

342/H/06



TESIS
(LS 2340)

**PENERAPAN FUZZY KONTROLER
PADA SISTEM INJEKSI BAHAN BAKAR
MESIN DIESEL KAPAL**

Oleh :

EMIE SANTOSO

NRP : 4103 204 002

RTPe

9.801 SII 313

San

p-1

2005



| PERPUSTAKAAN ITS | |
|---------------------|---------|
| Tgl. Terima | 10-3-06 |
| Terima Dari | H |
| No. Agenda Prp. | 724945 |

**PROGRAM PASCA SARJANA
PROGRAM STUDI TEKNIK KELAUTAN
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2005**

LEMBAR PENGESAHAN

PENERAPAN FUZZY KONTROLER PADA SISTEM INJEKSI BAHAN BAKAR MESIN DIESEL KAPAL

Tesis ini disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Teknik (M.T.)

di

FAKULTAS TEKNOLOGI KELAUTAN
INSTITUT TEKNOLOGI SEPULUH NOPEMBER SURABAYA

Oleh :

EMIE SANTOSO

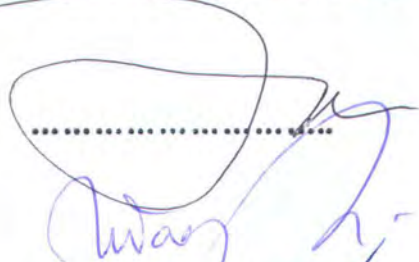
NRP : 4103 204 002

Disetujui oleh Tim Penguji Tesis :

Tanggal Ujian : 27 Juli 2005

Periode Wisuda : September 2005

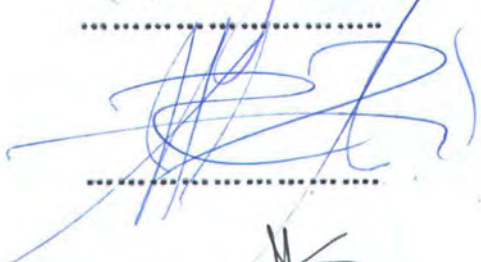
1. Ir. Aguk Zuhdi, MF, M.Eng.

.....


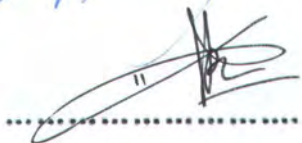
2. Dr. A.A. Masroeri, M.Eng.

.....

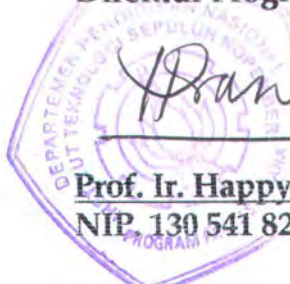
3. Ir. Surjo Widodo Adji, MSc.



4. Indra Ranu K, ST, MSc.

.....


Direktur Program Pascasarjana





Prof. Ir. Happy Ratna S., M.Sc., Ph.D.

NIP. 130 541 829

LEMBAR PERSEMBAHAN

*Dengan segenap rasa cinta kupersembahkan kepada
Ibunda, Suami dan anak-anakku tercinta.*

THE APPLICATION OF FUZZY CONTROLLER
AT THE FUEL INJECTION SYSTEM OF SHIP DIESEL ENGINE

By : Emie Santoso
Under the supervision I : Indra Ranu Kusuma ST, M.Sc
II : Dr. Ir. AA Masroeri M.Eng

ABSTRACT

Diesel Engine is a primary engine in ship which drives propeller. An important mechanism of the diesel engine to study is its fuel injection system into the cylinder. In the following study, the application of fuzzy logic controller into the fuel injection system of marine diesel engine would be discussed. First, details of the fuel injection system, in this case the EUI (Electronic Unit Injector), the working principles and its system elements, must be learned. Then, a mathematical model of its system elements including the use of sensor and actuator on its control system was developed. From the model, block diagrams and the sub-systems, and eventually the block diagram of the total system were found. Simulation of fuel injection control system using the fuzzy logic controller was done by utilizing matlab 6.1 program. To simplify the simulation, "GUIDE" facility in matlab 6.1 developed, to show the system block diagrams and its responses. The result is time response obtained, is much better than the result of without fuzzy logic controller.

Keywords : EUI, fuzzy logic control, fuel injection system.

KATA PENGANTAR

Syukur Alhamdulillah kami panjatkan ke hadirat Allah SWT atas segala rahma dan hidayahnya, serta sholawat serta salam kami tujukan kepada Nabi Muhammad Saw, sehingga kami dapat menyelesaikan tesis dengan judul “ Penerapan Fuzzy Kontroler Pada Sistem Injeksi bahan bakar Mesin Diesel Kapal “ tepat pada waktu yang ditentukan.

Tesis ini merulakan salah satu syarat untuk memperoleh gelar Magister Teknik (S2) yang terdapat dalam kurikulum Program Pasca Sarjana, Bidang keahlian Teknik Sistem dan Pengendalian Kelautan, Institut Teknologi Sepuluh Nopember Surabaya.

Tesis ini berisikan tentang penerapan kontroler logika fuzzy pada sistem injeksi bahan bakar mesin diesel kapal, dan mensimulasikan sisyem pengendalian tersebut dengan menggunakan program mathlab, untuk mengetahui time responnya.

Tesis ini dimaksudkan untuk dapat dijadikan bahan pertimbangan dalam menentukan kontroler terbaik untuk sistem injeksi bahan bakar mesin diesel kapal dan sebagai alternatif lain dalam menentukan kontroler pada sistem injeksi bahan bakar.

Harapan kami semoga laporan tesis ini dapat bermanfaat bagi civitas akademik Institut Teknologi Sepuluh Nopember pada umumnya. Kami menyadari bahwa dalam penulisan laporan tesis ini jauh dari kesempurnaan dan banyak kekurangannya, sehingga kami berharap masukan-masukan dan

kritik-kritik yang bersifat membangun yang nantinya akan bermanfaat bagi kesempurnaan dan pengembangan lebih lanjut dari tesis ini.

Akhir kata, semoga tesis ini bermanfaat untuk kemajuan ilmu pengetahuan dan teknologi, bangsa, dan agama, amin.

Surabaya, Juli 2005

Emie Santoso

4103 204 002

UCAPAN TERIMA KASIH

Terselesaikannya tesis ini tidak lepas dari dukungan dan bantuan dari banyak pihak, baik dukungan moral dan material. Pada kesempatan ini penulis dengan tulus , mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Yang terhormat dan tersayang orang tua, ibu Murti, ibu Sri Jasaningsih, Bpk Hadi Suratin, atas dukungan dan doa restu yang selalu diberikan kepada kami, sehingga kami bisa menyelesaikan tesis ini.
2. Suami dan anak-anakku tercinta, yang selama ini selalu memberikan dukungan, pengertian, pengorbanan dan kasih sayang sehingga menguatkan tekad kami dalam menyelesaikan tesis ini. *“Maafkan mama, waktu untuk kalian banyak yang tersita. Mama mencintai kalian semua”*.
3. Bapak Indra Ranu Kusuma, ST,M.Sc selaku Pembimbing I, atas segala kesabaran, dan kesediaannya meluangkan banyak waktu untuk membimbing dan mengarahkan kami.
4. Bapak Dr. Ir. AA Masroeri M.Eng, selaku Pembimbing II atas masukan-masukan dan bimbingannya dalam menyelesaikan tesis ini.
5. Bapak Indrajaya, yang memberikan masukan-masukan pada kami, sehingga tesis ini menjadi lebih baik.

6. Bapak Imam Sutrisno MT, yang banyak membantu kami dalam menyelesaikan tesis ini.
7. Rekan-rekan kerja, di Politeknik Perkapal Negeri Surabaya, atas dukungan dan bantuannya
8. Agung Sukma Sedayu, yang banyak membantu dikala ada masalah dengan komputer kami.
9. Teman-teman di Teknik Sistem dan Pengendalian Kelautan, Beny, Pak Ferry, Pak Dani, terima kasih atas kekompakan, dan kerjasamanya.

Surabaya, Juli 2005

Emie Santoso

4103 204 002

DAFTAR ISI

| | Halaman |
|------------------------------------|---------|
| ABSTRAK | i |
| ABSTRACT | ii |
| KATA PENGANTAR | iii |
| UCAPAN TERIMAKASIH | v |
| DAFTAR ISI | vii |
| DAFTAR GAMBAR | x |
| DAFTAR TABEL | xiv |
| BAB. I. PENDAHULUAN | 1 |
| 1.1 Latar Belakang | 1 |
| 1.2 Permasalahan | 2 |
| 1.3 Batasan Masalah | 3 |
| 1.4 Tujuan | 4 |
| 1.5 Manfaat | 5 |
| 1.6 Sistematika Penulisan | 5 |
| BAB. II TEORI DASAR | 7 |
| 2.1 EUI (Electronic Unit Injektor) | 7 |
| 2.2 Teori Logika Fuzzy | 12 |
| 2.2.1 Definisi Himpunan Fuzzy | 14 |
| 2.2.2 Fungsi Keanggotaan | 16 |
| 2.2.3 Variabel Linguistik | 17 |
| 2.2.4 Linguistic Hedge | 19 |

| | |
|---|----|
| 2.2.5 Operasi Himpunan Fuzzy | 19 |
| 2.2.6 Kontroler Logika Fuzzy | 24 |
| 2.2.7 Struktur Dasar Kontroler Logika Fuzzy | 25 |
| BAB. III METODOLOGI | 30 |
| 3.1 Diagram Alir | 30 |
| 3.2 Penjelasan Diagram Alir | 31 |
| BAB.IV PEMODELAN MATEMATIS UNTUK DINAMIKA ELEMEN SISTEM INJEKSI BAHAN BAKAR MESIN DIESEL KAPAL DAN PERENCANAAN KONTROLER LOGIKA FUZZY | 34 |
| 4.1 EUI (Electronic Unit Injector) Sistem Bahan Bakar | 34 |
| 4.2 Dinamika Mesin Diesel | 36 |
| 4.3 Dinamika Transducer Kecepatan | 39 |
| 4.4 Dinamika Aktuator (Solenoid) | 40 |
| 4.5 Data dan Pemodelan Sistem | 45 |
| 4.5.1 Diagram Blok Sistem Injeksi Bahan Bakar dalam State-space | 46 |
| 4.5.2 Pemodelan Matematis Blok diagram Sistem dalam Model Fungsi Alih, tanpa Kontroler Logika Fuzzy | 48 |
| 4.6 Kontroler Logika Fuzzy | 50 |
| 4.6.1 Variabel Masukan dan Variabel Keluaran | 50 |
| 4.6.2 Strategi Pengaturan Kontroler Logika Fuzzy | 50 |
| 4.6.3 Basis Data | 50 |
| 4.6.4 Penyusunan Aturan Dasar | 53 |
| 4.6.5 Logika Pengambilan Keputusan | 56 |
| 4.6.6 Defuzzifikasi | 57 |

| | |
|---|-----------|
| 4.6.7 Diagram Blok Sistem dalam Model Fungsi Alih Dengan Kontroler Logika Fuzzy | 57 |
| BAB. V SIMULASI DAN ANALISA SISTEM | 58 |
| 5.1 Metode Pengujian | 63 |
| 5.2 Analisa Hasil Pengujian | 64 |
| 5.2.1 Pengujian Sistem dengan Model State-Space dibanding dengan Model Fungsi Alih (Sebelum di fuzzy) | 64 |
| 5.2.2 Pengujian Sistem dengan Model Fungsi Alih, Tanpa Kontroler Logika Fuzzy Dibanding dengan Sistem yang Menggunakan Kontroler Logika Fuzzy | 65 |
| BAB. VI KESIMPULAN DAN SARAN | 75 |
| 6.1 Kesimpulan | 75 |
| 6.2 Saran | 77 |
| DAFTAR PUSTAKA | 78 |
| LAMPIRAN A. PROGRAM SIMULASI DENGAN MATLAB | 80 |
| LAMPIRAN B. HASIL SIMULASI | 88 |

DAFTAR GAMBAR

| | Halaman |
|---|---------|
| Gambar 2.a dan 2.b Gambar keseluruhan sistem EUI | 8 |
| Gambar 2.2 Konsep dasar kerja rocker arm dan solenoid pengontrol valve pada EUI. | 9 |
| Gambar 2.3 Cara kerja sistem injektor (EUI) | 11 |
| Gambar 2.4 Penyajian secara grafis himpunan fuzzy A | 15 |
| Gambar 2.5 Grafik fungsi keanggotaan | 16 |
| Gambar 2.6 Definisi himpunan fuzzy dengan variabel linguistik lambat, Sedang dan cepat. | 18 |
| Gambar 2.7 Definisi himpunan fuzzy lambat, sedang, dan cepat | 20 |
| Gambar 2.8 Grafik irisan dari himpunan fuzzy lambat dan sedang | 21 |
| Gambar 2.9 Grafik produk aljabar dari himpunan fuzzy lambat dan sedang | 22 |
| Gambar 2.10 Grafik gabungan (union) antara himpunan fuzzy sedang dan cepat | 23 |
| Gambar 2.11 Grafik produk aljabar antara himpunan fuzzy sedang dan cepat | 23 |
| Gambar 2.12 Grafik komplemen antara himpunan fuzzy cepat dan tidak cepat | 24 |
| Gambar 2.13 Kontroler logika fuzzy | 25 |
| Gambar 4.1 Gambar penyederhanaan sistem injeksi bahan bakar yang dikendalikan secara elektronik | 35 |
| Gambar 4.2 Gambar diagram blok sistem injeksi bahan bakar yang dikendalikan secara elektronik. | 35 |
| Gambar 4.3 Diagram blok sistem state-space plan (mesin diesel kapal) | 37 |

| | |
|--|----|
| Gambar 4.4 Blok diagram dinamika mesin diesel kapal dalam bentuk transfer function. | 39 |
| Gambar 4.5 Diagram blok untuk sensor | 40 |
| Gambar 4.6 Magnet solenoida | 41 |
| Gambar 4.7 Diagram blok state-space dari subsistem solenoid | 43 |
| Gambar 4.8 Diagram blok solenoida dalam transfer function. | 45 |
| Gambar 4.9 Diagram blok sistem injeksi bahan bakar dalam bentuk State-space. | 46 |
| Gambar 4.10 Diagram blok solenoid | 47 |
| Gambar 4.11 Diagram blok mesin diesel | 47 |
| Gambar 4.12 Diagram blok subsistem sensor. | 48 |
| Gambar 4.13 Diagram blok subsistem amplifier | 48 |
| Gambar 4.14 Diagram blok sistem injeksi bahan bakar dalam model fungsi alih tanpa kontroler logika fuzzy. | 48 |
| Gambar 4.15 Diagram blok subsistem mesin diesel kapal | 49 |
| Gambar 4.16 Diagram blok subsistem solenoid | 49 |
| Gambar 4.17 Diagram blok subsistem sensor | 49 |
| Gambar 4.18 Diagram blok subsistem amplifier | 49 |
| Gambar 4.19 Diagram blok sistem dalam model fungsi alih dengan kontroler logika fuzzy | 57 |
| Gambar 5.1 Fungsi keanggotaan error | 59 |
| Gambar 5.2 Fungsi keanggotaan kecepatan perubahan putaran mesin | 59 |
| Gambar 5.3 Fungsi keanggotaan output kontroler | 60 |
| Gambar 5.4 Tampilan dalam pembuatan aturan (rule) | 61 |

| | | |
|-------------|---|----|
| Gambar 5.5 | Aturan-aturan dalam bentuk grafis | 62 |
| Gambar 5.6 | Aturan-aturan dalam bentuk tiga dimensi | 62 |
| Gambar 5.7 | Tampilan untuk mensimulasikan sistem, yang dinuat dengan GUIDE | 63 |
| Gambar 5.8 | Grafik sstep respon hasil simulasi sistem dalam model fungsi alih Dengan input step 1000 rpm, dan pembebanan 1 Nm | 64 |
| Gambar 5.9 | Grafik step respon hasil simulasi sistem dengan model state-space Saat diberi input step 1000 rpm, dengan beban 1 Nm | 65 |
| Gambar 5.10 | Grafik step respon hasil simulasi sistem tanpa kontroler logika Fuzzy, dengan setting point 1000 rpm, dan beban konstan 100 Nm | 66 |
| Gambar 5.11 | Grafik step respon hasil simulasi sistem dengan kontroler logika Fuzzy, dengan setting point 1000 rpm, dan beban konstan 100 Nm | 67 |
| Gambar 5.12 | Grafik step respon hasil simulasi sistem tanpa kontroler logika Fuzzy, dengan setting point 1000 rpm, dan beban sinusoidal dengan amplitudo 500 Nm | 67 |
| Gambar 5.13 | Hasil zoom grafik step respon hasil simulasi sistem tanpa kontroler logika fuzzy, dengan setting point 1000 rpm, dan beban sinusoidal dengan amplitudo 500 Nm | 68 |
| Gambar 5.14 | Grafik step respon hasil simulasi sistem dengan kontroler logika Fuzzy, dengan setting point 1000 rpm, dan beban sinusoidal dengan amplitudo 500 Nm | 68 |
| Gambar 5.15 | Hasil zoom grafik step respon hasil simulasi sistem | |

Dengan kontroler logika fuzzy, dengan setting point 1000 rpm,
dan beban sinusoidal dengan amplitudo 500 Nm 68

Gambar 5.16 Grafik hubungan antara perubahan setting point (putaran mesin),
Terhadap settling time (T_s), pada keadaan beban konstan
(1000 Nm) 72

Gambar 5.17 Grafik hubungan antara perubahan beban terhadap settling time (T_s)
Pada keadaan setting point konstan (1000 rpm) 73

DAFTAR TABEL

| | Halaman |
|---|---------|
| Tabel 4.1 Pendefinisian bentuk segitiga uniform. | 52 |
| Tabel 4.2 Tabel Mac Vicar-Whelan. | 54 |
| Tabel 5.1 Tabel hasil simulasi step respon sistem pengendalian tanpa Kontroler logika fuzzy, dan dengan kontroler logika fuzzy, pada Sistem bahan bakar mesin diesel kapal. | 70 |
| Tabel 5.2 Hasil simulasi step respon sistem pengendalian tanpa kontroler fuzzy dan dengan kontroler logika fuzzy, dengan setting point 1000 rpm, pada kondisi beban(konstan) yang berbeda-beda. | 70 |
| Tabel 5.3 Hasil simulasi step respon sistem pengendalian tanpa kontroler fuzzy dan dengan kontroler logika fuzzy, dengan kondisi beban konstan 1000 Nm, pada kondisi setting point yang berbeda-beda. | 70 |



BAB I
PENDAHULUAN

BAB I

PENDAHULUAN

1.1. LATAR BELAKANG

Pada saat ini sejumlah aplikasi logika *fuzzy* telah banyak digunakan oleh banyak kalangan di industri berat. Di Jepang, sudah banyak ditawarkan produk komersial yang diperkaya dengan teknologi *fuzzy* seperti autofocus, transmisi otomatis dari mobil, kontrol AC, kulkas dan elevator.

Berangkat dari kenyataan diatas, maka perlu kiranya teknologi kontroler logika *fuzzy* ini diterapkan lebih lanjut pada teknologi perkapalan, salah satu aplikasi teknologi ini adalah pengendalian sistem injeksi bahan bakar ke dalam silinder mesin diesel kapal. Pengendalian sistem injeksi bahan bakar merupakan mekanisme terpenting dalam mesin diesel yang harus dipelajari. Waktu penginjeksian dan jumlah bahan bakar yang tepat akan mempengaruhi performansi mesin, misalnya kecepatan putaran mesin, daya yang dihasilkan, SFOC, bahkan gas buang yang dihasilkan. Pengendalian system injeksi bahan bakar harus berjalan secara efektif dan otomatis untuk menghindari kemungkinan- kemungkinan yang tidak dikehendaki, dan untuk mendapatkan performansi mesin yang bagus, sesuai dengan yang dikehendaki.

Pada keadaan tertentu, beban baling-baling kapal tidak konstan, karena disebabkan oleh adanya gelombang. Untuk mengatasi adanya perubahan beban tersebut, maka setiap mesin diesel kapal dilengkapi dengan kontroler yang mengatur kecepatan putaran motor. Pengendalian kecepatan putaran mesin diesel ini dapat dilakukan dengan cara mengendalikan sistem injeksi bahan bakarnya.

Penerapan kontroler logika *fuzzy* merupakan salah satu solusi yang diharapkan bisa mengendalikan system injeksi bahan bakar dengan lebih baik dari pada system pengendalian system injeksi bahan bakar lain yang ada saat ini.

1.2. PERMASALAHAN

Sistem injeksi bahan bakar yang ada saat ini ada dua macam yaitu sistem injeksi bahan bakar yang dikendalikan secara mekanik, dan system injeksi bahan bakar yang dikendalikan secara elektrik, yang biasa disebut dengan EUI (electronic unit injector).

Pada sistem injeksi bahan bakar yang dikendalikan secara mekanik, kontroler yang digunakan adalah governor. Pada system ini bahan bakar diinjeksikan ketika *rocker arm* menekan bagian atas dari injektor bahan bakar kebawah. Gerakan *rocker arm* tersebut dikontrol oleh *camshaft* melalui *lifter assembly* dan kemudian menekan *control rod*. Sedang jumlah bahan bakar yang diinjeksikan dikontrol oleh *rack*. Gerakan *rack* ini menyebabkan gerak memutar dari gear alat pengunci dari plunyer, sehingga plunyer berputar dari kedudukan semula. Berputarnya plunyer ini merubah *effective stroke* (stroke selama bahan bakar secara actual diinjeksikan ke silinder) dari plunyer. Waktu penginjeksian dipengaruhi oleh dua factor yaitu angular location dari camshaft dan lokasi dari plunyer. Angular location dari camshaft dikontrol oleh camshaft drive gears di bagian belakang mesin, dan lokasi plunyer dapat diatur dengan screw.

Pada sistem injeksi bahan bakar yang dikendalikan secara elektrik (EUI), waktu atau awal penginjeksian, lamanya penginjeksian semuanya dikontrol oleh ECM (electronic control modul). ECM ini menerima sinyal dari sensor-sensor yang

digunakan seperti *boost* mesin turbo, temperatur saluran masuk udara, temperatur bahan bakar, temperatur oli, temperatur agen pendingin, dan kecepatan putaran mesin [1]. Kemudian ECM menentukan dan menghitung sinyal listrik PWM (pulse-width modulation) berdasar pada daftar kalibrasi yang sudah ditentukan yang ada dalam memorinya untuk mengontrol saat penginjeksian bahan bakar dalam ruang bakar. Aktuator yang digunakan adalah solenoid. Saat solenoid menerima sinyal dari ECM, solenoid bekerja, dan menutup *fuel spill valve*, sehingga tekanan bahan bakar dalam injektor cukup besar untuk membuka *nozle check valve* dan bahan bakar dapat disemprotkan kedalam ruang bakar. Jika sinyal yang diberikan ECM ke solenoid selesai maka *fuel spill valve* terbuka, tekanan bahan bakar drop, seketika itu bahan bakar berhenti diinjeksikan. Semakin lama sinyal yang diterima oleh solenoid, maka *fuel spill valve* akan tertutup lebih lama dan jumlah bahan bakar yang disemprotkan menjadi lebih banyak. Lamanya *fuel spill valve* menutup menentukan jumlah bahan bakar yang diinjeksikan dalam ruang bakar.

Pada thesis ini akan dibahas penerapan kontroler logika *fuzzy* pada sistem injeksi bahan bakar yang menggunakan kontrol elektrik (EUI). Dengan titik berat pembahasan pada pengendalian jumlah bahan bakar yang diinjeksikan dalam ruang bakar. Dalam thesis ini juga dibuat software untuk mendemokan hasil pengendalian dengan menggunakan fasilitas matlab, GUI.

1.3. BATASAN MASALAH

Untuk mencegah melebarnya permasalahan dalam thesis ini, maka perlu adanya batasan batasan dalam pembahasannya yaitu:

1. Pokok pembahasan dalam tesis ini adalah penerapan kontroler logika *fuzzy* untuk mengendalikan jumlah bahan bakar yang diinjeksikan kedalam ruang bakar, yaitu dengan cara mengontrol berapa lama ECM memberikan sinyal ke solenoid yang bertindak sebagai aktuator pada aksi pengendalian ini.
2. Penerapan kontroler logika *fuzzy* yang dibahas dalam tesis ini hanya terbatas pada sistem injeksinya saja, dengan masukan tunggal yaitu kecepatan putaran mesin. Sedang masukan –masukan lain yang ada dikompensasikan dalam sebuah kompensator.
3. Didalam tesis ini tidak dibahas tentang pembuatan system fisik dan keuntungan secara ekonomis dari penerapan kontroler logika *fuzzy* ini.

1.4. TUJUAN

Tujuan penelitian dalam tesis ini adalah:

- Untuk menerapkan teknologi kontroler logika *fuzzy* pada sistem injeksi bahan bakar mesin diesel kapal, sehingga mampu mengatur dan mempertahankan kecepatan putaran mesin diesel sesuai dengan yang diinginkan.
- Mendapatkan grafik respon dinamik sistem kontrol, untuk mengetahui bagaimana kemampuan kontroler yang digunakan dalam mengendalikan system injeksi bahan bakar.
- Membuat software dengan menggunakan fasilitas GUI dalam matlab, untuk memudahkan mensimulasikan hasil pengendalian.

1.5. MANFAAT

Penelitian ini diharapkan dapat dijadikan sebagai bahan pertimbangan dalam menentukan kontroler terbaik untuk mesin diesel kapal, serta sebagai alternatif lain dalam menentukan system pengendalian system injeksi bahan bakar mesin diesel kapal. Disamping itu software yang dibuat dapat dimanfaatkan untuk mempelajari dan mendapatkan dengan lebih mudah respon dinamik system pengendalian system injeksi bahan bakar .

1.6 SISTEMATIKA PENULISAN

Laporan tesis ini terdiri atas:

Halaman Judul

Lembar Pengesahan

Abstrak

Kata Pengantar

Daftar Isi

Daftar Gambar

Daftar Tabel

BAB I Pendahuluan

Bab ini akan membahas tentang latar belakang penulisan tesis ini, perumusan masalah, pembatasan masalah, serta manfaat dan tujuan penulisan tesis ini.

BAB II Dasar Teori

Bab ini membahas tentang teori-teori yang mendasari dan digunakan untuk menyelesaikan permasalahan yang diangkat dalam tesis ini. Sesuai dengan



permasalahan yang diangkat maka dalam bab ini akan dijelaskan tentang sistem injeksi bahan bakar mesin diesel yang dikontrol secara elektrik, lebih tepatnya injektornya yang dikenal dengan *Electrical Unit Injector* (EUI), kemudian juga dibahas tentang teori logika *fuzzy*.

BAB III Metodologi

Pada metodologi ini akan dibahas tahapan-tahapan yang harus dilakukan untuk menyelesaikan permasalahan yang ada. Dijelaskan dengan menggunakan diagram alir dan kemudian dijelaskan lagi lebih rinci.

BAB IV Dinamika Mesin Diesel dan Pengontrol Logika Fuzzy

Dalam bab ini ada dua pokok bahasan, yaitu mengenai dinamika mesin diesel kapal, dan pengontrol logika *fuzzy*. Pada pokok bahasan dinamika mesin diesel, akan diturunkan model matematis semua komponen dalam sistem, dan dimodelkan dalam bentuk diagram blok. Dalam pokok bahasan pengontrol logika *fuzzy* akan dibahas proses perancangan kontroler logika *fuzzy*, mulai dari menentukan input dan output nya, sampai membuat aturan-aturan *fuzzy*.

BAB V Simulasi dan Analisa Sistem.

Dalam bab ini akan ditampilkan beberapa hasil pengujian, yang menggunakan kontroler logika *fuzzy*, maupun yang menggunakan kompensator P, PI, PID.

BAB VII Kesimpulan dan Saran

Bab ini merupakan penutup dari tesis ini.



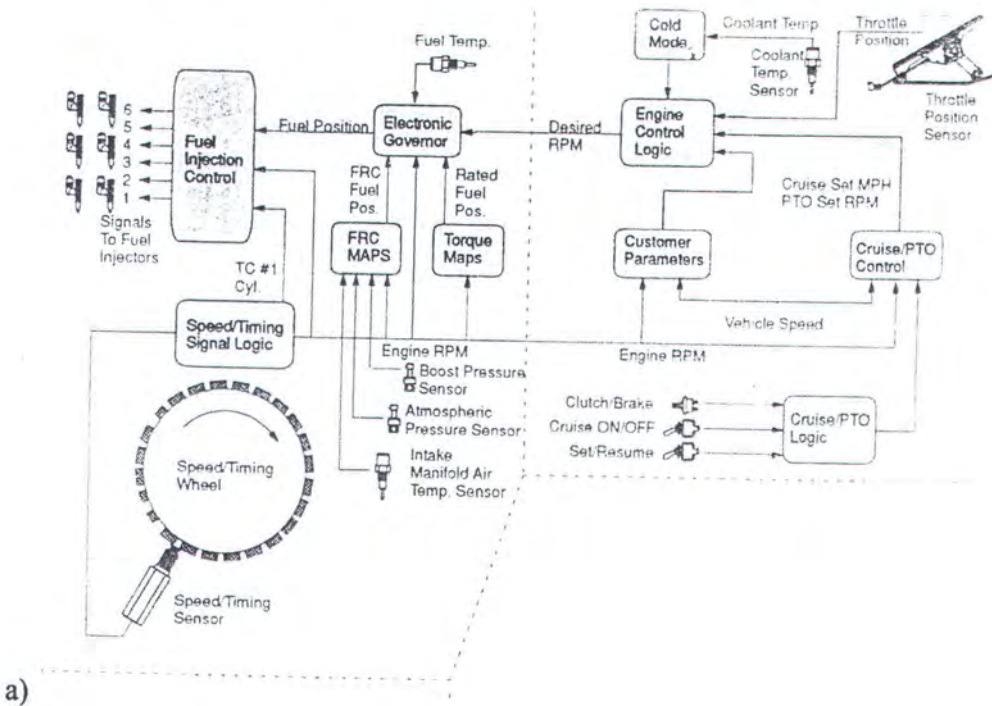
BAB II
TEORI DASAR

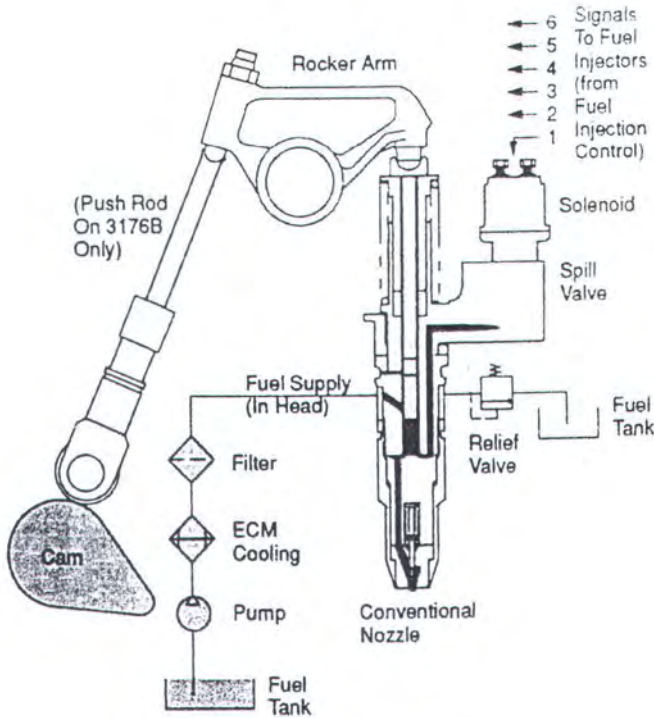
BAB II

TEORI DASAR

2.1 EUI (Electronic Unit Injector)

Caterpillar memperkenalkan sistem injeksi bahan bakar yang dikontrol secara elektronik (EUI) pertama kali pada tahun 1987, yaitu pada model 3406 B. Tidak ada *rack* bahan bakar yang dijalankan secara mekanik pada unit injektor yang dikendalikan secara elektronik. Oleh karena itu, ukuran jumlah bahan bakar dan penentuan waktu penginjeksian semua dikendalikan oleh sinyal yang dibangkitkan di kontroler unit injektor bahan bakar yang disebut ECM (electronic control module). Gambar keseluruhan sistem EUI yang digunakan oleh Caterpillar dapat dilihat pada gambar 2.1 dibawah ini.





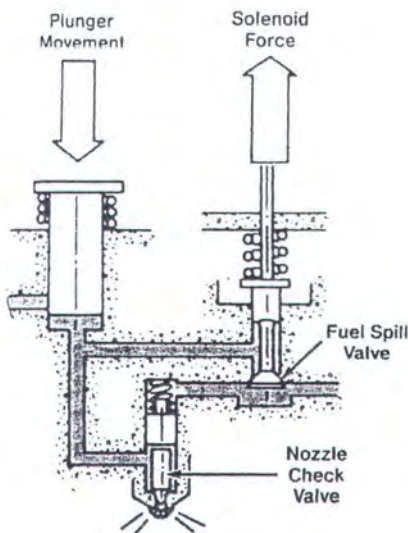
b)

Gambar 2.1 a dan b Gambar keseluruhan sistem EUI

ECM bekerja berdasarkan berbagai inputan dari berbagai sensor antara lain, *boost* mesin turbo, temperatur saluran masuk udara, temperatur bahan bakar, temperatur bahan bakar, temperatur oli, temperatur agen pendingin, kecepatan putaran mesin, dan posisi *throttlet*. Kemudian ECM memberikan sinyal PWM (pulse – width modulation) ke solenoid yang menjadi actuator pada sistem ini. Semakin lama solenoid diberi tenaga atau menerima sinyal dari ECM, semakin besar jumlah bahan bakar yang diinjeksikan ke dalam ruang bakar. Sebaliknya semakin pendek sinyal PWM nya, semakin sedikit volume bahan bakar yang diinjeksikan ke dalam ruang bakar.

Pembangkitan tekanan bahan bakar yang cukup tinggi untuk tujuan penginjeksian membutuhkan gerakan dari *rocker arm assembly*. Gambar 2.2 menunjukkan plunyer injektor internal, yang dipaksa turun oleh *rocker arm* dalam

barrel / pakin-nya. Perhatikan bahwa *spill valve* ukuran kecil ditunjukkan disebelah kanan dari diagram. *Spill valve* ini dijaga agar tetap terbuka oleh pegas yang akan menjaga peningkatan tekanan bahan bakar yang dihasilkan oleh pompa penyedia bahan bakar. Saat plunyer injektor bahan bakar bergerak turun, bahan bakar akan mengalir tumpah dari katup ini dan kembali ke tangki bahan bakar. Supaya penginjeksian dapat dilakukan, *spill valve* ini harus ditutup oleh sinyal dari ECM yang memberi daya solenoid elektrik, yang terletak diatas injektor. Saat solenoid diberi daya oleh sinyal PWM dari ECM, pergerakan turun plunyer injektor akan membuat peningkatan yang cepat pada tekanan bahan bakar yang terjebak dibawahnya. Saat tekanan ini cukup tinggi *needle valve* dalam ujung injektor penyemprot akan terbuka dan menyebabkan bahan bakar diinjeksikan ke dalam ruang bakar. Begitu solenoid tidak diberi daya lagi dari ECM, *spill valve* terbuka oleh pegasnya dan tekanan dalam injektor turun seketika, dan penginjeksian bahan bakar akan berhenti seketika.

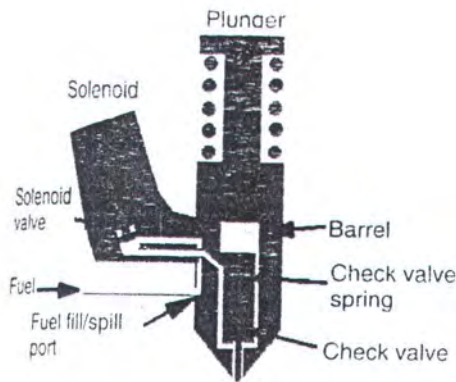


Gambar 2.2 Konsep dasar kerja rocker arm dan solenoid pengontrol valve, pada EUI.

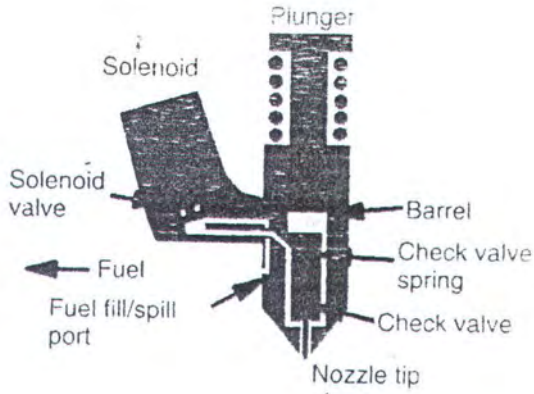
Perbedaan dasar antara plunyer injektor yang dijalankan secara mekanik dan plunyer injektor yang digunakan pada EUI adalah bahwa tidak adanya *helix* pada plunyer injektor elektronik. Plunyerinya hanya berupa plunyer solid.

Lebih jelasnya, empat langkah cara kerja dari EUI (electronic unit injector) ditunjukkan dan dijelaskan dalam gambar 2.3 a s/d d. dibawah ini.

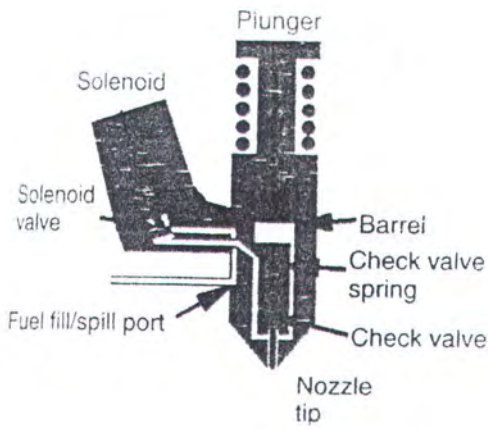
a. Tanpa tekanan yang diberikan ke plunyer dari *cam* yang menggerakkan *rocker arm*, pegas akan menjaga plunyer tertarik kembali. Didorong oleh tekanan bahan bakar yang lemah dari pompa bahan bakar, bahan bakar mengalir masuk ke injektor melalui *fill/spill port*. Dari sini aliran melalui *solenoid valve* turun melewati saluran injektor internal dan kembali kedalam tangki. Tekanan pompa terlalu lemah untuk mengangkat *check valve* pada ujung injektor.



