

**PENGATURAN KECEPATAN MOTOR INDUKSI
TIGA FASA MENGGUNAKAN LOGIKA FUZZY**

SKRIPSI

Oleh

ANWAR HUSNAWAN

04 02 23 0085



**TUGAS AKHIR INI DIAJUKAN UNTUK MELENGKAPI
SEBAGIAN PERSYARATAN MENJADI SARJANA TEKNIK**

**DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS INDONESIA
GANJIL 2007/2008**

PERNYATAAN KEASLIAN TUGAS AKHIR

Saya menyatakan dengan sesungguhnya bahwa skripsi dengan judul:

PENGATURAN KECEPATAN MOTOR INDUKSI TIGA FASA MENGUNAKAN LOGIKA FUZZY

yang dibuat untuk melengkapi sebagian persyaratan menjadi Sarjana Teknik pada Program Studi Teknik Elektro Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia, sejauh yang saya ketahui bukan merupakan tiruan atau duplikasi dari tugas akhir yang sudah dipublikasikan dan atau pernah dipakai untuk mendapat gelar kesarjanaan di lingkungan Universitas Indonesia maupun di Perguruan Tinggi atau Instansi manapun, kecuali bagian yang sumber informasinya dicantumkan sebagaimana mestinya.

Depok, 08 Januari 2008

Anwar Husnawan
NPM. 04 02 23 0085

PENGESAHAN

Skripsi dengan judul :

PENGATURAN KECEPATAN MOTOR INDUKSI TIGA FASA MENGUNAKAN LOGIKA FUZZY

dibuat untuk melengkapi sebagian prasyarat menjadi Sarjana Teknik pada Program Studi Teknik Elektro Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia. Skripsi ini telah diujikan pada sidang ujian skripsi pada tanggal 04 Januari 2007 dan dinyatakan memenuhi syarat/sah sebagai skripsi pada Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia.

Depok, 08 Januari 2008

Dosen Pembimbing

Dr. Ir. Ridwan Gunawan, MT
NIP. 130 869 978

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada:

Dr. Ir. Ridwan Gunawan, MT.

selaku dosen pembimbing yang telah bersedia meluangkan waktu untuk memberikan pengarahan, diskusi, dan bimbingan serta persetujuan sehingga skripsi ini dapat selesai dengan baik.

Anwar Husnawan
NPM 04 02 23 0085
Departemen Teknik Elektro

Dosen Pembimbing
Dr. Ir. Ridwan Gunawan, MT

PENGATURAN KECEPATAN MOTOR INDUKSI TIGA FASA MENGUNAKAN LOGIKA FUZZY

ABSTRAK

Dalam makalah ini disajikan aplikasi kendali logika fuzzy untuk pengaturan kecepatan motor induksi dalam bentuk simulasi. Kendali logika fuzzy disimulasikan dengan menggunakan program Matlab. Pengaturan kecepatan motor dilakukan dengan mengatur tegangan motor dan menggunakan metode Pulse Width Modulation (PWM) yang ideal. Pembebanan pada motor dilakukan dengan cara bertahap. Feedback sistemnya adalah sinyal frekuensi dari alat ukur yang ideal. Sistem logika fuzzy mempunyai 2 crisp input yaitu error dan perubahan error kecepatan motor dan mempunyai 1 crisp output yaitu perubahan tegangan. Metode defuzzifikasi yang digunakan adalah center of area. Jumlah label dari membership function bervariasi yaitu 7 label. Jumlah rule mengikuti jumlah label yang digunakan. Respon sistem ditampilkan dalam bentuk grafik kecepatan motor terhadap waktu. Hasil pengujian dibandingkan dengan pengendali PI yang telah ada dan menunjukkan FLC sistem dapat menangani kecepatan tanpa overshoot dan steady-state error dengan perubahan kecepatan dan beban yang mendadak.

Kata kunci: Motor Induksi, Logika Fuzzy

Anwar Husnawan
NPM 04 02 23 0085
Electro Engineering

Dosen Pembimbing
Dr. Ir. Ridwan Gunawan, MT

PENGATURAN KECEPATAN MOTOR INDUKSI TIGA FASA MENGUNAKAN LOGIKA FUZZY

ABSTRACT

This paper present the application of fuzzy logic control for induction motor speed control simulation. Fuzzy logic control simulated by Matlab program. The speed control is done by adjusting motor voltage and using Pulse Width Modulation (PWM) Method. The loading on the motor is executed with step torque. The system feedback is a signal frequency from the ideal. Fuzzy logic system applies two crisps of input: error and error change of the motor speed; and an output crisp, i.e. voltage change. The defuzzification methods used is center of area. The numbers of rules is 7 labels. The system response is displayed by graphic of the motor speed toward time. The testing result has been compare with PI controller and showed FLC system able to control the speed without overshoot and steady-state error on the different load and speed.

Keywords: Induction Motor, Fuzzy Logic

DAFTAR ISI

	Halaman
PERNYATAAN KEASLIAN TUGAS AKHIR	ii
PENGESAHAN	iii
UCAPAN TERIMA KASIH	iv
ABSTRAK	v
ABSTRACT	vi
DAFTAR ISI	vii
DAFTAR GAMBAR	ix
DAFTAR TABEL	x
DAFTAR SIMBOL	xi
BAB I PENDAHULUAN	1
1.1. LATAR BELAKANG	1
1.2. PERUMUSAN DAN BATASAN MASALAH	2
1.3. TUJUAN PENULISAN	3
1.4. METODE PENELITIAN	3
1.5. SISTEMATIKA PEMBAHASAN	4
BAB II LANDASAN TEORI	5
2.1 MOTOR INDUKSI	5
2.1.1 Komponen	5
2.1.2 Klasifikasi motor induksi	6
2.1.3 Kecepatan motor induksi	6
2.1.4 Hubungan antara beban, kecepatan dan torque	7
2.1.5 Rangkaian Pengganti Motor Induksi	8
2.2 LOGIKA FUZZY	10
2.2.1 Himpunan Fuzzy	11
2.2.2 Fungsi Keanggotaan	11
2.2.3 Operator dan Operator Himpunan Fuzzy	14
2.2.4 Basis Aturan	15
2.2.5 Inferensi Fuzzy	15
2.2.6 Fuzzifikasi	16

2.2.7 Defuzzifikasi	16
2.2.8 Sistem Fuzzy mamdani	17
2.3 MODULASI LEBAR PULSA (PWM)	18
2.3.1 Kontrol Kecepatan Motor dengan PWM inverter	21
BAB III PERANCANGAN SISTEM	22
3.1 DESAIN PLANT	22
3.2 STRUKTUR PENGATURAN LOGIKA FUZZY	23
3.3 PERANCANGAN UNIT PENGENDALI LOGIKA FUZZY	26
3.3.1 Metode MAX-MIN	27
3.3.2 Metode Center of Area	28
3.4 FUZZY SET DAN FORMASI ATURAN	28
3.5 PEMODELAN SISTEM	32
BAB IV PENYAJIAN DATA DAN PEMBAHASAN	37
4.1 DASAR ATURAN (RULE BASE)	37
4.2 HASIL SIMULASI DAN PENGUJIAN RESPON	39
4.2.1 Simulasi dan Analisa Uji Respon Input dengan Torsi Beban 0 N.m	41
4.2.2 Simulasi dan Analisa Uji Respon Input dengan Torsi Beban 100 N.m	43
4.2.3 Simulasi dan Analisa Uji Respon Input dengan Torsi Beban 200 N.m	45
4.2.4 Simulasi dan Analisa Uji Respon Input dengan Torsi Step	47
4.3 PERBANDINGAN PENGENDALI PI DAN FUZZY	49
BAB V KESIMPULAN	51
DAFTAR ACUAN	52
DAFTAR PUSTAKA	53
LAMPIRAN	54

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Bagian-bagian Motor Induksi	5
Gambar 2.2 Grafik Torsi – Kecepatan Motor Induksi tiga fasa [10]	8
Gambar 2.3 Rangkaian ekivalen motor induksi per-fasa	8
Gambar 2.4 Contoh Fungsi keanggotaan	13
Gambar 2.5 Mekanisme inferensi Fuzzy Mamdani	17
Gambar 2.6 Diagram proses pembuatan sinyal ke Gate pada IGBT	20
Gambar 2.7 Diagram pembentukkan sinyal PWM	20
Gambar 2.8 Sistem Pengaturan Kecepatan Motor Menggunakan PWM	21
Gambar 3.1 Respon Sistem dari suatu Plant	22
Gambar 3.2 Konfigurasi Sistem Logika Fuzzy	24
Gambar 3.3 Diagram Blok Pengaturan Kecepatan Motor Induksi	24
Gambar 3.4 Membership Function	26
Gambar 3.5 Variabel Input Q_e dan ΔQ_e	26
Gambar 3.6 Proses kuantisasi	27
Gambar 3.7 Metode center of area	28
Gambar 3.8 Pemodelan sistem dengan MATLAB Simulink	34
Gambar 3.9 Model Pengendali Logika Fuzzy	35
Gambar 3.10 Diagram alir Pengaturan kecepatan dengan logika fuzzy	36
Gambar 4.1 FIS Editor untuk metode Mamdani	37
Gambar 4.2 Fungsi keanggotaan input Q_e dan ΔQ_e	38
Gambar 4.3 Surface Viewer	38
Gambar 4.4 Grafik respon PI untuk beban 0 N.m	41
Gambar 4.5 Grafik respon FLC untuk beban 0 N.m	42
Gambar 4.6 Grafik respon PI untuk beban 100 N.m	43
Gambar 4.7 Grafik respon FLC untuk beban 100 N.m	44
Gambar 4.8 Grafik respon PI untuk beban 200 N.m	45
Gambar 4.9 Grafik respon FLC untuk beban 200 N.m	46
Gambar 4.10 Grafik respon PI untuk beban torsi step	47
Gambar 4.11 Grafik respon FLC untuk beban torsi step	48

DAFTAR TABEL

	Halaman
Tabel 3.1 Aturan dasar pengkoreksian berbasis fuzzy	31
Tabel 3.2 Spesifikasi motor induksi	32
Tabel 4.1 Hasil Percobaan Pengendali PI	49
Tabel 4.2 Hasil Percobaan Pengendali FL	50

DAFTAR SIMBOL

Simbol	Keterangan	Dimensi
a, b, c	sumbu fasa	
ns	kecepatan medan putar	rpm
nr	kecepatan rotor	rpm
s	slip	
V, v	tegangan	volt
I, i	arus	amper
f	frekuensi stator	Hz
p	jumlah kutub	
F	gaya gerak magnet	
R	tahanan	ohm
L	induktansi	henry
X	reaktansi	ohm
Te	torsi elektromagnetik	N.m
Pn	daya nominal	watt
J	rotor inertia	kg m ²
w	motor speed	rad/sec
Rs, Rr	stator dan rotor resistansi	ohm
Ls, Lr	stator dan rotor induktansi	henry
Lm	mutual induktansi	henry
Tr	waktu naik	detik
Tp	waktu puncak	detik
Ts	waktu tetap	detik
FLC	fuzzy logic controller	
PWM	Pulse Width Modulation	
NB	negative big	
NM	negative medium	
NS	negative small	
ZE	zero	
PS	positive small	
PM	positive medium	
PB	positive big	

BAB I

PENDAHULUAN

1.1 LATAR BELAKANG

Motor induksi merupakan jenis motor yang paling banyak digunakan di industri dibandingkan dengan motor listrik yang lain, karena jenis motor ini memiliki beberapa keuntungan diantaranya konstruksinya yang sederhana dan kokoh harga relatif murah dan tidak memerlukan pemeliharaan yang rumit. Di samping keunggulan diatas kelemahan dari motor induksi adalah sulitnya mengatur kecepatan karena karakteristik motor induksi itu sendiri sifatnya tidak linear. Pada prinsipnya motor induksi dioperasikan pada kecepatan yang konstan, bila beban berubah kecepatan motor juga akan berubah. Karena itu untuk mempertahankan agar kecepatan tetap konstan maka tegangan dan frekuensi harus diatur. Agar didapatkan unjuk kerja yang diharapkan dari motor induksi diperlukan suatu kontroler yang berupa sistem fuzzy. Algoritma menggunakan sistem ini cukup baik, tetapi penggunaan sistem kontrol ini tidak dapat mengatasi lingkungan kontrol dengan beban yang berubah-ubah atau nonlinear.

Dalam Perencanaan suatu sistem kontrol kecepatan motor induksi haruslah diperhatikan tingkat *performance* sistem dalam mengatur variabel kontrol tersebut. Penggunaan sistem *Fuzzy Logic Control* (FLC) dalam kinerjanya sangatlah baik dengan tingkat respon yang cepat dalam mengatasi kinerja sistem kontrol yang mengalami perubahan beban. Penggunaan *fuzzy logic control* adalah untuk mengantisipasi atau memperbaiki kesalahan (*error*) yang terjadi pada sistem (*plant*) yang diakibatkan oleh adanya perubahan beban. Untuk melengkapi proses kontrol kecepatan motor induksi, digunakan sebuah inverter PWM. Hal ini dimaksudkan untuk memberikan frekuensi variabel dengan konstanta tegangan dan frekuensi (V/F) dalam pengaturan kecepatan motor yang diberikan pada input motor.

1.2 PERUMUSAN DAN BATASAN MASALAH

Pengendalian membutuhkan ketangguhan dalam performansi kerja kendali yang diharapkan dapat menangani model *plant* sesulit apapun, dalam jangkauan apapun, dan dalam skala yang dapat direalisasikan secara mudah.

Karena itu, dibutuhkan suatu pengendalian pada *plant* proses tertentu yang dapat mengendalikan *plant* proses yang memiliki kesulitan dalam pemodelan secara matematik dan memiliki tingkat supervisi dan tingkat jangkauan yang beragam sampai pada lokasi yang tak dapat dicapai secara langsung oleh manusia. Dengan pengendalian yang menggunakan aturan *fuzzy*; diharapkan memudahkan penanganan *plant* proses.

Dalam pengerjaan tugas akhir ini, dilakukan pembatasan masalah sebagai berikut:

1. Tugas akhir ini hanya berupa simulasi yang hanya menampilkan kondisi *plant* dan pengendali dalam keadaan ideal.
2. Menekankan pada integrasi sistem *fuzzy* dan *plant* simulasi. Aspek performansi dan kuantitatif sistem memerlukan bahasan tersendiri.
3. Menentukan kontrol keluaran dari sistem tersebut.
4. Membandingkan respon sistem terhadap perubahan *setpoint* pada beban.
5. Motor induksi yang digunakan adalah motor induksi tiga fasa asimetri.
6. PWM inverter yang mensuplai motor adalah PWM inverter yang ideal.
7. Pengidera kecepatan atau posisi adalah pengidera yang ideal.
8. Pengukur besaran-besaran listrik yang digunakan adalah pengukur yang ideal.

1.3 TUJUAN PENULISAN

- Merancang desain konfigurasi sistem pengaturan kecepatan motor induksi dengan logika fuzzy.
- Simulasi desain sistem dengan perubahan masukan baik kecepatan dan torsi acuan.
- Mencari respon plant yang paling baik dengan berbagai masukan.
- Membuat simulasi unjuk kerja logika fuzzy pada motor induksi dengan Matlab dan simulink.

1.4 METODE PENELITIAN

Metode penelitian yang dilaksanakan dalam pengerjaan tugas akhir ini dapat dibagi menjadi tiga, yaitu:

- Studi literatur

Antara lain dilakukan dengan:

1. Mempelajari sistem pengendali yang sudah ada saat ini, dengan tujuan memahami konsep sistem tersebut.
2. Mempelajari referensi perangkat lunak *Matlab*, yang dapat digunakan sebagai agen pengendali *miniplant*.

- Penentuan spesifikasi

Berdasarkan studi literatur yang dilakukan sebelumnya, dilakukan proses penentuan spesifikasi lunak yang dibutuhkan sesuai dengan kriteria perancangan. Didalamnya termasuk juga penentuan bahasa pemrograman yang digunakan dalam sistem ini. Tahap ini menghasilkan spesifikasi perancangan.

- Simulasi sistem

Simulasi sistem akan menghasilkan data-data yang kita inginkan. Data-data itu berupa basis pengetahuan yang diubah kedalam bentuk tabel fungsional antara sinyal masukan (harga *error*) sebagai masukan dan sinyal kontrol sebagai keluaran. Membuat grafik dari tabel fungsional sinyal kontrol untuk melihat respon sistem. Dari perbandingan dalam bentuk tabel dan grafik dapat dibandingkan keunggulan dan kelemahan pada kedua sistem sehingga didapat kesimpulan pengendali mana yang paling baik yang digunakan dalam proses pengendalian sistem ini.

1.5 SISTEMATIKA PEMBAHASAN

Agar pembahasan masalah dapat dilakukan secara sistematis, maka penulisan dilakukan dengan susunan sebagai berikut:

Pada bab satu adalah pendahuluan, berisi latar belakang, perumusan masalah, tujuan penulisan, metode pengerjaan dan sistematika pembahasan. Bab dua menjelaskan konsep dasar teknologi kendali dalam dunia industri, berisi gambaran mengenai kecenderungan penerapan teknologi kendali dalam dunia industri. Bab tiga menguraikan perancangan dan realisasi sistem, berisi penentuan spesifikasi sistem, desain dasar komponen subsistem, dan realisasinya. Bab tiga menjelaskan pengujian dan analisis, berisi hasil pengujian dan analisis kinerja fungsional dan waktu respon sistem. Bab lima merupakan kesimpulan dan saran, berisi kesimpulan yang didapat selama pengerjaan tugas akhir ini dan saran mengenai hal-hal yang perlu diperhatikan untuk pengembangan selanjutnya.

BAB II

LANDASAN TEORI

2.1 MOTOR INDUKSI

Motor induksi merupakan motor yang paling umum digunakan pada berbagai peralatan industri. Popularitasnya karena rancangannya yang sederhana, murah dan mudah didapat, dan dapat langsung disambungkan ke sumber daya AC

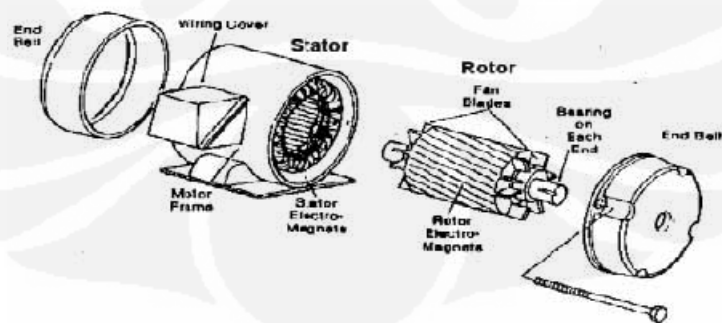
2.1.1 Komponen

Motor induksi memiliki dua komponen listrik utama seperti pada gambar 2.1[1]

Rotor. Motor induksi menggunakan dua jenis rotor:

- Rotor kandang tupai terdiri dari batang penghantar tebal yang dilekatkan dalam petak-petak *slots* paralel. Batang-batang tersebut diberi hubungan pendek pada kedua ujungnya dengan alat cincin hubungan pendek.
- Lingkaran rotor yang memiliki gulungan tiga fase, lapisan ganda dan terdistribusi. Dibuat melingkar sebanyak kutub stator. Tiga fase digulungi kawat pada bagian dalamnya dan ujung yang lainnya dihubungkan ke cincin kecil yang dipasang pada batang as dengan sikat yang menempel padanya.

