasi yang berbeda, padahal kedua usia tersebut tidak memiliki beda yang banyak.

Hal ini memperlihatkan kelemahan logika Boole untuk mengklasifikasikan suatu anggota yang terletak pada titik perbatasan, lagipula penentuan batas tersebut banyak tergantung pada tiap individu, karena seseorang mungkin saja memiliki klasifikasi dan nilai batas yang berbeda. Pada fuzzy usia 55 dan 56 tidak dibedakan dengan jelas namun samar, sehingga usia 55 dapat dianggap terletak antara usia menengah dan usia tua begitupula dengan usia 56. Jadi usia 55 dan 56 memiliki tingkat keanggotaan yang berbeda, beda tingkat keanggotaan tersebut akan menentukan kecenderungan klasifikasinya. Karena itu usia 55 dapat dikatakan lebih cenderung sebagai usia menengah sedangkan usia 56 cenderung lebih tua.

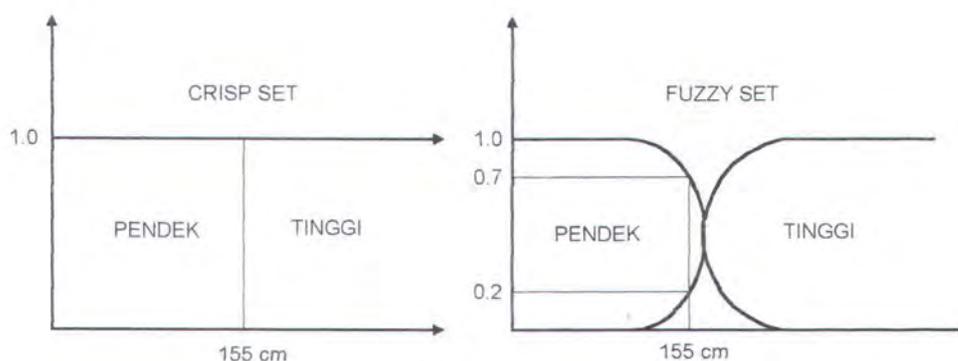
Dengan makin bertambah atau berkurangnya usia maka tingkat keanggotaannya untuk suatu himpunan akan semakin bertambah sehingga lebih jelas kondisinya.

2.3.2 Teori Logika Fuzzy

Pengembangan logika fuzzy dilakukan untuk menjawab permasalahan “two valued logic” yaitu logika yang hanya mengenal suatu keadaan dalam 2 kemungkinan, seperti benar dan salah. Teori “two valued logic” terbukti efektif

untuk memecahkan masalah, dengan syarat permasalahannya dapat dideskripsikan kuantitasnya secara tepat.

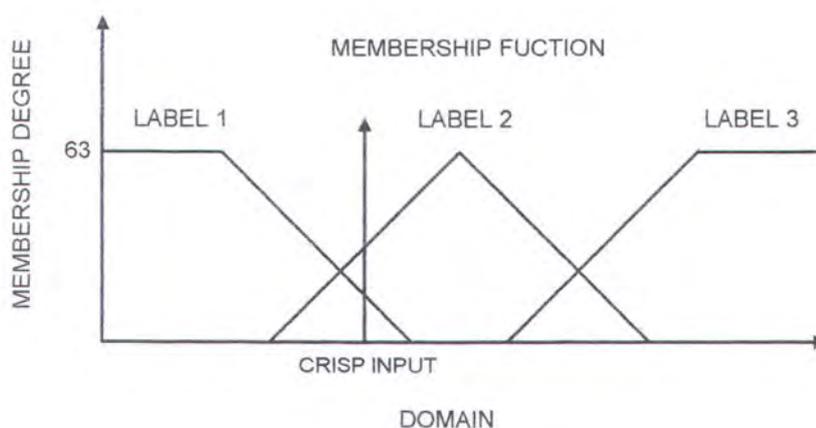
Sebagai contoh adalah menentukan apakah orang dengan tinggi 160 cm termasuk tinggi, sedang atau pendek. Apabila menggunakan metode “two valued logic” maka pemecahannya adalah memberi range dibawah 155 cm sebagai pendek dan diatas 155 sebagai tinggi, kesulitan yang pertama adalah menentukan batas pendek dan tinggi secara kuantitatif, karena setiap orang mempunyai nilai yang berbeda. Kesulitan yang kedua adalah menentukan akan dimasukkan kemanakah nilai 155 cm tersebut (karena tepat berada di perbatasan). Logika fuzzy membagi suatu keadaan dalam interval $[0,1]$ yang secara intuitif dapat dinyatakan dalam contoh diatas dengan mengelompokkan orang sebagai pendek, agak pendek, sedang, agak tinggi dan tinggi. Jika logika fuzzy ini diterapkan pada permasalahan tinggi badan diatas maka dapat dikatakan bahwa orang dengan tinggi badan 155 cm mempunyai nilai kebenaran 0,7 pendek dan 0,2 tinggi. Hal ini akan memberikan kesimpulan bahwa orang tersebut cenderung pendek. Untuk lebih jelas, permasalahan tersebut dapat dilihat pada grafik gambar 2.3.



Gambar 2.3. Perbandingan Metode Logika Konvensional dan Logika Fuzzy

Dalam logika fuzzy terdapat istilah-istilah sebagai berikut (lihat gambar)

1. Fungsi keanggotaan (*membership Function*) adalah fungsi yang memetakan masukan nyata (*crisp input*) dengan domainnya pada derajat keanggotaan.
2. Skala keanggotaan / derajat keanggotaan (*Degrees of Membership*) merupakan skala dimana nilai nyata masukan setara dengan fungsi keanggotaannya. Skala ini bernilai antara 0 sampai 1.
3. Nilai masukan (*Crisp Input*) adalah nilai masukan yang bernilai skalar, tertentu dan tunggal. Untuk sistem kontrol nilai ini dapat berupa masukan dari transducer. Misalnya nilai suhu sebesar 27 derajat celcius.
4. Label adalah nama yang merupakan penggambaran dari fungsi keanggotaan.
5. Domain adalah lebar total dari fungsi keanggotaan yang merupakan jangkauan dari suatu konsep angka-angka tertentu, dimana fungsi keanggotaan itu diperjelas.
6. Nilai tengah merupakan nilai dari fungsi keanggotaan dimana mempunyai nilai kebenaran sempurna ('1' atau '0').



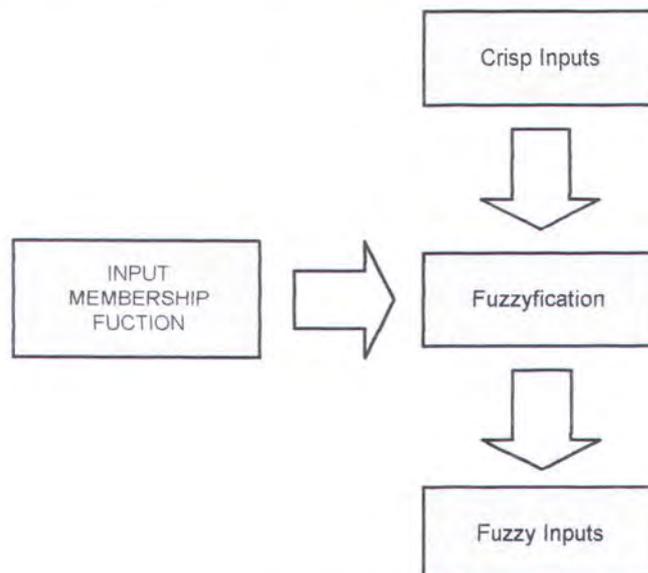
Gambar 2.4. Istilah dalam Logika Fuzzy

2.3.3 Proses Logika Fuzzy

Dalam pemecahan permasalahan menggunakan metode logika fuzzy diperlukan tiga tahap proses, yaitu:

- a. Proses fuzzifikasi yaitu mengubah variabel input yang berupa variabel *crisp* ke dalam variabel fuzzy, yang dimaksud dengan variabel *crisp* adalah variabel yang berorientasi numerik.
- b. Proses evaluasi aturan (*rule evaluation*) yaitu mencari nilai aksi (*action*) dengan memberikan bobot pada setiap aturan yang diberikan.
- c. Proses defuzzification yaitu merubah variabel fuzzy yang terbentuk dari proses evaluasi aturan menjadi variabel *crisp*.

Proses fuzzification adalah rangkaian usaha untuk mentransformasi variabel input yang mulanya bersifat numerik (*crisp input*) ke variabel fuzzy (*fuzzy inputs*). Transformasi ini dipengaruhi oleh fungsi keanggotaan yang digunakan. Untuk lebih jelasnya, proses fuzzifier diperlihatkan pada gambar 2.5.



Gambar 2.5. Proses Fuzzifikasi³

³ ----, FUZZY LOGIC EDUCATION PROGRAM (Motorolla)

Proses evaluasi aturan (*rule evaluation*) adalah rangkaian usaha untuk menentukan nilai aksi sebagai tanggapan atas setiap input atau kombinasi input yang diberikan dengan memberi bobot pada masing-masing aturan yang telah ditetapkan. Di dalam proses ini terdapat dua komponen utama yaitu himpunan aturan (*rule set*) dan metode evaluasi aturan.

Himpunan aturan (*rule set*) adalah kombinasi semua aturan yang diperlukan untuk menentukan tanggapan terhadap input atau kombinasi input yang diberikan. Aturan ini bersifat linguistik dan mempunyai bentuk “jika...maka.” (Ifthen...) Bentuk aturan ini diciptakan berdasarkan keinginan untuk⁴ :

1. Menyediakan cara yang mudah bagi para ahli untuk mengekspresikan pengetahuan dan pengalaman mereka.
2. Menyediakan cara yang mudah bagi para desainer untuk menyusun dan memprogram aturan fuzzy.
3. Mengurangi biaya desain.

Bentuk umum aturan fuzzy adalah : jika x_1 adalah A_{k1} dan X_2 adalah A_{k2} atau X_3 adalah A_{k3} ... maka y_1 adalah B_{k1} , dimana x_1 , x_2 , dan x_3 adalah input kejadian 1 (*antecedent*), kejadian 2, dan kejadian 3; A_{k1} , A_{k2} dan A_{k3} adalah himpunan fuzzy yang berkorelasi dengan kejadian; y_1 adalah output kejadian dan B_{k1} adalah himpunan fuzzy yang berkorelasi dengan output.

Metode evaluasi aturan adalah metode yang digunakan dalam mengevaluasi aturan yang telah ditetapkan. Ada beberapa metode evaluasi aturan

⁴ Jun Yan, Michael Ryan, James Power, op.cit., p.18.

yang sering dipakai seperti mini rule (*Mamadani*), product rule (*Larsen*), Max-Min (*Zadeh*)⁵, Arithmetic rule (*Zadeh*) dan Boolean.

Metode yang digunakan dalam tugas akhir ini adalah metode Max-Min sesuai dengan metode yang dipakai oleh fuzzy microcontroller NLX220⁶ yang digunakan. Konsep dari metode ini adalah mencari nilai minimum pada setiap rule, kemudian mencari nilai maksimum dari himpunan aturan yang berkorelasi dengan satu kejadian output sehingga dapat ditentukan nilai aksi yang seharusnya, demikian diulangi untuk setiap kejadian output.

Nilai minimum pada setiap rule menggambarkan derajat ke-fuzzian aturan tersebut, sedangkan nilai maksimum dari nilai minimum himpunan aturan yang berkorelasi dengan suatu kejadian output menggambarkan kejadian yang paling dapat “dipercaya” karena mempunyai derajat ke-fuzzian paling tinggi sehingga aturan yang mempunyai derajat ke-fuzzian paling tinggi diambil sebagai aturan paling dapat dipercaya (*the winning rule*).

Proses defuzzification adalah tahap terakhir proses logika fuzzy yang berfungsi untuk mengubah output yang berupa output fuzzy menjadi output crisp. Ada beberapa metode defuzzification, yaitu Center of Gravity (*COG*), fuzzy singleton, accumulate, dan Immediate sesuai dengan spesifikasi NLX220 yang digunakan.

Metode accumulate pada output berarti nilai output sama dengan nilai aksi aturan yang menang ditambah dengan nilai output sebelumnya sehingga metode ini dapat digunakan sebagai pendekatan proses integrasi. Sedangkan metode immediate berarti nilai output sama dengan nilai aksi aturan yang menang.

⁵ Ibid., p.32

⁶ Neural Logic, FUZZY MIKRO DEVELOPMENT SYSTEM (American NeuraLogic), p. 6-3

2.3.4 Pengontrolan Dengan Logika Fuzzy

Tujuan utama suatu sistem kontrol adalah menghasilkan suatu keluaran yang dikehendaki untuk setiap masukan yang diberikan. Rangkaian usaha yang dilakukan untuk mengolah masukan menjadi keluaran yang dikehendaki disebut dengan proses kontrol. Secara konvensional dikenal beberapa proses kontrol yaitu metode look-up table, metode pemodelan secara matematis untuk mencari fungsi transfer antara masukan dan keluaran, dan metode dengan menggunakan logika fuzzy.

Model look-up table efektif hanya jika sistem yang dikontrol hanya mempunyai resolusi yang rendah dan variabel masukan sedikit. Kelemahan utama sistem ini adalah jika sistem rumit maka akan menghabiskan banyak tempat di memori, menimbulkan gangguan pada sistem karena adanya event yang tidak tertanggapi, dan meloncatnya nilai keluaran dari tabel ke tabel yang lainnya.

Model matematis dari suatu sistem harus secara tepat menggambarkan perilaku sistem terhadap masukan yang diberikan. Untuk sistem yang rumit hal ini menjadikan pembuatan model matematis amat sulit, dengan hasil suatu persamaan yang kompleks dan sulit diaplikasikan. Pembuatan model matematis juga membutuhkan keahlian khusus sehingga meningkatkan biaya desain. Kelemahan lain dari model ini adalah keterbatasan data yang ada sehingga tidak memungkinkan pembuatan model matematis tapi amat mengurangi keakuratan sistem.

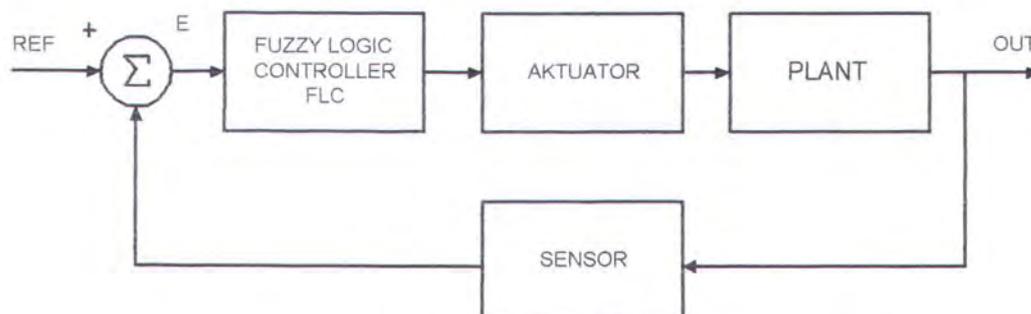
Penggunaan teknologi fuzzy dalam rekayasa proses dan sistem informasi akan menghasilkan alat-alat yang handal, tahan, luwes dan lebih canggih dibandingkan dengan alat-alat digital biasa. Hal ini akan memproduksi sistem

pengambilan keputusan, sistem kontrol otomatis yang akan membawa kepada mesin yang mempunyai daya pikir (*intelligence machine*).

Akibat lebih lanjut dari penerapan logika fuzzy dalam kontrol akan mempunyai keuntungan-keuntungan sebagai berikut :

- Kemudahan bagi pemakai yang lebih baik.
- Kemampuan menyesuaikan diri yang lebih baik.
- Kemampuan untuk diagnosa sendiri.
- Kinerja yang lebih baik dengan konsumsi daya yang lebih rendah.

Struktur dasar penggunaan logika fuzzy untuk sistem kontrol, ditunjukkan seperti gambar di bawah ini :



Gambar 2.6. Blok Diagram Sistem Kontrol Logika Fuzzy

Sistem kontrol otomatis biasanya terdiri dari empat bagian utama; sensor, controller, aktuator dan plant. Sensor berfungsi sebagai pengambil data perilaku sistem. Aktuator memberikan daya untuk menggerakkan plant (yaitu peralatan yang dikontrol), sehingga dicapai suatu harga yang diinginkan. Kontroler berfungsi memberikan sinyal perintah ke aktuator sesuai aksi kontrol menurut

besarnya deviasi (error) yaitu selisih antara referensi dan keluaran yang terukur oleh sensor.

Sistem kontrol yang digunakan adalah sistem kontrol dengan umpan balik⁷ karena sistem ini cenderung menjaga hubungan yang telah ditentukan antara keluaran dan masukan acuan dengan membandingkannya dan menggunakan selisihnya sebagai alat pengontrolan. Dengan demikian sistem kontrol dengan menggunakan umpan balik mempunyai keunggulan dibandingkan dengan sistem kontrol tanpa umpan balik.

2.4 Fuzzy Microcontroller NLX220⁸

2.4.1 Deskripsi Umum

NLX220 merupakan perangkat (*device*) yang membentuk kalkulasi logika fuzzy secara langsung di hardware. Karena memang dibuat untuk controller, maka mudah dalam pemakaian, performansi, feature, dan tangguh dalam lingkungan yang kasar. Perangkat ini terdiri dari 4 analog input dan 4 analog output dengan sumber clock internal. NLX220 akan menyerap daya yang rendah saat operasi normal dan mempunyai mode power-down yang akan mengurangi daya dengan faktor 10. Fuzzy logic sangat sesuai dengan proses-proses yang mempunyai input data yang acak dan sistem tidak linier untuk laju sistem kontrol yang tangguh. Metodologinya memakai deskripsi secara linguistik dari sistem, sehingga menjadikannya sangat intuitif dan mudah untuk dipakai. Dapat juga dipakai untuk menambahkan kecerdasan pada produk-produk industri, misalnya untuk

⁷ Katsuhito Ogata, Edi Leksono, TEKNIK KONTROL AUTOMATIK JILID 1, (Jakarta, Erlangga, 1991),. P.3.

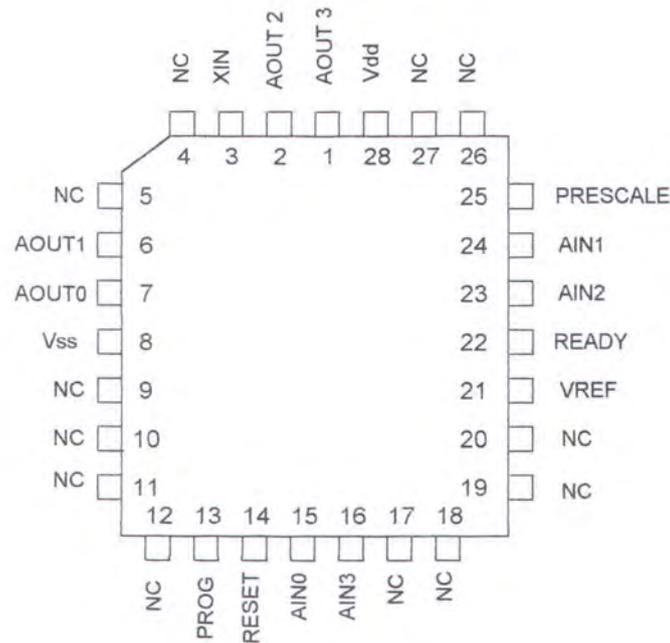
⁸ Neural Logic, FUZZY MICROCONTROLLER DEVELOPMENT SYSTEM (American NeuraLogic),. p. 1.

meningkatkan performansi, menambah feature, dan meningkatkan efisiensi. NLX220 bisa diprogram yang sesuai untuk development dan produksi yang terbatas. Kompatibilitas pin NLX220 memakai teknologi OTP untuk storage dan sesuai untuk produksi yang beragam.

Memori menyimpan fungsi keanggotaan fuzzy (*MF Fuzzy*) dan parameter-parameter rule. Pengorganisasian memori fleksibel dan efisien mengadaptasi terhadap berbagai keperluan aplikasi. Perangkat ini menyimpan 111 variabel Fuzzy yang diorganisasikan menjadi rule seperti yang dikehendaki.

Perangkat menyediakan 6 tipe fungsi keanggotaan yang berbeda untuk berbagai aplikasi. Fungsi keanggotaannya memiliki kemiringan yang konstan dan hanya perlu spesifikasi tipe, lebar, dan center. NLX220 juga menyediakan fungsi keanggotaan *floating*, dimana lebar dan center dari fungsi keanggotaan dapat dibuat “mengambang” (*float*) atau bervariasi secara dinamis. Fungsi keanggotaan *floating* dapat digunakan untuk mengukur penurunan, membuat timer, atau meng-*adjust* untuk men-drive sensor.

Ada dua metode Defuzzifikasi, yaitu *immediate* dan *accumulate*. Mode *Immediate* akan men-drive output untuk harga yang sudah tertentu dan *accumulate* untuk menambahkan harga yang telah ada.



Gambar 2.7. NLX220 dengan 28 pin.