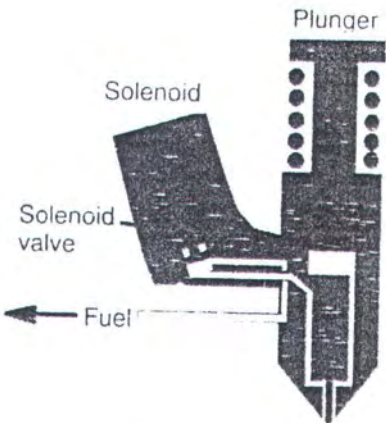
a)



b)



c)



d)

Gambar 2.3 Cara kerja sistem injektor (EUI)

b. Putaran *cam* mulai menggerakkan plunyer ke bawah. Injeksi bahan bakar bisa terjadi pada saat setelah plunyer mulai bergerak kebawah. Sampai ECM memberikan sinyal untuk penginjeksian, bahan bakar yang dipindahkan didorong keluar melalui *solenoid valve* kembali ke *fill/spill port*.

c. Ketika ECM memberikan sinyal untuk memulai penginjeksian, solenoid menarik *fuel valve* menjadi tertutup, dan menutup saluran bahan bakar menuju *fill/spill port*.

Dengan tertutupnya *valve* tersebut, tekanan bertambah pada ujung injektor sampai 37.931 kPa (5500 psi), yang diperlukan untuk menggeser atau mengangkat *check valve*, dan bahan bakar diinjeksikan ke dalam ruang bakar.

d. Bahan bakar akan terus diinjeksikan ke dalam ruang bakar sampai ECM berhenti mengirim sinyal ke solenoid. Pada saat itu solenoid membuka valvenya dan tekanan dalam injektor tiba-tiba drop, dan *check valve* berhenti menginjeksikan bahan bakar ke dalam ruang bakar. Kemudian plunyer melanjutkan gerakannya dan memindahkan bahan bakar melalui *valve* yang terbuka dan kembali ke dalam tangki. Aliran bahan bakar ini membantu mendinginkan injektor.

Waktu penginjeksian (awal penginjeksian) dan jumlah bahan bakar yang diinjeksikan dikontrol oleh sinyal dari ECM menuju ke EUI. Semakin lama solenoid diberi daya (menerima sinyal dari ECM) semakin lama *valve* solenoid tertutup, dan semakin lama pula waktu penginjeksian terjadi. Akibatnya semakin banyak jumlah bahan bakar yang diinjeksikan ke dalam ruang bakar.

2.2 TEORI LOGIKA FUZZY

Konsep teori *fuzzy* diperkenalkan oleh Lotfi A Zadeh, pada tahun 1965 untuk mempresentasikan dan menganalisa fenomena-fenomena di alam nyata yang serba

samar atau tidak tepat ditinjau dari cara berfikir manusia mengenai fenomena-fenomena tersebut.

Pada dasarnya logika manusia dalam pengenalan atau pendekatan terhadap suatu obyek, selalu didasarkan pada klasifikasi secara kualitatif atau sering disebut sebagai *terminology linguistik*. Contohnya, manusia diklasifikasikan dengan variabel linguistik besar, sedang, kecil, tinggi, pendek dan sebagainya. Sifat kebenaran yang dikandung oleh suatu variabel linguistik tersebut adalah samar/ kabur, karena batas antara kebenaran yang satu dengan kebenaran yang lain tidak tegas atau tidak jelas.

Sebagai contoh: suatu pernyataan menyebutkan “ Seandainya cuaca *cerah*, saya akan *sering* berolahraga”. Seberapa *cerah* dan seberapa *sering*, hal itu sulit untuk dibuat batasan dengan dasar-dasar kuantitatif. Jika *cerah* dikategorikan sebagai kondisi langit yang terang dengan tidak ada awan sama sekali, lalu bagaimana dengan kondisi langit yang sekitar 1-5% berawan, apakah bisa dikategorikan *tidak cerah*. Hal tersebut tidak dapat dibenarkan karena pada dasarnya selisih 1-5% berawan tidak dapat dijadikan suatu batasan untuk tidak dikatakan *cerah*. Sehingga semua pernyataan tersebut dapat diterima oleh logika tanpa memperhitungkan batas-batas kondisi sebenarnya. Kebenaran inilah yang disebut dengan *kebenaran Fuzzy*.

Konsep kekaburan dan ketidakpastian tersebut yang melatarbelakangi konsep logika *fuzzy*. Ide awalnya adalah bagaimana cara menyajikan ke-kabur-an. Penyajian tersebut harus cukup menggambarkan ke-kabur-an tetapi dilain pihak harus cukup sederhana sehingga komputasinya lebih mudah. Logika *fuzzy* atau logika kabur adalah metodologi untuk menyatakan hukum operasional dari suatu sistem dengan ungkapan-ungkapan bahasa (variabel linguistik), bukan dengan persamaan matematis.

Banyak sistem yang terlalu kompleks untuk dimodelkan secara akurat, meskipun dengan persamaan yang kompleks. Dalam kasus semacam ini, ungkapan bahasa yang digunakan dalam logika *fuzzy* dapat membantu dalam mendefinisikan karakteristik operasional sistem dengan lebih baik. Variabel linguistik tersebut merubah sinyal-sinyal yang berupa variabel numerik *crisp*, diubah kedalam suatu pernyataan yang menggambarkan kekaburan dan ketidakjelasan batasan-batasan suatu kebenaran.

Aturan yang dipakai dalam logika *fuzzy* tidak berdasarkan pada logika yang hanya mempunyai dua nilai kebenaran saja, yaitu benar dan salah, ya atau tidak, akan tetapi kebenaran dalam logika *fuzzy* adalah multi nilai yang bervariasi dari suatu kebenaran ke kebenaran yang lain atau bervariasi dari 0 (salah), sampai 1 (benar).

2.2.1. Definisi Himpunan Fuzzy

Pada dasarnya manusia mengenal obyek dengan memberi klasifikasi secara kualitatif seperti: besar, kecil, sedang, tinggi, rendah dan sebagainya. Sifat kebenaran yang dikandungnya samar karena batas kebenaran yang satu dengan yang lainnya tidak tegas. Kebenaran inilah yang disebut dengan kebenaran *fuzzy*.

Suatu himpunan fuzzy A dalam suatu semesta pembicaraan $U\{u\}$ didefinisikan oleh pasangan:

$$A : \{(u, \mu_A(u)/u \in A)\} \tag{2.1}$$

Dimana $(u) : u \rightarrow [0, 1]$ adalah derajat keanggotaan dari u .

Jika A adalah himpunan fuzzy diskrit terbatas maka,

$$A : \mu_A(u) u_1 + \dots + \mu_A(u_n) u_n \tag{2.2}$$

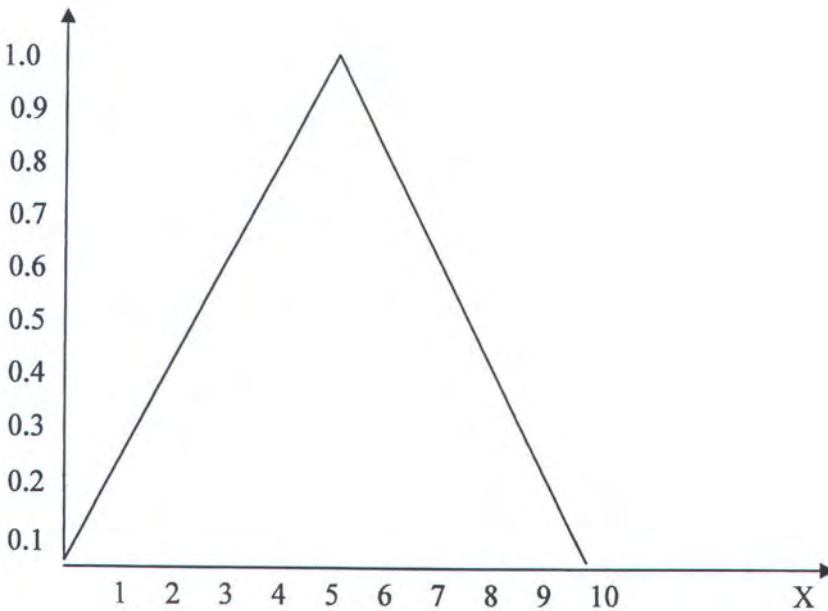
Dan jika u kontinyu maka himpunan fuzzy bisa dituliskan sebagai berikut

$$A : \int_x u_A(u)/x \quad (2.3)$$

Tanda garis miring diatas bukan menyatakan suatu pembagian melainkan bahwa nilai keanggotaan terurut relatif terhadap u yang bersangkutan. Sebagai contoh misalkan X adalah semesta pembicaraan bilangan cacah yang kurang dari 10, $X = \{0, 1, 2, 3, \dots, 10\}$, sedangkan himpunan fuzzy A didefinisikan sebagai bilangan yang dekat 5. Maka himpunan fuzzy A adalah:

$$A = 0/0 + 0.2/1 + 0.4/2 + 0.6/3 + 0.8/4 + 0.6/7 + 0.4/8 + 0.2/9 + 0/1$$

Penyajian secara grafis adalah:



Gambar 2.4 Penyajian secara grafis himpunan fuzzy A

2.2.2 Fungsi Keanggotaan (Membership Function)

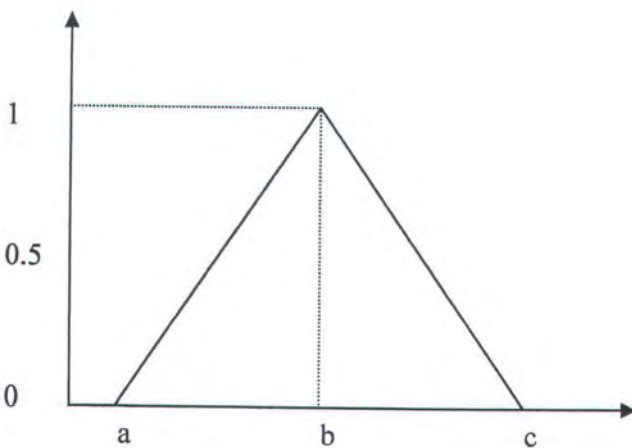
Sebuah fungsi keanggotaan pada suatu himpunan fuzzy A dinoyasikan dengan $\mu_A(x)$ yang mengkarakteristikkan suatu derajat keanggotaan x pada himpunan fuzzy A dalam semesta pembicaraan X.

Fungsi keanggotaan biasanya diekspresikan dalam grafik fungsi keanggotaan. Grafik fungsi keanggotaan yang sering digunakan dalam mendefinisikan fungsi keanggotaan adalah : Triangular.

Fungsi keanggotaan triangular dispesifikasikan dengan 3 parameter (a,b,c) yaitu:

$$\mu_A(x) = \begin{cases} 0 & x < a \\ (x - a) / (b - a) & a \leq x \leq b \\ (c - x) / (c - b) & b \leq x \leq c \\ 0 & x > c \end{cases}$$

Bentuk grafik triangular ditunjukkan pada gambar 2.2.



Gambar 2.5 Grafik Fungsi Triangular

2.2.3. Variabel Linguistik

Lotfi Zadeh menekankan, bahwa pendekatan teknik konvensional dalam analisa system pada dasarnya tidak cocok untuk menyelesaikan system humanistic. Karena pada system humanistic perilaku system sangat kuat dipengaruhi oleh keputusan, persepsi dan emosi manusia. Hal tersebut merupakan manifestasi dari apa yang disebut prinsip ketidakcocokan (Principle Of Incompatibility) yaitu jika kompleksitas dari system bertambah, kemampuan kita untuk membuat pernyataan – pernyataan yang teliti dan signifikan tentang perilaku system menjadi berkurang. Hal ini yang mendasari Lotfi Zadeh untuk memproporsionalkan konsep variable linguistik sebagai alternatif dalam pendekatan untuk memodelkan pikiran manusia. Seperti contoh dalam bahasa sehari-hari, kita lebih sering mengatakan “ Mobil itu melaju dengan kecepatan tinggi”, dari pada mengatakan “Mobil itu melaju dengan kecepatan 120 km/jam”.

Suatu himpunan fuzzy dapat didefinisikan dengan dasar variable linguistik tertentu yang dikarakteristikan sebagai:

$$(x, V(x), X, G, M); \quad (2.5)$$

dimana x = nama variable linguistik

$V(x)$ = himpunan term (term set) dari x yang merupakan linguistik value

X = semesta pembicaraan

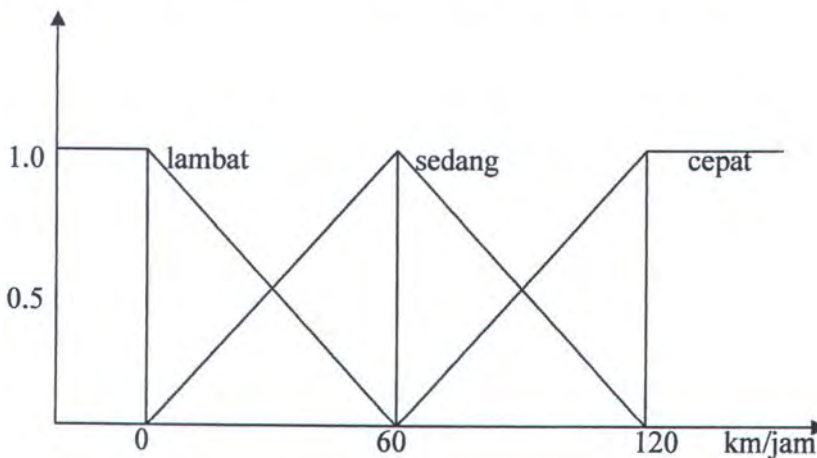
G = aturan syntactic untuk menghasilkan nama dalam $T(x)$

M = aturan semantic untuk menghubungkan tiap nilai dengan artinya

$M(A)$; dimana $M(A)$ menyatakan himpunan fuzzy A dalam X .

Suatu contoh: Suatu semesta pembicaraan kecepatan V (kecepatan), dinyatakan dengan variable linguistik *lambat*, *sedang*, *cepat*. Maka perangkat istilah $V(\text{kecepatan})$ dapat ditulis: $V(x) = (\text{lambat}, \text{sedang}, \text{cepat})$.

Selanjutnya dimisalkan variable linguistik berada pada semesta pembicaraan $0 - 120$ km/jam. Dalam semesta ini, *lambat* menyatakan kecepatan sekitar 0 km/jam, *sedang* menyatakan kecepatan sekitar 60 km/jam dan *cepat* menyatakan kecepatan sekitar 120 km/jam. Secara grafis digambarkan pada gambar 2.3.



Gambar 2.6 Definisi himpunan fuzzy dengan variable linguistik *lambat*, *sedang*, dan *cepat*.

Pada harga kecepatan 0 km/jam, derajat keanggotaan *lambat* adalah 1 sedangkan derajat keanggotaan untuk *sedang* dan *cepat* adalah 0. Pada kecepatan 60 km/jam derajat keanggotaan *sedang* adalah 1, sedang derajat keanggotaan *lambat* dan *cepat* adalah 0. Pada kecepatan 120 km/jam derajat keanggotaan *cepat* adalah 1, sedang derajat keanggotaan *lambat* dan *sedang* adalah 0. Misalkan pada kecepatan 90 km/jam, derajat keanggotaan *lambat* adalah 0, derajat keanggotaan *sedang* 50% dan *cepat* 50%.

2.2.4. Linguistic Hedge

Untuk mengembangkan himpunan yang lebih banyak dari satu variable linguistik dengan hanya menggunakan sedikit term, dipakai metode Linguistic Hedge, yaitu : metode operasi pengintensifan, pergeseran dan sebagainya.

Beberapa Hedge yang telah didefinisikan antara lain:

$$\text{Very, very } x = x^2 = \int \mu_x^2(x) / x \quad (2.6)$$

$$\text{More or Less } (x) = \int_x \mu_x^a(x) / x \quad (2.7)$$

$$\text{Rather, } \mu_{\text{rather}}(x) = \mu_A(x + c), \quad (2.8)$$

Dimana c = konstanta pergeseran.

2.2.5. Operasi Himpunan Fuzzy

Himpunan fuzzy memiliki basis atau aturan untuk memanipulasi secara sistematis konsep kekaburan dan ketidakpasian dengan menggunakan operasi himpunan fuzzy yang dilakukan dengan memanipulasi fungsi keanggotaannya.

Beberapa operasi dasar himpunan, untuk himpunan fuzzy A dan B pada semesta pembicaraan X antara lain:

Gabungan (Union) atau Disjunction, dapat ditulis:

$$\mu_{(A \cup B)}(x) = \max \{ \mu_A(x), \mu_B(x) \} = \mu_A(x) \cup \mu_B(x), \quad x \in X \quad (2.9)$$

Irisan (Intersection) atau Konjunction, dapat ditulis:

$$\mu_{(A \cap B)}(x) = \min \{ \mu_A(x), \mu_B(x) \} = \mu_A(x) \cap \mu_B(x), \quad x \in X \quad (2.10)$$

Komplemen atau negation

$$\mu_A(x) = 1 - \mu_A(x); \quad x \in X \quad (2.11)$$

Produk Aljabar (Aljabre Product)

Produk aljabar dari himpunan fuzzy A dan B, fungsi keanggotaan $\mu_A(x)$ dan $\mu_B(x)$ ditulis:

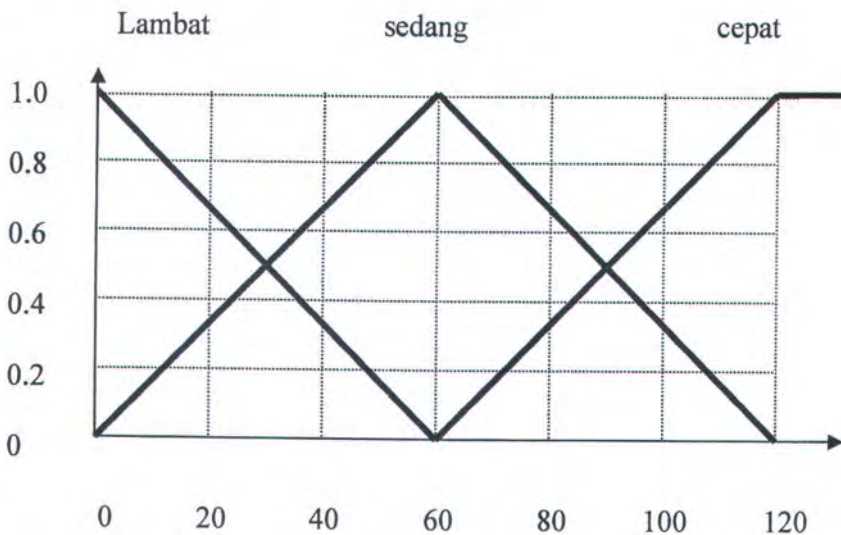
$$\mu_{(A*B)}(x) = \mu_A(x) * \mu_B(x), \quad x \in X \quad (2.12)$$

Sebagai contoh: misalnya X menyatakan suatu kecepatan antara 0 sampai 120 km/jam, atau ditulis $X = \{0, 20, 40, 60, 80, 100, 120\}$ dengan variable linguistik kecepatan adalah :*lambat*, *sedang*, *cepat* dan ketiga himpunan fuzzy tersebut didefinisikan sebagai:

$$A = \text{Lambat} = 1/0 + 0,66/20 + 0,33/40 + 0/60 + 0/80 + 0/100 + 0/120$$

$$B = \text{Sedang} = 0/0 + 0,33/20 + 0,66/40 + 1/60 + 0,66/80 + 0,33/100 + 0/120$$

$$C = \text{Cepat} = 0/0 + 0/20 + 0/40 + 0/60 + 0,33/80 + 0,66/100 + 1/120$$



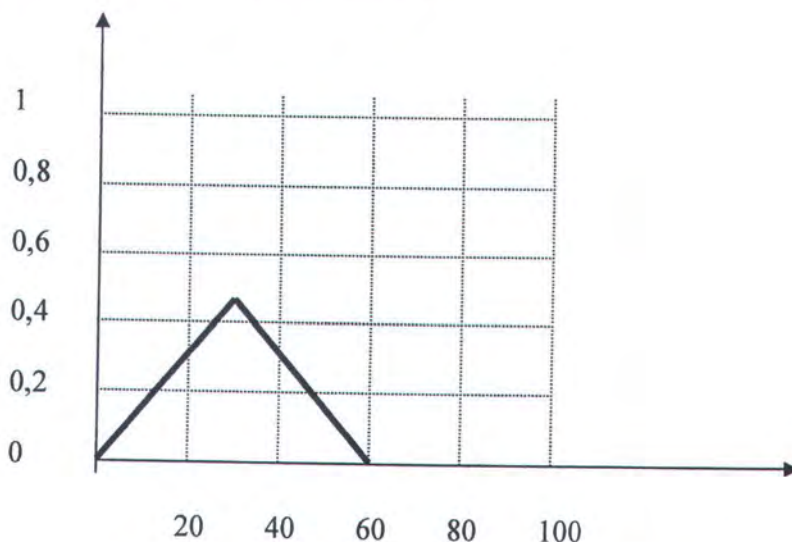
Gambar 2.7 Definisi himpunan fuzzy *lambat*, *sedang*, dan *cepat*.

Dari beberapa operasi himpunan fuzzy yang telah dikemukakan di atas maka dapat diketahui:

Irisan (Intersection) atau Conjunction antara lambat dan sedang adalah:

$$\begin{aligned} \mu_{(A \cap B)}(x) &= \min \{ \mu_A(x), \mu_B(x) \} = \mu_A(x) \cap \mu_B(x), \\ &= \min(0, 1)/0 + \min(0,33, 0,66)/20 + \min(0,66, 0,33)/40 + \min(1, 0)/60 + \\ &\quad \min(0,66, 0)/80 + \min(0,33, 0)/100 + \min(0, 0)/120 \\ &= 0/0 + 0,33/20 + 0,33/40 + 0/60 + 0/80 + 0/100 + 0/120. \end{aligned}$$

Secara grafis dapat dilihat pada gambar 2.5.



Gambar 2.8 Grafik irisan dari himpunan fuzzy lambat dan sedang.

Sedang produk aljabar dari lambat dan sedang adalah:

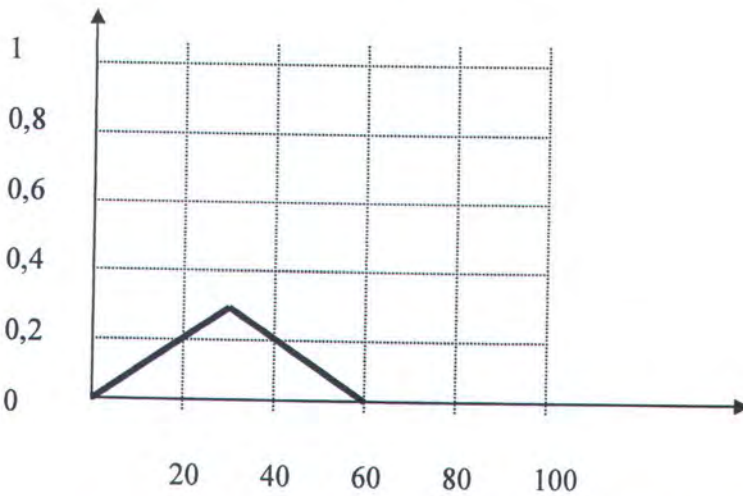
$$\begin{aligned} \mu_{(A * B)}(x) &= \mu_A(x) * \mu_B(x), \\ &= (0 \times 1)/0 + (0,33 \times 0,66)/20 + (0,66 \times 0,33)/40 + (1 \times 0)/60 + (0,66 \times 0)/80 + \\ &\quad (0,33 \times 0)/100 + (0 \times 0)/120. \\ &= 0/0 + 0,2178/20 + 0,2178/40 + 0/60 + 0/80 + 0/100 + 0/120 \end{aligned}$$

atau secara grafis terlihat pada gambar 2.6

Gabungan (Union) atau Disjunction antara sedang dan cepat.

$$\begin{aligned} \mu_{(A \cup B)}(x) &= \max \{ \mu_A(x), \mu_B(x) \} = \mu_A(x) \cup \mu_B(x), \\ &= \max(0, 0)/0 + \max(0,33, 0)/20 + \max(0,66, 0)/40 + \max(1, 0)/60 + \\ &\quad \max(0,66, 0,33)/80 + \max(0,5, 0,5)/90 + \max(0,33, 0,66)/100 + \\ &\quad \max(0, 1)/120. \\ &= 0/0 + 0,33/20 + 0,66/40 + 1/60 + 0,66/80 + 0,5/90 + 0,66/100 + 1/120. \end{aligned}$$

Secara grafis digambarkan pada gambar 2.7

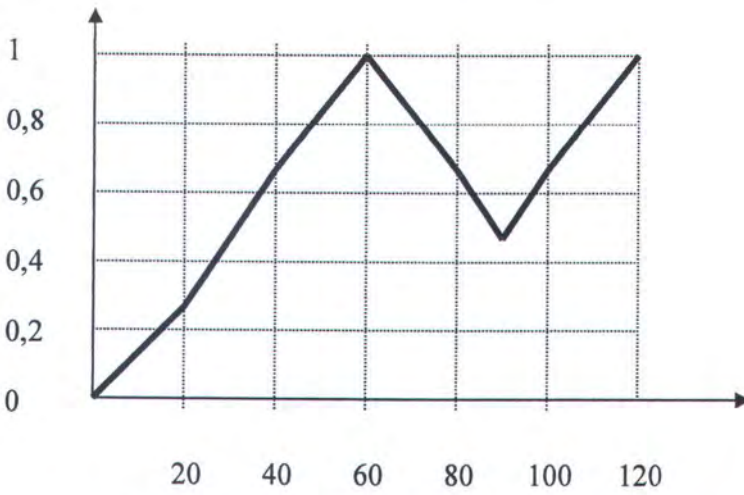


Gambar 2.9 Grafik produk aljabar dari himpunan fuzzy lambat dan sedang.

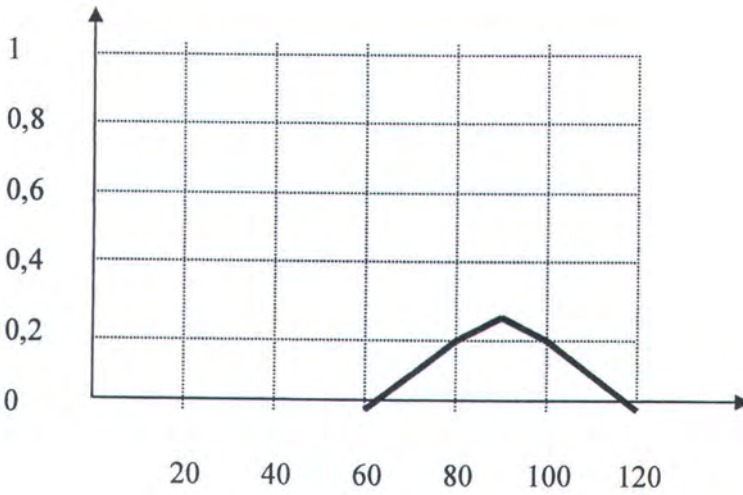
Produk Aljabar antara himpunan fuzzy sedang dan cepat.

$$\begin{aligned} \mu_{(A * B)}(x) &= \mu_A(x) * \mu_B(x), \\ &= (0 \times 0)/0 + (0,33 \times 0)/20 + (0,66 \times 0)/40 + (1 \times 0)/60 + (0,66 \times 0,33)/80 + \\ &\quad (0,5 \times 0,5)/90 + (0,33 \times 0,66)/100 + (0 \times 1)/120 \\ &= 0/0 + 0/20 + 0/40 + 0/60 + 0,2178/80 + 0,25/90 + 0,2178/100 + 0/120 \end{aligned}$$

Atau secara grafis seperti pada gambar 2.8



Gambar 2.10 Grafik Gabungan (Union) antara himpunan fuzzy sedang dan cepat.

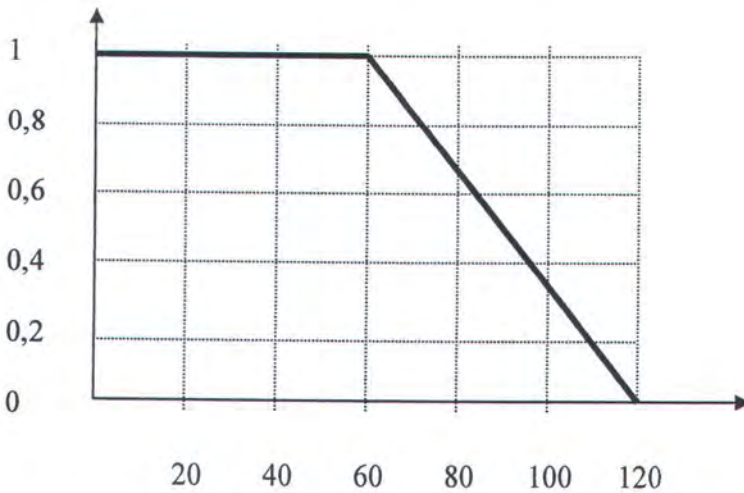


Gambar 2.11 Grafik produk aljabar antara himpunan fuzzy sedang dan cepat.

Komplemen (Negation) antara himpunan fuzzy cepat dan tidak cepat

$$\begin{aligned}
 \mu_A(x) &= 1 - \mu_A(x); \\
 &= (1 - 0)/0 + (1 - 0)/20 + (1 - 0)/40 + (1 - 0)/60 + (1 - 0,33)/80 + \\
 &\quad (1 - 0,5)/90 + (1 - 0,66)/100 + (1 - 1)/120 \\
 &= 1/0 + 1/20 + 1/40 + 1/60 + 0,66/80 + 0,5/90 + 0,33/100 + 0/120
 \end{aligned}$$

Secara grafis seperti pada gambar 2.9



Gambar 2.12 Grafik Komplemen antara himpunan fuzzy cepat dan tidak cepat.

2.2.6 Kontroller Logika Fuzzy

Beberapa sistem kontrol loop terbuka sering dijumpai dapat diubah menjadi sistem kontrol loop tertutup, jika operator manusia dipandang sebagai kontroler, membandingkan masukan dan keluaran, kemudian melakukan aksi koreksi yang berdasarkan selisih atau kesalahan yang diperoleh.

Jika kita berusaha menganalisa system kontrol loop tertutup yang melibatkan operator manusia semacam itu, kita akan menjumpai persoalan yang sulit dalam menuliskan persamaan matematis yang menggambarkan perilaku dan kemampuan nalar manusia. Belum lagi kebanyakan proses yang ada adalah sistem yang kompleks, non linier, dan mudah dipengaruhi oleh factor-faktor gangguan sekitar. Proses-proses tersebut pada kenyataannya bisa dikendalikan secara manual dengan hasil yang cukup baik. Operator manusia tersebut biasanya adalah tenaga terampil yang mengandalkan pengalaman praktis, tanpa dilatarbelakangi teori-teori pengendalian yang rumit. Hal ini disebabkan operator tersebut mengendalikan proses dengan dasar logika yang juga non linier dan kompleks, yang dibangun oleh

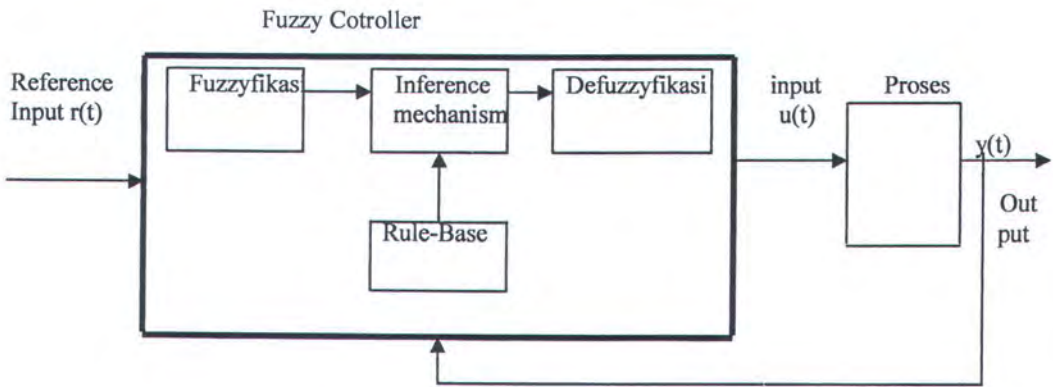
pengalaman dalam waktu yang panjang, sehingga pengendalian yang dilakukan sepenuhnya bersifat intuitif.

Untuk mendisain system pengendalian otomatis bagi proses-proses tersebut, diperlukan metode khusus, dalam hal ini digunakan teori logika *Fuzzy*.

Kontrol logika fuzzy merupakan system kontrol cerdas (*Intelligent Control System*) yang memanfaatkan / mengimplementasikan nalar/ logika manusia (*human operator*) sebagai pengendali proses dengan berdasarkan logika *fuzzy*.

2.2.7 Struktur Dasar Kontroller Logika Fuzzy

Salah satu hal penting yang harus difahami dalam menyelesaikan tesis ini adalah tentang bagaimana kontroler logika *fuzzy* itu. Kontroler logika *fuzzy* meliputi 4 elemen. Ke empat elemen itu digambarkan dalam gambar 2.11 dibawah ini.



Gambar 2.13. Kontroler Logika Fuzzy

A. Fuzzifikasi

Proses fuzzifikasi merupakan proses untuk mengubah variable nonfuzzy (variable numeric, crisp) menjadi variable fuzzy. Pada setiap sistem kontrol, besaran

input yang terjadi baik itu besaran set point maupun besaran input yang keluar dari sensor akan selalu berupa variable numeric, yang bersifat pasti dan kuantitatif. Atau dengan kata lain fuzzifikasi merupakan pemetaan titik numeric (crisp point)

$\underline{x} = (x^1, \dots, x^n)^T \in X$ ke himpunan fuzzy A di x , X adalah semesta pembicaraan.

Metode fuzzifikasi yang populer dipakai adalah *Fuzzifikasi Singleton*, A adalah fuzzy singleton dengan support \underline{x} . Artinya,

$$\mu_A(x) = 1 \text{ untuk } \underline{x} = x \text{ dan } \mu_A(x) = 0 \text{ untuk } \underline{x} = X \text{ yang lain dengan } \underline{x} \neq x. \quad (2.13)$$

B. Kuantisasi Dan Normalisasi

Pada proses kuantisasi terjadi penggolongan-penggolongan masukan menjadi daerah diskrit tertentu, yang berarti mendiskritkan masukan semesta pembicaraan yang biasanya kontinu kedalam sejumlah segmen-segmen tertentu yang disebut level kuantisasi dengan metode pelabelan atau penomoran level-level, sehingga membentuk pembicaraan baru yang bersifat diskrit.

Normalisasi memiliki fungsi untuk membatasi semesta pembicaraan pada jangkauan tertentu misalnya dari 1 sampai -1 , dengan kata lain normalisasi merupakan pemetaan semesta pembicaraan masukan ke semesta pembicaraan baru yang terbatas. Skala pembicaraan bisa seragam (uniform), bisa tidak seragam (ununiform).

C. Penyusunan Aturan / Rule Kontrol Fuzzy

Aturan / rule dasar fuzzy dapat ditulis dalam bentuk :

$$R^{(1)} : \text{Jika } x_1 \text{ adalah } F_1^1 \text{ DAN....DAN....}x_n \text{ adalah } F_n^1, \text{ maka } y \text{ adalah } G^1 \quad (2.14)$$

Dimana F_1^1 dan G^1 adalah himpunan fuzzy masing-masing di $X \subset R$ dan $Y \subset R$,

$(x = x^1, \dots, x^n)^T \in X_1, \dots, X_n$, dan $y \in Y$ adalah variable linguistik.

F_1^1, \dots, F_n^1 adalah premise atau antecedent (pernyataan yang mendasari penalaran) sedangkan G^1 adalah consequence (hasil penalaran). Karena permasalahan yang dihadapi oleh system kontrol cerdas kebanyakan kompleks, sehingga aturan yang terlibat relatif banyak, maka relasi implikasi fuzzy dapat dibentuk dengan mengkombinasikan relasi-relasi fuzzy dengan menghubungkannya dengan kata penghubung. Kata hubung yang sering dipakai adalah AND, OR, dan ELSE.

Operasi yang dipakai untuk implikasi “jika x adalah A Maka y adalah B”.

adalah : Operasi Implikasi Min [Mamdani], yang dapat ditulis :

$$R_C [\mu_A(x), \mu_B(y)] = \mu_A(x) \cap \mu_B(y) \quad (2.15)$$

D. Unit Penalaran Fuzzy

Salah satu unit penalaran *fuzzy* adalah Generalized Modus Ponens (GMP).

Jika 2 unit himpunan fuzzy dimotasikan dengan A, A', B, B', variable linguistik dinotasikan dengan x dan implikasi fuzzy dinotasikan dengan R(x,y), maka GMP dapat didefinisikan:

Pernyataan 1 (aturan) : jika x adalah A, maka y adalah B

Pernyataan 2 (fakta) : jika x adalah A'

Kesimpulan : y adalah B'

Secara analitis, GMP diatas dibentuk dengan komposisi A' dengan implikasi R(x,y) yang menghasilkan keputusan atau dapat ditulis:

$$B' = A' \cdot R(x,y) \quad (2.16)$$

Keterangan A' dan B' adalah himpunan fuzzy dengan linguistik hedge : sangat, lebih/kurang dari, tidak dan lain-lain.

Bentuk GMP diatas lalu disesuaikan dengan struktur kontrol, menjadi aturan kontrol fuzzy berbentuk multi input, single output (MISO), bentuk ini diperoleh dari skema umum pendekatan penalaran multikondisi, yaitu:

Jika x adalah A dan y adalah B **Maka** z adalah C

Dimana pada kontroller logika Fuzzy x dan y adalah masukan data crisp, z adalah aksi kontrol dan A,B,C adalah diskripsi linguistik dari x,y,z.

Berdasarkan metode implikasi min-max Mamdani, digunakan metode inferensi max-min Mamdani yang dinyatakan sebagai berikut :

$$\mu_c^k(z) = \max_k \{ \min[\mu_{Ai}(x), \mu_{Bi}(y)] \}$$

$$\mu_c^k(z) = \max_k (\alpha_k) \quad k = 1, 2, 3, \dots, r$$

dimana r, I adalah jumlah aturan dan variable linguistik yang dipakai, dan α_k adalah factor pembobot (fire strength) yang menunjukkan nilai kontribusi aturan ke r pada aksi kontrol fuzzy yang secara grafis digambarkan seperti pada gambar 2.12.

E. Defuzzifikasi

Metode defuzzifikasi merupakan metode untuk mengubah kembali variable fuzzy yang dihasilkan dari proses penalaran (fuzzy reason) menjadi suatu variable numeric non fuzzy kembali.