Stator. Stator dibuat dari sejumlah *stampings* dengan *slots* untuk membawa gulungan tiga fase. Gulungan ini dilingkarkan untuk sejumlah kutub yang tertentu. Gulungan diberi spasi geometri sebesar 120 derajat.



Gambar 2.1 Bagian-bagian Motor Induksi

2.1.2 Klasifikasi motor induksi

Motor induksi dapat diklasifikasikan menjadi dua kelompok utama [1]:

- Motor induksi satu fase. Motor ini hanya memiliki satu gulungan *stator*, beroperasi dengan pasokan daya satu fase, memiliki sebuah rotor kandang tupai, dan memerlukan sebuah alat untuk menghidupkan motornya. Sejauh ini motor ini merupakan jenis motor yang paling umum digunakan dalam peralatan rumah tangga, seperti kipas angin, mesin cuci dan pengering pakaian, dan untuk penggunaan hingga 3 sampai 4 Hp.
- Motor induksi tiga fase. Medan magnet yang berputar dihasilkan oleh pasokan tiga fase yang seimbang. Motor tersebut memiliki kemampuan daya yang tinggi, dapat memiliki kandang tupai atau gulungan rotor (walaupun 90% memiliki rotor kandang tupai); dan penyalaan sendiri. Diperkirakan bahwa sekitar 70% motor di industri menggunakan jenis ini, sebagai contoh, pompa, kompresor, *belt conveyor*, jaringan listrik, dan *grinder*. Tersedia dalam ukuran 1/3 hingga ratusan Hp.

2.1.3 Kecepatan motor induksi

Motor induksi bekerja sebagai berikut. Listrik dipasok ke stator yang akan menghasilkan medan magnet. Medan magnet ini bergerak dengan kecepatan sinkron disekitar rotor. Arus rotor menghasilkan medan magnet kedua, yang berusaha untuk melawan medan magnet stator, yang menyebabkan rotor berputar. Walaupun begitu, didalam prakteknya motor tidak pernah bekerja pada kecepatan sinkron namun pada “kecepatan dasar” yang lebih rendah. Terjadinya perbedaan antara dua kecepatan tersebut disebabkan adanya “*slip*/geseran” yang meningkat dengan meningkatnya beban. *Slip* hanya terjadi pada motor induksi. Untuk menghindari slip dapat dipasang sebuah cincin geser/ *slip ring*, dan motor tersebut dinamakan “motor cincin geser/ *slip ring motor*”. Persamaan berikut dapat digunakan untuk menghitung persentase *slip*/geseran [1]:

$$S = \frac{N_s - N_r}{N_s} \times 100 \dots\dots\dots(2.1)$$

Dimana: N_s = kecepatan sinkron dalam rpm
 N_r = kecepatan dasar dalam rpm

Kecepatan medan-magnet putar adalah:

$$N_m = \frac{f \cdot 120}{p} \dots\dots\dots(2.2)$$

Keterangan : N_m = Kecepatan medan magnet (rpm)
 f = Frekuensi stator(Hz)
 p = Jumlah kutub motor(buah)

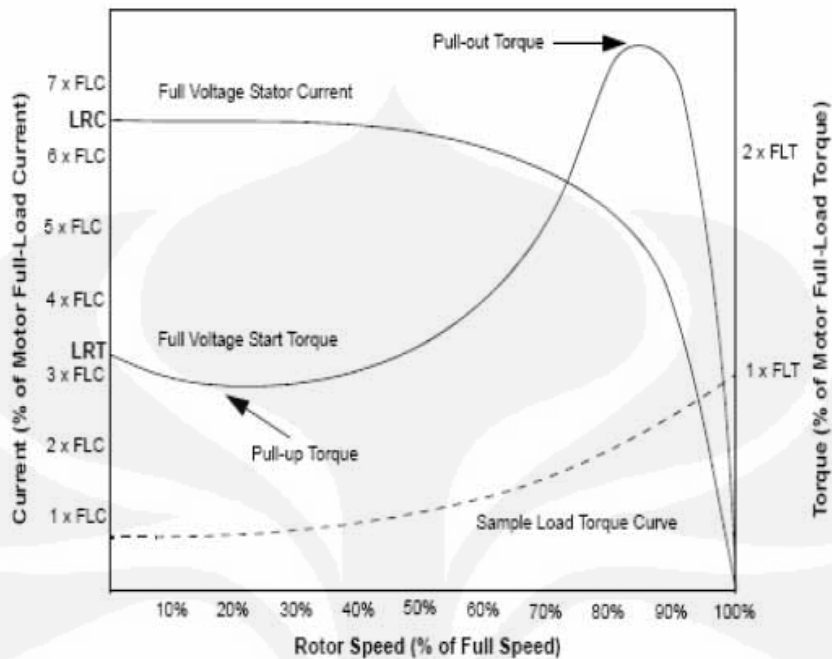
Pada waktu rotor berputar maka frekuensinya f' akan dipengaruhi oleh slip dengan persamaan:

$$f' = S \cdot f \dots\dots\dots(2.3)$$

2.1.4 Hubungan antara beban, kecepatan dan torque

Gambar 2.2 menunjukkan grafik *torque*-kecepatan motor induksi AC tiga fase dengan arus yang sudah ditetapkan. Bila motor [1]:

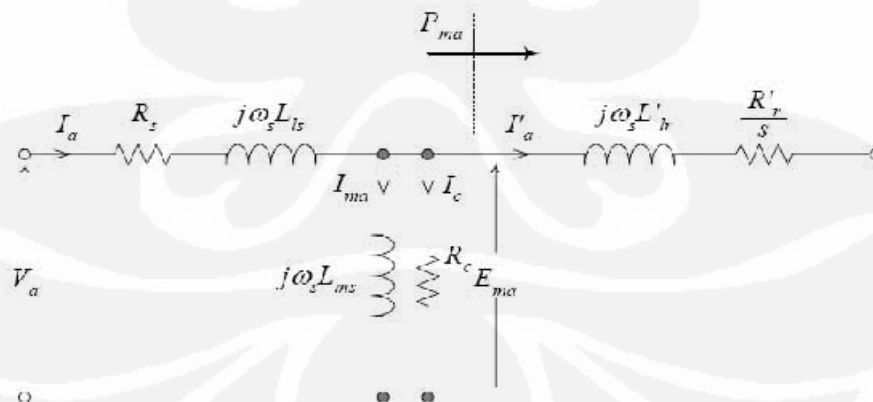
- Mulai menyala ternyata terdapat arus nyala awal yang tinggi dan *torque* yang rendah (*pull-up torque*).
- Mencapai 80% kecepatan penuh, *torque* berada pada tingkat tertinggi (*pull-out torque*) dan arus mulai turun.
- Pada kecepatan penuh, atau kecepatan sinkron, arus *torque* dan *stator* turun ke nol.



Gambar 2.2 Grafik Torsi – Kecepatan Motor Induksi tiga fasa [10]

2.1.5 Rangkaian Pengganti Motor Induksi

Motor induksi pada dasarnya merupakan transformator dengan kumparan sekunder (rotor) yang berputar, oleh karena itu rangkaian pengganti motor induksi serupa sebagaimana transformer, hal ini bisa dilihat pada gambar 2.3.



Gambar 2.3. Rangkaian ekivalen motor induksi per-fasa

Parameter motor induksi dalam notasi vector sebagai berikut:[1]

Persamaan tegangan stator:

$$\bar{V}_s = \bar{R}_s \bar{I}_s + \frac{d}{dt} \bar{\lambda}_s + J \omega_s \bar{\lambda}_s \dots\dots\dots(2.4)$$

$$\bar{I}_s = i_{ds} + J i_{qs} \dots\dots\dots(2.5)$$

$$\bar{\lambda}_s = \lambda_{ds} + J \lambda_{qs} \dots\dots\dots(2.6)$$

Persamaan Tegangan Rotor

$$\bar{V}_r = R_r \bar{I}_r + \frac{d}{dt} \bar{\lambda}_r + J(\omega_s - \omega_r) \bar{\lambda}_r \dots\dots\dots(2.7)$$

$$\bar{I}_r = i_{dr} + J i_{qr} \dots\dots\dots(2.8)$$

$$\bar{\lambda}_r = \lambda_{dr} + J \lambda_{qr} \dots\dots\dots(2.9)$$

Pada motor induksi dengan rotor sangkar, tegangan yang terdapat pada rotor sama dengan nol ($V_r = 0$)

- *Flux Linkage*

Flux linkage didefinisikan sebagai besarnya medan putar (flux) pada kumparan stator dan rotor motor.

$$\bar{\lambda}_s = L_s \cdot \bar{I}_s + M \cdot \bar{I}_r \dots\dots\dots(2.10)$$

$$\bar{\lambda}_r = L_r \cdot \bar{I}_r + M \cdot \bar{I}_s \dots\dots\dots(2.11)$$

- *Persamaan Torsi Elektromagnetik*

$$T_e = \left(\frac{3}{2}\right) \left(\frac{P}{2}\right) \left(\frac{1}{\omega b}\right) (\varphi_{qr} \cdot i_{dr} - \varphi_{dr} \cdot i_{qr}) \dots\dots\dots(2.12)$$

$$\bar{V}_s = R_s \bar{I}_s + L_s \frac{d}{dt} \bar{I}_s + M \frac{d}{dt} \bar{I}_r + j \omega_s L_s \bar{I}_s + j \omega_s M \bar{I}_r \dots\dots\dots(2.13)$$

$$\bar{V}_r = R_r \bar{I}_r + L_r \frac{d}{dt} \bar{I}_r + M \frac{d}{dt} \bar{I}_s + j(\omega_s - \omega_r) L_r \bar{I}_r + j(\omega_s - \omega_r) M \bar{I}_s \dots\dots\dots(2.14)$$

apabila persamaan 2.13 dan persamaan 2.14 dinyatakan kedalam sumbu d-q serta dengan memasukan tegangan rotor ($V_r = 0$), maka akan didapatkan besaran

tegangan stator dan tegangan rotor yang merupakan fungsi dari arus stator dan rotor dalam bentuk metrik sebagai berikut:

$$\begin{pmatrix} V_{ds} \\ V_{qs} \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} R_s + pL_s & -\omega_s L_s & pM & -\omega_s M \\ \omega_s L_s & R_s + pL_s & \omega_s M & pM \\ pM & -(\omega_s - \omega_r)M & R_r + pL_r & -(\omega_s - \omega_r)L_r \\ -\omega_r M & pM & (\omega_s - \omega_r)L_r & R_r + pL_r \end{pmatrix} \begin{pmatrix} i_{ds} \\ i_{qs} \\ i_{dr} \\ i_{qr} \end{pmatrix} \dots\dots(2.15)$$

dimana $p = d/dt$

jika ditinjau pada koordinat stationer ($\omega_s = 0$),

maka persamaan diatas akan berubah menjadi

$$\begin{pmatrix} V_{ds} \\ V_{qs} \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} R_s + pL_s & 0 & pM & 0 \\ 0 & R_s + pL_s & 0 & pM \\ pM & \omega_r M & R_r + pL_r & -\omega_r L_r \\ -\omega_r M & pM & -\omega_r L_r & R_r + pL_r \end{pmatrix} \begin{pmatrix} i_{ds} \\ i_{qs} \\ i_{dr} \\ i_{qr} \end{pmatrix} \dots\dots\dots(2.16)$$

2.2 LOGIKA FUZZY

Teori himpunan *fuzzy* diperkenalkan pertama kali oleh Lotfi A. Zadeh pada tahun 1965 [2]. Beberapa alasan pembenaran mengapa himpunan *fuzzy* dipergunakan :

- Adanya kesulitan dalam mendapatkan deskripsi yang persis dan lengkap dari sistem yang ada di dunia nyata karena itu diperlukan suatu pendekatan aproksimasi untuk menangani *fuzziness* yang ada.
- Informasi berupa pengetahuan dan pengalaman manusia mempunyai peranan penting dalam mengenali perilaku sistem di dunia nyata karena itu dibutuhkan suatu teori yang sistematis untuk memformulasikan pengetahuan manusia bersama dengan informasi lain berupa model matematis dan hasil pengukuran.

2.2.1 Himpunan Fuzzy

Himpunan *fuzzy* A pada semesta X dinyatakan sebagai himpunan pasangan berurutan (*set of ordered pairs*) baik diskrit maupun kontinu :

$$A = \{(x, \mu_A(x)) | x \in X\} \dots\dots\dots(2.17)$$

dimana $\mu_A(x)$ fungsi keanggotaan himpunan *fuzzy* A . Fungsi keanggotaan memetakan setiap $x \in X$ pada suatu nilai antara $[0,1]$ yang disebut derajat keanggotaan (*membership grade* atau *membership value*).

Support dari himpunan *fuzzy* A adalah kumpulan semua titik $x \in X$ yang memberikan nilai $\mu_A(x) > 0$, atau

$$Support(A) = \{x | \mu_A(x) > 0\} \dots\dots\dots(2.18)$$

Core dari himpunan *fuzzy* A adalah kumpulan semua titik $x \in X$ yang memberikan nilai $\mu_A(x) = 1$, atau

$$Core(A) = \{x | \mu_A(x) = 1\} \dots\dots\dots(2.19)$$

Titik *crossover* dari himpunan *fuzzy* A adalah titik dimana $\mu_A(x) = 0.5$, atau

$$Crossover(A) = \{x | \mu_A(x) = 0.5\} \dots\dots\dots(2.20)$$

Fungsi *singleton* adalah himpunan *fuzzy* yang memiliki *support* pada satu titik $x \in X$ dengan $\mu_A(x) = 1$.

2.2.2 Fungsi Keanggotaan

Dalam sistem *fuzzy* banyak dikenal bermacam-macam fungsi keanggotaan (*membership function*). Beberapa fungsi keanggotaan yang sering digunakan adalah:

- Fungsi keanggotaan segitiga

$$\text{segitiga}(x, a, b, c) = \begin{cases} 0, & x \leq a. \\ \frac{x-a}{b-a}, & a \leq x \leq b. \\ \frac{c-x}{c-b}, & b \leq x \leq c. \\ 0, & c \leq x. \end{cases} \dots\dots\dots(2.21)$$

atau dengan menggunakan min dan max

$$\text{segitiga}(x, a, b, c) = \max \left(\min \left(\frac{x-a}{b-a}, \frac{c-x}{c-b} \right), 0 \right) \dots\dots\dots(2.22)$$

- Fungsi keanggotaan trapesium

$$\text{trapesium}(x, a, b, c, d) = \begin{cases} 0, & x \leq a. \\ \frac{x-a}{b-a}, & a \leq x \leq b. \\ 1, & b \leq x \leq c. \\ \frac{d-x}{d-c}, & c \leq x \leq d. \\ 0, & d \leq x. \end{cases} \dots\dots\dots(2.23)$$

atau dengan menggunakan min dan max

$$\text{trapesium}(x, a, b, c, d) = \max \left(\min \left(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c} \right), 0 \right) \dots\dots\dots(2.24)$$

- Fungsi keanggotaan *Gaussian*

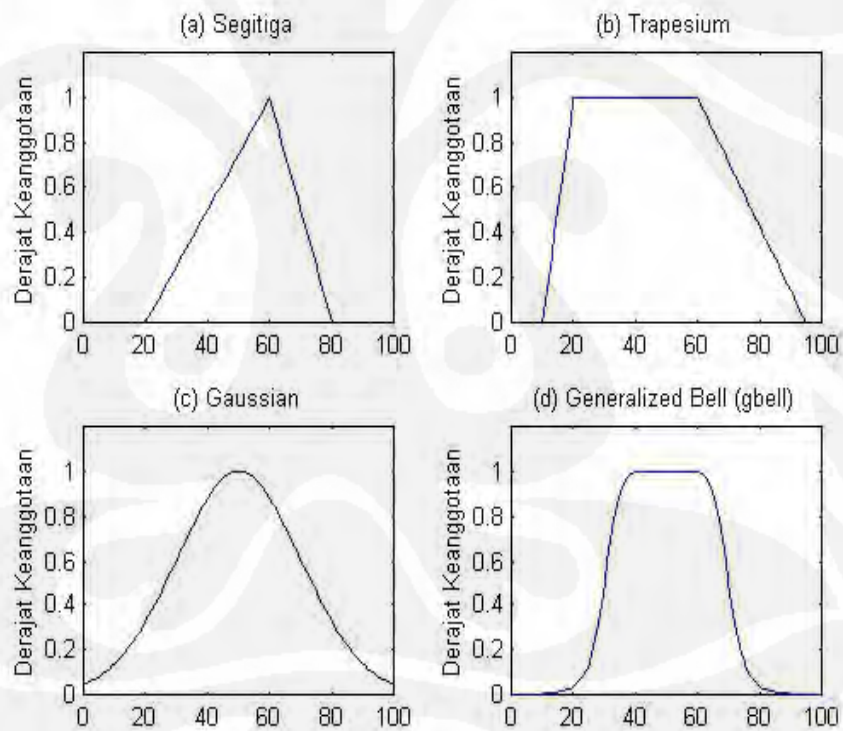
$$\text{gaussian}(x; c, \sigma) = e^{-\frac{1}{2} \left(\frac{x-c}{\sigma} \right)^2} \dots\dots\dots(2.25)$$

- Fungsi keanggotaan *Generalized Bells*

$$gbell(x, a, b, c) = \frac{1}{1 + \left| \frac{x-c}{a} \right|^{2b}} \dots\dots\dots(2.26)$$

dengan x adalah *variabel* sedangkan a, b , dan c merupakan bilangan ril. Untuk fungsi *Gaussian*, nilai b adalah positif, jika b negatif maka bentuk fungsi keanggotaan menjadi terbalik.

Keempat fungsi keanggotaan dapat dilihat pada Gambar 2.2.



Gambar 2.4 Contoh Fungsi keanggotaan : (a) segitiga($x;20,60,80$);
 (b) trapesium($x;10,20,60,95$); (c) *gaussian*($x;50,20$); (d) *gbell*($x;20,4,50$)

2.2.3 Operator dan Operasi Himpunan Fuzzy

Pada himpunan *fuzzy* dikenal dua kelompok operator yang sering digunakan yaitu kelompok *triangular norm (t-norm)* dan *triangular conorm (t-conorm)* atau biasa disebut juga *s-norm*.