Informasi aplikasi dimasukkan secara mudah menggunakan sistem *software INSIGHT* yang dapat dijalankan pada *Windows*. Diperlukan sedikit pengetahuan tentang logika fuzzy untuk dapat menggunakan perangkat sistem tersebut

2.4.2 Features

- Sebuah IC controller Fuzzy Logic yang lengkap
- Flexibel dan self-adapting
- EEPROM atau OTP (One Time Programming)
- Empat (4) bit input analog
- Empat (4) bit output analog
- Enam (6) tipe Membership Function (MF)
- 111 variabel Fuzzy

- 50 Rules
- PC-based development system

2.4.3 Aplikasi

- Power dan Management baterai
- Kontrol Motor
- Kontrol Heater
- Otomotif
- Industri Kontrol

2.4.4 Fungsi Pin-pin dari NLX220

Input

RESET, untuk menginisialisasi perangkat dengan sinyal aktif low. Harus tetap aktif hingga sedikitnya 8 clock untuk memastikan operasi yang lama telah habis. Dapat diaktifkan dengan rangkaian delay power-up. Dengan Reset akan mengaktifkan mode low-power.

AIN (0:3), Analog Input Data, Input data analog yang secara internal akan dikonversikan ke data digital 8 bit. Input yang tidak dipakai harus dihubungkan ke ground.

XIN (Clock input), boleh dipakai eksternal input clock atau dengan kristal, di mana ujung satunya di-ground-kan.

PROG, untuk saat pemrograman NLX220. Pin ini tidak dipakai pada NLX220. Sehingga pin ini dalam operasinya harus dihubungkan ke ground.

PRESCALE, input logika '1' menandakan dalam mode prescale dan '0' dalam operasi normal. Pin ini di-ground-kan saat mode prescale tidak pernah digunakan atau dihubungkan dengan pin *READY* untuk operasi kontinyu. Mode juga bisa dipanggil selama pengoperasian oleh logika eksternal. Setelah *RESET* diaktifkan, pin *PRESCALE* harus diberi logika rendah untuk selama paling sedikit empat siklus clock. Cara operasi prescale dijelaskan pada akhir bab ini.

Output

AOUT (0:3), Analog Output Data, 8 bit data digital dikonversikan secara internal ke level analog.

READY, setelah reset pin ini akan menunjukkan bahwa NLX220 akan memulai mensampel dan memproses data. Pin ini seharusnya tidak dihubungkan atau disambungkan dengan *PRESCALE* selama pengoperasian.

VREF, memfilter referensi tegangan internal, dihubungkan ke ground dengan 0,1uF kapasitor.

Table 2.1 Absolute maximum Ratings Ta = 25 C

Parameter	Min	Max	Unit
Vdd	-0,5	7,0	V
Vss	0	0	V
Digital Input	0	Vdd	V
Analog Input	0	Vdd	V
Power Dissipation		100	mW
Storage temperatur	-50	150	C

Tabel 2.2 Analog Conversion Specifications

Parameter	Value	Units
Resolution	1	Bit
Slew Rate, Tracking	1,6	V/ms max
Zero Code Error	1x	LSB
Full Scale Error	1x	LSB
Signal to Noise Ratio	45	dBmin
Sampling Rate	10KHz	Per Channel

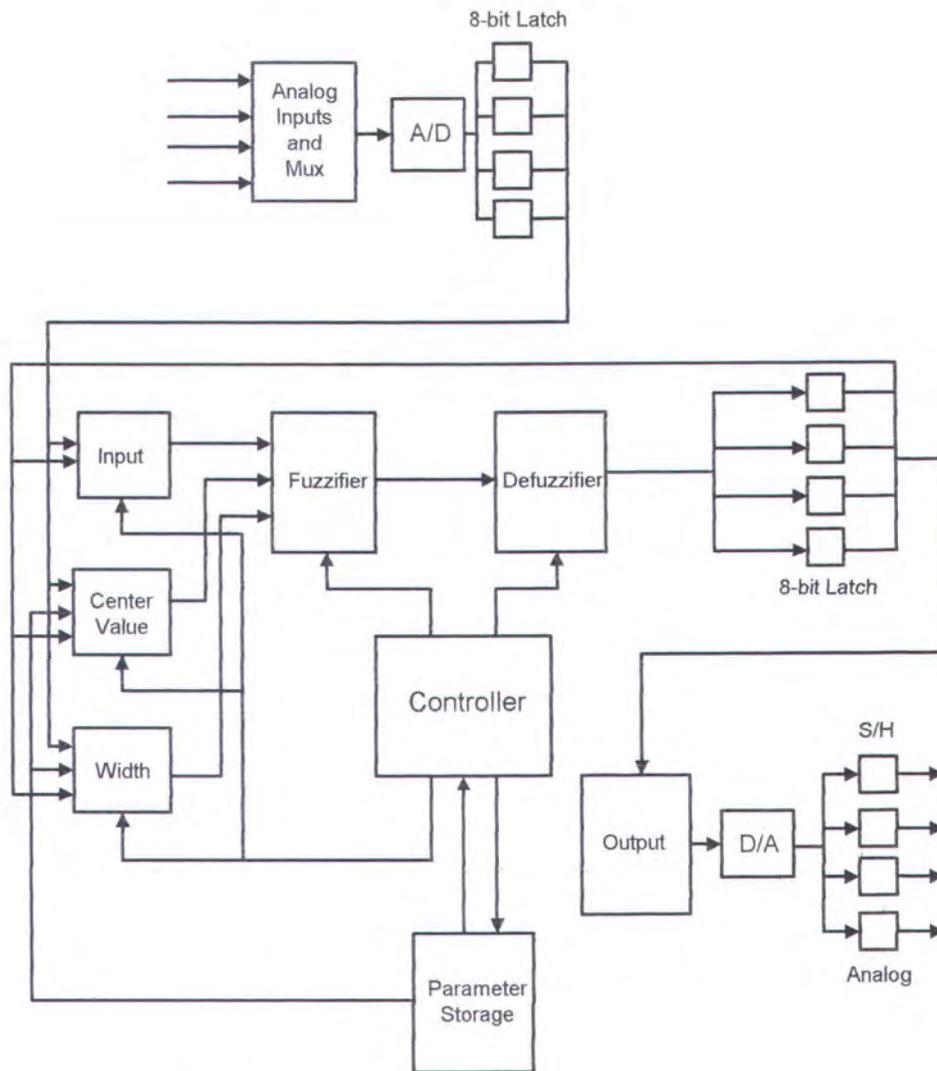
Tabel 2.3 Specifications and Recommended Operating Conditions

	Parameter	Min	Norm	Max	Unit
Vdd	Supply Voltage	4,75	5,0	5,25	V
Idd	Supply Current				mA
Iol	Digital output Low Level Current			15	mA
Ioh	Digital output High Level Current			-40	uA
F	Clock Frequency	1		10	MHz
Vil	Digital input Low Level voltage	0		0,8	V
Vih	Digital input High Level voltage	3,5		Vdd	V
Iil	Digital input low Level voltage			-40	uA
Iih	Digital input High Level voltage				uA
Zin	Analog Input impedance	100	150	250	kOhm
Vin	Analog input Voltage	0		Vdd-0,5	V
Vo	Analog Output voltage range	Vss+0,5			V
Io	Analog Output Current	-5		5	mA
Tw	Reset Pulse Width	100			ms
Tsv	Reset inactive before clock	10			ms
Ta	Operating Ambient temperature	0		70	C

2.4.5 Arsitektur Perangkat

Perangkat ini adalah stand alone kontroller Fuzzy logic yang membentuk semua kalkulasi di dalam hardware dan tidak memerlukan software. Input dapat secara langsung dihubungkan ke sensor atau switch, demikian juga outputnya langsung dihubungkan dengan piranti analog atau digunakan untuk fungsi kontrol. Komponen utama NLX220 adalah Fuzzifier, Deffuzzifier, dan Kontroller. Fuzzifier mengkonversikan input data ke dalam data Fuzzy dan dalam hubungannya dengan kontroller akan mengevaluasi data fuzzy dengan definisi

rule set yang dimasukkan, yang menggambarkan sistem kontrol yang dimaksud. Setelah rule-rule dievaluasi, defuzzifier memberikan nilai aksi ke output yang bersesuaian.



Gambar 2.8. Blok Diagram NLX220

2.4.5.1 Pengembangan Sistem Logika Fuzzy

Untuk mengerti operasi perangkat, perlu dimengerti bagaimana memasukkan model logika fuzzy dan bagaimana melakukan perhitungan logika fuzzy. Bagian berikut akan menjelaskan konsep dasar sistem fuzzy.

2.4.5.2 Fungsi Keanggotaan (Membership Function)

Bagian pertama yang harus dipahami adalah mengenai fungsi keanggotaan. Fungsi keanggotaan digunakan untuk membagi-bagi input, sehingga input akan memiliki suatu range yang tertentu sesuai dengan apa yang diinginkan.

Setiap input akan dibandingkan untuk mengetahui letaknya di dalam range yang telah didefinisikan. Input yang masuk akan memiliki suatu nilai fuzzy dimana semakin besar nilainya menunjukkan bahwa input tersebut merupakan anggota dari membership function itu, dan sebaliknya semakin kecil nilainya maka input tersebut bukan merupakan anggota dari range yang didefinisikan.

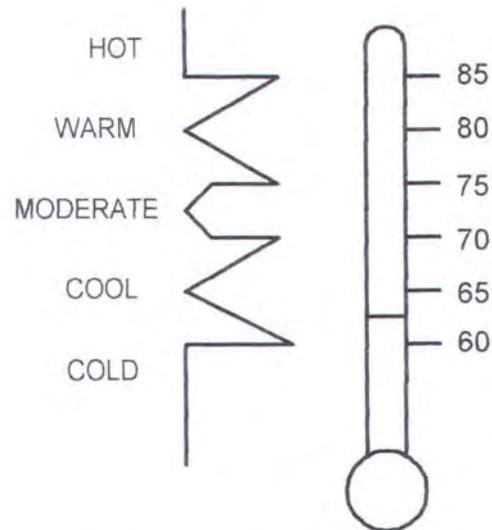
Membership function mempunyai nama yang dipilih oleh perancangnya sendiri, seperti panas, cepat atau tinggi, untuk mengklasifikasikan data tersebut. Termometer dapat digunakan untuk menggambarkan konsep fungsi keanggotaan dan menunjukkan bagaimana logika fuzzy bekerja seperti manusia.

Dalam hal ini termometer, pembagian suhunya dibuat sehalus mungkin, misal :

1. Di bawah 60 F = Dingin
2. 60 F - 70 F = Cool
3. 70 F - 75 F = Moderat
4. 75 F - 85 F = Warm
5. Di atas 85 F = Panas

Pembagian tersebut adalah suatu cara intuisi seseorang untuk membagi-bagi temperatur berdasarkan perasaan. Seseorang dapat menjelaskan bahwa ruang dengan suhu 60 derajat Fahrenheit adalah sejuk. Dalam logika Fuzzy lima pembagian tersebut disebut fungsi keanggotaan dan digambarkan pada gambar 2.9

Fungsi keanggotaan dapat dipisah-pisahkan seperti ditunjukkan atau dapat pula bisa saling tindih atau overlap. Hal tersebut memungkinkan data jatuh di daerah overlap yang termasuk di dua fungsi keanggotaan. Suhu sebagai contoh dapat diterangkan agak sejuk dan agak dingin.



Gambar 2.9. Fungsi Keanggotaan Temperatur

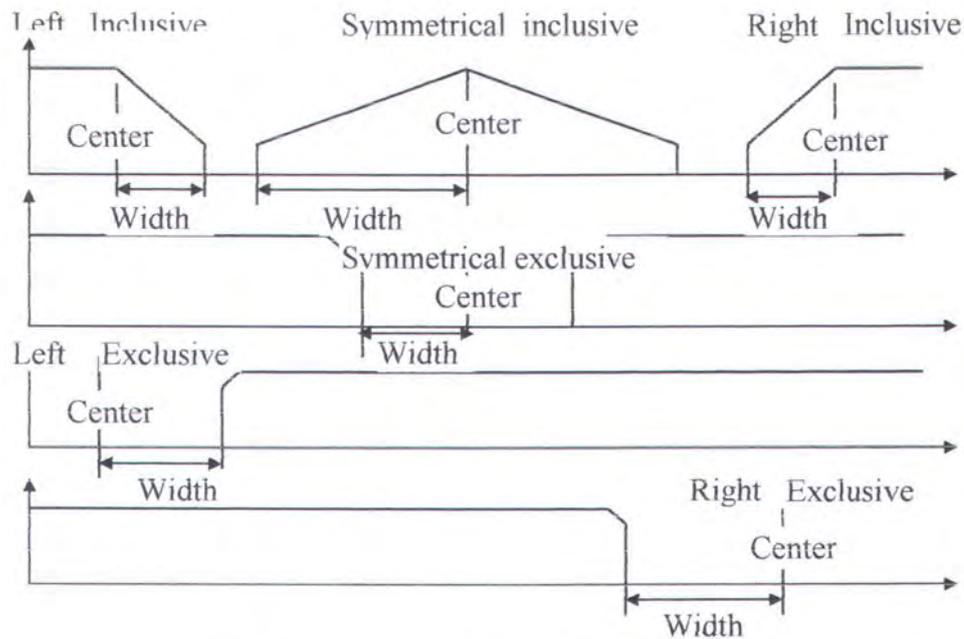
NLX220 memiliki enam fungsi keanggotaan yang berbeda dengan kemiringan yang konstan seperti gambar.

Fungsi keanggotaan tersebut terdiri dari :

1. Left Inclusive
2. Symmetrical Inclusive
3. Right Inclusive
4. Symmetrical Exclusive
5. Left Exclusive
6. Right Exclusive

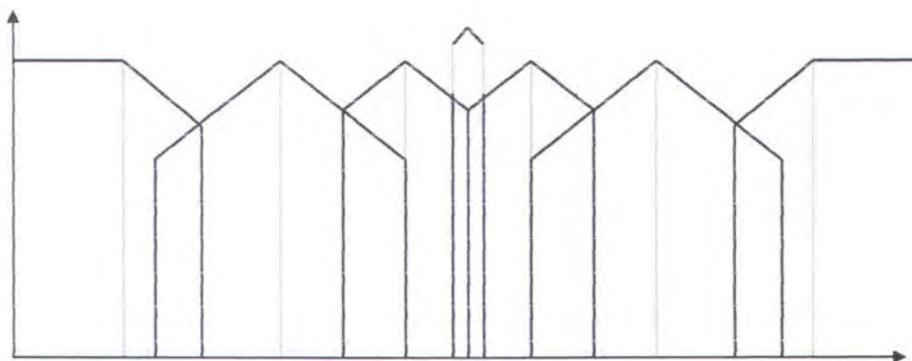
Di dalam aplikasinya didefinisikan dengan nama, tipe bentukannya, dan nilai numerik center dan width-nya. Pemilihan fungsi keanggotaan harus hati-hati

agar dapat menyederhanakan banyak model. Misalnya, dalam termometer Dingin adalah left inclusive dan Panas adalah right Inclusive MF.



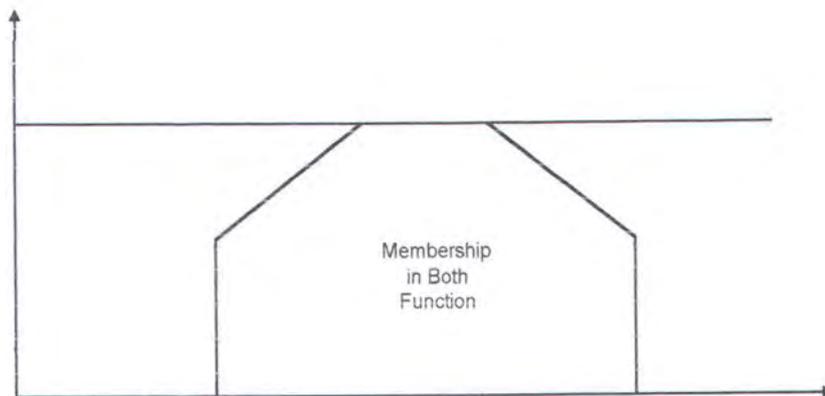
Gambar 2.10. Tipe Fungsi Keanggotaan

Untuk pengontrolan yang teliti pada titik operasi yang diinginkan dapat dibuat fungsi keanggotaan inclusive simetrik yang sempit. Contoh aplikasinya kontrol motor, yang perlu sekali kepresisian. Sebuah contoh dari gabungan berbagai tipe yang berbeda lebar fungsinya digunakan untuk memonitor kecepatan motor seperti gambar 2.11.



Gambar 2.11. Fungsi Keanggotaan Kecepatan

Fungsi Keanggotaan tersebut dapat di-overlap-kan agar membentuk tipe baru seperti trapezoidal seperti pada gambar 2.12. Trapezium tersebut dibentuk oleh adanya fungsi keanggotaan gabungan dari Left Inclusive dan Right Inclusive. Data input dapat masuk ke dalam tipe trapesium yang merupakan dua fungsi keanggotaan.



Gambar 2.12 Fungsi Keanggotaan Overlap

2.4.5.3 Variabel Fuzzy

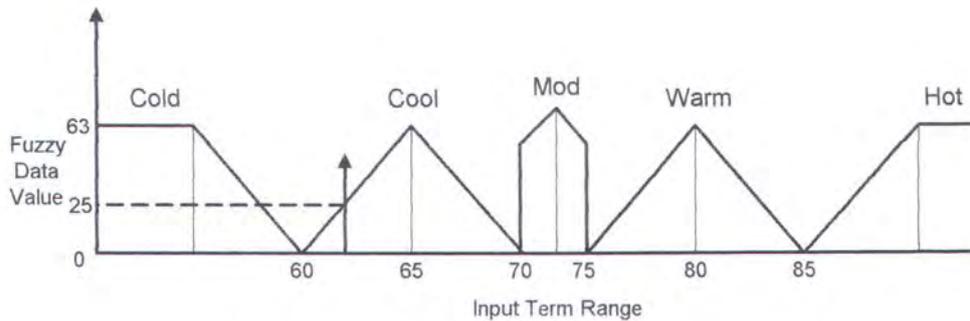
Variabel fuzzy adalah pernyataan bahasa yang menggambarkan hubungan antara nilai input terhadap fungsi keanggotaan yang meliputi seluruh sumbu. Variabel fuzzy mereferensi sebuah fungsi keanggotaan dan sebuah nilai variabel input. Sebuah nilai variabel fuzzy sebagai berikut :

if Temperatur is Cool

Di dalam contoh ini 'Temperatur' adalah input dan 'Cool' adalah sebuah fungsi keanggotaan.

Hubungannya dikerjakan oleh Fuzzifier, hasilnya adalah data Fuzzy yang menunjukkan derajat mana data input yang sesuai dengan fungsi keanggotaan. Data Fuzzy adalah numerik dan berkisar antara 0 sampai 63 di dalam NLX220.

Gambar 2.13. merupakan contoh evaluasi variabel fuzzy dimana terdapat suatu input kemudian nilai fuzzinya dilihat pada sumbu vertikal.



Gambar 2.13. Fuzzifikasi Temperatur Input

2.4.5.4 Rule

Rule terdiri dari satu atau lebih variabel Fuzzy dan sebuah nilai aksi ke outputnya. *Rule* dipakai untuk menggambarkan bagaimana kontroler harus bereaksi jika terjadi perubahan data input. Dalam sebuah contoh dibawah, kedua *Rule* berisi dua variabel fuzzy. *Rule* dimasukkan ke software *INSIGHT* dengan format sebagai berikut :

Output -5 if Velocity is Fast and Acceleration is Positive

Output +5 if Velocity is Little_Slow and Acceleration is Zero

pada *rule* pertama, variabel fuzzy pertama adalah 'Velocity is Fast' dan variabel fuzzy kedua adalah 'Acceleration is Positive'. Aksi '-5' dan '+5' diberikan ke output untuk mengurangi atau mempercepat motor. Jika memakai tanda '±' berarti memakai mode output accumulate yang menunjukkan bahwa output bisa ditambah atau dikurangi.

2.4.5.5 Evaluasi Rule

Ada beberapa metode untuk mengevaluasi *Rule* Fuzzy Logika fuzzy. NLX220 mengevaluasi *Rule* menggunakan teknik dua langkah MAX-of-MIN. Langkah pertama - MIN, semua nilai untuk variabel Fuzzy dalam *Rule* dibandingkan dan nilai paling rendah akan mewakili nilai *Rule* tersebut. Pada step kedua - MAX, nilai satu *Rule* dibandingkan dengan *Rule* yang lain dan *Rule* dengan nilai tertinggi yang menjadi pemenang dan nilai aksinya akan dikeluarkan pada outputnya.

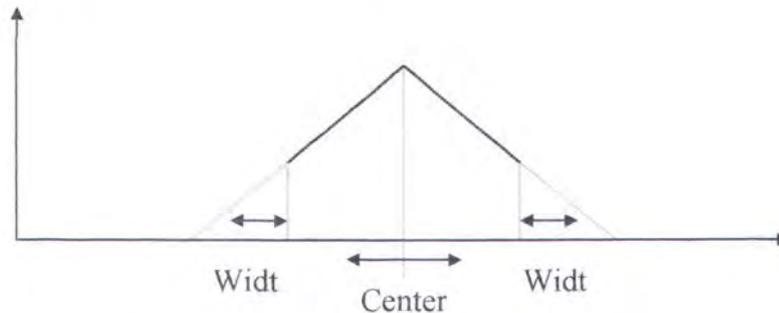
Cara atau metode penentuan fungsi keanggotaan, variabel fuzzy dan *Rule-Rule* yang didefinisikan dan diorganisasi bergantung pada keperluan. Sifat fisik sistem yang dikendalikan harus benar-benar dimengerti sebelum memasukkan model fuzzy. Namun dengan banyaknya pengalaman maka memasukkan sebuah model akan lebih mudah.

Membership function, variabel Fuzzy, dan *Rule* dibuat dan dikelompokkan menurut keperluan aplikasi. Sifat-sifat fisik sistem yang akan dikontrol harus dipahami sebelum memasukkan model Fuzzy.

