



TUGAS AKHIR - TE 091399

***TRACTION CONTROL PADA PARALLEL HYBRID
ELECTRIC VEHICLE (HEV) DENGAN MENGGUNAKAN
METODE KONTROL NEURO-FUZZY PREDIKTIF***

Bayu Prasetyo Wibowo
NRP 2210 100 139

Dosen Pembimbing
Ir. Rusdhianto Effendi A.K., MT.
Eka Iskandar, ST., MT.

JURUSAN TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2014



FINAL PROJECT - TE 091399

***TRACTION CONTROL IN PARALLEL HYBRID
ELECTRIC VEHICLE (HEV) USING NEURO-FUZZY
PREDICTIVE CONTROL***

Bayu Prasetyo Wibowo
NRP 2210 100 139

Advisor
Ir. Rusdhianto Effendi A.K., MT.
Eka Iskandar, ST., MT.

*ELECTRICAL ENGINEERING DEPARTMENT
Faculty of Industrial Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2014*

**TRACTION CONTROL PADA PARALLEL HYBRID
ELECTRIC VEHICLE (HEV) DENGAN
MENGUNAKAN METODE KONTROL NEURO-
FUZZY PREDIKTIF**

TUGAS AKHIR

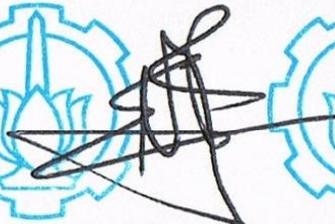
Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik
Pada
Bidang Studi Teknik Sistem Pengaturan
Jurusan Teknik Elektro
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember

Menyetujui :

Dosen Pembimbing I,

Dosen Pembimbing II,


Ir. Rusdhianto Effendi A.K., MT.
NIP. 19570424 198502 1 001


Eka Iskandar, ST., MT.
NIP. 19800528 200812 1 001

**SURABAYA
JANUARI, 2014**

TRACTION CONTROL PADA PARALLEL HYBRID ELECTRIC VEHICLE (HEV) MENGGUNAKAN METODE KONTROL *NEURO-FUZZY* PREDIKTIF

Nama : Bayu Prasetyo Wibowo
Pembimbing I : Ir. Rusdhianto Effendi AK,MT.
Pembimbing II : Eka Iskandar, ST., MT.

ABSTRAK

Penggunaan bahan bakar minyak untuk kendaraan bermotor yang semakin meningkat menyebabkan semakin menurunnya stok bahan bakar minyak di dunia. Hal ini menyebabkan terjadinya krisis energi di dunia. Selain itu, semakin banyaknya penggunaan bahan bakar minyak pada kendaraan bermotor menyebabkan semakin meningkatnya polusi udara. Polusi yang dihasilkan oleh kendaraan bermotor akan menyebabkan *global warming* sehingga suhu di atmosfer bumi akan meningkat.

Dalam upaya menanggulangi krisis energi dan bahaya *global warming* yang dihasilkan oleh kendaraan bermotor, maka diciptakan suatu kendaraan alternatif yang hemat energi dan ramah lingkungan yang disebut *Hybrid Electric Vehicle (HEV)*. HEV merupakan suatu kendaraan yang menggunakan *Internal Combustion Engine (ICE)* dan motor listrik sebagai motor penggerakannya. Salah satu permasalahan yang masih terjadi pada HEV yaitu akselerasi.

Kontroler *Neuro-Fuzzy* Prediktif digunakan untuk mengatur kecepatan motor DC ketika proses akselerasi HEV. Berdasarkan hasil pengujian didapatkan respon kecepatan HEV dapat mencapai model referensi yang diberikan pada $t = 0,051$ s sehingga dapat disimpulkan kontroler *Neuro-Fuzzy* Prediktif dapat mempercepat akselerasi pada HEV.

Kata kunci: akselerasi, HEV, motor DC, *Neuro-Fuzzy* Prediktif



Halaman ini sengaja dikosongkan

TRACTION CONTROL IN PARALLEL HYBRID ELECTRIC VEHICLE (HEV) USING NEURO-FUZZY PREDICTIVE CONTROL

Name : Bayu Prasetyo Wibowo
1st Advisor : Ir. Rusdhianto Effendi AK,MT.
2nd Advisor : Eka Iskandar, ST., MT.

ABSTRACT

The use of fuel for vehicles has increased causing the decreasing stocks of fuel oil in the world. This leads to the energy crisis in the world. In addition, the increasing use of fuel in vehicles used an increasing of air pollution. Pollution produced by vehicles so that global warming will cause temperatures to increase in the Earth's atmosphere.

In an effort to overcome the energy crisis and global warming dangers generated by vehicles, then created an alternative vehicle that is energy efficient and environmentally friendly called Hybrid Electric Vehicle (HEV). HEV is a vehicle that uses the Internal Combustion Engine (ICE) and an electric motor as the driving motor. One of the problems that still exist in the HEV is acceleration.

Therefore, designed a Neuro-Fuzzy Predictive controller to control the speed of a DC motor in order to obtain the desired performance. By applying Neuro-Fuzzy Predictive controller in HEV, the results obtained are in accordance with the desired performance.

Keywords: acceleration, HEV, DC motor, Neuro-Fuzzy Predictive



KATA PENGANTAR

Segala Puji bagi Allah SWT atas segala nikmat dan rahmat-Nya, sehingga penulis dapat menyelesaikan laporan Tugas Akhir ini dengan judul: ***“Traction Control Pada Parallel Hybrid Electric Vehicle (HEV) Dengan Menggunakan Metode Kontrol Neuro-Fuzzy Prediktif”***. Tugas Akhir ini disusun guna memenuhi persyaratan untuk mencapai Gelar Sarjana Teknik pada Bidang Studi Teknik Sistem Pengaturan, Jurusan Teknik Elektro Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember (ITS) Surabaya.

Pada kesempatan yang berbahagia ini penulis mengucapkan terima kasih kepada semua pihak yang telah banyak membantu baik secara langsung maupun tidak langsung, hingga penelitian Tugas Akhir ini dapat diselesaikan. Khususnya kepada Ibu, Bapak, dan Mbak Risma. Terimakasih atas doa dan dukungannya, sehingga penulis dapat menyelesaikan Tugas Akhir ini. Terima kasih juga kepada Bapak Rusdhianto Effendi dan Bapak Eka Iskandar selaku dosen pembimbing penulis yang terus memberikan bimbingan terbaiknya kepada penulis. Terima kasih kepada Widhayaka, Ilmiyah Elrosa, dan Anisa E yang telah banyak membantu penulis, serta teman-teman e50 yang terus memberikan dukungan kepada penulis.

Penulis menyadari masih banyak kekurangan dalam pembuatan laporan Tugas Akhir ini, mohon maaf atas segala kekurangan pada Tugas Akhir ini. Semoga Tugas Akhir ini dapat memberikan manfaat dalam pengembangan keilmuan ke depannya.

Surabaya, Januari 2014

Penulis



DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN JUDUL.....	iii
PERNYATAAN KEASLIAN TUGAS AKHIR.....	v
HALAMAN PENGESAHAN.....	vii
ABSTRAK.....	ix
<i>ABSTRACT</i>	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR.....	xix
DAFTAR TABEL.....	xxiii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang Masalah.....	1
1.2 Rumusan Masalah.....	1
1.3 Tujuan.....	2
1.4 Batasan Masalah.....	2
1.5 Sistematika Penulisan.....	2
1.6 Relevansi.....	3
BAB 2 DASAR TEORI.....	5
2.1 <i>Hybrid Electric Vehicle (HEV)</i>	5
2.2 Simulator HEV.....	7
2.2.1 <i>Internal Combustion Engine (ICE) 2 Tak</i>	7
2.2.2 Motor Listrik DC.....	8
2.2.3 Rem Elektromagnetik Arus Eddy.....	12
2.2.4 Perbandingan <i>Gear</i>	14
2.2.5 <i>Tachogenerator</i>	17
2.3 Sistem Minimum Arduino Uno R3.....	18
2.4 Identifikasi Sistem.....	19
2.4.1 Metode Identifikasi.....	20
2.4.2 Validasi Sistem.....	22
2.5 Logika <i>Fuzzy</i>	23
2.5.1 Himpunan <i>Fuzzy</i>	23
2.5.2 Fungsi Keanggotaan.....	23
2.6 Kontroler <i>Fuzzy</i>	25
2.6.1 Fuzzifikasi.....	26
2.6.2 Basis Aturan.....	26
2.6.3 Mekanisme Inferensi.....	26
2.6.4 Defuzzifikasi.....	27

2.7 Jaringan Syaraf Tiruan	28
2.8 Kontroler <i>Neuro-Fuzzy</i> (NF)	29
2.8.1 FMN Tipe I	30
2.8.2 FMN Tipe II	31
2.8.3 FMN Tipe III	33
BAB 3 PERANCANGAN SISTEM	37
3.1 Diagram Blok Sistem	37
3.2 Perancangan Perangkat Keras	37
3.2.1 Perancangan Mekanik	38
3.2.2 Perancangan Elektronik	46
3.3 Perancangan Perangkat Lunak	51
3.3.1 Matlab	51
3.4 Pengujian <i>Open Loop</i> Kecepatan Motor DC	52
3.5 Pengujian Sensor	53
3.5.1 Pengujian Sensor <i>Tachogenerator</i>	54
3.5.2 Pengujian Sensor Potensiometer	55
3.6 Proses Identifikasi dan Permodelan Sistem	56
3.6.1 Identifikasi ICE	56
3.6.2 Identifikasi Motor DC	65
3.6.3 Model Referensi	69
3.7 Perancangan Kontroler <i>Neuro-Fuzzy</i> Prediktif	69
3.7.1 Perancangan Tahap <i>Forward Propagation Neuro-Fuzzy</i>	71
3.7.2 Perancangan Tahap <i>Back Propagation Neuro-Fuzzy</i>	73
3.7.3 Model Reference Based Invers Model	76
3.7.4 Perancangan <i>Observer</i>	78
3.7.5 Perancangan Prediktor	81
BAB 4 PENGUJIAN DAN ANALISIS SISTEM	85
4.1 Gambaran Umum Pengujian Sistem	85
4.2 Simulasi Sistem	85
4.2.1 Pengujian ICE Berbeban Nominal Tanpa Bantuan Motor DC	85
4.2.2 Pengujian Kontroler <i>Neuro-Fuzzy</i> Terhadap Sinyal Target	86
4.2.3 Pengujian ICE Berbeban Nominal Dengan Bantuan Motor DC Menggunakan Kontroler <i>Neuro-Fuzzy</i> Prediktif	89
4.3 Implementasi Sistem	94

4.3.1 Pengujian ICE Berbeban Nominal Tanpa Bantuan Motor DC	94
4.3.2 Pengujian Menggunakan Kontroler <i>Neuro-Fuzzy</i> Tanpa Model Prediksi	94
4.3.3 Pengujian Menggunakan Kontroler <i>Neuro-Fuzzy</i> Menggunakan Model Prediksi	95
BAB 5 PENUTUP	97
5.1 Kesimpulan	97
5.2 Saran.....	97
DAFTAR PUSTAKA	99
RIWAYAT PENULIS	101



Halaman ini sengaja dikosongkan

DAFTAR GAMBAR

Gambar 2.1 Konfigurasi HEV	5
Gambar 2.2 Siklus Kerja ICE 2 Tak	8
Gambar 2.3 Kaidah Tangan Kiri.....	9
Gambar 2.4 Konstruksi Motor DC dengan Sikat (<i>Brushed</i>).....	10
Gambar 2.5 Rangkaian Ekuivalen Motor DC <i>Shunt</i> Penguat Sendiri	10
Gambar 2.6 Konfigurasi Motor DC	11
Gambar 2.7 Rem Elektromagnetik.....	12
Gambar 2.8 Gaya Pengereman yang Ditimbulkan Arus Eddy.....	13
Gambar 2.9 Bentuk Fisik <i>Gear</i>	14
Gambar 2.10 Konfigurasi Beda Poros.....	15
Gambar 2.11 Konfigurasi Satu Poros.....	16
Gambar 2.12 <i>Pitch Circle</i>	16
Gambar 2.13 <i>Gear</i> Dengan <i>Pitch Diameter 1"</i> dan <i>Diametral Pitch 2017</i>	
Gambar 2.14 <i>Tachogenerator</i>	17
Gambar 2.15 Rangkaian <i>Tachogenerator</i> DC	18
Gambar 2.16 Bentuk Fisik Sistem Minimum Arduino Uno R3.....	18
Gambar 2.17 Respon Step Sistem dengan waktu T_u , T_N , t_1 , dan t_2	20
Gambar 2.18 Diagram Blok Kontroler <i>Fuzzy</i>	25
Gambar 2.19 Jaringan Syaraf Tiruan Perseptron Lapis Jamak	29
Gambar 2.20 Struktur Jaringan FMN Tipe I.....	31
Gambar 2.21 Struktur Jaringan FMN Tipe II	32
Gambar 2.22 Struktur Jaringan FMN Tipe III.....	34
Gambar 3.1 Diagram Blok Sistem	37
Gambar 3.2 Diagram Perangkat Keras Simulator HEV	38
Gambar 3.3 Bentuk Fisik Simulator HEV	39
Gambar 3.4 ICE 2 Tak	40
Gambar 3.5 Motor DC	40
Gambar 3.6 Transmisi <i>Gear</i>	42
Gambar 3.7 Bentuk Fisik Sensor Posisi Pada Pedal Gas.....	44
Gambar 3.8 Rem Elektromagnetik.....	44
Gambar 3.9 Bentuk Fisik Rem Elektromagnetik	45
Gambar 3.10 Bentuk Fisik <i>Tachogenerator</i>	46
Gambar 3.11 Bentuk Fisik Rangkaian <i>Driver</i> Motor DC.....	46
Gambar 3.12 Rangkaian <i>Driver</i> Motor DC	47
Gambar 3.13 Bentuk Fisik Sistem Minimum Arduino Uno R3.....	48
Gambar 3.14 Skema Rangkaian Pembagi Arus-Tegangan.....	49
Gambar 3.15 Bentuk Fisik Rangkaian Pembagi Arus-Tegangan.....	49

Gambar 3.16 Skema Rangkaian Penyearah Setengah Gelombang.....	50
Gambar 3.17 Skema Rangkaian Penyearah Gelombang Penuh	50
Gambar 3.18 Bentuk Fisik Rangkaian Penyearah Gelombang Penuh ...	51
Gambar 3.19 Blok Simulink Perancangan Kontroler <i>Neuro Fuzzy</i> Prediktif Pada Simulator HEV	52
Gambar 3.20 Blok Simulink Implementasi Kontroler <i>Neuro Fuzzy</i> Prediktif Pada Simulator HEV	52
Gambar 3.21 Hubungan Antara Tegangan <i>Input</i> dan Kecepatan Putar	53
Gambar 3.22 Hubungan Antara Kecepatan Putar Poros Motor DC dengan Tegangan <i>Tachogenerator</i>	54
Gambar 3.23 Hubungan Pijakan Pedal terhadap Tegangan Potensiometer	55
Gambar 3.24 Respon Kecepatan Putar ICE	56
Gambar 3.25 Respon Kecepatan Putar ICE dengan Beban Minimal.....	58
Gambar 3.26 Hasil Identifikasi dan Permodelan ICE dengan Beban Minimal.....	59
Gambar 3.27 Respon Kecepatan Motor Bakar dengan Beban Nominal	60
Gambar 3.28 Hasil Identifikasi dan Permodelan ICE dengan Beban Nominal	62
Gambar 3.29 Respon Kecepatan Putar ICE dengan Beban Maksimal ..	64
Gambar 3.30 Hasil Identifikasi dan Permodelan ICE dengan Beban Maksimal	65
Gambar 3.31 Respon Kecepatan Motor DC	66
Gambar 3.32 Hasil Identifikasi dan Permodelan Kecepatan Motor DC	69
Gambar 3.33 Struktur Jaringan FMN Tipe II.....	70
Gambar 4.1 Respon ICE Berbeban Tanpa Bantuan Motor DC Terhadap <i>Sinyal Step</i>	86
Gambar 4.2 Respon ICE Berbeban Tanpa Bantuan Motor DC Terhadap <i>Set Point</i> Dengan Perubahan Amplitudo.....	86
Gambar 4.3 <i>Output</i> Kontroler <i>Neuro-Fuzzy</i> Dengan $\alpha=0,0001$	87
Gambar 4.4 <i>Output</i> Kontroler <i>Neuro-Fuzzy</i> Dengan $\alpha=0,001$	88
Gambar 4.5 <i>Output</i> Kontroler <i>Neuro-Fuzzy</i> Dengan $\alpha=0,01$	88
Gambar 4.6 Respon ICE Berbeban Dengan Bantuan Motor DC Tanpa Model Prediksi Terhadap <i>Sinyal Step</i>	89
Gambar 4.7 <i>Sinyal</i> Kontrol Motor DC Tanpa Model Prediksi Terhadap <i>Sinyal Step</i>	90
Gambar 4.8 Respon ICE Berbeban Dengan Bantuan Motor DC Tanpa Model Prediksi Terhadap <i>Set Point</i> Dengan Perubahan Amplitudo	90