T-Norm (Fuzzy Intersection)

T-Norm merupakan operator yang menampilkan operasi irisan (*intersection*) dalam himpunan *fuzzy*.

Misalkan *t* operator *t-norm* : $[0,1] \times [0,1] \rightarrow [0,1]$

$$t[\mu_A(x), \mu_B(x)] = \mu_{A \cap B}(x) \dots\dots\dots(2.27)$$

Operator yang sering dipakai dari sekian banyak operator *t-norm* adalah

$$\text{Produk aljabar : } t[\mu_A(x), \mu_B(x)] = \mu_A(x) \cdot \mu_B(x) \dots\dots\dots(2.28)$$

$$\text{Minimum : } t[\mu_A(x), \mu_B(x)] = \min[\mu_A(x), \mu_B(x)] \dots\dots\dots(2.29)$$

S-Norm (Fuzzy Union)

S-Norm adalah operator yang menampilkan operasi gabungan (*union*) dalam himpunan *fuzzy*.

Misalkan *s* operator *s-norm* : $[0,1] \times [0,1] \rightarrow [0,1]$

$$s[\mu_A(x), \mu_B(x)] = \mu_{A \cup B}(x) \dots\dots\dots(2.30)$$

Operator yang sering dipakai dari sekian banyak operator *s-norm* adalah

$$\text{Jumlah aljabar : } s[\mu_A(x), \mu_B(x)] = \mu_A(x) + \mu_B(x) - \mu_A(x) \cdot \mu_B(x) \dots\dots\dots(2.31)$$

$$\text{Maksimum : } s[\mu_A(x), \mu_B(x)] = \max[\mu_A(x), \mu_B(x)] \dots\dots\dots(2.32)$$

2.2.4 Basis Aturan

Basis aturan (*Rule Base*) merupakan inti dari sistem *fuzzy*. Basis aturan berisi sekumpulan aturan *if-then* dalam bentuk

$$R^i : \text{if } x_1 \text{ is } F_1^i \text{ and } \dots \text{ and } x_n \text{ is } F_n^i \text{ then } y \text{ is } G^i$$

dengan F dan G merupakan himpunan *fuzzy* pada U dan V , x dan y adalah variabel linguistik, i menyatakan banyaknya jumlah aturan.

2.2.5 Inferensi Fuzzy

Prinsip dasar dari inferensi *fuzzy* (*Fuzzy Inference*) adalah menggunakan kumpulan aturan yang terdapat dalam basis aturan untuk memetakan himpunan *fuzzy* pada semesta pembicaraan masukan menjadi himpunan *fuzzy* lain yang terdapat pada semesta pembicaraan keluaran. Pada prinsipnya inferensi *fuzzy* menginterpretasikan aturan *if-then* dengan menggunakan komposisi *sup-star*.

Pada Basis Aturan, masukan x dipetakan menjadi keluaran y . Aturan *if-then* diinterpretasikan sebagai implikasi fuzzy $F_1^i x \dots x F_n^i \rightarrow G^i$. Terdapat banyak sekali model interpretasi implikasi yang telah dikembangkan. Salah satu yang sederhana dan mudah digunakan adalah implikasi Mamdani $p \rightarrow q = t[p, q]$. Sedangkan interpretasi atas kata hubung *and* adalah irisan yang dapat menggunakan operator *t-norm*.

Jika diberikan himpunan A' pada U yang menjadi masukan inferensi dari Aturan maka dapat ditentukan derajat keanggotaan B pada keluaran V dengan menggunakan komposisi *sup-star*.

$$\mu_B^i(y) = \sup[\mu_{A'}(x) * \mu_Q(x, y)] \dots \dots \dots (2.33)$$

Q adalah implikasi fuzzy $F_1^i x \dots x F_n^i \rightarrow G^i$. Tanda "*" adalah operator *t-norm*. Derajat keanggotaan B pada V untuk m jumlah aturan dinyatakan dengan

$$\mu_B(y) = s[\mu_B^1(y) \dots \mu_B^m(y)] \dots\dots\dots(2.34)$$

2.2.6 Fuzzifikasi

Fuzzifikasi (*Fuzzification*) berfungsi untuk memetakan titik *crisp*(*non fuzzy*) $x \in U$ ke dalam himpunan fuzzy A' pada U . Ada dua macam fuzzifikasi yaitu : fuzzifikasi *singleton* dan fuzzifikasi *nonsingleton*. Pada fuzzifikasi *singleton*, titik *crisp* $x \in U$ dipetakan menjadi himpunan fuzzy yang *supportnya* hanya pada satu titik x . Pada fuzzifikasi *nonsingleton*, titik *crisp* $x \in U$ dipetakan menjadi himpunan fuzzy yang *supportnya* lebih dari satu titik. Fuzzifikasi yang banyak digunakan adalah fuzzifikasi *singleton*. Dengan menggunakan fuzzifikasi *singleton* himpunan fuzzy A' hanya mempunyai *support* hanya pada satu titik x .

2.2.7 Defuzzifikasi

Defuzzifikasi (*Defuzzification*) memetakan besaran yang berupa himpunan fuzzy menjadi titik *crisp*. Defuzzifikasi dibutuhkan dalam penerapan sistem fuzzy karena yang digunakan dalam aplikasi adalah besaran *crisp*. Beberapa metode dari sekian banyak metode yang digunakan dalam defuzzifikasi adalah :

- Metoda defuzzifikasi maksimum

$$y = \arg \sup[\mu_B(y)] \dots\dots\dots(2.35)$$

- Metoda defuzzifikasi rata-rata tengah

$$y = \frac{\sum_{i=1}^m \bar{y}^i \cdot \mu_{B^i}(\bar{y}^i)}{\sum_{i=1}^m \mu_{B^i}(\bar{y}^i)} \dots\dots\dots(2.36)$$

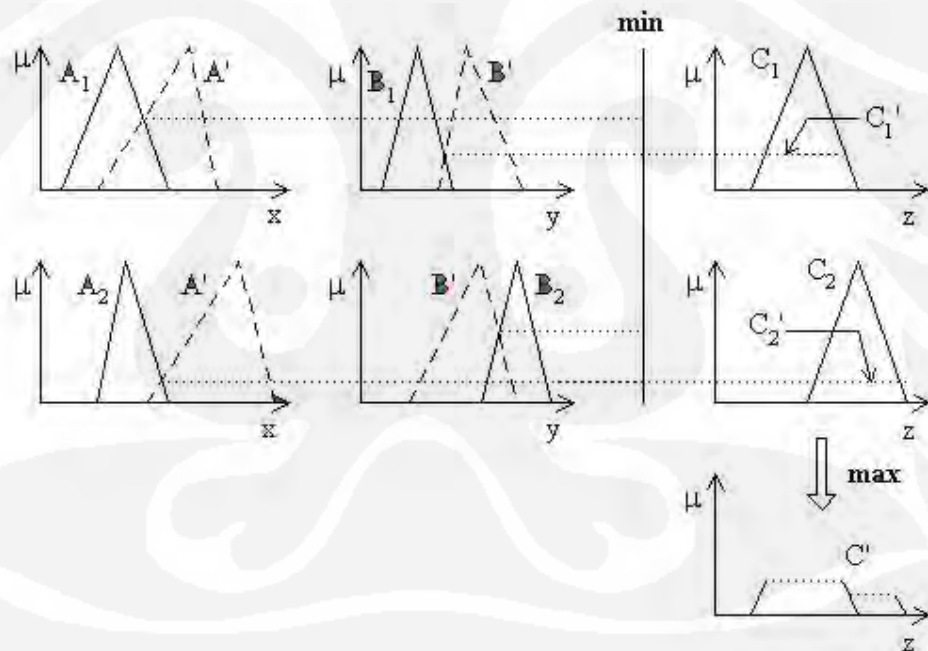
dengan $\mu_{B^i} \cdot \mu_{B^i}$ masing-masing dinyatakan oleh Persamaan sebelumnya \bar{y} adalah titik dimana derajat keanggotaan himpunan fuzzy G menjadi maksimum.

2.2.8 Sistem Fuzzy Mamdani

Sistem *fuzzy* yang paling banyak dikenal adalah sistem *fuzzy* Mamdani. Sistem *fuzzy* Mamdani terdiri dari [3] :

- Basis aturan yang berisi sekumpulan aturan if-then
- Basis data yang mendefinisikan fungsi keanggotaan himpunan *fuzzy*
- Mesin inferensi yang melakukan operasi inferensi
- Fuzzifikasi dan defuzzifikasi

Sistem *fuzzy* Mamdani menggunakan basis aturan seperti pada Persamaan sebelumnya. Pada sistem *fuzzy* ini terdapat banyak kebebasan dalam memilih jenis fuzzifikasi, defuzzifikasi, mesin inferensi (*inference engine*) beserta operator yang sesuai dengan masalah yang dihadapi dapat dilihat pada Gambar 2.5 [3].



Gambar 2.5. Mekanisme inferensi Fuzzy Mamdani

Di bawah ini diilustrasikan bagaimana sistem inferensi *fuzzy* Mamdani memberikan *reasoning* terhadap masukannya.

Pernyataan : x adalah A' dan y adalah B'

Aturan 1 : jika x adalah A₁ dan y adalah B₁ maka z = C₁

Aturan 2 : jika x adalah A₂ dan y adalah B₂ maka z = C₂

Kesimpulan : z adalah C'

Prinsip dasar dari bagian mesin inferensi (pengambil keputusan) pada sistem *fuzzy* adalah menggunakan kumpulan aturan yang terdapat dalam basis aturan untuk memetakan kumpulan himpunan *fuzzy* yang ada pada semesta pembicaraan masukan, ke himpunan *fuzzy* lain yang terdapat pada semesta pembicaraan keluaran sistem, yang dalam beberapa aspek seperti manusia pada waktu mengambil keputusan.

Selama pengambilan keputusan, tiap-tiap aturan *fuzzy* melakukan operasi-operasi di bawah ini, yaitu:

1. Menentukan seberapa besar tingkat keanggotaan himpunan *fuzzy* pada sistem variabel masukan.
2. Melakukan perhitungan untuk tiap-tiap basis aturan.
3. Penggunaan hubungan yang dipakai variabel masukan pada bagian sebab (*antecedent*), seperti AND, OR.
4. Aplikasi dari operator implikasi *fuzzy*
5. Mengagregasikan keluaran untuk keseluruhan masing basis aturan.
6. Melakukan defuzzifikasi dari hasil agregasi.

2.3 MODULASI LEBAR PULSA (PWM)

Kebutuhan untuk mengatur kecepatan motor AC dengan menggunakan pengaturan frekuensi dapat diterapkan untuk tegangan dan arus. Penerapan ini harus menghasilkan distorsi yang sekecil mungkin. Hal ini dapat di atasi apabila inverter yang digunakan dapat menghasilkan gelombang sinusoidal.

Kebutuhan yang sangat ideal pada saat ini adalah dengan menerapkan suatu sistem penyelidikan terhadap bentuk gelombang tegangan yang di namakan teknik Modulasi Lebar Pulsa Sinusoidal atau lebih dikenal dengan Pulse Width Modulation (PWM). Pada dasarnya komponen dalam inverter yang menghasilkan gelombang PWM ini dapat mengontrol frekuensi dan besarnya sudut penyulutan. Harmonik merupakan salah satu masalah dalam sistem pengontrolan motor induksi, untuk itu dalam sistem PWM harmonik dapat di kurangi tetapi tidak total hilang sama sekali.

Pengaturan yang dilakukan oleh PWM ini dapat berupa phasa tunggal dan tiga phasa. PWM yang di gunakan untuk sistem ESP ini adalah pengaturan tiga phasa.

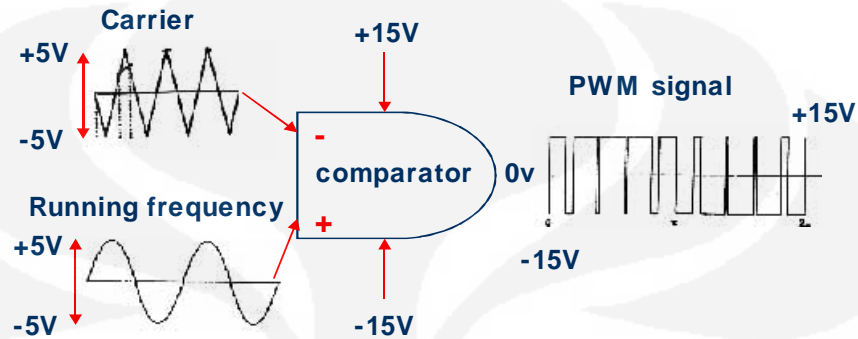
Pada pengaturan tiga phasa dengan menggunakan PWM pada dasarnya memiliki perbedaan bentuk antara sumber tegangan yang tidak menggunakan sistem PWM. Sebuah inverter yang tidak menghasilkan PWM menggunakan sebuah penyearah yang diatur untuk memberikan suplai DC yang dapat diatur dari masukan DC tersebut ke DC link (berupa rangkaian kapasitor). Tingkatan tegangan dari tegangan keluaran inverter tersebut dikontrol dengan mengatur tegangan VDC link, dimana frekuensi dikontrol secara sendiri oleh gate dari inverter switching.

Sebuah inverter PWM menggunakan rangkaian penyearah yang terdiri dari dioda-dioda untuk memberikan tegangan yang tetap VDC di DC link. Frekuensi dan tegangan inverter dikontrol oleh gate dengan inverter switching. Secara lengkap digambarkan seperti dibawah ini. Ketika tegangan keluaran dari penyearah tidak murni DC maka diperlukan filter untuk menyerap faktor ripple.

Penggunaan tegangan DC tetap maksudnya bahwa beberapa inverter dapat beroperasi secara simultan dari sumber DC yang sama. Pada saat tegangan kecil penggunaan transistor (lebih baik dari thyristor) membolehkan switching dengan cepat dan arus cepat dan respon dari transient torque dibanding dengan inverter yang tidak menghasilkan pulsa PWM.

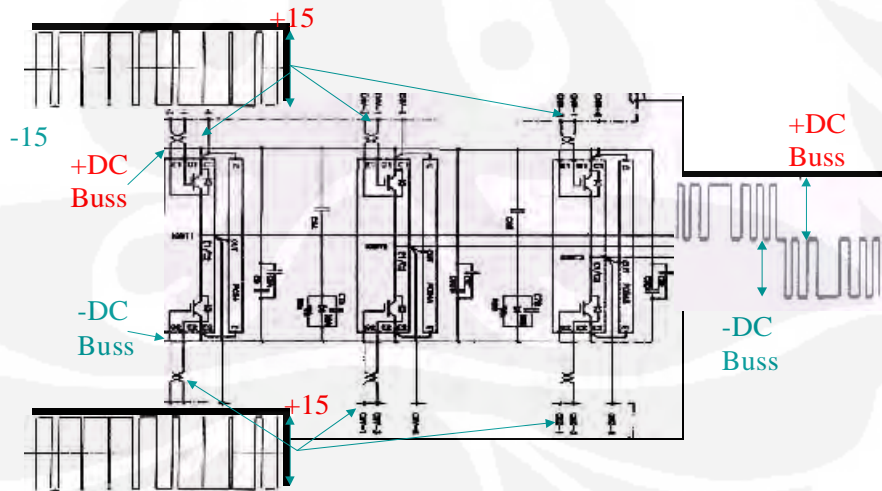
Pembuatan PWM didasari atas dua komponen penting, yaitu carrier frekuensi dan running frekuensi. Gabungan atau kombinasi dari keduanya akan membentuk suatu pulsa dimana pulsa tersebut dapat diatur dengan menaikkan

dan menurunkan banyaknya carrier frekuensi yang di berikan ke sistem dan mengatur target dari running frekuensi yang diinginkan. Hal ini dapat dijelaskan pada gambar 2.8 [7]



Gambar 2.6 Diagram proses pembuatan sinyal ke Gate pada IGBT

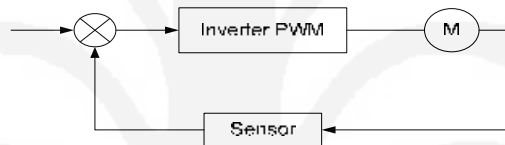
Dari gambar 2.8 diatas maka akan dilanjutkan pada proses pada gambar 2.9 di bawah ini [7]



Gambar 2.7 Diagram pembentukan sinyal PWM

2.3.1 Kontrol Kecepatan Motor dengan PWM inverter

Untuk mengatur kecepatan motor induksi dengan menggunakan inverter PWM sebagai suplai daya yang mengubah tegangan searah menjadi tegangan variabel dan frekuensi variabel dalam rangka memperoleh kecepatan yang sesuai dengan perubahan beban [7]. Gambar 2.7 dibawah ini menunjukkan sebuah sistem pengaturan kecepatan motor induksi menggunakan inverter PWM.



Gambar 2.8 Sistem Pengaturan Kecepatan Motor Menggunakan PWM

2.3.2 IGBT Inverter

IGBT inverter atau insulated gate bipolar transistor adalah teknologi inverter yang dapat meminimalisasi power loss dan meningkatkan switching PWM sehingga menghasilkan torsi start yang tinggi namun noise rendah. IGBT merupakan implementasi semikonduktor yang diatur oleh sinyal gate.

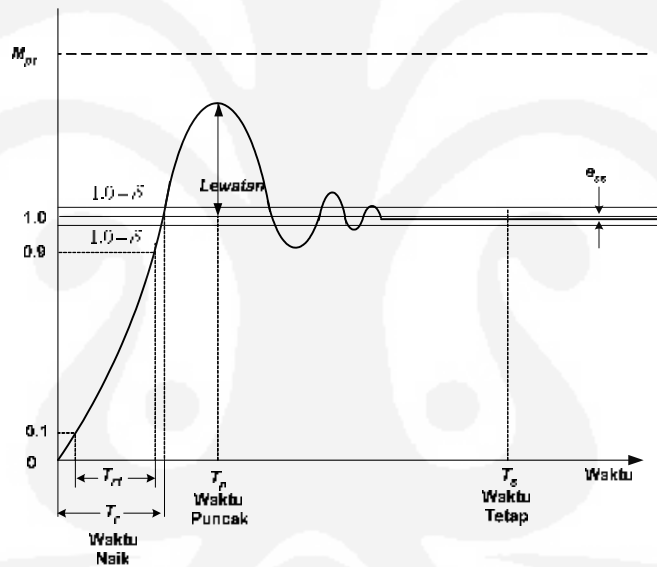
BAB III

PERANCANGAN SISTEM

3.1 DESAIN PLANT

Perancangan yang diinginkan untuk *respon plant*:

- *Overshoot* $\leq 0.5\%$
- *Settling time* (T_s) ≤ 4 detik
- *Steady-state error* $\leq 2.63\%$ untuk masukan adalah unit *step*



Gambar 3.1. Respon Sistem dari suatu Plant

Ukuran penampilan yang standard biasanya, didefinisikan dengan dinyatakan dalam *respon step* seperti yang ditunjukkan pada gambar 3.1. Cepatnya suatu tanggapan diukur dengan suatu variabel yang dinamakan waktu naik (*rise time*) T_r dan waktu puncak (*peak time*). Untuk sistem yang kurang teredam dengan suatu *overshoot*, waktu naik 0 sampai 100% merupakan suatu tolak ukur yang berguna. Bila sistemnya terlalu teredam, maka waktu puncaknya tak didefinisikan, dan waktu naik 10-100% lazimnya digunakan.