2.4.5.6 Floating Membership Function

Rancangan unik dari NLX220 adalah fungsi keanggotaan *Floating* (mengambang). Seperti ditunjukkan pada gambar 2.14 di bawah ini, fungsi keanggotaan mengambang mempunyai nilai center dan Width yang bervariasi secara dinamik. Pada fungsi keanggotaan biasa nilai *Center* dan *Width* tetap (fixed) yang disimpan di dalam memori, sedangkan fungsi keanggotaan mengambang nilai *center* atau *width-nya* berasal dari suatu input atau output.

Suatu fungsi keanggotaan dapat dipilih sebagai *floating* pada saat memasukkan rancangan. Fungsi keanggotaan mengambang akan berubah nilai *center* dan *widthnya* sesuai dengan data dari nilai input atau outputnya.



Gambar 2.14. Fungsi Keanggotaan Mengambang

Sebagai contoh, dua variabel fuzzy dengan fungsi keanggotaannya yang didefinisikan secara konvensional dengan menggunakan Rule sebagai berikut:

IN1 is small (0, 25, Symmetrical Inclusive)

IN2 is small (0, 25, Symmetrical Inclusive)

dimana angka pertama adalah nol menunjukkan nilai Center dan yang kedua, 25 adalah Widthnya. Dua variabel tersebut dapat digabungkan menjadi sebuah rule dengan menggunakan fungsi floating sebagai berikut:

IN1 is small_difference (IN2, 25, symmetrical Exclusive)

Output +1 if IN1 is small_difference

Dimana variabel fuzzy 'IN1 is small' membandingkan input IN1 terhadap fungsi keanggotaan konvensional 'small'. Fungsi keanggotaan mengambang dapat memberi penjelasan yang sama namun lebih ringkas.

Di dalam variabel Fuzzy, fungsi keanggotaan center *small_difference* didefinisikan oleh nilai IN2 yang disimpan dalam latch input. Saat proses *Fuzzifikasi*, sebuah input dikurangkan dari center dan hasilnya dirubah untuk

mengukur nilai absolut seberapa dekat nilai tersebut terhadap center. Jika menggunakan sebuah fungsi keanggotaan *Floating Center* maka akan dikurangkan satu input dari yang lainnya. Fungsi keanggotaan *mengambang* (*floating*) memungkinkan menggunakan variabel fuzzy secara langsung mengukur perbedaan antara dua input. Teknik ini dapat digunakan untuk mengkalibrasi perubahan sebuah sensor. Nilai sensor yang tetap dibandingkan terhadap himpunan tegangan. Rule-rule kalibrasi mengecek derajat ketidaksetaraan dan menyimpan nilai koreksi di sebuah latch output. Jika inputnya dalam keadaan kalibrasi, center tersebut akan sesuai dan nilai koreksinya nol. Ketidaksetaraan yang besar akan menyimpan nilai koreksi yang besar. Koreksi digunakan untuk mengatur fungsi keanggotaan *Floating Center* dalam Rule-Rule yang memproses data.

Fungsi keanggotaan *Floating* dapat digabungkan atau dikombinasi dengan nilai output aksi *Floating* untuk memperoleh derivative dari sebuah nilai input. Rule dapat mereferensi sebuah input sehingga akan langsung dilewatkan ke output. Pada sampling input berikutnya, nilai latch output memilih nilai Center, yang mengurangkan nilai input sebelumnya dari nilai yang sekarang (*current value*). Perbedaan atau selisih itu dibagi-bagi melalui interval sampling sehingga merupakan nilai derivative. Sebagai contoh menggunakan sebuah nilai aksi/input yang akan mengukur percepatan motor. Sebuah rule yang menyimpan sebuah input ke latch output dapat ditulis sebagai berikut:

$$VALUE_T0 = IN1 \text{ if } IN1 \text{ is } MUST_WIN (0, 0, \text{Right Inclusive})$$

Rule akan menyimpan IN1 sebagai nilai aksi . Fungsi keanggotaan *MUST_WIN* adalah tipe Right Inclusive yang dimulai dari nol sedemikian rupa dengan tidak

memandang nilai IN1, Rule tersebut harus menang (win) dan nilai IN1 disimpan di output latch. Rule kedua menghitung derivatif dan mengatur output yang menggerakkan motor .

ACCEL ± if IN1 is VALUE_T1 (VALUE_T0, 25, Symmetrical Inclusive)

Rule ini menentukan apakah nilai input pada T1 masih berada di dalam range 25 dari nilai awal saat T0. Di dalam aplikasi nyata, ada fungsi keanggotaan lain yang menentukan polaritas derivative dan Rule-Rule lain untuk mengatasi pengaturan dengan variasi yang lebar. Contoh di atas adalah fungsi keanggotaan floating yang langsung (*strightfoward*) . Dalam aplikasi nyata, fungsi keanggotaan floating secara instensif untuk menghemat memory karena hanya menggunakan beberapa variabel fuzzy dan Rule-Rule mendeteksi perbedaan antara input-input dibanding yang dilakukan fungsi konvensional .

2.4.6 Operasional Perangkat

Pemrosesan data meliputi beberapa langkah. Pertama, data sampel analog dikoversikan ke digital dan ditahan (dilatch). Berikutnya Fuzzifier membandingkan isi dari input latch dengan variabel fuzzy untuk menemukan nilai variabel fuzzy. Fuzifier juga melakukan kalkulasi MAX-of-MIN untuk menentukan Rule yang menang. Akhirnya, Defuzzifier menentukan nilai aksi rule pemenang dan menahannya untuk mengkonversi menjadi output analog atau untuk feedback internal.

2.4.6.1 Fuzzifier

Fuzzifier membandingkan data output yang ditahan dengan fungsi keanggotaan untuk menghitung nilai variabel fuzzy. Ketika penghitungan MIN rule telah dilakukan terhadap semua variabel fuzzy dalam sebuah Rule, nilai yang mewakili Rule disimpan. Ketika penghitungan MAX telah dilakukan pada seluruh variabel fuzzy yang mereferensikan nilai output, maka nilai aksi rule pemenang akan dilewatkan ke Defuzzifier.

2.4.6.2 Pembaharuan Data Output Yang di Latch

Rule-Rule untuk setiap output dievaluasi dan ditentukan pemenangnya. Setiap output dapat menggunakan satu atau beberapa buah rule. Ketika Rule atau sekelompok rule mempengaruhi sekelompok output yang telah dievaluasi dan Rule berikutnya dievaluasi untuk output yang lain, maka kompiler secara otomatis menyisipkan kode untuk Rule terakhir (Last Rule) menyebabkan output yang dilatch diperbaharui oleh aksi rule pemenang. Data yang telah dilatch dapat segera dipakai sebagai internal feedback. dengan output latch untuk di-update dengan nilai pemenang yang baru. Latch data juga bisa dengan cepat dipakai sebagai feedback.

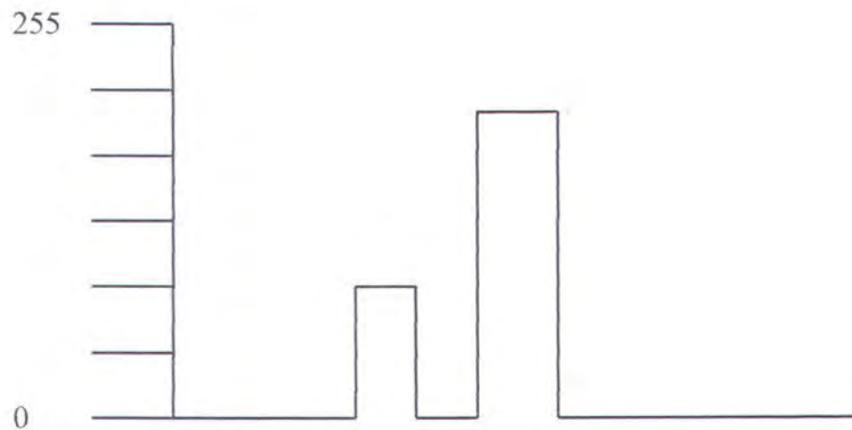
Sebuah data latch output kemudian dapat diperbaharui (di update) selama masih ada sekelompok Rule yang terpisah yang menggunakan output tersebut. Seperti yang telah disinggung sebelumnya, sampling input berlangsung kontinyu. Nilai output analog juga di update secara kontinyu. Selama siklus pemrosesan, variabel Fuzzy dapat menggunakan sebuah sample data dari siklus sampel sebelumnya atau dari siklus yang sedang berlangsung (current) bergantung

dimana letak siklus sampling relatif terhadap siklus pemrosesan. Akan ada lebih dari sekelompok rule yang menggunakan input dan output yang sama, kemudian nilai output dapat berubah lebih dari satu kali selama siklus pemrosesan berdasarkan data input yang berbeda.

2.4.6.3 Defuzzifier

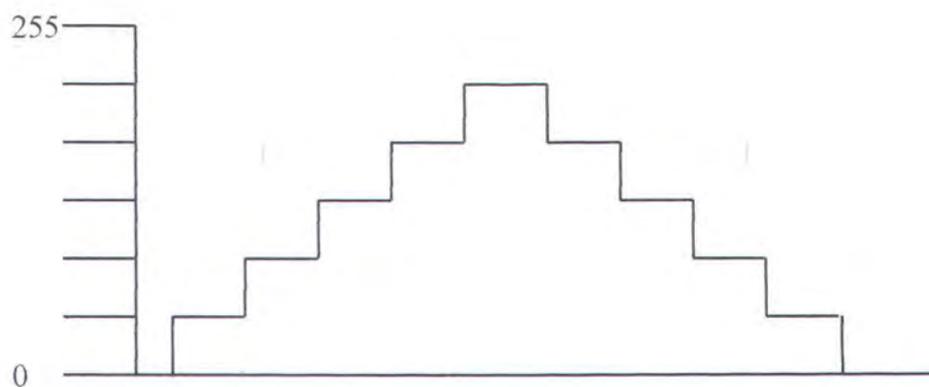
Pada bagian ini data output yang bernilai fuzzy diubah menjadi data crisp. Nilai aksi rule yang menang dan mode data diberikan ke defuzzifier. Data digital dari defuzzifier di-latch dan dikonversikan ke analog untuk mendrive output atau diinputkan kembali secara internal (looped back internally). Jika semua kelompok rule menghasilkan sebuah output yang bernilai nol (zero), maka output tidak akan nilainya. Jika lebih dari satu rule menghasilkan suatu nilai bukan nol yang sama, maka rule yang pertama masuk yang akan menang dan aksinya akan menentukan output.

Defuzzifikasi menyebabkan nilai aksi rule pemenang mendrive (menggerakkan) sebuah output. Ada dua metode defuzzifikasi, Immediate dan Accumulate. Dua mode tersebut terlihat pada gambar 2.15 dan 2.16 dalam menyeleksi rule. Fungsi model Immediate seperti suatu susunan tabel, dimana nilai aksinya tertentu untuk sebuah rule. Defuzzifikasi Immediate sangat berguna jika nilai outputnya harus absolut.



Gambar 2.15. Defuzzifikasi Immediate

Mode accumulate adalah untuk menaikkan atau menurunkan nilai output yang ada dengan nilai pemenang rule. Output adalah fungsi aksi yang sedang berlangsung ditambah atau dikurangi output sebelumnya. *Defuzzifikasi* akumulasi dapat digunakan untuk perubahan pada perubahan halus pada output ketika sistem dikendalikan dekat titik operasi yang diinginkan. Hal ini juga berguna untuk fungsi-fungsi waktu.



Gambar 2.16. Defuzzifikasi akumulasi

2.4.7 Organisasi Memori

NLX220 berisi 256 byte memory untuk aplikasi penyimpanan parameter. 32 byte terakhir menyimpan nilai fungsi keanggotaan tetap (fixed member function) Center dan Width. Sisanya 224 byte diorganisasikan sebagai satu atau lebih rule dengan satu atau lebih variabel fuzzy per rule. Setiap rule memerlukan dua byte, plus dua byte tambahan untuk setiap variabel fuzzy dalam rule itu. Sebuah rule berisi lima variabel fuzzy untuk contoh akan menggunakan 12 byte. Memory diorganisasikan menjadi tiga seksi yang didefinisikan sebagai penyimpan variabel fuzzy/rule, penyimpan Center dan penyimpan Width.

Tabel 2.4 Organisasi Memory

Alamat (Desimal)	(Alamat Hexadesimal)	Fungsi
0	00	Rule
-----	-----	-----
223	DF	Rule
224	E0	Center
-----	-----	-----
239	EF	Center
240	F0	Width
-----	-----	-----
225	FF	Width

2.4.7.1 Rule dan Penyimpan Variabel Fuzzy

Rule diorganisasikan sebagai sebuah atau lebih kelompok variabel-variabel fuzzy. Setiap variabel fuzzy terdiri dari dua byte, seperti dijelaskan dalam tabel 3.5 dan tabel 3.6. Byte pertama disimpan pada alamat genap dan yang kedua pada alamat ganjil. Byte-byte tersebut dibagi-bagi menjadi bidang-bidang yang mengendalikan bagaimana data diproses. Tiga bit LSB untuk byte genap

mendefinisikan tipe fungsi keanggotaan atau apakah variabel fuzzy sebelumnya adalah rule terakhir atau variabel fuzzy terakhir dari rule terakhir yang mereferensi output. Ketika bagian least significant memilih tipe fungsi keanggotaan, lima bit-bit MSB dibagi-bagi menjadi tiga bagian bit yang memilih sumber satu input dari empat pin input atau latch output. Dua sisa bit-bit MSB mendefinisikan baik itu center dan Width dari fungsi keanggotaan floating atau fixed. Tipe kode signal-signal variabel fuzzy terakhir (001) adalah variabel fuzzy terakhir dari rule yang telah diproses. Ketika hal ini terjadi, hanya dua bit-bit MSB dari lima bagian bit yang digunakan. Bit-bit MSB memilih apakah nilai aksi yang berasal dari lokasi memory fixed atau dari latch I/O. Bit-bit MSB berikutnya memilih mode output apakah Immediate atau Accumulate. Kode (000) menunjukkan variabel fuzzy terakhir dari rule terakhir. Dua bit-bit MSB digunakan seperti dijelaskan pada paragraf diatas. Suatu tambahan, dua bit diatas bagian pemilihan tipe digunakan untuk memilih output. Byte kedua selalu terjadi pada alamat ganjil dan berisi alamat Center dan alamat Width bagian Index jika byte sebelumnya dipilih sebagai tipe fungsi keanggotaan dan nilai Center atau Width fixed. Jika salah satu baik itu Center atau width dipilih sebagai floating, maka mereka mengambil byte ganjil digunakan untuk memilih input atau output. Ketika tipe byte pertama adalah variabel fuzzy terakhir atau variabel fuzzy terakhir dari rule terakhir dan aksinya adalah fixed, maka byte kedua mengandung nilai aksi. Jika aksinya adalah floating, maka byte ganjil memilih input atau output yang memberikan nilai aksi.

Tabel 2.5 Commond Byte / Alamat Ganjil

7	6	5	4	3	2	1	0
center		select		width		select	
	I/O cont	I/O select center			I/O cont	I/O select width	
ACTION							
					I/O cont	I/O select action	
Width Select		(0 - 3) Used as address index (E0-EF) for fixed 6-bit value Width when type= 2-7 and WF= 0.					
Center Select		(4 - 7) Used as Address index (F0-FF) for fixed 8-bit CENTER value when type = 2-7 and CF = 0					
I/O Select Width		<u>1 0</u> 0 0 I/O port 0 as Width (Type=2-7 and WF=1) 0 1 I/O port 1 as Width (Type=2-7 and WF=1) 1 0 I/O port 2 as Width (Type=2-7 and WF=1) 1 1 I/O port 3 as Width (Type=2-7 and WF=1)					
I/O Control		<u>2</u> 0 Select from Inputs (Type = 2-7 and WF=1) 1 Select from outputs (Type = 2-7 and WF=1)					
I/O Select Center		<u>5 4</u> 0 0 I/O port 0 as Input (type=2-7 and WF=1) 0 1 I/O port 1 as Input (Type=2-7 and WF=1) 1 0 I/O port 2 as Input (Type=2-7 and WF=1) 1 1 I/O port 3 as Input (Type=2-7 and WF=1)					
I/O Control		<u>6</u> 0 Select from Inputs (Type = 2-7 and WF=1) 1 Select from outputs (Type = 2-7 and WF=1)					
ACTION		(0 - 7) 8-Bit Action value to be applied to an output due to winning of Last Term of Last Rule (Type = 1) or Last Term of Last Rule of given output (Type = 0), and AF =0 (Fixed)					
I/O Select Action		<u>1 0</u> 0 0 I/O port 0 as Action (type=0-1 and AF=1) 0 1 I/O port 1 as Action (type=0-1 and AF=1) 1 0 I/O port 2 as Action (type=0-1 and AF=1) 1 1 I/O port 3 as Action (type=0-1 and AF=1)					
I/O Control		<u>2</u> 0 Select from Inputs (Type = 0 - 1 and WF=1) 1 Select from outputs (Type = 0 - 1 and WF=1)					

Tabel 2.6 Commond Byte / Alamat Genap

7	6	5	4	3	2	1	0
WF	CF	I/O cont	I/O select		type	2-7	
AF	Mode				type	1	
AF	Mode		output select		type	0	
Type		<u>2</u> <u>1</u> <u>0</u>	0 0 0 Last Term of last Rule of given output 0 0 1 Last Term of Current Rule 0 1 0 MF, Symmetrical, Inclusive 0 1 1 MF, Symmetrical, Exclusive 1 0 0 MF, Left, Inclusive 1 0 1 MF, Left, Exclusive 1 1 0 MF, Right, Inclusive 1 1 1 MF, Right, Exclusive				
I/O Select		<u>4</u> <u>3</u>	0 0 I/O port 0 as Input 0 1 I/O port 1 as Input 1 0 I/O port 2 as Input 1 1 I/O port 3 as Input				
I/O Control		<u>5</u>	0 Select from Inputs 1 Select from outputs				
Mode		<u>6</u>	0 Immediate, Output equals Action 1 Accumulate, Output equals current output plus two's complement Action (-128 to +127)				
AF		<u>7</u>	0 Select Action from select Byte (FIXED) 1 Select Action from I/O via select Byte (FLOAT)				
Ouput Select		<u>4</u> <u>3</u>	0 0 ACTION from current RULE set to Output 0 0 1 ACTION from current RULE set to Output 1 1 0 ACTION from current RULE set to Output 2 1 1 ACTION from current RULE set to Output 3				
CF		<u>6</u>	0 Select Center from memory via Select byte (FIXED) 1 Select Center from I/O via Select byte (FLOAT)				
WF		<u>7</u>	0 Select Width from memory via Select byte (FIXED) 1 Select Width from I/O via Select byte (FLOAT)				

2.4.8 Timming (pewaktuan)

Gambar 3.11 menunjukkan pewaktuan NLX220. Ada tiga blok untuk pewaktuan meliputi pemultiplekan konverter A/D input, controller fuzzy dan pemultiplekan konverter D/A output. Kecepatan pemrosesan adalah fungsi kecepatan clock dan banyaknya clock (1024) yang diperlukan untuk penyamplingan data secara lengkap dan siklus pemrosesan. Kecepatan maksimum clock adalah 10 MHz dan minimum 1MHz.

2.4.8.1 Pewaktuan Operasi

Reset,

Ketika pin *RESET* aktif, semua latch dihapus, output digital ada dalam logika rendah dan output analog bertahan pada levelnya terutama pada saat reset. Jika *RESET* aktif untuk seratus clock atau lebih, input analog akan nol ketika sampling dimulai lagi. Jika *RESET* aktif atau kurang dari seratus clock ada beberapa sisa data yang disampel terakhir yang masih ada pada input analog ketika penyamplingan dimulai lagi. Ketika *RESET* tidak aktif lagi, maka penyamplingan input dimulai lagi selama 1024 clock.

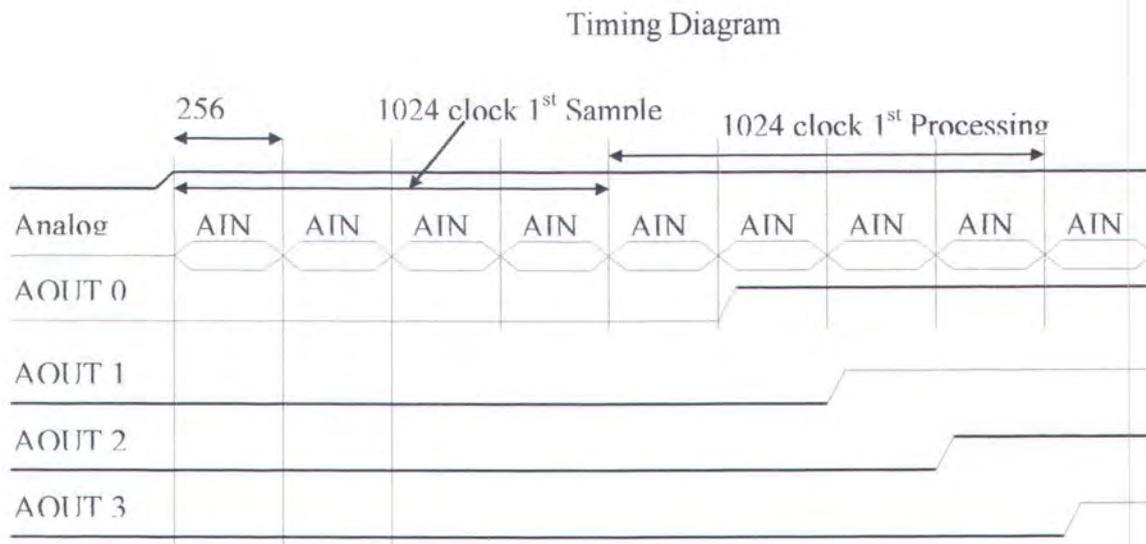
Konversi Input,

Nilai analog input diubah menjadi data digital dan dilatch secara internal dalam periode setiap 256 clock. Jumlah total 1024 clock yang diperlukan untuk mengkonversi empat input setiap pengulangan proses konversi.