Metode defuzzifikasi yang populer dipakai adalah: Centroid Of Area (COA). Dalam metode COA, besarnya variable numeric crisp u^* didapat dari pusat geometric dari besarnya output fuzzy $\mu_{\text{output}}(x)$, secara rumus dapat ditulis:

$$u^* = \frac{\sum_{i=1}^N u_i \mu_{out}(u_i)}{\sum_{i=1}^N \mu_{out}(u_i)} \quad (2.17)$$

dimana:

u^* = nilai keluaran

n = tingkat kuantisasi

u_i = elemen ke-1

$\mu_{out}(u_i)$ = derajat keanggotaan elemen-elemen pada fuzzy set

U = semesta pembicaraan

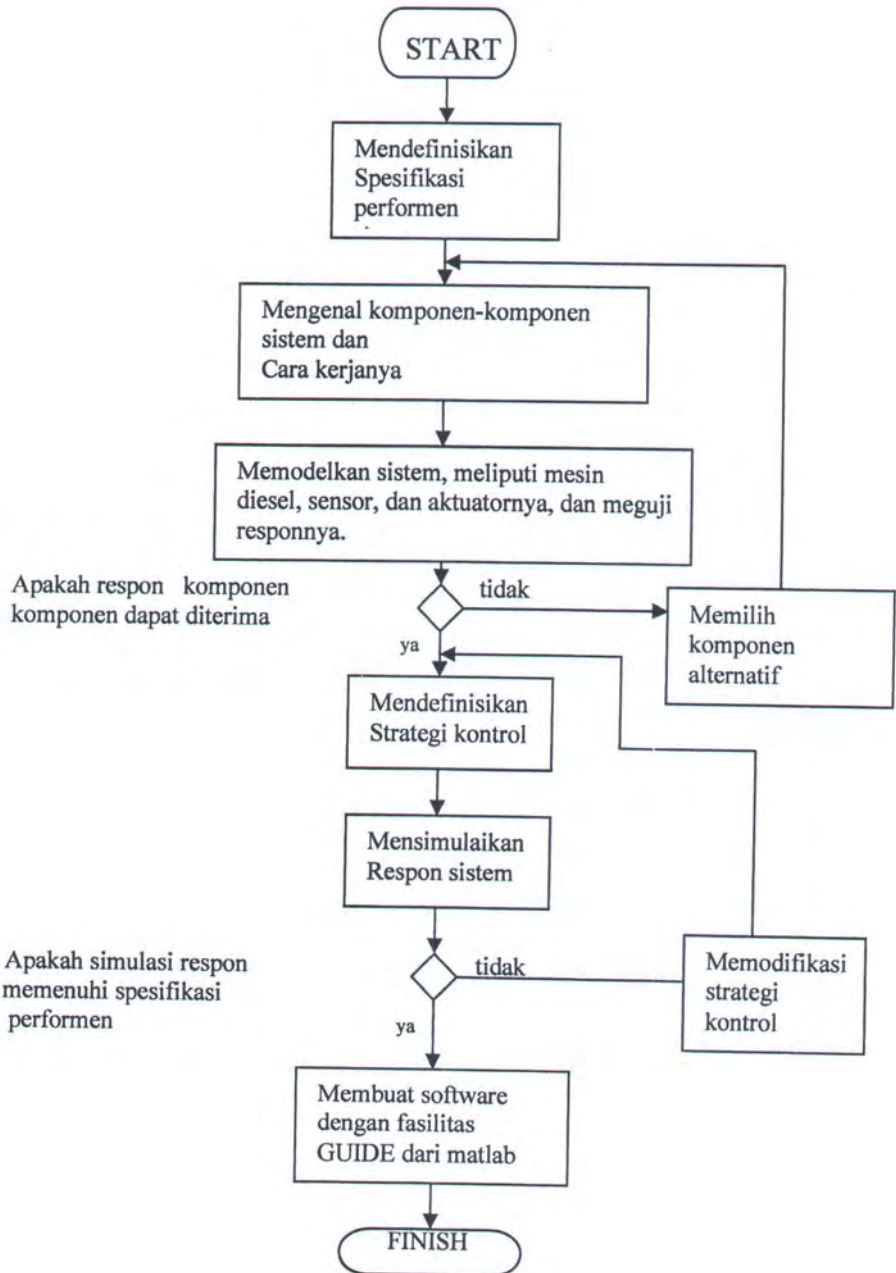


BAB III
METODOLOGI

BAB III

METODOLOGI

3.1 Diagram Alir (Flow Chart)



3.2 Penjelasan Diagram Alir

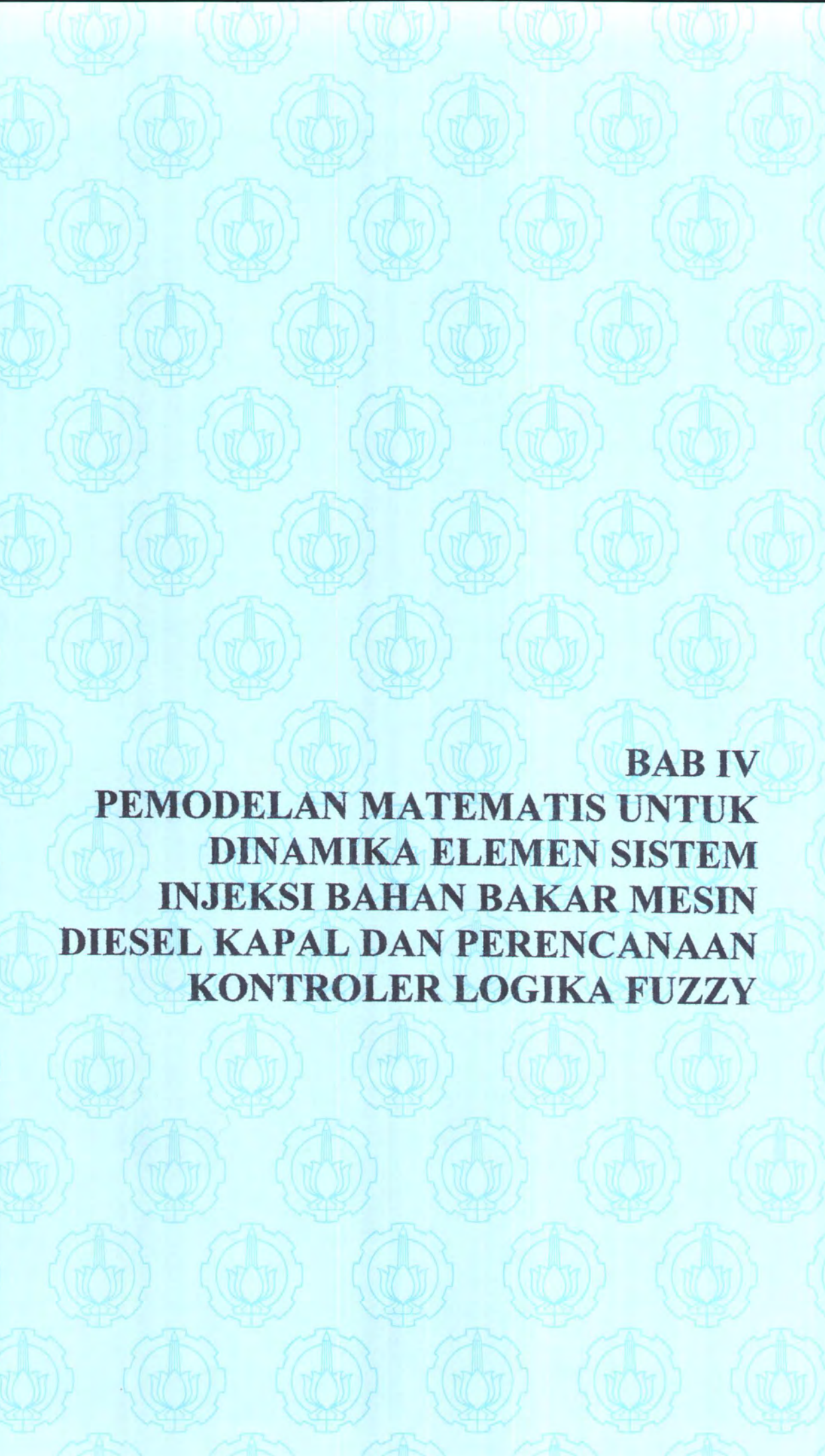
Secara umum metodologi penyelesaian masalah dalam tesis ini dapat diuraikan sebagai berikut:

1. Mendapatkan informasi dari beberapa sumber yaitu buku-buku yang berkenaan dengan pengendalian system bahan bakar mesin diesel kapal dan buku-buku tentang kontroler *fuzzy*. Dari sini akan diperoleh secara pasti komponen-komponen yang digunakan dalam sistem injeksi bahan bakar mesin diesel baik yang mekanik maupun yang elektrik, bagaimana cara kerjanya, sehingga bisa didapatkan model dinamis yang benar dari sistem injeksi bahan bakar. Sistem injeksi yang dibahas dalam tesis ini sistem injeksi yang umum digunakan, baik oleh Caterpillar, Detroit maupun lainnya.
2. Disamping itu makalah-makalah atau paper –paper yang berhubungan dengan pengendalian system bahan bakar juga diperlukan sebagai referensi dan memperluas wawasan terutama dalam memodelkan system dan menganalisanya.
3. Mengumpulkan data-data penunjang yang diperlukan dari berbagai sumber yang ada, diantaranya dari TRAKINDO yang memproduksi Caterpillar Marine Engine yang sudah menggunakan EUI (*electronic unit injector*). Data ini berkenaan dengan spesifikasi komponen-komponen yang digunakan dalam sistem injeksi bahan bakar secara elektronik (EUI).
4. Mendefinisikan spesifikasi performen sistem yang akan dikendalikan, dalam hal ini sistem injeksi bahan bakar EUI.
5. Mengenali komponen-komponen sistem injeksi bahan bakar EUI. Bagaimana prinsip kerjanya sehingga bisa didapatkan pemodelan matematis yang benar

dari komponen-komponen tersebut. Termasuk sensor-sensor , dan aktuator yang digunakan.

6. Membuat model matematis dari system. Pemodelan ini meliputi pemodelan dari plant (bagian yang dikendalikan), dan pemodelan actuator juga sensor yang akan digunakan. Setelah model sistem dibuat, sebelum merencanakan kontrolernya, model sistem diuji terlebih dahulu apakah apakah model dapat diterima. Jika belum, maka akan dipilih komponen-komponen alternatif lain. Dan jika sudah dapat diterima, maka bisa dilanjutkan tahapan berikutnya, yaitu mendefinisikan strategi kontrolnya.
7. Kontroler yang digunakan adalah kontroler logika *fuzzy*. Dalam hal ini digunakan fasilitas simulink yang ada dalam software matlab. Dalam hal ini ada beberapa tahapan yaitu:
 - Mendefinisikan dan menentukan jumlah variabel linguistik yang akan digunakan dalam aksi atur ini.
 - Membuat tabel pandang (look up table) untuk menentukan kaidah-kaidah atur pengaturan logika *fuzzy* sesuai dengan perintah variabel linguistik yang ditentukan.
8. Mensimulasikan respon sistem yang telah dibuat. Jika simulasi respon sistem belum sesuai dengan spesifikasi performen, maka perlu merubah strategi kontrolnya. Dilakukan perbaikan-perbaikan sampai didapat hasil respon yang sesuai dengan spesifikasi performen sistem. Jika sudah sesuai, maka bisa dilanjutkan ke tahapan berikutnya.
9. Tahapan terakhir dalam penyelesaian permasalahan tesis ini adalah membuat software dengan menggunakan fasilitas GUI dalam matlab untuk

memudahkan dalam mensimulasikan respon sistem pengendalian yang telah dibuat.



BAB IV
PEMODELAN MATEMATIS UNTUK
DINAMIKA ELEMEN SISTEM
INJEKSI BAHAN BAKAR MESIN
DIESEL KAPAL DAN PERENCANAAN
KONTROLER LOGIKA FUZZY

BAB IV

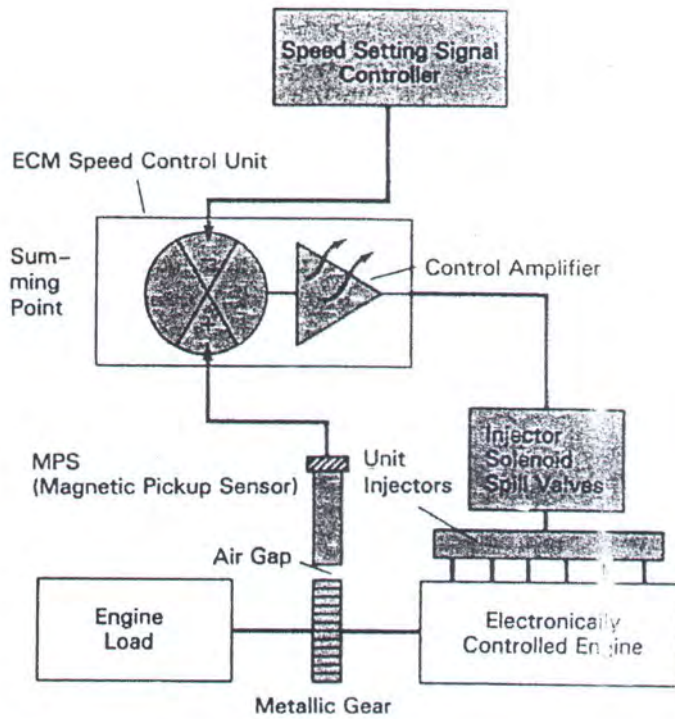
PEMODELAN MATEMATIS UNTUK DINAMIKA ELEMEN SISTEM INJEKSI BAHAN BAKAR MESIN DIESEL KAPAL DAN PERENCANAAN KONTROLER LOGIKA FUZZY

4.1 EUI (Electronic Unit Injektor) Sistem bahan bakar

Seperti telah diterangkan di bab II, tidak ada *rack* pada sistem bahan bakar yang dikendalikan secara elektronik. Ukuran jumlah bahan bakar yang diinjeksikan ke dalam ruang bakar, dan waktu awal penginjeksian semuanya dikendalikan secara elektronik. Pada gambar 2.1 a dan b (pada bab II) dapat dilihat bahwa dalam mengendalikan sistem injeksi bahan bakar ke dalam silinder atau ruang bakar, ECM menerima beberapa sinyal dari beberapa sensor yang digunakan, antara lain sensor posisi pedal (throttlet), *speed / timing sensor*, temperatur bahan bakar, *boost* mesin turbo, temperatur saluran masuk udara, temperatur oli dan lain-lain. Sesuai dengan batasan dibab 1, maka dalam tesis ini ditekankan bahwa yang dikendalikan utamanya adalah jumlah bahan bakar yang diinjeksikan. Masukan pada kontroler adalah kecepatan putaran mesin. Sehingga secara lebih sederhana bisa digambarkan sistem pengendalian sistem injeksi bahan bakar seperti gambar 4.1 dibawah ini.

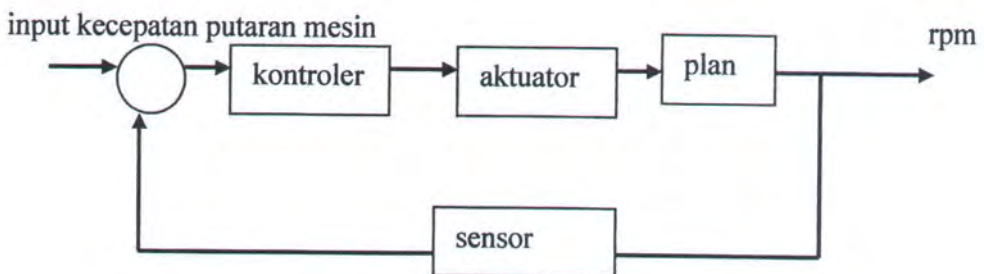
Sensor kecepatan putara mesin yang digunakan adalah *magnetic pickup sensor (MPS)*. MPS adalah sebuah perangkat magnet permanen kutub tunggal. Konsep *magnetic pickup* ini digunakan pada semua sistem elektronik yang ada, maka operasinya bisa dianggap sama untuk semuanya. MPS dipasang disebelah suatu gear transmisi dan dibuat dari bahan yang bereaksi terhadap medan magnet. Ketika gear memotong medan magnet MPS ini, maka akan dihasilkan sinyal arus bolak-

balik yang disesuaikan dengan rpm mesin. Sinyal ini yang nantinya dikirim ke kontroler.



Gambar 4.1 Gambar penyederhanaan sistem injeksi bahan bakar yang dikendalikan secara elektronik.

Sistem injeksi bahan bakar yang dikendalikan secara elektronik seperti pada gambar 4.1, bisa digambarkan dalam diagram blok dibawah ini.



Gambar 4.2 Gambar diagram blok sistem injeksi bahan bakar yang dikendalikan secara elektronik



4.2 Dinamika Mesin Diesel

Dari literature yaitu Munif [1993], Fossen [1994], Heru [1999], diperoleh perumusan tentang dinamika mesin diesel, dinamika kecepatan mesin diesel N_o merupakan fungsi dari bahan bakar yang disuplay ke silinder mesin diesel Q dan torsi mesin diesel T , jadi:

$$N_o = f(Q, T) \quad (4.1)$$

Linierisasi dari persamaan 4.1 adalah

$$N_o = C_1 q - C_2 t \quad (4.2)$$

Dimana :

$$C_1 = \left. \frac{\partial N_o}{\partial Q} \right|_T$$

$$C_2 = \left. \frac{\partial N_o}{\partial T} \right|_Q$$

Dimana:

C_1 = Gradien kurva perubahan kecepatan pada setiap perubahan aliran bahan bakar
Dengan torsi dijaga konstan

C_2 = Gradien kurva perubahan kecepatan pada setiap perubahan torsi dengan aliran
bahan bakar dijaga konstan.

Untuk T yang konstan maka kecepatan N_o bertambah sehingga harga C_1 berharga positif. Sedangkan untuk Q yang konstan maka N_o akan berkurang apabila torsi bertambah sehingga C_2 akan bernilai negatif.

Torsi T yang dihasilkan oleh mesin diesel dikurangi torsi beban T_L adalah torsi bersih yang dipergunakan oleh mesin diesel untuk menghasilkan percepatan. Perumusan matematikanya yaitu:

$$T - T_L = J\alpha = J \cdot \frac{dn_o}{dt} \quad (4.3)$$

Dimana

T_L = torsi beban

J = momen inersia rotasi mesin diesel

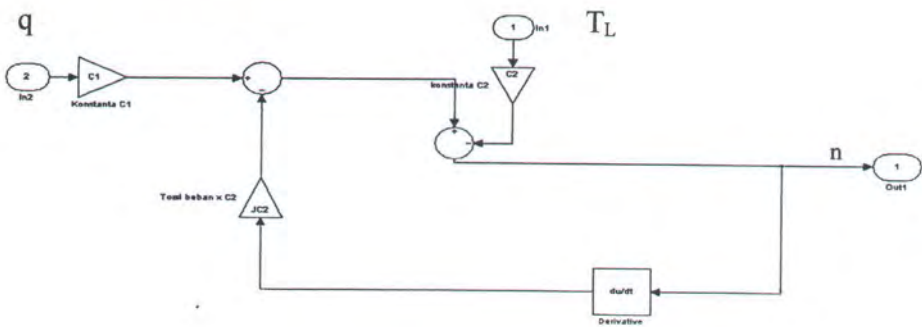
α = percepatan sudut mesin diesel

Dari persamaan 4.2 da. 4.3 dapat dirumuskan:

$$T_L + J \frac{dN}{dt} = \frac{C_1 Q - N_o}{C_2}$$

$$N_o = C_1 Q - T_L C_2 - J C_2 \frac{dN}{dt} \quad (4.4)$$

Dari persamaan 4.4 diatas dapat digambarkan diagram blok untuk mesin diesel seperti gambar dibawah ini.



Gambar 4.3 Gambar diagram blok state-space plant (mesin diesel kapal)

Dimana gain 2 (K) adalah JC_2

Karena keluaran dari sistem diatas masih dalam radian per detik maka perlu dikalikan lagi dengan $60/2\pi$, untuk menjadi putaran per detik.

Untuk mendapatkan model dinamika dari plant (mesin diesel kapal) dalam bentuk *transfer function* , maka persamaan 4.2 dan 4.3 diubah dalam bentuk laplace sebagai berikut:

$$T(s) = \frac{C_1q(s) - n_o(s)}{C_2} \quad (45)$$

dan

$$T - T_L = Jn_o(s)s \quad (46)$$

Substitusi persamaan 45 kedalam persamaan 46 menghasilkan;

$$\frac{C_1q(s) - n_o(s)}{C_2} = Jn_o(s)s + T_L$$

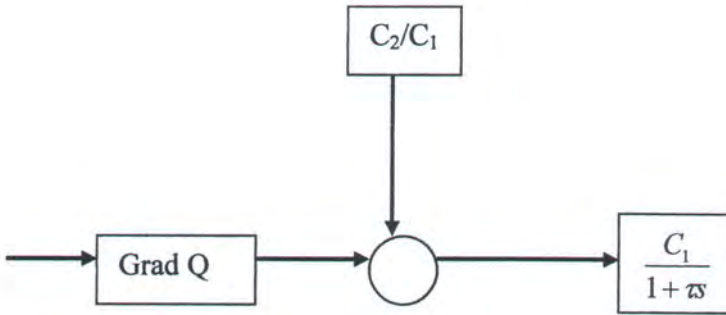
$$n_o(s) = \frac{C_1q(s) - C_2T_L}{1 + JC_2s}$$

$$n_o(s) = \frac{C_1}{1 + \tau_1s} \left(q - \frac{C_2}{C_1} T_L \right) \quad (47)$$

dimana

$$\tau = C_2J$$

Berdasarkan persamaan 47 diatas digambarkan sebuah diagram blok seperti pada gambar 4.3 yang menyatakan dinamika motor diesel dengan jumlah bahan bakar sebagai inputnya.



Gambar 4.4 Blok diagram dinamika mesin diesel kapal dalam bentuk *transfer function*.

4.3 Dinamika Transducer Kecepatan

Merupakan alat untuk mengubah variable system menjadi sinyal yang ditransmisikan ke rangkaian kontroler. Dalam hal ini variable terkontrol adalah kecepatan putaran motor. Dalam system pengendalian ini digunakan tachometer sebagai sensor kecepatan yang membangkitkan tegangan sebanding dengan kecepatan putar poros. Tegangan yang terbangkit didalam peralatan ini adalah tegangan induksi yang dirumuskan dengan persamaan:

$$E_a = cn\phi$$

Dimana

ϕ = fluks

n = putaran

c = konstanta yang harganya bergantung pada jumlah kutub, jalur parallel konduktor jangkar dan jumlah konduktor jangkar. Atau secara rumus

$$c = (p/a) \times (z/60)$$

p = jumlah kutub

a = jalur parallel konduktor jangkar

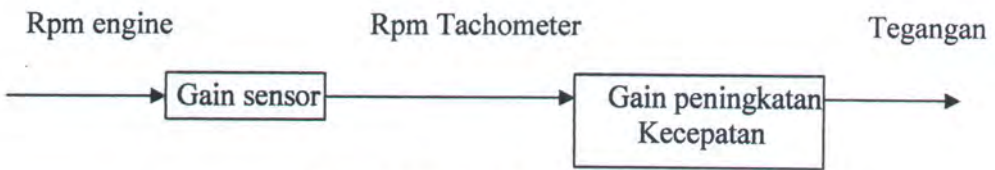
z = jumlah konduktor jangkar

Jika input tachometer adalah rpm mesin n_o dan output nya adalah tegangan (e) maka transfer function dari sensor adalah:

$$\frac{E(s)}{n_o(s)} = K_g \tag{4.8}$$

dimana K_g adalah konstanta proporsional.

Diagram blok untuk sensor dengan masukan berupa rpm dan keluaran adalah tegangan ditunjukkan pada gambar 4.3 dibawah.



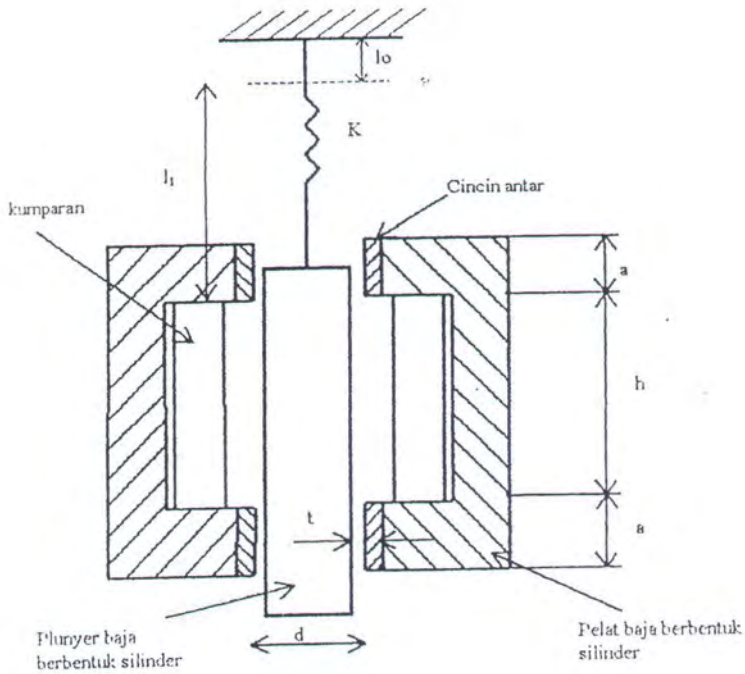
Gambar 4.5 Diagram blok untuk sensor

Dimana gain sensor didapatkan dari spesifikasi tachometer yang digunakan, dan gain peningkatan kecepatan didapat perbandingan antara diameter cranckshaft dan diameter tachometer.

4.4 Dinamika Aktuator.

Aktuator dalam sistem pengendalian ini terdiri adalah solenoid yang akan mengubah besaran listrik menjadi gerak linier, yaitu berupa gerak solenoid valve yang membuka dan menutup spill valve, sehingga penginjeksian dapat diatur.

Besar gaya yang dihasilkan oleh solenoid adalah proporsional dengan arus [Fitzgerald, 1990]. Gambar 4.4 memperlihatkan penampang sebuah magnet solenoid berbentuk silinder dimana plunyer berbentuk silinder bermassa M bergerak vertical dalam cincin- antar (guide rings) dari kuningan (brass) yang tebalnya t dan diameter rata-ratanya d .



Gambar 4.6 Magnet solenoida.

Permiabilitas kuningan mendekati harga permiabilitas ruang hampa [Fitzgerald, 1990], yaitu $4 \pi \times 10^{-7}$ H/m.

Plunyer tersebut disangga oleh pegas yang tetapannya K . Tegangan terminalnya adalah e dan arus yang melalui kumparan yang mempunyai lilitan N adalah i .

Penurunan secara detail gaya magnetik yang bekerja ke atas pada plunyer terdapat pada [Fitzgerald, 1990] sedangkan dibawah ini hasil akhir dari penurunan rumus tersebut.

$$f_{\pi d} = \frac{1}{2} L' \frac{ai^2}{(a+x)^2} \quad (4.9)$$

$$\text{dengan } L' = \frac{\mu_o \pi ad N^2}{t}$$

Sedangkan tegangan terminal pada kumparan sama dengan drop tegangan pada kumparan ditambah dengan tegangan yang diinduksikan dalam kumparan [Fritzgerald, 1990] sebagaimana dirumuskan dibawah ini.

$$e = Ri + L' \frac{x}{(a+x)} \frac{di}{dt} + L' \frac{ai}{(a+x)^2} \frac{dx}{dt} \quad (4.10)$$

Dua persamaan terakhir ini merupakan persamaan non linier. Untuk mendapatkan karakteristik yang linier maka simpangan plunyer harus dibatasi pada simpangan yang kecil disekitar titik keseimbangan. Teknik linierisasi secara terinci untuk magnet ssolenoida terdapat pada [fritzgerald, 1990] dan [Janusz Zieba]. Hasil akhir dari linierisasi tersebut adalah:

$$f_{\pi d} = K_o I$$

$$e = Ri + L_o \frac{di}{dt} + K_o \frac{dx}{dt} \quad (4.10)$$

dimana

$$K_o = \frac{L' a I_o}{(a + X_o)^2}$$

$$L_o = \frac{L' X_o}{a + X_o}$$

I_o = arus listrik pada titik keseimbangan

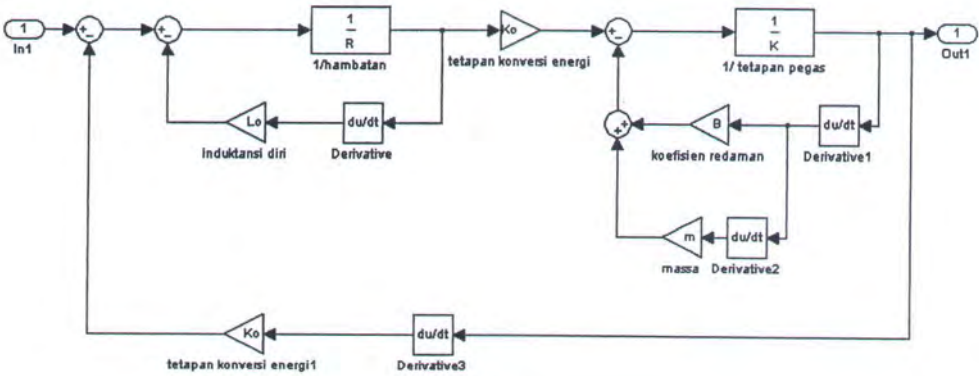
X_o = posisi plunyer pada titik keseimbangan

Dari perumusan gaya magnetik dan persamaan tegangan terminal pada kumparan maka dapat dibuat persamaan dinamika untuk gerak magnet solenoida sebagai berikut:

$$f_{\pi d} = M \frac{d^2 x}{dt^2} + B \frac{dx}{dt} + Kx \quad (4.11)$$

$$e = Ri + L_o \frac{di}{dt} + K_o \frac{dx}{dt} \quad (4.12)$$

Dari persamaan diatas, dapat digambarkan diagram blok dari sub sistem yang berfungsi sebagai aktuator yaitu solenoid, seperti gambar dibawah ini.



Gambar 4.7 Gambar diagram blok state-space dari subsistem solenoid

Untuk mendapatkan diagram blok solenoid dalam bentuk *transfer function*, maka persamaan 4.11 dan 4.12 dapat diubah dalam bentuk laplace seperti dibawah ini.

$$\begin{aligned} K_o I(s) &= Ms^2 X(s) + BsX(s) + KX(s) \\ &= (Ms^2 + Bs + K) X(s) \end{aligned}$$

dan

$$\begin{aligned} E(s) &= RI(s) + L_o sI(s) + K_o sX(s) \\ &= (R + L_o s) I(s) + K_o sX(s) \end{aligned}$$

Dengan mengatur dua persamaan diatas didapat:

$$\frac{X(s)}{I(s)} = \frac{K_o}{Ms^2 + Bs + K} \quad (4.13)$$

$$\text{dan } I(s) = \frac{E(s) - K_o s X(s)}{R + L_o s} \quad (4.14)$$

Substitusi persamaan 4.13 dan persamaan 4.14 akan menghasilkan

$$\frac{X(s)}{\frac{E(s) - K_o s X(s)}{R + L_o s}} = \frac{K_o}{Ms^2 + Bs + K}$$

$$X(s) = \frac{K_o (E(s) - K_o s X(s))}{(Ms^2 + Bs + K)(R + L_o s)}$$

$$X(s) + \frac{K_o s X(s)}{(Ms^2 + Bs + K)(R + L_o s)} = \frac{K_o E(s)}{(Ms^2 + Bs + K)(R + L_o s)}$$

$$\{(Ms^2 + Bs + K)(R + L_o) + K_o s\} X(s) = K_o E(s)$$

Sehingga apabila untuk magnet solenoida yang bertindak sebagai input sistem adalah tegangan terminal (E) dan sebagai outputnya adalah displasemen plunyer (gerakkan valve), maka *transfer function* untuk solenoida adalah:

$$\frac{X(s)}{E(s)} = \frac{K_o}{(Ms^2 + Bs + K)(R + L_o s) + K_o s^2} \quad (4.15)$$

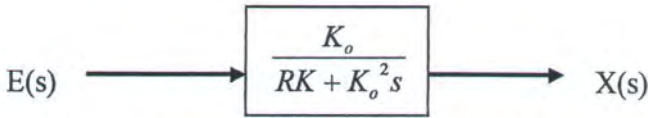
Pada operasi normal $L_o s \ll R$ dan $Ms^2 + Bs \ll K$, asumsi ini diperkuat pula pada [Fritzgeral, 1990] halaman 105 dan [Yeaple, 1990] halaman 243 yang menyebutkan bahwa B dapat diabaikan, maka dinamika magnet solenoida dapat didekati dengan sistem orde satu sebagai berikut:

$$\begin{aligned} \frac{X(s)}{E(s)} &= \frac{K_o}{RK + K_o s^2} \\ &= \frac{K_v}{1 + \tau s} \end{aligned} \quad (4.16)$$

dengan $K_v = K_o / RK$

$$\tau = K_o^2 / RK$$

Sehingga diagram blok untuk solenoid dalam bentuk *transfer function* dapat digambarkan sebagai berikut:



Gambar 4.8 Diagram blok solenoida dalam *transfer function*

4.5 Data Dan Pemodelan Sistem

Pada tesis ini diperlukan data-data dari mesin diesel, solenoida sebagai aktuatornya juga tachometer sebagai sensor. Data data tersebut sebagian didapat dari penelitian sebelumnya [Heru, 1999], dan untuk data yang spesifik sekali dan dari bagian disain, maka data tersebut hasil perkiraan peneliti.

Data-data tersebut adalah:

A. Mesin Diesel

1. Gradien kurva perubahan kecepatan pada setiap perubahan aliran bahan bakar dengan torsi dijaga konstan (C_1) adalah 2.45
2. Gradien kurva perubahan kecepatan pada setiap perubahan torsi dengan aliran bahan bakar dijaga konstan (C_2) adalah 0.0305
3. Gradien kurva laju aliran bahan bakar terhadap lamanya valve solenoida menutup adalah 10 kg/bkW hr² m
4. Inersia rotasi mesin adalah 983.6 kg m².
5. Diameter crankshaft : 46 cm

B. Solenoida

1. Jumlah lilitan : 737 lilitan

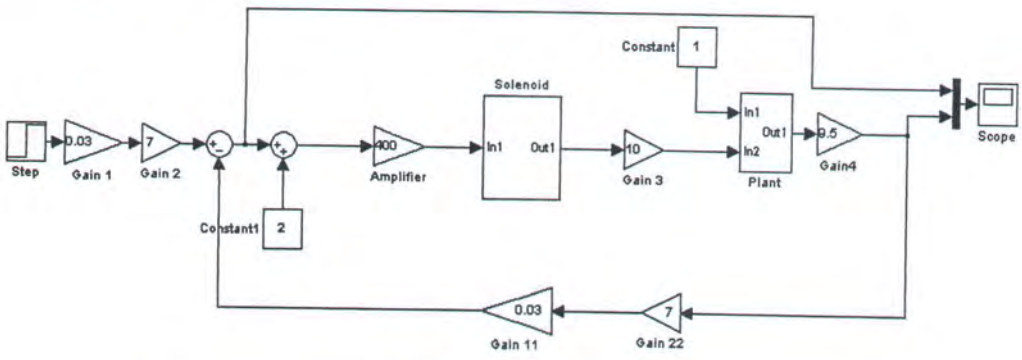
2. Tahanan total lilitan : 95.3 ohm
3. Diameter luar kumparan : 3.15 cm
4. Diameter dalam lilitan : 1.11 cm
5. Tinggi kumparan : 1.9 cm
6. Tegangan pada titik keseimbangan : 38 volt
7. Konstanta pegas : 41.9 N/m²
8. permeabilitas kuningan mendekati permeabilitas ruang hampa : $4\pi \times 10^{-7}$
9. tetapan redaman : 0.05 N dt/m
10. Massa : 0.01 kg
10. Simpangan saat plunyer keadaan diam : 0.25 cm

C. Transduser

1. Gain voltage : 30 volt per 1000 rpm
2. rpm maksimum : 7500 rpm
3. Diameter tachogenerator : 7 cm

Dari data-data diatas maka dapat digambarkan sistem injeksi bahan bakar mesin diesel kapal secara lengkap seperti gambar-gambar dibawah ini.

4.5.1 Diagram blok sistem injeksi bahan bakar dalam state-space



Gambar 4.9 Diagram blok sistem injeksi bahan bakar dalam bentuk state-space.

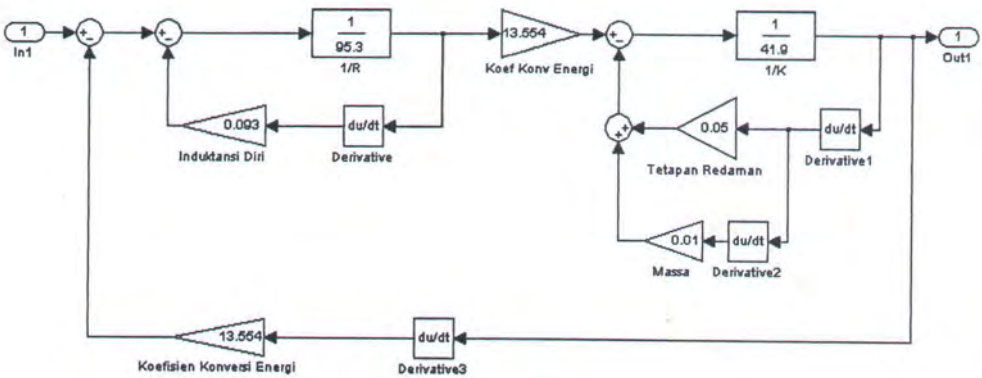
Gain 1 = gain 11 adalah gain tachogenetaror

Gain 2 = gain 22 adalah gain peningkatan kecepatan tachogenerator dari perbandingan antara diameter crankshaft dan diameter tachogenerator.

Gain 3 adalah garadien laju aliran bahan bakar terhadap lamanya solenoida diberi sinyal.

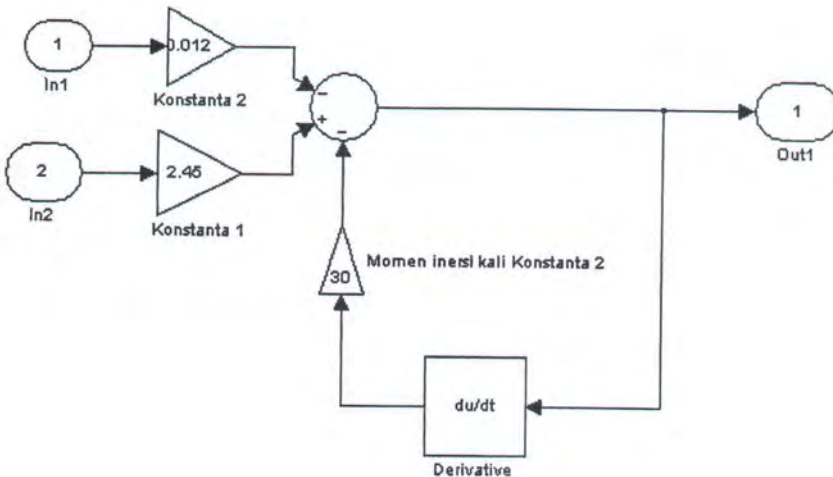
Gain 4 adalah konversi satuan dari radian perdetik ke rpm, sebesar $60/2\pi$.