Kemiripan respon sebenarnya terhadap *respon step* dapat diukur dalam suatu prosentase *overshoot*, dan waktu tetap (*settling-time*). Prosentase *overshoot*, OS, dapat didefinisikan sebagai berikut [12]:

$$OS(\%) = e^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}} \times 100\% \dots\dots\dots(3.1)$$

Untuk suatu masukan *step* satuan, dimana ζ adalah faktor redaman. Waktu tetap (*settling-time*), T_s , didefinisikan sebagai waktu yang dibutuhkan oleh sistem supaya tetap harganya, dan berubah sebanyak suatu prosentase δ tertentu dari amplitudo masukan. Jalur (*band*) $\pm\delta$ ini tampak pada gambar 3.1. Untuk sistem tingkat kedua dengan tetapan redaman untuk loop tertutup $\xi\omega_n$, responnya akan tetap dalam range 2% setelah 4 kali tetapan waktu, atau [12]:

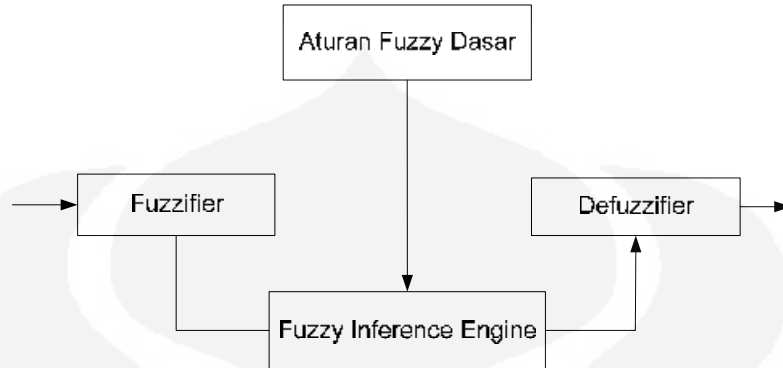
$$T_s = 4\tau = \frac{4}{\xi\omega_n} \dots\dots\dots(3.2)$$

Karena itu dapat didefinisikan waktu tetap sebagai empat kali tetapan waktu respons dominan. Akhirnya, kesalahan keadaan mantap suatu sistem dapat diukur dengan suatu tanggapan step seperti diperlihatkan gambar 3.1. Karena itu, tanggapan sistem dapat dilukiskan dengan:

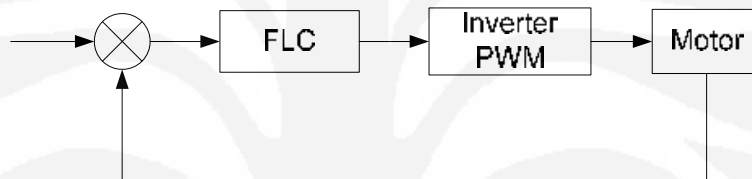
1. Kecepatan tanggapan, T_r dan T_p ;
2. Dekatnya tanggapan terhadap harga ζ dan T_s yang dikehendaki.
3. *Overshoot*

3.2 STRUKTUR PENGATURAN LOGIKA FUZZY

Struktur pengaturan logika fuzzy (*Fuzzy Logic Control*) menggabungkan aspek pendefinisian himpunan *fuzzy* dengan aspek logika *fuzzy*, dengan menggabungkan kedua sistem tersebut, didapatkan sistem yang andal yang dapat merepresentasikan bagaimana cara kerja operator manusia dengan prosedur perancangan tertentu. Kedua aspek diterapkan pada masukan dan keluaran untuk membentuk hasil perancangan yang berbentuk suatu algoritma aturan *fuzzy*, perancangan pengaturan logika *fuzzy* selama ini tidak mempunyai prosedur yang kaku, hal ini disebabkan oleh fleksibilitas pengatur logika *fuzzy* terhadap berbagai macam *plant*. Secara umum pengatur logika *fuzzy* mempunyai empat buah blok seperti gambar 3.2.



Gambar 3.2. Konfigurasi sistem logika fuzzy



Gambar 3.3. Diagram blok pengaturan kecepatan motor induksi

Perancangan pengendali logika *fuzzy* memiliki struktur dasar umum yang terdiri dari unit fuzzifikasi, mekanisme inferensi fuzzy, basis pengetahuan, dan unit defuzzifikasi yang ditunjukkan oleh gambar. 3.2 fungsi masing-masing unit tersebut adalah:

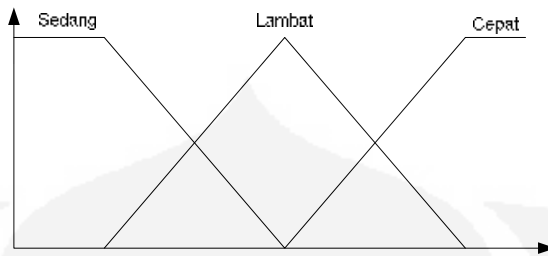
- Unit fuzzifikasi, memetakan sinyal input berupa yang bersifat *crisp* (*nonfuzzy*) kedalam semesta pembicaraan himpunan fuzzy dengan menggunakan operator fuzzifikasi (*fuzzifier*).
- Basis pengetahuan, meliputi basis data dan aturan dasar yang mendefinisikan himpunan fuzzy atas sinyal input dan sinyal output yang digunakan. Kemudian menyusunnya dalam perangkat aturan kendali fuzzy yang dikarakteristikan dengan istilah-istilah linguistik serta didasarkan pada pengetahuan ahli sebagai aturan sebab-akibat atau pernyataan jika-maka (*if-then*).
- Mekanisme inferensi fuzzy, memberikan aksi kendali fuzzy dengan menggunakan implikasi fuzzy dan kesimpulan aturan kendali fuzzy.

- Unit defuzzifikasi, memetakan kembali nilai inferensi yang masih dalam bentuk aksi kendali fuzzy menjadi sinyal output yang bersifat *crisp* dengan menggunakan operator defuzzifikasi (*defuzzifier*). Dari unit ini dihasilkan sebuah aksi kendali *crisp* yang paling baik mewakili keluaran inferensi fuzzy yang siap digunakan untuk mengendalikan sistem pada *plant*.

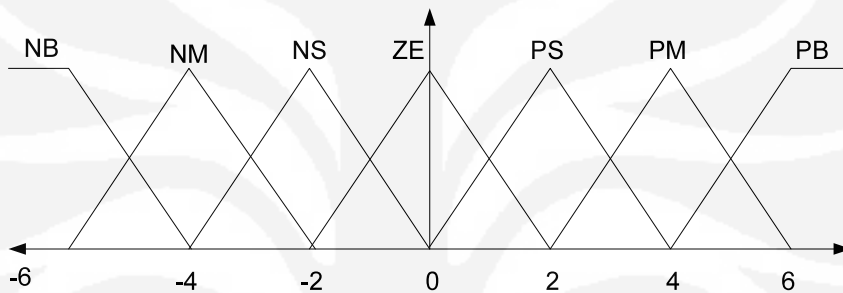
Penerapan fuzzifikasi pada suatu sistem pengaturan, besaran suatu masukan yang diperoleh dari *plant* melalui sensor selalu merupakan *crisp* yang sifatnya tidak pasti dan kuantitatif, sedangkan untuk pengolahan data dalam pengaturan logika fuzzy didasarkan pada teori himpunan fuzzy yang menggunakan variabel linguistik yang bersifat fuzzy.

Pada tahap awal pengaturan logika fuzzy diperlukan adanya fuzzifikasi yang dilakukan *fuzzifier*. Fuzzifikasi dapat dikatakan sebagai pemetaan masukan kemestaan himpunan fuzzy. Metode yang sering dipakai adalah dengan memperlakukan masukan *crisp* yang diperoleh sebagai fuzzy tunggal (*singlelition fuzzy*). Untuk basis data berfungsi untuk mendefinisikan himpunan fuzzy dari sinyal masukan dan sinyal keluaran yang akan digunakan oleh variabel linguistik dalam aturan dasar. Sedangkan perencanaan basis data meliputi kuantisasi dan normalisasi yaitu mendiskretkan semesta pembicaraan yang kontinyu ke dalam sejumlah segmen tertentu yang disebut level kuantitas. Level- level ini membentuk pendukung himpunan fuzzy secara berhingga atau semesta pembicaraan bersifat *crisp*. Pendefinisian ini dinyatakan dalam bentuk tabulasi. Pemilihan jumlah level kuantitas sangat mempengaruhi kepekaan pengaturan logika fuzzy terhadap masukan dan keluaran aksi pengaturan pada output. Semakin banyak level kuantitas yang dihasilkan semakin peka pengaturan logika fuzzy dan semakin halus aksi atur. Pembagian ruang masukan dan keluaran: membagi-bagi semesta pembicaraan atas nilai variabel linguistik.

Pemilihan fungsi keanggotaan yaitu pembagian ruangan dilakukan dengan tujuh nilai linguistik dari fungsi keanggotaan (*Membership Function*) dari logika fuzzy. Untuk penggunaan algoritma pengendalian fuzzy dengan bahasa natural, NB, NM, NS, ZE, PS, PM, PB yang berbentuk segitiga seperti pada gambar 3.5.



Gambar 3.4. Membership Function



Gambar 3.5. Variabel Input Q_e dan ΔQ_e

3.3 PERANCANGAN UNIT PENGENDALI LOGIKA FUZZY

Pada penelitian ini perancangan pengendali logika fuzzy dilakukan dengan bantuan perangkat lunak MATLAB versi 7.1 khususnya fuzzy editor. Metode inferensi yang digunakan adalah MAX-MIN, sedangkan defuzifikasinya menggunakan metode *center of area* (COA). Dalam merancang unit pengendali logika fuzzy, yang perlu diperhatikan bahwa masukkan unit pengendali fuzzy adalah *error* dan perubahan *error*, yang dinyatakan dengan:

$$e(k) = sp - y(k) \dots\dots\dots(3.3)$$

$$\Delta e(k) = e(k) - e(k - 1) \dots\dots\dots(3.4)$$

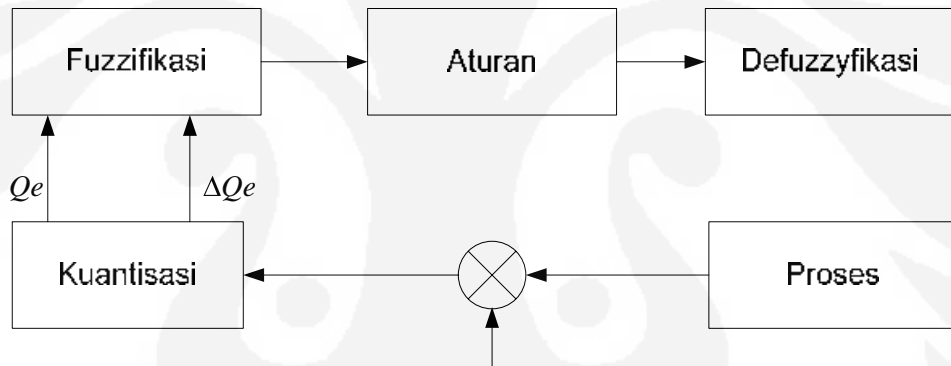
dimana sp dan y adalah *setpoint* dari output sistem. Sedangkan indeks k dan $k-1$ adalah urutan kejadian dari pencuplikan data sistem. Error (e) dan perubahan error yang terjadi selama sistem beroperasi kedalam sinyal yang difuzzifikasikan agar dapat menunjukkan pada tingkat keanggotaan. Kemudian proses perubahan sinyal masukkan dalam hal ini error dan delta error dikuantisasikan dahulu untuk

dipetakan menjadi error yang terkuantisasi (Qe) dan perubahan error terkuantisasi (ΔQe) untuk proses selanjutnya.

Level atau tingkat kuantisasi diproses menjadi penggolongan-penggolongan masukan menjadi daerah-daerah diskrit tertentu melalui interpolasi biasa dengan rentang kerja dari -6 sampai dengan 6. Pengatur fuzzy akan menentukan aksi pengendalian Δu (perubahan sinyal masukan sistem) dari keadaan sistem berupa kesalahan dan perubahan kesalahan yang merupakan masukan dari unit pengatur fuzzy, secara matematika u dapat dinyatakan:

$$\Delta u = F[e(k) - \Delta e(k - 1)] \dots\dots\dots (3.5)$$

dimana F adalah tindakan dari fungsi fuzzy berdasarkan aturan-aturan yang diberikan kepada sistem.



Gambar 3.6. Proses kuantisasi

3.3.1 Metode MAX-MIN

Komposisi aturan-aturan fuzzy untuk inferensi untuk perancangan ini yang digunakan adalah metode Max-Min. Pada metode ini solusi himpunan fuzzy diperoleh dengan cara mengambil nilai maksimum aturan, kemudian menggunakannya untuk memodifikasi daerah fuzzy dan mengaplikasikannya ke output dengan menggunakan operator OR (union). Jika semua proposisi telah dievaluasi, maka output akan berisi suatu himpunan fuzzy yang merefleksikan kontribusi dari tiap-tiap proposisi. Secara umum dapat dituliskan sebagai berikut:

$$\mu_{sf}[Xi] \leftarrow \max(\mu_{sf}[Xi], \mu_{kf}[Xi]) \dots \dots \dots (3.6)$$

Dengan:

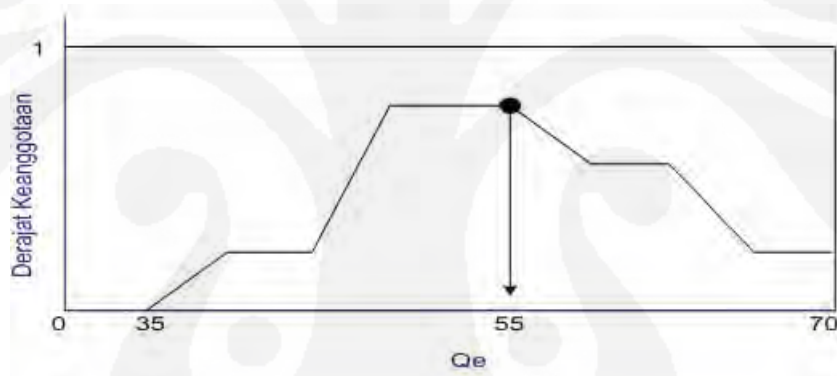
$\mu_{sf}[Xi]$ adalah nilai keanggotaan solusi fuzzy sampai aturan ke-i

$\mu_{kf}[Xi]$ adalah nilai keanggotaan konsekuen fuzzy sampai aturan ke-i

Apabila digunakan fungsi implikasi MIN, maka metode komposisi ini sering disebut dengan nama MAX-MIN atau MAMDANI. Komposisi aturannya seperti pada gambar 2.5 pada bab sebelumnya [9].

3.3.2 Metode Center of Area

Ada beberapa metode defuzzifikasi pada komposisi aturan Mamdani, salah satunya adalah metode *center of area* atau *centroid*. Pada metode ini, solusi crisp diperoleh dengan cara mengambil titik pusat daerah fuzzy.



Gambar 3.7. Metode center of area

Gambar 3.7 memperlihatkan metode center of area bekerja, ada keuntungan menggunakan metode ini, yaitu nilai defuzzy akan bergerak secara halus sehingga perubahan dari suatu topologi himpunan fuzzy ke topologi berikutnya juga akan berjalan dengan halus [9].

3.4 FUZZY SET DAN FORMASI ATURAN

Kaidah pengendalian fuzzy merupakan kumpulan aturan-aturan “*IF-THEN*” yang merupakan inti dari relasi fuzzy, pengendalian sebagai acuan untuk menyatakan aksi pengendalian. Aturan-aturan tersebut disusun berdasarkan pengamatan atau pikiran terhadap respon dinamik sistem. Perancangan kaidah

pengendali didalam unit logika fuzzy terdiri dari basis data yang merupakan media manipulasi data fuzzy. Basis aturan dari fuzzy merupakan hukum aksi pengendali fuzzy. Aturan dasar dibuat dari sehimpunan aturan linguistik yang didapatkan dengan berdasarkan pengalaman “*Human Expert*” dan menggunakan algoritma pelatihan.

Himpunan keanggotaan untuk masing-masing masukan dan FLC direncanakan tujuh keanggotaan Sedangkan batas-batas keanggotaan keluaran ditentukan berdasarkan coba-coba. Karena himpunan keanggotaan masing-masing ada tujuh keanggotaan maka aturan pengendaliannya (*if-then rule*) dibuat 49 aturan (7×7). Aturan-aturan tersebut disusun berdasarkan tujuan pengendalian yaitu memutar motor untuk mendapatkan torsi kecepatan tertentu dengan kecepatan maksimal dan permulaan pengereman yang ditentukan.

Berdasarkan data-data masukan dan keluaran. Pada pengendalian kecepatan motor yang dinyatakan dalam persamaan 3.3 dan 3.4 untuk semesta pembicaraan dan aksi pengendalian adalah torsi referensi.