Pada clock maksimum, kecepatan penyamplingan untuk setiap input adalah 10 KHz atau 100 μ s.

2.4.8.2 Pewaktuan Internal

Siklus pemrosesan 1024 clock pertama mulai setelah siklus konversi input yang pertama komplit. Siklus pemrosesan terdiri 1024 clock dengan tidak memandang banyaknya variabel fuzzy dan rule yang dipakai. Evaluasi variabel fuzzy dan rule masing-masing memerlukan empat clock. Sebagai contoh, rule dengan dua variabel memerlukan 12 clock untuk pemrosesan. Selama siklus pemrosesan baik variabel fuzzy atau rule diproses setiap empat clock, kecuali 64 clock terakhir pada akhir siklus pemrosesan.



Gambar 2.17. Pewaktuan I/O

2.4.8.3 Waktu Tunda Loopback Internal

Ketika data berada dalam latch output diloopback secara internal sebagai input, maka data tersebut akan mengalami kelambatan sebagai input analog sebesar 1024 clock dari siklus sampling yang pertama. Setelah itu, pada saat latch

output di update selama pemrosesan maka data feedback digunakan sebagai input.

2.4.8.4 Pewaktuan Output

Output di update pada batas 256 byte setelah pemrosesan dimulai seperti terlihat dalam gambar 2.17. Setiap pin output di update sekali setiap 1024 clock. Pewaktuan update output sangat tidak berubah. Latch output di update secara immediate setelah evaluasi Rule yang relevan lengkap.

2.4.8.5 Operasi Prescale

Perangkat ini berisi suatu yang dapat diprogram, yaitu suatu counter prescale 8-bit yang memungkinkan perangkat ini menjadi inaktif untuk periode tertentu. Rancangan ini dapat digunakan pada berbagai kecepatan sampling dan pemrosesan. Lokasi terakhir dalam memory, yang dalam keadaan normal menyimpan data width dari fungsi keanggotaan fixed, dapat menyimpan suatu nilai yang diberikan ke counter. Pin *PRESCALE* memilih operasi normal atau operasi prescale. Dalam mode prescale, kontroller inaktif untuk periode 1024 clock setelah counter ditambah. Ketika counter berjalan, kontroler aktif untuk periode tunggal 1024 clock untuk melakukan komputasi fuzzy dan counter diload lagi. Dalam mengatur scale dengan memasukkan bilangan komplemen dari interval 1024 clock, yang diharapkan diantara siklus sampling dan pemrosesan. Sebagai contoh, dipilih penyamplingan yang dipisahkan dengan 2 (dua) interval, berarti memasukkan bilangan komplemennya adalah FD. Interval-interval prescale dimasukkan pada saat kompilasi disain file. Pin dapat dikondisikan low

ketika tidak digunakan atau dikondisikan ke pin *READY* untuk operasi prescale kontinyu. Logika eksternal juga dapat digunakan untuk mengoperasikan pin yang memungkinkan fungsi prescale dipertahankan atau tidak selama operasi yang berbeda.

2.4.8.6 Mode Inaktif

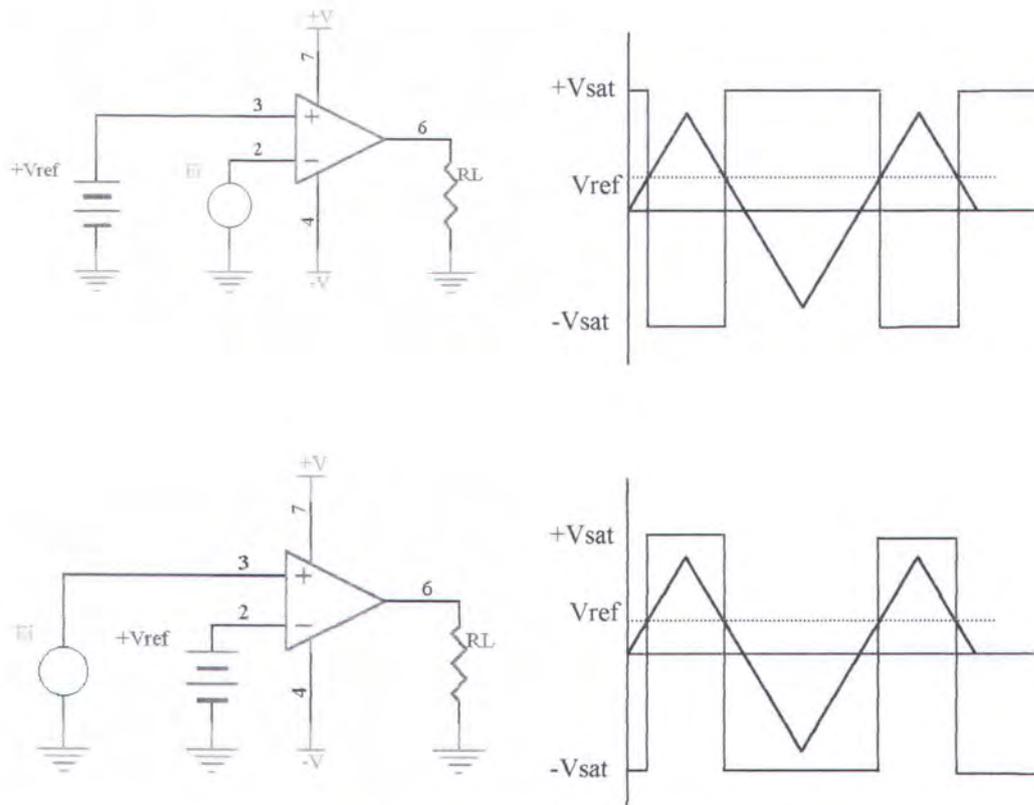
Konsumsi power dapat diperkecil dari mode operasi ke mode standby yaitu dengan mempertahankan pin clock high. Menghentikan clock berarti menunda pemrosesan dan membiarkan output pada kondisi seting terakhir. Nilai output analog akan menjadi nol. Pemrosesan berlanjut lagi ketika clock memulai lagi.

2.5 Sensor

Salah satu bagian yang penting dalam sistem kontrol adalah sensor. Sensor berfungsi untuk mengukur parameter yang diperlukan pada sistem kontrol. Dalam tugas akhir ini parameter yang diukur adalah posisi dan besar sudut benda.

Sensor yang digunakan untuk mendeteksi posisi dan besar sudut tersebut digunakan transducer yang ekonomis tetapi cukup memenuhi kebutuhan yaitu Variabel Resistor (VR) yang non linear. Perubahan resistansinya disebabkan karena perubahan posisi yang mengakibatkan terjadinya perubahan tegangan DC yang merupakan input Fuzzy. Disinilah kehandalan dari logika fuzzy yang dapat melinierkan suatu sensor yang non linear dengan hanya mengatur fungsi keanggotaan (MF) pada programnya.

2.6 Op-amplifier sebagai Positif Level Detector⁹



Gambar 2.18. Positif Level Detektor

Dalam gambar 2.18, tegangan acuan positif V_{ref} diumpangkan pada salah satu masukan op-amp. Ini berarti op-amp tersebut sebagai sebuah pembanding untuk mendeteksi suatu tegangan positif. Jika tegangan yang harus dideteksi, E_i diumpangkan pada masukan non inverting (+) op-amp tersebut maka hasilnya adalah **non-inverting positif detector**. Operasinya diperlihatkan oleh bentuk gelombang dalam gambar. Bila E_i berada diatas V_{ref} , V_o menyamai $+V_{sat}$. Bila E_i dibawah V_{ref} , V_o menyamai $-V_{sat}$. Jika E_i diterapkan pada masukan Inverting, maka pada saat E_i melebihi V_{ref} , V_o menyamai $-V_{sat}$ dan sebaliknya.

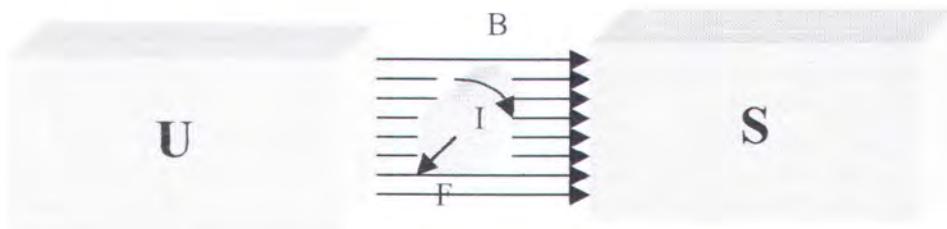
⁹Robert F, Frederick F, OPERATIONAL AMPLIFIER AND LINEAR INTEGRATED CIRCUITS, (PrenticeHall), P.19

2.7 Motor Arus Searah (DC)

Motor arus searah merupakan suatu alat yang berfungsi mengubah daya listrik arus searah menjadi daya mekanik. Motor arus searah mempunyai prinsip kerja berdasarkan percobaan Lorentz¹⁰ yang menyatakan “ Jika sebatang penghantar listrik yang berarus berada di dalam medan magnet maka pada kawat penghantar tersebut akan terbentuk suatu gaya “. Gaya yang terbentuk sering dinamakan **Gaya Lorentz** Untuk menentukan arah gaya dapat digunakan kaidah tangan kiri dan besarnya adalah :

$$F = I(L \times B) \text{ Newton.}$$

$$= I.L.B \sin \theta \text{ Newton.}$$



Gambar 2.19. Prinsip kerja motor dc

Persamaan dari motor dc adalah sebagai berikut :



Gambar 2.20. Persamaan motor dc

¹⁰, TEKNOLOGI MOTOR DAN GENERATOR,, 1988, p.85

$$V = I_a.R_a + E_a$$

$$E_a = N.\phi.C$$

Dimana :

R_a = Tahanan jangkar.

E_a = GGL lawan jangkar.

N = kecepatan putar (rps).

C = konstanta.

ϕ = Flux.

Besarnya daya mekanik adalah :

$$P_m = V.I_a - I_a^2.R_a$$

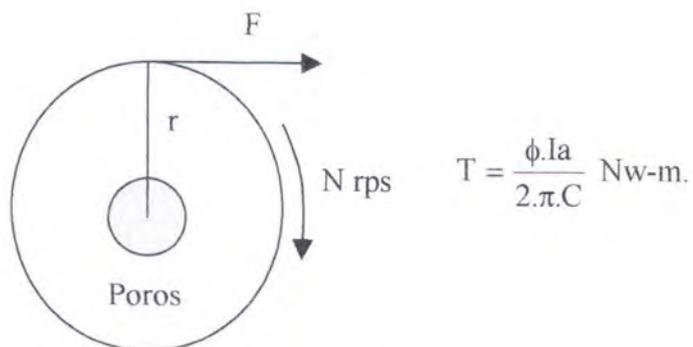
Untuk kondisi daya mekanik maksimum ialah :

$$\frac{P_m}{dI_a} = 0, \quad \text{maka } E_a = I_a.R_a = \frac{V}{2}$$

Dari persamaan tegangan motor maka di dapat kecepatan motor dc adalah :

$$N = \frac{V - I_a.R_a}{\phi.C} \text{ rps}$$

Torsi jangkar motor dc adalah kerja yang dilakukan motor dc dengan kecepatan N rps



Gambar 2.21. Torsi motor dc

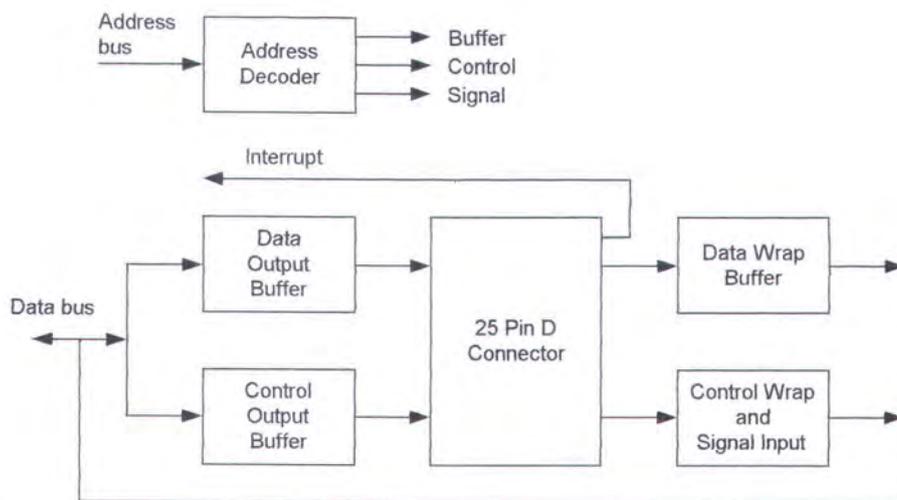
2.8 LPT1 IBM PC

Pada tugas akhir ini proses akuisisi data dilakukan IBM PC melalui port paralel (LPT1) menggunakan metoda HDL.

Port paralel disebut juga adapter paralel merupakan suatu media untuk menghubungkan komputer dengan unit di luarnya. Biasanya setiap PC memiliki sebuah port paralel yang dapat ditambahkan menjadi LPT1 dan LPT2.

Pada umumnya port paralel LPT dihubungkan dengan printer. Selain untuk mengatur kerja printer, penggunaan port paralel sebagai sarana komunikasi dan proses transfer data (input/output) hampir terabaikan. Hal ini disebabkan karena pada dasarnya jalur data yang ada pada port paralel LPT hanya berfungsi sebagai jalur data yang bersifat satu arah (unidirectional), yaitu sebagai jalur data output.

Blok diagram port paralel adalah sebagai berikut :



Gambar 2.22 Blok diagram port paralel⁸

¹¹ ---, "IBM PC AT Technical Reference", IBM, 1984, hal. 20

Pada tugas akhir, port paralel LPT1 akan dimanfaatkan untuk mengatur modul – modul pengukuran serta proses akuisisi datanya. Karena digunakan untuk mengatur kerja suatu alat, maka diperlukan suatu jalur data yang berfungsi sebagai input. Oleh karena jalur data yang disediakan oleh port hanya berfungsi sebagai output, maka sebagai jalur input digunakan jalur sinyal status dan kontrol, yang dapat berfungsi sebagai input atau output.

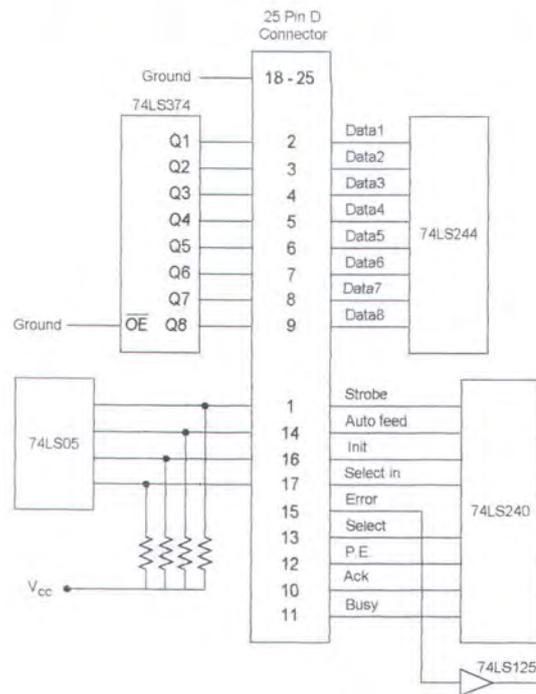
2.8.1 Konfigurasi Pin

Port paralel LPT yang terdapat pada IBM PC terdiri dari sebuah konektor DB25 pin dengan 17 jalur sinyal yang terbagi atas 3 kelompok jalur sinyal dan 8 jalur yang terhubung dengan ground. Adapun 17 jalur sinyal yang terdapat pada port paralel LPT dan terhubung melalui konektor DB25 tersebut adalah :

1. Jalur sinyal kontrol (4 jalur)
2. Jalur sinyal status (5 jalur)
3. Jalur sinyal data (8 jalur)

Jalur kontrol digunakan sebagai jalur pengatur perantara dan proses handshaking sinyal dari PC ke printer. Jalur status digunakan untuk sinyal handshake dan indikator status, seperti kertas habis (paper-end) dan sibuk (busy). Jalur data (D0 – D7) digunakan untuk memindahkan data dari PC ke printer.

Deskripsi masing-masing pin ditunjukkan pada gambar berikut :



Gambar 2.23 Sinyal input dan output dari port paralel¹²

2.8.2 Alamat Memori

Pada saat kita menghubungkan IBM PC dengan suatu peralatan luar melalui port paralel LPT, maka pertama kali kita perlu mengetahui alamat memori yang digunakan oleh port paralel LPT. Terdapat 3 alamat memori (base address) yang digunakan oleh port paralel pada komputer, yaitu¹³:

1. 956 atau 3BCh, yaitu alamat port dengan MBA Card
2. 888 atau 378h, yaitu alamat port dengan CGA Card
3. 632 atau 278h, yaitu alamat port 3ed

¹² Ibid

¹³ Paul Bergsman. "Micocomputer Journal", Januari 1995, hal. 30

Sedangkan pendefinisian alamat dasar (base address) pada pin-pin port paralel LPT ditunjukkan pada tabel berikut ini.

Tabel 2.7 Alamat Memori Pin Pada Port Printer

Pin	Alamat Memori (Base Address)
1	D0, Base + 2, Bit 1
2	D0, Base Address, Bit 1
3	D1, Base Address, Bit 2
4	D2, Base Address, Bit 3
5	D3, Base Address, Bit 4
6	D4, Base Address, Bit 5
7	D5, Base Address, Bit 6
8	D6, Base Address, Bit 7
9	D7, Base Address, Bit 8
10	D6, Base + 1, Bit 7
11	D7, Base + 1, Bit 8
12	D5, Base + 1, Bit 6
13	D4, Base + 1, Bit 5
14	D1, Base + 2, Bit 2
15	D3, Base + 1, Bit 4
16	D2, Base + 2, Bit 3
17	D3, Base + 2, Bit 4

Alamat-alamat pada tabel kesemuanya harus didefinisikan dan diatur secara program pada saat kita akan mengakses pin yang bersangkutan.

Sebagai gambaran, misalnya digunakan base address 378h, maka penggunaan memori ini akan mengatur fungsi-fungsi sebagaimana yang ditunjukkan pada gambar, adalah sebagai berikut :

- Port 378h berupa IC 74LS374 (output) dan IC 74LS244 (input). IC 74LS374 mampu memberikan logika 1 (source) hingga 2,6 mA dan logika 0 (sink) hingga 24 mA. Port ini sebenarnya bisa sebagai input maupun output, tetapi karena pin OE disambung permanen ke ground, maka port ini tidak bisa sebagai input. Port hanya dapat mengirim sinyal ke luar. Karena terdapat IC 74LS244, port dapat dibaca oleh komputer. Dalam hal ini komputer membaca output 74LS374, jadi komputer membaca dirinya sendiri.

- Port 379h hanya berfungsi sebagai input. Normalnya port ini membaca status printer, yaitu sinyal Error, SLCT, Paper End, Acknowledge dan Busy. Jadi hanya data bit ke 3 sampai bit ke 7 saja yang dipakai.
- Port 37Ah berupa IC 74LS05 (output open collector) dan IC 74LS240 (input), dimana output 74LS05 berhubungan dengan input 74LS240. IC 74LS05 mampu memberikan logika 0 hingga 7 mA (dengan R pull-up sebesar 4,7 k Ω). Karena output dari 74LS05 bersifat open collector, maka port ini dapat bertindak sebagai input maupun output. Agar tidak merusak sinyal dari luar, terlebih dahulu dikirim data 0 ke alamat ini dan diterima sebagai data 1 sehingga ketika diberikan data luar 0 atau 1 maka data tersebut mengikuti data luar. Khusus untuk bit 2 memakai dua buah Not open collector, sehingga dikeluarkan dengan data 1. Jadi untuk menjadi port ini sebagai input, terlebih dahulu dikeluarkan data 0000 0100b atau 04h ke port ini.