4.5.1.1 Substistem aktuator (Solenoida)



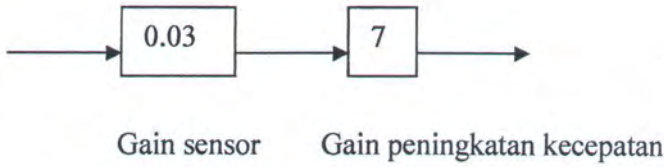
Gambar 4.10 Diagram blok solenoida

4.5.1.2 Substistem plan (Mesin Diesel)



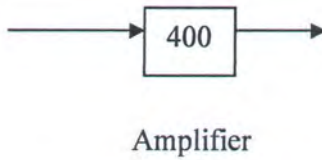
Gambar 4.11 Diagram blok mesin diesel.

4.5.1.3 Subsisten sensor



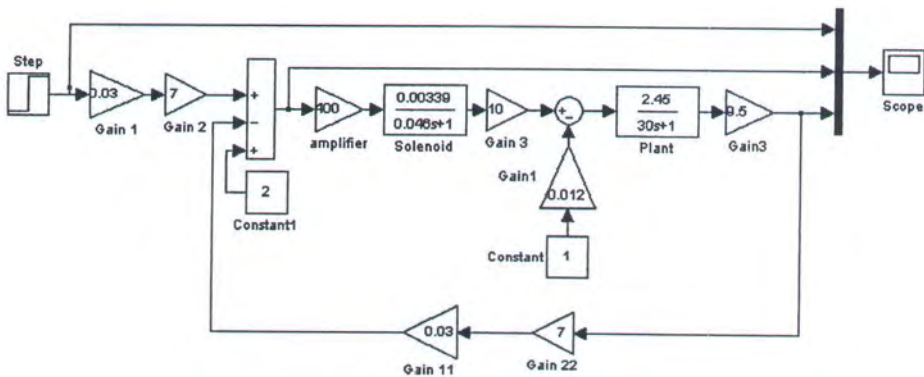
Gambar 4.12 Diagram blok subsistem sensor.

4.5.1.4 Subsistem Amplifier



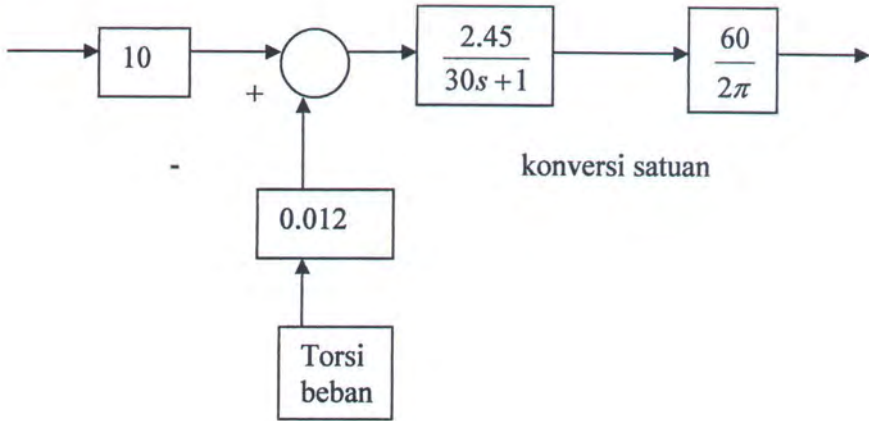
Gambar 4.13 Diagram blok subsistem amplifier

4.5.2 Pemodelan matematis blok diagram sistem dalam model fungsi alih tanpa kontroler fuzzy



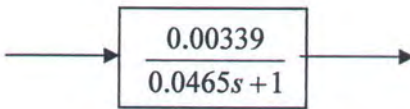
Gambar 4.14 Diagram Blok sistem injeksi bahan bakar dalam model fungsi alih tanpa kontroler logika fuzzy.

4.5.2.1 Subsystem plan



Gambar 4.15 Gambar subsystem mesin diesel kapal

4.5.2.2 Subsystem Solenoida



Gambar 4.16 Diagram blok subsystem solenoida

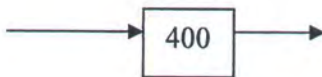
4.5.2.3 Subsystem sensor



Gain sensor Gain peningkatan kecepatan

Gambar 4.17 Diagram blok subsystem sensor.

4.5.2.4 Subsystem Amplifier



Amplifier

Gambar 4.18 Diagram blok subsystem amplifie

4.6 Kontroler logika fuzzy

4.6.1 Variabel masukan dan variabel keluaran

Pada sistem kontrol, selalu dibutuhkan umpan balik. Sinyal impan balik ini dibandingkan dengan sinyal referensi, dan error yang ada kemudian akan menjadi masukan bagi sistem kontrol, dalam hal ini kontrol *fuzzy*. Selain itu ada masukan lain yaitu variabel kecepatan perubahan putaran mesin, dan keluaran dari kontroler logika fuzzy adalah variabel sinyal atur.

4.6.2 Strategi Pengaturan Kontroler Logika Fuzzy

Dalam pemakaian kontroler logika fuzzy terdapat beberapa patameter yang menentukan kemampuan dari kontroler. Parameter-parameter tersebut adalah:

- Pemilihan variabel masukan dan keluaran.
- Himpunan fuzzy dan fungsi keanggotaan.
- Mekanisme inferensi dan implikasi.
- Operator *defuzzyfikasi*.

4.6.3 Basis Data

Penyusunan basis data berarti mendefinisikan himpunan-himpunan fuzzy ke dalam daerah masukan dan keluaran. Prosedur ini ditekankan pada pembagian ruang masukan dan keluaran dengan sejumlah nilai variabel-variabel linguistik (himpunan fuzzy) yang menggunakan fungsi keanggotaan tertentu. Pada tesis ini dalam menyusun basis data fungsi keanggotaan yang akan dipakai adalah bentuk fungsi segitiga seragam.

1. Skala pemetaan

Untuk fungsi seragam, kuantisasi dan normalisasi tidak diperlukan karena masukannya kontinyu dan fungsi keanggotaannya didefinisikan secara fungsional dan bersifat simetris.

2. Pembagian ruang masukan dan keluaran

Dalam tesis ini ruang masukan dan keluaran dibagi dalam 7 variabel linguistik, yaitu: NB (Negatif Besar), NS (Negatif Sedang), NK (Negatif Kecil), NL (Nol), PK (Positif Kecil), PS (Positif Sedang), dan PB (Positif Besar).

3. Pemilihan fungsi keanggotaan

Fungsi keanggotaan yang akan digunakan adalah fungsi segitiga dengan parameter yang telah ditentukan. Pemilihan fungsi segitiga didasari oleh pertimbangan bahwa fungsi ini membatasi jangkauan semesta pembicaraan tiap himpunan fuzzy sehingga jumlah perhitungan untuk menurunkan sinyal atur dapat dikurangi dan waktu perhitungan (eksekusi) dapat ditekan. Penggunaan parameter fungsi yang seragam mengakibatkan pembagian ruang secara merata. Pembagian ini mengandung arti bahwa penggolongan suatu besaran harus proporsional karena mempengaruhi linieritas perilaku kontroler. Berdasarkan ketentuan-ketentuan tersebut dilakukan penyusunan basis data dan hasilnya diungkapkan dalam tabel 4.1 dibawah ini.

Tabel 4.1 Pendefinisian bentuk fungsi segitiga uniform.

| Range Segmen | Jangkauan | a | b | c | Himpunan Fuzzy |
|--------------|---------------|-------|-------|-------|----------------|
| (-R, R) | (-R, -2/3 R) | -4/3R | -R | -2/3R | NB |
| | (-R, -1/3R) | -R | -2/3R | -1/3R | NS |
| | (-2/3R, 0) | -2/3R | -1/3R | 0 | NK |
| | (-1/3R, 1/3R) | -1/3R | R | 1/3R | NL |
| | (0, 2/3R) | 0 | 1/3R | 2/3R | PK |
| | (1/3R, R) | 1/3R | 2/3R | R | PS |
| | (2/3R, R) | 2/3R | R | 4/3R | PB |

Secara rinci pembagian ruang masukan dan keluaran untuk menghasilkan basis data fungsi keanggotaan segitiga adalah sebagai berikut:

Input 1 : error

Range = [-200 200]

NumMFs = 7

MF1 = 'NB' : 'trimf', [-266.6 -200 -133.3]

MF2 = 'NS' : 'trimf', [-200 -133.3 -66.66]

MF3 = 'NK' : 'trimf', [-133.3 -66.66 0]

MF4 = 'NL' : 'trimf', [-66.66 0 66.66]

MF5 = 'PK' : 'trimf', [0 66.66 133.3]

MF6 = 'PS' : 'trimf', [66.66 133.3 200]

MF7 = 'PB' : 'trimf', [133.3 200 266.6]

Input 2 : kecepatan perubahan putaran mesin

Range = [-375 375]

NumMFs = 7

MF1 = 'NB' : 'trimf', [-499.9 -375 -250]

MF2 = 'NS' : 'trimf', [-375 -250 -125]

MF3 = 'NK' : 'trimf', [-250 -125 0]

MF4 = 'NL' : 'trimf', [-125 0 125]

MF5 = 'PK' : 'trimf', [0 125 250]

MF6 = 'PS' : 'trimf', [125 250 375]

MF7 = 'PB' : 'trimf', [250 375 499.9]

Output : Sinyal atur

Range = [-200 200]

NumMFs = 7

MF1 = 'NB' : 'trimf', [-266.6 -200 -133.3]

MF2 = 'NS' : 'trimf', [-200 -133.3 -66.66]

MF3 = 'NK' : 'trimf', [-133.3 -66.66 0]

MF4 = 'NL' : 'trimf', [-66.66 0 66.66]

MF5 = 'PK' : 'trimf', [0 66.66 133.3]

MF6 = 'PS' : 'trimf', [66.66 133.3 200]

MF7 = 'PB' : 'trimf', [133.3 200 266.6]

4.6.4 Penyesuaian Aturan Dasar

Setelah dipahami perilaku pengaturannya, maka berdasarkan pemahaman tersebut penentuan kaidah atur dapat dilakukan. Penurunan prototipe kaidah atur dengan penyempurnaannya secara tuntas menghasilkan suatu *look up* tabel yang disebut tabel Mac Vicar-Whelan.

Tabel 4.2 Tabel Mac Vicar-Whelan.

| Error \ Δ Error | NB | NS | NK | NL | PK | PS | PB |
|------------------------|----|----|----|----|----|----|----|
| NB | NB | NB | NB | NB | NS | NK | NL |
| NS | NB | NB | NB | NS | NK | NL | PK |
| NK | NB | NB | NS | NK | NL | PK | PS |
| NL | NB | NS | NK | NL | PK | PS | PB |
| PK | NS | NK | NL | PK | PS | PB | PB |
| PS | NK | NL | PK | PS | PB | PB | PB |
| PB | NL | PK | PS | PB | PB | PB | PB |

Bila tabel diatas dijabarkan dalam bentuk linguistik maka diperoleh seperangkat pernyataan linguistik fuzzy dengan nomor urutan sesuai dengan nomor baris-kolom pada tabel diatas, sebagai berikut:

- R1 : jika error adalah NB dan Δ error adalah NB maka u (keluaran) adalah NB
- R2 : jika error adalah NB dan Δ error adalah NS maka u (keluaran) adalah NB
- R3 : jika error adalah NB dan Δ error adalah NK maka u (keluaran) adalah NB
- R4 : jika error adalah NB dan Δ error adalah NL maka u (keluaran) adalah NB
- R5 : jika error adalah NB dan Δ error adalah PK maka u (keluaran) adalah NS
- R6 : jika error adalah NB dan Δ error adalah PS maka u (keluaran) adalah NK
- R7 : jika error adalah NB dan Δ error adalah PB maka u (keluaran) adalah NL
- R8 : jika error adalah NS dan Δ error adalah NB maka u (keluaran) adalah NB
- R9 : jika error adalah NS dan Δ error adalah NS maka u (keluaran) adalah NB
- R10 : jika error adalah NS dan Δ error adalah NK maka u (keluaran) adalah NB
- R11 : jika error adalah NS dan Δ error adalah NL maka u (keluaran) adalah NS

- R12 : jika error adalah NS dan Δ error adalah PK maka u (keluaran) adalah NK
- R13 : jika error adalah NS dan Δ error adalah PS maka u (keluaran) adalah NL
- R14 : jika error adalah NS dan Δ error adalah PB maka u (keluaran) adalah PK
- R15 : jika error adalah NK dan Δ error adalah NB maka u (keluaran) adalah NB
- R16 : jika error adalah NK dan Δ error adalah NS maka u (keluaran) adalah NB
- R17 : jika error adalah NK dan Δ error adalah NK maka u (keluaran) adalah NS
- R18 : jika error adalah NK dan Δ error adalah NL maka u (keluaran) adalah NK
- R19 : jika error adalah NK dan Δ error adalah PK maka u (keluaran) adalah NL
- R20 : jika error adalah NK dan Δ error adalah PS maka u (keluaran) adalah PK
- R21 : jika error adalah NK dan Δ error adalah PB maka u (keluaran) adalah PS
- R22 : jika error adalah NL dan Δ error adalah NB maka u (keluaran) adalah NB
- R23 : jika error adalah NL dan Δ error adalah NS maka u (keluaran) adalah NS
- R24 : jika error adalah NL dan Δ error adalah NK maka u (keluaran) adalah NK
- R25 : jika error adalah NL dan Δ error adalah NL maka u (keluaran) adalah NL
- R26 : jika error adalah NL dan Δ error adalah PK maka u (keluaran) adalah PK
- R27 : jika error adalah NL dan Δ error adalah PS maka u (keluaran) adalah PS
- R28 : jika error adalah NL dan Δ error adalah PB maka u (keluaran) adalah PB
- R29 : jika error adalah PK dan Δ error adalah NB maka u (keluaran) adalah NS
- R30 : jika error adalah PK dan Δ error adalah NS maka u (keluaran) adalah NK
- R31 : jika error adalah PK dan Δ error adalah NK maka u (keluaran) adalah NL
- R32 : jika error adalah PK dan Δ error adalah NL maka u (keluaran) adalah PK
- R33 : jika error adalah PK dan Δ error adalah PK maka u (keluaran) adalah PS
- R34 : jika error adalah PK dan Δ error adalah PS maka u (keluaran) adalah PB
- R35 : jika error adalah PK dan Δ error adalah PB maka u (keluaran) adalah PB

- R36 : jika error adalah PS dan Δ error adalah NB maka u (keluaran) adalah NK
- R37 : jika error adalah PS dan Δ error adalah NS maka u (keluaran) adalah NL
- R38 : jika error adalah PS dan Δ error adalah NK maka u (keluaran) adalah PK
- R39 : jika error adalah PS dan Δ error adalah NL maka u (keluaran) adalah PS
- R40 : jika error adalah PS dan Δ error adalah PK maka u (keluaran) adalah PB
- R41 : jika error adalah PS dan Δ error adalah PS maka u (keluaran) adalah PB
- R42 : jika error adalah PS dan Δ error adalah PB maka u (keluaran) adalah PB
- R43 : jika error adalah PB dan Δ error adalah NB maka u (keluaran) adalah NL
- R44 : jika error adalah PB dan Δ error adalah NS maka u (keluaran) adalah PK
- R45 : jika error adalah PB dan Δ error adalah NK maka u (keluaran) adalah PS
- R46 : jika error adalah PB dan Δ error adalah NL maka u (keluaran) adalah PB
- R47 : jika error adalah PB dan Δ error adalah PK maka u (keluaran) adalah PB
- R48 : jika error adalah PB dan Δ error adalah PS maka u (keluaran) adalah PB
- R49 : jika error adalah PB dan Δ error adalah PB maka u (keluaran) adalah PB

4.6.5 Logika Pengambilan Keputusan

Secara praktis logika pengambilan keputusan adalah suatu operator untuk menghitung sinyal atur berdasarkan basis data dan aturan dasar yang telah disusun. Prosedur ini berupa operasi matematika yaitu operasi mini mamdani terhadap pasangan masukan error dan perubahan error yang akan menghasilkan kesimpulan sinyal atur u .

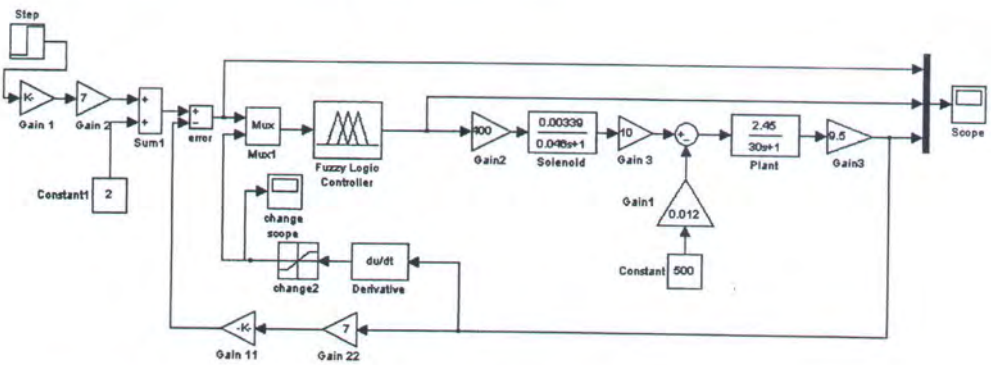
Berdasarkan basis pengetahuan yang telah disusun, setiap pasangan input error dan perubahan error akan menghasilkan sinyal atur yang merupakan keluaran yang akan diumpangkan ke sistem yang diatur (dikendalikan).

4.6.6 Defuzzifikasi

Untuk mengubah sinyal atur fuzzy menjadi sinyal crips, sinyal atur dari logika pengambilan keputusan dilakukan operasi defuzzifikasi. Metode defuzzifikasi yang akan digunakan adalah metode COA.

4.6.7 Diagram Blok Sistem dalam Model Fungsi Alih dengan Kontroler Fuzzy

Setelah kontroler logika fuzzy dirancang maka diagram blok sistem injeksi bahan bakar menjadi seperti pada gambar 4.19.



Gambar 4.19 Diagram blok sistem dalam model fungsi alih dengan kontroler logika fuzzy



BAB V
SIMULASI DAN ANALISA SISTEM

BAB V

SIMULASI DAN ANALISA SISTEM

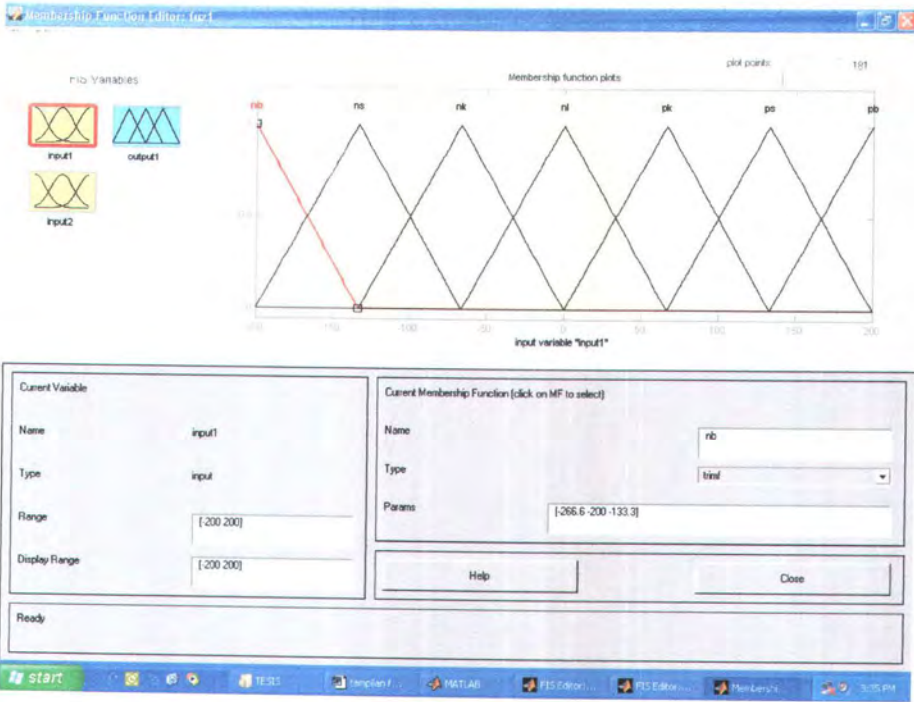
Setelah perancangan kontroler selesai, maka selanjutnya dilakukan simulasi sistem. Simulasi dilakukan dengan menggunakan bahasa pemrograman Matlab 6.1 yang menyediakan fasilitas toolbox yang memuat fungsi-fungsi untuk mendapatkan solusi dari sistem, antara lain:

1. Fuzzy Logic Toolbox, menyediakan fasilitas untuk perancangan kontroler logika fuzzy.
2. Simulink, yang menyediakan blok-blok diagram untuk memodelkan sistem yang akan disimulasikan, dan menjalankan simulasi,
3. GUIDE, untuk membuat tampilan-tampilan guna memudahkan kita mensimulasikan sistem.

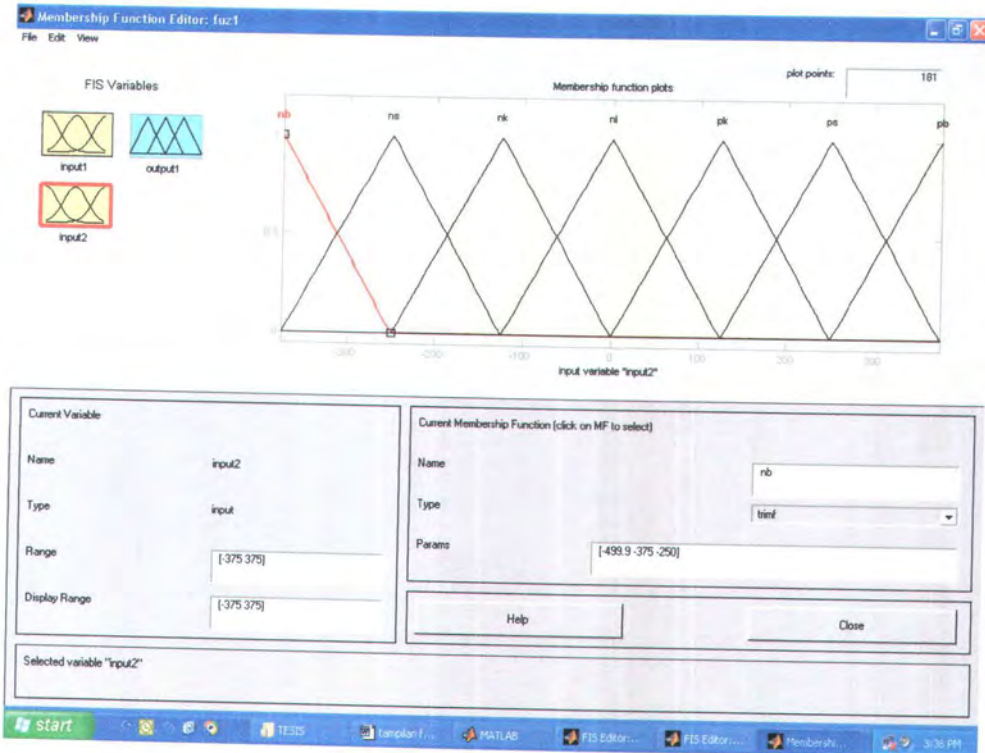
Setelah penentuan variabel masukan pengatur logika fuzzy yaitu variabel error dan kecepatan perubahan putaran mesin, serta variabel keluaran yaitu sinyal atur dengan fungsi keanggotaan segitiga seperti telah disebutkan dalam bab IV, selanjutnya hal tersebut diimplementasikan kedalam program matlab 6.1, seperti ditampilkan pada gambar 5.1 sampai dengan 5.3. Dalam gambar 5.1 dibawah telah ditentukan:

- 7 (tujuh) variabel linguistik yaitu NB, NS, NK, NL, PK, PS, dan PB.
- Range segmen untuk variabel error ditentukan sebesar (-200 200).

Penentuan segmen ini didasarkan pada error hasil simulasi sebelum dipasang kontroler fuzzy.



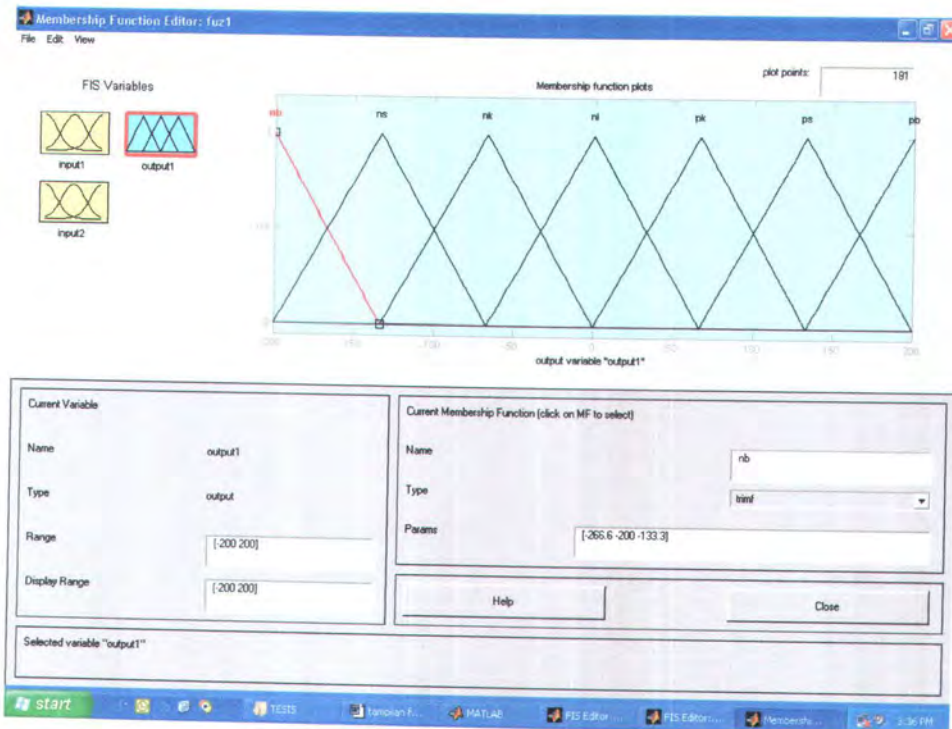
Gambar 5.1 Fungsi Keanggotaan error



Gambar 5.2 Fungsi keanggotaan kecepatan perubahan putaran mesin

Gambar 5.2 diatas merupakan implementasi fungsi keanggotaan kecepatan perubahan putaran mesin pada program matlab 6.1, dimana telah ditentukan:

- 7 (tujuh) variabel linguistik yaitu NB, NS, NK, NL, PJ, PS, dan PB.
- Range segmen untuk variabel kecepatan perubahan putaran mesin ini ditentukan antara (-375 375). Nilai ini merupakan nilai yang ditentukan oleh perancang menurut pengenalan perancang tentang perilaku sistem (plant) dan perilaku kontrolernya.



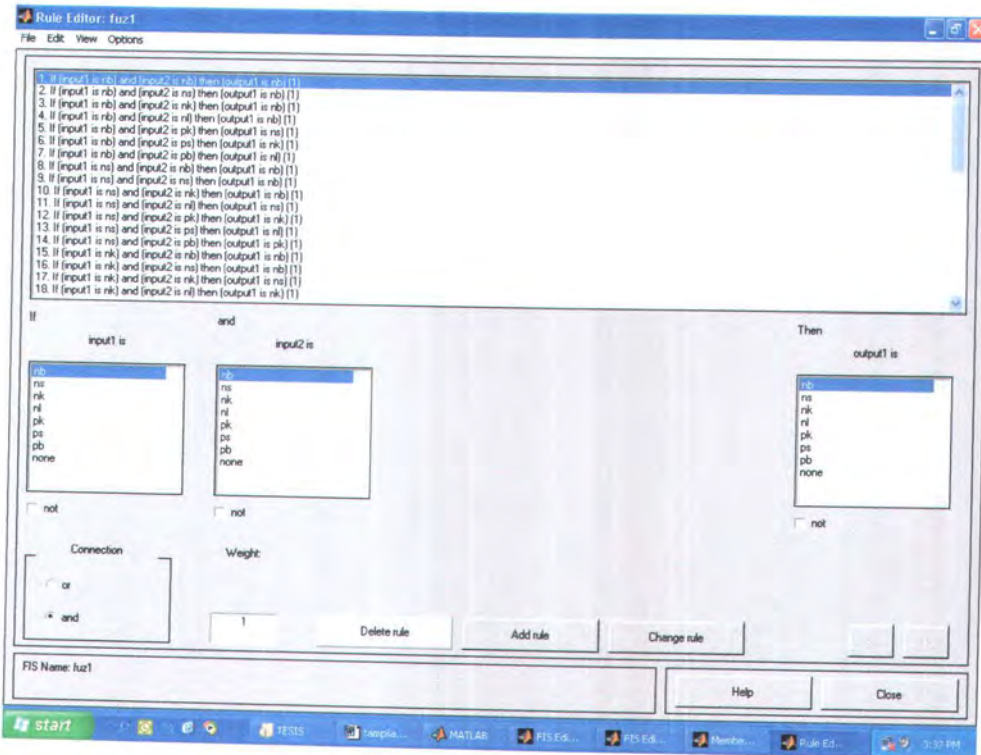
Gambar 5.3 Fungsi keanggotaan output kontroler

Gambar 5.3 diatas merupakan implementasi dari fungsi keanggotaan sinyal atur pada program mathlab 6.1, dimana telah ditentukan:

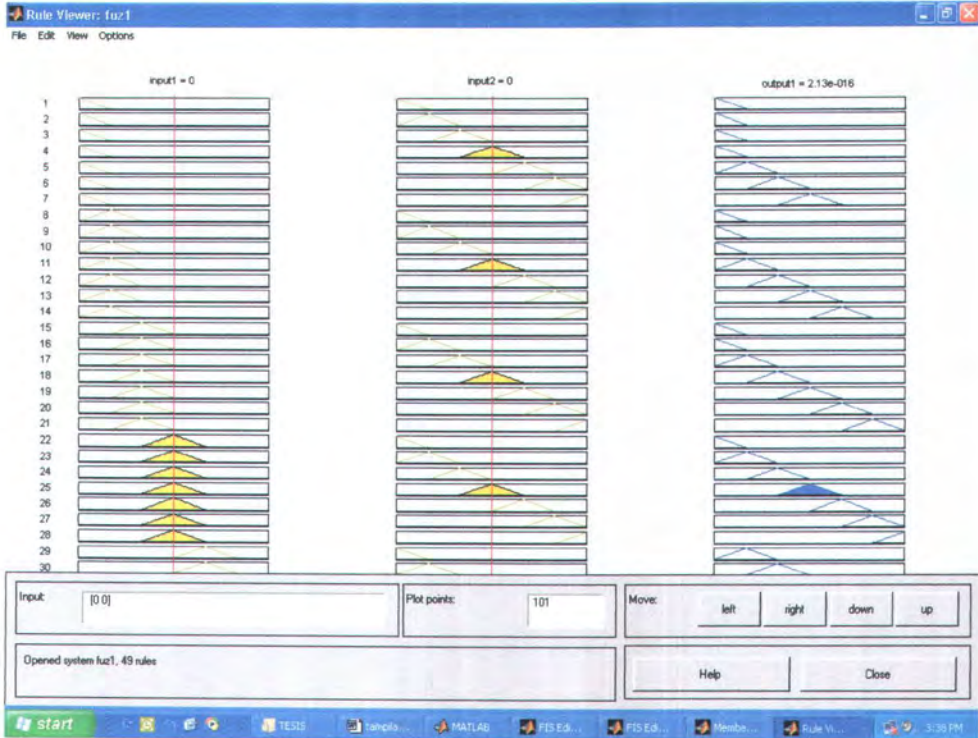
- 7 (tujuh) variabel linguistik yaitu NB, NS, NK, NL, PK, PS dan PB.

- Range segmen variabel sinyal atur yang mengatur gerakan solenoid, berapa lama solenoid diberi sinyal, ditentukan antara (-200 200) yaitu harga yang diperlukan untuk menggerakkan, atau memberi sinyal pada solenoid dalam waktu terpendek, sampai memberi sinyal pada solenoid dalam waktu terlama.

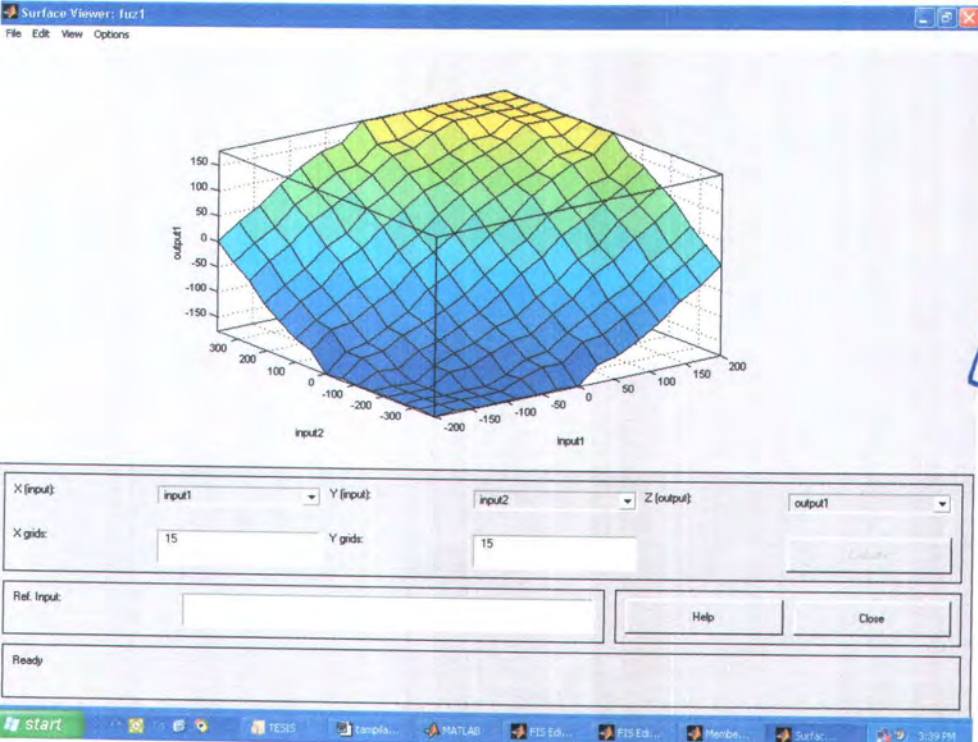
Bila perancangan kontoler logika fuzzy ini dikerjakan dengan program matlab, maka dengan mudah dapat disusun aturan-aturan fuzzy seperti yang telah diterangkan pada bab IV. Gambar 5.4 sampai 5.7 menunjukkan tampilan pembuatan aturan dengan matlab.



Gambar 5.4 Tampilan dalam pembuatan aturan fuzzy (rule)



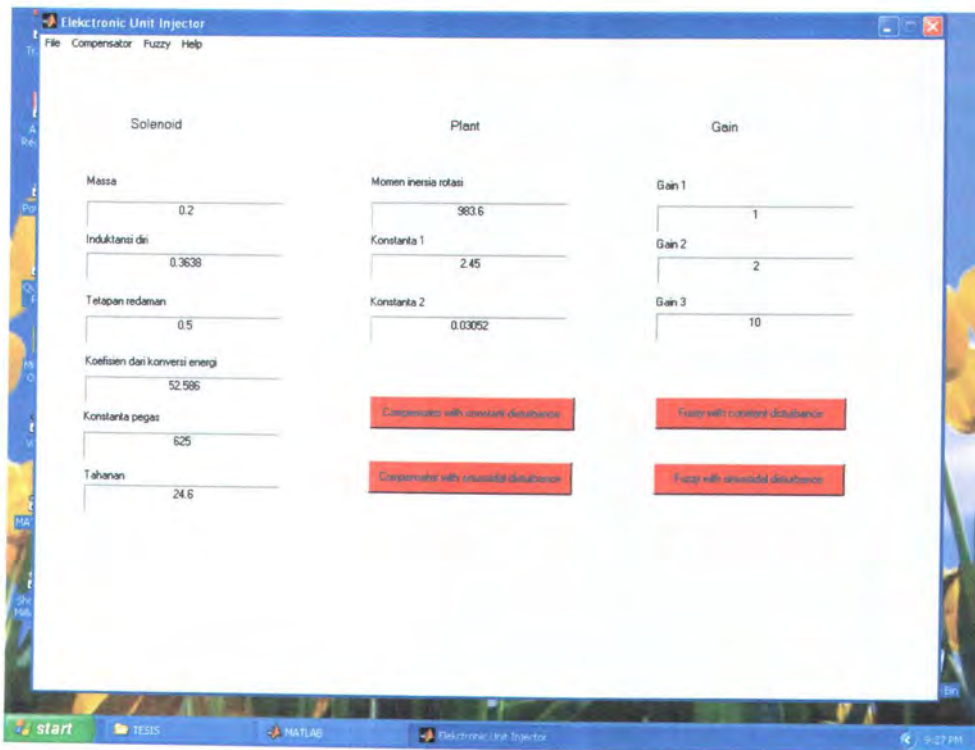
Gambar 5.5 Aturan-aturan dalam bentuk grafis



Gambar 5.6 Gambar Aturan-aturan dalam bentuk tiga dimensi.



Untuk memudahkan simulasi maka dibuat sebuah tampilan, dengan menggunakan fasilitas dalam program matlab yaitu GUIDE. Tampilan yang dinuat seperti pada gambar 5.7.