Aturan-aturan jika-maka (*if-then rule*) yang digunakan adalah sebagai berikut:

1. Jika (Qe adalah NB dan perubahan Qe adalah NB) maka output adalah NB
2. Jika (Qe adalah NB dan perubahan Qe adalah NM) maka output adalah NB
3. Jika (Qe adalah NB dan perubahan Qe adalah NS) maka output adalah NB
4. Jika (Qe adalah NB dan perubahan Qe adalah ZE) maka output adalah NM
5. Jika (Qe adalah NB dan perubahan Qe adalah PS) maka output adalah NS
6. Jika (Qe adalah NB dan perubahan Qe adalah PM) maka output adalah NVS
7. Jika (Qe adalah NB dan perubahan Qe adalah PB) maka output adalah ZE
8. Jika (Qe adalah NM dan perubahan Qe adalah NB) maka output adalah NB
9. Jika (Qe adalah NM dan perubahan Qe adalah NM) maka output adalah NB
10. Jika (Qe adalah NM dan perubahan Qe adalah NS) maka output adalah NM
11. Jika (Qe adalah NM dan perubahan Qe adalah ZE) maka output adalah NS
12. Jika (Qe adalah NM dan perubahan Qe adalah PS) maka output adalah NVS
13. Jika (Qe adalah NM dan perubahan Qe adalah PM) maka output adalah ZE
14. Jika (Qe adalah NM dan perubahan Qe adalah PB) maka output adalah PVS
15. Jika (Qe adalah NS dan perubahan Qe adalah NB) maka output adalah NB

16. Jika (Qe adalah NS dan perubahan Qe adalah NM) maka output adalah NM
17. Jika (Qe adalah NS dan perubahan Qe adalah NS) maka output adalah NS
18. Jika (Qe adalah NS dan perubahan Qe adalah ZE) maka output adalah NVS
19. Jika (Qe adalah NS dan perubahan Qe adalah PS) maka output adalah ZE
20. Jika (Qe adalah NS dan perubahan Qe adalah PM) maka output adalah PVS
21. Jika (Qe adalah NS dan perubahan Qe adalah PB) maka output adalah PS
22. Jika (Qe adalah ZE dan perubahan Qe adalah NB) maka output adalah NM
23. Jika (Qe adalah ZE dan perubahan Qe adalah NM) maka output adalah NS
24. Jika (Qe adalah ZE dan perubahan Qe adalah NS) maka output adalah NVS
25. Jika (Qe adalah ZE dan perubahan Qe adalah ZE) maka output adalah ZE
26. Jika (Qe adalah ZE dan perubahan Qe adalah PS) maka output adalah PVS
27. Jika (Qe adalah ZE dan perubahan Qe adalah PM) maka output adalah PS
28. Jika (Qe adalah ZE dan perubahan Qe adalah PB) maka output adalah PM
29. Jika (Qe adalah PS dan perubahan Qe adalah NB) maka output adalah NS
30. Jika (Qe adalah PS dan perubahan Qe adalah NM) maka output adalah NVS
31. Jika (Qe adalah PS dan perubahan Qe adalah NS) maka output adalah ZE
32. Jika (Qe adalah PS dan perubahan Qe adalah ZE) maka output adalah PVS
33. Jika (Qe adalah PS dan perubahan Qe adalah PS) maka output adalah PS
34. Jika (Qe adalah PS dan perubahan Qe adalah PM) maka output adalah PM
35. Jika (Qe adalah PS dan perubahan Qe adalah PB) maka output adalah PB
36. Jika (Qe adalah PM dan perubahan Qe adalah NB) maka output adalah NVS
37. Jika (Qe adalah PM dan perubahan Qe adalah NM) maka output adalah ZE
38. Jika (Qe adalah PM dan perubahan Qe adalah NS) maka output adalah PVS
39. Jika (Qe adalah PM dan perubahan Qe adalah ZE) maka output adalah PS
40. Jika (Qe adalah PM dan perubahan Qe adalah PS) maka output adalah PM
41. Jika (Qe adalah PM dan perubahan Qe adalah PM) maka output adalah PB
42. Jika (Qe adalah PM dan perubahan Qe adalah PB) maka output adalah PB
43. Jika (Qe adalah PB dan perubahan Qe adalah NB) maka output adalah ZE
44. Jika (Qe adalah PB dan perubahan Qe adalah NM) maka output adalah PVS
45. Jika (Qe adalah PB dan perubahan Qe adalah NS) maka output adalah PS
46. Jika (Qe adalah PB dan perubahan Qe adalah ZE) maka output adalah PM
47. Jika (Qe adalah PB dan perubahan Qe adalah PS) maka output adalah PB

48. Jika (Q_e adalah PB dan perubahan Q_e adalah PM) maka output adalah PB
 49. Jika (Q_e adalah PB dan perubahan Q_e adalah PB) maka output adalah PB

Dari aturan-aturan *if-then* yang digunakan diatas maka dapat dibuat suatu tabel yang lebih ringkas seperti pada tabel 3.1 dibawah ini.

Tabel 3.1. Aturan dasar pengkoreksian berbasis fuzzy

$e/\Delta e$	NB	NM	NS	ZE	PS	PM	PB
NB	NB	NB	NB	NM	NS	NVS	ZE
NM	NB	NB	NM	NS	NVS	ZE	PVS
NS	NB	NM	NS	NVS	ZE	PVS	PS
ZE	NM	NS	NVS	ZE	PVS	PS	PM
PS	NS	NVS	ZE	PVS	PS	PM	PB
PM	NVS	ZE	PVS	PS	PM	PB	PB
PB	ZE	PVS	PS	PM	PB	PB	PB

Fuzzy set didefinisikan sebagai berikut:

Z = Zero

PS = Positive Small

PM = Positive Medium

PB = Positive Big

NS = Negative Small

NM = Negative Medium

NB = Negative Big

PVS = Positive Very Small

NVS = Negative Very Small

Seluruh membership function adalah asimetrik karena dekat dengan steady state, sinyal-sinyal tersebut membutuhkan ketelitian yang lebih. Tujuh membership function dipilih untuk sinyal $e(pu)$ dan delta $e(pu)$ dan sembilan untuk keluarannya. Semua membership function adalah simetrik untuk nilai positif dan negatif dari variabel. Jadi maximum ada $7 \times 7 = 49$ aturan yang digunakan.

3.5 PEMODELAN SISTEM

Pemodelan sistem menggunakan simulink MATLAB 7.1 dengan menggunakan teknik sampling. Masukan pada sistem adalah kecepatan dan torsi referensi pada motor induksi. Adapun spesifikasi motor [8] yang dipakai adalah sbb: Motor induksi 50 HP 460 V, 4 kutub, 60 Hz, dan 1780 RPM

Tabel 3.2. Spesifikasi motor induksi

Rated values	Power	50	HP
	Frequency	60	Hz
	Voltage	460	V
	Motor Speed	1760	rpm
	pole pair (p)	2	
Rated parameters	Rs	0.087	Ω
	Rr	0.228	Ω
	Ls	8.00E-04	H
	Lr	8.00E-04	H
	M.	3.47E-02	H
Constant	j	1.662	kg.m ²

Motor induksi disuplai oleh arus yang di kendalikan oleh PWM inverter yang menggunakan universal bridge block yaitu IGBT inverter. Motor induksi menggerakkan beban mekanis yang memiliki inersia J, koefisien friksi B dan torsi beban T_L . Pengendali kecepatan menggunakan Fuzzy Logic termasuk pengendali PI yang menghasilkan referensi arus quadrature-axis i_q^* yang mengendalikan torsi motor. Flux motor dikendalikan oleh arus referensi direct-axis i_d^* . Blok DQ-ABC digunakan untuk mengubah arus i_d^* dan i_q^* menjadi arus referensi i_a^* , i_b^* , dan i_c^* sebagai pengendali arus.

Kecepatan medan putar stator:

$$N_s = \frac{120f}{p} = \frac{120 \times 60}{4} = 1800 \text{ rpm}$$

Slip:

$$S = \frac{N_s - N_r}{N_s} = \frac{1800 - 1760}{1800} = 0.02 = 2\%$$

Frekuensi rotor:

$$f' = S \cdot f = 0.02 \times 60 = 1.2 \text{ Hz}$$

Beban dan Pengaturan kecepatan motor dapat dicari dengan formula berikut dimana Torsi adalah T dan kecepatan ω .

$$T = k \times \omega^2$$

Nominal Torsi dari motor adalah

$$T_n = \frac{50 \times 746}{157} = 237,5 \text{ N.m}$$

Maka dapat dicari konstantanya adalah

$$k = \frac{T_n}{\omega_s^2} = \frac{237,5}{157^2} = 9,63 \cdot 10^{-3}$$

Formula yang digunakan untuk membangkitkan pulsa PWM yang digunakan untuk membandingkan tegangan keluaran dengan frekuensi gelombang tiga fasa adalah:

$$V_{LLrms} = \frac{m}{2} \times \frac{\sqrt{3}}{\sqrt{2}} V_{dc} = m \times 0,612 \times V_{DC}$$

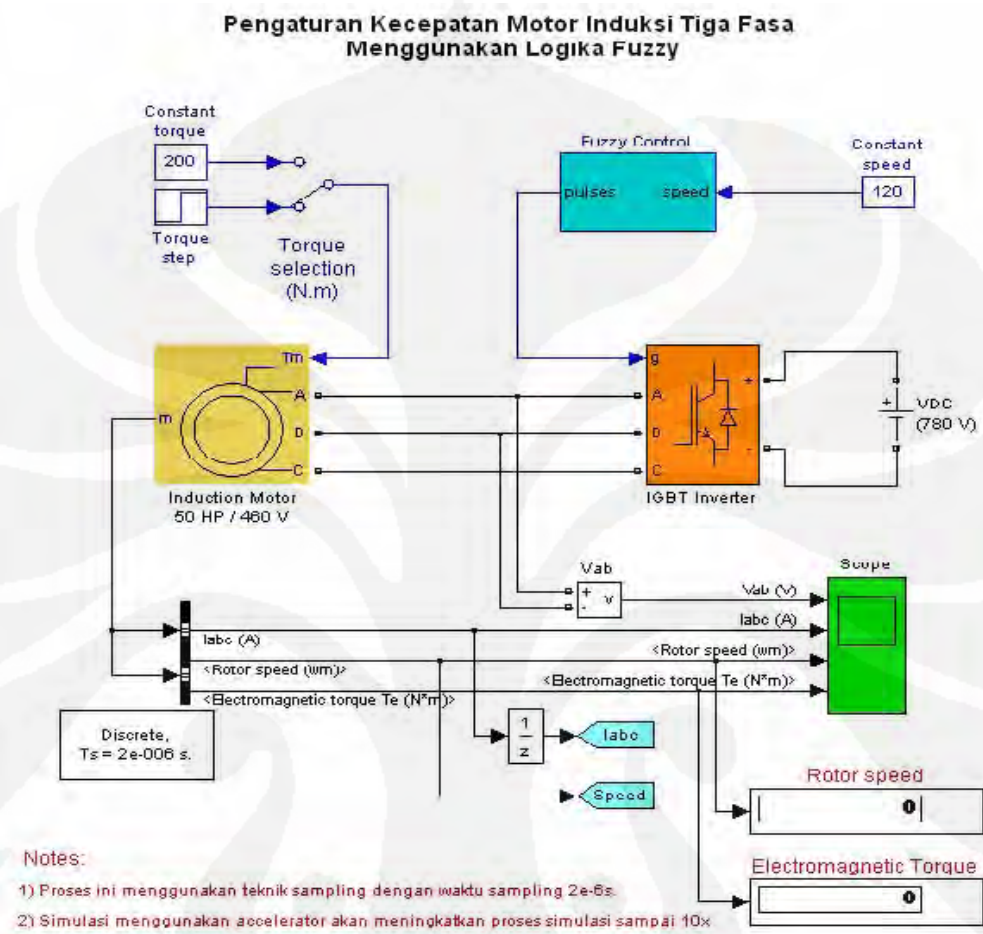
Dimana :

1. Tegangan keluaran line-line V_{LLrms} adalah tegangan nominal dari motor induksi.
2. m adalah faktor modulasi yaitu 0,97 yang dihasilkan oleh tegangan keluaran line-line.

Maka dapat dicari tegangan V_{DC} :

$$V_{DC} = \frac{V_{LLrms}}{m \times 0,612} = \frac{460}{0,97 \times 0,612} = 774,8 \text{ Volt}$$

Gambar dibawah adalah diagram sirkuit pengendali motor induksi menggunakan logika fuzzy pada Matlab.

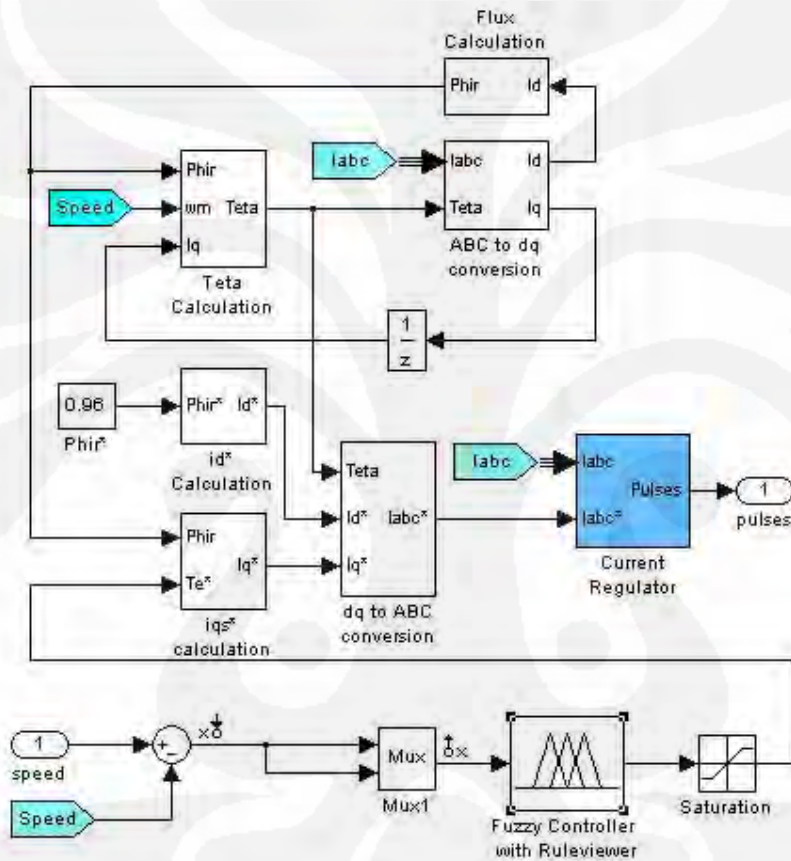


Gambar 3.8. Pemodelan sistem dengan MATLAB Simulink

Pada gambar 3.8 *Speed Reference* yang digunakan pada model ini adalah konstan pada 120 rad/s yang merupakan kecepatan masukan. Sedangkan *Torque Reference* yang memberikan referensi nilai beban torsi kepada pengendali logika fuzzy, Sedangkan model rangkaian PWM inverter dipilih adalah IGBT dan ABC sebagai terminal keluaran. DC link sebagai sumber tegangan ke IGBT memberikan tegangan 780 Vdc. Blok pengukur arus dan tegangan digunakan untuk menampilkan visualisasi sinyal yang akan diukur. Arus motor, kecepatan dan sinyal torsi merupakan keluaran dari motor induksi. Sistem kendali menggunakan teknik sampling selama 2 mikrodetik. Konversi antara rpm menjadi rad/sec menggunakan formula sbb:

$$1 \text{ rpm} = \frac{2\pi}{60} \text{ rad/sec} \dots\dots\dots(3.7)$$

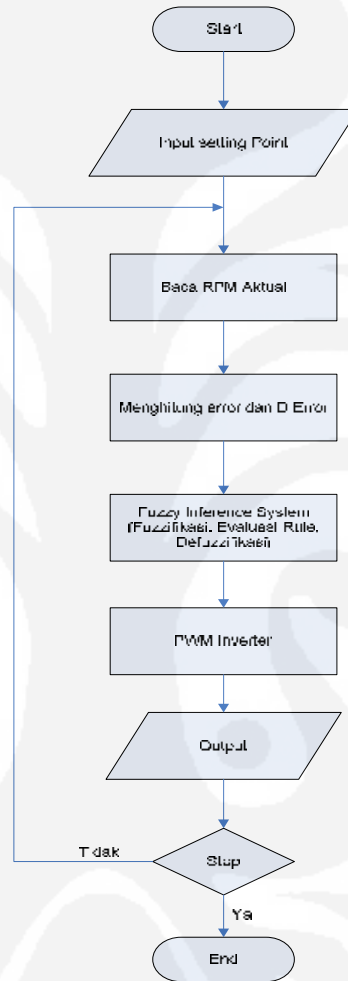
Sedangkan pada blok model Pengendali Logika Fuzzy pada gambar 3.8 didalamnya terdapat rangkaian pengendali logika fuzzy yang lebih detail sekaligus merupakan tempat nilai masukan dan umpan balik diterima seperti pada gambar 3.9.



Gambar 3.9. Model Pengendali Logika Fuzzy

Diagram alir sederhana pada gambar 3.10 dibawah ini menjelaskan proses kendali pengaturan kecepatan motor induksi dengan logika fuzzy. Dengan memasukan nilai *setpoint* berupa referensi kecepatan maka sistem kendali ini akan membaca setiap kecepatan aktual keluaran sebagai umpan baliknya. Untuk masukan dan umpan balik error berupa tegangan AC maka perubahan error-nya juga berupa tegangan AC. Kemudian masuk kedalam pengendali logika fuzzy

yang diubah ke tegangan DC. Hal ini dilakukan karena perubahan secara cepat dapat dilakukan dengan mudah bila diubah ke dalam tegangan DC. Keluaran FLC (*Fuzzy Logic Controller*) dimasukkan kedalam PWM Inverter untuk mengubah kembali ke dalam tegangan AC.



Gambar 3.10. Diagram alir Pengaturan kecepatan dengan logika fuzzy

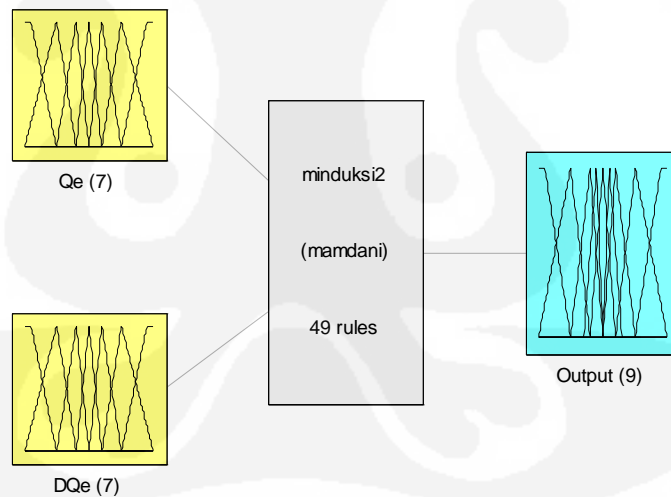
BAB IV

PENYAJIAN DATA DAN PEMBAHASAN

4.1 DASAR ATURAN (RULE BASE)

Penalaran dengan metode Mamdani hampir sama dengan metode penalaran Sugeno, hanya saja output (konsekuen) sistem tidak berupa himpunan fuzzy, melainkan berupa konstanta atau persamaan linear. Metode ini diperkenalkan pada tahun 1985. Penggunaan logika fuzzy pada sistem ini adalah untuk mengatur kecepatan motor induksi tiga fasa dimana akan dilihat respon dinamik dari motor dengan memasukan *setpoint* referensi kecepatan (rpm).