2.9 ADC MAX 155

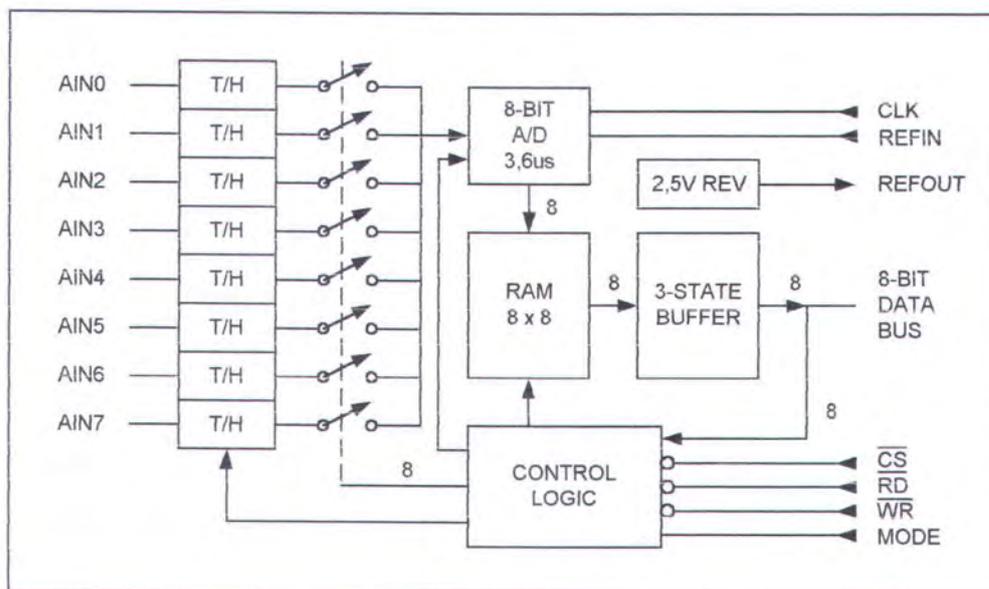
MAX155 adalah pengubah analog ke digital kecepatan tinggi tipe *successive approximation*, 8-bit, serta memiliki 8 kanal input dengan track/hold (T/H) simultan untuk mengeliminasi perbedaan waktu di antara sampel tiap kanal inputnya. Setiap kanal memiliki T/H terpisah. ADC mengkonversi tiap kanal dalam 3,6 μ s dan menyimpan hasilnya pada RAM internal 8 x 8. Selain itu MAX155 juga memiliki referensi internal 2,5V dan kemampuan power down untuk keperluan sistem sampling akuisisi data yang lengkap.

Pada operasi dengan supply tunggal +5V, MAX155 dapat bekerja pada mode unipolar atau bipolar, konversi input tunggal (single-ended) atau diferensial.

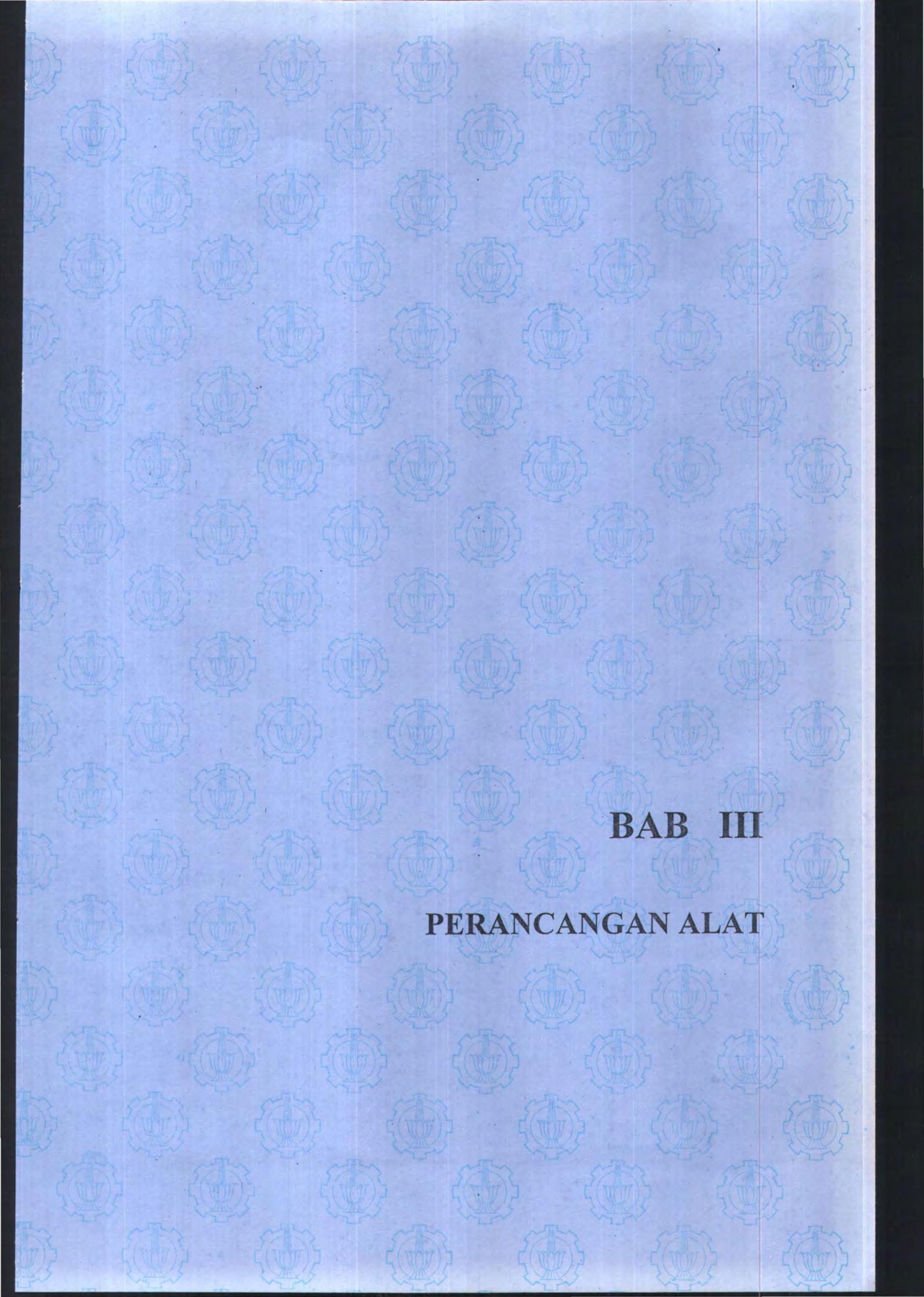
Pada pemakaian range dinamis yang lebih lebar atau konversi bipolar di sekitar ground, terminal V_{SS} harus diberi tegangan supply $-5V$.

Konversi ADC ditandai dengan pulsa rendah pada pin WR, dan data diakses dari RAM ADC dengan pulsa ke pin RD. Interface dua arah (bidirectional) memperbaharui konfigurasi kanal dan menyediakan data output.

Diagram fungsi ADC MAX155 ditunjukkan pada gambar berikut ini :



Gambar 2.24 Diagram fungsi IC MAX155



BAB III

PERANCANGAN ALAT

BAB III

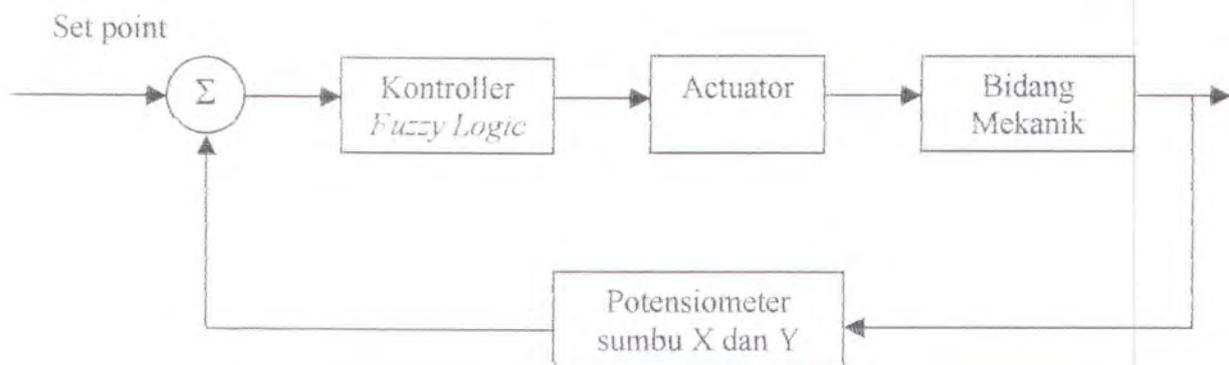
PERANCANGAN ALAT

Setelah mempelajari teori-teori tentang sistem kontrol dan teori logika fuzzy, dalam bab ini akan dibahas tentang perencanaan mekanik, perangkat keras dan perangkat lunak yang dibuat adalah :

- Merancang diagram blok sistem secara keseluruhan dan cara kerja secara umum.
- Merancang bentuk mekanik sesuai yang diharapkan.
- Merancang rangkaian elektronika tiap blok sesuai dengan yang dibutuhkan.
- Merancang perangkat lunak untuk menunjang kerja dari sistem yang dibuat.

3.1 Perancangan Sistem

Diagram blok sistem pengendalian posisi benda terhadap arah gravitasi bumi ini ditunjukkan pada gambar 3.1.

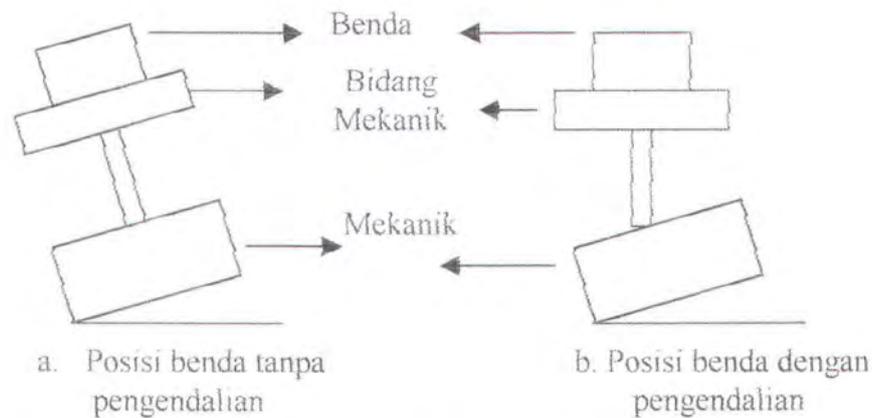


Gambar 3.1 Diagram Blok Pengendalian posisi benda terhadap arah gravitasi bumi.

Pada blok diagram rangkaian diatas terdapat empat blok yaitu blok *Kontroller Fuzzy Logic*, blok actuator yaitu driver kecepatan dan arah gerak motor, blok mekanik serta blok potensiometer sebagai tranducer kemiringan bidang pada mekanik.

Cara kerja umum tugas akhir ini adalah apabila mekanik tersebut miring, maka sensor posisi akan mendeteksi kemiringan tersebut kemudian sinyal dari sensor tersebut akan diberikan pada kontroller. Kontroller tersebut akan memberikan respon dengan menggerakkan arah motor yang berlawanan dengan kemiringan mekanik serta kecepatan motor akan berdasarkan posisi kemiringan bidang pada mekanik tersebut, sehingga benda yang diatas bidang mekanik tersebut diharapkan tidak ikut miring. Kecepatan motor untuk sumbu Y selalu digunakan kecepatan tinggi karena motor membawa beban motor yang bergerak ke sumbu X sedangkan kecepatan motor untuk sumbu X terbagi dalam tiga kecepatan yaitu:

- Kecepatan rendah apabila besar sudut kemiringan bidang mekanik adalah 0 s/d 30° atau 0 s/d -30° .
- Kecepatan sedang apabila besar sudut kemiringan bidang mekanik adalah 30 s/d 60° atau -30 s/d -60° .
- Kecepatan tinggi apabila besar sudut kemiringan bidang mekanik adalah 60 s/d 90° atau -60 s/d -90° .



Gambar 3.2 Perbedaan posisi benda terhadap arah gravitasi bumi.

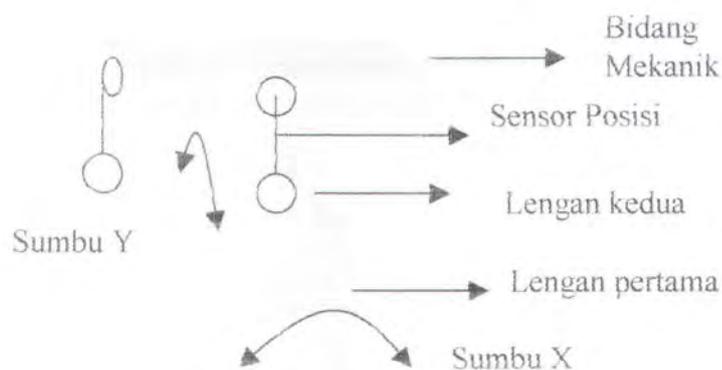
Fungsi dari masing-masing blok adalah sebagai berikut .

1. Kontroller Fuzzy Logic berfungsi untuk mengatur arah putaran dan kecepatan motor DC terhadap sinyal yang diberikan oleh sensor posisi yaitu potensiometer sumbu X dan Y.
2. Actuator merupakan driver untuk arah putaran dan kecepatan motor DC.
3. Bidang Mekanik merupakan bagian mekanik yang akan digunakan serta akan diuji.
4. Potensiometer yang merupakan sensor posisi yang berfungsi untuk mendeteksi arah kemiringan mekanik dan perubahan besar sudut kemiringan bidang mekanik. Sensor posisi ini terdapat 2 buah yaitu mendeteksi kemiringan mekanik pada posisi sumbu X dan sumbu Y.

3.2 Perancangan Mekanik

Pada perancangan mekanik tersebut dibuat dengan bahan aluminium karena sifat aluminium yang ringan tetapi cukup kuat. Mekanik tersebut terdiri dari dua buah motor DC untuk menggerakkan lengan terhadap sumbu X dan

sumbu Y. Lengan pertama yang digerakkan oleh motor pertama berfungsi untuk menjadikan bidang mekanik dapat berputar terhadap sumbu X apabila bidang mekanik miring terhadap sumbu X dari arah gravitasi demikian juga untuk lengan kedua yang digerakkan oleh motor kedua yang digunakan terhadap sumbu Y. Lengan kedua terkait dengan lengan pertama sebagai suatu kesatuan. Untuk gambaran secara umum dapat dilihat pada gambar di bawah ini dan lebih jelasnya dapat dilihat pada lampiran.



Gambar 3.3 Bentuk mekanik secara umum.

Untuk menggerakkan tiap-tiap lengan digunakan motor DC yang nantinya dapat berputar ke kiri maupun ke kanan terhadap sumbu X maupun Y. Pemilihan motor DC ini karena alasan murah, mudah didapat dan torsi yang cukup kuat.

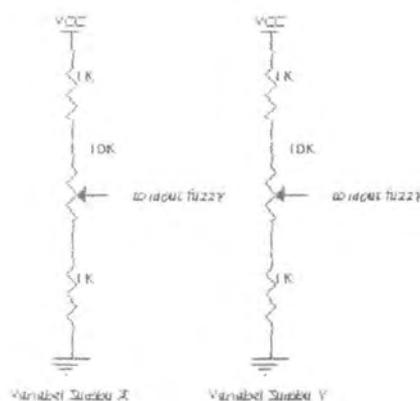
Karena pergerakan lengan nantinya diharapkan pelan atau halus maka diperlukan perubahan putaran yaitu putaran tinggi (motor) menjadi putaran rendah atau dikenal dengan reduksi. Fungsi utama dari reduksi selain melambatkan

putaran motor juga untuk meningkatkan torsi. Perbandingan gigi yang dirancang adalah 1 : 66 yaitu 66 putaran motor akan memutar 1 putaran penuh gigi reduksinya.

3.3 Perancangan Perangkat Keras

3.3.1 Rangkaian Sensor Posisi

Salah satu bagian kontrol sistem yang penting adalah sensor. Sensor berfungsi untuk mengukur parameter yang diperlukan oleh sistem kontrol. Dalam tugas akhir ini hanya terdapat satu parameter yang akan diukur yaitu posisi dari suatu gerakan mekanis. Transducer yang digunakan digunakan VR (variabel resistor) karena transducer tersebut mudah dicari dan harganya ekonomis. Walaupun komponen ini terkadang banyak yang tidak linear, maka disini lah fungsinya logika fuzzy yang dapat melinierkan suatu masukan atau transducer yang non linear.



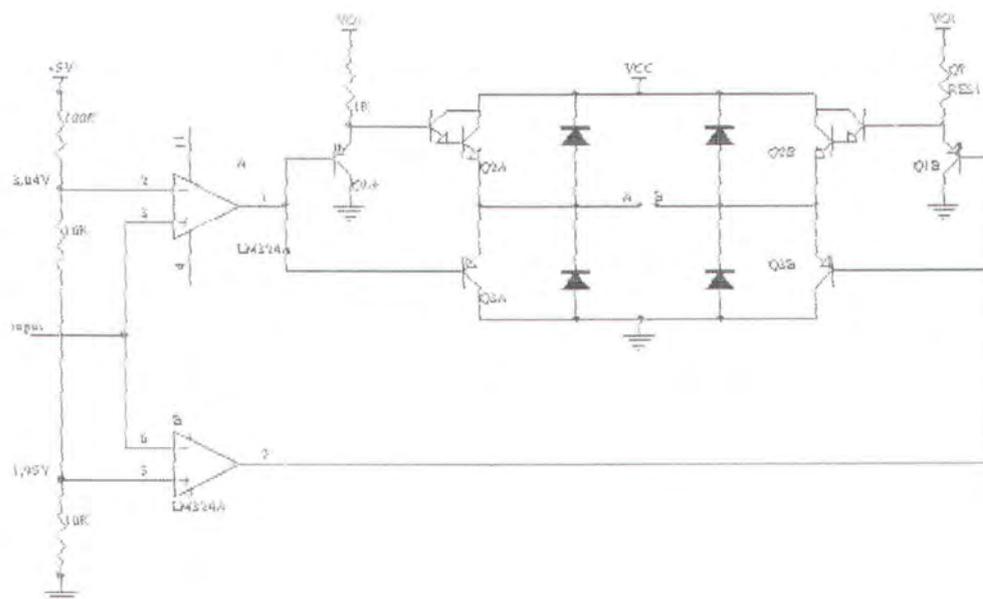
Gambar.3.4 Rangkaian sensor Posisi.

Pada perancangan ini digunakan dua buah sensor untuk mengetahui posisi bidang pada sumbu X dan sumbu Y untuk menentukan arah gerakan dan kecepatan mekanis motor.

Fungsi dari resistor $1K\Omega$ adalah selain sebagai pembatas arus, juga sebagai pengaman bila terjadi short circuit atau kerusakan pada VR-nya agar tidak merusakkan catu dayanya. Jangkauan dari perputaran VR tiap-tiap masukan dari minimum sampai maksimum yaitu dari $0V - 5Vdc$.

3.3.2 Rangkaian Positif Level Detektor.

Rangkaian tersebut digunakan sebagai driver untuk menentukan arah gerakan motor. Motor akan berputar ke kiri apabila input diberikan tegangan 0 s/d 2 Vdc dan akan berputar ke kanan apabila input diberikan tegangan 3 s/d 5 Vdc. Motor akan berhenti apabila input diberikan tegangan 2 s/d 3 Vdc.

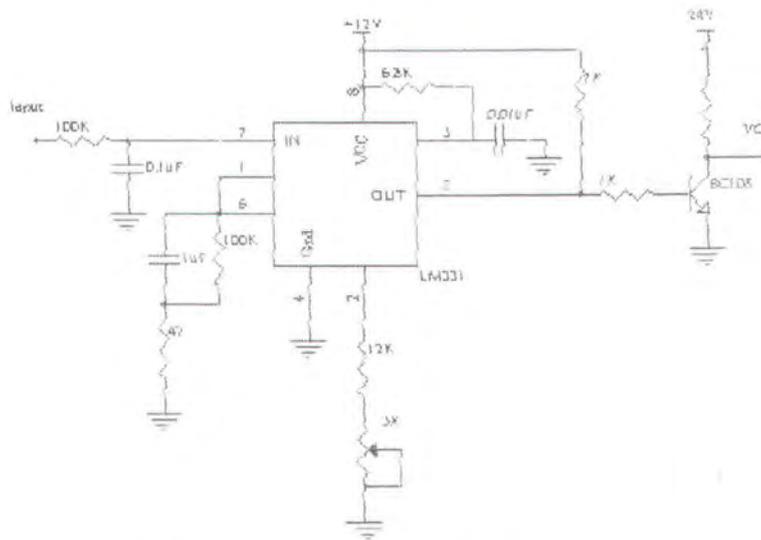


Gambar 3.5 Rangkaian Level Detector (switching)

Op-amp yang digunakan adalah LM324 digunakan sebagai pembanding masukan terhadap tegangan referensinya. Tegangan referensi untuk Op-amp A adalah 3,04 Vdc dan untuk op-amp B adalah 1,95 Vdc. Dan untuk menentukan arah putaran motor DC adalah mengubah polaritas pada keluaran driver tersebut. Apabila input kurang dari 1,95 Vdc maka Op-amp A akan $-V_{sat}$ sedangkan Op-amp B akan $+V_{sat}$. Dengan demikian Q1A, Q3A, Q2B akan OFF sedangkan Q2A dan Q3B akan ON, maka arus mengalir dari titik A ke titik B. Apabila input lebih dari 3,04 Vdc maka Op-amp A akan $+V_{sat}$ sedangkan Op-amp B akan $-V_{sat}$, sehingga akan Q1B, Q3B, Q2A akan OFF sedangkan Q2B dan Q3A akan ON maka arus mengalir dari titik B ke titik A. Dan apabila diberi tegangan diantaranya karena Op-amp A dan B akan $-V_{sat}$ sehingga Q1A, Q3A, Q1B, Q3B akan ON dan Q2A, Q2B akan OFF.

3.3.3 Rangkaian Pengatur Kecepatan Motor DC.

Untuk mengatur kecepatan motor dc maka digunakan rangkaian PWM yang menggunakan IC LM 331 yang berfungsi untuk mengubah tegangan menjadi frekuensi yang besarnya frekuensi tersebut adalah 10 Hz s/d 11 KHz.



Gambar 3.6 Rangkaian Pengatur Kecepatan Motor DC.