Gambar 5.7 Gambar tampilan untuk mensimulasikan sistem, yang dibuat dengan GUIDE

5.1 Metode Pengujian

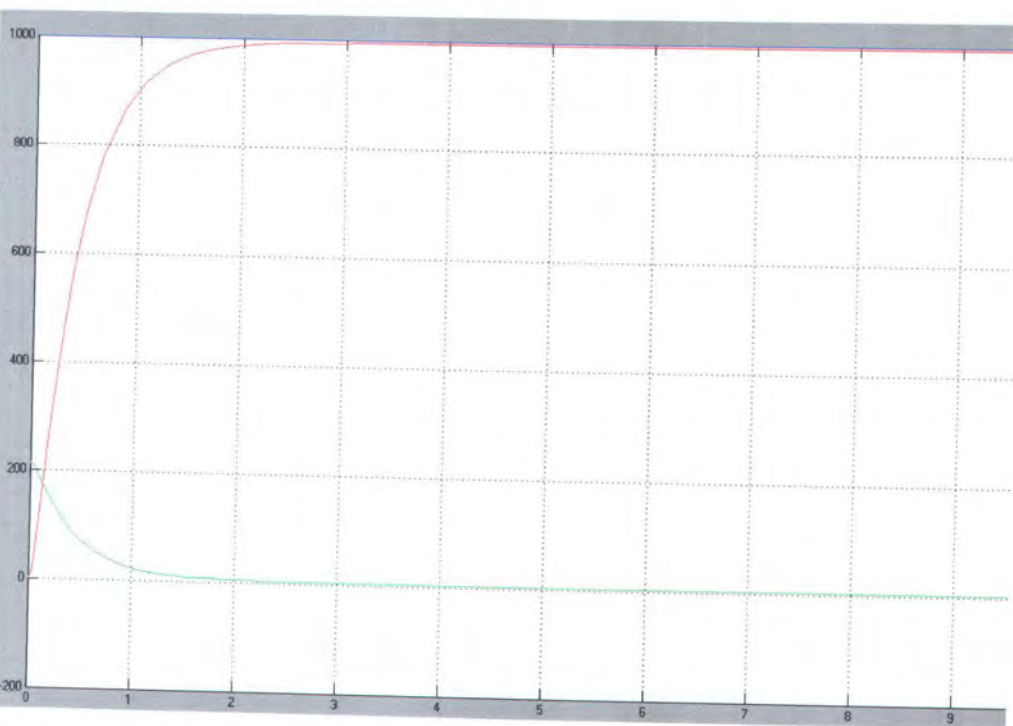
- Pengujian dilakukan pada semua pemodelan yang telah dibuat yaitu meliputi model yang menggunakan state-space dan model dalam bentuk transfer function, baik yang belum diberi kontroler logika fuzzy dan setelah dikontrol dengan kontroler logika fuzzy.
- Semua model tersebut diuji dengan memberi input step 750 rpm dan 1000 rpm, dengan bermacam-macam pembebanan.

- c. Beban yang diberikan berupa beban (torsi) konstan dan dalam bentuk gelombang sinus.

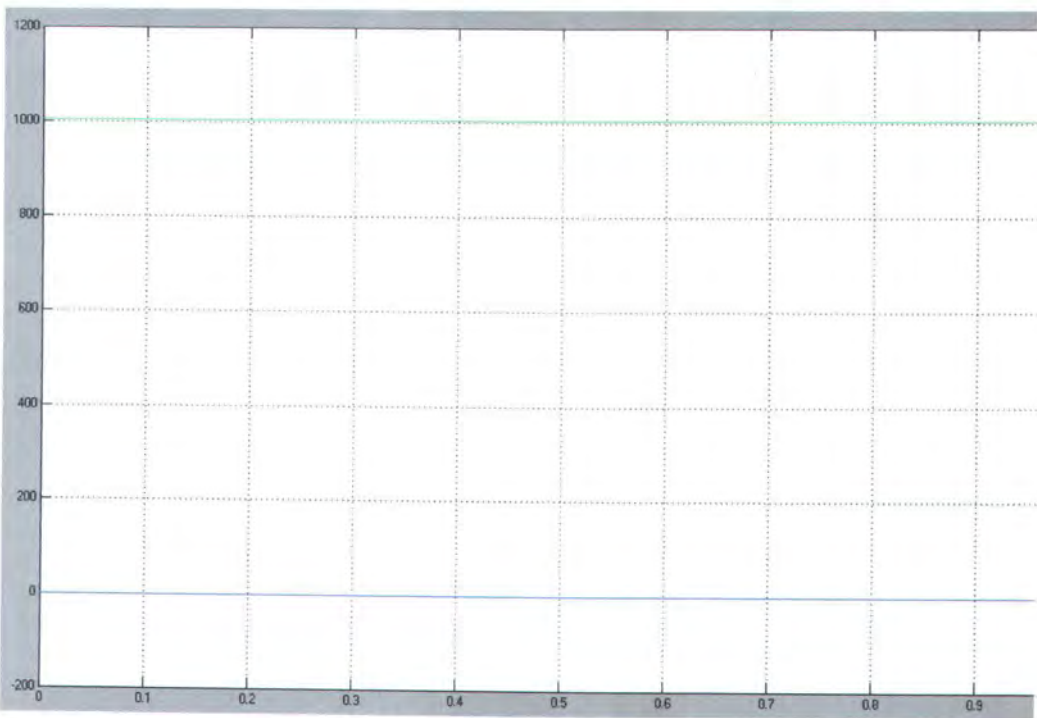
5.2 Analisa Hasil Pengujian.

5.2.1 Pengujian Sistem dengan Model State-Space dibandingkan dengan Model Fungsi Alih (sebelum dikontrol dengan fuzzy)

Hasil pengujian dapat dilihat pada gambar 5.8 dan gambar 5.9 dibawah. Disini tampak perbedaan yang sangat mencolok, dimana untuk pemodelan dengan state-space, memberikan respon spontan, sesuai dengan input yang diberikan. Sedangkan dalam model fungsi alih, diperlukan waktu 3.2 detik untuk mencapai harga *steady-state* yang sesuai dengan *setting point* nya.



Gambar 5.8 Grafik step respon hasil simulasi sistem dalam model fungsi alih dengan input step 1000 rpm, dan pembebanan yang kecil 1 Nm.



Gambar5.9 Grafik step respon hasil simulasi sistem dengan model state-space saat diberi input step 1000 rpm, dengan beban 1 Nm

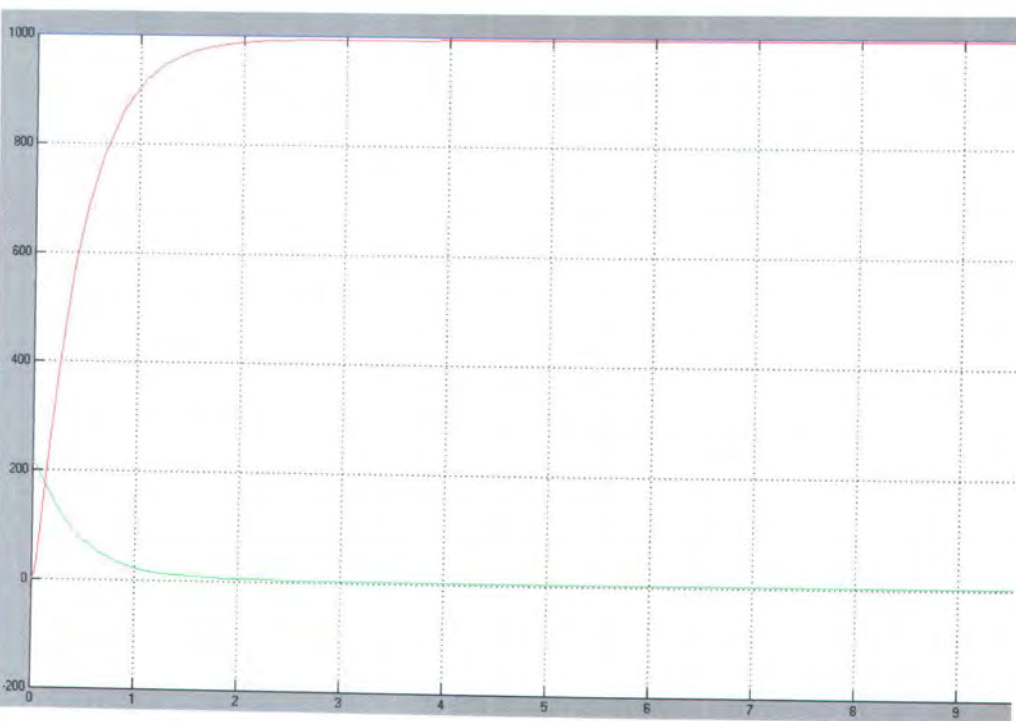
Hal ini disebabkan karena pada pemodelan dalam bentuk fungsi alih, ada beberapa penyederhanaan seperti waktu transformasi laplace dilakukan maka initial condition nya dibuat nol, dan ada pensederhanaan lain dengan menghilangkan beberapa parameter yang dianggap bernilai jauh lebih kecil dari parameter lainnya. Karena itu, untuk mempelajari bagaimana pengaruh penerapan kontroler logika fuzzy pada sistem injeksi bahan bakar mesin diesel kapal, maka model sistem yang akan dikendalikan dengan logika fuzzy adalah model sistem yang dalam bentuk transfer function.

2.2 Pengujian sistem dengan Model Fungsi Alih,tanpa kontroler logika fuzzy dibanding dengan sistem yang menggunakan kontroler logika fuzzy.

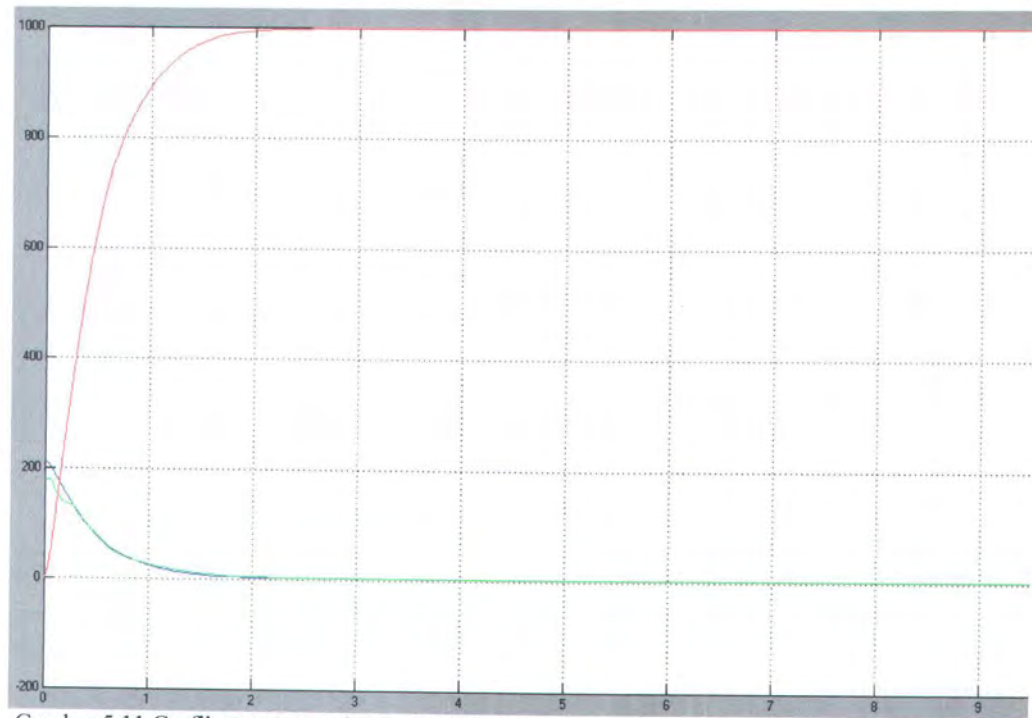
Pengujian ini dilakukan dengan melakukan simulasi sistem baik yang menggunakan kontroler logika fuzzy maupun tidak, dengan beban konstan dan beban

sinusoidal, dengan harga beban yang bermacam-macam. Kemudian dilihat T_s (*settling time*) nya atau waktu yang diperlukan untuk mencapai harga *steady-state*. Disamping itu juga dilihat T_r (*Rise Time*)nya, yaitu waktu yang diperlukan untuk mencapai harga 10% sampai 90% dari harga *steady-state*.

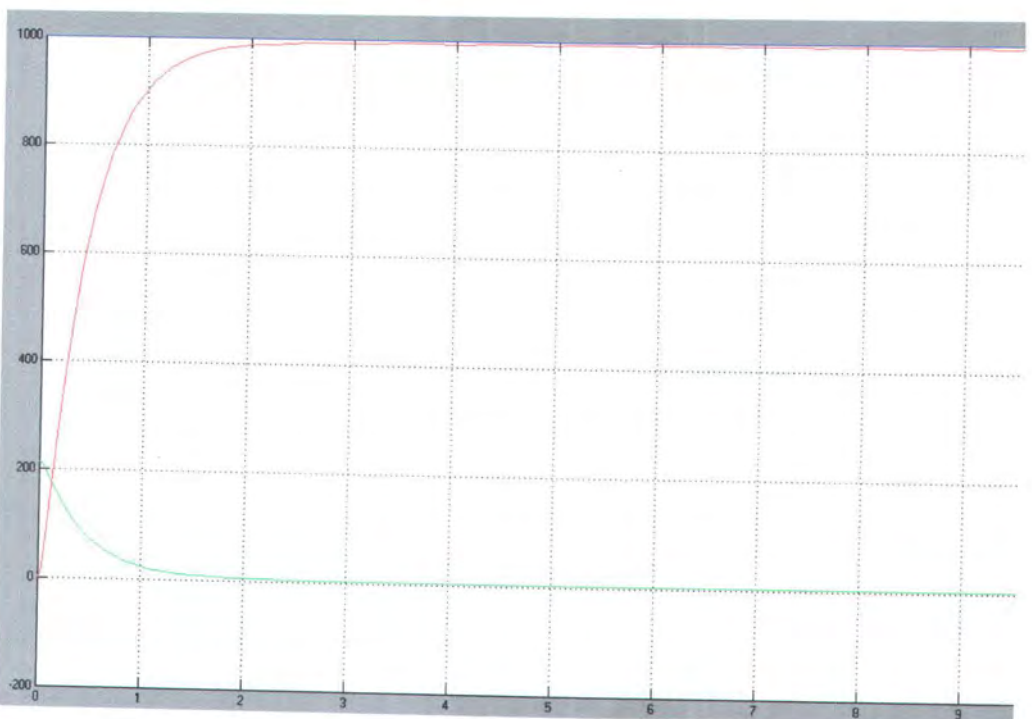
Gambar-gambar dibawah ini merupakan contoh grafik hasil simulasi yang dilakukan. Hasil selengkapnya diuraikan dalam tabel 5.1. Grafik hasil simulasi dari keseluruhan simulasi yang dilakukan, ditampilkan dalam lampiran.



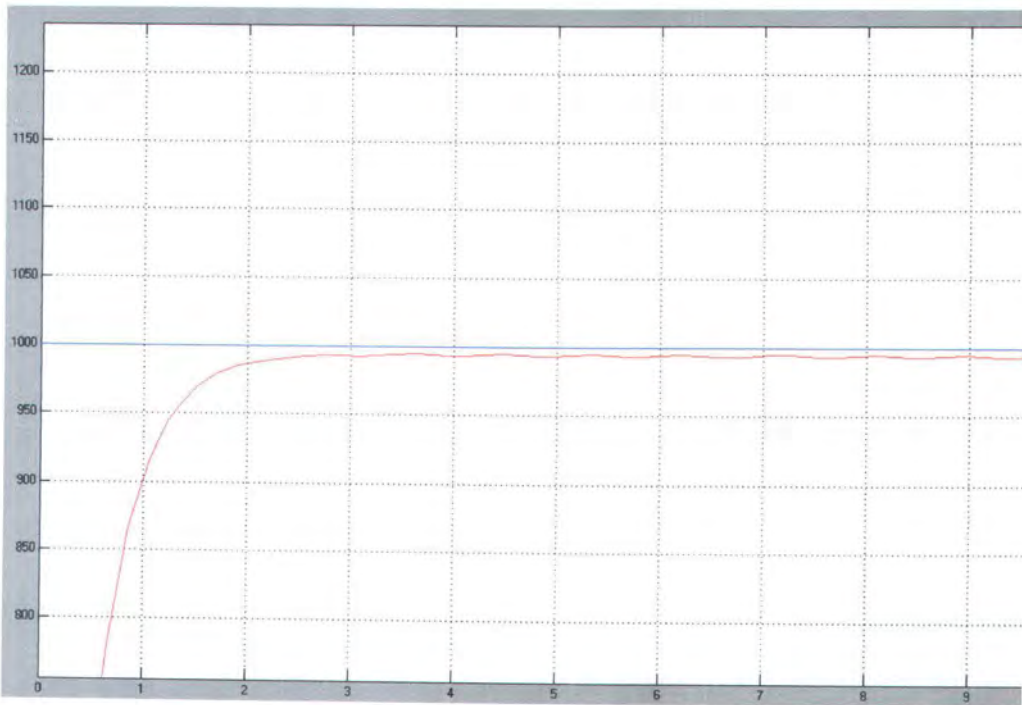
Gambar 5.10 Grafik step respon hasil simulasi sistem tanpa kontroler logika fuzzy, dengan setting point 1000 rpm, dan beban konstan 100 Nm



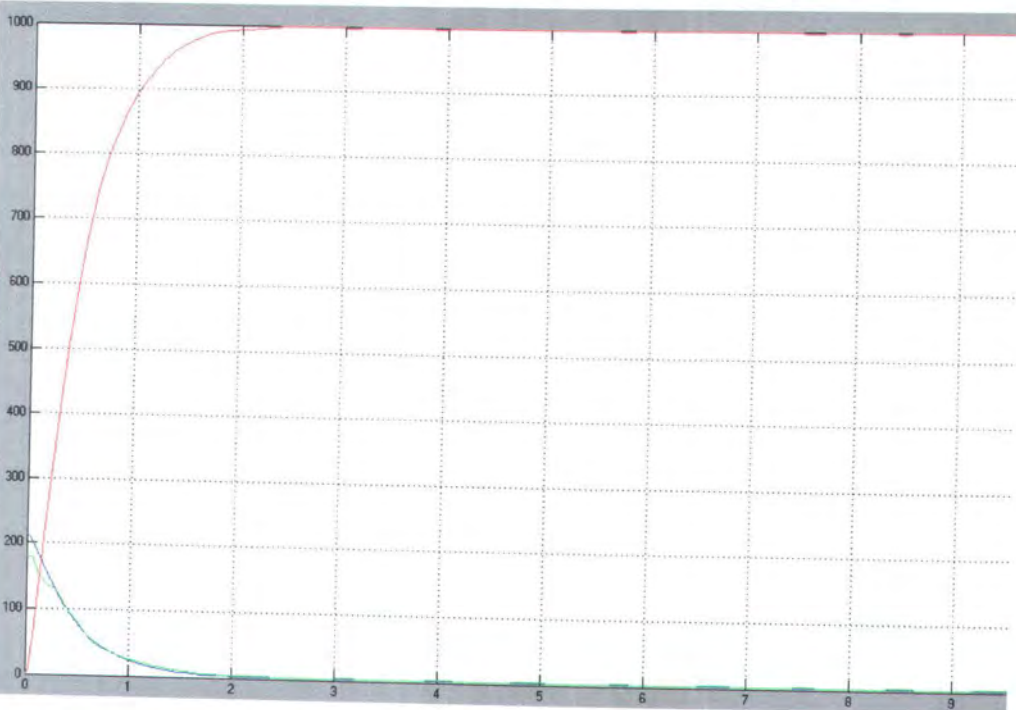
Gambar 5.11 Grafik step respon hasil simulasi sistem dengan kontroler logika fuzzy, dengan setting point 1000 rpm, dan beban konstan 100 Nm



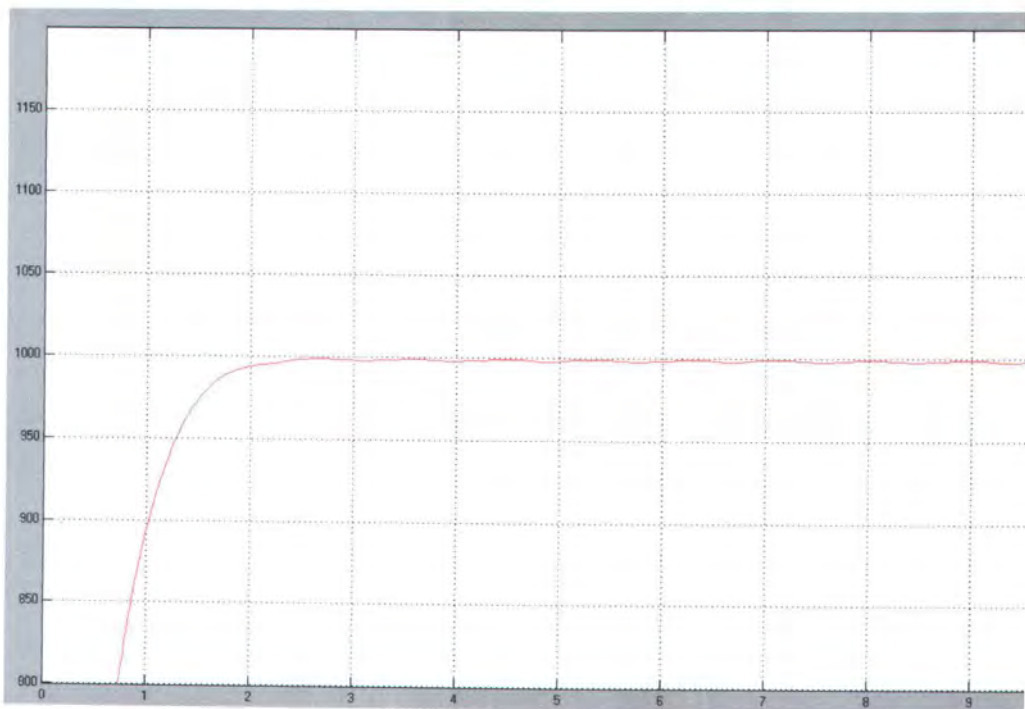
Gambar 5.12 Grafik step respon simulasi sistem tanpa kontroler logika fuzzy, dengan setting point 1000 rpm, dan beban sinusoidal dengan amplitudo 500.



Gambar 5.13 Gambar hasil zoom grafik step respon simulasi sistem tanpa kontroler logika fuzzy, dengan setting point 1000 rpm, dan beban sinusoidal dengan amplitudo 500.



Gambar 5.14 Grafik step respon simulasi sistem dengan kontroler logika fuzzy, dengan setting point 1000 rpm, dan beban sinusoidal dengan amplitudo 500.



Gambar 5.15 Gambar hasil zoom grafik step respon simulasi sistem tanpa kontroler logika fuzzy, dengan setting point 1000 rpm, dan beban sinusoidal dengan amplitudo 500.

Dari gambar 5.10 dan gambar 5.11 bisa dilihat bahwa hasil simulasi menunjukkan waktu yang diperlukan untuk mencapai harga steady-state dari sistem dengan beban konstan 100 Nm, untuk sistem yang tanpa fuzzy adalah 3.2 detik, sedangkan yang dengan kontroler logika fuzzy 2.5 detik. Dan jika gambar tersebut diperbesar maka akan tampak bahwa pada hasil simulasi sistem yang tanpa kontroler logika fuzzy tampak ada error dengan setting point sebesar -10 rpm (hasil simulasi menunjukkan harga steady-state pada 990 rpm, sedang pada sistem yang dikontrol dengan logika fuzzy tidak terdapat penyimpangan. Sedang rise time untuk kedua sistem berharga sama yaitu 0.9 detik.

| | Tanpa Fuzzy | | | | | | Dengan Fuzzy | | | | | |
|-------|---------------|-----------|-----------|-----------|-------------|-----------|---------------|--------|--------|----------|-------------|----------|
| | Beban Konstan | | | | Beban sinus | | Beban Konstan | | | | Beban sinus | |
| | 1 Nm | 100 Nm | 500 Nm | 1000 Nm | Amp 500 | Amp 1000 | 1 Nm | 100 Nm | 500 Nm | 1000 Nm | Amp 500 | Amp 1000 |
| Ts | 3.1 dt | 3.2 dt | 3.3 dt | 3.2 dt | 3 dt | 3.1 dt | 2.5 dt | 2.5 dt | 2.6 dt | 2.7 dt | 2.4 dt | 2.5 dt |
| Tr | 0.9 dt | 0.9 dt | 0.9 dt | 0.9 dt | 0.9 dt | 0.9 dt | 0.9 dt | 0.9 dt | 0.9 dt | 0.9 dt | 0.9 dt | 0.9 dt |
| error | (-10 rpm) | (-10 rpm) | (-10 rpm) | (-15 rpm) | (-10 rpm) | (-15 rpm) | 0 | 0 | 0 | (-5 rpm) | 0 | 0 |

Tabel 5.1 Hasil simulasi step respon sistem pengendalian tanpa kontroler fuzzy dan dengan kontroler logika fuzzy pada setting point 1000 rpm, dengan beban yang berbeda-beda.

| | Tanpa Fuzzy | | | | | | Dengan Fuzzy | | | | | |
|----|-------------|--------|--------|--------|---------|---------|--------------|--------|--------|---------|---------|---------|
| | 1 Nm | 100 Nm | 500 Nm | 700 Nm | 1000 Nm | 1200 Nm | 1 Nm | 100 Nm | 500 Nm | 700 Nm | 1000 Nm | 1200 Nm |
| Ts | 3,1 dt | 3,2 dt | 3,3 dt | 3,5 dt | 3,7 dt | 3,9 dt | 2,5 dt | 2,5 dt | 2,6 dt | 2,65 dt | 2,7 dt | 2,7 dt |

Tabel 5.2 Hasil simulasi step respon sistem pengendalian tanpa kontroler fuzzy dan dengan kontroler logika fuzzy, dengan setting point 1000 rpm, pada kondisi beban(konstan) yang berbeda-beda.

| | Tanpa Fuzzy | | | | | | Dengan Fuzzy | | | | | |
|----|-------------|---------|---------|---------|----------|----------|--------------|---------|---------|---------|----------|----------|
| | 500 rpm | 700 rpm | 800 rpm | 900 rpm | 1000 rpm | 1200 rpm | 500 rpm | 700 rpm | 800 rpm | 900 rpm | 1000 rpm | 1200 rpm |
| Ts | 2,35 dt | 2,45 dt | 2,55 dt | 2,6 dt | 2,7 dt | 2,75 | 3,12 dt | 3,4 dt | 3,5 dt | 3,55 dt | 3,7 dt | 3,7 dt |

Tabel 5.3 Hasil simulasi step respon sistem pengendalian tanpa kontroler fuzzy dan dengan kontroler logika fuzzy, dengan kondisi beban konstan 1000 Nm, pada kondisi setting point yang berbeda-beda.

Dari tabel 5.1 bisa dilihat bahwa rise time (T_r) untuk semua simulasi yang dilakukan berharga sama yaitu lebih kurang 0,9 detik. Sedangkan waktu yang diperlukan untuk mencapai harga steady-state (T_s) berbeda-beda demikian pula simpangan (error) yang terjadi.

Sistem dengan beban kecil atau mendekati nol disini diwakili beban 1 Nm, yang tanpa kontroler logika fuzzy waktu untuk mencapai harga steady-state (T_s) adalah 3,1 detik, sedang yang dikontrol dengan logika fuzzy 2,5 detik. Dan ada penyimpangan sebesar -10 rpm dari harga setting pointnya untuk sistem yang tanpa kontroler logika fuzzy.

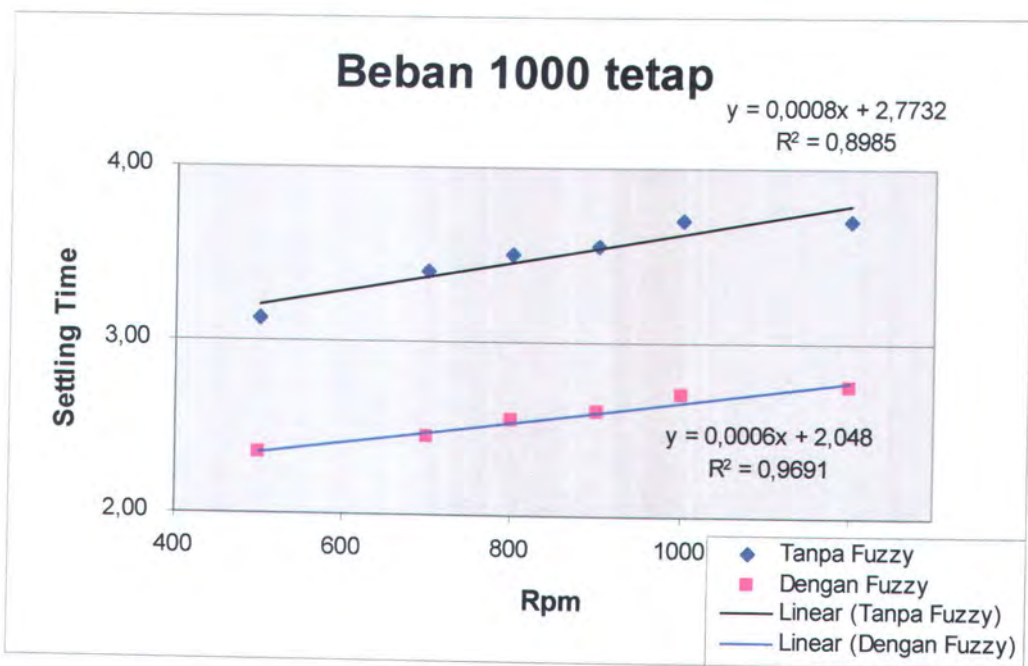
Sistem dengan beban konstan 500 Nm, yang tanpa kontroler logika fuzzy waktu untuk mencapai harga steady-state (T_s) adalah 3,3 detik, sedang yang dikontrol dengan logika fuzzy 2,6 detik. Dan ada penyimpangan sebesar -10 rpm dari harga setting pointnya untuk yang tanpa kontroler logika fuzzy. Sedangkan yang dikontrol dengan logika fuzzy penyimpangannya kecil yaitu -2 rpm.

Sistem dengan beban konstan 1000 Nm, yang tanpa kontroler logika fuzzy waktu untuk mencapai harga steady-state (T_s) adalah 3,2 detik, sedang yang dikontrol dengan logika fuzzy 2,7 detik. Dan ada penyimpangan sebesar -15 rpm dari harga setting pointnya untuk yang tanpa kontroler logika fuzzy. Sedangkan yang dikontrol dengan fuzzy penyimpangannya -5 rpm.

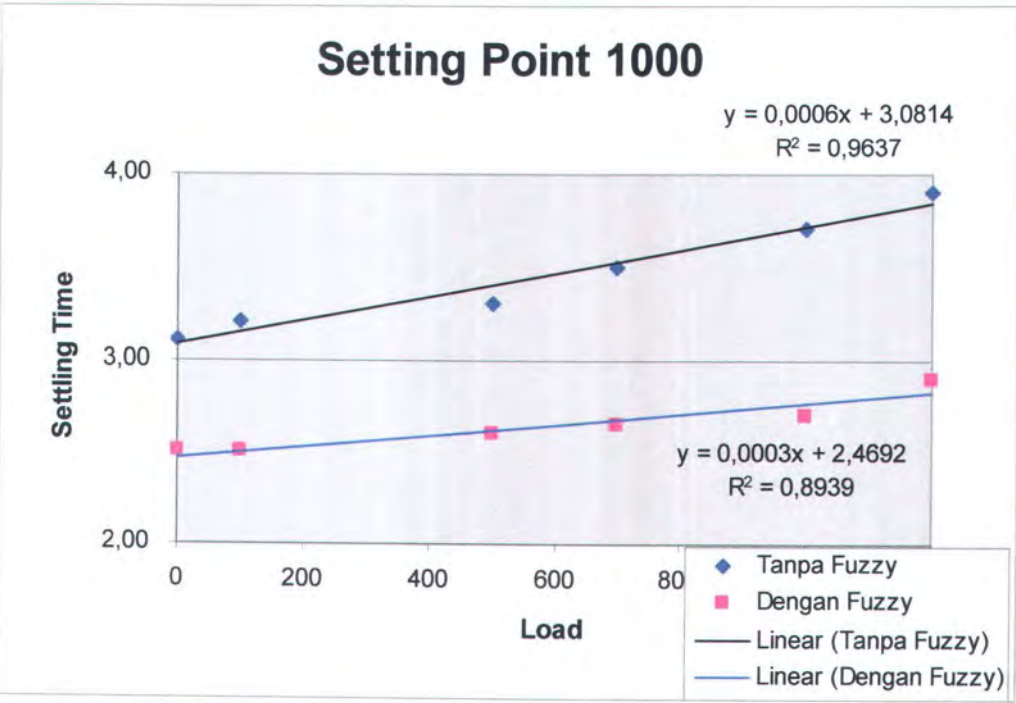
Sistem dengan beban sinusoidal dengan amplitudo 500 Nm, yang tanpa kontroler logika fuzzy waktu untuk mencapai harga steady-state (T_s) adalah 3 detik, sedang yang dikontrol dengan logika fuzzy 2,4 detik. Dan ada penyimpangan sebesar 10 rpm dari harga setting pointnya untuk yang tanpa kontroler logika fuzzy.

Sistem dengan beban sinusoidal dengan amplitudo 1000 Nm, yang tanpa kontroler logika fuzzy waktu untuk mencapai harga steady-state (T_s) adalah 3,1 detik, sedang yang dikontrol dengan logika fuzzy 2,5 detik. Dan ada penyimpangan sebesar -15 rpm dari harga setting pointnya untuk yang tanpa kontroler logika fuzzy.

Dari tabel 5.2 dan tabel 5.3 dapat digambarkan grafik hubungan antara perubahan beban terhadap settling time, pada keadaan setting point tetap yaitu 1000 rpm, dan grafik hubungan antara perubahan setting point (rpm) terhadap settling time, pada keadaan beban konstan (1000 Nm). Grafik tersebut dapat dilihat pada gambar dibawah ini.



Gambar 5.16 Grafik hubungan antara perubahan setting point (putaran mesin), terhadap settling time(T_s), pada keadaan beban konstan (1000 Nm)



Gambar 5.17 Grafik hubungan antara perubahan beban terhadap settling time (Ts), pada keadaan setting point konstan (1000 rpm).

Dari gambar 5.16 dapat dilihat bahwa pada keadaan beban konstan, semakin besar setting point yang ditentukan, waktu untuk mencapai harga steady-state (settling time Ts) semakin lama. Dan dari gambar 5.17 dapat dilihat bahwa pada keadaan setting point tetap (1000 rpm), semakin besar beban mesin, waktu untuk mencapai harga steady-state (Ts) semakin lama.

Dari gambar 5.16 dan 5.17 juga bisa dilihat bahwa grafik yang menggambarkan hasil simulasi sistem dengan kontroler logika fuzzy, memiliki gradien yang lebih kecil jika dibandingkan dengan gradien grafik yang menggambarkan hasil simulasi sistem tanpa kontroler logika fuzzy. Dari sini dapat disimpulkan bahwa penerapan kontroler logika fuzzy pada sistem injeksi bahan bakar mesin diesel kapal lebih menguntungkan, karena dengan penambahan beban

yang dilakukan , peningkatan settling timenya lebih kecil jika dibandingkan dengan sistem yang tanpa kontroler logika fuzzy. Demikian juga dengan penambahan harga seting point, pada sistem dengan kontroler logika fuzzy, peningkatan settling pointnya lebih kecil, jika dibanding dengan sistem yang tanpa kontroler logika fuzzy.



BAB VI
KESIMPULAN DAN SARAN

BAB VI

KESIMPULAN DAN SARAN

KESIMPULAN

1. Pada pemodelan sistem injeksi bahan bakar mesin diesel kapal yang dibahas dalam tesis ini, yang menggunakan model state-space yang dilinierisasikan tidak terdapat parameter-parameter yang dihilangkan, sehingga pada saat disimulasikan, responnya sangat spontan.
2. Pada pemodelan sistem injeksi bahan bakar mesin diesel kapal yang menggunakan menggunakan model transfer function, ada beberapa parameter yang dihilangkan seperti massa plunyer, konstanta redaman dan lainnya yang dianggap berharga relatif jauh lebih kecil dibanding dengan harga parameter-parameter lainnya. Disamping itu initial condition dalam penurunan model laplacenya juga dianggap nol. Sehingga sekalipun semua parameter-parameter yang digunakan sama dengan yang digunakan pada pemodelan dengan state-space, responnya tidak sama, tidak spontan, tetapi ada selang waktu yang diperlukan untuk mencapai harga steady-state.
3. Waktu yang diperlukan untuk mencapai harga steady-state (T_s) untuk sistem yang tidak dikontrol dengan kontroler logika fuzzy antara 3,1 sampai 3,3 detik. Sedang yang dikontrol dengan kontroler logika fuzzy butuh waktu antara 2,4 sampai 2,7 detik. Dari sini bisa disimpulkan bahwa settling time sistem yang

dikontrol dengan kontroler logika fuzzy lebih bagus dari pada sistem yang tidak dikontrol dengan kontroler logika fuzzy.

4. Waktu untuk menempuh harga respon dari 10% sampai 90% (rise time T_r), untuk sistem yang dikontrol dengan fuzzy maupun tidak, harganya relatif sama (pengamatannya sangat sulit untuk dicari selisihnya), yaitu 0,9 detik.
5. Sistem yang tidak dikontrol dengan logika fuzzy, harga steady-statenya terjadi penyimpangan dari setting point yang diberikan, penyimpangan yang terjadi sekitar -10 sampai -15 rpm. Sedang pada sistem yang dikontrol dengan logika fuzzy, relatif tidak terjadi penyimpangan. Terjadi penyimpangannya hanya pada simulasi dengan beban konstan sebesar 1000 Nm yaitu penyimpangannya -5 rpm, masih lebih kecil jika dibanding dengan penyimpangan yang terjadi pada sistem yang tidak dikontrol dengan fuzzy.
6. Dari semua analisa yang telah dilakukan maka sistem injeksi bahan bakar mesin diesel kapal, yang menggunakan electrical unit injector, akan bekerja lebih baik jika padanya diterapkan kontroler logika fuzzy.
7. Pada beban konstan yaitu 1000 rpm, baik dengan kontroler logika fuzzy maupun tidak, semakin besar setting point yang ditentukan maka semakin lama waktu yang diperlukan untuk mencapai harga steady-state.
8. Pada setting point yang tetap yaitu 1000 rpm, baik sistem dengan kontroler logika fuzzy maupun tidak, semakin besar torsi beban yang diberikan, maka semakin lama waktu yang digunakan untuk mencapai harga steady-state.

9. Software untuk simulasi telah dibuat, dan programnya terlampir pada lampiran A.
10. Penerapan kontroler logika fuzzy pada sistem injeksi mesin diesel kapal lebih menguntungkan, karena pada keadaan setting point konstan, dengan penambahan beban mesin, peningkatan settling timenya lebih kecil jika dibandingkan dengan sistem yang tanpa kontroler logika fuzzy. Demikian juga pada keadaan beban konstan, dengan penambahan setting point (rpm), peningkatan settling timenya lebih kecil jika dibanding dengan sistem yang tanpa kontroler logika fuzzy.

SARAN

1. Penelitian ini bisa dikembangkan lagi yaitu dengan membuat demo bagaimana sistem electrical unti injektor bekerja, dan bagaimana aksi kontrolnya.
2. Untuk lebih sempurnanya penelitian semacam ini, maka lebih bagus lagi jika data yang digunakan adalah data primer dari uji mesin secara langsung.