Pada sistem ini, *error* proses (Q_e) dan perubahan *error* (ΔQ_e) digunakan sebagai input pada basis pengetahuan, sedangkan aksi kontrol dijadikan sebagai output yang akan dicari [2].

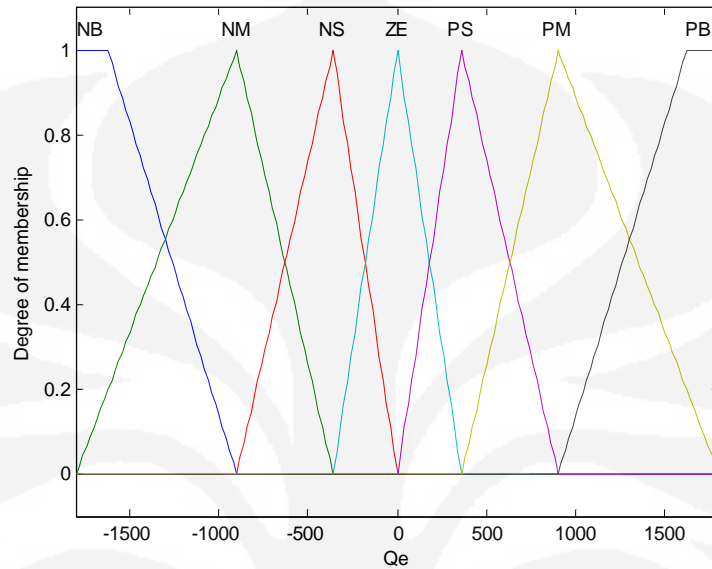


System minduksi2: 2 inputs, 1 outputs, 49 rules

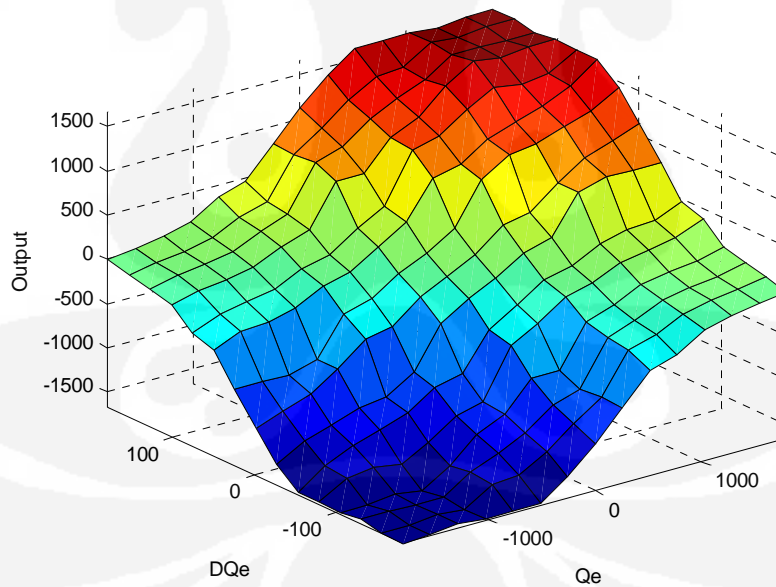
Gambar 4.1. FIS Editor untuk metode Mamdani

Gambar 4.1 menjelaskan bahwa pada pengendali logika fuzzy ini memiliki dua masukan yaitu error dan perubahan error. Kemudian dimasukkan kedalam rule

base yang telah dibuat dimana masing masukan memiliki tujuh himpunan fuzzy (NB, NM, NS, ZE, PS, PM dan PB). Hal ini dapat dilihat pada gambar 4.2.



Gambar 4.2. Fungsi keanggotaan input Q_e dan ΔQ_e



Gambar 4.3. Surface Viewer

Gambar 4.3 adalah *surface viewer* yang menunjukkan kaitan antara input dan output. Dimana keduanya sudah ternormalisasi.

4.2 HASIL SIMULASI DAN PENGUJIAN RESPON

Perkembangan teori kontrol juga diikuti oleh software pendukungnya. mulai dari software untuk pemrograman sistem sampai dengan software untuk proses simulasinya. Salah satu software yang dapat dipergunakan untuk simulasi tersebut adalah Matlab.

Software ini dilengkapi dengan berbagai toolbox yang memudahkan pemakai untuk melakukan perhitungan-perhitungan tertentu. Bahkan saat ini sudah dikembangkan toolbox khusus untuk simulasi yang diberi nama Simulink.

Aplikasi MatLab dalam bidang pengaturan dilengkapi Kontrol Toolbox. Toolbox ini sudah dilengkapi dengan berbagai macam fungsi pendukung yang dipergunakan dalam analisa sistem kontrol. Beberapa fungsi pendukung yang sering dipergunakan untuk menganalisa suatu sistem adalah: *feedback*, *step*, *rlocus*, *series*, dll. Untuk menganalisa suatu sistem, software hanya memerlukan masukan berupa *transfer function* yang ditulis dalam Laplace Transform (dalam s domain) atau matriks. Untuk selanjutnya, pemakai tinggal memilih analisa yang akan dipergunakan. Tulisan ini akan membahas penggunaannya secara khusus untuk merancang *Fuzzy Logic Controller* pada suatu sistem.

Sebagai contoh, suatu sistem kontrol memiliki *transfer function* dengan kriteria perancangan sebagai berikut :

1. Memiliki *rise time* yang cepat
2. *Overshoot* sekecil mungkin
3. Tidak memiliki *steady state error*.

Dari fungsi di atas, maka parameter-parameter yang dimasukkan berupa koefisien pembilang dan penyebutnya. Biasanya dipergunakan variabel *num* untuk pembilang dan *den* untuk penyebut. Kedua nama variabel tersebut tidak mutlak, jadi penggunaan nama variabel yang lain juga diperbolehkan. Setelah itu komputer sudah siap untuk menganalisa sistem kontrol. Langkah kedua yang perlu dilakukan adalah memilih jenis input yang akan dimasukkan ke dalam sistem. Input ini bisa berupa *step*, *pulse*, *ramp*, *sinus*, dan sebagainya.

Dalam pengaturan motor induksi industri kontroler logika fuzzy sering digunakan karena dapat mengurangi adanya *overshoot* yang berlebihan, *dead time* dan *rise time* yang terlalu lama. Penalaan parameter proses kerja motor induksi dilakukan dengan cara mencoba-coba, sehingga dihasilkan nilai parameter yang optimal. Sedangkan penalaran parameter unit pengendali fuzzy, terutama gain error dan gain perubahan error, dilakukan secara *heuristik*, yaitu dengan cara mempelajari perkiraan respon yang akan dihasilkan oleh sistem motor induksi

yang meliputi *maximum overshoot*, *settling time*, *rise time* dan *steady state error*. Respon dinamik ditentukan oleh beberapa parameter yaitu waktu naik (*rise time*), waktu tetap (*settling time*), max overshoot dan error keadaan tunak (*steady state*). Semakin cepat waktu naik dan waktu tetap dan kecilnya *overshoot* serta *error steady state* adalah keadaan ideal yang diinginkan dalam sistem pengaturan.

Pengujian sistem dilakukan dengan melihat respon sistem terhadap fungsi waktu selama selang waktu tertentu akan diamati respon sistem terhadap pemberian beban dan pelepasan beban dengan berbagai kondisi meliputi

1. Variasi bentuk *membership function*
2. Variasi jumlah label *membership function*
3. Variasi metode defuzzifikasi

Simulasi dilakukan dengan menggunakan simulink dan dilakukan dengan pada kondisi-kondisi sebagai berikut:

1. Simulasi dan Analisa Uji Respon Input variasi beban Torsi.
2. Simulasi dan Analisa Uji Respon Input variasi Torsi Step.
3. Analisa perbandingan antara Pengendali PI yang sudah ada dengan Pengendali Logika Fuzzy.

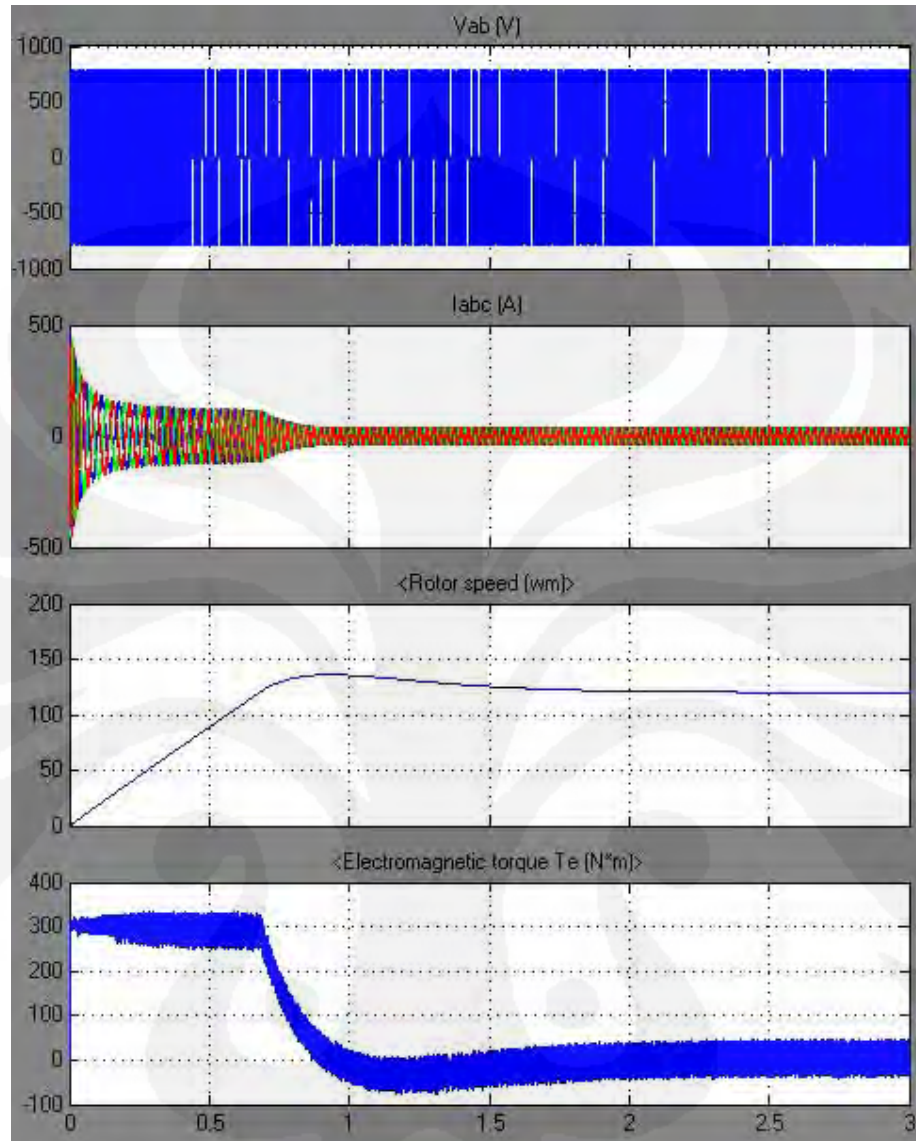
Dari simulasi-simulasi ini diharapkan dapat dilihat pengaruh perubahan torsi beban berbasis fuzzy.

Motor induksi yang digunakan adalah motor induksi tiga fasa, 4 kutub, 50 HP, 460 V, 60 Hz, 1760 rpm.

Berikut adalah grafik-grafik respon sistem terhadap waktu, hasil dari pengujian yang telah dilakukan dengan kecepatan referensi 120 rad/s dengan waktu simulasi 3 detik.

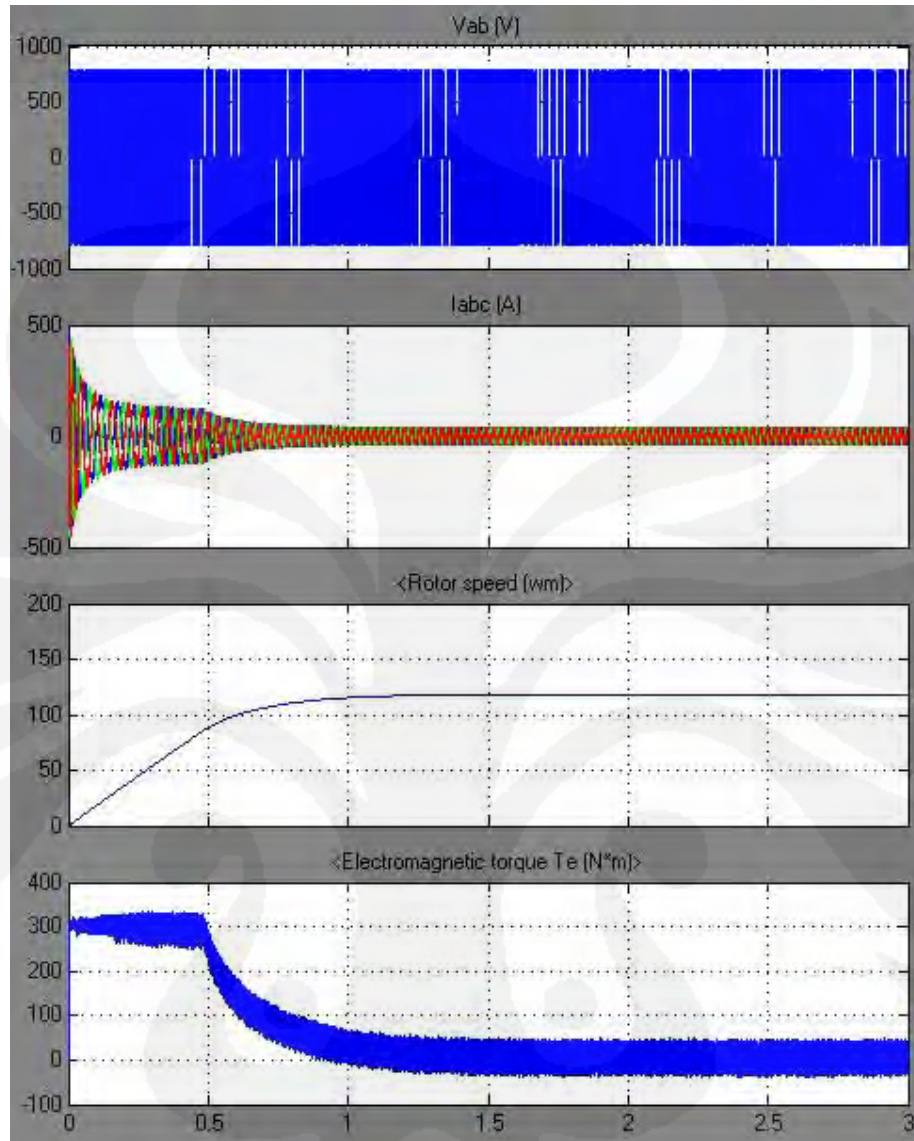
4.2.1 Simulasi dan Analisa Uji Respon Input dengan Torsi Beban 0 N.m

Dengan waktu simulasi adalah 3 detik maka akan dihasilkan grafik respon sebagai berikut:



Gambar 4.4. Grafik respon PI untuk beban 0 N.m

Pada gambar 4.4 memperlihatkan pengendali motor menggunakan PI. Motor mulai berputar dengan torsi beban nol dan kecepatan masukan 120 rad/s. Terjadi overshoot sekitar 0.4%, waktu naik T_r 0.7 detik, waktu tetap T_s 2.5 detik dan steadystate error 0.005. Pada arus dan torsi terjadi kondisi maksimum saat start dan menurun setelah 0.7 detik.