Gambar 4.9 Sinyal Kontrol Motor DC Tanpa Model Prediksi Terhadap Sinyal Masukan Dengan Perubahan Amplitudo	91
Gambar 4.10 Respon ICE Berbeban Dengan Bantuan Motor DC Menggunakan Model Prediksi Terhadap Sinyal <i>Step</i>	92
Gambar 4.11 Sinyal Kontrol Motor DC Menggunakan Model Prediksi Terhadap Sinyal <i>Step</i>	92
Gambar 4.12 Respon ICE Berbeban Dengan Bantuan Motor DC Menggunakan Model Prediksi Terhadap <i>Set Point</i> Dengan Perubahan Amplitudo.....	93
Gambar 4.13 Sinyal Kontrol Motor DC Menggunakan Model Prediksi Terhadap Sinyal Masukan Dengan Perubahan Amplitudo	93
Gambar 4.14 Respon ICE Berbeban Nominal Tanpa Bantuan Motor DC	94
Gambar 4.15 Respon Hasil Implementasi Kontroler <i>Neuro-Fuzzy</i> Tanpa Model Prediksi	95
Gambar 4.16 Respon Hasil Implementasi Kontroler <i>Neuro-Fuzzy</i> Tanpa Model Prediksi	96



DAFTAR TABEL

Tabel 2.1 Perbandingan Kapasitas Konfigurasi Seri, Paralel, dan Seri-Paralel.....	6
Tabel 2.2 Spesifikasi <i>Board</i> Arduino Uno	19
Tabel 2.3 Nilai Konstan <i>Ratio T</i>	21
Tabel 2.4 Pendekatan Orde ke-n dan Nilai Titik Singgung.....	22
Tabel 2.5 Perbandingan logika <i>fuzzy</i> dan jaringan syaraf tiruan	30
Tabel 3.1 Spesifikasi ICE.....	39
Tabel 3.2 Spesifikasi Motor DC	40
Tabel 3.3 Spesifikasi <i>Tachogenerator</i>	45
Tabel 3.4 Hubungan Antara Tegangan <i>Input</i> dan Kecepatan Putar Motor DC	53
Tabel 3.5 Data Pengujian Sensor <i>Tachogenerator</i>	54
Tabel 3.6 Hubungan Tegangan Potensiometer dengan Kecepatan Putar ICE.....	55
Tabel 3.7 Pemodelan dan Validasi <i>Plant</i> ICE dengan Beban Minimal. 58	
Tabel 3.8 Pemodelan dan Validasi <i>Plant</i> ICE dengan Beban Nominal. 61	
Tabel 3.9 Pemodelan dan Validasi <i>Plant</i> ICE dengan Beban Maksimal	64
Tabel 3.10 Pemodelan dan Validasi <i>Plant</i> Motor DC.....	68
Tabel 3.11 Basis Aturan <i>Neuro-Fuzzy Controller</i>	73



Halaman ini sengaja dikosongkan

BAB 1

PENDAHULUAN

1.1 Latar Belakang Masalah

Penggunaan bahan bakar minyak untuk kendaraan bermotor yang semakin meningkat menyebabkan semakin menurunnya stok bahan bakar minyak di dunia. Hal itu menyebabkan terjadinya krisis energi di dunia. Selain itu, semakin banyaknya penggunaan bahan bakar minyak pada kendaraan bermotor menyebabkan semakin meningkatnya polusi udara. Polusi yang dihasilkan oleh kendaraan bermotor akan menyebabkan *global warming* sehingga suhu di atmosfer bumi akan meningkat.

Dalam upaya menanggulangi krisis energi dan bahaya *global warming* yang dihasilkan oleh kendaraan bermotor, maka diciptakan suatu kendaraan alternatif yang hemat energi dan ramah lingkungan yang disebut *Hybrid Electric Vehicle (HEV)*. HEV merupakan suatu kendaraan yang menggunakan *internal combustion engine (ICE)* dan motor listrik sebagai motor penggerak. Dengan adanya HEV ini menyebabkan ICE tidak bekerja sendirian dalam menggerakkan kendaraan sehingga penggunaan bahan bakar dapat diminimalkan.

Pada proses akselerasi, dibutuhkan kerja motor listrik untuk membantu kerja ICE agar akselerasi HEV dapat tercapai sesuai dengan yang diinginkan dengan penggunaan bahan bakar yang lebih minimal. Pada Tugas Akhir ini digunakan simulator HEV untuk merepresentasikan HEV. Simulator HEV yang digunakan telah dirancang pada penelitian Tugas Akhir sebelumnya [1][2].

1.2 Rumusan Masalah

Performa dan kinerja HEV ketika mengalami proses akselerasi masih perlu untuk ditingkatkan kembali. Peningkatan performa HEV ini tentunya membutuhkan suatu algoritma kontrol tertentu. Proses peningkatan performa HEV dilakukan dengan cara mengatur kecepatan motor listrik sehingga akan mencapai suatu nilai referensi tertentu. Proses percepatan pada HEV membutuhkan waktu yang cepat dalam mencapai nilai referensi. Pada penelitian ini akan digunakan metode kontrol *Neuro-Fuzzy* Prediktif untuk meningkatkan performa HEV ketika terjadi proses percepatan.

1.3 Tujuan

Pada Tugas Akhir kali ini akan digunakan metode *Neuro-Fuzzy* Prediktif untuk mengatur kecepatan motor listrik untuk meningkatkan akselerasi dari simulator HEV. Kontroler *Neuro-Fuzzy* memiliki keunggulan pada kemampuan adaptasi, sedangkan kontroler yang bersifat prediktif dapat memperkirakan nilai *error* yang akan datang. Sehingga akan diperoleh respon sistem yang lebih cepat daripada kontroler lainnya. Dengan adanya kontroler ini diharapkan motor listrik dapat membantu ICE dalam mencapai akselerasi yang diinginkan pada simulator HEV

1.4 Batasan Masalah

Permasalahan pada Tugas Akhir ini dibatasi oleh beberapa hal antara lain:

- a. ICE dioperasikan secara manual dan tidak diberikan aksi kontrol.
- b. Motor listrik diberikan tegangan dari jala jala PLN, tidak menggunakan baterai sehingga tidak ada mekanisme *charging* dan *discharging*.
- c. Pengaturan traksi simulator HEV dilakukan oleh motor listrik saat keadaan akselerasi.

1.5 Sistematika Penulisan

Sistematika penulisan buku Tugas Akhir kali ini adalah sebagai berikut:

BAB 1 : PENDAHULUAN

Dalam bab ini dijelaskan mengenai latar belakang, rumusan masalah, tujuan, batasan masalah, sistematika penulisan, dan relevansi.

BAB 2 : DASAR TEORI

Dalam bab ini dijelaskan mengenai teori yang menunjang penelitian, berupa teori tentang HEV dan komponennya, serta metode yang digunakan untuk pengaturan HEV.

BAB 3 : PERANCANGAN SISTEM

Dalam bab ini dijelaskan mengenai perancangan perangkat keras simulator HEV, perangkat lunak, dan perancangan kontroler .

BAB 4 : PENGUJIAN DAN ANALISIS SISTEM

Dalam bab ini dijelaskan mengenai hasil simulasi kontroler dan analisisnya. Selain itu, dijelaskan mengenai hasil implementasi kontroler pada simulator HEV beserta analisa hasil implementasi.

BAB 5 : KESIMPULAN DAN SARAN

Dalam bab ini dijelaskan mengenai kesimpulan hasil penelitian dan saran untuk penelitian selanjutnya.

1.6 Relevansi

Hasil dari Tugas Akhir ini diharapkan dapat berkontribusi dalam pengembangan penelitian tentang HEV serta diharapkan dapat meningkatkan teknologi otomotif dalam negeri.

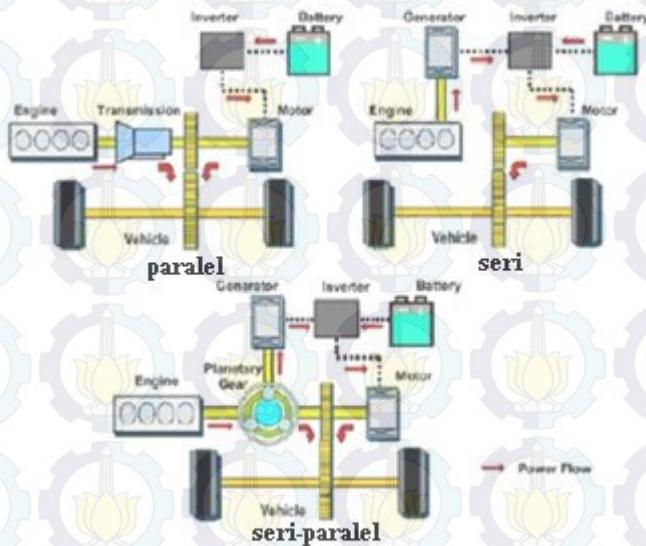


Halaman ini sengaja dikosongkan

BAB 2 DASAR TEORI

2.1 Hybrid Electric Vehicle (HEV) [1][2]

Hybrid Electric Vehicle (HEV) merupakan kendaraan yang mengombinasikan antara mesin bahan bakar dan mesin listrik. Saat ini HEV sedang dikembangkan di beberapa negara karena dengan adanya HEV ini polusi yang dihasilkan semakin kecil, selain itu pemakaian minyak bumi sebagai bahan bakar kendaraan bermotor akan semakin berkurang. Berdasarkan arsitektur mekanik, HEV dibagi menjadi tiga kategori yaitu: konfigurasi seri, konfigurasi paralel, dan konfigurasi seri-paralel.



Gambar 2.1 Konfigurasi HEV

Pada konfigurasi seri, motor listrik berfungsi penuh sebagai penggerak HEV. Energi yang dihasilkan oleh mesin bahan bakar dikonversikan menjadi energi listrik. Pada konfigurasi paralel, dua sumber energi bisa menggerakkan HEV secara bergantian maupun

secara bersamaan. Sedangkan pada konfigurasi seri-paralel, yaitu mengombinasikan keduanya dimana mesin bahan bakar dapat berfungsi sebagai penggerak HEV dan dapat juga berfungsi sebagai sumber energy bagi motor listrik.

Pada HEV dengan konfigurasi paralel, sumber energi yang ada pada HEV ini bekerja sesuai dengan porsinya masing-masing. Pada saat HEV akan bergerak, maka mesin listrik yang menggerakkan HEV sedangkan mesin bahan bakar tidak bekerja pada kondisi ini. Hal ini dikarenakan pada kondisi tersebut kecepatan HEV tidak terlalu tinggi sehingga energi yang dibutuhkan tidak terlalu besar. Pada saat kondisi akselerasi, ICE dan mesin listrik bekerja secara bersamaan karena energi yang dibutuhkan besar. Pada saat HEV dalam kondisi kecepatan tinggi konstan, ICE yang bertugas untuk menggerakkan roda. Hal ini dikarenakan pada saat ini energi yang dibutuhkan sangat besar, apabila motor listrik juga bekerja pada kondisi tersebut maka dikhawatirkan akan mengganggu kecepatan dari HEV.

Pada konfigurasi dengan menggunakan *power split device*, maka kelemahan konfigurasi paralel dan seri bisa diatasi. Pada Tabel 2.1 dapat dilihat perbandingan kapasitas konfigurasi paralel, seri, dan *power split device* (seri-paralel).

Tabel 2.1 Perbandingan Kapasitas Konfigurasi Seri, Paralel, dan Seri-Paralel [1]

Tipe	Efisiensi Bahan Bakar			Performansi	
	<i>Idling Stop</i>	<i>Recovery Energy</i>	Efisiensi	Akselerasi	Kestabilan Kecepatan
Seri	Baik	Sangat baik	Baik	Kurang baik	Kurang baik
Paralel	Baik	Baik	Baik	Baik	Kurang baik
Seri-Paralel	Sangat baik	Sangat baik	Sangat baik	Baik	Baik

Dilihat dari Tabel 2.1, maka dapat disimpulkan bahwa konfigurasi seri-paralel bisa mereduksi kelemahan dari konfigurasi paralel maupun konfigurasi seri. Pada penelitian Tugas Akhir ini akan digunakan simulator HEV dengan konfigurasi paralel.

2.2 Simulator HEV

Simulator HEV yang digunakan pada Tugas Akhir ini terdiri dari *internal combustion engine* (ICE) dan motor DC sebagai elemen penggerak. Simulator HEV ini memiliki konfigurasi paralel dimana ICE bertindak sebagai penggerak utama.

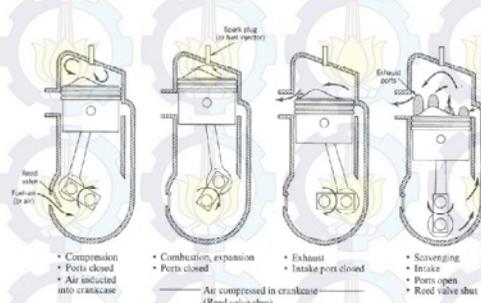
2.2.1 Internal Combustion Engine (ICE) 2 Tak [3]

Mesin bakar merupakan suatu perangkat yang dapat mengubah energi kalor menjadi energi gerak. Pembakaran yang terjadi pada ICE ini dilakukan di dalam mesin tersebut. Mesin ini disebut sebagai *internal combustion engine* (ICE). Sedangkan, mesin bakar yang proses pembakarannya terjadi di luar mesin disebut sebagai *external combustion engine* (ECE). Konstruksi ICE lebih sederhana dibandingkan dengan ECE, selain itu tingkat efisiensi ICE juga lebih tinggi daripada ECE. ICE biasanya digunakan pada kendaraan bermotor, seperti: mesin dua langkah (dua tak), mesin empat langkah (empat tak), dan mesin diesel. Pada subbab ini akan dibahas mengenai ICE dua langkah (dua tak) dan ICE empat langkah (empat tak).

ICE empat tak merupakan mesin bensin dimana pada suatu kerja diperlukan empat gerakan *piston* dan dua kali putaran poros. Langkah hisap awal, *piston* bergerak pada Titik Mati Atas (TMA) menuju Titik Mati Bawah (TMB). Pada langkah ini katup hisap terbuka sedangkan katup buang tertutup sehingga udara dan bahan bakar masuk. Setelah itu, *piston* yang telah mencapai TMB akan kembali bergerak menuju TMA. Pada saat terjadi langkah ini katup hisap dan katup buang tertutup sehingga tekanan dan suhu di dalam *piston* naik. Beberapa saat sebelum *piston* mencapai TMA, terjadi proses pengapian yang disebabkan oleh busi. Hal ini menyebabkan udara memuai sehingga *piston* akan didorong menuju TMB. Sebelum *piston* bergerak menuju TMB, katup buang terbuka sehingga gas sisa pembakaran akan keluar dari *piston*. Setelah *piston* mencapai TMB, maka *piston* akan bergerak menuju TMA. Pada saat ini, *piston* akan mendorong keluar sisa gas pembakaran.

ICE dua tak merupakan mesin bensin dimana pada suatu kerja diperlukan dua kali gerakan *piston* dan satu kali putaran poros. Langkah awal pada ICE dua tak disebut sebagai langkah hisap dan kompresi. Pada langkah ini, *piston* bergerak menuju TMA sehingga menyebabkan ruang engkol membesar dan menjadikan ruang tersebut hampa (*vacuum*). Katup hisap terbuka sehingga campuran udara dan bahan

bakar masuk menuju ruang engkol. Ketika *piston* bergerak menuju TMA menyebabkan katup hisap tertutup sehingga akan menyebabkan tekanan dan suhu bahan bakar di atas *piston* naik. Beberapa saat sebelum *piston* sampai di TMA, akan terjadi percikan api dari busi dan terjadi proses pengapian. Hal ini menimbulkan gaya yang mendorong *piston* bergerak menuju TMB. Pada langkah kedua disebut sebagai langkah kerja dan buang. Sesaat setelah *piston* bergerak menuju TMB, katup buang akan terbuka dan menyebabkan sisa gas pembakaran keluar dari ruang silinder. Ketika *piston* bergerak menuju TMB akan menyebabkan campuran bahan bakar dan udara bergerak ke atas. Siklus kerja ICE dua tak dapat dilihat pada Gambar 2.3.



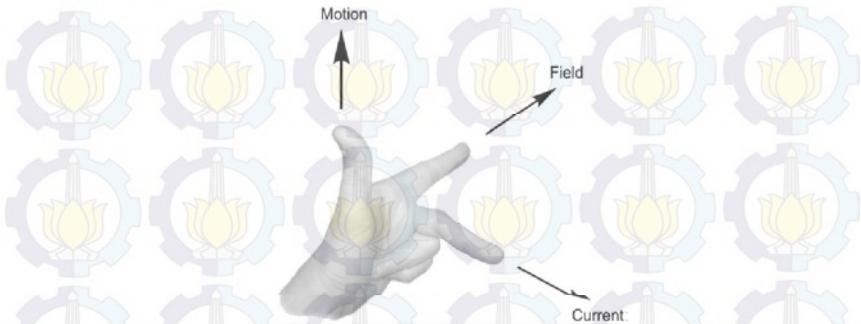
Gambar 2.2 Siklus Kerja ICE 2 Tak [1]

2.2.2 Motor Listrik DC [4]

Motor listrik merupakan suatu perangkat yang dapat mengubah energi listrik menjadi energi mekanik. Berdasarkan sumber yang dibutuhkan, motor listrik dibagi menjadi dua yaitu: motor listrik AC dan motor listrik DC. Motor listrik AC merupakan motor listrik yang membutuhkan sumber berupa tegangan bolak-balik untuk diubah menjadi energi mekanik. Sedangkan motor DC merupakan motor listrik yang membutuhkan sumber berupa tegangan searah untuk diubah menjadi energi mekanik. Pada tugas akhir ini, digunakan motor listrik DC untuk membantu kinerja motor bakar pada simulator HEV.