Untuk memperbesar arus output maka digunakan rangkaian driver Transistor BC 108 dan rangkaian darlington 2N3055. Tegangan output ditentukan oleh besarnya frekuensi output LM 331 dan besarnya frekuensi output adalah

$$f_{out} = \frac{V_{in}}{2,09V} \cdot \frac{R_s}{R_1} \cdot \frac{1}{R_f \cdot C_f}$$

Dengan berubahnya tegangan output maka akan mengubah kecepatan motor DC tersebut.

3.3.4 ADC MAX155

Dalam tugas akhir ini, ADC MAX155 direncanakan untuk menerima sinyal dari ke-8 buah kanal inputnya pada mode *hard-wired*, dengan perincian:

- Untuk pembacaan sudut X dan Y pada bidang dengan kanal 0 dan 1
- Untuk pembacaan sudut X dan Y pada mekanik dengan kanal 2 dan 3.

Tabel 3.1 Pemilihan Multiplexer mode hard-wired

MODE	V _{SS}	TIPE KONVERSI
Terbuka	X	Register konfigurasi multiplexer menentukan tipe konversi
0	AGND	8 kanal, single-ended, konversi unipolar
1	AGND	4 kanal, diferensial, konversi unipolar
0	-5V	8 kanal, single-ended, konversi bipolar
1	-5V	4 kanal, diferensial, konversi bipolar

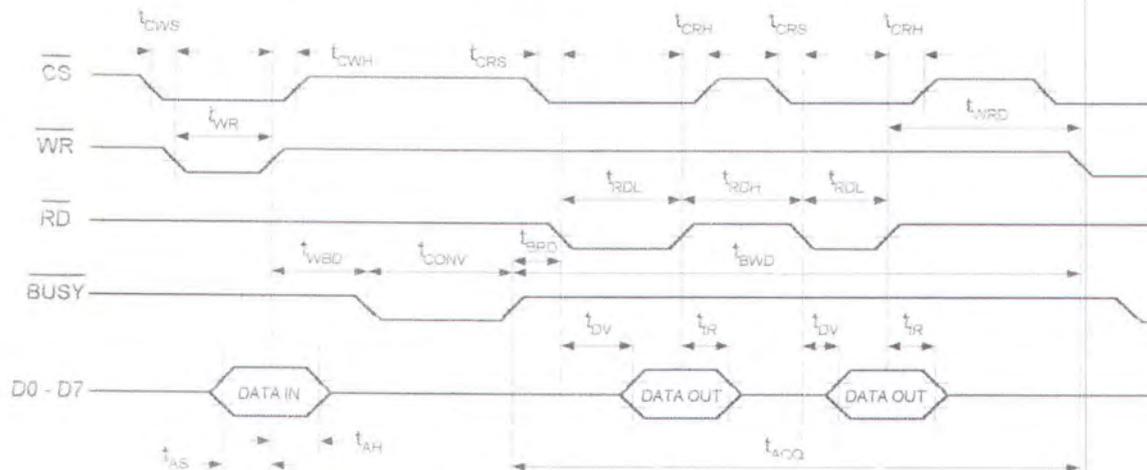
3.3.4.1 Prinsip Kerja

Saat konversi dimulai, semua input AIN secara simultan melakukan proses sampling meskipun dipilih atau tidak untuk proses konversi. Hasilnya akan disimpan dalam RAM internal. Konversi multi-kanal diawali dengan satu operasi tulis.

Untuk mengakses hasil konversi, pulsa RD successive otomatis sekuensial melalui RAM, dimulai dari kanal 0. Tiap pulsa RD akan menaikkan counter alamat RAM, dan kembali reset ke 0 saat WR menjadi rendah pada konversi multi-kanal.

Tabel 3.2 Karakteristik Timing

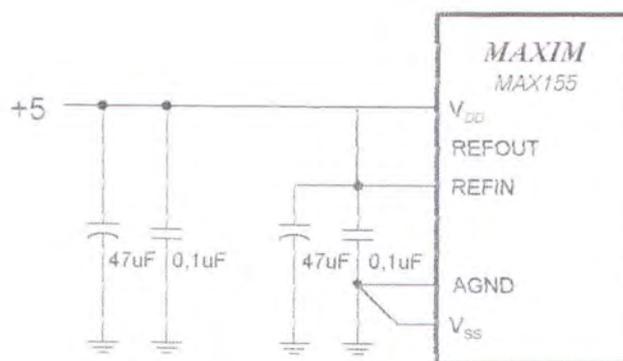
PARAMETER	SIMBOL	MIN (ns)	MAX (ns)
Waktu setup CS ke WR	t _{CWS}	0	-
Waktu hold CS ke WR	t _{CWH}	0	-
Waktu setup CS ke RD	t _{CRS}	0	-
Waktu hold CS ke RD	t _{CRH}	0	-
Lebar pulsa rendah WR	t _{WR}	100	2000
Lebar pulsa rendah RD	t _{RDL}	100	-
Lebar pulsa tinggi RD	t _{RDH}	180	-
Delay WR ke RD	t _{WRD}	280	-
Delay WR ke BUSY rendah	t _{WBD}	-	220
Delay BUSY tinggi ke WR (mode I/O)	t _{BWD}	50	-
Delay CLK ke WR	t _{CCW}	800	-
Delay BUSY tinggi ke RD	t _{BDD}	50	-
Waktu setup address	t _{AS}	120	-
Waktu hold address	t _{AH}	0	-



Gambar 3.7 Diagram waktu tulis dan baca

3.3.4.2 Referensi External

Referensi 5V external (REFOUT) maka REFIN dihubungkan dengan V_{DD} dengan tegangan sebesar 5V dan REFOUT dibiarkan atau dihubungkan dengan capacitor 4,7uF terhadap ground.



Gambar 3.8 Referensi external

3.3.4.3 Bypassing

Kapasitor elektrolit 4,7uF dan keramik 0,1uF digunakan untuk bypass V_{DD} ke AGND. Karena sinyal input dibawah ground diperlukan, supply negatif

dibutuhkan. Di sini, VSS harus dibypass ke AGND menggunakan kapasitor elektrolit 4,7 μ F dan keramik 0,1 μ F.

3.3.4.4 Amplifier Track/Hold

Amplifier T/H MAX155 tidak membutuhkan bufer input. Semua T/H melakukan sample secara simultan. Untuk hasil terbaik, input analog tidak boleh melebihi 50mV dari power-supply rail (VDD, VSS)

Waktu yang dibutuhkan T/H untuk mengambil sinyal input satu kanal adalah sebagai fungsi dari kecepatan kapasitansi kanal input diisi. Jika impedansi sumber sinyal input tinggi, akuisisi akan lebih lama. Waktu akuisisi dihitung dari :

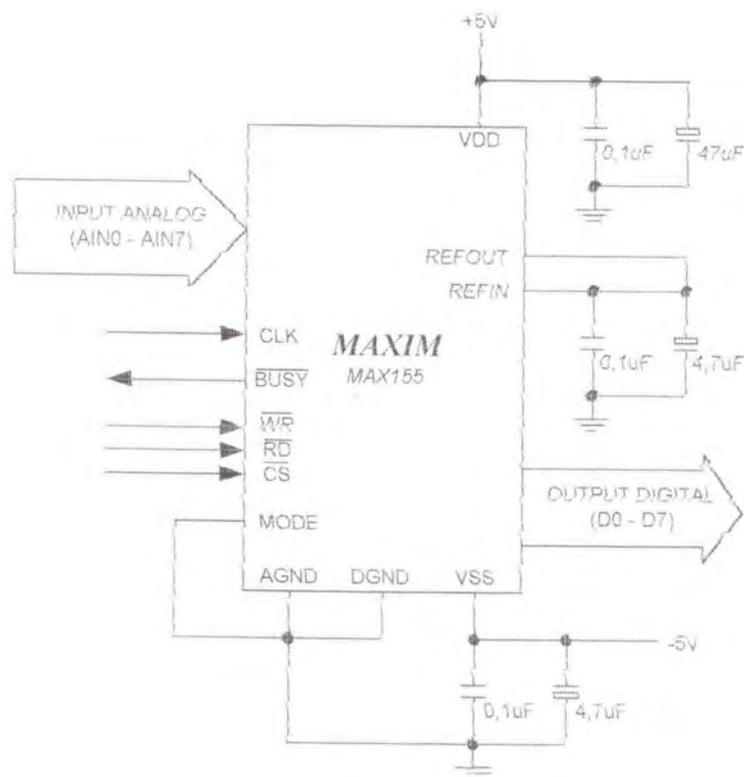
$$t_{ACQ} = 8(R_S + R_{IN}) \times 4pF \quad (\text{tidak pernah kurang dari } 800ns)$$

dimana $R_{IN} = 15k\Omega$ dan $R_S =$ impedansi sumber sinyal input ADC

Waktu konversi dihitung dari :

$$t_{CONV} = \frac{(9 \times N \times 2)}{f_{CLK}}$$

dimana N adalah jumlah kanal yang dikonversi. Untuk satu kanal dengan clock 5MHz, waktu konversinya adalah 3,6 μ s. Untuk MAX155, waktu konversi maksimum untuk 8 kanal adalah 28,8 μ s.



Gambar 3.9 Rangkaian ADC MAX155

3.3.5 INTERFACE KE LPT1 IBM PC

Interface ke LPT1 IBM PC meliputi dua hal penting yaitu sinyal data output dari ADC yang akan dikirimkan ke komputer dan sinyal kontrol dari komputer untuk mengatur proses kerja alat. LPT digunakan untuk menampilkan besarnya sudut mekanik dan bidang mekanik untuk arah X dan Y.

3.4 Perancangan Perangkat Lunak

Perangkat penunjang aplikasi ini adalah *Insight* yang digunakan untuk mendefinisikan :

1. Input
2. Output

3. Variabel fuzzy
4. Rules
5. Simulasi dan down load rule ke IC

3.4.1 Input Dan Output

Penentuan definisi input ditentukan berdasarkan kebutuhan. Dalam hal ini terdapat input internal dan external. Input internal merupakan umpan balik internal yang secara fisik tidak dihubungkan ke komponen luar, namun secara internal dihubungkan dengan software (loop back). Sedangkan input external harus dihubungkan secara fisik.

Pada perancangan alat ini tidak memakai umpan balik internal tetapi digunakan umpan balik external untuk itu jenis input yang dihubungkan secara perangkat keras dan input tersebut adalah:

1. Arah, berfungsi untuk mengetahui arah kemiringan dari mekanik dan bidang mekanik.
2. Sudut, berfungsi untuk mengetahui besar sudut kemiringan dari bidang mekanik dan mekanik.

Output yang langsung dihubungkan dengan komponen di luar adalah 2 buah motor DC yang diatur arah putaran dan kecepatannya.



Gambar 3.10 Input dan Output Pada NLX 220

INPUT

Arah_X
Sudut_X
Arah_Y
Sudut_Y

Output

Putar_X
Cepat_X
Putar_Y
Cepat_Y

3.4.2 Fuzzy Variable

Langkah selanjutnya merancang variabel fuzzy yang diperlukan dalam proses. Format penulisan variabel fuzzy pada software Insight adalah :

antecedent > is <fuzzy set> (center, width, type membership function).

Di dalam variabel *Arah_X* dan *Arah_Y* menentukan arah ke kanan, ke kiri dan arah di tengah. Sedangkan variabel *Sudut_X* dan *Sudut_Y* menentukan kecepatan rendah, menengah dan tinggi. Antecedent yang dipakai sebanyak 10 buah untuk sumbu X dan 6 antecendent untuk sumbu Y.



Gambar 3.11 Variabel Fuzzy untuk antecedent Arah_X dan Arah_serta sudut Y



Gambar 3.12 Variabel Fuzzy untuk antecedent Sudut_X

Fuzzy Variables

Arah_X is Tengah_X (128,7,Symmetric Inclusive)

Arah_X is Kanan_X (128,7,Left Exclusive)

Arah_X is Kiri_X (128,7,Right Enclusive)

Arah_Y is Tengah_Y (128,7,Symmetric Inclusive)

Arah_Y is Kanan_Y (128,7,Left Exclusive)

Arah_Y is Kiri_Y (128,7,Right Enclusive)

Sudut_X is Nol_X (128,7, Symmetric Inclusive)

Sudut_X is P1 3_X (128,61, Left Exclusive)

Sudut_X is N1 3_X (128,60, Right Exclusive)

Sudut_X is P2 3_X (128,40, Left Exclusive)

Sudut_X is N2 3_X (128,40, Right Exclusive)

Sudut_X is P3 3_X (128,7, Left Exclusive)

Sudut_X is N3 3_X (128,7, Right Exclusive)

Sudut_X is Nol_Y (128,7, Symmetric Inclusive)

Sudut_Y is P1 3_Y (128,7, Left Exclusive)

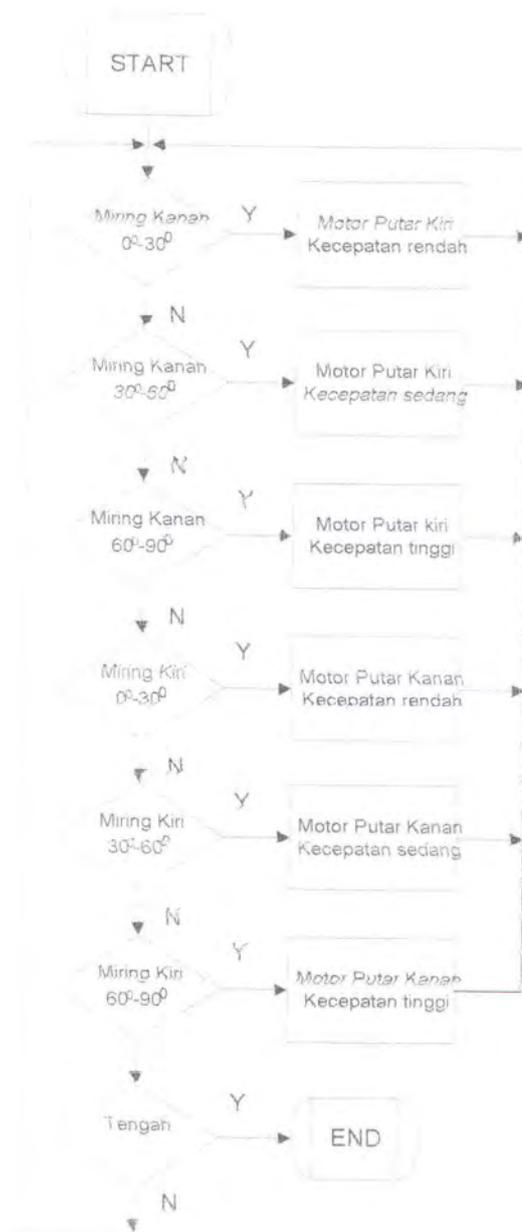
Sudut_Y is N1 3_Y (128,7, Right Exclusive)

3.4.3 Aturan (Rule Sets)

Langkah terakhir adalah membuat sekumpulan aturan (rule sets). Pada sistem pengendalian posisi ini, untuk sumbu X digunakan 10 terms dan sumbu Y digunakan 6 term, masing-masing terdiri dari 7 buah term untuk Sudut_X yang menentukan kecepatan yaitu : Nol (tanpa kecepatan), P1/3 & N1/3 (rendah), P2/3 & N2/3(sedang), P3/3 & N3/3 (tinggi) dan 3 term untuk Arah_X dan sedangkan untuk sumbu Y digunakan 3 term untuk arah yaitu : Tengah, Kanan, Kiri serta 3 term untuk menentukan kecepatan sumbu Y.

Rule yang disusun berdasarkan Algoritma Fuzzy adalah sebagai berikut :





Gambar 3.13 Algoritma Fuzzy.

Jika posisi bidang mekanik berada di tengah maka bidang mekanik tersebut akan berhenti. Apabila posisi bidang mekanik miring ke kiri dengan sudut kemiringan $0^\circ - 30^\circ$ maka motor akan berputar ke kanan dengan kecepatan rendah sampai bidang mekanik di tengah. Dan untuk kemiringan $30^\circ - 60^\circ$ maka motor akan

berputar dengan kecepatan sedang, sedangkan untuk kemiringan $60^\circ - 90^\circ$ maka motor akan berputar dengan kecepatan tinggi. Hal ini untuk sumbu X sedangkan untuk sumbu Y digunakan kecepatan tinggi atau torsi yang maksimum karena motor yang berputar pada sumbu Y menanggung beban motor yang berputar pada sumbu X.

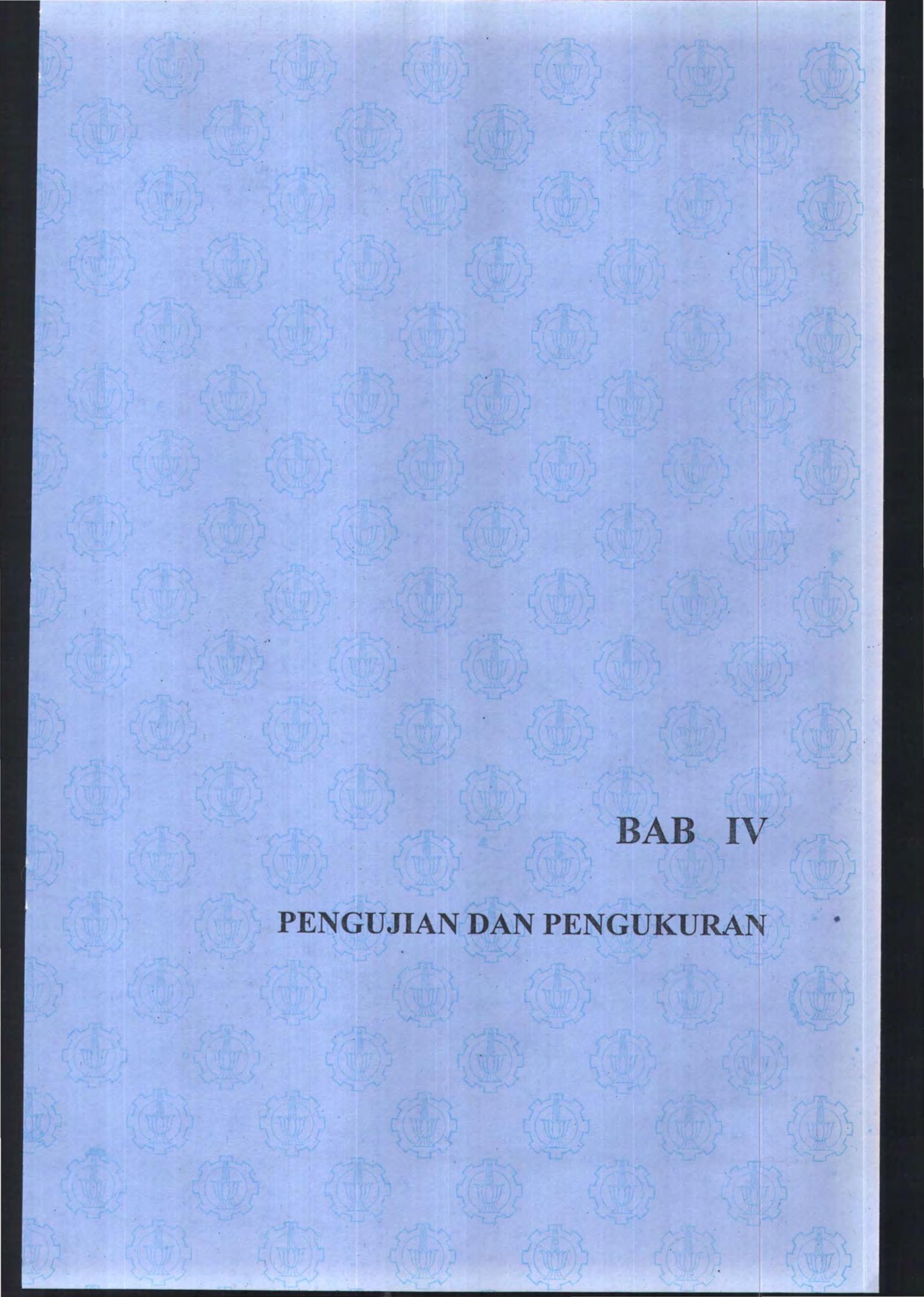
3.5 Perangkat Lunak Akuisisi Data

Proses pembuatan perangkat lunak untuk akuisisi data menggunakan program RAD (*Rapid Application Development*) yaitu Delphi. Beberapa kelebihanannya antara lain bahasa pemrogramannya yang sudah dikenal yaitu Pascal, pembuatan tampilan yang cepat serta program dijalankan dalam basis Windows. Untuk proses akuisisi data mulai dari awal konversi hingga menampilkannya ke komputer ditunjukkan pada diagram alir berikut :



Gambar 3.15 Diagram alir proses akuisisi data

Proses akuisisi data dimulai dengan inialisasi perangkat, yang dilanjutkan dengan pengiriman sinyal WR pada ADC yang menandai awal dari konversi seluruh kanal yang ada. Data analog hasil konversi akan disimpan pada RAM internal ADC. Pulsa RD rendah yang dikirimkan selanjutnya akan menempatkan isi memori tersebut ke buffer output tri-state. Kemudian komputer dapat membaca harga yang tersimpan dalam buffer tri-state ini melalui LPT1. Proses pengiriman sinyal RD rendah serta pembacaan harga isi buffer tri-state tadi diulang sebanyak tujuh kali untuk membaca keseluruhan hasil konversi ADC. Selanjutnya dengan metoda multitasking, ke - 8 data hasil pembacaan ADC ini yang ditampilkan pada layar monitor hanya 4 kanal saja yaitu kanal 1 s/d 4.