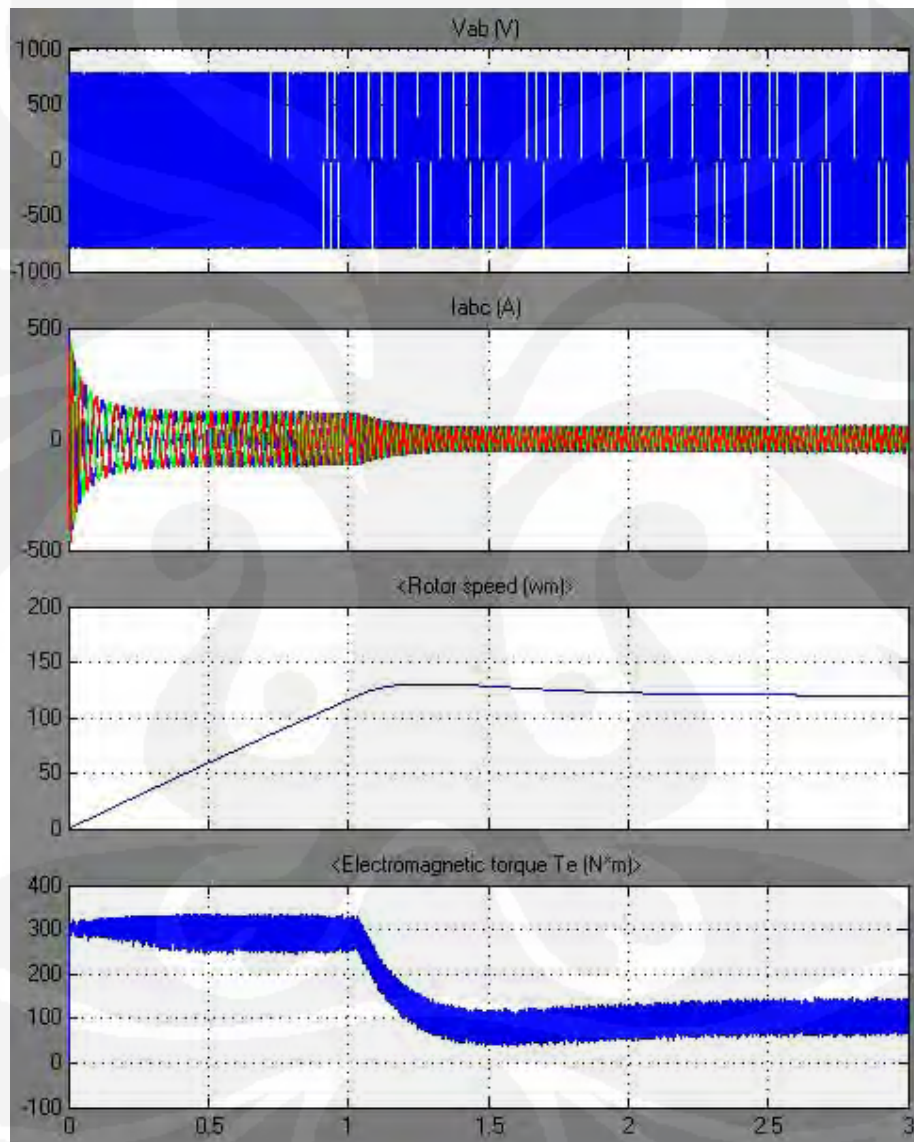


Gambar 4.5. Grafik respon FLC untuk beban 0 N.m

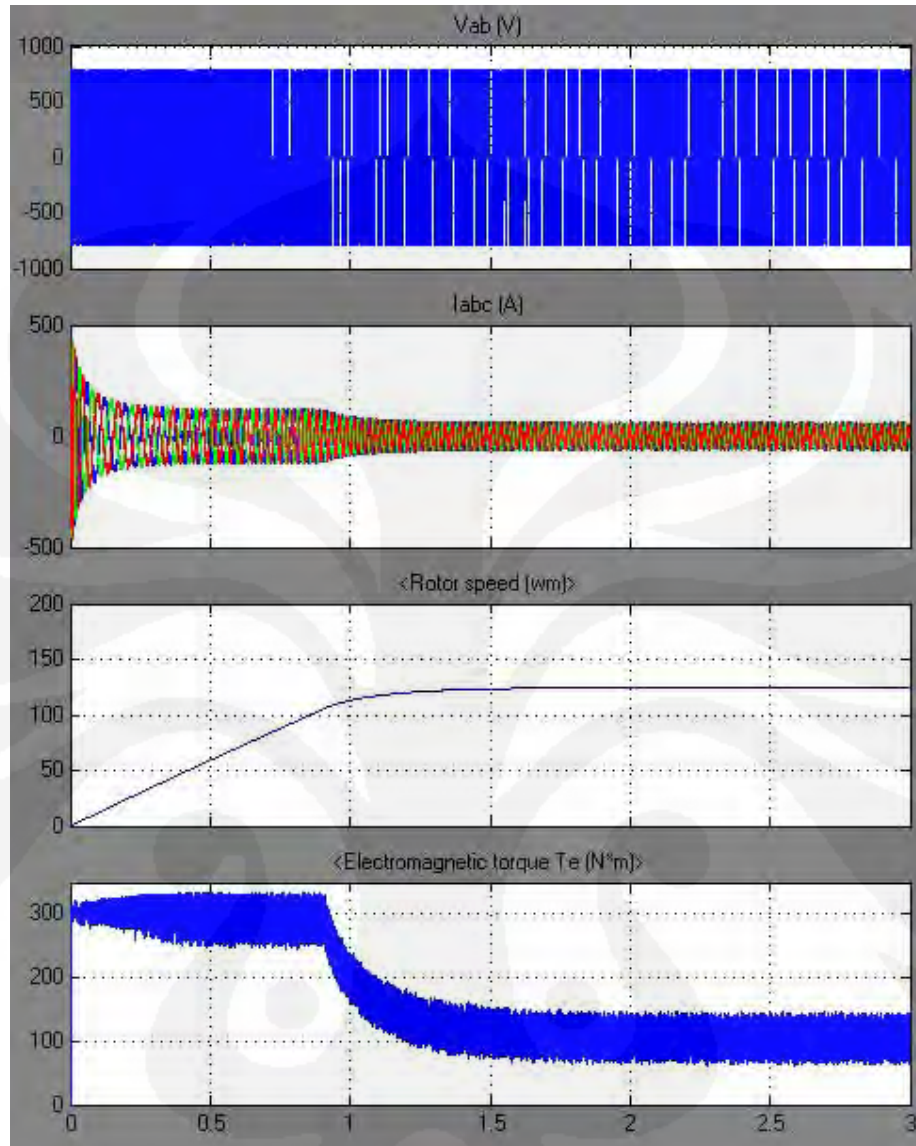
Pada gambar 4.5 memperlihatkan pengendali motor menggunakan FLC. Motor mulai berputar dengan torsi beban nol dan kecepatan masukan 120 rad/s. Terjadi tanpa overshoot 0.0%, waktu naik T_r 0.5 detik, waktu tetap T_s 1.5 detik dan steadystate error 0.003%. Pada arus dan torsi terjadi kondisi maksimum saat start dan menurun setelah 0.5 detik.

4.2.2 Simulasi dan Analisa Uji Respon Input dengan Torsi Beban 100 N.m.

Pada gambar 4.6 memperlihatkan pengendali motor menggunakan PI. Motor mulai berputar dengan torsi beban 100 N.m dan kecepatan masukan 120 rad/s. Terjadi overshoot sekitar 0.3%, waktu naik T_r 1.1 detik, waktu tetap T_s 2.7 detik dan steadystate error 0.005. Pada arus dan torsi terjadi kondisi maksimum saat start dan menurun setelah 1.1 detik.



Gambar 4.6. Grafik respon PI untuk beban 100 N.m

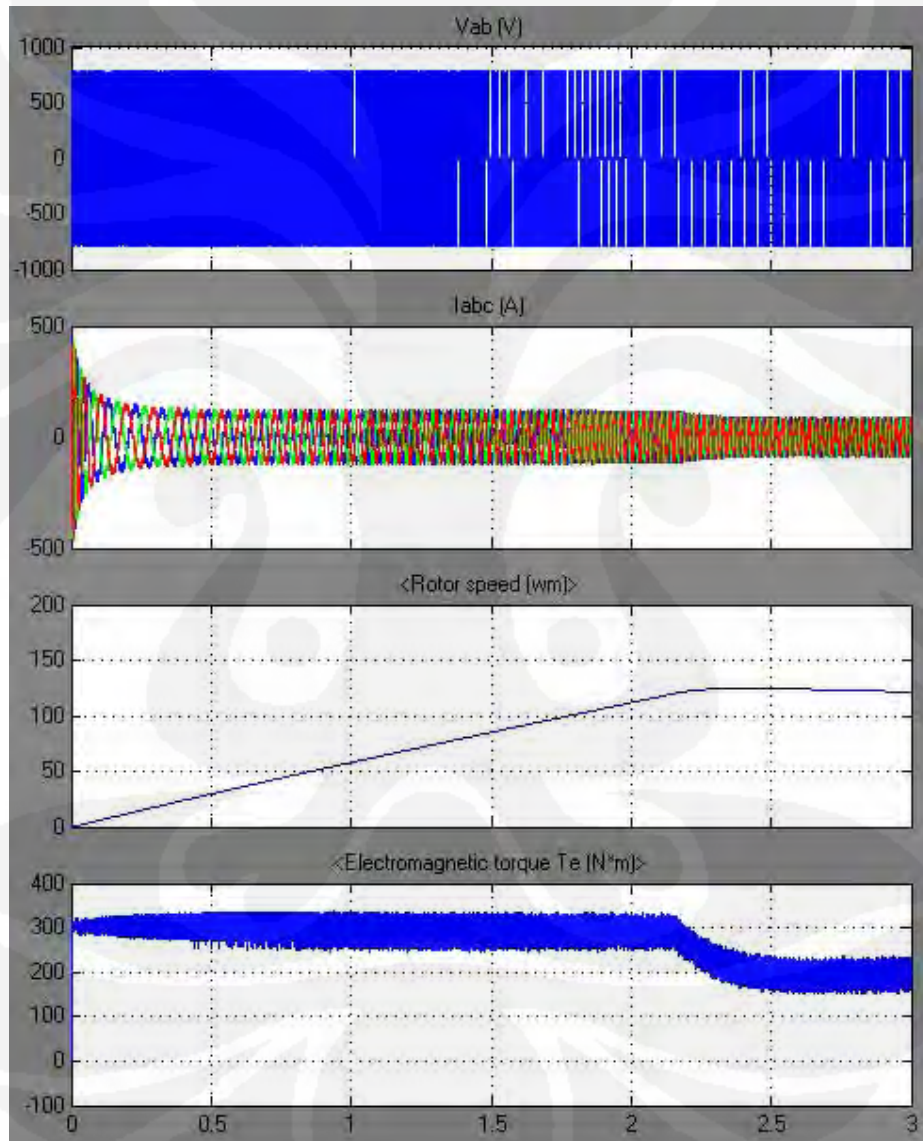


Gambar 4.7. Grafik respon FLC untuk beban 100 N.m

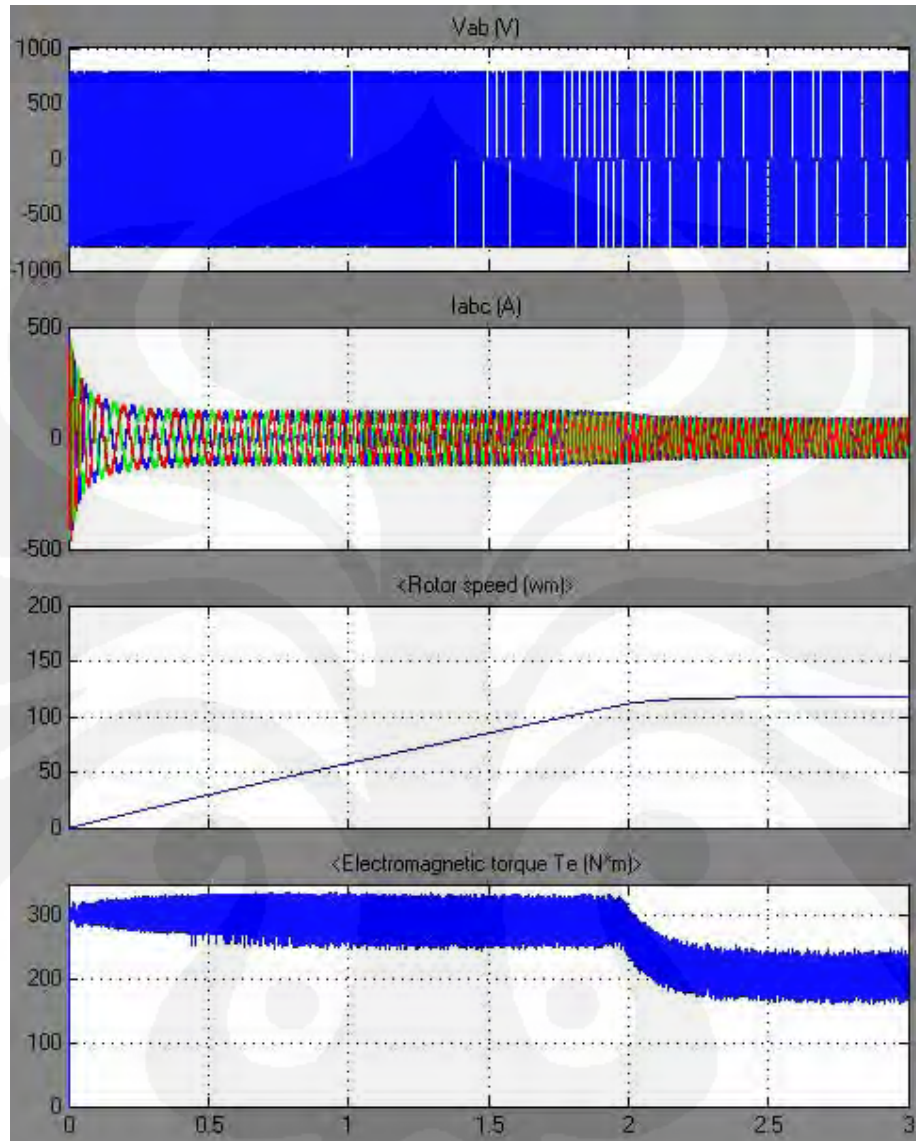
Pada gambar 4.7 memperlihatkan pengendali motor menggunakan FLC. Motor mulai berputar dengan torsi beban 100 N.m dan kecepatan masukan 120 rad/s. Terjadi tanpa overshoot 0.0%, waktu naik T_r 0.8 detik, waktu tetap T_s 1.7 detik dan steadystate error 0.003%. Pada arus dan torsi terjadi kondisi maksimum saat start dan menurun setelah 0.8 detik.

4.2.3 Simulasi dan Analisa Uji Respon Input dengan Torsi Beban 200 N.m.

Pada gambar 4.8 memperlihatkan pengendali motor menggunakan PI. Motor mulai berputar dengan torsi beban 200 N.m dan kecepatan masukan 120 rad/s. Terjadi overshoot sekitar 0.2%, waktu naik T_r 2.2 detik, waktu tetap T_s 3.2 detik dan steadystate error 0.005. Pada arus dan torsi terjadi kondisi maksimum saat start dan menurun setelah 2.2 detik.



Gambar 4.8. Grafik respon PI untuk beban 200 N.m



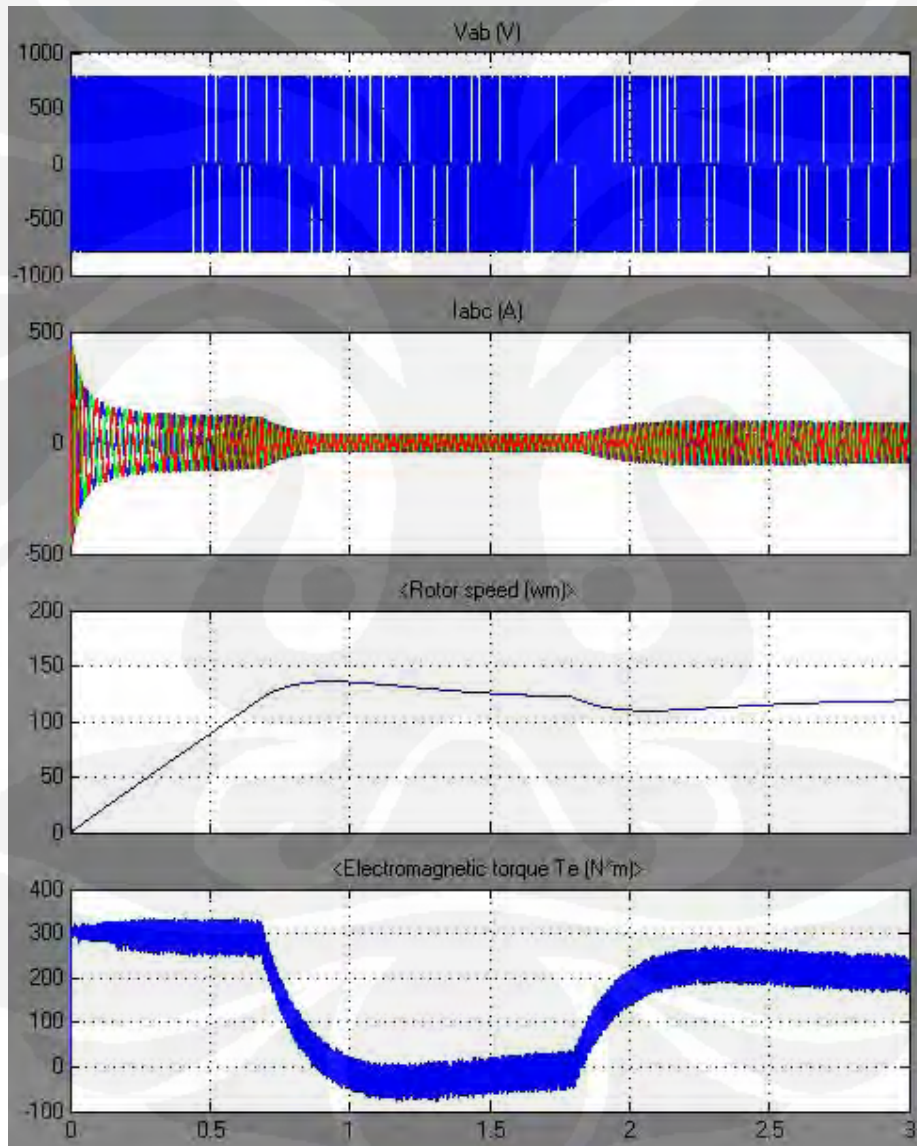
Gambar 4.9. Grafik respon FLC untuk beban 200 N.m

Pada gambar 4.9 memperlihatkan pengendali motor menggunakan FLC. Motor mulai berputar dengan torsi beban 200 N.m dan kecepatan masukan 120 rad/s. Terjadi tanpa overshoot 0.0%, waktu naik T_r 1.9 detik, waktu tetap T_s 2.7 detik dan steadystate error 0.003%. Pada arus dan torsi terjadi kondisi maksimum saat start dan menurun setelah 1.9 detik.

4.2.4 Simulasi dan Analisa Uji Respon Input dengan Torsi Step

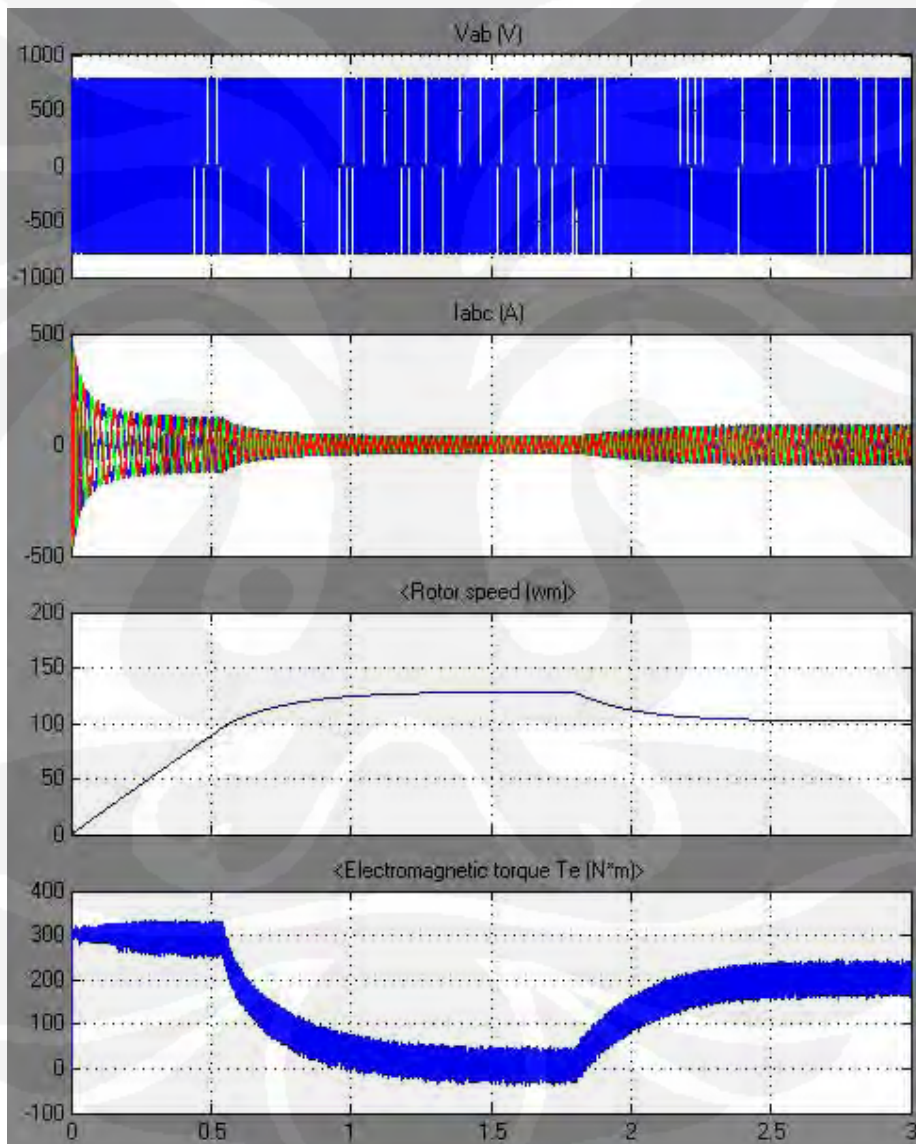
Input step memiliki parameter sebagai berikut:

1. Step time 1.8
2. Initial value 0
3. Final value 200
4. Sample time 0



Gambar 4.10. Grafik respon PI untuk beban torsi step

Pada gambar 4.10 memperlihatkan pengendali motor menggunakan PI. Terjadi perubahan kecepatan, torsi, arus dan tegangan terjadi pada step time 1,8 detik dengan perubahan beban dari nol menjadi 200 N.m. Penurunan kecepatan rotor terjadi pada saat $t = 1,8$ detik hal ini disebabkan oleh perubahan beban secara mendadak setelah itu kembali pada kecepatan referensi. Terjadi undershoot sekitar 0.2%, waktu naik T_r 2.1 detik, waktu tetap T_s 3.5 detik dan steadystate error 0.005.