Prinsip kerja motor DC berawal dari prinsip bahwa apabila terdapat konduktor yang menghantarkan arus dan diletakkan pada medan magnet maka akan menghasilkan suatu gaya. Arah gaya yang dihasilkan ditentukan melalui kaidah tangan kiri, dimana ibu jari diibaratkan sebagai arah gaya, jari telunjuk sebagai arah medan magnet,

dan jari tengah diibaratkan sebagai arah arus yang mengalir pada konduktor.

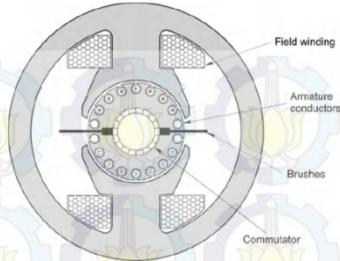


Gambar 2.3 Kaidah Tangan Kiri

Berdasarkan Gambar 2.3 dapat dilihat bahwa terdapat hubungan antara gaya, medan magnet, dan arus listrik. Gaya yang dihasilkan konduktor dapat dirumuskan dalam Persamaan 2.1.

$$F = BIl \quad (2.1)$$

F adalah gaya yang dihasilkan oleh konduktor dengan satuan Newton (N). B merupakan medan magnet dengan satuan Tesla, I merupakan arus yang mengalir pada konduktor dengan satuan Ampere (A), dan l merupakan panjang dari konduktor dengan satuan meter. Persamaan tersebut dapat diaplikasikan apabila arah medan magnet tegak lurus dengan arah arus. Apabila arah medan magnet tidak tegak lurus dengan arus maka gaya yang dihasilkan akan semakin kecil dan apabila arah medan magnet sejajar dengan arah arus maka gaya yang dihasilkan akan bernilai nol. Gaya yang dihasilkan ini dapat menyebabkan terjadinya gerak rotasi pada motor.



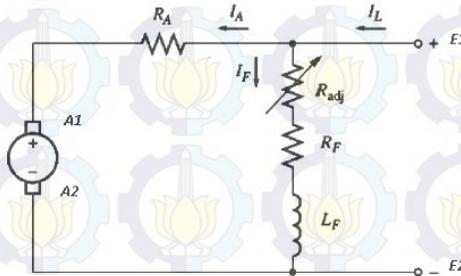
Gambar 2.4 Konstruksi Motor DC dengan Sikat (*Brushed*)

Komponen utama pada motor DC, antara lain: stator, rotor, dan komutator. Bagian stator dapat berupa magnet permanen maupun lilitan medan. Keuntungan dan kerugian apabila pada bagian stator menggunakan magnet permanen antara lain:

- a. Tidak dibutuhkan *supply* listrik
- b. Tidak ada rugi-rugi I^2R

Kekurangan:

- a. Kekuatan dari medan selalu konstan

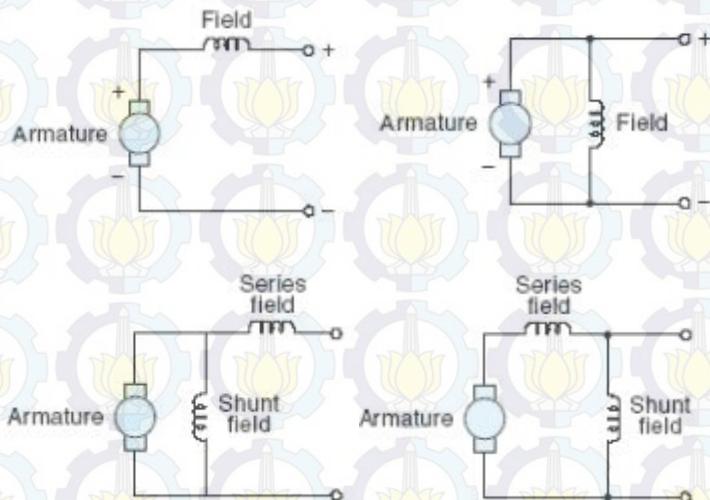


Gambar 2.5 Rangkaian Ekuivalen Motor DC *Shunt* Penguat Sendiri

Rangkaian pada motor DC dibagi menjadi dua, yaitu terminal E1,E2 dan terminal A1,A2. Terminal E1,E2 dihubungkan dengan kumparan medan di sekitar kedua kutubnya. Kumparan ini akan menghasilkan gaya magnet untuk mengatur fluks magnet. Pada kondisi *steady*, daya masukan terminal E1,E2 akan menghasilkan energi panas dan tidak bisa menghasilkan energi gerak. Terminal A1,A2 dihubungkan dengan kumparan pada rotor dan akan menghasilkan torsi yang dapat menyebabkan gerak rotasi pada rotor. Torsi dihasilkan oleh interaksi

pada arus yang mengalir pada rotor dan fluks magnet yang dihasilkan oleh rotor.

Bagian rotor merupakan bagian pada motor DC yang bergerak. Rotor terdiri dari konduktor yang dililitkan pada magnet. Konduktor ini akan dialiri arus yang akan menyebabkan timbulnya gaya pada konduktor ini. Pada motor DC *brushed*, menggunakan komutator dan sikat untuk memberikan daya listrik pada rotor. Motor DC *brushed*, lebih banyak digunakan pada dunia industri karena rangkaiannya sederhana dan berharga lebih murah. Motor DC *brushed* memiliki beberapa konfigurasi, antara lain: seri, *shunt*, dan *compound*.



Gambar 2.6 Konfigurasi Motor DC

Perbedaan konfigurasi dari keempat tipe motor DC dapat dilihat pada Gambar 2.6. Pada motor DC konfigurasi seri, kumparan medan dirangkai seri terhadap kumparan jangkar. Pada motor DC konfigurasi *shunt*, kumparan medan dirangkai paralel terhadap kumparan jangkar. Hal ini menyebabkan berbedanya torsi yang dibutuhkan pada saat *start* awal motor DC. Pemilihan konfigurasi motor DC yang digunakan tergantung dari spesifikasi yang diinginkan karena kedua konfigurasi ini memiliki kelebihan dan kekurangan masing-masing. Untuk menutupi kekurangan yang dimiliki masing-masing

konfigurasi, maka terdapat konfigurasi motor DC gabungan yang disebut konfigurasi *compound*.

2.2.3 Rem Elektromagnetik Arus Eddy [5]

Sistem pengereman merupakan suatu sistem yang bertujuan untuk memperlambat maupun menghambat gerakan. Pada kendaraan bermotor umumnya terdapat beberapa sistem pengereman, antara lain: pengereman friksi maupun pengereman mesin (*engine brake*). Pengereman friksi pada kendaraan bermotor menggunakan piringan cakram maupun tromol. Dengan adanya gesekan maka akan menyebabkan terjadinya pengereman. *Engine brake* yang umumnya terdapat pada kendaraan roda empat, tidak memakai prinsip gesekan. *Engine brake* hanya bekerja pada kecepatan tertentu, pada saat itu secara tiba-tiba *supply* bahan bakarnya akan dihentikan. Akibatnya, pada ruang pembakaran tidak terjadi proses pembakaran sehingga kecepatan kendaraan akan berkurang. Saat ini, tidak hanya dua sistem pengereman tersebut yang digunakan dalam dunia otomotif. Terdapat suatu sistem pengereman yang tidak menggunakan prinsip gesekan. Sistem tersebut menggunakan rem elektromagnetik arus eddy .

Rem elektromagnetik arus eddy menggunakan prinsip hukum Faraday dan hukum Lenz. Arus eddy merupakan suatu loop tertutup dari arus induksi pada suatu permukaan yang tegak lurus terhadap fluks magnet. Arus ini akan sejajar dengan kumparan dan alirannya akan terbatas oleh daerah induksi medan magnet.



Gambar 2.7 Rem Elektromagnetik

Komponen dasar dari rem elektromagnetik adalah sumber magnet yang diam dan konduktor yang terhubung dengan poros. Pada hukum Faraday dijelaskan bahwa besar ggl induksi atau arus induksi tergantung pada laju perubahan fluks dan banyaknya lilitan. Pada hukum

Lenz dijelaskan bahwa “arus induksi yang timbul arahnya sedemikian sehingga menimbulkan medan magnet induksi yang melawan arah perubahan medan magnet.” Dari penjelasan ini dapat disimpulkan bahwa arus induksi akan timbul jika terdapat perubahan fluks. Perubahan fluks ini akan terjadi apabila konduktor tersebut berputar. Semakin cepat putaran maka akan semakin besar pula laju perubahan fluks sehingga arus induksi yang dihasilkan akan semakin besar, begitu juga sebaliknya. Arus induksi yang dihasilkan tadi akan menimbulkan medan magnet induksi yang berlawanan arah dengan arah perubahan medan magnet sehingga akan menjadikan putaran dari konduktor melambat.

Berdasarkan hukum Lenz dapat direpresentasikan dalam Persamaan (2.2)

$$W = E \cdot I \cdot \Delta t \quad (2.1)$$

$$W = -F \cdot s \quad (2.2)$$

maka:

$$E \cdot I \cdot \Delta t = -B \cdot l \cdot l \cdot v \cdot \Delta t \quad (2.3)$$

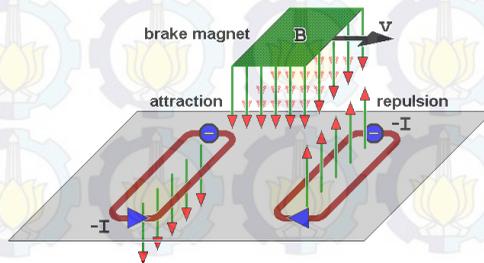
Dimana $B \cdot l \cdot v \cdot \Delta t$ merupakan banyaknya garis gaya yang dipotong oleh konduktor, dapat dirumuskan perubahan garis gaya $\Delta\Phi$. Sehingga dapat dirumuskan dalam Persamaan (2.4)

$$E = -\frac{\Delta\Phi}{\Delta t} \quad (2.4)$$

di mana:

E : gaya gerak listrik induksi

$\frac{\Delta\Phi}{\Delta t}$: perubahan garis gaya listrik setiap satuan waktu



Gambar 2.8 Gaya Pengereman yang Ditimbulkan Arus Eddy

2.2.4 Perbandingan Gear [1]

Gear merupakan suatu lingkaran yang memiliki gerigi di bagian luarnya. *Gear* ini digunakan untuk mengubah kecepatan sudut atau torsi. Bentuk fisik *gear* dapat dilihat pada Gambar 2.9.



Gambar 2.9 Bentuk Fisik *Gear*

Torsi pada *gear* dapat dihitung melalui Persamaan (2.5)

$$\tau = F \times R \quad (2.5)$$

Dimana τ merupakan torsi dengan satuan Nm. F merupakan gaya yang bekerja pada *gear* dan R merupakan jari-jari *gear*. Konfigurasi dasar penggunaan *gear* ada dua, yaitu:

a. Beda poros (*gear* reduksi)

Aturan yang berlaku pada *gear* beda poros dapat dilihat pada Persamaan (2.6)

$$V_A = V_B$$

$$\omega_A R_A = \omega_B R_B$$

$$\frac{\omega_A}{\omega_B} = \frac{R_B}{R_A} \quad (2.6)$$

dimisalkan $R_A = 4R_B$

$$\frac{\omega_A}{\omega_B} = \frac{R_B}{4R_B}$$

$$\omega_A = \frac{1}{4}\omega_B$$

$$F_A = F_B$$

$$\frac{\tau_A}{R_A} = \frac{\tau_B}{R_B}$$

$$\frac{\tau_A}{\tau_B} = \frac{R_A}{R_B}$$

$$\frac{\tau_A}{\tau_B} = \frac{4R_B}{R_B}$$

$$\tau_A = 4\tau_B$$

(2.7)

di mana:

V_A : Kecepatan linier pada gear A

V_B : Kecepatan linier pada gear B

ω_A : Kecepatan sudut pada gear A

ω_B : Kecepatan sudut pada gear B

R_A : Jari-jari pada gear A

R_B : Jari-jari pada gear B

F_A : Gaya yang bekerja pada gear A

F_B : Gaya yang bekerja pada gear B

τ_A : Torsi pada gear A

τ_B : Torsi pada gear B

Konfigurasi dua buah gear ditunjukkan dengan Gambar 2.10. Kedua buah poros gear ini dihubungkan dengan menggunakan chain (belt).



Gambar 2.10 Konfigurasi Beda Poros

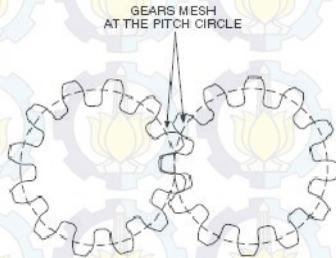
b. Satu poros (*compound gear*)

Dalam pemilihan kombinasi *gear* yang akan digunakan terdapat dua hal yang harus diperhatikan *diametral pitch* dan jumlah pada *gear*. Konfigurasi *gear* dengan sistem satu poros terlihat pada Gambar 2.11.



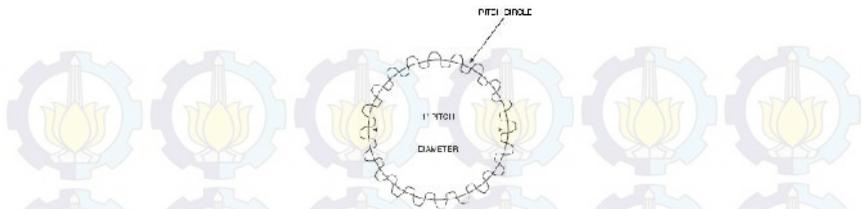
Gambar 2.11 Konfigurasi Satu Poros

Pitch Circle adalah lingkaran yang terjadi ketika dua buah *gear* bersinggungan seperti ditampilkan pada Gambar 2.12.



Gambar 2.12 *Pitch Circle*

Pitch diameter adalah diameter dari *pitch circle*, sedangkan *diameter pitch* adalah rasio jumlah gigi terhadap *pitch diameter* dari *gear*. Penjelasan dari *pitch diameter* dan *diametral pitch* diperhatikan pada Gambar 2.13.



Gambar 2.13 Gear Dengan *Pitch Diameter 1"* dan *Diametral Pitch 20*

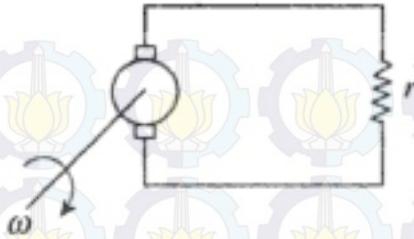
2.2.5 Tachogenerator [6]

Tachogenerator merupakan suatu instrumen yang dapat mengubah energi mekanik menjadi sinyal listrik. *Tachogenerator* digunakan sebagai sensor kecepatan. *Tachogenerator* dibagi menjadi dua macam, yaitu: *tachogenerator* AC dan *tachogenerator* DC. Prinsip dari *tachogenerator* kerja ini didasarkan pada fluks magnet yang konstan dan gaya gerak listrik yang bernilai proporsional terhadap kecepatan sudut. Pada tugas akhir ini digunakan *tachogenerator* DC sebagai sensor kecepatan pada simulator HEV. *Tachogenerator* DC pada simulator HEV dipasang *couple* dengan poros motor DC. Pada Gambar 2.14 ditunjukkan gambar fisik dari *tachogenerator* DC



Gambar 2.14 *Tachogenerator*

Struktur jangkar pada *tachogenerator* DC sama dengan generator DC pada umumnya. Untuk menghasilkan *output* yang lebih halus digunakan generator dua kutub atau lebih. Pada Gambar 2.15 ditunjukkan rangkaian *tachogenerator* DC dengan medan magnet permanen.



Gambar 2.15 Rangkaian *Tachogenerator* DC

2.3 Sistem Minimum Arduino Uno R3 [7]

Arduino Uno merupakan suatu *board* mikrokontroler berbasis Atmega328. Arduino Uno memiliki 14 *input/output* pin (6 pin sebagai PWM *output*), 6 *input* analog, 16 MHz *ceramic resonator*, koneksi USB, *jack power*, *ICSP header*, dan tombol *reset*. Arduino Uno berisi semua yang diperlukan untuk mendukung mikrokontroler, hanya terhubung ke komputer dengan kabel USB atau menggunakan sumber dari AC ke DC *adapter*. Pada Gambar 2.16 ditunjukkan bentuk fisik dari Arduino Uno R3.



Gambar 2.16 Bentuk Fisik Sistem Minimum Arduino Uno R3

Arduino Uno R3 merupakan seri dari Arduino Uno Revisi Ketiga dengan memiliki fitur-fitur baru sebagai berikut:

- a. Pinout menambahkan pin SDA dan SCL yang dekat dengan pin AREF dan dua pin baru lainnya ditempatkan dekat dengan pin

- RESET*. IOREF memungkinkan untuk beradaptasi dengan tegangan yang disediakan dari papan. Untuk kedepannya, *shield* akan kompatibel baik dengan papan yang menggunakan AVR, yang beroperasi dengan 5V
- b. Sirkuit *RESET* yang lebih memadai
 - c. Atmega 16U2 mengganti 8U2

Tabel 2.2 Spesifikasi *Board* Arduino Uno

Klasifikasi	Spesifikasi
Mikrokontroler	ATMega 328
Tegangan Operasi	5 V
Tegangan Masukan (rekomendasi)	7-12 V
Tegangan Masukan (batas)	6-20 V
Pin I/O Digital	14 (6 di antaranya bisa digunakan untuk <i>output</i> PWM)
Pin Analog <i>Input</i>	6
Arus DC setiap pin I/O	40 mA
Flash Memory	32 kB (ATMega 328) dimana 0,5 kB digunakan sebagai <i>Bootloader</i>
SRAM	2 kB (ATMega 328)
EEPROM	1 kB (ATMega 328)
Kecepatan Clock	16 MHz

Arduino Uno memiliki beberapa fasilitas untuk berhubungan dengan komputer, Arduino lainnya, dan mikrokontroler lainnya. ATMega328 menyediakan UART TTL (5V) komunikasi serial, dimana tersedia pin *digital* 0 (RX) dan 1 (TX). Komunikasi serial melalui USB akan menghasilkan *virtual com port* pada *software* di komputer. Pada *software* Arduino termasuk monitor yang terdiri dari data sederhana untuk dikirim maupun diterima oleh *board* Arduino. LED RX dan TX pada *board* akan menyala ketika data dikirimkan melalui *chip* USB to Serial dan koneksi USB ke komputer.