BAB IV

PENGUJIAN DAN PENGUKURAN

BAB IV

PENGUJIAN DAN PENGUKURAN

Sebelum digabung menjadi suatu sistem, pada tiap-tiap blok dilakukan pengujian terlebih dahulu. Untuk merealisasikan sistem yang telah direncanakan maka dibuat rangkaian-rangkaian yang terpisah dengan memiliki suatu fungsi yang telah ditentukan. Bagian yang dilakukan pengujian adalah sensor posisi, rangkaian level detektor, rangkaian kecepatan motor dc.

Pengujian dilakukan dengan memberikan input simulasi pada alat yang diuji kemudian melihat hasilnya, apabila hasilnya telah sesuai dengan perencanaan maka alat tersebut dapat bekerja dengan baik dan siap untuk digabung menjadi suatu sistem keseluruhan. Kemudian dilakukan pengujian dari sistem yang tergabung apakah telah sesuai dengan perancangan yang diinginkan.

4.1 Pengujian Sensor Posisi

Pengukuran dilakukan dengan memutar VR (Variable Resistor) dari posisi minimum sampai maksimum, di dapatkan hasil pengukuran 0 V sampai dengan 5 V. Kemudian dilakukan pengukuran tegangan VR untuk tiap-tiap posisi sudut. Dari hasil pengukuran tersebut didapatkan :

Tabel 4.1 Pengujian sensor untuk sumbu X

Sudut ($^{\circ}$)	Tegangan Sensor (V)	Tegangan output fuzzy (V)		Tegangan PWM (V)		Tegangan motor (V)
		2	3	Tanpa switching	Dengan switching	
0 – 10	2,5 – 2,61	0,25	2,52	3,2	2,43	0,252
10 – 30	2,61 – 3,22	3,25	10,7	13,59	9,25	7,6
30 – 60	3,22 – 3,61	3,33	10,7	15,16	9,77	8,1
60 – 90	3,61 – 4,54	4,54	10,7	23,2	14,77	9,5
0 – (-10)	2,5 – 2,36	0,25	2,52	3,2	2,36	-0,252
0 – (-30)	2,36 – 1,77	3,07	0,25	11,64	7,79	-7,3
-30 – (-60)	1,77 – 1,37	3,31	0,25	14,7	10,3	-8,5
-60 – (-90)	1,33 – 0,54	4,54	0,25	23,2	14,8	-9,5

Tabel 4.2 Pengujian sensor untuk sumbu Y

Sudut ($^{\circ}$)	Tegangan Sensor (V)	Tegangan output fuzzy (V)		Tegangan PWM (V)		Tegangan motor (V)
		4	1	Tanpa switching	Dengan switching	
0 – 10	2,5 – 2,61	0,25	2,46	2,75	2,43	1,7
10 – 30	2,61 – 3,22	10,8	10,7	23,2	14,76	9,2
30 – 60	3,22 – 3,61	10,8	10,7	23,2	14,76	9,2
60 – 90	3,61 – 4,54	10,8	10,7	23,2	14,76	9,2
0 – (-10)	2,5 – 2,36	0,25	2,46	2,75	2,43	2,1
0 – (-30)	2,36 – 1,77	10,8	0,25	23,2	14,76	-9,5
-30 – (-60)	1,77 – 1,37	10,8	0,25	23,2	14,76	-9,5
-60 – (-90)	1,33 – 0,54	10,8	0,25	23,2	14,76	-9,5

4.2 Pengujian Positif Level Detector

Mula-mula input rangkaian diberi tegangan 0V kemudian pada titik A diukur terhadap titik B, didapatkan nilai +Vcc (motor putar kanan), dan apabila

diberikan tegangan input 5V didapatkan nilai $-V_{cc}$ (motor putar kiri). Kemudian diberikan input 2,5V didapatkan hasil pengukuran 0V. Sehingga dari hasil uji coba tersebut didapatkan hasil yang sesuai dengan perancangan.

Tabel 4.3 Pengujian switching

Tegangan input (V)	Tegangan motor Ref. $V_{cc} = 12\text{ V}$
0 - 2,36	10,76
2,4 - 2,6	0,25
2,61- 5	-10,76

4.3 Pengujian Rangkaian Pengatur Kecepatan Motor DC

Untuk menguji rangkaian tersebut yaitu dengan memberikan input mulai minimum sampai dengan maksimum (0 s/d 5V). Dari hasil pengukuran tersebut maka tegangan output tersebut adalah :

Tabel 4.4 Tegangan PWM dan pengendali motor DC sumbu X

Input PWM (V)	Tegangan motor (V)		Tegangan PWM (V)			Frekuensi (Hz)	
	V_{ab}	V_{ba}	Tanpa switching	Dengan switching		V1	V0
				V1	V0		
0	2,1	1,8	1,8	1,69	1,7	0	0
0,5	3,2	2,8	4,22	3,23	3,17	598	600
1	3,9	4	5,62	4,12	4,05	1190	1201
1,5	4,8	5	7,09	5	4,93	1780	1798
2	5,6	6	8,61	6	5,71	2392	2388
2,5	6,2	6,7	9,9	6,69	6,7	2989	3010
3	6,9	7,3	11,4	7,75	7,4	3589	3610
3,5	9	9,4	21,4	12,86	12,6	4059	4048
4	9,5	9,8	23,2	14,76	14,79	4786	4750
4,5	9,5	9,8	23,2	14,76	14,79	5383	5363
5	9,5	9,8	23,2	14,76	14,79	5990	5982

Tabel 4.5 Tegangan PWM dan pengendali motor DC sumbu Y

Input PWM (V)	Tegangan motor (V)		Tegangan PWM (V)			Frekuensi (Hz)	
	V _{ab}	V _{ba}	Tanpa switching	Dengan switching		V1	V0
				V1	V0		
0	0,2	0,2	1,33	0	0	0	0
0,5	2	2,5	3,15	1,07	1,08	594	592
1	2,5	3,7	4,27	1,64	1,6	1182	1180
1,5	3,9	4,5	5,31	2,19	2,2	1779	1769
2	4,5	5,2	6,34	2,75	2,75	2353	2376
2,5	5	6	8,72	3,22	3,32	2995	3048
3	6,3	7	10,23	4,54	4,5	3576	3550
3,5	9	9,2	19,78	9,6	9,7	4183	4163
4	9,5	9,4	23,2	11,86	9,5	4783	4792
4,5	9,5	9,4	23,2	11,86	9,5	5366	5375
5	9,5	9,4	23,2	11,86	9,5	5990	6001

Keterangan :

V1 : input switching diberi tegangan +5V (putar kanan)

V0 : input switching diberi tegangan 0V (putar kiri)

V_{ab} : apabila input switching = 4,54 V

V_{ab} : apabila input switching = 0,25 V

Kemudian untuk menentukan besarnya kecepatan yang diinginkan maka dilakukan pemberian tegangan input untuk mendapatkan tegangan output yang diinginkan untuk sumbu X dan putar kanan, yaitu :

1. Kecepatan rendah, tegangan output fuzzy = 3,25 V dan tegangan motor = 7,6 V dan frekuensi = 3878 Hz.

2. Kecepatan sedang, tegangan output fuzzy = 3,33 V dan tegangan motor = 8,1 V dan frekuensi = 3982 Hz.
3. Kecepatan tinggi, tegangan output fuzzy = 4,54 V dan tegangan motor = 9,5 V dan frekuensi = 5435 Hz.

Sumbu X putar kiri adalah :

1. Kecepatan rendah, tegangan output fuzzy = 3,07 V dan tegangan motor = 7,3 V dan frekuensi = 3571 Hz.
2. Kecepatan sedang, tegangan output fuzzy = 3,31 V dan tegangan motor = 8,5 V dan frekuensi = 3975 Hz.
3. Kecepatan tinggi, tegangan output fuzzy = 4,54 V dan tegangan motor = 9,5 V dan frekuensi = 5450 Hz.

Dan untuk sumbu Y putar kiri maupun kanan hampir sama kecepatan, yaitu ;

1. Kecepatan tinggi, tegangan output fuzzy = 4,54 V dan tegangan motor = 9,5 V dan frekuensi = 5463 Hz.

4.4 Pengujian Modul NLX 220

Untuk menguji modul ini, dibuat rule atau aturan yang sederhana yaitu harga tegangan input sama dengan harga tegangan output. Sehingga dari hal ini dapat mengetahui apakah modul NLX220 sudah berjalan atau belum. Selain itu dapat juga diketahui urutan pin input dan pin output sehingga mempermudah pengujian sistem.

Variabel Fuzzy yang digunakan adalah sebagai berikut :

- ◆ Input is inp1 (10,0,right inclusive)
- ◆ Input is inp2 (0,0,right inclusive)
- ◆ Input is inp3 (0,0,right inclusive)
- ◆ Input is inp4 (0,0,right inclusive)

Sedangkan rule-rulanya adalah sebagai berikut :

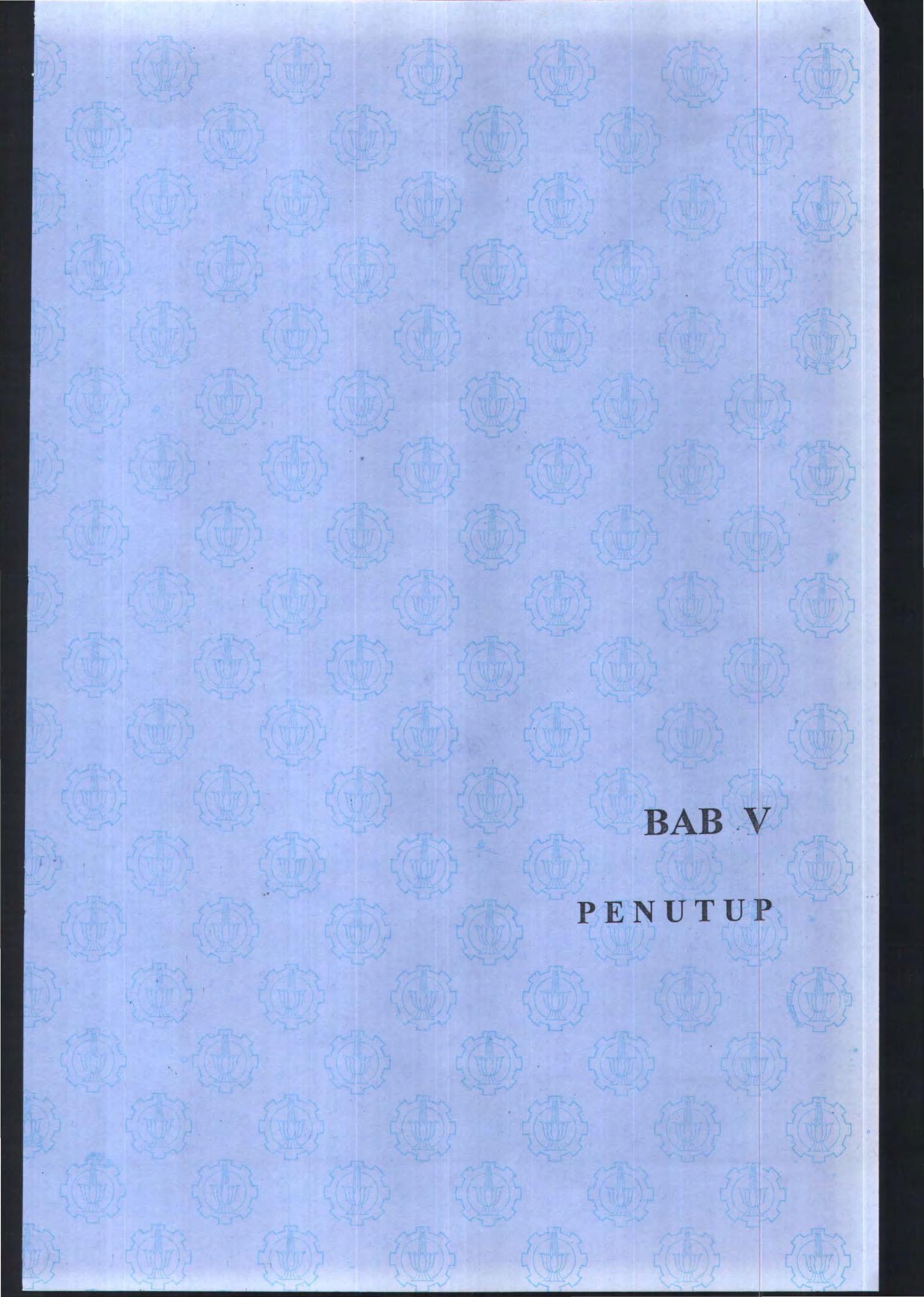
- ◆ If input1 is inp1 then output1 = input1
- ◆ If input2 is inp2 then output2 = input2
- ◆ If input3 is inp3 then output3 = input3
- ◆ If input4 is inp4 then output4 = input4

Dengan menggunakan kelompok rule tersebut maka pada setiap input diberi sumber tegangan dan outputnya diukur. Dari hasil pengukuran diperoleh harga output sama dengan harga input. Hal ini membuktikan bahwa modul telah bekerja dengan baik. Nilai digital yang dimasukkan ke dalam rule apabila dikeluarkan akan menghasilkan tegangan output :

$$V_{out} \approx \text{Digital.value} \times \frac{5}{256}$$

4.5 Pengujian Sistem

Setelah dilakukan pengujian tiap blok sistem rangkaian maka dilakukan pengujian untuk sistem secara keseluruhan.



BAB V

PENUTUP

BAB V

PENUTUP

5.1 Kesimpulan

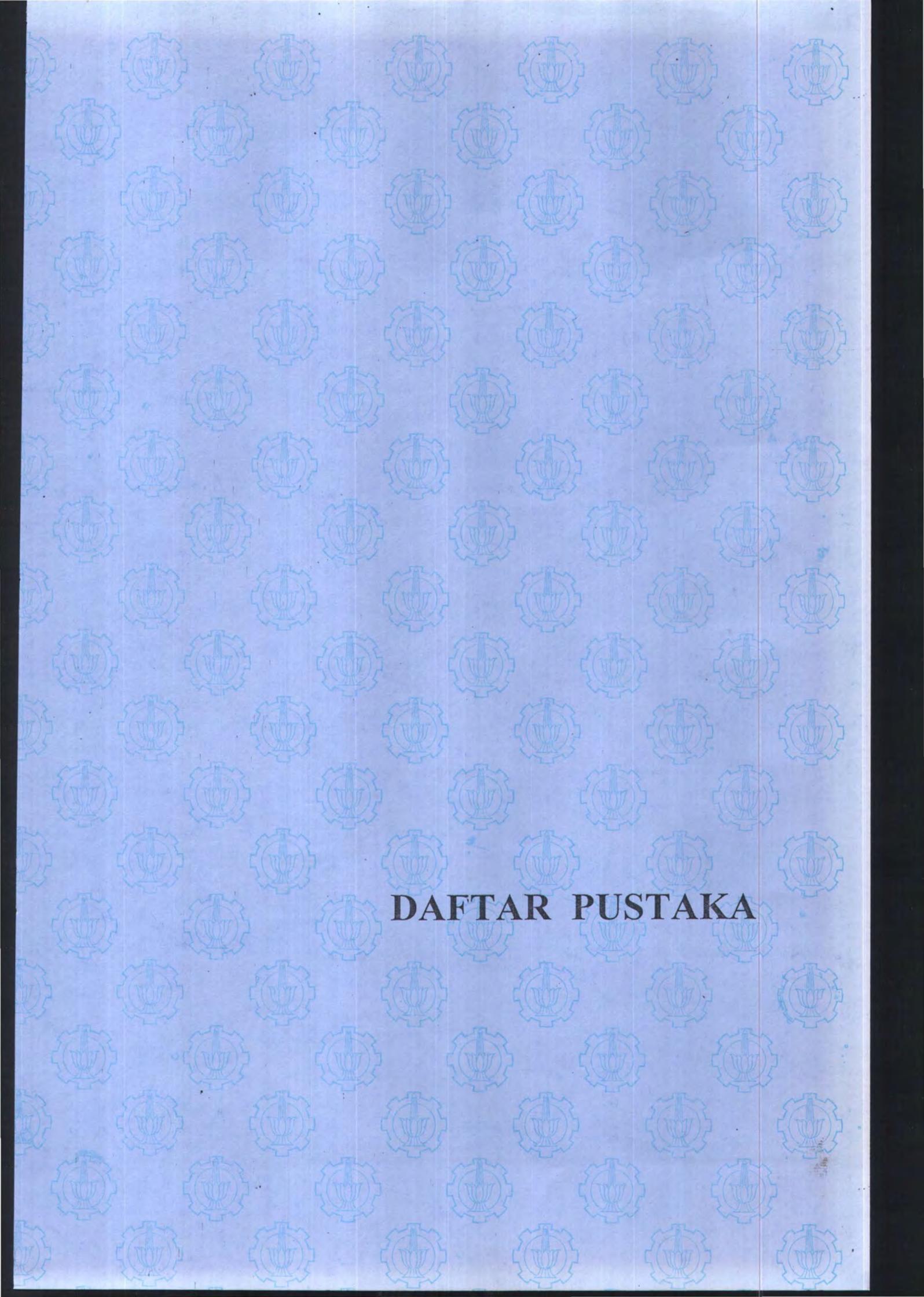
Dari perencanaan dan pembuatan alat pengendali posisi terhadap arah gravitasi ini dapat diambil suatu kesimpulan ;

1. Pengontrolan dengan suatu sistem, terutama sistem mekanis dengan fuzzy logic akan bekerja sesuai dengan yang diharapkan hanya jika bagian sensor masukan dan bagian keluaran komponen mekanis bekerja dengan baik.
2. Dengan transducer resistor variabelnya yang biasanya tidak linier ternyata fuzzy logic dapat melinierkannya, sehingga untuk aplikasi cukup baik sebagai transducer.
3. Di dalam pembuatan aturan-aturan rule yang digunakan terlebih dahulu harus tahu mekanik yang akan dikontrol, yaitu parameter input dan outputnya.
4. Dengan menggunakan fuzzy logic controller, mekanisme pengontrolan dilakukan seperti cara berfikir manusia.
5. PWM digunakan sebagai pengatur kecepatan motor karena kenaikan tegangan akan lebih halus daripada penguat biasa. PWM bersifat sebagai penambah, sedangkan penguat bersifat sebagai pengali.

5.2 Saran

Untuk pengembangan dan peningkatan lebih lanjut dari alat yang telah dibuat, kami sampaikan beberapa saran :

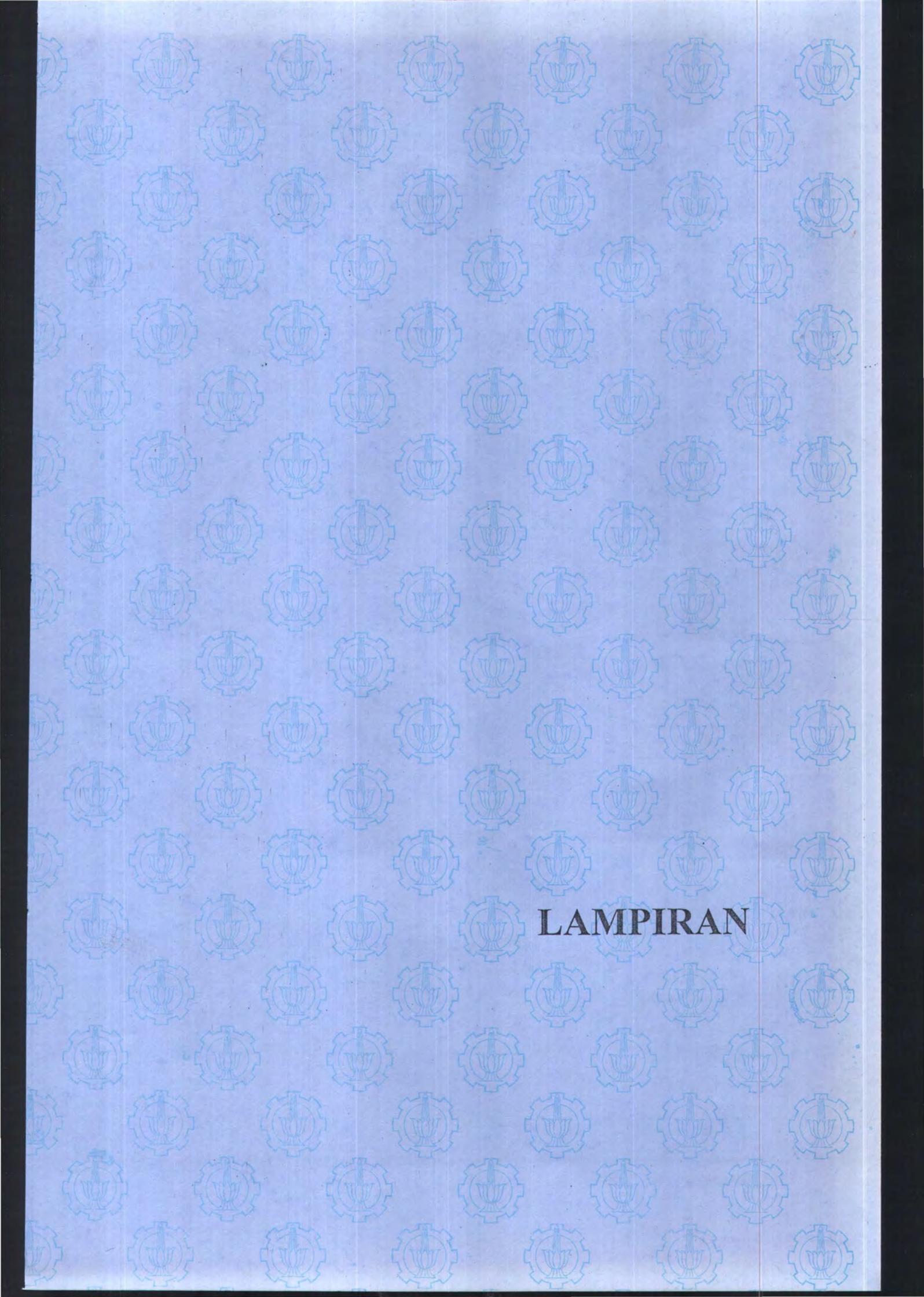
1. Perancangan dan penggunaan mekanis yang lebih presisi.
2. Penggunaan teknologi fuzzy dengan metode lain, sehingga dapat dilakukan perbandingan yang tepat untuk suatu permasalahan.