Gambar 4.11. Grafik respon FLC untuk beban torsi step

Pada gambar 4.11 memperlihatkan pengendali motor menggunakan FLC. Terjadi perubahan kecepatan, torsi, arus dan tegangan terjadi pada step time 1,8 detik dengan perubahan beban dari nol menjadi 200 N.m. Penurunan kecepatan rotor terjadi pada saat $t = 1,8$ detik hal ini disebabkan oleh perubahan beban secara mendadak setelah itu kembali pada kecepatan referensi. Terjadi tanpa undershoot sekitar 0.0%, waktu naik T_r 2.3 detik, waktu tetap T_s 3.2 detik dan steadystate error 0.003.

4.3 PERBANDINGAN PENGENDALI PI DAN FUZZY

Dari hasil percobaan diatas didapat data-data keluaran sebagai berikut:

Tabel 4.1. Hasil Percobaan Pengendali PI

Beban	Torsi (N.m)	OS/US (%)	T_r (dt)	T_s (dt)	Ess (%)
Nol	-5.545	0.4	0.7	2.5	0.005
25	4.871	0.4	0.8	2.6	0.005
50	64.43	0.4	0.9	2.6	0.005
75	86.37	0.3	1.0	2.7	0.005
100	112.7	0.3	1.1	2.7	0.005
125	137.1	0.3	1.4	2.8	0.005
150	141.1	0.2	1.8	2.9	0.005
175	197.6	0.2	2.0	3.0	0.005
200	208.1	0.2	2.2	3.2	0.005
Step	226.5	0.2	2.1	3.5	0.005

Pada tabel 4.1 merupakan data yang didapat pada saat 3 detik yang diambil dari display kecepatan dan torsi. Nilai torsi yang ditampilkan fluktuasi dan kecepatan referensi selalu tetap yaitu 120 rad/s. Tabel tersebut menjelaskan bahwa dengan perubahan beban dari 0, 25, 50, 75, 100, 125, 150, 175 dan 200 menyebabkan torsi meningkat, overshoot menurun, waktu naik T_r meningkat, waktu tetap T_s meningkat dan steadystate error sebesar 0.005%. Untuk masukan beban step menyebabkan terjadinya undershoot dan torsi menjadi naik karena perubahan beban secara mendadak dari nol menjadi 200 N.m

Tabel 4.2. Hasil Percobaan Pengendali FL

Beban	Torsi (N.m)	OS/US (%)	Tr (dt)	Ts (dt)	Ess (%)
Nol	8.084	0.0	0.5	1.5	0.003
25	28.65	0.0	0.6	1.5	0.003
50	62.72	0.0	0.6	1.6	0.003
75	92.08	0.0	0.7	1.6	0.003
100	117.3	0.0	0.8	1.7	0.003
125	125.9	0.0	1.0	1.9	0.003
150	154	0.0	1.3	2.2	0.003
175	183.4	0.0	1.6	2.4	0.003
200	213	0.0	1.9	2.7	0.003
Step	197.8	0.0	2.3	3.2	0.003

Pada tabel 4.2 merupakan data yang didapat pada saat 3 detik yang diambil dari display kecepatan dan torsi. Nilai torsi yang ditampilkan fluktuasi dan kecepatan referensi selalu tetap yaitu 120 rad/s. Tabel tersebut menjelaskan bahwa dengan perubahan beban dari 0, 25, 50, 75, 100, 125, 150, 175 dan 200 menyebabkan torsi meningkat, waktu naik T_r meningkat, waktu tetap T_s meningkat dan steady state error sebesar 0.003%. Pada Pengendali FLC tidak ada overshoot. Untuk masukan beban step menyebabkan torsi menjadi naik karena perubahan beban secara mendadak dari nol menjadi 200 N.m tapi tanpa adanya undershoot.

Perbandingan antara pengendali PI dan Fuzzy saat motor mulai berputar menunjukkan pengendali fuzzy mempunyai kemampuan yang lebih baik dari pada pengendali PI dan memiliki waktu respon yang lebih cepat.

Perbandingan antara pengendali PI dan Fuzzy saat perubahan beban step menunjukkan kecepatan motor mengikuti referensinya dengan tidak ada steady state error dan respon yang cepat menggunakan pengendali logika fuzzy. Hal itu memperlihatkan bahwa respon kecepatan dipengaruhi oleh kondisi dari beban. Pengendali fuzzy juga memberikan respon yang lebih baik untuk overshoot, steady state error dan respon yang cepat. Pengendali logika fuzzy juga dapat menangani perubahan kecepatan secara tiba-tiba tanpa adanya overshoot, undershoot dan steady state error.

BAB V

KESIMPULAN

Setelah melakukan simulasi pengendalian kecepatan pada motor induksi menggunakan logika fuzzy maka dapat disimpulkan beberapa hal yaitu:

1. Motor induksi yang digunakan adalah motor induksi tiga fasa, 4 kutub, 50 HP, 460 V, 60 Hz, 1760 rpm.
2. Simulasi pengendali fuzzy dengan beban bervariasi dan masukan kecepatan 120 rad/sec yang selalu tetap yang menghasilkan keluaran dengan waktu simulasi selama 3 detik. Keluaran yang dihasilkan adalah tanpa overshoot, rise time 0.5 detik, settling time 1.5 detik dan steady state error 0,003%.
3. Simulasi pengendali fuzzy dengan beban torsi step dan masukan kecepatan 120 rad/sec menghasilkan tanpa overshoot. Kondisi perubahan step terjadi pada $t=1.8$ detik dan dengan waktu simulasi selama 3 detik.
4. Perbandingan antara pengendali PI dan Fuzzy saat motor mulai berputar menunjukkan pengendali fuzzy mempunyai kemampuan yang lebih baik dari pada pengendali PI dan memiliki waktu respon yang lebih cepat.
5. Sistem pengendali logika fuzzy dapat menangani kecepatan tanpa overshoot, under-shoot dan steady-state error dengan perubahan beban yang mendadak.
6. Fungsi keanggotaan dalam pengaturan kecepatan motor induksi sangat dibutuhkan dimana fungsi keanggotaan tersebut sangat menentukan keadaan kestabilan dari kontrol motor yang digunakan.

DAFTAR ACUAN

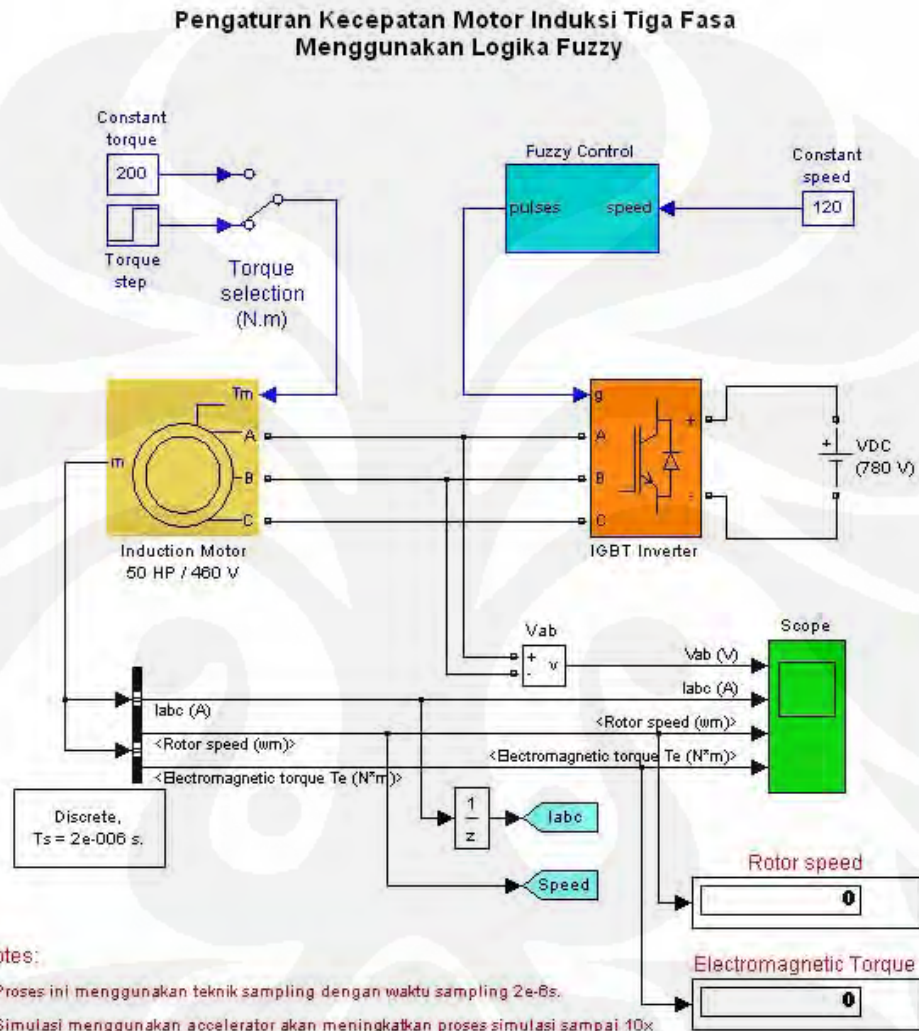
- [1]. Parekh, R., *AC Induction Motors Fundamentals*, Microchip Technology Inc. AN887. 2003.
- [2]. Lee, C.-C., *Fuzzy logic in control systems: fuzzy logic controller-parts 1 and 2*, IEEE Transactions on Systems, Man, and Cybernetics, Vol. 20, No. 2, pp 404-435, 1990.
- [3]. Mamdani, E.H., *Applications of fuzzy logic to approximate reasoning using linguistic synthesis*, IEEE Transactions on Computers, Vol. 26, No. 12, pp. 1182-1191, 1977.
- [4]. Bose, B. K., *Modern Power Electronics and AC Drives*, Prentice-Hall, N.J., 2002.
- [5]. Krause, P.C., O. Wasynczuk, and S.D. Sudhoff, *Analysis of Electric Machinery*, IEEE Press, 2002.
- [6]. Harunur, M. R., *Power Electronics*, Prentice-Hall, 1988.
- [7]. Lowery F. Thomas, *Application Consideration for PWM inverter-fed low-voltage induction motors*, IEEE transaction on industri application, April 1994
- [8]. Simulink, *Fuzzy Logic Toolbox*, MATLAB ver. 7.1, The Mathwork.
- [9]. Sri Kusumadewi, *Analisis dan Desain Sistem Fuzzy*, Graha Ilmu Yogyakarta, 2002

DAFTAR PUSTAKA

- Micio sugeno and takahiro Yasukawa, *A Fuzzy-logic-base Approach to qualitative Modelling*, IEEE transaction on fuzzy system, Vol. 1, No. 1, February 1993
- Li-Xing Wang, *Adaptive Fuzzy Control of Non Linear System*, IEE Transaction on Fuzzy System Vol. 1, No. 2, May 1993
- C.J. Harris, C.G. Morre & M. Brown, *Intellegent Control, advanced System*, Research Group, University of Southhampton, UK. 1994
- Timothy J. Ross, *Fuzzy Logic with Engineering Applications*, McGraw-Hill International Editions. Singapore 1997
- Ricardo Rovanti and Roberto Guerrierri, *Fuzzy Set of Rule for System Identification*, IEE Transaction On Fuzzy System Vol. 1. No. 2, May 1996
- David Brown, E.P. Hamilton III. *Electro mechanical energy conversion*, General Electric Company and University of Texas, Austin 1984
- Zadeh, L.A., *Fuzzy Logic Computer*, Vol. 1, No. 4, pp. 83-93, 1988.
- Khater, F.M.H., Novotny, D.W., *An Equivalent Circuit Model for Phase Back Voltage Control of AC Machines*, III Transaction on Industry Application Vol. IA-22No. 5, September/October 1986.
- Leonhard, W., *Control of Electrical Drives*, Springer-Verlag, Berlin, 1996
- Jang, J.-S. R., *ANFIS: Adaptive-Network-based Fuzzy Inference Systems*, IEEE Transactions on Systems, Man, and Cybernetics, Vol. 23, No. 3, pp. 665-685, May 1993.

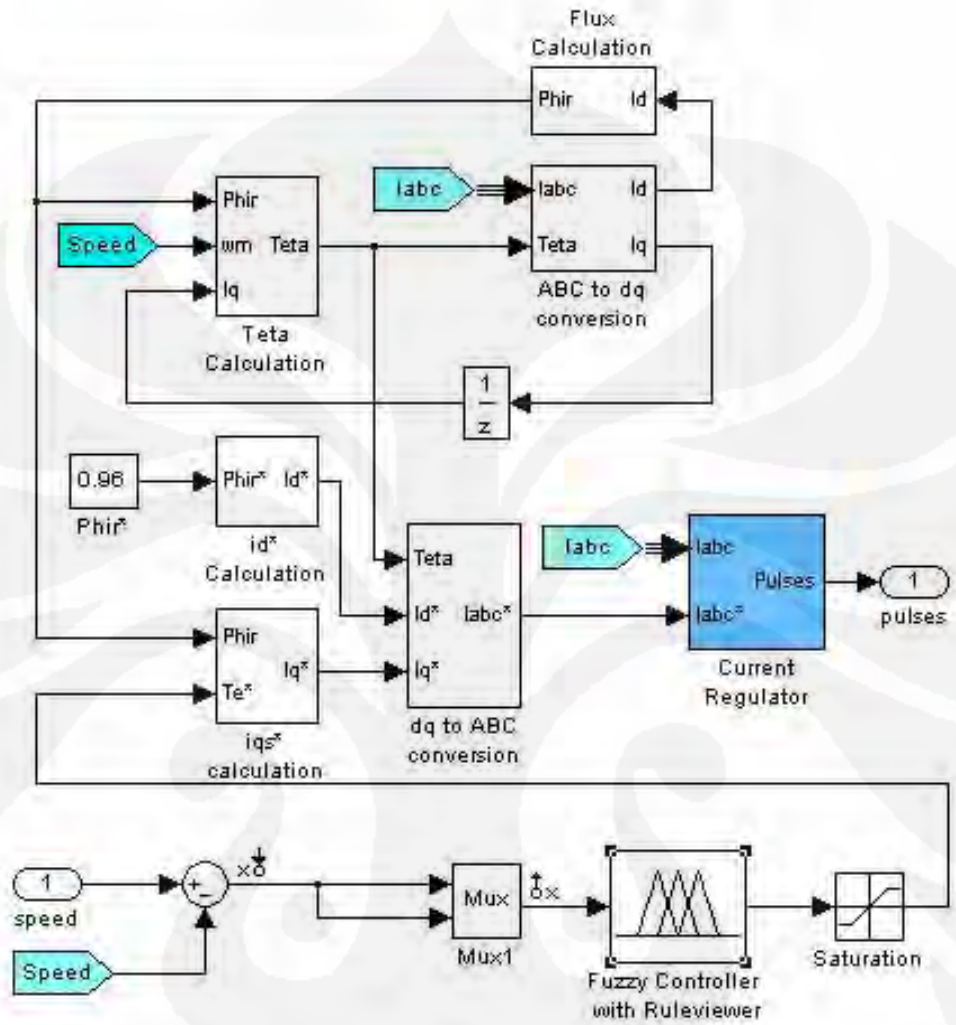
LAMPIRAN 1

Pengaturan Motor Induksi Tiga Fasa Menggunakan Logika Fuzzy



LAMPIRAN 2

FUZZY LOGIC CONTROLLER



LAMPIRAN 3

FIS EDITOR

```
[System]
Name='minduksi'
Type='mamdani'
Version=2.0
NumInputs=2
NumOutputs=1
NumRules=49
AndMethod='min'
OrMethod='max'
ImpMethod='min'
AggMethod='max'
DefuzzMethod='centroid'
```

```
[Input1]
Name='Qe'
Range=[-1800 1800]
NumMFs=7
MF1='NB': 'trapmf', [-1800 -1800 -1620 -900]
MF2='NM': 'trimf', [-1800 -900 -360]
MF3='NS': 'trimf', [-900 -360 0]
MF4='ZE': 'trimf', [-360 0 360]
MF5='PS': 'trimf', [0 360 900]
MF6='PM': 'trimf', [360 900 1800]
MF7='PB': 'trapmf', [900 1620 1800 1800]
```

```
[Input2]
Name='DQe'
Range=[-180 180]
NumMFs=7
MF1='NB': 'trapmf', [-180 -180 -162 -90]
MF2='ZE': 'trimf', [-36 0 36]
MF3='PM': 'trimf', [36 90 180]
MF4='NM': 'trimf', [-180 -90 -36]
MF5='NS': 'trimf', [-90 -36 0]
MF6='PS': 'trimf', [0 36 90]
MF7='PB': 'trapmf', [90 162 180 180]
```

```
[Output1]
Name='Output'
Range=[-600 600]
NumMFs=9
MF1='NB': 'trapmf', [-600 -600 -540 -300]
MF2='NS': 'trimf', [-300 -120 0]
MF3='ZE': 'trimf', [-120 0 120]
MF4='PS': 'trimf', [0 120 300]
MF5='PB': 'trapmf', [300 540 600 600]
MF6='NM': 'trimf', [-600 -300 -120]
MF7='NVS': 'trimf', [-180 -60 0]
MF8='PVS': 'trimf', [0 60 180]
MF9='PM': 'trimf', [120 300 600]
```