2.4 Identifikasi Sistem [8]

Identifikasi sistem bertujuan untuk memperoleh parameter-parameter dari suatu *plant* berdasarkan data hasil pengukuran. Identifikasi sistem ini selanjutnya akan digunakan untuk memodelkan sistem karena sebagian besar kontroler berbasis model (*model based controller*) dimana pada tahap perancangannya dibutuhkan model

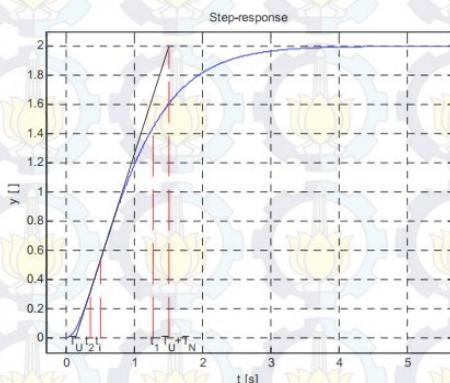
matematis dari *plant*. Untuk memodelkan suatu sistem, dilakukan identifikasi sistem secara *open loop* maupun *close loop*. Identifikasi *close loop* biasanya digunakan pada sistem yang tidak stabil, sedangkan identifikasi *open loop* digunakan pada sistem yang sudah stabil.

2.4.1 Metode Identifikasi

Metode identifikasi dapat dilakukan dengan cara statis dan dinamis. Metode identifikasi statis dilakukan dengan memberikan *input* berupa sinyal *step* pada *plant*. Sedangkan metode identifikasi dinamis dilakukan dengan memberikan *input* berupa sinyal acak. Identifikasi dinamis ini akan menghasilkan parameter-parameter yang digunakan untuk membuat model matematis *plant*. Proses identifikasi dapat dilakukan melalui beberapa metode, antara lain: metode Harriot, metode Viteckova, metode Smith, metode Strejc, dan lain-lain. Pada Tugas Akhir ini proses identifikasi ICE dilakukan dengan menggunakan metode Strejc.

2.4.1.1 Metode Strejc [8]

Sistem tanpa osilasi dapat dijadikan model matematis dengan menggunakan pendekatan metode Strejc. Penghitungan model matematis menggunakan metode Strejc diperoleh dengan bantuan waktu T_u dan T_N seperti ditunjukkan pada Gambar 2.17.



Gambar 2.17 Respon Step Sistem dengan waktu T_u , T_N , t_1 , dan t_2

Perbandingan nilai waktu T_u dan T_N akan menghasilkan parameter τ .

$$\tau = \frac{T_u}{T_N} \quad (2.8)$$

Pada kondisi $\tau < 0$, sistem akan didekati dengan model orde dua seperti pada Persamaan (2.9)

$$G(s) = \frac{K}{(\tau_{ST1}s + 1)(\tau_{ST2}s + 1)} \quad (2.9)$$

Parameter τ_{ST1} dan τ_{ST2} didapatkan dari perhitungan di bawah ini:

$$\tau_{ST1} + \tau_{ST2} = \frac{t_1}{1,2564} \quad (2.10)$$

a. Nilai t_1 merupakan waktu ketika respon sistem mencapai 72% dari nilai *steady-state* ($0,72y_\infty$).

b. Menentukan nilai t_2 dari Persamaan (2.11)

$$t_2 = 0,3574(\tau_{ST1} + \tau_{ST2}) \quad (2.11)$$

c. Menentukan nilai T menggunakan Persamaan (2.12)

$$T = \frac{\tau_{ST1}}{\tau_{ST2}} \quad (2.12)$$

d. Menentukan nilai $y(t_2)$ berdasarkan Tabel 2.3

Tabel 2.3 Nilai Konstan *Ratio T*

$Y(t_2)$	T	$Y(t_2)$	T
0,30	0,000	0,22	0,183
0,29	0,023	0,21	0,219
0,28	0,043	0,20	0,264
0,27	0,063	0,19	0,322
0,26	0,084	0,18	0,403
0,25	0,105	0,17	0,538
0,24	0,128	0,16	1,000
0,23	0,154		

e. Nilai τ_{ST1} dan τ_{ST2} diperoleh dari Persamaan (2.10) dan Persamaan (2.12)

Pada kondisi $\tau > 0$, sistem akan didekati dengan model orde ke- n seperti pada Persamaan (2.13)

$$G(s) = \frac{K}{(\tau_{ST1}s + 1)^n} \quad (2.13)$$

- a. Nilai *ratio* τ didapatkan dari Persamaan (2.14)

$$\tau = \frac{T_u}{T_N} \quad (2.14)$$

- b. Dari penghitungan τ , ditentukan orde sistem berdasarkan Tabel 2.4

Tabel 2.4 Pendekatan Orde ke- n dan Nilai Titik Singgung

N	τ	y_i	n	τ	y_i
2	0,014	0,264	7	0,57	0,394
3	0,218	0,327	8	0,642	0,401
4	0,319	0,359	9	0,709	0,407
5	0,41	0,371	10	0,773	0,413
6	0,493	0,384			

- c. Nilai *time constant* τ_{ST} didapatkan dari Persamaan (2.15)

$$\tau_{ST} = \frac{t_1}{n - 1} \quad (2.15)$$

2.4.2 Validasi Sistem [9]

Hasil identifikasi sistem kemudian dibuat menjadi suatu model matematis. Suatu model matematis yang sudah dibuat kemudian dicocokkan dengan data aktual yang didapatkan dari proses identifikasi sistem. Untuk menghitung tingkat keakuratan pendekatan melalui model matematis dilakukan proses penghitungan *error* yang disebut sebagai *Root Mean Square Error (RMSE)*. Error yang diperoleh merupakan selisih antara nilai pada model matematis dengan nilai aktual hasil identifikasi sistem. Penghitungan RMSE dapat dituliskan dengan Persamaan (2.16)

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_a - y_m)^2} \quad (2.16)$$

dimana y_a : nilai keluaran aktual, y_m : nilai keluaran hasil permodelan, n : jumlah data.

Dari Persamaan (2.16), didapatkan bahwa untuk mencari nilai RMSE dibutuhkan nilai keluaran aktual dan nilai keluaran hasil permodelan. Kemudian selisih kedua nilai tersebut dikuadratkan dan dibagi dengan jumlah data yang didapatkan. Sampai tahap ini telah didapatkan nilai *Mean Square Error (MSE)*. Untuk memperoleh nilai RMSE, hasil MSE diakarkuadratkan. Permodelan matematis dari suatu sistem dikatakan mendekati sistem yang aktual apabila didapatkan nilai RMSE yang kecil.

2.5 Logika Fuzzy [10]

Logika *fuzzy* (samar) merupakan perluasan dari logika Boolean. Pada teori logika Boolean hanya terdapat dua definisi, yaitu nilai nol dan satu untuk merepresentasikan suatu keadaan. Pada logika *fuzzy* suatu keadaan direpresentasikan dalam rentang nilai nol dan satu.

2.5.1 Himpunan Fuzzy

Pada dasarnya, teori himpunan *fuzzy* merupakan perluasan dari teori himpunan klasik. Pada teori himpunan klasik, keberadaan elemen (x) pada suatu himpunan A hanya memiliki dua kemungkinan, yaitu merupakan anggota dari himpunan A atau bukan merupakan anggota dari himpunan A. Suatu nilai yang dapat menunjukkan seberapa besar tingkat keanggotaan (derajat keanggotaan) suatu elemen dinotasikan dalam $\mu_A(x)$. Pada teori himpunan klasik nilai $\mu_A(x)=1$ untuk x merupakan anggota himpunan A dan $\mu_A(x)=0$ untuk x bukan merupakan anggota himpunan A.

Pada teori himpunan *fuzzy*, nilai derajat keanggotaan $\mu_A(x)$ tidak hanya terbatas pada dua kemungkinan (nol atau satu). Nilai $\mu_A(x)$ pada himpunan *fuzzy* memiliki nilai pada rentang $[0,1]$.

2.5.2 Fungsi Keanggotaan

Fungsi keanggotaan (*membership function*) adalah suatu kurva yang menunjukkan pemetaan titik-titik *input* data ke dalam nilai keanggotaannya. Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaannya adalah dengan melalui pendekatan fungsi. Beberapa fungsi biasanya digunakan dalam merepresentasikan nilai keanggotaan dari himpunan *fuzzy*, antara lain: segitiga, trapesium, *sigmoid*, *gauss*, dan *generalized bell*.

a. Representasi Kurva Segitiga

Kurva segitiga pada dasarnya merupakan gabungan dari dua garis linear. Fungsi keanggotaan pada kurva segitiga dapat dituliskan dalam Persamaan (2.17)

$$\mu_A(x) = \begin{cases} 0 & x \leq a \\ \frac{x-a}{b-a} & a \leq x \leq b \\ \frac{c-x}{c-b} & b \leq x \leq c \\ 0 & x \geq c \end{cases} \quad (2.17)$$

b. Representasi Kurva Trapesium

Kurva trapesium memiliki kemiripan dengan kurva segitiga. Akan tetapi, pada kurva trapesium terdapat derajat keanggotaan yang bernilai satu pada beberapa titik. Fungsi keanggotaan pada kurva trapesium dapat dituliskan dalam Persamaan (2.18)

$$\mu_A(x) = \begin{cases} 0 & x \leq a \\ \frac{x-a}{b-a} & a \leq x \leq b \\ 1 & b \leq x \leq c \\ \frac{c-x}{c-b} & x \geq d \end{cases} \quad (2.18)$$

c. Representasi Kurva Sigmoid

Kurva sigmoid biasa disebut juga sebagai kurva *s-shaped* yang berhubungan dengan kenaikan atau penurunan secara tidak linear. Kurva sigmoid yang menunjukkan pertumbuhan akan bergerak dari sisi paling kiri (fungsi keanggotaan = 1) menuju sisi paling kanan (fungsi keanggotaan = 0). Fungsi keanggotaannya akan tertumpu pada 50% keanggotaannya yang sering disebut dengan titik infleksi. Kurva sigmoid yang menunjukkan penurunan bergerak dari sisi paling kanan (fungsi keanggotaan = 1) menuju sisi paling kiri (fungsi keanggotaan = 0). Keanggotaan dari kurva sigmoid direpresentasikan dalam tiga parameter, yaitu: nilai keanggotaan nol (α), titik infleksi (β), dan nilai keanggotaan penuh (γ). Fungsi keanggotaan pada kurva *sigmoid* dapat dituliskan dalam Persamaan (2.19)

$$\mu_A(x) = \left\{ \frac{1}{1 + e^{-\alpha(x-\beta)}} \right\} \quad (2.19)$$

d. Representasi Kurva Gauss

Keanggotaan pada kurva *Gauss* direpresentasikan dalam dua parameter yaitu γ dan k . Dimana γ menunjukkan nilai domain pada pusat kurva dan k menunjukkan lebar kurva. Fungsi keanggotaan kurva *Gauss* dapat dituliskan dalam Persamaan (2.20)

$$\mu_A(x) = \{e^{-k(\gamma-x)^2}\} \quad (2.20)$$

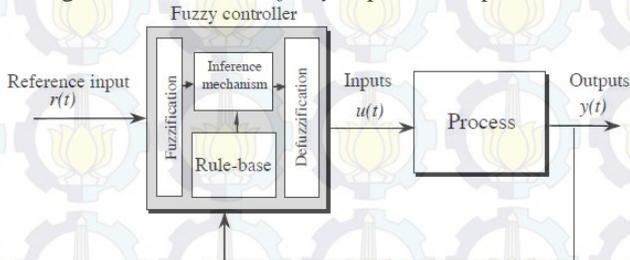
e. Representasi Kurva Generalized Bell

Kurva ini berbentuk menyerupai lonceng. Pada kurva ini terdapat tiga parameter, yaitu parameter titik yang memiliki nilai keanggotaan nol (a), lebar kurva ketika nilai $\mu_A(x)=0,5$, dan parameter titik tengah kurva (c). Fungsi keanggotaan kurva *Generalized Bell* dapat dituliskan dalam Persamaan (2.21)

$$\mu_A(x) = \left\{ \frac{1}{1 + \left| \frac{x-c}{a} \right|^{2b}} \right\} \quad (2.21)$$

2.6 Kontroler Fuzzy [11]

Diagram blok kontroler *fuzzy* dapat dilihat pada Gambar 2.18.



Gambar 2.18 Diagram Blok Kontroler Fuzzy

Dalam kontroler *fuzzy* terdapat empat komponen penyusun, yaitu:

- Fuzzifikasi
- Basis Aturan
- Mekanisme Inferensi
- Defuzzifikasi

2.6.1 Fuzzifikasi

Proses pengubahan nilai *input* menjadi nilai *fuzzy* yang memiliki rentang 0 sampai dengan 1. Aktivasi nilai *fuzzy* dilakukan dengan menggunakan fungsi keanggotaan *fuzzy*.

Kontroler *fuzzy* termasuk dalam kontroler yang bersifat *heuristik*. Hal ini menyebabkan penentuan *range input* dan jumlah fungsi keanggotaan *fuzzy* didasarkan dari pengetahuan terhadap *plant* yang akan dikontrol. Jumlah fungsi keanggotaan akan berpengaruh pada *output* kontroler *fuzzy*.

2.6.2 Basis Aturan

Basis aturan merupakan deskripsi linguistik terhadap variabel *input* dan *output*. Penentuan basis aturan yang dipakai dalam mengontrol suatu *plant* dapat melalui metode heuristik maupun deterministik. Metode heuristik didasarkan pada pengetahuan terhadap *plant* dan perilaku dari *plant* yang akan dikontrol. Sedangkan metode deterministik didapatkan melalui identifikasi struktur dan parameter dari aturan kontrol.

Pemetaan *input* dan *output* pada sistem *fuzzy* direpresentasikan dalam Persamaan (2.22)

If premis *Then* konsekuen (2.22)

Pada kontroler *fuzzy*, *input* kontroler direpresentasikan dalam premis dan *output* kontroler direpresentasikan dalam konsekuen. Jumlah basis aturan dari suatu sistem *fuzzy* ditentukan dari jumlah variabel pada *input* dan jumlah *membership function* pada variabel *input*, dirumuskan dalam Persamaan (2.23)

$$\prod_{i=1}^n N_i = N_1 \cdot N_2 \dots N_n \quad (2.23)$$

Dimana N_i merupakan jumlah *membership function* pada variabel *input* i . Sebagai contoh apabila variabel *input* pertama memiliki tiga *membership function* dan variabel *input* kedua memiliki tiga *membership function*, maka jumlah basis aturan adalah $3 \times 3 = 9$ aturan.

2.6.3 Mekanisme Inferensi

Mekanisme inferensi merupakan mekanisme operasi pada logika *fuzzy*. Operasi matematika yang dilakukan sesuai dengan premis

yang diberikan. Pada basis aturan terdapat terdapat dua operasi matematika yang dapat dilakukan adalah *AND* dan *OR*.

Mekanisme inferensi pada logika *fuzzy* dibagi menjadi dua tipe, yaitu Mamdani dan Takagi Sugeno. Perbedaan kedua metode ini terletak pada pengambilan kesimpulan logika *fuzzy*. Pada metode Mamdani, kesimpulan logika *fuzzy* berupa derajat keanggotaan sehingga dalam menyimpulkan suatu logika *fuzzy* dibutuhkan proses defuzzifikasi. Sedangkan pada tipe Takagi Sugeno, kesimpulan logika *fuzzy* berupa suatu persamaan sehingga tidak diperlukan proses defuzzifikasi. Kelebihan pada logika *fuzzy* tipe Mamdani lebih sederhana, akan tetapi diperlukan kemampuan untuk mengetahui karakteristik *plant* untuk menentukan *range output* kontroler. Pada tipe Takagi-Sugeno tidak diperlukan pengetahuan mengenai karakteristik dari *plant* akan tetapi diperlukan perhitungan yang lebih rumit untuk persamaan pada bagian konsekuen.