DAFTAR PUSTAKA

DAFTAR PUSTAKA

1. Coughlin, Robert F. Frederick F. Driscoll. 1982, **Operational Amplifier Linear Integrated Circuit**, Englewood Cliffs. New York : Prentice Hall Inc.
2. Marcelo, A, Edward J. F, **Dasar-Dasar Fisika Universitas**, Erlangga, 1990.
3. Jun Yan, Ryan. M dan Power. J, **Using Fuzzy Logic**, London, Prentice Hall,1994.
4. Katsuhito Ogata, Edi Leksono, **Teknik Kontrol Automatik Jilid II**, Jakarta, Erlangga, 1990
5. Pacheco, Xavier. Steve Teixeira, 1996, **Delphi 2 Developer's Guide. 2nd Edition**, Indianapolis, Borland International Inc. Sam Publishing.
6. Paul Bergsman, **Microcomputer Journal**, Januari 1985
7. ---. **Fuzzy Logic Education Program**, Motorola
8. ---. **Fuzzy Microcontroller Development System**, American Neural Logic.
9. ---. **IBM PC AT Technical Reference**, IBM Co,1984.
10. ---. **MAXIM Full-Line Data Catalog – CDROM**. MAXIM – IC,1998.
11. ---. **National Operational Amplifier Databook**. National Semiconductor,1995.
12. ---. **Teknologi Motor dan Generator**, 1988



LAMPIRAN

INPUT

Arah_X

Sudut_X

Arah_Y

Sudut_Y

Output

Putar_X

Cepat_X

Putar_Y

Cepat_Y

Fuzzy Variables

Arah_X is Tengah_X (128,7,Symmetric Inclusive)

Arah_X is Kanan_X (128,7,Left Exclusive)

Arah_X is Kiri_X (128,7,Right Exclusive)

Arah_Y is Tengah_Y (128,7,Symmetric Inclusive)

Arah_Y is Kanan_Y (128,7,Left Exclusive)

Arah_Y is Kiri_Y (128,7,Right Exclusive)

Sudut_X is Nol_X (128,7, Symmetric Inclusive)

Sudut_X is P1/3_X (128,61, Left Exclusive)

Sudut_X is N1/3_X (128,60, Right Exclusive)

Sudut_X is P2/3_X (128,40, Left Exclusive)

Sudut_X is N2/3_X (128,40, Right Exclusive)

Sudut_X is P3/3_X (128,7, Left Exclusive)

Sudut_X is N3/3_X (128,7, Right Exclusive)

Sudut_X is Nol_Y (128,7, Symmetric Inclusive)

Sudut_Y is P1/3_Y (128,7, Left Exclusive)

Sudut_Y is N1/3_Y (128,7, Right Exclusive)

Rules

If Arah_X is tengah

Then Putar_X =128
If Arah_X is kanan
Then Putar_X =255
If Arah_X is kiri
Then Putar_X = 0
If Arah_Y is tengah
Then Putar_Y =128
If Arah_Y is kanan
Then Putar_Y =255
If Arah_Y is kiri
Then Putar_Y = 0
If Arah_X is nol
Then Cepat_X = 0
If Arah_X is P1/3
Then Cepat_X = 255
If Arah_X is N1/3
Then Cepat_X = 255
If Arah_X is P2/3
Then Cepat_X = 182
If Arah_X is N2/3
Then Cepat_X = 181
If Arah_X is P3/3
Then Cepat_X = 177
If Arah_X is N3/3
Then Cepat_X = 167
If Arah_Y is nol
Then Cepat_Y = 0
If Arah_Y is P1/3
Then Cepat_Y = 255
If Arah_Y is N1/3
Then Cepat_Y = 255

```

unit takul;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
  Dialogs,
  OleCtrls, AGAUGELib_TLB,Control,StdCtrls, ExtCtrls;

type
  TForm1 = class(TForm)
    AGauge1: TAGauge;
    AGauge2: TAGauge;
    AGauge4: TAGauge;
    AGauge3: TAGauge;
    Timer1: TTimer;
    Timer2: TTimer;
    Button1: TButton;
    Button2: TButton;
    Image1: TImage;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    GroupBox1: TGroupBox;
    GroupBox2: TGroupBox;
    Label4: TLabel;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
    procedure Timer2Timer(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

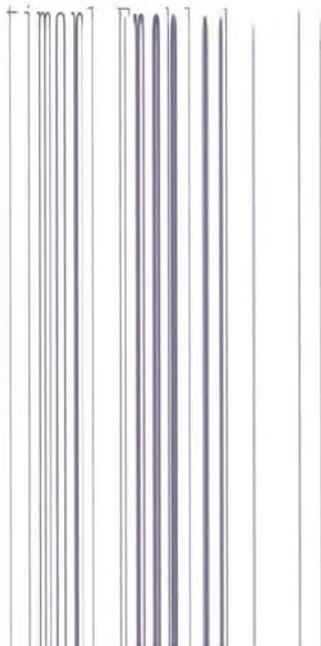
var
  Form1: TForm1;

implementation

{$R *.DFM}

procedure TForm1.Button1Click(Sender: TObject);
begin

```



```

unit takul;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
  Dialogs,
  OleCtrls, AGAUGELib_TLB, Control, StdCtrls, ExtCtrls;

type
  TForm1 = class(TForm)
    AGauge1: TAGauge;
    AGauge2: TAGauge;
    AGauge4: TAGauge;
    AGauge3: TAGauge;
    Timer1: TTimer;
    Timer2: TTimer;
    Button1: TButton;
    Button2: TButton;
    Image1: TImage;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    GroupBox1: TGroupBox;
    GroupBox2: TGroupBox;
    Label4: TLabel;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
    procedure Timer2Timer(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.DFM}

procedure TForm1.Button1Click(Sender: TObject);
begin
  timer1.Enabled := true;
  timer2.Enabled := true;
  label1.visible := true;
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
  timer1.Enabled := false;
  timer2.Enabled := false;
  label1.visible := false;
end;

procedure TForm1.Timer1Timer(Sender: TObject);
var

```

```
DataInt0,DataInt1,DataInt2,DataInt3 : real;
begin
    Proses;
end;

procedure TForm1.Timer2Timer(Sender: TObject);
var
    hour,min,sec,msec :word;
begin
    decodetime(time, hour, min, sec, msec);
    AGauge1.NeedleValue := DataInt2;
    AGauge2.NeedleValue := DataInt3;
    AGauge3.NeedleValue := DataInt0;
    AGauge4.NeedleValue := DataInt1;
end;

end.
```

Unit Control;

Interface
uses Winprocs;

var

Data_378h,Data_379h: byte;
Data8bit,detect_high: byte;

DataADC0,DataADC1,DataADC2,DataADC3: byte;
DataADC4,DataADC5,DataADC6,DataADC7: byte;

DataInt0,DataInt1,DataInt2,DataInt3: real;
DataInt4,DataInt5,DataInt6,DataInt7: real;

const

write : byte = \$01;
read: byte = \$02;
idle: byte = \$03;

procedure InisialisasiLPT1;
procedure bacastatus_379h;
procedure Sampling;
procedure Bacadata8bit;
procedure BacaADC;
procedure Konversi;
procedure Proses;

Implementation

uses takul, SysUtils;

{AX-Accumulator-menampung hasil dari operasi aritmatika(juga unt
keperluan lainnya)
BX-Base-digunakan unt. pengalamatan di memory
CX-Count-menyimpan informasi jumlah perulangan yg akan dilakukan
DX-Data-menyimpan data secara umum}

procedure InisialisasiLPT1;

begin

asm
push ax
push dx
mov dx,378h
mov al,3h
out dx,al
pop dx
pop ax
end;

end;

procedure Sampling;

begin

asm
push ax
push dx
push cx
mov cx,0fffh

```

    mov dx,378h
@loop2:    mov al,1h
    out dx,al
    dec cx
    jnz @loop2

```

```

    mov al,3h
    out dx,al

```

```

    pop cx
    pop dx
    pop ax
end;
end;

```

```

procedure bacastatus_379h;

```

```

begin

```

```

    ASM

```

```

        push dx
        push ax
        pushf
        mov dx,379h
@loop1: in al,dx
        and al,8h
        mov bl,al
        mov al,8h
        cmp al,bl
        je @loop1
        popf
        pop ax

```

```

        mov dx,378h
        mov al,3h
        out dx,al
        pop dx

```

```

    end;
end;

```

```

procedure Bacadata8bit;

```

```

var LSB : byte;

```

```

begin

```

```

    ASM

```

```

        push ax
        push dx
        mov dx,378h
        mov al,2h
        out dx,al           {kirim pulsa awal baca ADC}

```

```

        mov dx,379h
        in al,dx
        and al,11110000b   {ambil nibble dari 4-bit MSB}
        xor al,80h         {XOR 80h}
        mov LSB,al        {selamatkan data 4-bit MSB}

```

```

        mov dx,37Ah
        mov al,4h
        out dx,al         {open collector output high}

```

```

        mov dx,37Ah

```

```

in al,dx
and al,00001111b {ambil nibble dari 4-bit LSB}
xor al,0Bh {XOR 0Bh}

or al,LSB {OR-kan 4-bit MSB dan 4-bit LSB}
mov Data8bit,al {ambil data 8-bit}

mov dx,378h
mov al,3h {kirim pulsa akhir baca ADC}
out dx,al
pop dx
pop ax
end;
end;

```

```

procedure BacaADC;
begin
  Bacadata8bit;
  DataADC0:= Data8bit;

  Bacadata8bit;
  DataADC1:= Data8bit;

  Bacadata8bit;
  DataADC2:= Data8bit;

  Bacadata8bit;
  DataADC3:= Data8bit;

  Bacadata8bit;
  DataADC4:= Data8bit;

  Bacadata8bit;
  DataADC5:= Data8bit;

  Bacadata8bit;
  DataADC6:= Data8bit;

  Bacadata8bit;
  DataADC7:= Data8bit;
end;

```

```

procedure Konversi;
begin
  begin
    if (DataADC0>=30) and (DataADC0<128) then
      DataInt0:=90/98*(DataADC0-128);
    if (DataADC1>=46) and (DataADC1<157) then
      DataInt1:=90/111*(DataADC1-157);
    if (DataADC2>=63) and (DataADC2<164) then
      DataInt2:=90/101*(DataADC2-164);
    if (DataADC3>=53) and (DataADC3<161) then
      DataInt3:=90/108*(DataADC3-161);
    if (DataADC0>=128) and (DataADC0<223) then
      DataInt0:=90/95*(DataADC0-128);
    if (DataADC1>=157) and (DataADC1<255) then
      DataInt1:=90/98*(DataADC1-157);
    if (DataADC2>=164) and (DataADC2<252) then
      DataInt2:=90/88*(DataADC2-164);
  end;
end;

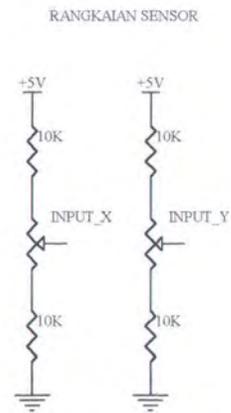
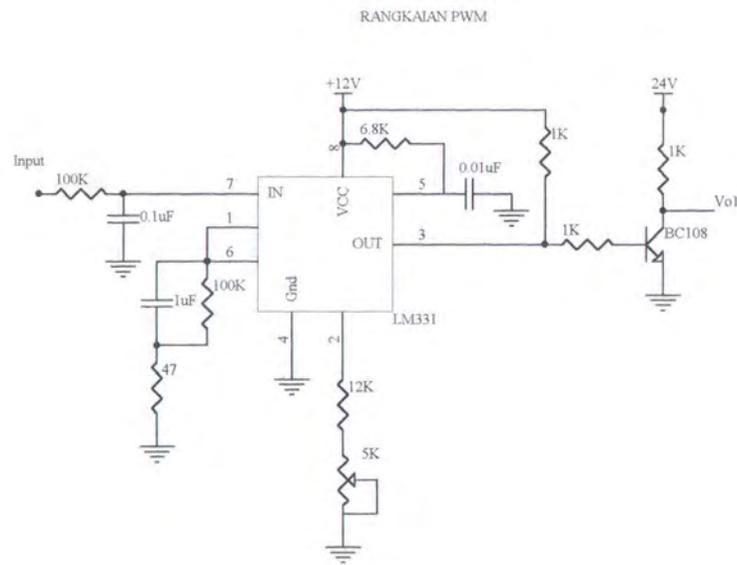
```

```
    if (DataADC3>=161) and (DataADC3<251) then
        DataInt3:=90/90*(DataADC3-161);

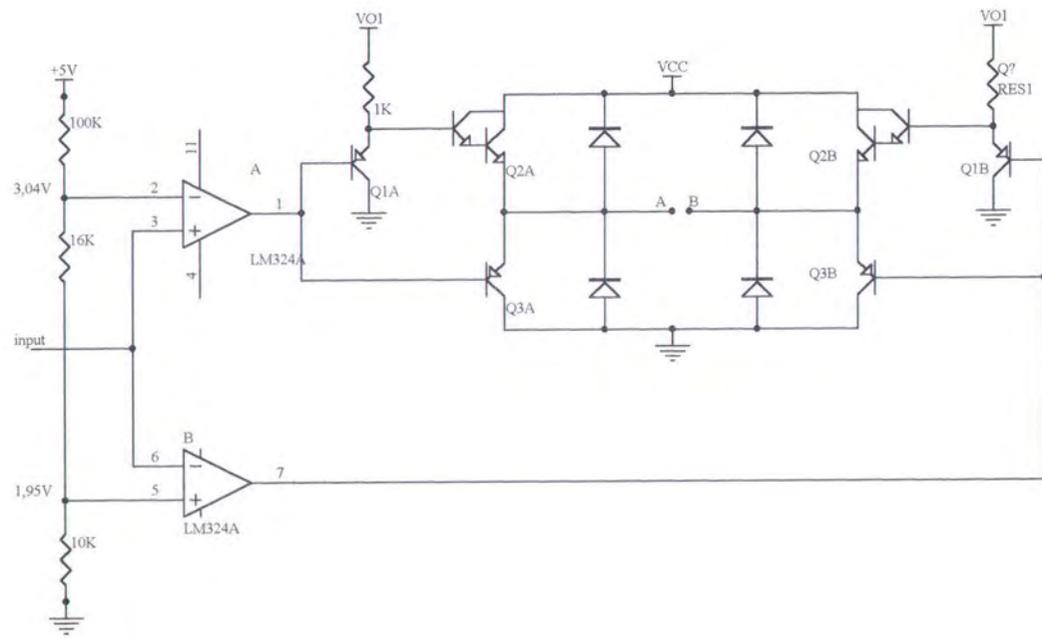
    end;
end;

procedure Proses;
begin
    inisialisasiLPT1;
    Sampling;
    bacastatus_379h;
    BacaADC;
    Konversi;
end;

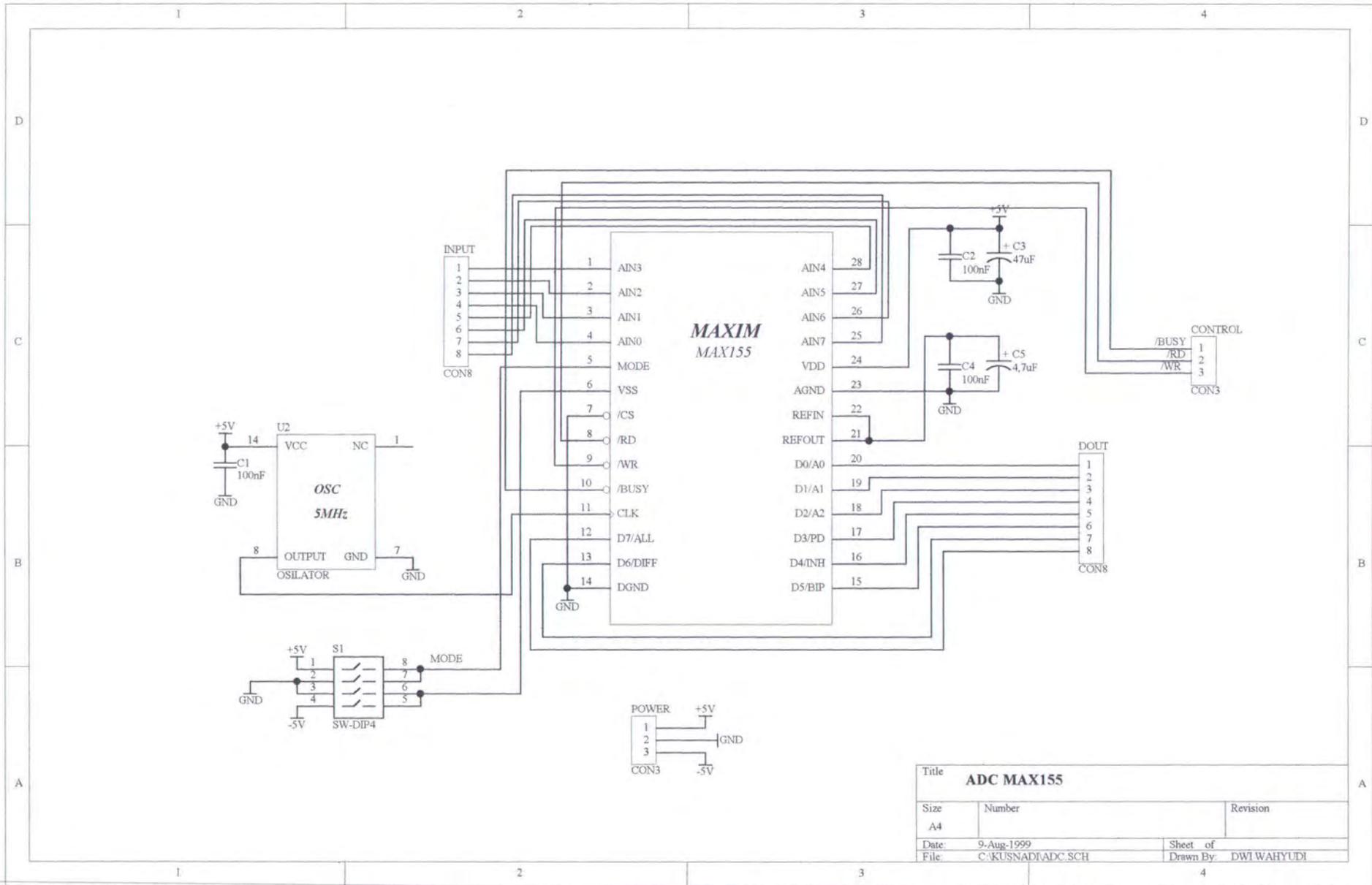
end.
```



Title			RANGKAIAN PWM DAN SENSOR		
Size	Number	Revision			
A4					
Date:	9-Aug-1999	Sheet of			
File:	C:\KUSNADI\PWM_TA.SCH	Drawn By:			



Title			
Size	Number	Revision	
A4	RANGKAIAN SWITCHING		
Date:	9-Aug-1999	Sheet of	
File:	C:\KUSNADEDMOT.SCH	Drawn By:	



DAFTAR RIWAYAT HIDUP



Kusnadi Sugiarto, dilahirkan di Surabaya, Jawa Timur pada tanggal 12 Mei 1973 merupakan putra kedua dari tiga bersaudara dari Bapak Djuremi Mardjuni dan Ibu Harini, yang bertempat tinggal di Surabaya, Jawa Timur.

Terdaftar sebagai mahasiswa Program Lintas Jalur Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya pada tahun 1996 dengan nomor pokok 2296100532.

Pendidikan yang telah ditempuh sampai saat ini adalah :

- SD Negeri 267 Surabaya, pada tahun 1979 – 1985
- SMP Negeri 13 Surabaya, pada tahun 1985 – 1988
- SMA Negeri 16 Surabaya, pada tahun 1988 – 1991
- Politeknik Elektronika Surabaya – ITS, pada tahun 1991 – 1994
- Institut Teknologi Sepuluh Nopember Surabaya, Fakultas Teknologi Industri, Jurusan Teknik Elektro, Bidang studi Elektronika sebagai mahasiswa program Lintas Jalur sejak tahun 1996.

Selain jenjang pendidikan di atas, penulis pernah bekerja di beberapa perusahaan, antara lain :

- PT. Indonesia Chemi-Con, sebagai utility foreman pada tahun 1994 – 1995.
- PT. Sorini Towa Berlian Corp, sebagai Hydrogen Gas Foreman pada tahun 1995 – 1996.
- PT. Maspion, sebagai EDP staff pada bulan Mei 1999 – Sekarang.

Selama menjadi mahasiswa program Lintas Jalur, aktif sebagai asisten Praktikum Rangkaian Listrik, asisten Praktikum Rangkaian Elektronika, dan asisten Praktikum Elektronika Lanjut II.