DAFTAR PUSTAKA

DAFTAR PUSTAKA

1. John F Dagel, Robert N. Brady. (2001), *Diesel Engine and Fuel System Repair* ; Ohio ; Edisi ke 5.
2. Bahram Shahian, Michael Hassul. (1993), *Control System Design Using Matlab* .
3. Ogata, Katsuhiko. (1994), *Solving Control Engineering Problems with MATLAB*; Prentice-Hall International Inc; New Jersey.
4. Gene F Franklin, J David Powel, Abbas Emami, Naeini. (2002), *Feedback Control of Dynamic System* ; USA ; Edisi ke 4.
5. Fitzgerald, A.E. (1990). *Mesin-mesin Listrik* ; diterjemahkan oleh Djoko Achyanto; Erlangga.
6. Passino, Kevin M, and Yurkovich, Stephen. (1998) *Fuzzy Control*. Addison Wesley Longman, Inc., 2725 Sand Hill Road Park, California 94025.
7. Sugeno, Michio. (1999), *Fuzzy Modeling And Control*, dalam Selected Work of M, Sugeno. Eds: Nguyen, Hung T; Prasad, Nadipuram R. CRC Press LLC. Boca Raton, London, New York, Washington, D.C.
8. Mohammad Jamshidi; Andre Titli, Lotfi Zadeh and Serge Boverie. (1997), *Application of Fuzzy Logic, Towards High Machine Intelligence Quotient System*.
9. Roland S Burns. (2001), *Advanced Control Engineering*.
10. Ogata, Katsuhiko. (1985), *Teknik Kontrol Automatik I* , diterjemahkan oleh Edi Leksono, Erlangga, Jakarta.

11. Ogata, Katsuhiko. (1985), *Teknik Kontrol Automatik II*, diterjemahkan oleh Edi Leksono, Erlangga, Jakarta.
12. Munif, Abd, & Santoso, Agus. (1993), *Analisa Respon Transien Sistem Pengendalian Aliran Bahan Bakar Pada mesin Diesel Penggerak Propeller*, Laporan Penelitian, ITS, Surabaya.
13. Dwilaksono, Widigdo. (2001), *Studi Perancangan Sistem Pengendalian Kecepatan Putaran Motor Diesel Kapal Dengan Menggunakan Kontroler Logika Fuzzy*, Tugas Akhir, ITS, Surabaya.
14. Heru Hermawan N. (1999), *Modifikasi Sistem Pengendalian Konvensional Kecepatan Motor Induk Kapal untuk Meningkatkan Time Respon Performance Menggunakan servomekanisme elektrohidrolik*. Tugas Akhir, ITS, Surabaya.
15. Janusz Zieba, *Simulation of Solenoid Actuator for a Device for Investigating Dynamic Air Permeability Through Flat Textile Product*. Technixal University of Textile Processes, Ul Zeromskiego 116, 90-543 Lodz, Poland.



LAMPIRAN

LAMPIRAN A PROGRAM SIMULASI DENGAN MATLAB

```
function varargout = tesis(varargin)
% TESIS Application M-file for tesis.fig
% FIG = TESIS launch tesis GUI.
% TESIS('callback_name', ...) invoke the named callback.

% Last Modified by GUIDE v2.0 27-Jun-2005 11:54:38

if nargin == 0 % LAUNCH GUI

    fig = openfig(mfilename,'reuse');

    % Generate a structure of handles to pass to callbacks, and store it.
    handles = guihandles(fig);
    guidata(fig, handles);
    % EUI picture
    h0=figure('Name','Electrical Unit Injector','NumberTitle','off',...
        'ToolBar','none','MenuBar','none');
    set(gca,'Position',[0 0 1 1])
    [x,map] = imread('gbeui','bmp');
    image(x, colormap(map));
    set(gca,'visible','off')

    if nargin > 0
        varargout{1} = fig;
    end

elseif ischar(varargin{1}) % INVOKE NAMED SUBFUNCTION OR CALLBACK

    try
        if (nargout)
            [varargout{1:nargout}] = feval(varargin{:}); % FEVAL
switchyard
        else
            feval(varargin{:}); % FEVAL switchyard
        end
    catch
        disp(lasterr);
    end

end

%| ABOUT CALLBACKS:
```



```

%| GUIDE automatically appends subfunction prototypes to this file, and
%| sets objects' callback properties to call them through the FEVAL
%| switchyard above. This comment describes that mechanism.
%|
%| Each callback subfunction declaration has the following form:
%| <SUBFUNCTION_NAME>(H, EVENTDATA, HANDLES, VARARGIN)
%|
%| The subfunction name is composed using the object's Tag and the
%| callback type separated by '_', e.g. 'slider2_Callback',
%| 'figure1_CloseRequestFcn', 'axis1_ButtondownFcn'.
%|
%| H is the callback object's handle (obtained using GCBO).
%|
%| EVENTDATA is empty, but reserved for future use.
%|
%| HANDLES is a structure containing handles of components in GUI using
%| tags as fieldnames, e.g. handles.figure1, handles.slider2. This
%| structure is created at GUI startup using GUIHANDLES and stored in
%| the figure's application data using GUIDATA. A copy of the structure
%| is passed to each callback. You can store additional information in
%| this structure at GUI startup, and you can change the structure
%| during callbacks. Call guidata(h, handles) after changing your
%| copy to replace the stored original so that subsequent callbacks see
%| the updates. Type "help guihandles" and "help guidata" for more
%| information.
%|
%| VARARGIN contains any extra arguments you have passed to the
%| callback. Specify the extra arguments by editing the callback
%| property in the inspector. By default, GUIDE sets the property to:
%| <MFILENAME>('<SUBFUNCTION_NAME>', gcbo, [], guidata(gcbo))
%| Add any extra arguments after the last argument, before the final
%| closing parenthesis.

```

```

% -----
function varargout = file_Callback(h, eventdata, handles, varargin)

```

```

% -----
function varargout = print_Callback(h, eventdata, handles, varargin)
print -f handles.figure1

```

```

% -----
function varargout = close_Callback(h, eventdata, handles, varargin)
delete(handles.figure1)

% -----
function varargout = comp_Callback(h, eventdata, handles, varargin)

% -----
function varargout = comp_const_Callback(h, eventdata, handles, varargin)
cmcn

% -----
function varargout = comp_sinus_Callback(h, eventdata, handles, varargin)
cmsin

% -----
function varargout = fuzzy_Callback(h, eventdata, handles, varargin)

% -----
function varargout = help_Callback(h, eventdata, handles, varargin)

% -----
function varargout = using_Callback(h, eventdata, handles, varargin)
HelpPath = which('eui_help.html');
web(HelpPath);

% -----
function varargout = about_Callback(h, eventdata, handles, varargin)
h0=figure('Name','About Electrical Unit Injector','NumberTitle','off',...
    'ToolBar','none','MenuBar','none');

```

```
set(gca,'Position',[0 0 1 1])
[x,map] = imread('gbeui','bmp');
image(x), colormap(map);
set(gca,'visible','off')
```

```
% -----
function varargout = edit1_Callback(h, eventdata, handles, varargin)
M = str2double(get(handles.edit1,'String'))
```

```
% -----
function varargout = edit2_Callback(h, eventdata, handles, varargin)
Lo = str2double(get(handles.edit2,'String'))
```

```
% -----
function varargout = edit3_Callback(h, eventdata, handles, varargin)
B = str2double(get(handles.edit3,'String'))
```

```
% -----
function varargout = edit4_Callback(h, eventdata, handles, varargin)
Ko = str2double(get(handles.edit4,'String'))
```

```
% -----
function varargout = edit5_Callback(h, eventdata, handles, varargin)
K = str2double(get(handles.edit5,'String'))
```

```
% -----
function varargout = edit6_Callback(h, eventdata, handles, varargin)
R = str2double(get(handles.edit6,'String'))
```

```
% -----
function varargout = edit7_Callback(h, eventdata, handles, varargin)
J = str2num(get(handles.edit7,'String'))
```

```
% -----  
function varargout = edit8_Callback(h, eventdata, handles, varargin)  
C1 = str2num(get(handles.edit8,'String'))
```

```
% -----  
function varargout = edit9_Callback(h, eventdata, handles, varargin)  
C2 = str2num(get(handles.edit9,'String'))
```

```
% -----  
function varargout = edit10_Callback(h, eventdata, handles, varargin)  
g1 = str2double(get(handles.edit10,'String'))
```

```
% -----  
function varargout = edit11_Callback(h, eventdata, handles, varargin)  
g2 = str2double(get(handles.edit11,'String'))
```

```
% -----  
function varargout = edit12_Callback(h, eventdata, handles, varargin)  
g3 = str2double(get(handles.edit12,'String'))
```

```
% -----  
function varargout = comcons_Callback(h, eventdata, handles, varargin)  
find_system('Name','ajar');  
open_system('ajar');  
set_param('ajar/Gain 1','Gain',get(handles.edit10,'String'))  
set_param('ajar/Gain 11','Gain',get(handles.edit10,'String'))  
set_param('ajar/Gain 2','Gain',get(handles.edit11,'String'))  
set_param('ajar/Gain 22','Gain',get(handles.edit11,'String'))  
set_param('ajar/Gain 3','Gain',get(handles.edit12,'String'))  
%open_system('ajar/Solenoid');  
%set_param('ajar/Solenoid/Massa','Gain',get(handles.edit1,'String'))  
%set_param('ajar/Solenoid/Induktansi Diri','Gain',get(handles.edit2,'String'))  
%set_param('ajar/Solenoid/Tetapan Redaman','Gain',get(handles.edit3,'String'))  
%set_param('ajar/Solenoid/Koef Konv Energi','Gain',get(handles.edit4,'String'))  
%set_param('ajar/Solenoid/Koefisien Konversi  
Energi','Gain',get(handles.edit4,'String'))
```

```

%set_param('ajar/Solenoid/Transfer Fcn1','Denominator',get(handles.edit5,'String'))
%set_param('ajar/Solenoid/Transfer Fcn','Denominator',get(handles.edit6,'String'))
%open_system('ajar/Plant');
set_param('ajar/Plant','Numerator',get(handles.edit8,'String'))
%set_param('ajar/Plant/Konstanta 2','Gain',get(handles.edit9,'String'))
%J = str2num(get(handles.edit7,'String'))
%C2 = str2num(get(handles.edit9,'String'))
%JC2=J*C2
%nJC2=num2str(JC2)
%set_param('ajar/Plant','Denominator',[nJC2 1])
sim('ajar')

```

```

% -----
function varargout = fuzzyconst_Callback(h, eventdata, handles, varargin)
find_system('Name','ajar2');
open_system('ajar2');
set_param('ajar2/Gain 1','Gain',get(handles.edit10,'String'))
set_param('ajar2/Gain 11','Gain',get(handles.edit10,'String'))
set_param('ajar2/Gain 2','Gain',get(handles.edit11,'String'))
set_param('ajar2/Gain 22','Gain',get(handles.edit11,'String'))
set_param('ajar2/Gain 3','Gain',get(handles.edit12,'String'))
%open_system('ajar2/Solenoid');
%set_param('ajar2/Solenoid/Massa','Gain',get(handles.edit1,'String'))
%set_param('ajar2/Solenoid/Induktansi Diri','Gain',get(handles.edit2,'String'))
%set_param('ajar2/Solenoid/Tetapan Redaman','Gain',get(handles.edit3,'String'))
%set_param('ajar2/Solenoid/Koef Konv Energi','Gain',get(handles.edit4,'String'))
%set_param('ajar2/Solenoid/Koefisien Konversi
Energi','Gain',get(handles.edit4,'String'))
%set_param('ajar2/Solenoid/Transfer Fcn1','Denominator',get(handles.edit5,'String'))
%set_param('ajar2/Solenoid/Transfer Fcn','Denominator',get(handles.edit6,'String'))
%open_system('ajar2/Plant');
set_param('ajar2/Plant','Numerator',get(handles.edit8,'String'))
%set_param('ajar2/Plant/Konstanta 2','Gain',get(handles.edit9,'String'))
%J = str2num(get(handles.edit7,'String'));
%C2 = str2num(get(handles.edit9,'String'));
%JC2=J*C2;
%set_param('ajar2/Plant/Momen inersi kali Konstanta 2','Gain',num2str(JC2))
sim('ajar2')

```

```

% -----
function varargout = compsin_Callback(h, eventdata, handles, varargin)
find_system('Name','ajar1');
open_system('ajar1');
set_param('ajar1/Gain 1','Gain',get(handles.edit10,'String'))

```

```

set_param('ajar1/Gain 11','Gain',get(handles.edit10,'String'))
set_param('ajar1/Gain 2','Gain',get(handles.edit11,'String'))
set_param('ajar1/Gain 22','Gain',get(handles.edit11,'String'))
set_param('ajar1/Gain 3','Gain',get(handles.edit12,'String'))
%open_system('ajar1/Solenoid');
%set_param('ajar1/Solenoid/Massa','Gain',get(handles.edit1,'String'))
%set_param('ajar1/Solenoid/Induktansi Diri','Gain',get(handles.edit2,'String'))
%set_param('ajar1/Solenoid/Tetapan Redaman','Gain',get(handles.edit3,'String'))
%set_param('ajar1/Solenoid/Koef Konv Energi','Gain',get(handles.edit4,'String'))
%set_param('ajar1/Solenoid/Koefisien Konversi
Energi','Gain',get(handles.edit4,'String'))
%set_param('ajar1/Solenoid/Transfer Fcn1','Denominator',get(handles.edit5,'String'))
%set_param('ajar1/Solenoid/Transfer Fcn','Denominator',get(handles.edit6,'String'))
%open_system('ajar1/Plant');
set_param('ajar1/Plant','Numerator',get(handles.edit8,'String'))
%set_param('ajar1/Plant/Konstanta 2','Gain',get(handles.edit9,'String'))
%J = str2num(get(handles.edit7,'String'));
%C2 = str2num(get(handles.edit9,'String'));
%JC2=J*C2;
%set_param('ajar1/Plant/Momen inersi kali Konstanta 2','Gain',num2str(JC2))
sim('ajar1')

```

```

% -----
function varargout = fuzzysin_Callback(h, eventdata, handles, varargin)
find_system('Name','ajar4');
open_system('ajar4');
set_param('ajar4/Gain 1','Gain',get(handles.edit10,'String'))
set_param('ajar4/Gain 11','Gain',get(handles.edit10,'String'))
set_param('ajar4/Gain 2','Gain',get(handles.edit11,'String'))
set_param('ajar4/Gain 22','Gain',get(handles.edit11,'String'))
set_param('ajar4/Gain 3','Gain',get(handles.edit12,'String'))
%open_system('ajar4/Solenoid');
%set_param('ajar4/Solenoid/Massa','Gain',get(handles.edit1,'String'))
%set_param('ajar4/Solenoid/Induktansi Diri','Gain',get(handles.edit2,'String'))
%set_param('ajar4/Solenoid/Tetapan Redaman','Gain',get(handles.edit3,'String'))
%set_param('ajar4/Solenoid/Koef Konv Energi','Gain',get(handles.edit4,'String'))
%set_param('ajar4/Solenoid/Koefisien Konversi
Energi','Gain',get(handles.edit4,'String'))
%set_param('ajar4/Solenoid/Transfer Fcn1','Denominator',get(handles.edit5,'String'))
%set_param('ajar4/Solenoid/Transfer Fcn','Denominator',get(handles.edit6,'String'))
%open_system('ajar4/Plant');
set_param('ajar4/Plant','Numerator',get(handles.edit8,'String'))
%set_param('ajar4/Plant/Konstanta 2','Gain',get(handles.edit9,'String'))
%J = str2num(get(handles.edit7,'String'));
%C2 = str2num(get(handles.edit9,'String'));

```

```
%JC2=J*C2;  
%set_param('ajar4/Plant/Momen inersi kali Konstanta 2','Gain',num2str(JC2))  
sim('ajar4')
```

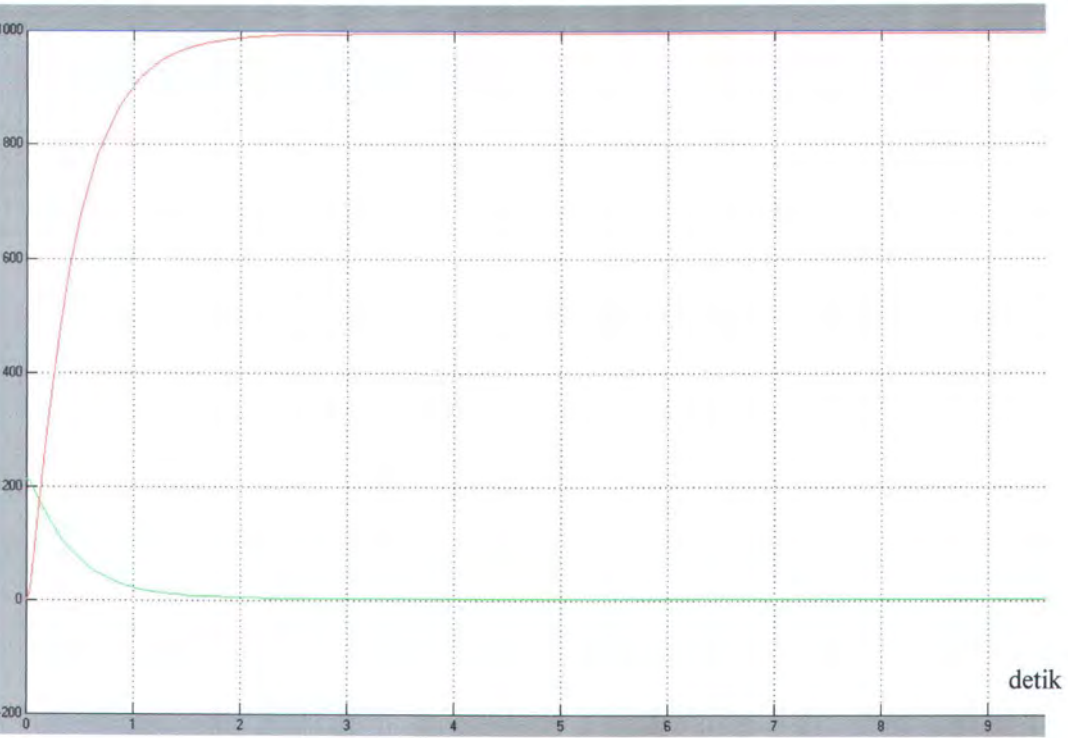
```
% -----  
function varargout = tfld_Callback(h, eventdata, handles, varargin)  
tfld
```

```
% -----  
function varargout = flsd_Callback(h, eventdata, handles, varargin)  
flsd
```

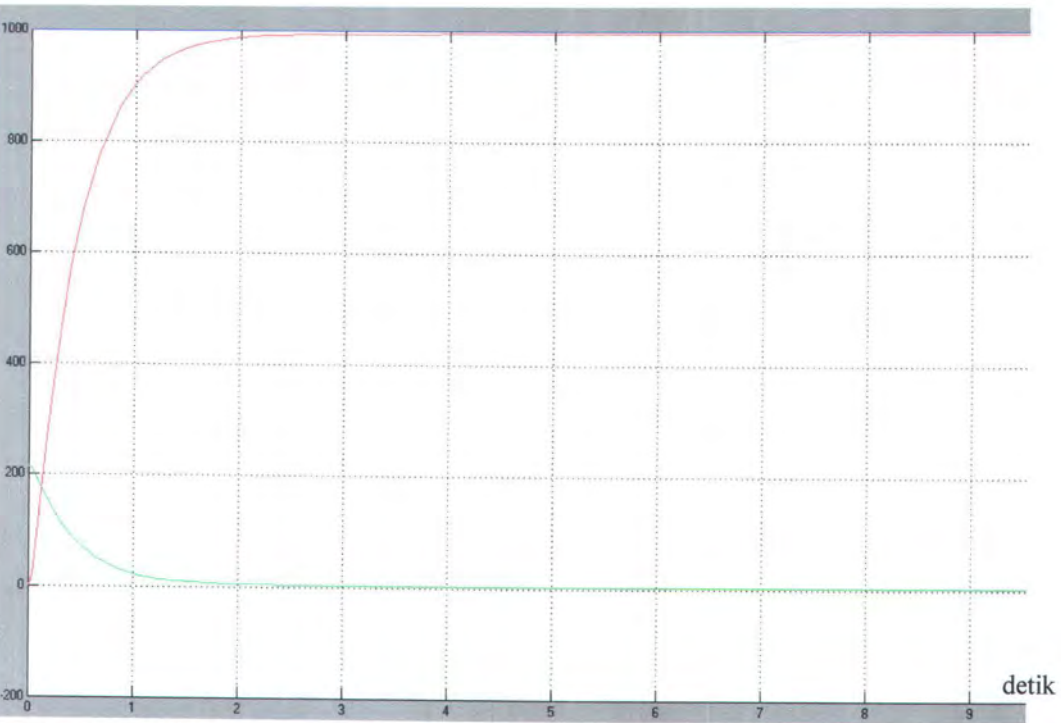
```
% -----  
function varargout = fld_Callback(h, eventdata, handles, varargin)  
fld
```

```
% -----  
function varargout = flsd_Callback(h, eventdata, handles, varargin)  
flsd
```

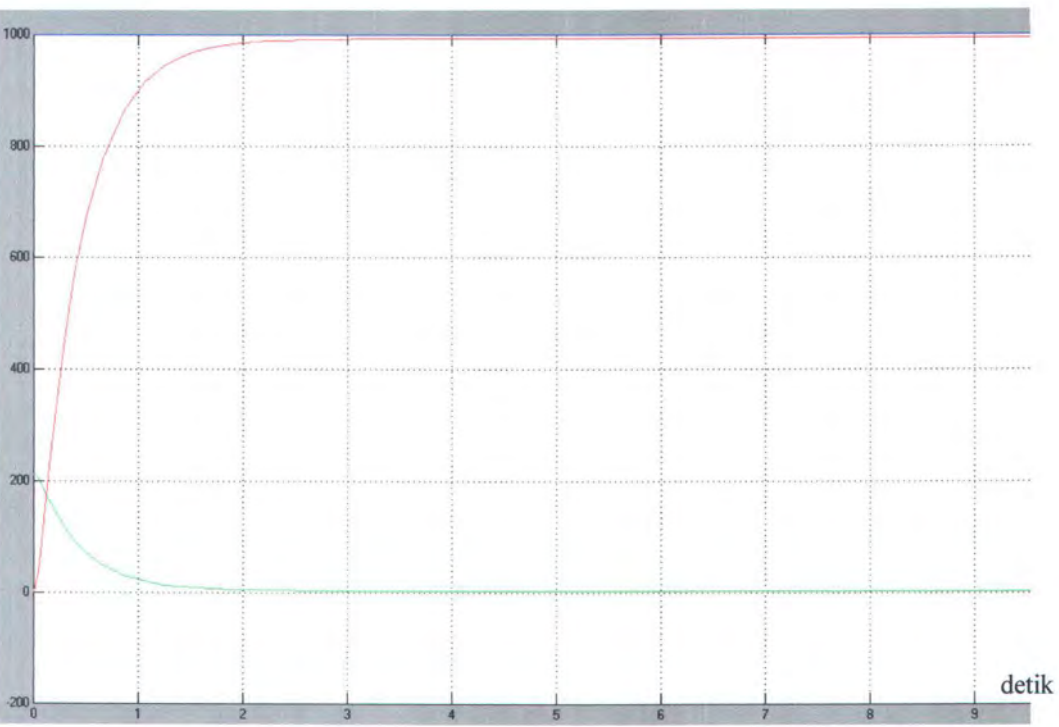
LAMPIRAN B. HASIL SIMULASI



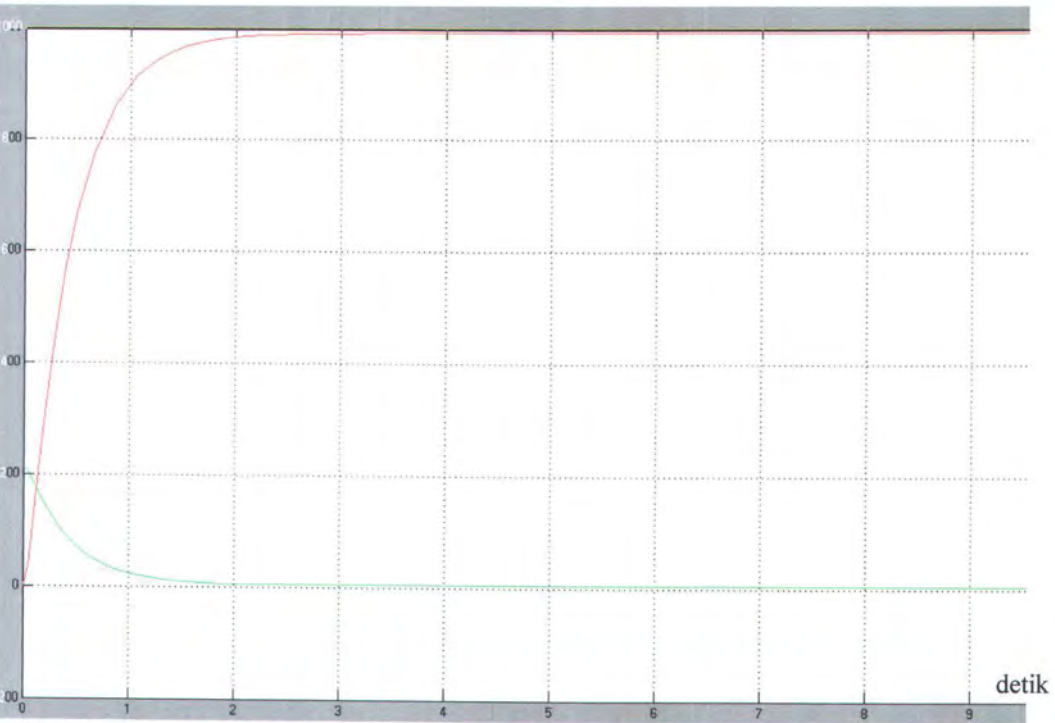
step respon hasil simulasi sistem tanpa kontroler logika fuzzy, dengan setting point 1000 rpm, dan konstan 1Nm.



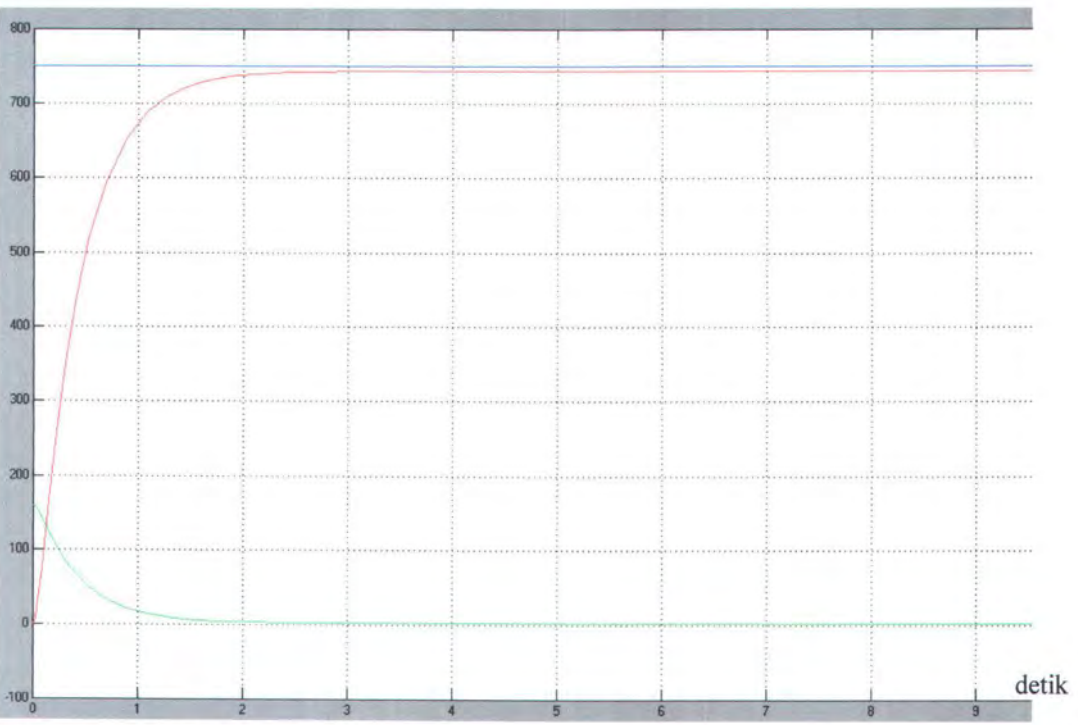
step respon hasil simulasi sistem tanpa kontroler logika fuzzy, dengan setting point 1000 rpm, dan konstan 100 Nm.



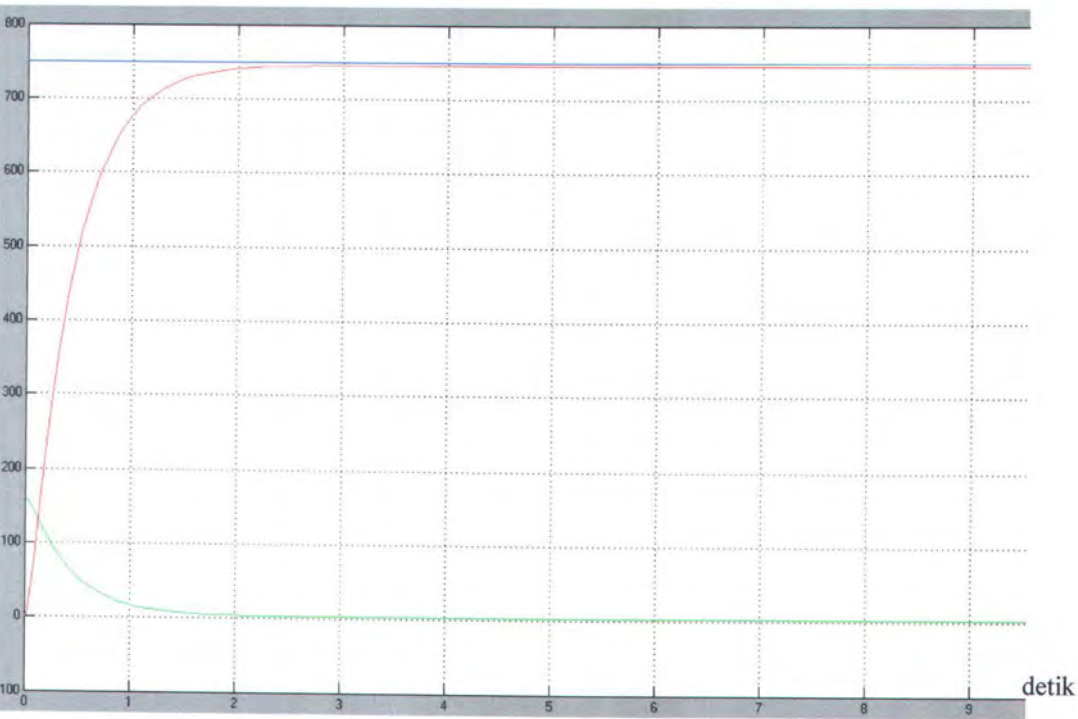
step respon hasil simulasi sistem tanpa kontroler logika fuzzy, dengan setting point 1000 rpm, dan konstan 500 Nm.



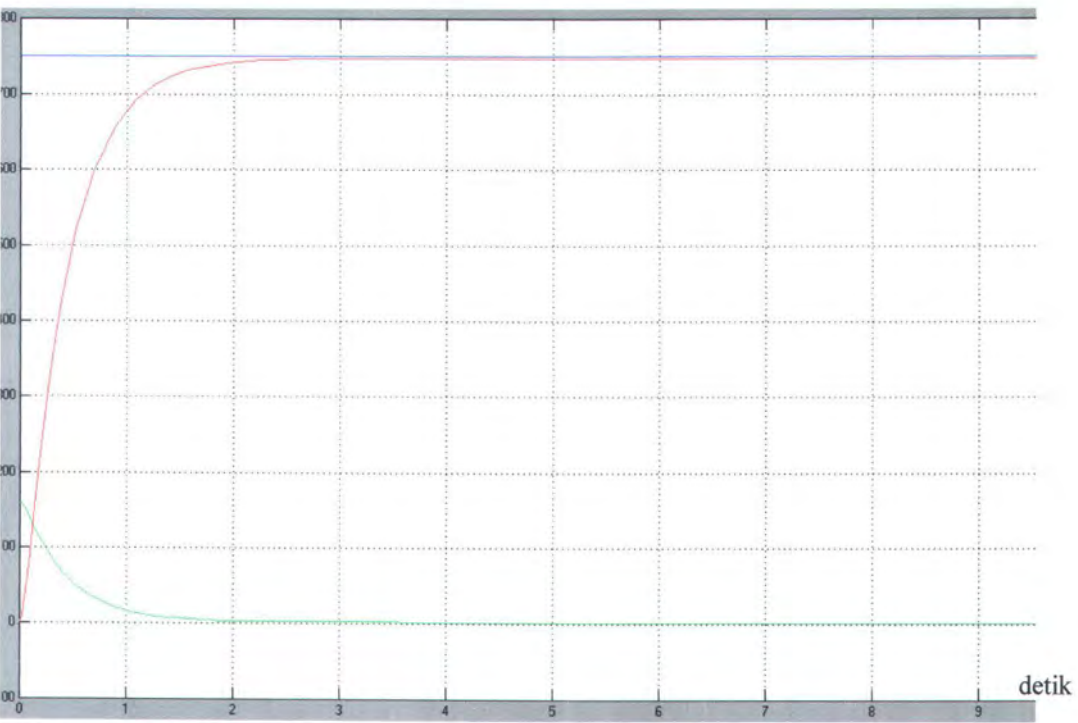
step respon hasil simulasi sistem tanpa kontroler logika fuzzy, dengan setting point 1000 rpm, dan konstan 1Nm.



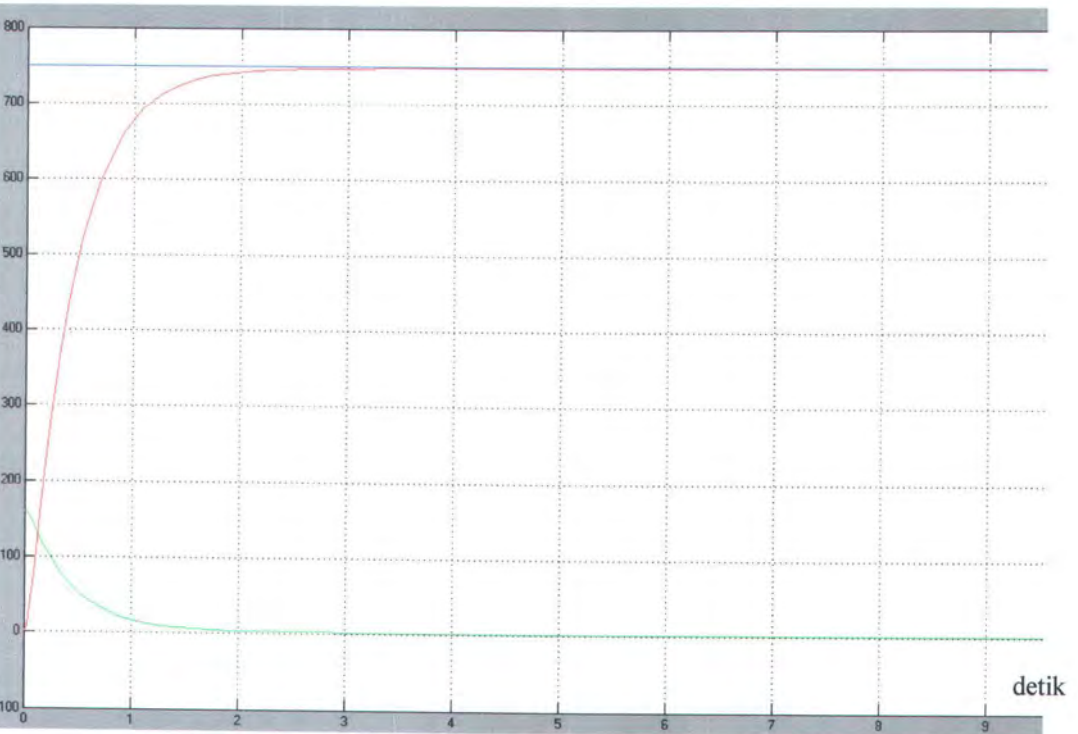
Step respon hasil simulasi sistem tanpa kontroler logika fuzzy, dengan setting point 750 rpm, dan konstan 1000 Nm.



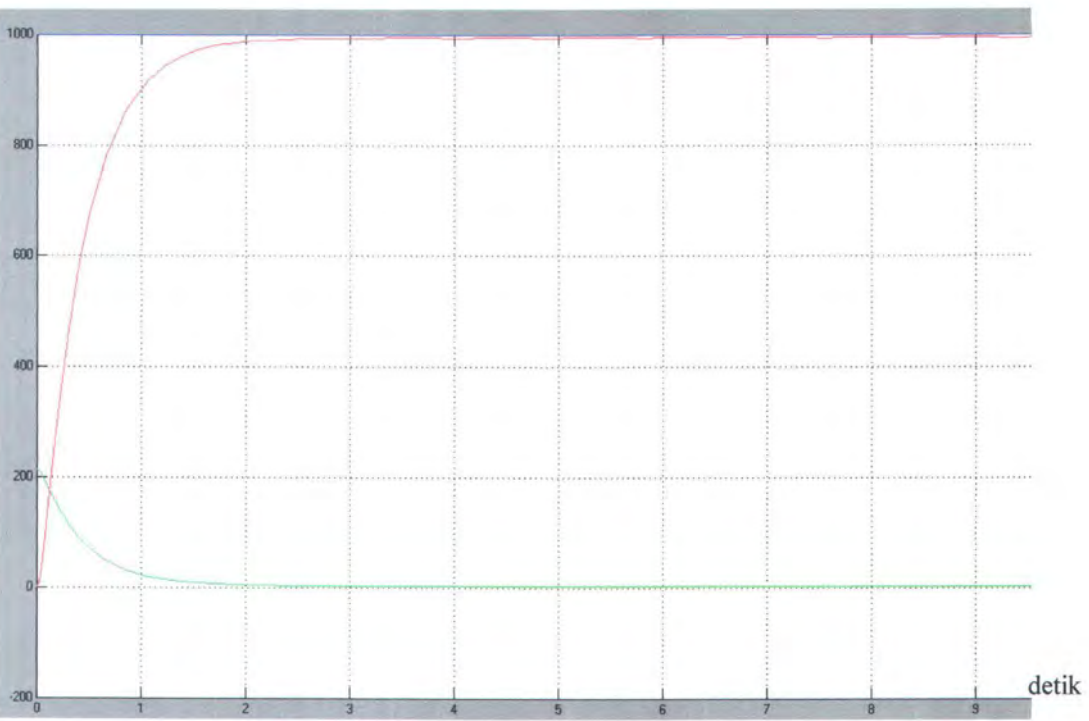
Step respon hasil simulasi sistem tanpa kontroler logika fuzzy, dengan setting point 750 rpm, dan konstan 500 Nm.