2.6.4 Defuzzifikasi

Hasil dari proses agregasi pada sistem *fuzzy*, berupa nilai derajat keanggotaan. Sebelum masuk ke dalam *plant*, nilai derajat keanggotaan harus diubah menjadi nilai tegas (*crisp*). Proses konversi nilai *fuzzy* menjadi nilai tegas disebut sebagai proses defuzzifikasi. Terdapat beberapa metode yang digunakan dalam proses defuzzifikasi, antara lain: *center of gravity (COG)*, *center of area (COA)*, dan *weighted sum (fuzzy singleton)*.

a. *Center of Gravity (COG)*

$$u_{crisp} = \frac{\sum_i b_i \int_u \mu_i(u) du}{\sum_i \int_u \mu_i(u) du} \quad (2.24)$$

dimana b_i merupakan titik tengah *membership function*, sedangkan $\int_u \mu_i(u) du$ merupakan luasan daerah di bawah fungsi keanggotaan $\mu_i(u)$.

b. *Center of Area (COA)*

$$u_{crisp} = \frac{\sum_i \int_u y \mu_i(u) du}{\sum_i \int_u \mu_i(u) du} \quad (2.25)$$

dimana $\int_u y\mu_i(u)du$ merupakan luasan daerah di bawah fungsi keanggotaan $\mu_i(u)$.

c. **Weighted Sum**

$$u_{crisp} = \frac{\sum_i \mu_i(u) s_i}{\sum_i \mu_i(u)} \quad (2.26)$$

dimana $\mu_i(u)$ merupakan fungsi keanggotaan hasil agregasi, sedangkan s_i merupakan nilai *fuzzy singleton*.

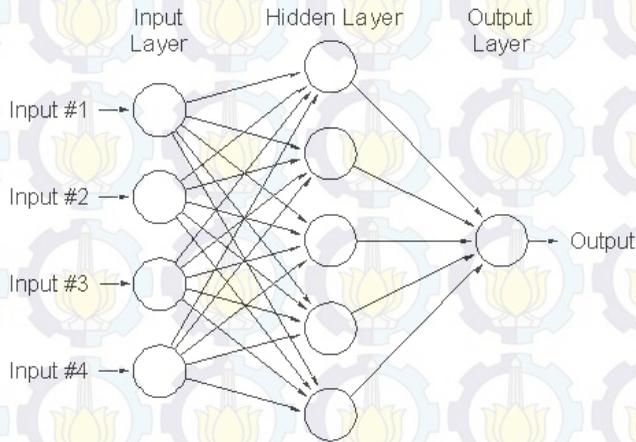
2.7 Jaringan Syaraf Tiruan [12]

Jaringan Syaraf Tiruan (JST) adalah suatu program komputer yang dibuat berdasarkan cara kerja otak manusia. Pada dasarnya, JST hampir sama dengan metode komputasi lainnya. Satu hal yang paling membedakan JST dengan metode komputasi lainnya adalah dalam melakukan proses pelatihan yang meniru kerja otak manusia. Proses komputasi pada JST diilhami dari struktur dan cara kerja otak manusia. Pada jaringan syaraf otak manusia, informasi disalurkan dari satu *neuron* ke *neuron* lainnya. Sementara pada JST proses penyaluran informasi dari satu *neuron* ke *neuron* lainnya diimplementasikan pada program komputer. Proses pelatihan JST umumnya menggunakan metode pelatihan *backpropagation* yang sudah banyak diterapkan pada semua proses pelatihan yang sederhana sampai yang rumit.

Metode pelatihan *backpropagation* termasuk ke dalam metode pelatihan terawasi (*supervisory learning*). *Supervisory learning* adalah metode pelatihan yang memasukkan target keluaran dalam data untuk proses pelatihannya. Metode pelatihan *backpropagation* telah banyak diaplikasikan dalam berbagai bidang, antara lain: bidang finansial, pengenalan pola tulis tangan, sistem kontrol, dan lain sebagainya. Metode *backpropagation* banyak diaplikasikan dalam berbagai proses karena metode ini didasarkan pada interkoneksi yang sederhana. Apabila keluaran JST tidak sesuai dengan keluaran yang diinginkan, maka metode *backpropagation* akan memperbaiki bobot (*weight*) yang ada pada lapisan tersembunyi (*hidden layer*) untuk mencapai keluaran yang JST yang sesuai dengan target keluaran.

Pada dasarnya JST akan diberikan pola masukan sebagai pola pelatihan maka pola akan menuju ke unit-unit lapisan tersembunyi dan akan diteruskan ke lapisan keluaran. Ketika keluaran JST tidak sama

dengan keluaran yang diinginkan maka keluaran akan disebarkan mundur yang selanjutnya dilanjutkan ke lapisan masukan. Oleh karena itu, metode ini dinamakan sebagai metode pelatihan *backpropagation*.



Gambar 2.19 Jaringan Syaraf Tiruan Persepton Lapis Jamak

2.8 Kontroler *Neuro-Fuzzy* (NF) [10]

Sistem hibrida yang menggabungkan logika *fuzzy*, jaringan syaraf tiruan, algoritma genetika, dan kecerdasan buatan lainnya sudah banyak dikembangkan untuk menyelesaikan berbagai masalah yang berkaitan dengan optimalisasi. Konsep hibrida dikembangkan karena pada metode kecerdasan buatan ini memiliki kelebihan dan kekurangan masing-masing, sehingga dengan konsep hibrida akan didapatkan suatu metode yang lebih baik dalam menyelesaikan permasalahan optimasi. Salah satu penggunaan konsep hibrida adalah pada *Neuro-Fuzzy* (NF) yang digunakan sebagai pengendali suatu sistem (kontroler).

Kontroler NF merupakan sistem hibrida yang menggabungkan konsep logika *fuzzy* dengan jaringan syaraf tiruan (JST). Logika *fuzzy* memiliki kelebihan dalam pengambilan keputusan, sedangkan JST memiliki kelebihan dalam adaptasi melalui kemampuan pembelajaran yang dimilikinya. Kekurangan logika *fuzzy* dalam hal penentuan parameter yang sifatnya intuitif dapat diatasi dengan JST. Pada kontroler NF, logika *fuzzy* akan direpresentasikan dalam JST yang memungkinkan terjadinya pembelajaran di dalamnya dan akan terjadi perubahan bobot

untuk mengubah parameter pada logika *fuzzy*. Berikut ini merupakan tabel perbandingan kemampuan logika *fuzzy* dan JST dalam penyelesaian suatu masalah.

Tabel 2.5 Perbandingan Logika *Fuzzy* dan Jaringan Syaraf Tiruan [10]

Masalah	Teknik Penyelesaian	
	Logika <i>Fuzzy</i>	Jaringan Syaraf Tiruan
Model matematika	Agak baik	Buruk
Kemampuan pembelajaran	Buruk	Baik
Representasi pengetahuan	Baik	Buruk
Pengetahuan pakar	Baik	Buruk
Non-liniaritas	Baik	Baik
Kemampuan optimasi	Buruk	Agak baik
Toleransi kegagalan	Baik	Baik
Toleransi ketidakpastian	Baik	Baik
Operasi waktu nyata	Baik	Agak baik

Teknik adaptasi kontroler NF berdasarkan metode pembelajaran *back propagation* pada JST. Terdapat beberapa struktur NF yang biasa dipakai sebagai kontroler, antara lain: *Adaptive Neuro Fuzzy Inference System (ANFIS)*, *Fuzzy Modelling Networks (FMN) Tipe I*, *FMN Tipe II*, dan *FMN Tipe III*. Perbedaan ketiga tipe FMN ini terletak pada konsekuennya. FMN ini akan mengidentifikasi aturan-aturan *fuzzy* dan fungsi keanggotaan secara otomatis dengan cara memodifikasi bobot-bobot jaringan syaraf melalui algoritma pembelajaran *backpropagation*.

2.8.1 FMN Tipe I [10]

Pada FMN Tipe I, aturan *fuzzy* memiliki konsekuen berupa variabel *fuzzy*. Format aturan untuk FMN tipe I adalah

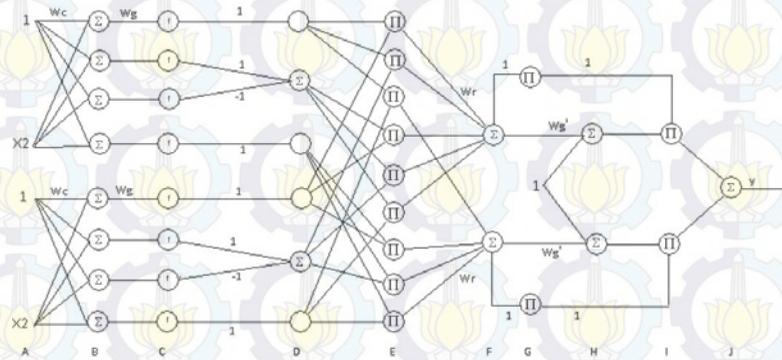
$$R_i^k = (IF x_1 \text{ is } A_{1i} \text{ AND } x_2 \text{ is } A_{2i} \text{ THEN } y \text{ is } B_k) \text{ is } \tau_{R_i^k}$$

$i = 1, 2, \dots, n$; $k = 1, 2$

$$\mu_k = \sum_i^n \hat{\mu}_i \tau_{R_i^k} \quad (2.27)$$

$$y^* = \frac{\sum_{k=1}^2 \mu'_k \mu_{B_k}^{-1}(\mu'_k)}{\sum_{k=1}^2 \mu'_k} \quad (2.28)$$

Dengan R_i^k adalah aturan *fuzzy* ke- i konsekuen ke- k , A_{1i} dan A_{2i} adalah himpunan *fuzzy* pada bagian IF (anteseden), B_k adalah himpunan *fuzzy* pada bagian konsekuen. $\tau_{R_i^k}$ adalah nilai kebenaran linguistik dari aturan R_i^k dalam bentuk konstanta yang bernilai pada rentang $[0 \ 1]$. μ'_k adalah nilai kebenaran pada konsekuen, dan $\mu_{B_k}^{-1}(\cdot)$ adalah fungsi invers dari fungsi keanggotaan pada konsekuen $\mu_{B_k}(y)$. Struktur jaringan FMN tipe I dapat dilihat pada Gambar 2.20.



Gambar 2.20 Struktur Jaringan FMN Tipe I

Pada Gambar 2.20, lapisan A sampai E merupakan bagian premis dan lapisan F sampai J merupakan bagian konsekuen. Bobot w_g dan w_c merupakan bobot yang menentukan nilai derajat keanggotaan dari *input* x . Bobot w_r merepresentasikan nilai $\tau_{R_i^k}$, sedangkan w_g' merepresentasikan gradien dari fungsi keanggotaan konsekuen. Selanjutnya, untuk melatih jaringan syaraf ini bisa digunakan algoritma *backpropagation* untuk perbaikan bobot w_c , w_g , w_r , w_c' , dan w_g' .

2.8.2 FMN Tipe II [10]

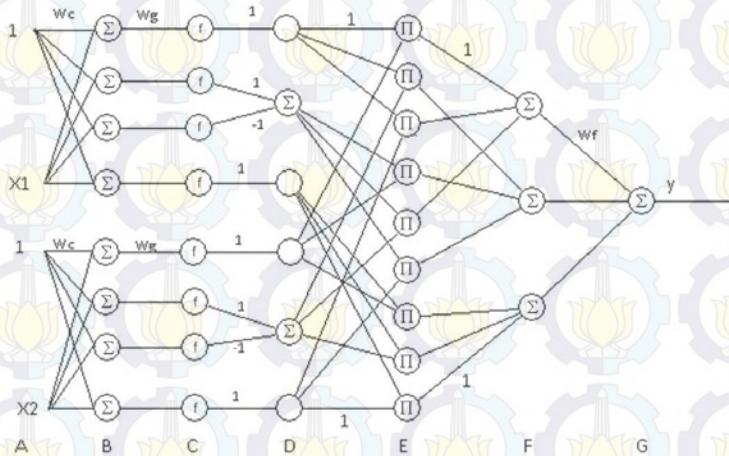
Pada FMN Tipe II, aturan *fuzzy* memiliki konsekuen berupa nilai tunggal (konstanta). Format aturan untuk FMN tipe II adalah

$R_i = \text{IF } x_1 \text{ is } A_{1i} \text{ AND } x_2 \text{ is } A_{2i} \text{ THEN } y \text{ is } f_i$

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

$$y^* = \frac{\sum_{i=1}^n \mu_i f_i}{\sum_{i=1}^n \mu_i} = \sum_{i=1}^n \hat{\mu}_i f_i \quad (2.29)$$

Dengan R_i adalah aturan *fuzzy* ke- i , A_{1i} dan A_{2i} adalah himpunan *fuzzy* pada bagian IF (anteseden), f_i adalah konstanta pada konsekuen, n adalah jumlah aturan, dan y^* adalah nilai inferensi. Struktur jaringan FMN tipe II dapat dilihat pada Gambar 2.21



Gambar 2.21 Struktur Jaringan FMN Tipe II

Pada Gambar 2.21, lapisan A sampai E merupakan bagian premis, sedangkan lapisan F merupakan bagian konsekuen. Bobot w_r dan w_g merupakan nilai yang menentukan derajat keanggotaan dari *input* x , sedangkan w_f menentukan nilai *output fuzzy*. Selanjutnya, untuk melatih jaringan syaraf ini bisa digunakan algoritma *backpropagation* untuk perbaikan bobot w_c , w_g , dan w_f .

2.8.3 FMN Tipe III [10]

Pada FMN tipe III, aturan *fuzzy* memiliki konsekuen berupa persamaan linear orde pertama. Format aturan untuk FMN tipe III adalah

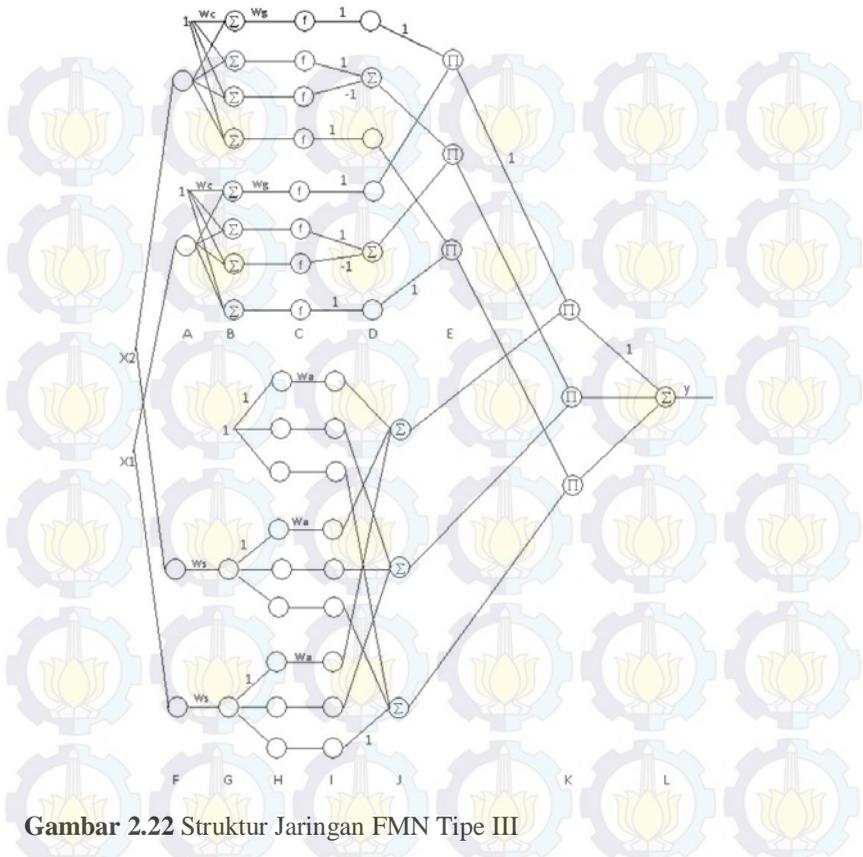
$$R_i = \text{IF } x_1 \text{ is } A_{1i} \text{ AND } x_2 \text{ is } A_{2i} \text{ THEN } y \text{ is } f(x_1, x_2)$$

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

$$y^* = \frac{\sum_{i=1}^n \mu_i f_i(x_1, x_2)}{\sum_{i=1}^n \mu_i} = \sum_{i=1}^n \hat{\mu}_i f_i(x_1, x_2) \quad (2.30)$$

$f_i(x_1, x_2) = a_{i0} + a_{i1}x_1 + a_{i2}x_2$; a_{ij} , $j = 0, 1, 2$ adalah konstanta

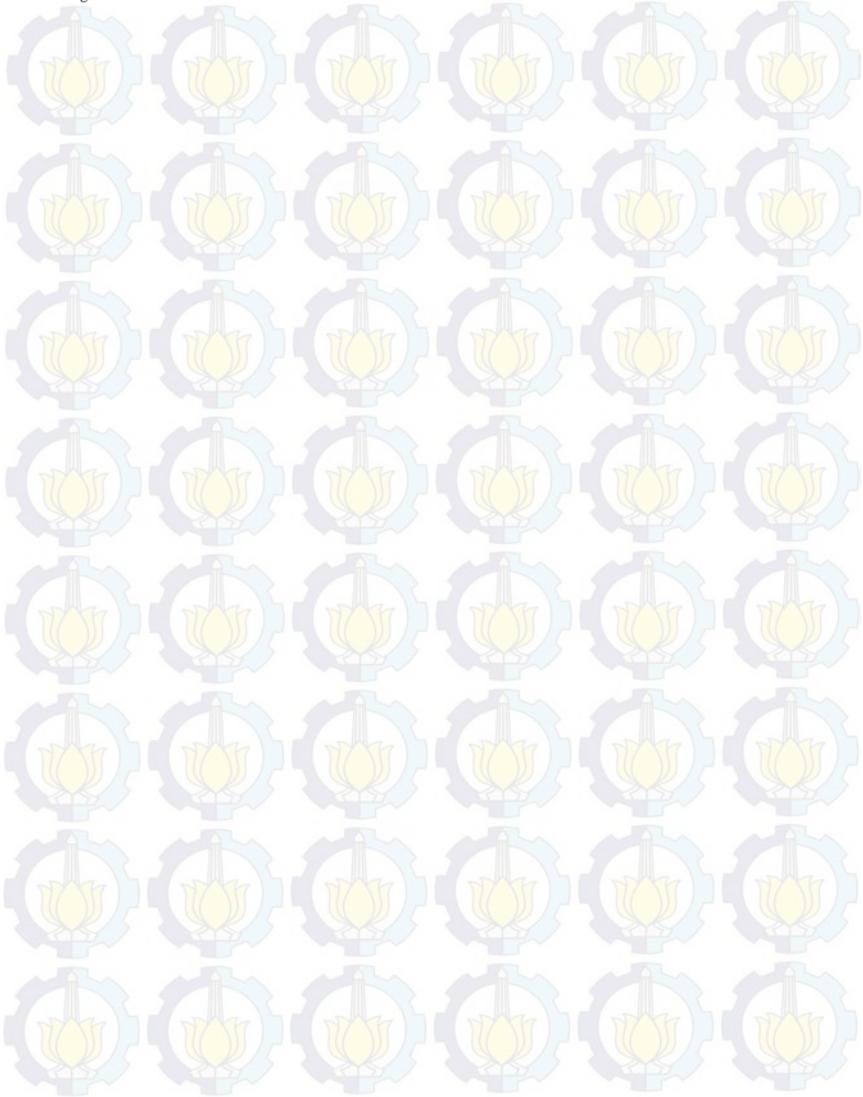
Dengan R_i adalah aturan *fuzzy* ke- i , A_{1i} dan A_{2i} adalah himpunan *fuzzy* pada bagian IF (anteseden), $f_i(x_1, x_2)$ adalah persamaan pada konsekuen, n adalah jumlah aturan, dan y^* adalah nilai inferensi. Struktur jaringan FMN tipe III dapat dilihat pada Gambar 2.22.



Gambar 2.22 Struktur Jaringan FMN Tipe III

Pada Gambar 2.22, lapisan A sampai E merupakan bagian premis, sedangkan lapisan F sampai L merupakan bagian konsekuen. Bobot w_r dan w_g merupakan nilai yang menentukan derajat keanggotaan dari *input* x . Bobot w_s merupakan faktor penskala untuk variabel *input* pada konsekuen. Nilai a_{ij} merupakan hasil perkalian antara w_s dan w_a . Hasil inferensi pada setiap aturan $f_i(x_1, x_2)$ dihitung sebagai *output* pada neuron lapisan J. Hasil perkalian antara $f_i(x_1, x_2)$ dan μ_i dihitung pada lapisan K. Penjumlahan dari perkalian tersebut dilakukan pada lapisan L sebagai hasil inferensi. Selanjutnya, untuk melatih jaringan syaraf ini

bisa digunakan algoritma *backpropagation* untuk perbaikan bobot w_c , w_g , w_s , dan w_a .

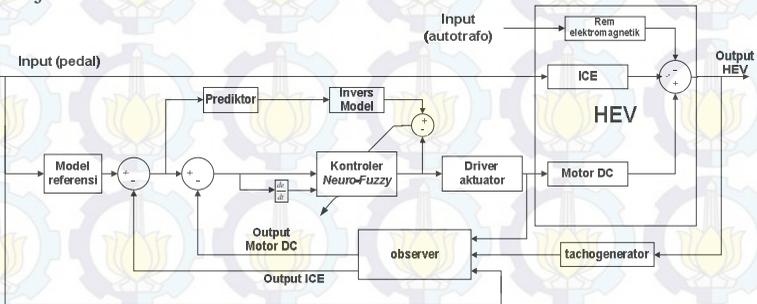




BAB 3 PERANCANGAN SISTEM

3.1 Diagram Blok Sistem

HEV merupakan suatu kendaraan dengan penggerak berupa gabungan ICE dan motor DC. Simulator HEV yang digunakan pada Tugas Akhir ini mengambil prinsip yang sama dengan HEV. Simulator HEV yang digunakan menggunakan konfigurasi paralel, dimana ICE bertindak sebagai penggerak utama sedangkan motor DC bertindak sebagai penggerak pembantu ketika proses akselerasi dan pemberian beban. Pada Tugas Akhir kali ini akan dibahas mengenai pengaturan kecepatan motor DC ketika proses akselerasi. Diagram blok sistem ditunjukkan oleh Gambar 3.1.



Gambar 3.1 Diagram Blok Sistem

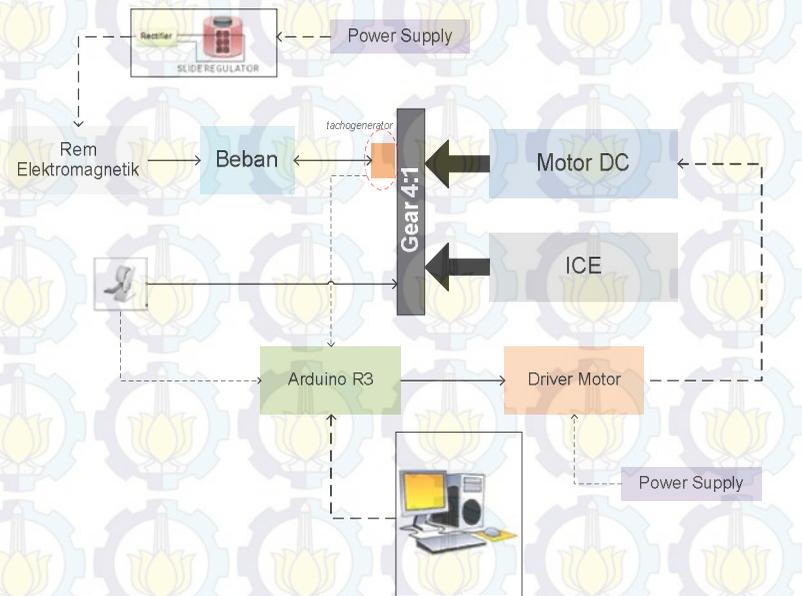
Model referensi yang digunakan merupakan karakteristik respon *plant* yang diinginkan. Model referensi harus memiliki *rise time* yang lebih cepat daripada *rise time* ICE dan tidak memiliki *error steady state*. Dengan adanya model referensi ini, diharapkan motor DC dapat membantu ICE agar dapat mencapai model referensi yang diinginkan.

3.2 Perancangan Perangkat Keras

Dalam Tugas Akhir ini digunakan dua jenis perangkat keras, yaitu perangkat mekanik dan perangkat elektronik.

3.2.1 Perancangan Mekanik [1][13]

Simulator HEV pada dasarnya menggabungkan tenaga dari ICE dan motor DC. Pada simulator HEV, poros ICE dan motor DC dipasang terpisah. Agar kedua motor ini dapat bekerja secara bersamaan maka ditambahkan suatu *belt* yang berfungsi untuk mereduksi kecepatan motor DC. Pedal gas merepresentasikan bukaan *throttle* pada karburator. Bukaan *throttle* karburator merupakan *input* dari sistem pada HEV. *Tachogenerator* yang dipasang pada poros motor DC merepresentasikan *output* dari sistem HEV. Pada simulator HEV, terdapat beban berupa rem elektromagnetik yang akan memberikan gaya berlawanan dengan arah putar motor. Diagram perangkat keras ditunjukkan pada Gambar 3.2.



Gambar 3.2 Diagram Perangkat Keras Simulator HEV

Pada simulator HEV ini, ICE berfungsi sebagai penggerak utama sedangkan motor DC berfungsi untuk membantu kinerja ICE. Kecepatan ICE lebih besar daripada motor DC, oleh karena itu

dibutuhkan suatu transmisi *gear* untuk menyamakan kecepatan putar dari keduanya.

Pada simulator HEV ini, diberikan juga rem elektromagnetik arus eddy. Rem ini akan menghasilkan gaya yang berlawanan dengan torsi pada HEV sehingga akan menyebabkan kecepatan putar HEV berkurang. Keseluruhan sistem simulator HEV ditunjukkan pada Gambar 3.3.



Gambar 3.3 Bentuk Fisik Simulator HEV

3.2.1.1 Internal Combustion Engine (ICE)

ICE pada simulator HEV berfungsi sebagai penggerak utama. Pada simulator HEV yang digunakan dalam Tugas Akhir ini digunakan mesin pemotong rumput. Mesin pemotong rumput diasumsikan sebagai ICE pada HEV yang sesungguhnya. Tabel spesifikasi mesin pemotong rumput yang digunakan ditunjukkan pada Tabel 3.1.

Tabel 3.1 Spesifikasi ICE

No.	Klasifikasi	Spesifikasi
1	Jenis	2 Tak
2	Bahan Bakar	Bensin dan Oli (25:1)
3	Kapasitas Silinder	33,8 cc
4	Pemasukan Bahan Bakar	Karburator
5	Sistem Starter	<i>Recoil Starter</i>
6	Sistem Kopling	Kering, Sentrifugal
7	Kecepatan Putar Maksimal	8000 rpm
8	Daya Keluaran	1,8 HP



Gambar 3.4 ICE 2 Tak

3.2.1.2 Motor DC

Motor listrik yang digunakan pada simulator HEV ini merupakan jenis motor *Direct Current (DC)*. Motor DC merupakan motor yang mengubah energi listrik DC menjadi energi gerak. Motor DC dipilih sebagai motor pembantu kerja ICE karena motor DC lebih mudah dikendalikan dibandingkan dengan motor *Alternating Current (AC)*. Pada Tugas Akhir ini, motor DC akan dikendalikan supaya dapat membantu kerja ICE pada proses akselerasi. Spesifikasi motor DC yang digunakan dapat dilihat pada Tabel 3.2. Bentuk fisik motor DC dapat dilihat pada Gambar 3.5.

Tabel 3.2 Spesifikasi Motor DC

No.	Klasifikasi	Spesifikasi
1	Tegangan Maksimal	190 V
2	Kecepatan Putar Maks.	1750 rpm
3	Tipe	Seri/Paralel
4	Daya Keluaran	0,5 HP



Gambar 3.5 Motor DC

3.2.1.3 Transmisi Gear

Pada simulator HEV ini, poros ICE dan motor DC terpasang secara terpisah. Oleh karena itu, untuk menyatukan kedua poros tersebut dibutuhkan pemasangan *belt* dan *gear*. Transmisi *gear* ini dibutuhkan untuk mereduksi kecepatan putar ICE agar sesuai *range* dari motor DC. Pemasangan *gear* dibutuhkan untuk membuat poros baru sehingga kecepatan putar pada poros akan sama.

Range kerja ICE dibatasi antara 0 – 8000 rpm sedangkan *range* kerja motor DC 0 – 2000 rpm, maka digunakan perbandingan *gear* dengan rumus:

$$\begin{aligned}\omega_A R_A &= \omega_B R_B \\ \frac{\omega_A}{\omega_B} &= \frac{R_B}{R_A} \\ \frac{2000}{8000} &= \frac{R_B}{R_A} \\ \frac{R_A}{R_B} &= \frac{4}{1}\end{aligned}\tag{3.1}$$

di mana:

ω_A : kecepatan putar motor DC

ω_B : kecepatan putar ICE

R_A : jari-jari pada poros motor DC

R_B : jari-jari pada poros ICE

Berdasarkan Persamaan (3.1) terlihat bahwa untuk menyamakan kecepatan putar yang sama antara ICE dan motor DC diperlukan perbandingan *gear* 1:4. Untuk efisiensi tempat, maka perbandingan *gear* dibagi menjadi dua dengan memberikan *gear* tambahan di antara keduanya seperti terlihat pada Gambar 3.6.



Gambar 3.6 Transmisi Gear

Dimisalkan gear pada poros motor bakar R_B dan poros transfer gear R_{C1} dengan perbandingan gear $R_B : R_{C1} = 1 : 2$.

$$\omega_B R_B = \omega_{C1} R_{C1}$$

$$\omega_A R_A = \omega_{C2} R_{C2}$$

$$\omega_{C1} = \omega_{C2}$$

$$\frac{\omega_A R_A}{R_{C2}} = \frac{\omega_B R_B}{R_{C1}}$$

$$\frac{\omega_A}{\omega_B} = \frac{R_B}{R_{C1}} \times \frac{R_{C2}}{R_A}$$

$$\frac{\omega_A}{\omega_B} = \frac{1}{2} \times \frac{1}{2} = \frac{1}{4} \quad (3.2)$$

di mana:

ω_{C1} : kecepatan putar gear 1 pada transfer gear

ω_{C2} : kecepatan putar gear 2 pada transfer gear

R_{C1} : jari-jari putar gear 1 pada transfer gear

R_{C2} : jari-jari putar gear 2 pada transfer gear

Berdasarkan Persamaan (3.2), dapat dilihat bahwa apabila ω_A diberikan nilai 2000 rpm maka akan didapatkan nilai ω_B sebesar 8000 rpm. Dapat disimpulkan bahwa dengan menggunakan konfigurasi seperti ini akan didapatkan nilai reduksi yang sama dengan menggunakan perbandingan radius secara langsung.

Selain kecepatan putar, torsi pada ICE dan motor DC akan mengalami perubahan dengan adanya reduksi ini. Hubungan antara

perbandingan torsi dengan jari-jari *gear* dapat dilihat pada Persamaan (3.3)

$$\begin{aligned}\frac{\tau_A}{R_A} &= \frac{\tau_B}{R_B} \\ \frac{\tau_A}{\tau_B} &= \frac{\omega_B}{\omega_A} \\ \frac{\tau_A}{\tau_B} &= \frac{8000}{2000} \\ \tau_A &= 4\tau_B\end{aligned}\quad (3.3)$$

Berdasarkan Persamaan (3.3) dapat disimpulkan bahwa poros utama akan memiliki torsi sebesar empat kali torsi yang dihasilkan ICE.

3.2.1.4 Sensor Posisi

Pada simulator HEV, *set point* yang diberikan berupa pedal gas. Pedal gas dihubungkan langsung ke karburator untuk menentukan bukaan *throttle* pada karburator. Pijakan pedal gas sebanding dengan bukaan *throttle* karburator. Oleh karena itu, untuk mengetahui bukaan *throttle* pada karburator dapat dilihat melalui pemberian sensor posisi pada pedal gas.

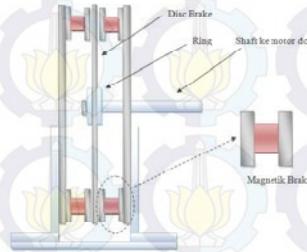
Perubahan derajat sudut akibat pijakan pedal gas merepresentasikan besar kecilnya bukaan *throttle* pada karburator. Perbandingan *gear* 4:1 ditempatkan pada poros potensiometer dan poros pedal gas agar potensiometer dapat mendeteksi semua perubahan sudut yang terjadi ketika pedal gas ditekan. Setiap satu derajat perubahan sudut pedal gas akan menghasilkan perubahan empat derajat pada potensiometer. Bentuk fisik sensor posisi pada HEV dapat dilihat pada Gambar 3.7.



Gambar 3.7 Bentuk Fisik Sensor Posisi Pada Pedal Gas

3.2.1.5 Rem Elektromagnetik

Rem elektromagnetik pada simulator HEV ini berfungsi sebagai beban. Rem elektromagnetik diletakkan pada poros motor DC. Rem elektromagnetik terdiri dari dua piringan besi yang diberi konduktor berupa kumparan yang dililit seperti Gambar 3.8.



Gambar 3.8 Rem Elektromagnetik

Ketika rem elektromagnetik diletakkan pada poros yang berputar dan dialiri arus listrik akan menghasilkan gaya yang berlawanan dengan arah putar poros. Hal ini akan menyebabkan terjadinya penurunan kecepatan putar pada motor DC. Gaya inilah yang disebut sebagai beban bagi simulator HEV. Bentuk fisik rem elektromagnetik dapat dilihat pada Gambar 3.8.



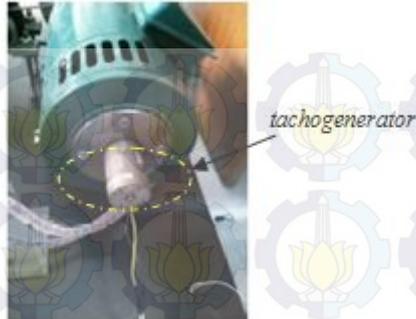
Gambar 3.9 Bentuk Fisik Rem Elektromagnetik

3.2.1.6 Tachogenerator

Sensor yang digunakan untuk menghitung kecepatan putar pada ICE dan motor DC adalah *tachogenerator*. *Tachogenerator* akan menghasilkan tegangan DC pada terminal keluaran yang nilainya proporsional dengan kecepatan putar pada rotor. Spesifikasi sensor *tachogenerator* ditunjukkan pada Tabel 3.3. Sedangkan gambar fisik *tachogenerator* ditunjukkan pada Gambar 3.10.