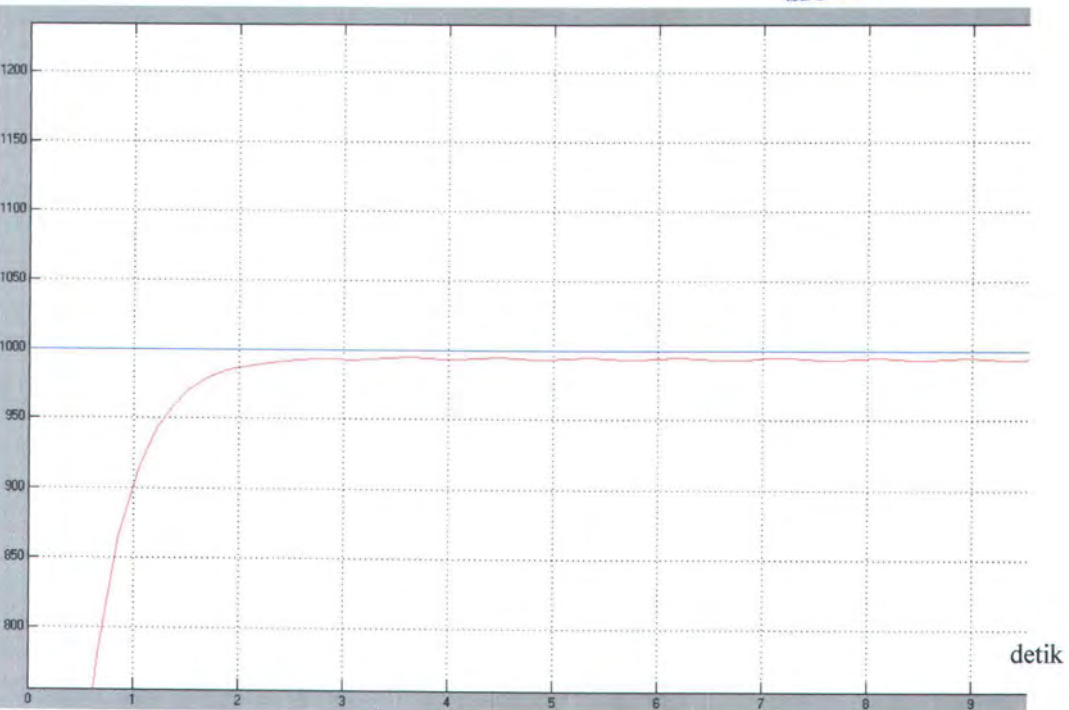
Step respon hasil simulasi sistem tanpa kontroler logika fuzzy, dengan setting point 750 rpm, dan konstan 100 Nm.



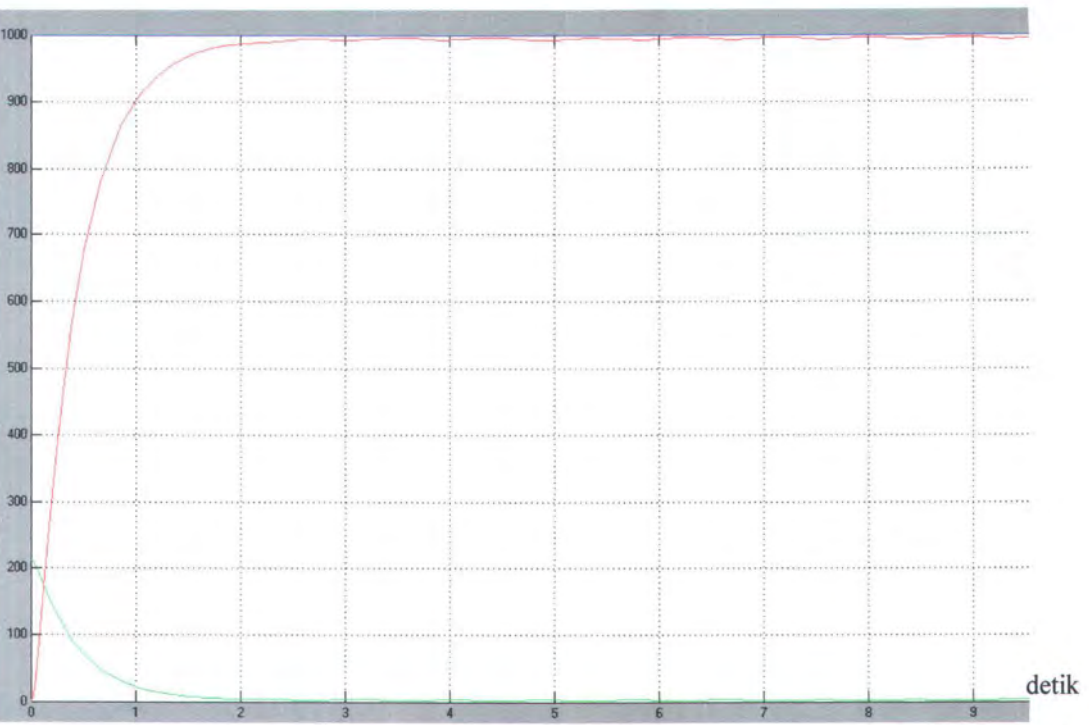
Step respon hasil simulasi sistem tanpa kontroler logika fuzzy, dengan setting point 750 rpm, dan konstan 1Nm.



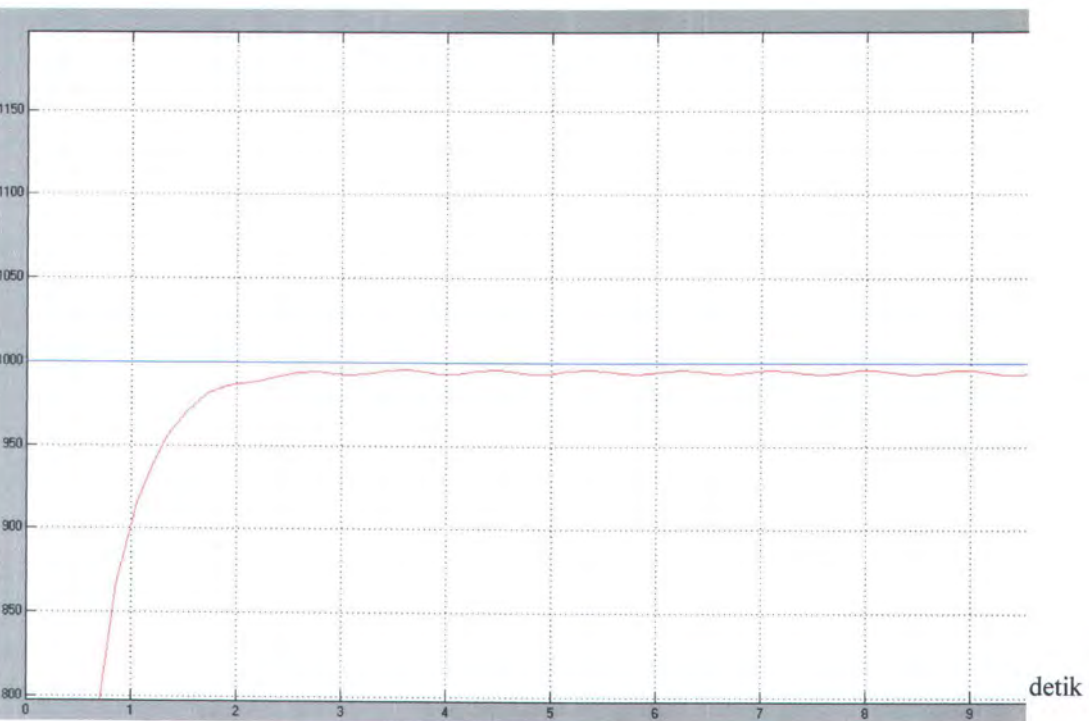
step respon hasil simulasi sistem tanpa kontroler logika fuzzy, dengan setting point 1000 rpm, dan sinusoidal dengan amplitudo 500 Nm.



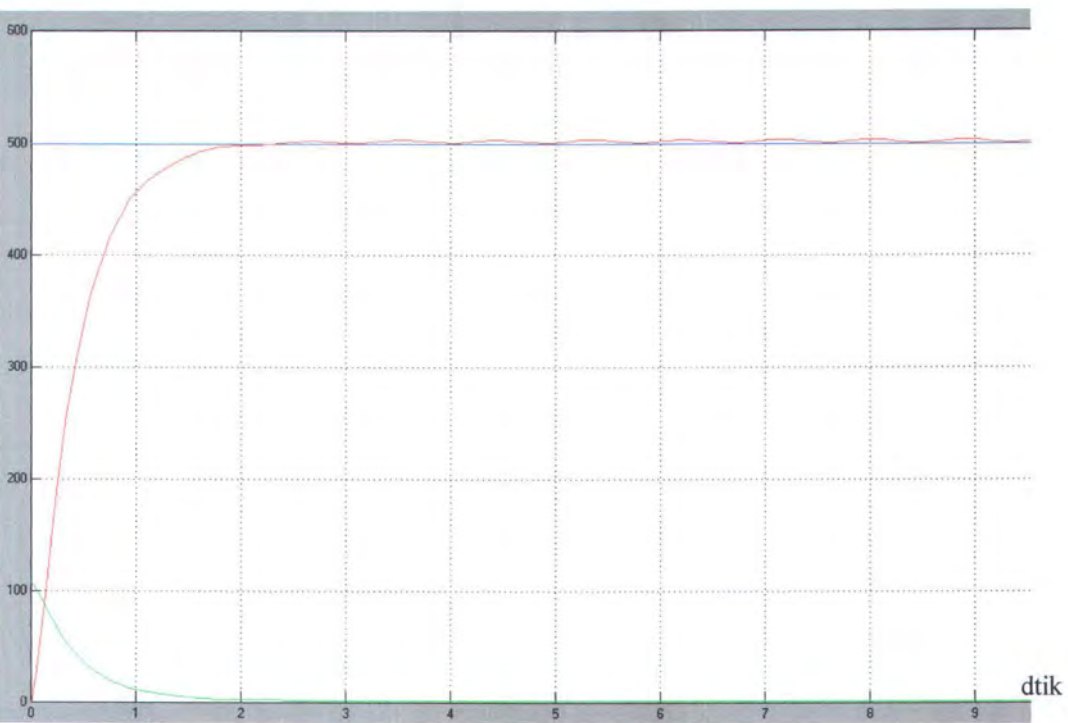
step respon hasil simulasi sistem tanpa kontroler logika fuzzy, dengan setting point 1000 rpm, dan sinusoidal dengan amplitudo 500 Nm, dan dizoom.



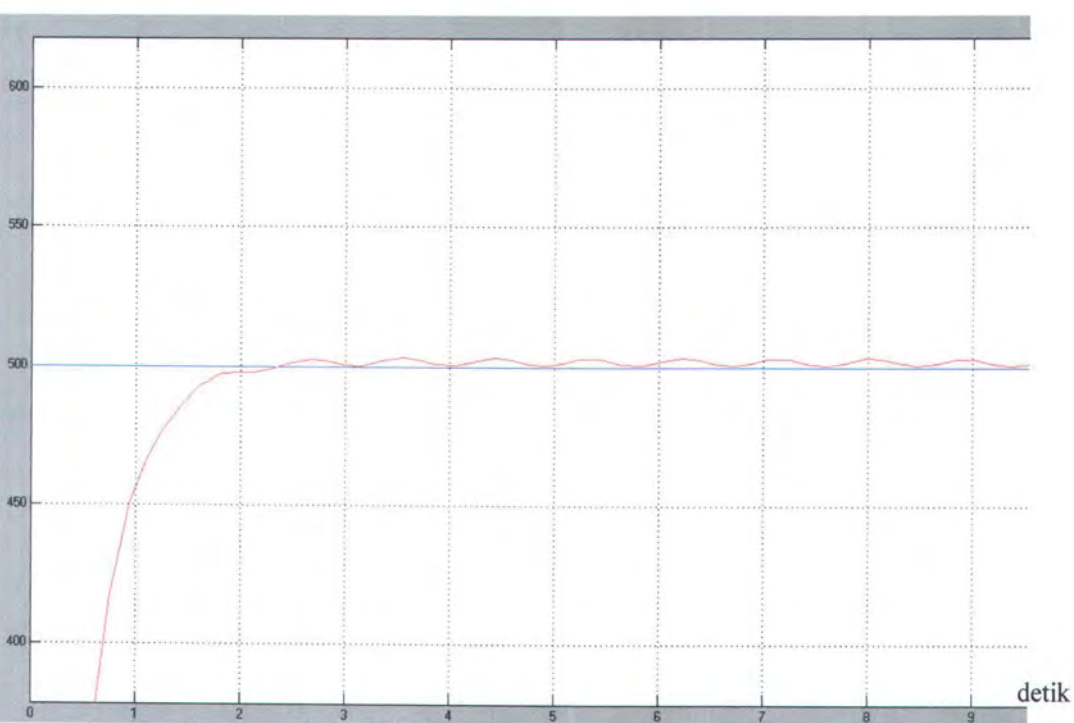
step respon hasil simulasi sistem tanpa kontroler logika fuzzy, dengan setting point 1000 rpm, dan sinusoidal dengan amplitudo 1000 Nm.



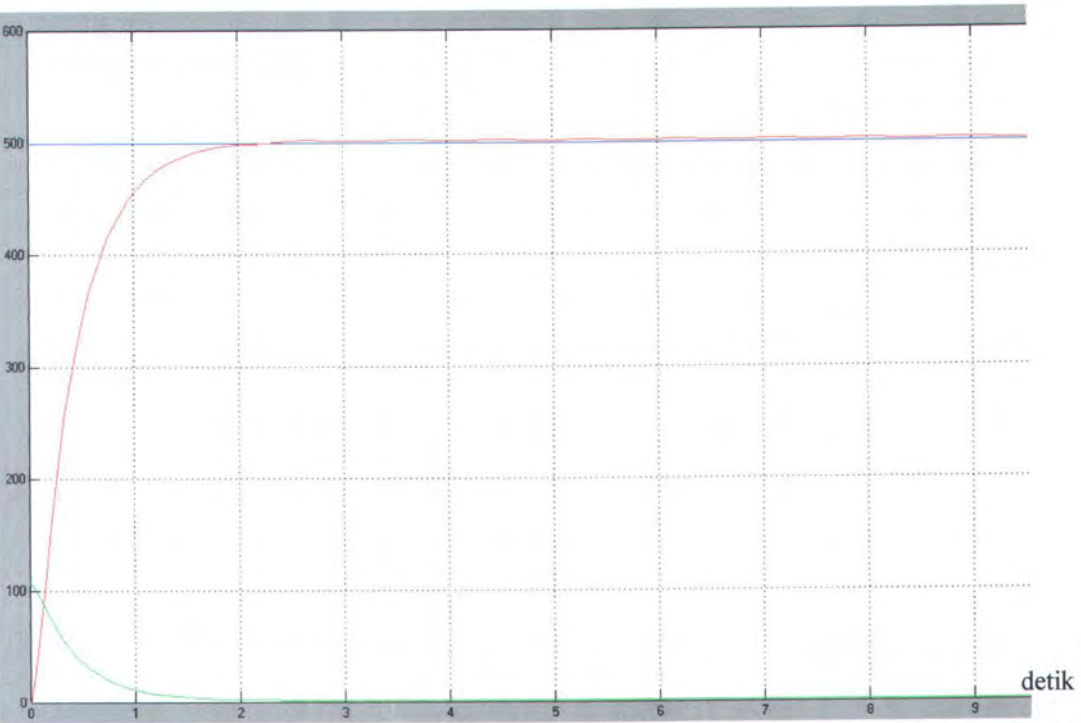
step respon hasil simulasi sistem tanpa kontroler logika fuzzy, dengan setting point 1000 rpm, dan sinusoidal dengan amplitudo 1000 Nm, dan dizoom.



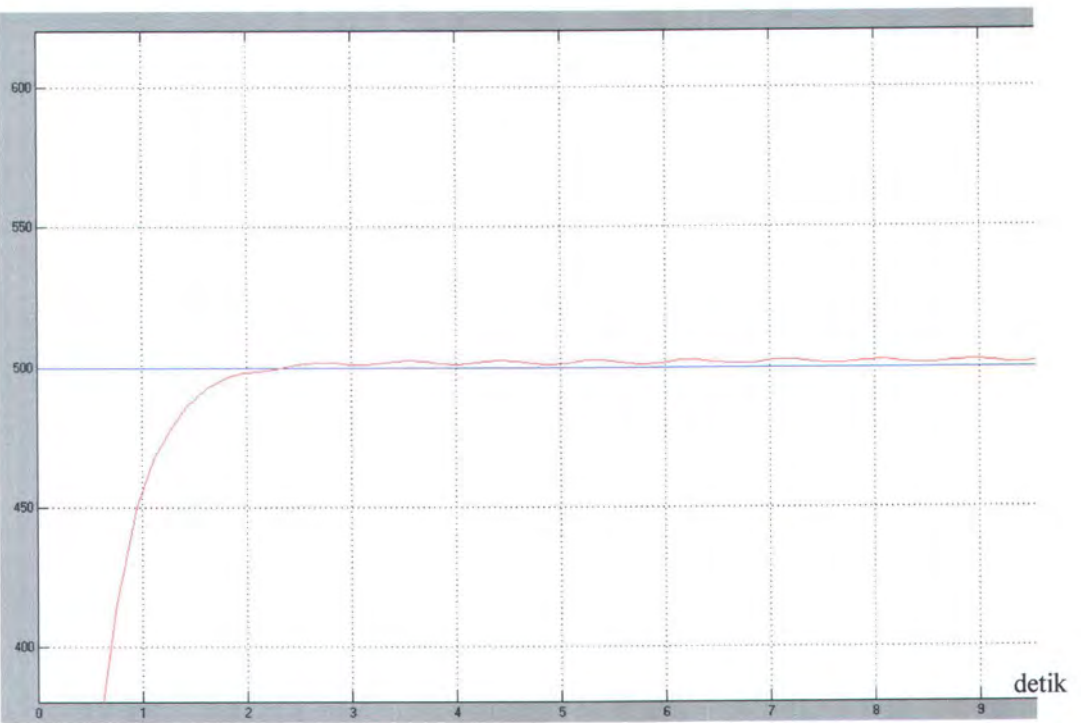
Step respon hasil simulasi sistem tanpa kontroler logika fuzzy, dengan setting point 500 rpm, dan sinusoidal dengan amplitudo 1000 Nm.



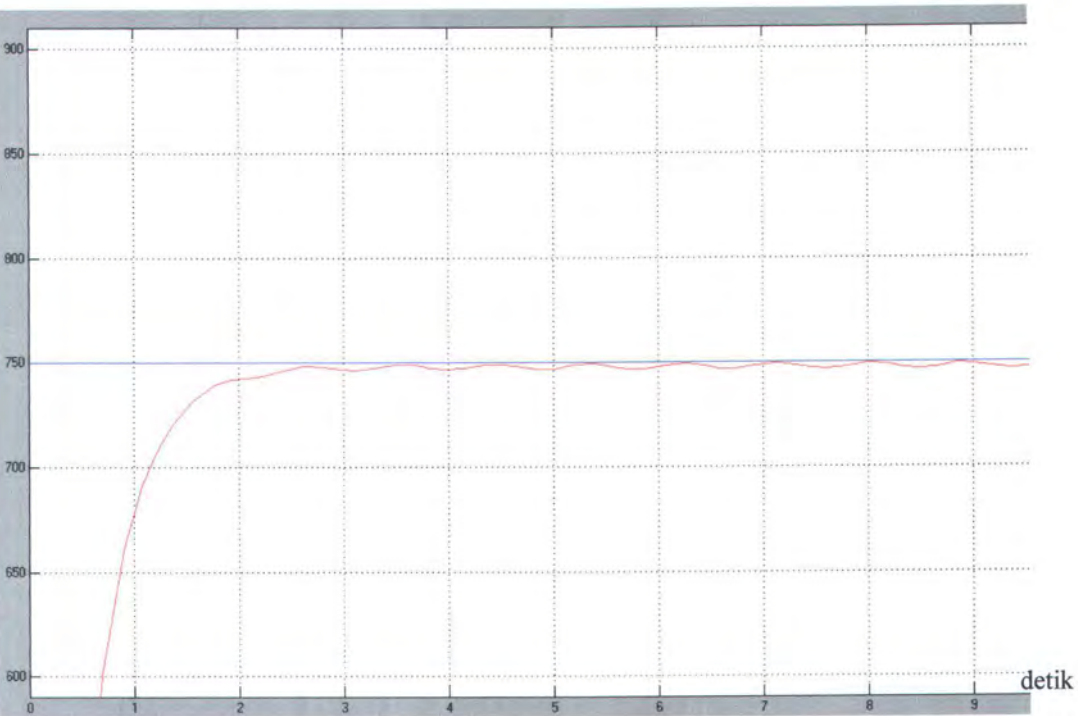
Step respon hasil simulasi sistem tanpa kontroler logika fuzzy, dengan setting point 500 rpm, dan sinusoidal dengan amplitudo 1000 Nm, dan dizoom.



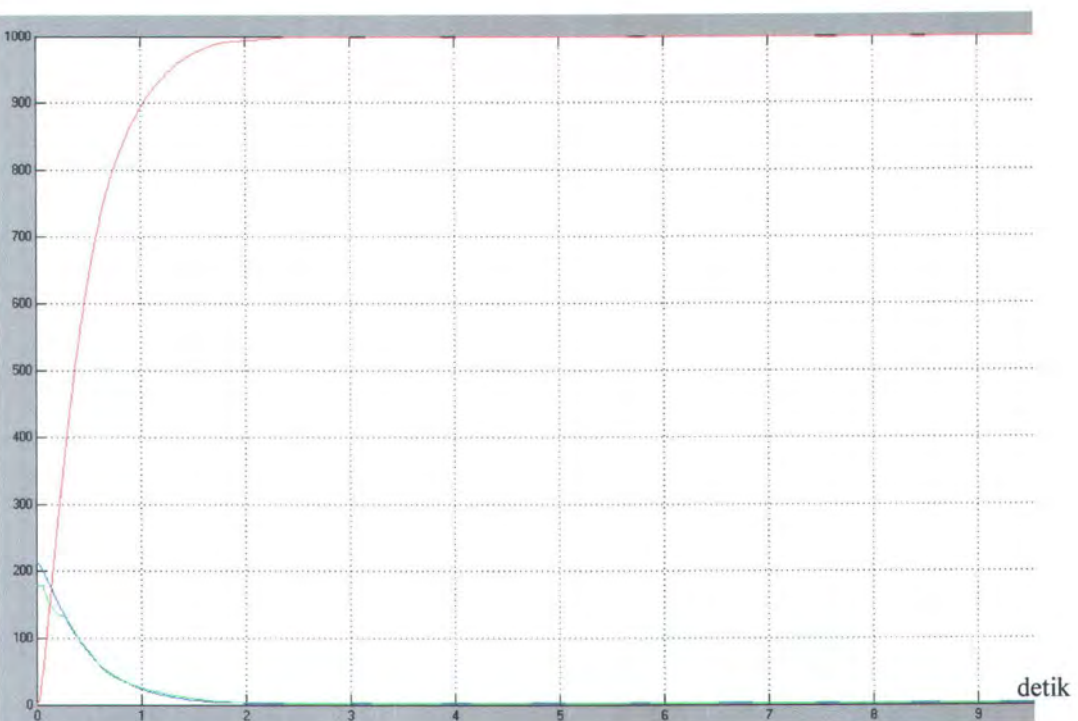
Step respon hasil simulasi sistem tanpa kontroler logika fuzzy, dengan setting point 500 rpm, dan sinusoidal dengan amplitudo 500 Nm.



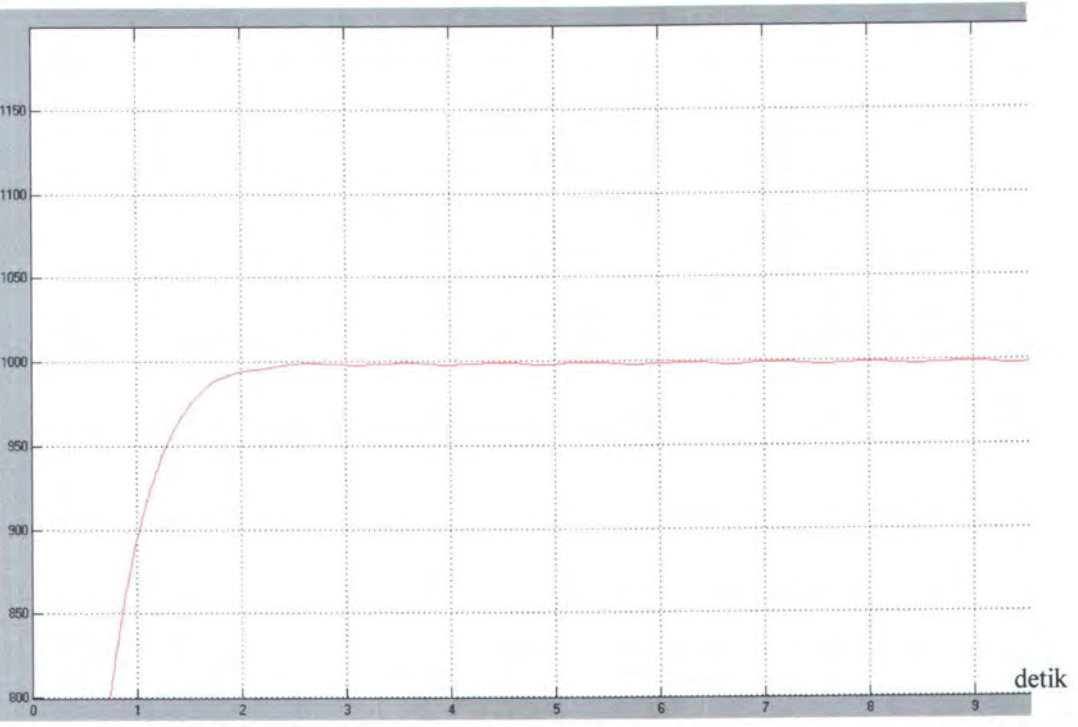
Step respon hasil simulasi sistem tanpa kontroler logika fuzzy, dengan setting point 500 rpm, dan sinusoidal dengan amplitudo 500 Nm, dan dizoom.



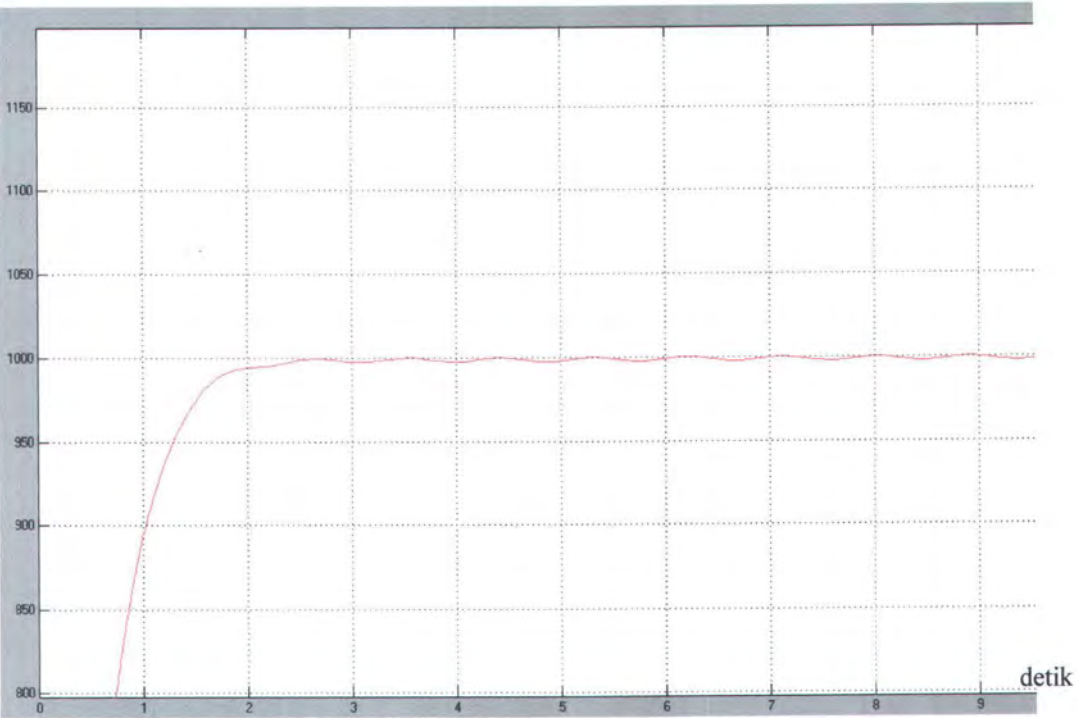
Step respon hasil simulasi sistem tanpa kontroler logika fuzzy, dengan setting point 750 rpm, dan sinusoidal dengan amplitudo 1000 Nm, dan dizoom.



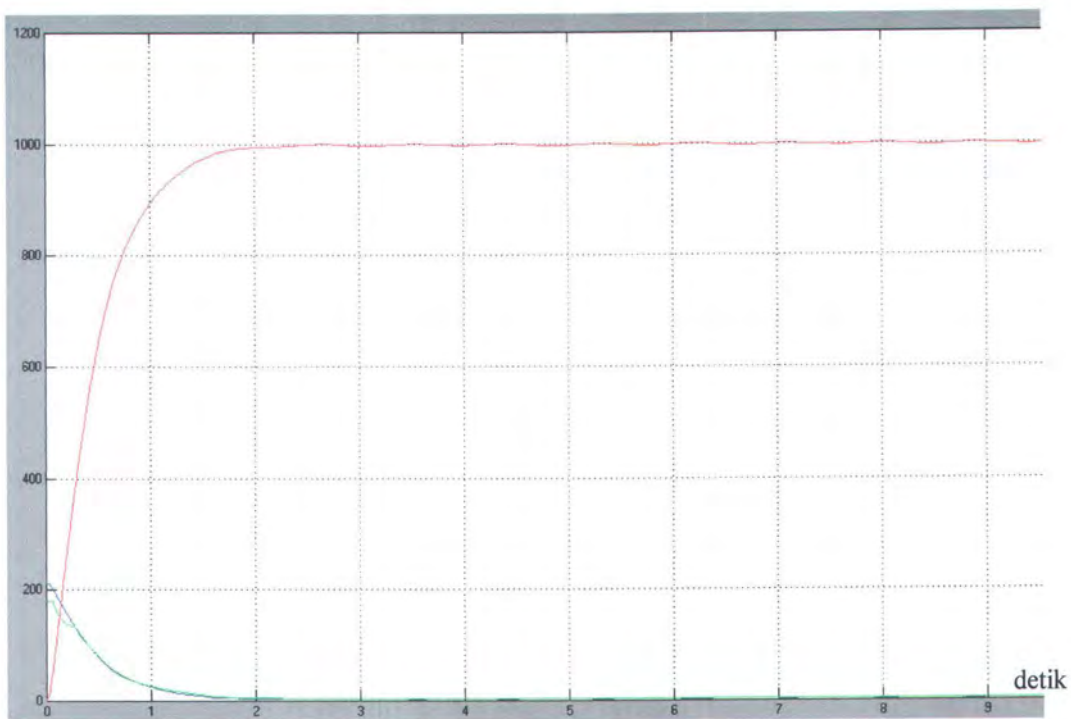
Step respon hasil simulasi sistem dengan kontroler logika fuzzy, dengan setting point 1000 rpm, dan sinusoidal dengan amplitudo 500 Nm.



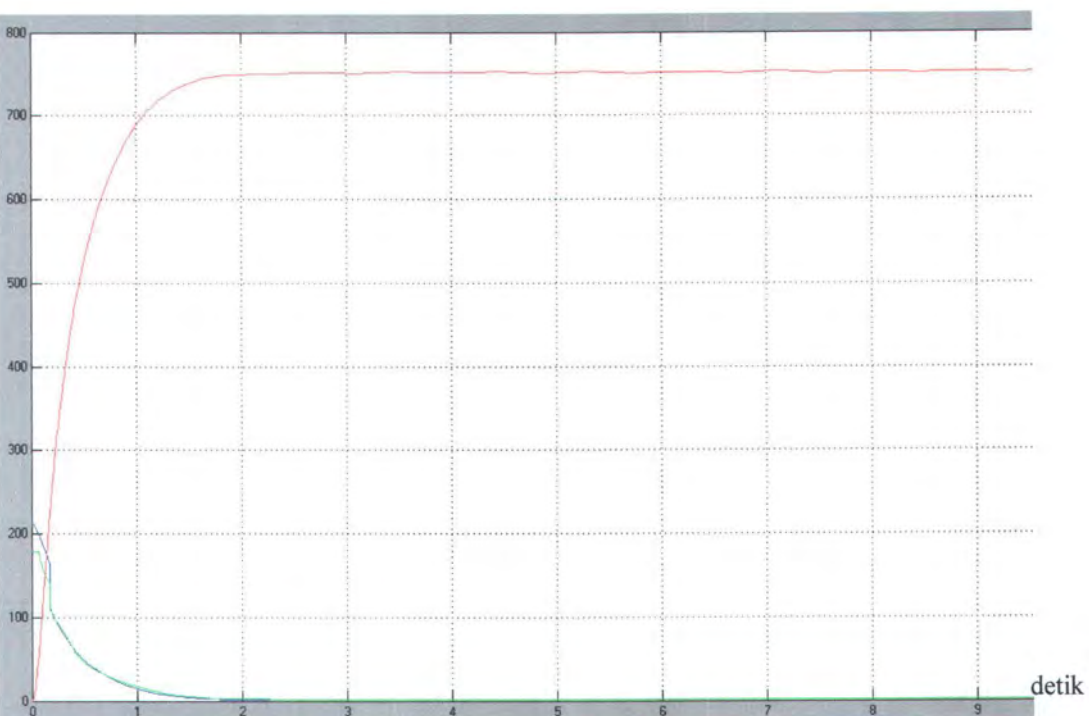
Step respon hasil simulasi sistem dengan kontroler logika fuzzy, dengan setting point 1000 rpm, beban sinusoidal dengan amplitudo 500 Nm, dan dizoom.



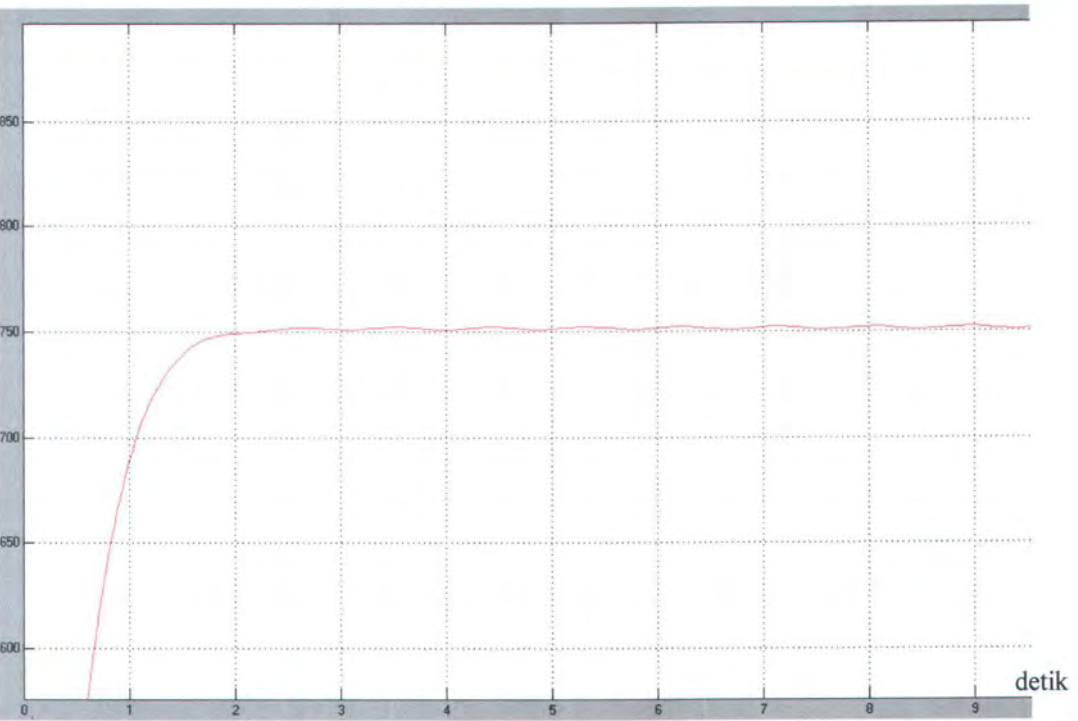
Step respon hasil simulasi sistem dengan kontroler logika fuzzy, dengan setting point 1000 rpm, dan beban sinusoidal dengan amplitudo 1000 Nm, dan dizoom.



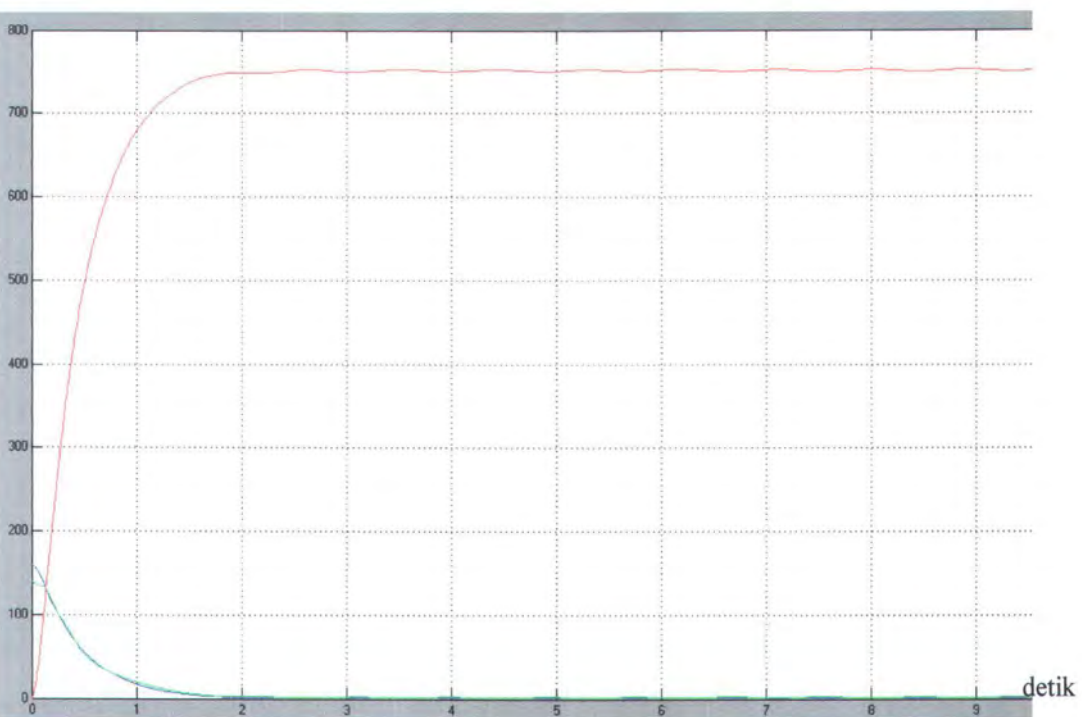
Step response plot of the system with a fuzzy logic controller, with a setpoint of 1000 rpm, and a sinusoidal load with an amplitude of 1000 Nm.