Tabel 3.3 Spesifikasi *Tachogenerator*

No.	Klasifikasi	Spesifikasi
1.	Tipe	<i>Brush</i>
2.	Torsi	29,9 Nm
3.	Konstruksi	Magnet Permanen
4.	Kecepatan Maksimal	13570 Rpm
5.	Tegangan Kerja	6 - 24 V
6.	Daya Keluaran	7 - 35 W



Gambar 3.10 Bentuk Fisik *Tachogenerator*

3.2.2 Perancangan Elektronik

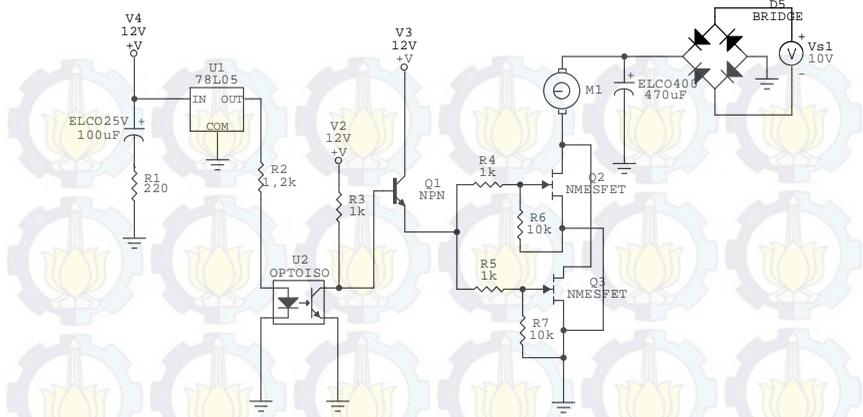
Perancangan elektronik dibutuhkan pada simulator HEV untuk pengaturan kecepatan pada motor DC. Pengaturan kecepatan ini bertujuan untuk membantu kerja ICE pada saat akselerasi dan pemberian beban.

3.2.2.1 Rangkaian *Driver* Motor DC

Rangkaian *driver* motor DC berfungsi sebagai aktuator pada sistem HEV. Rangkaian ini berfungsi untuk mengubah sinyal kontrol tegangan menjadi tegangan yang sesuai dengan tegangan kerja motor DC. Tegangan masukan *driver* motor DC berupa tegangan *auto trafo* yang sudah disearahkan. Untuk menghasilkan tegangan yang sesuai dengan yang dibutuhkan, diperlukan *Pulse Width Modulation (PWM)* dari sistem minimum Arduino. Bentuk fisik rangkaian *driver* motor DC dapat dilihat pada Gambar 3.11.



Gambar 3.11 Bentuk Fisik Rangkaian *Driver* Motor DC



Gambar 3.12 Rangkaian *Driver Motor DC*

3.2.2.2 Sistem Minimum Arduino Uno R3 [7]

Sistem minimum Arduino merupakan mikrokontroler berbasis Atmega 328. Sistem minimum Arduino Uno memiliki 14 *input/output* pin (6 pin sebagai *PWM output*), 6 *input* analog, 16 MHz *ceramic resonator*, koneksi *USB*, *jack power*, *ICSP header*, dan tombol *reset*. Pada simulator HEV ini, sistem minimum Arduino digunakan untuk membaca analog *input* dari potensiometer dan *tachogenerator*. Pembacaan dari potensiometer akan merepresentasikan sinyal masukan pada simulator HEV, sedangkan pembacaan dari *tachogenerator* merepresentasikan kecepatan putar pada simulator HEV. Selain itu, sistem minimum Arduino juga digunakan untuk mengeluarkan sinyal *output* PWM sebagai sinyal kontrol dari motor DC. Pengaturan *duty cycle* pada PWM disesuaikan dengan sinyal kontrol yang dibutuhkan oleh motor DC.

Pengiriman data dari sistem minimum Arduino ke PC/laptop dilakukan melalui *USB*. Pemrograman pada Arduino menggunakan bahasa C sehingga lebih mudah dipahami oleh pengguna. Bentuk fisik sistem minimum Arduino dapat dilihat pada Gambar 3.13.



Gambar 3.13 Bentuk Fisik Sistem Minimum Arduino Uno R3

3.2.2.3 Rangkaian Pembagi Arus-Tegangan

Identifikasi respon dari sensor *tachogenerator* dilakukan dengan menggunakan mikrokontroler Arduino. Mikrokontroler Arduino digunakan agar data yang diambil dari sensor *tachogenerator* dapat disimpan di dalam komputer/laptop. Mikrokontroler Arduino hanya dapat membaca sinyal analog pada *range* 0-5 V, sedangkan tegangan keluaran *tachogenerator* berada pada *range* 0 - 7,8 V. Oleh karena itu, dibutuhkan suatu rangkaian pembagi arus-tegangan yang dapat mengubah *range* tegangan 7,8 V menjadi 5 V.

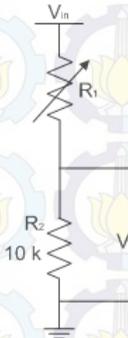
Rangkaian pembagi arus-tegangan terdiri dari satu buah resistor dan satu buah potensiometer. Potensiometer digunakan untuk mengubah-ubah nilai resistansi agar didapatkan nilai tegangan keluaran sebesar 5 V. Besarnya nilai resistansi dapat dicari melalui Persamaan (3.4), sedangkan skema rangkaian pembagi arus-tegangan dapat dilihat pada Gambar 3.14. Bentuk fisik rangkaian pembagi arus-tegangan dapat dilihat pada Gambar 3.15.

$$V_{out} = \frac{R_2}{R_1 + R_2} \times V_{in} \quad (3.4)$$

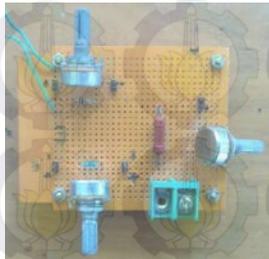
Jika dipilih nilai R_2 sebesar 10 k Ω , maka nilai R_1 agar didapatkan nilai tegangan keluaran 5 V adalah

$$5 = \frac{10k}{R_1 + 10k} \times 7,8$$

$$R_1 = 5,6 k$$



Gambar 3.14 Skema Rangkaian Pembagi Arus-Tegangan



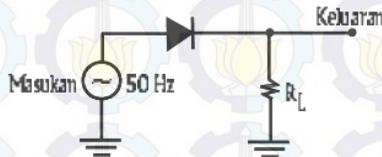
Gambar 3.15 Bentuk Fisik Rangkaian Pembagi Arus-Tegangan

3.2.2.4 Rangkaian Penyearah Gelombang Penuh

Pengoperasian motor DC pada simulator HEV membutuhkan sumber tegangan 0 – 190 VDC. Sumbertegangan tersebut didapatkan dari tegangan jala-jala PLN yang dihubungkan dengan *autotrafo* kemudian diubah menjadi tegangan DC menggunakan rangkaian penyearah (*rectifier*). Rangkaian penyearah menggunakan dioda sebagai komponen utama. Dioda memiliki karakteristik *short circuit* ketika *forward bias* dan *open circuit* ketika *reverse bias*. Karakteristik ini yang digunakan sebagai dasar dalam rangkaian penyearah. Terdapat beberapa

macam rangkaian penyearah yang dapat digunakan, antara lain: rangkaian penyearah setengah gelombang dan rangkaian penyearah gelombang penuh.

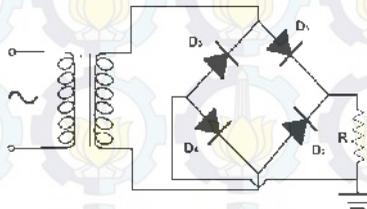
Pada rangkaian penyearah setengah gelombang, sumber tegangan AC dihubungkan ke resistor melalui sebuah dioda. Skema rangkaian penyearah setengah gelombang dapat dilihat pada Gambar 3.16.



Gambar 3.16 Skema Rangkaian Penyearah Setengah Gelombang

Ketika tegangan masukan bernilai positif maka dioda akan berada pada posisi *forward bias* sehingga tegangan keluaran akan bernilai positif. Sedangkan ketika tegangan masukan bernilai negatif maka posisi dioda berada pada posisi *reverse bias* sehingga tegangan keluaran akan bernilai nol.

Rangkaian penyearah gelombang penuh yang banyak dijual di pasaran terdiri dari empat buah dioda yang dirangkai seperti pada Gambar 3.17.



Gambar 3.17 Skema Rangkaian Penyearah Gelombang Penuh

Ketika tegangan masukan bernilai positif, maka dioda D1 dan D4 berada pada posisi *forward bias* sehingga tegangan keluaran pada R bernilai positif. Ketika tegangan masukan bernilai negatif, maka dioda D2 dan D3 berada pada posisi *forward bias* sehingga tegangan keluaran pada R akan bernilai positif. Tegangan keluaran rangkaian penyearah tidak sepenuhnya berupa tegangan DC murni melainkan tegangan DC yang berubah secara periodik. Bentuk fisik rangkaian penyearah gelombang penuh dapat dilihat pada Gambar 3.18.



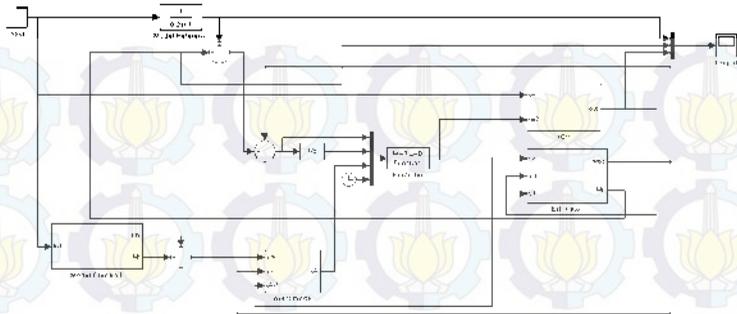
Gambar 3.18 Bentuk Fisik Rangkaian Penyearah Gelombang Penuh

3.3 Perancangan Perangkat Lunak

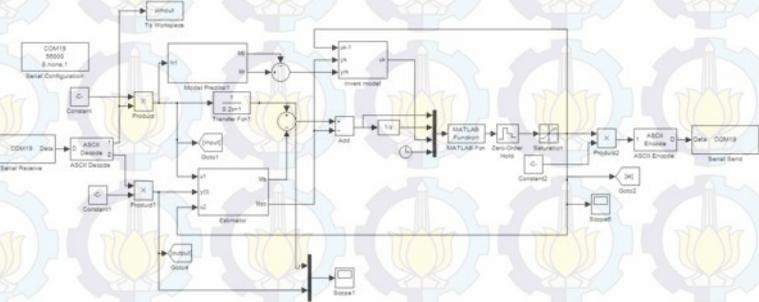
Perangkat lunak diperlukan dalam perancangan sistem sebagai *interface* antara *plant* dan komputer. Perangkat lunak yang digunakan adalah Matlab 2010b. Perangkat lunak ini digunakan untuk pengambilan data identifikasi *plant*, simulasi kontroler dan implementasi kontroler pada *plant*. Proses identifikasi dan implementasi dilakukan dengan menggunakan fasilitas komunikasi *serial* pada Matlab 2010b.

3.3.1 Matlab

Perancangan kontroler pada simulator HEV digunakan dengan menggunakan bahasa pemrograman dan *toolbox* Simulink pada perangkat lunak Matlab. Simulink digunakan untuk mensimulasikan kontroler pada model matematis yang merepresentasikan *plant* motor DC. Simulink juga dilengkapi fasilitas komunikasi dengan mikrokontroler Arduino, sehingga proses implementasi kontroler dapat dilakukan melalui *toolbox* Simulink. Pada Gambar 3.19 dapat dilihat blok Simulink simulator HEV.



Gambar 3.19 Blok Simulink Perancangan Kontroler *Neuro Fuzzy* Prediktif Pada Simulator HEV



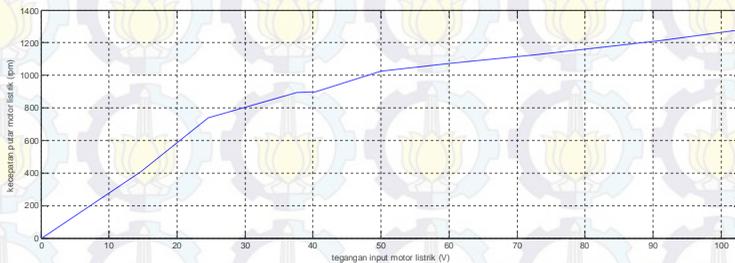
Gambar 3.20 Blok Simulink Implementasi Kontroler *Neuro Fuzzy* Prediktif Pada Simulator HEV

3.4 Pengujian *Open Loop* Kecepatan Motor DC

Pengujian *Open Loop* Kecepatan Motor DC bertujuan untuk mengetahui hubungan antara tegangan *input* dengan kecepatan putar poros motor DC. Pengujian dilakukan tanpa memberikan beban berupa rem elektromagnetik. Tabel hubungan antara tegangan *input* dan kecepatan putar dapat dilihat pada Tabel 3.4, sedangkan grafik hubungan antara tegangan *input* dan kecepatan putar dapat dilihat pada Gambar 3.21.

Tabel 3.4 Hubungan Antara Tegangan *Input* dan Kecepatan Putar Motor DC

Tegangan DC <i>Input</i> (V)	Kecepatan Putar (rpm)
14,7	406,5639
24,6	739,0251
37,6	893,239
40,4	897,2445
50	1025,422
60,4	1073,489
72,2	1125,561
82,6	1171,625
91,2	1215,686
102,1	1273,767



Gambar 3.21 Hubungan Antara Tegangan *Input* dan Kecepatan Putar

3.5 Pengujian Sensor

Simulator HEV pada Tugas Akhir kali ini menggunakan dua buah sensor, yaitu sensor *tachogenerator* dan sensor potensiometer. Sensor *tachogenerator* digunakan untuk merepresentasikan putaran dari HEV. Sedangkan sensor potensiometer digunakan untuk merepresentasikan bukaan *throttle* pada karburator. Pengujian sensor digunakan untuk mengetahui nilai-nilai keluaran sensor yang akan dikonversikan menjadi putaran maupun bukaan *throttle* pada karburator.

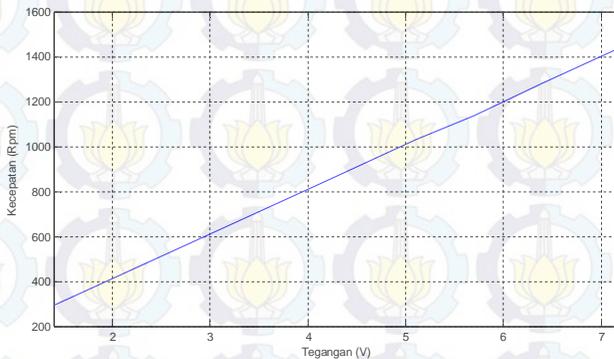
3.5.1 Pengujian Sensor *Tachogenerator*

Tachogenerator merupakan sensor yang digunakan untuk menghitung kecepatan putar motor DC maupun ICE dengan cara memasang *tachogenerator* pada poros motor DC. Pengujian pada sensor *tachogenerator* dilakukan dengan cara membandingkan antara nilai *output* tegangan pada *tachogenerator* dengan kecepatan putar sesungguhnya yang diukur dengan menggunakan tachometer digital pada poros motor DC.

Berdasarkan hasil pengujian didapatkan hubungan antara nilai keluaran *tachogenerator* dengan kecepatan putar pada poros motor DC. Data hasil pengujian *tachogenerator* ditunjukkan pada Tabel 3.5.

Tabel 3.5 Data Pengujian Sensor *Tachogenerator*

No	Tegangan <i>tachogenerator</i> (V)	Kecepatan (rpm)
1	1,4	296
2	4,2	849
3	5,1	1029
4	5,7	1140
5	6,4	1282
6	7,2	1442



Gambar 3.22 Hubungan Antara Kecepatan Putar Poros Motor DC dengan Tegangan *Tachogenerator*

Berdasarkan data pengujian hubungan *tachogenerator* dengan kecepatan putar motor DC, dilakukan linierisasi menggunakan metode interpolasi sehingga didapatkan hasil pada Persamaan (3.5)

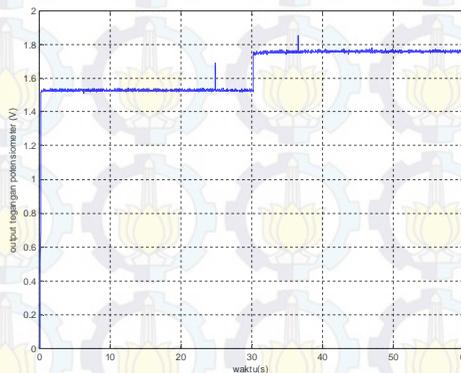
$$f(x) = 200,2778x \quad (3.5)$$

3.5.2 Pengujian Sensor Potensiometer

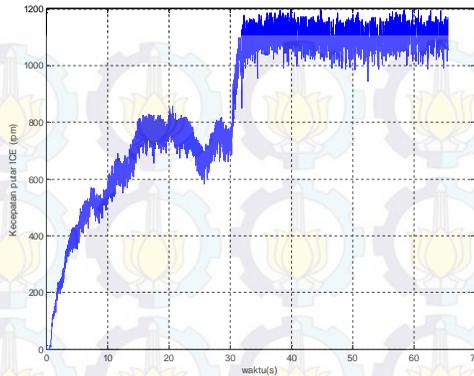
Input dari sistem HEV direpresentasikan dalam pijakan pedal. Sensor potensiometer dipasang untuk mengetahui derajat bukaan *throttle* pada karburator. Pengujian sensor potensiometer dilakukan dengan cara memberikan tegangan pada potensiometer kemudian dihubungkan ke Arduino. Data dari Arduino akan dikirimkan ke PC untuk melihat grafik pijakan pedal dan grafik kecepatan putar ICE. Grafik pijakan pedal dapat dilihat pada Gambar 3.23, sedangkan grafik kecepatan putar ICE dapat dilihat pada Gambar 3.24.