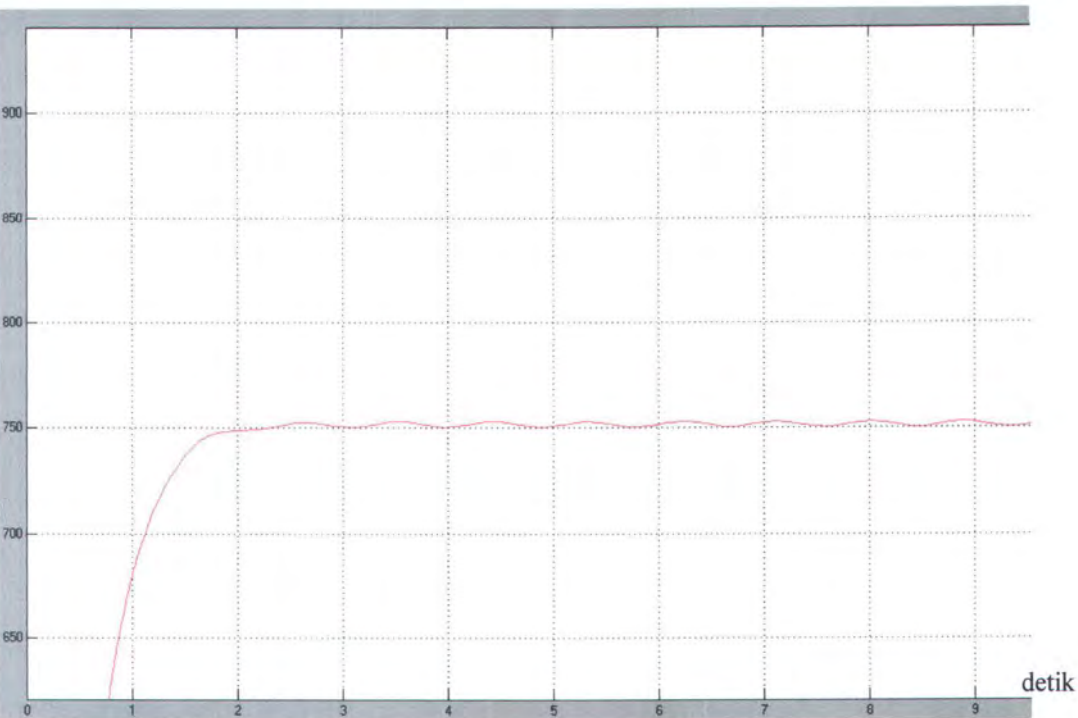
Step response plot of the system with a fuzzy logic controller, with a setpoint of 750 rpm, and a sinusoidal load with an amplitude of 500 Nm.



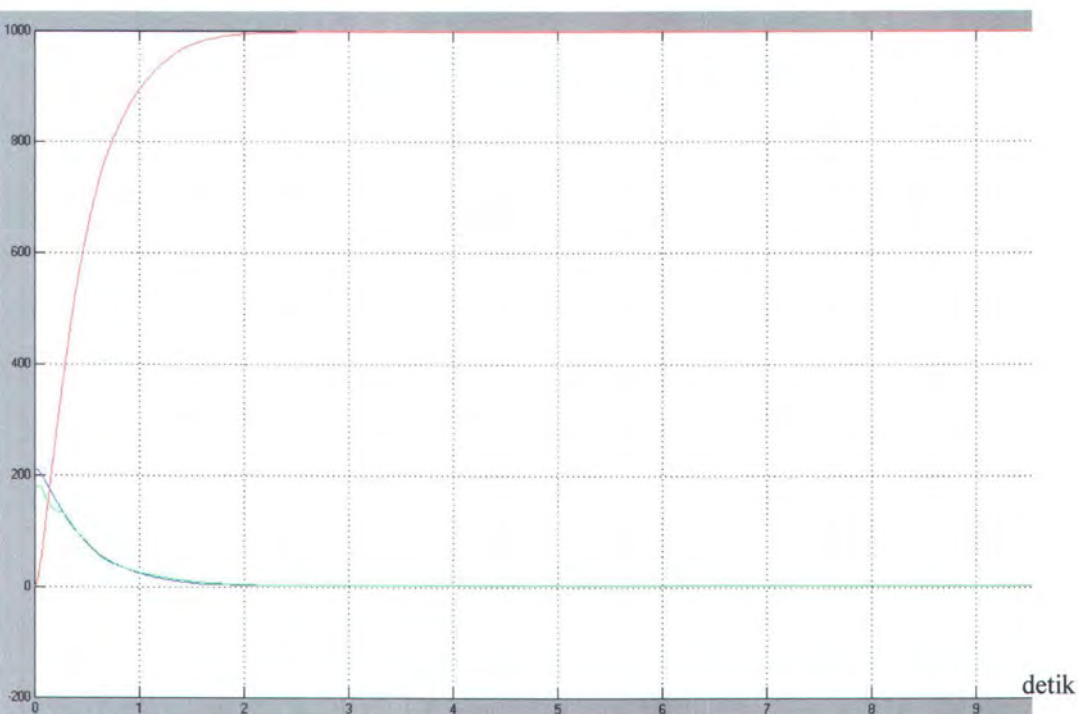
Step respon hasil simulasi sistem dengan kontroler logika fuzzy, dengan setting point 750 rpm, dan sinusoidal dengan amplitudo 500 Nm., dan dizoom.



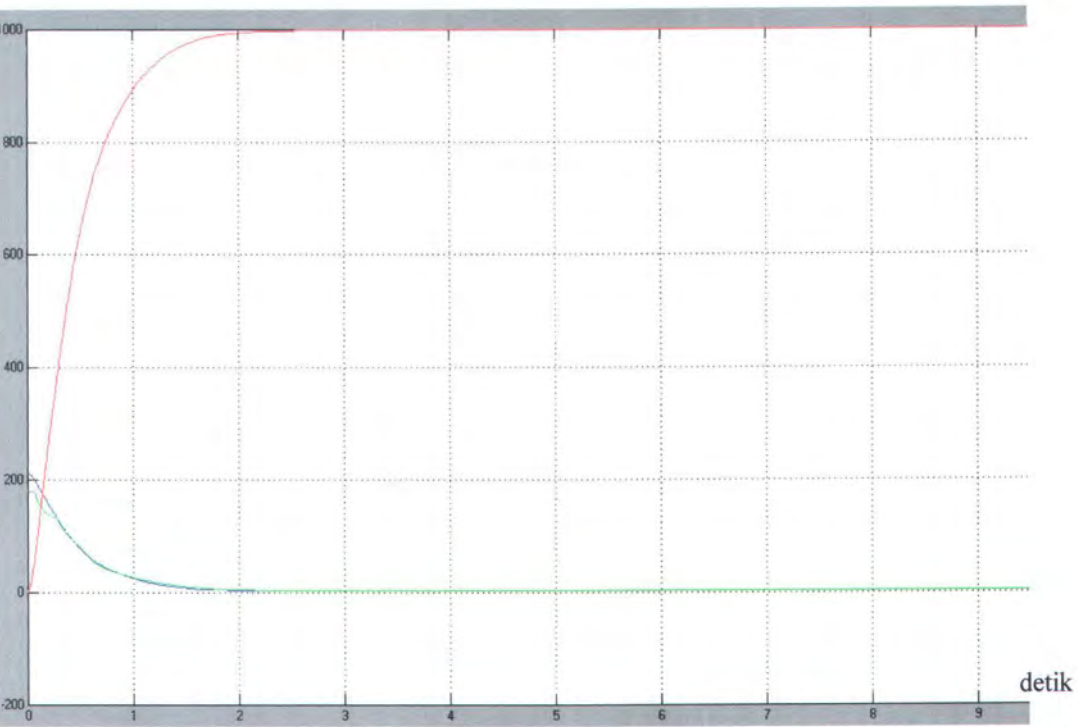
Step respon hasil simulasi sistem dengan kontroler logika fuzzy, dengan setting point 750 rpm, dan sinusoidal dengan amplitudo 1000 Nm.



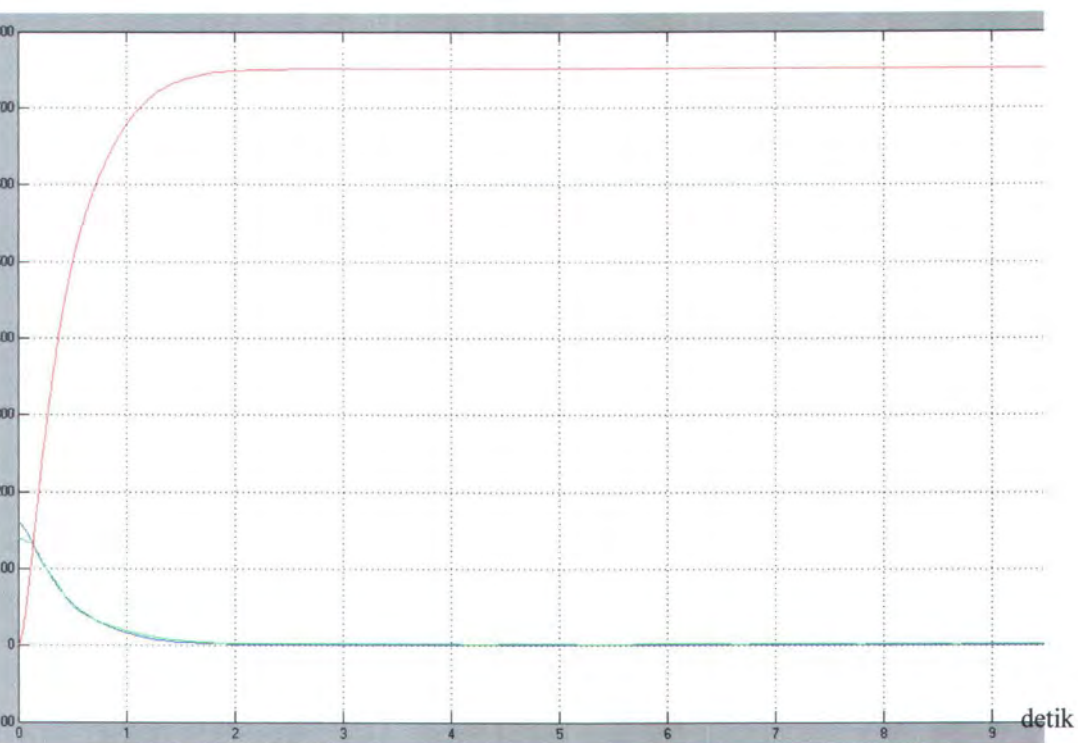
Step respon hasil simulasi sistem dengan kontroler logika fuzzy, dengan setting point 750 rpm, dan sinusoidal dengan amplitudo 1000 Nm, dan dizoom.



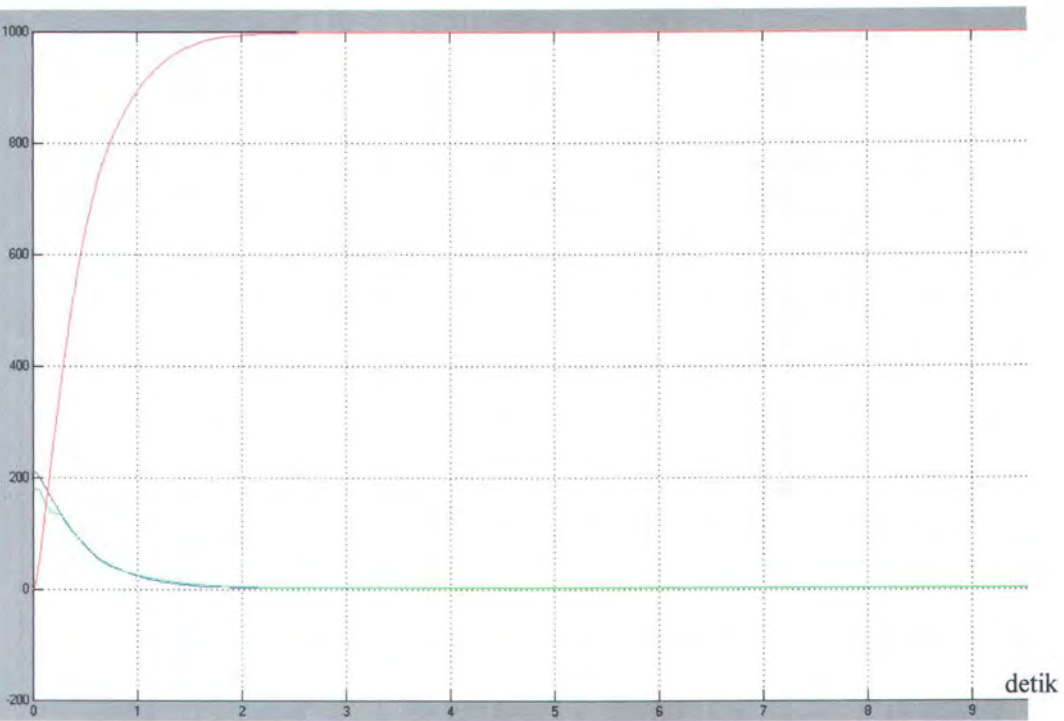
Step respon hasil simulasi sistem dengan kontroler logika fuzzy, dengan setting point 1000 rpm, dan konstan 1 Nm.



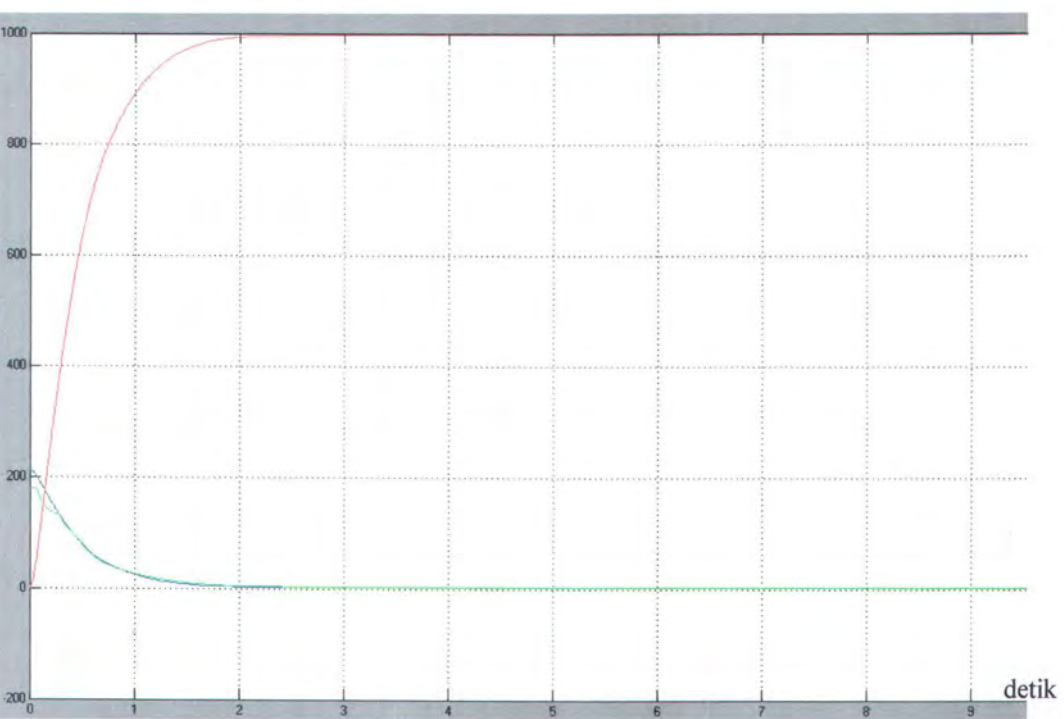
Step respon hasil simulasi sistem dengan kontroler logika fuzzy, dengan setting point 1000 rpm, dan konstan 100 Nm.



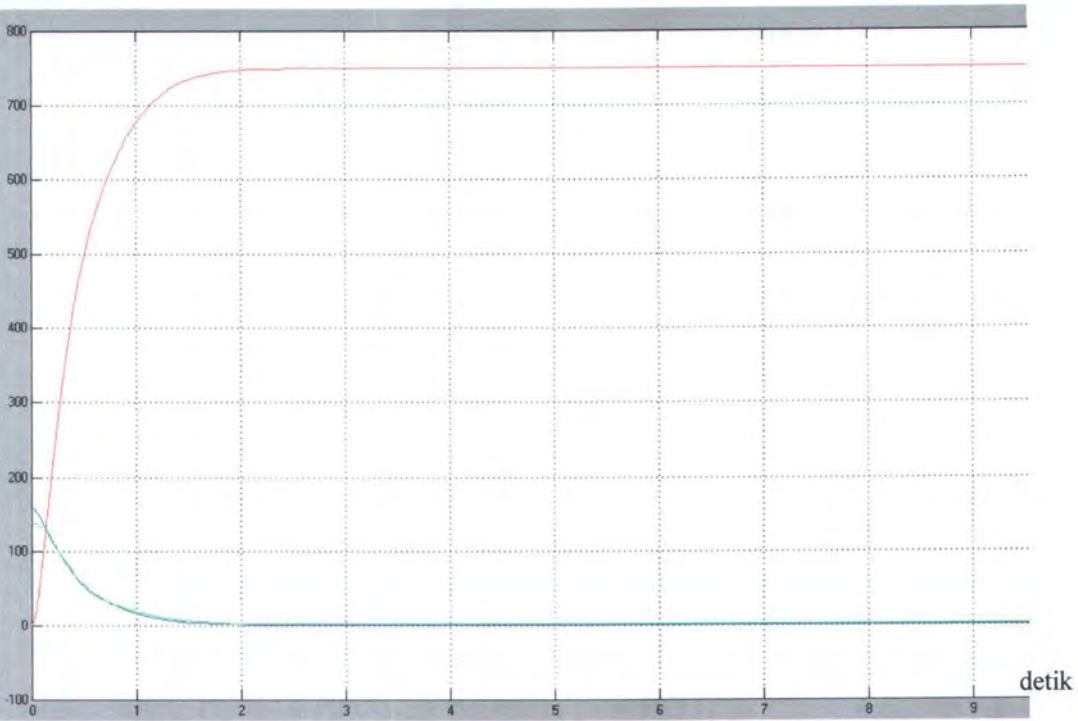
Step respon hasil simulasi sistem dengan kontroler logika fuzzy, dengan setting point 750 rpm, dan konstan 100 Nm.



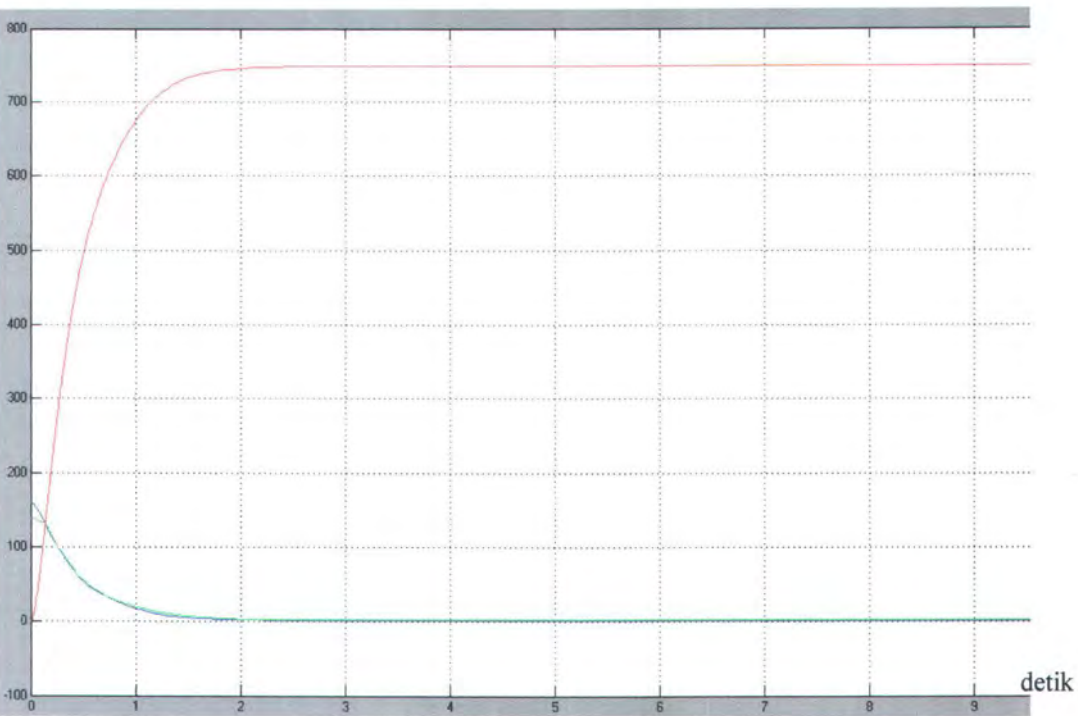
step respon hasil simulasi sistem dengan kontroler logika fuzzy, dengan setting point 1000 rpm, dan konstan 500 Nm.



step respon hasil simulasi sistem dengan kontroler logika fuzzy, dengan setting point 1000 rpm, dan konstan 1000 Nm.

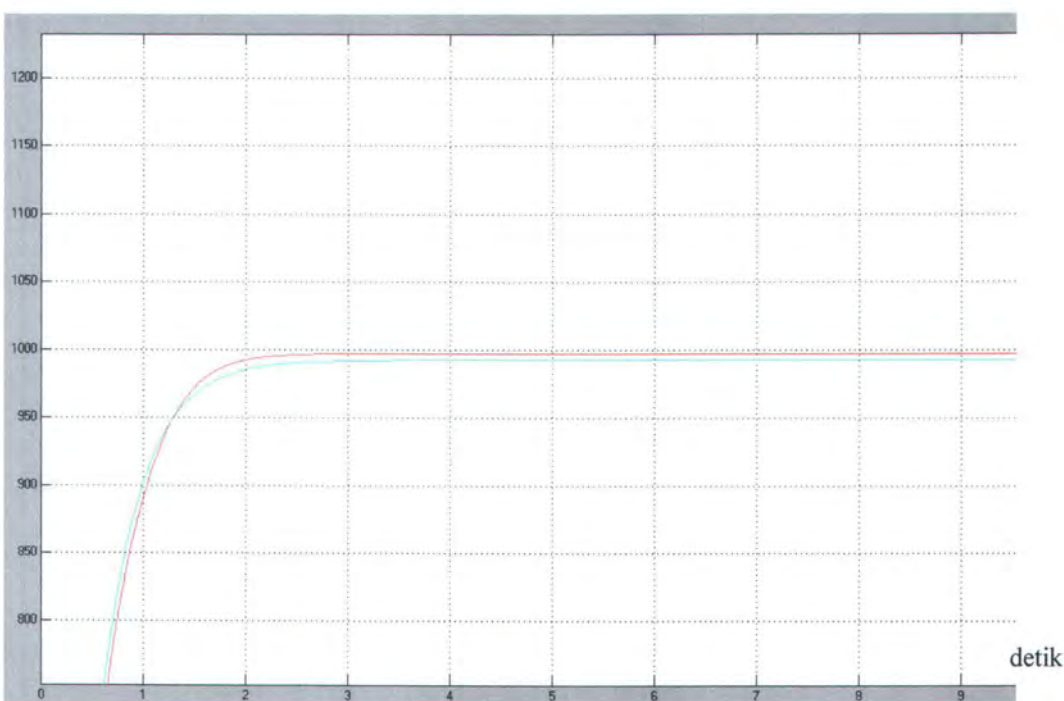
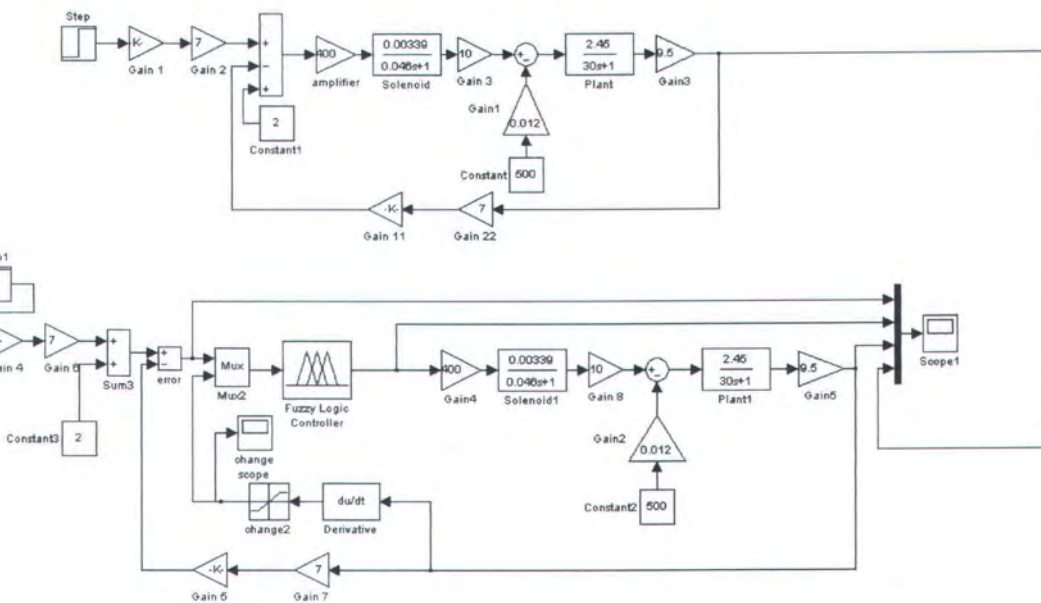


Step respon hasil simulasi sistem dengan kontroler logika fuzzy, dengan setting point 750 rpm, dan konstan 500 Nm.

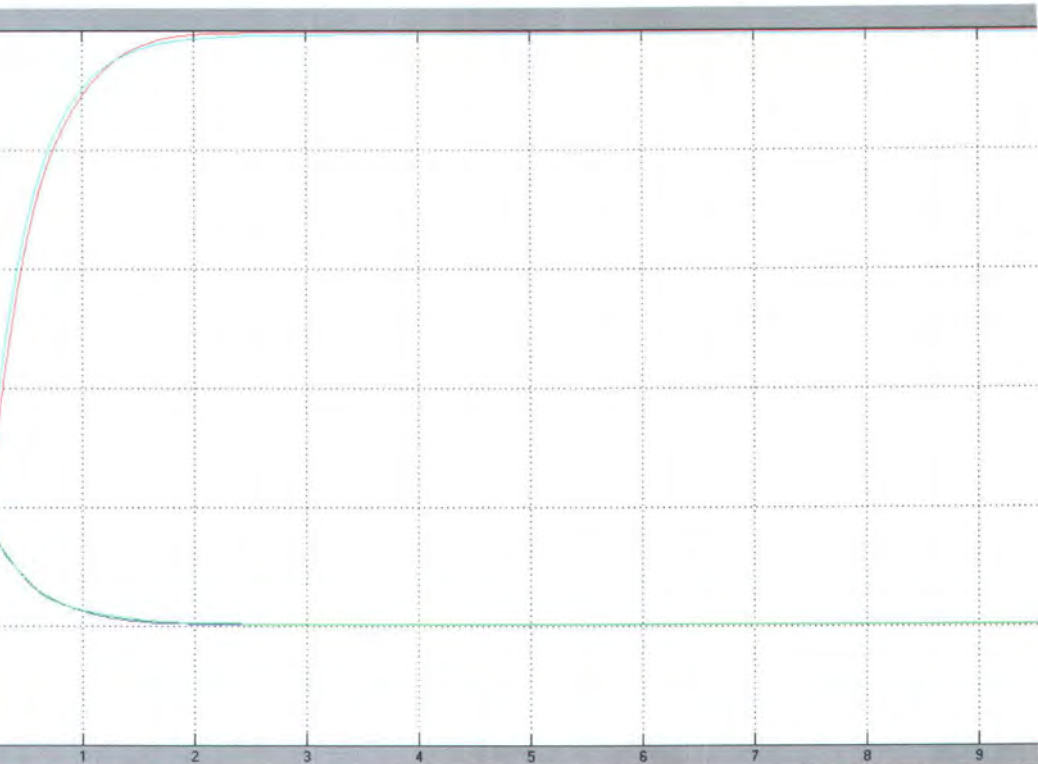


Step respon hasil simulasi sistem dengan kontroler logika fuzzy, dengan setting point 750 rpm, dan konstan 1000 Nm.

bandingkan sistem yang difuzzy dan sistem yang tidak difuzzy dengan beban konstan, jika ditampilkan dalam satu scop.

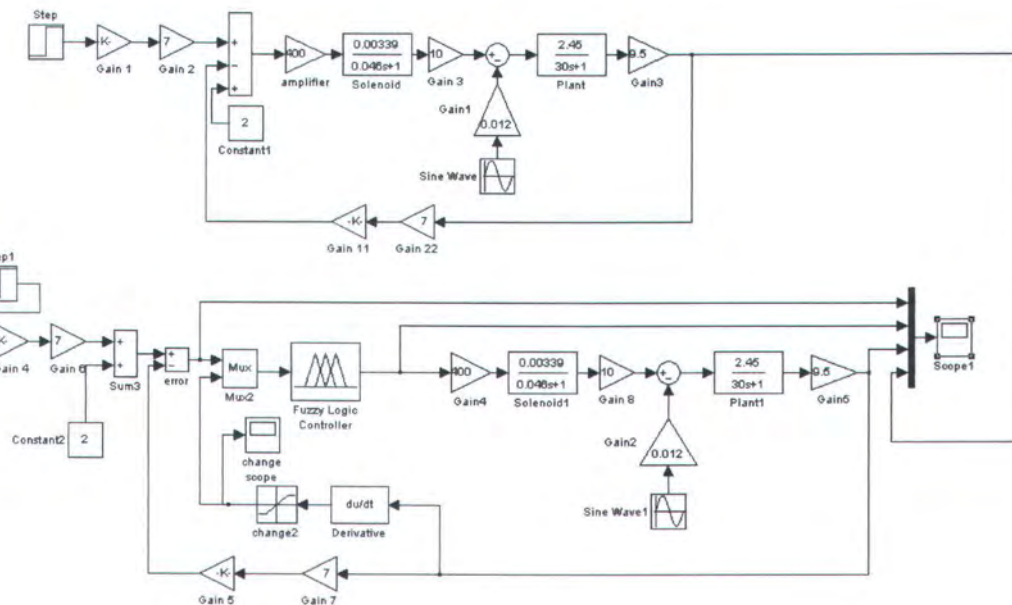


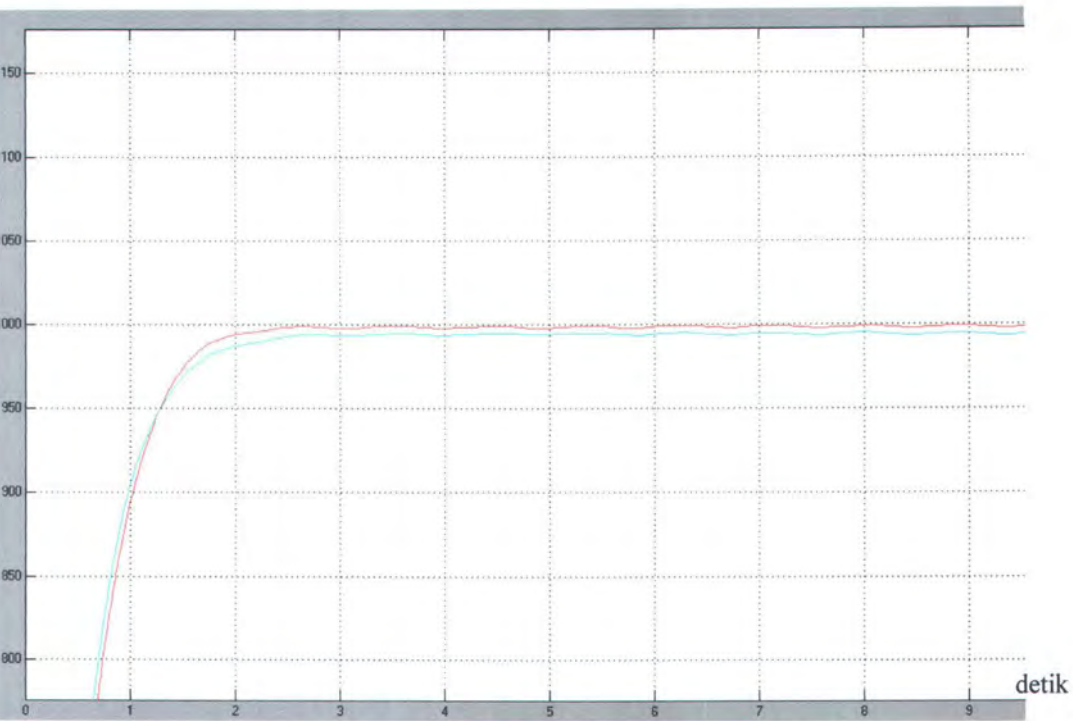
step respon hasil simulasi sistem dengan kontroler logika fuzzy (merah) dan yang tidak difuzzy, dengan setting point 1000 rpm, dan beban konstan 500 Nm, dan dizoom.



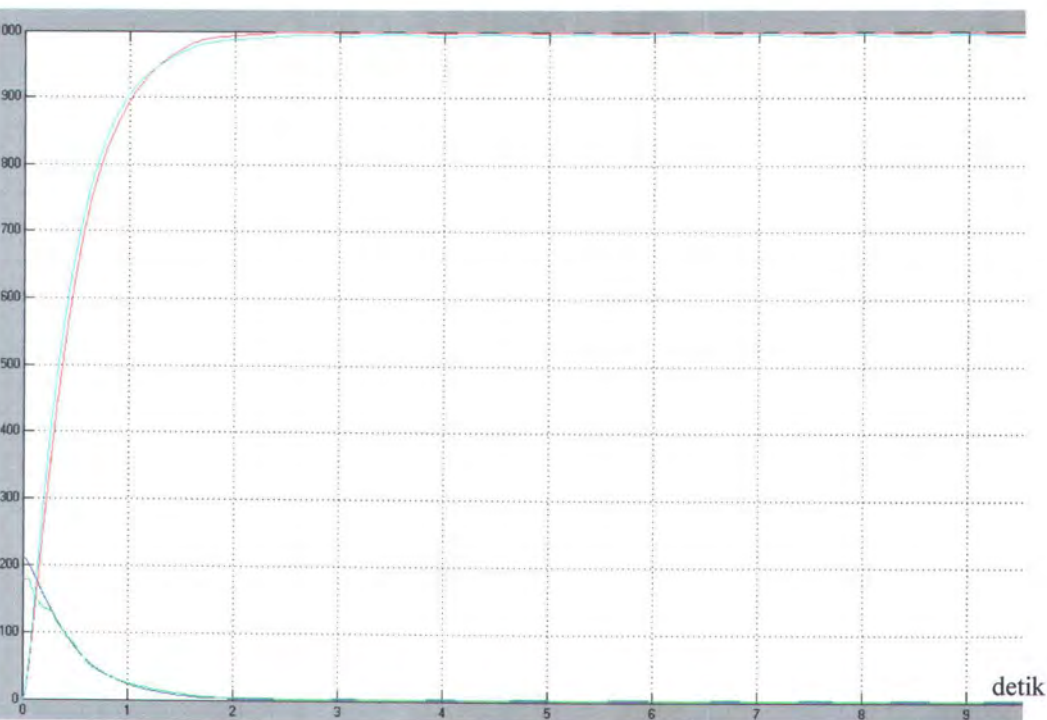
step respon hasil simulasi sistem dengan kontroler logika fuzzy (merah) dan yang tidak difuzzy, dengan setting point 1000 rpm, dan beban konstan 500 Nm.

bandingkan sistem yang difuzzy dan sistem yang tidak difuzzy dengan beban sinusoidal, jika ditampilkan dalam satu scop.

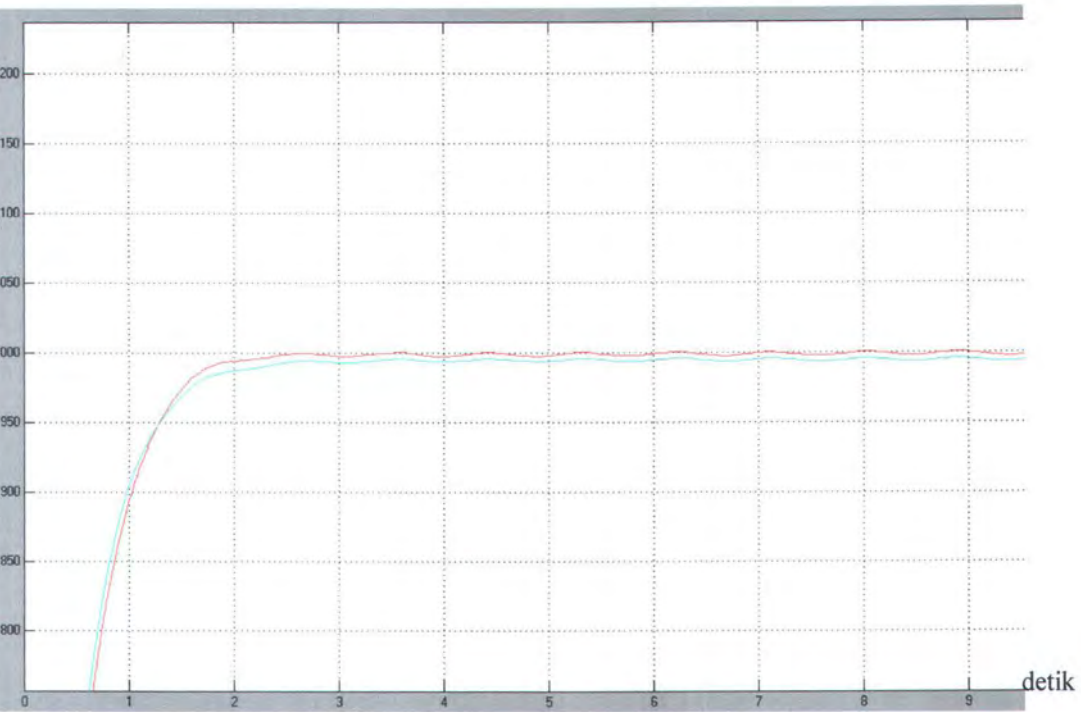




step respon hasil simulasi sistem dengan kontroler logika fuzzy (merah) dan yang tidak difuzzy, dengan setting point 1000 rpm, dan beban sinusoidal 500 Nm, dan dizoom.



step respon hasil simulasi sistem dengan kontroler logika fuzzy (merah) dan yang tidak difuzzy, dengan setting point 1000 rpm, dan beban sinusoidal dengan amplitudo 1000 Nm.



step respon hasil simulasi sistem dengan kontroler logika fuzzy (merah) dan yang tidak difuzzy, dengan setting point 1000 rpm, dan beban sinusoidal dengan amplitudo 1000 Nm, dan dizoom.