Tabel 3.6 Hubungan Tegangan Potensiometer dengan Kecepatan Putar ICE

Pijakan pedal (%)	Tegangan potensiometer (V)	Output ICE (rpm)
0	0	0
75	1,52	760,6
100	1,75	1095



Gambar 3.23 Hubungan Pijakan Pedal terhadap Tegangan Potensiometer



Gambar 3.24 Respon Kecepatan Putar ICE

3.6 Proses Identifikasi dan Permodelan Sistem

Proses identifikasi bertujuan untuk mencari parameter-parameter dari *plant*. Parameter-parameter yang sudah didapatkan digunakan untuk membuat model matematis dari sistem. Pada Tugas Akhir ini, identifikasi ICE dilakukan menggunakan metode *Strejc* sedangkan identifikasi motor DC menggunakan metode analitis.

3.6.1 Identifikasi ICE [7]

Identifikasi ICE dilakukan dengan identifikasi *open loop* respon ICE dengan memberikan sinyal uji *step* dan dilakukan identifikasi menggunakan metode *Strejc*. Identifikasi ICE dilakukan pada tiga kondisi, yaitu pada saat diberi beban minimal, nominal, dan maksimal.

3.6.1.1 Metode Pembebanan Minimal

Metode pembebanan minimal dilakukan dengan melihat respon *open loop* ICE tanpa diberi beban tambahan berupa rem elektromagnetik. Pada metode pembebanan minimum, motor DC yang telah dihubungkan dengan ICE melalui *gear* dapat dianggap sebagai beban minimal bagi ICE itu sendiri.

Pengujian dan validasi respon *open loop* ICE (dapat dilihat pada Gambar 3.26) dilakukan dengan identifikasi menggunakan metode *Strejc* sebagaimana pada perhitungan dibawah ini:

$$Y_{ss} = 1178$$

$$X_{ss} = 914$$

$$K = \frac{Y_{ss}}{X_{ss}} = \frac{1178}{914} = 1,289$$

Dari grafik respon pada Gambar 3.24, dan dengan mencari parameter respon seperti pada Gambar 2.17, maka didapat :

$$T_u = 0,0953$$

$$T_N = 0,7647$$

Maka nilai parameter τ didapatkan melalui Persamaan (2.14)

$$\tau = \frac{T_u}{T_N} = \frac{0,0953}{0,7647} = 0,125$$

Berdasarkan nilai τ , dapat dilakukan pendekatan model orde ke-n seperti pada Persamaan (2.13) dengan perhitungan sebagai berikut:

- a. Pada nilai τ dan Tabel 2.4 didapat bahwa orde untuk pendekatan model adalah orde-2 dan ditentukan koordinat $y_i = 0,264$ yang merupakan titik singgung antara respon sistem dengan garis singgung.
- b. Tentukan parameter t_i sesuai dengan mencari waktu saat respon mencapai 0,264 dari Y_{ss} .

$$y_i = 0,264(1178)$$

$$y_i = 310,992$$

Waktu saat respon mencapai y_i adalah

$$t_i = 0,3118$$

- c. Tentukan waktu konstan τ_{ST} menurut Persamaan (2.15)

$$\tau_{ST} = \frac{0,3118}{2 - 1}$$

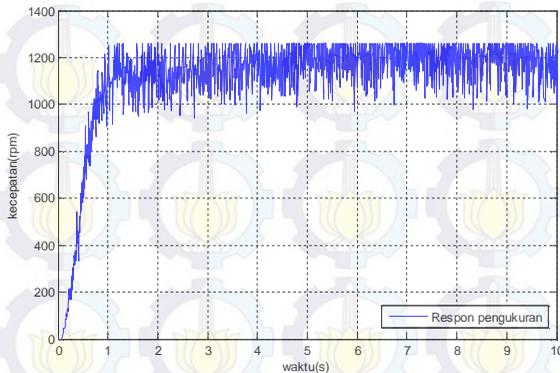
$$\tau_{ST} = 0,3118 \quad (3.6)$$

- d. Dari Persamaan (3.6) didapatkan pendekatan model orde ke-n seperti Persamaan (2.9) sehingga didapat model representasi sistem seperti ditunjukkan pada Persamaan (3.7)

$$G_{ST2}(s) = \frac{K}{(\tau_{ST}S + 1)^n} \quad (2.13)$$

$$G(s) = \frac{1}{(0,3118 + 1)^2} \quad (3.7)$$

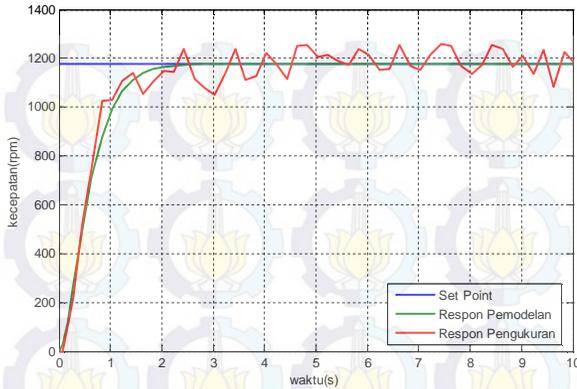
Pemodelan terhadap respon sistem dilakukan beberapa kali sebagai uji validasi terhadap nilai RMSE. Pengujian respon dapat dilihat pada Tabel 3.7 dan Gambar 3.26.



Gambar 3.25 Respon Kecepatan Putar ICE dengan Beban Minimal

Tabel 3.7 Pemodelan dan Validasi *Plant* ICE dengan Beban Minimal

No	Fungsi AlihPlant	RRMSE(%)
1	$G(s) = \frac{1,29}{(0,3118s + 1)^2}$	4,39
2	$G(s) = \frac{1,31}{(0,3133s + 1)^2}$	4,49
3	$G(s) = \frac{1,302}{(0,3128s + 1)^2}$	4,37
4	$G(s) = \frac{1,27}{(0,3107s + 1)^2}$	4,61
5	$G(s) = \frac{1,25}{(0,3088s + 1)^2}$	5,463



Gambar 3.26 Hasil Identifikasi dan Permodelan ICE dengan Beban Minimal

3.6.1.2 Metode Pembebanan Nominal

Metode pembebanan nominal dilakukan dengan memberi beban pada ICE berupa rem elektromagnetik. Besarnya tegangan yang diberikan pada rem elektromagnetik ditentukan dengan ketentuan ICE masih dapat mencapai *set point* awal sebelum adanya beban saat *throttle* dibuka penuh. Setelah melakukan beberapa percobaan, maka ditentukan bahwa beban nominal adalah beban dengan tegangan *input* pada rem elektromagnetik sebesar 56,3VDC.

Pengujian dan validasi respon *open loop* ICE dilakukan dengan identifikasi menggunakan metode Strejc pada perhitungan di bawah ini:

$$\begin{aligned}
 Y_{SS} &= 914 \\
 X_{SS} &= 914 \\
 K &= \frac{Y_{SS}}{X_{SS}} = \frac{914}{914} = 1
 \end{aligned}$$

Dari grafik respon pada Gambar 3.25, dan dengan mencari parameter respon seperti pada Gambar 2.17, maka didapat :

$$\begin{aligned}
 T_u &= 0,06133 \\
 T_N &= 0,5843
 \end{aligned}$$

Maka nilai parameter τ dapat dengan Persamaan (2.14)

$$\tau = \frac{T_U}{T_N} = \frac{0,06133}{0,5843} = 0,1049$$

Berdasarkan nilai τ , dapat dilakukan pendekatan model orde ke-n seperti pada Persamaan (2.4) dengan perhitungan sebagai berikut:

- a. Pada nilai τ dan Tabel 2.4 didapat bahwa orde untuk pendekatan model yang paling mendekati adalah orde-2 dan ditentukan koordinat $y_i = 0,264$ yang merupakan titik singgung antara respon sistem dengan garis singgung.
- b. Tentukan parameter t_i sesuai dengan mencari waktu saat respon mencapai 0,264 dari Y_{ss} .

$$y_i = 0,264(914)$$

$$y_i = 241,296$$

Waktu saat respon mencapai y_i adalah

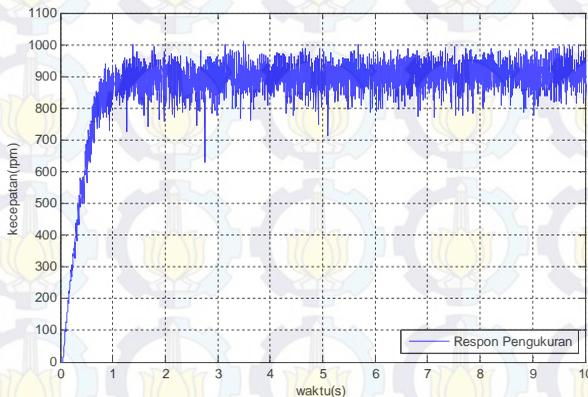
$$t_i = 0,1848$$

- c. Tentukan waktu konstan τ_{ST} menurut Persamaan (2.15)

$$\tau_{ST} = \frac{0,1848}{2 - 1}$$

$$\tau_{ST} = 0,1848$$

(3.8)



Gambar 3.27 Respon Kecepatan Motor Bakar dengan Beban Nominal

- d. Dari Persamaan (3.8) didapat pendekatan model orde ke-n seperti Persamaan (2.15) sehingga didapat Persamaan (3.9) sebagai berikut:

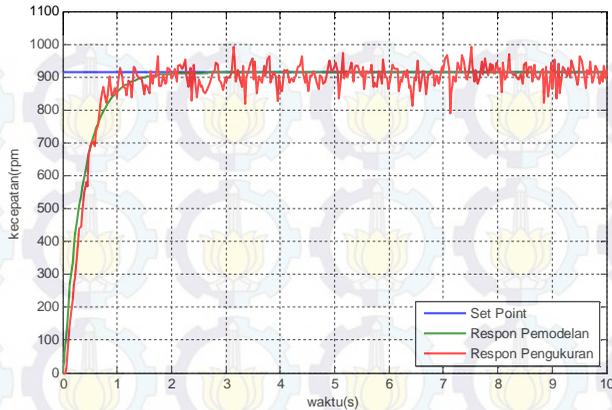
$$G_{ST2}(s) = \frac{K}{(\tau_{ST} s + 1)^n} \quad (2.15)$$

$$G(s) = \frac{1}{(0,1848 + 1)^2} \quad (3.9)$$

Pemodelan terhadap respon sistem dilakukan beberapa kali sebagai uji validasi terhadap nilai RMSE. Pengujian respon dapat dilihat pada Tabel 3.8 dan Gambar 3.28.

Tabel 3.8 Pemodelan dan Validasi *Plant* ICE dengan Beban Nominal

No	Fungsi Alih <i>Plant</i>	RRMSE (%)
1	$G(s) = \frac{0,946}{(0,17s + 1)^2}$	5,397
2	$G(s) = \frac{0,974}{(0,18s + 1)^2}$	3,852
3	$G(s) = \frac{0,996}{(0,1843s + 1)^2}$	3,576
4	$G(s) = \frac{1}{(0,1848s + 1)^2}$	3,208
5	$G(s) = \frac{1,01}{(0,1858s + 1)^2}$	3,717



Gambar 3.28 Hasil Identifikasi dan Permodelan ICE dengan Beban Nominal

3.6.1.3 Metode Pembebanan Maksimal

Metode pembebanan maksimal dilakukan dengan memberi beban pada ICE berupa rem elektromagnetik. Besarnya tegangan yang diberikan pada rem elektromagnetik ditentukan dengan ketentuan ICE sudah tidak dapat mencapai *set point* awal sebelum adanya beban saat *throttle* dibuka penuh. Setelah melakukan beberapa percobaan, maka ditentukan bahwa beban maksimal adalah beban dengan tegangan *input* pada rem elektromagnetik sebesar 82,3VDC.

Pengujian dan validasi respon *open loop* ICE dilakukan dengan identifikasi menggunakan metode Strejc pada perhitungan dibawah ini:

$$\begin{aligned}
 Y_{ss} &= 548,9 \\
 X_{ss} &= 914 \\
 K &= \frac{Y_{ss}}{X_{ss}} = \frac{548,9}{914} = 0,6
 \end{aligned}$$

Dari grafik respon pada Gambar 3.29, dan dengan mencari parameter respon seperti pada Gambar 2.17, maka didapat :

$$\begin{aligned}
 T_u &= 0,104 \\
 T_N &= 0,891
 \end{aligned}$$

Maka nilai parameter τ dapat dengan Persamaan (2.14)

$$\tau = \frac{T_U}{T_N} = \frac{0,104}{0,891} = 0,1167$$

Berdasarkan nilai τ , dapat dilakukan pendekatan model orde ke-n seperti pada Persamaan 2.14 dengan perhitungan sebagai berikut:

- a. Pada nilai τ dan Tabel 2.4 didapat bahwa orde untuk pendekatan model yang paling mendekati adalah orde-2 dan ditentukan koordinat $y_i = 0,264$ yang merupakan titik singgung antara respon sistem dengan garis singgung.
- b. Tentukan parameter t_i sesuai dengan mencari waktu saat respon mencapai 0,264 dari Y_{ss} .

$$y_i = 0,264(548,9)$$

$$y_i = 144,9$$

t saat respon mencapai y_i adalah

$$t_i = 0,289$$

- c. Tentukan waktu konstan τ_{ST} menurut Persamaan (2.15)

$$\tau_{ST} = \frac{0,289}{2 - 1}$$

$$\tau_{ST} = 0,289$$

(3.10)

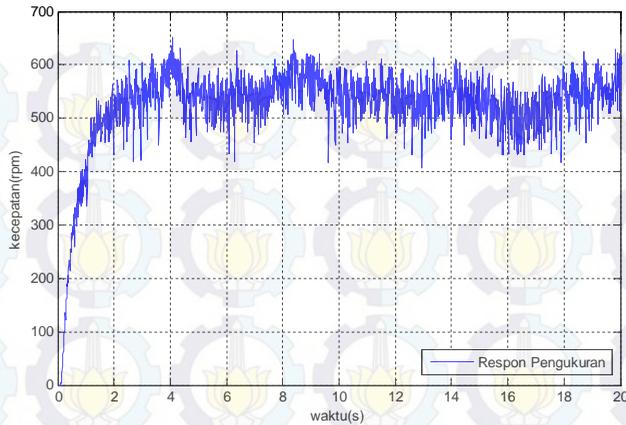
- d. Dari Persamaan (3.10) didapat pendekatan model orde ke-n seperti Persamaan (2.15) sehingga didapat Persamaan (3.11) sebagai berikut:

$$G_{ST2}(s) = \frac{K}{(\tau_{ST}s + 1)^n}$$

$$G(s) = \frac{1}{(0,289s + 1)^2}$$

(3.11)

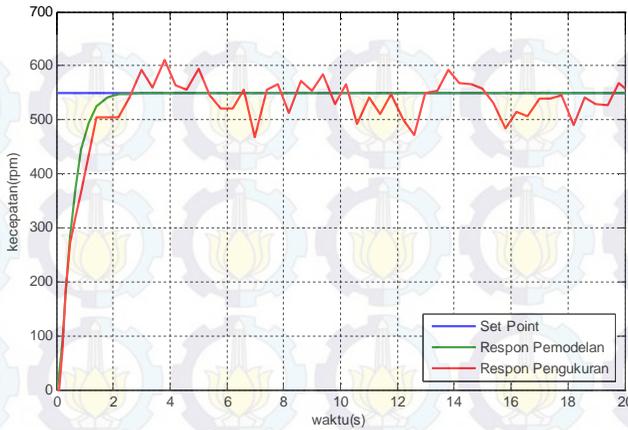
Pemodelan terhadap respon sistem dilakukan beberapa kali sebagai uji validasi terhadap nilai RMSE. Pengujian respon dapat dilihat pada Tabel 3.9 dan Gambar 3.28.



Gambar 3.29 Respon Kecepatan Putar ICE dengan Beban Maksimal

Tabel 3.9 Pemodelan dan Validasi *Plant* ICE dengan Beban Maksimal

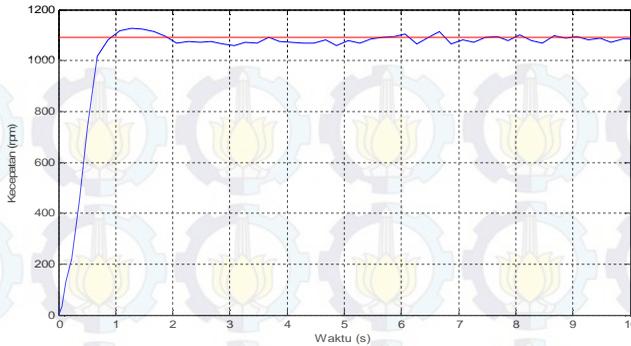
No	Fungsi Alih <i>Plant</i>	RRMSE (%)
1	$G(s) = \frac{0,6}{(0,286s + 1)^2}$	6,66
2	$G(s) = \frac{0,6}{(0,284s + 1)^2}$	7,3
3	$G(s) = \frac{0,6}{(0,2854s + 1)^2}$	6,83
4	$G(s) = \frac{0,6}{(0,289s + 1)^2}$	6,17
5	$G(s) = \frac{0,6}{(0,291s + 1)^2}$	6,39



Gambar 3.30 Hasil Identifikasi dan Permodelan ICE dengan Beban Maksimal

3.6.2 Identifikasi Motor DC

Identifikasi motor DC dilakukan dengan menggunakan metode pendekatan analitik untuk mencari parameter. Setelah dilakukan pengambilan data, didapatkan respon kecepatan motor DC seperti pada Gambar 3.31. Berdasarkan Gambar 3.31 diperoleh suatu kesimpulan bahwa motor DC memiliki respon *overshoot* sehingga dapat didekati dengan model matematis orde dua dengan *overshoot*. Parameter-parameter yang dibutuhkan untuk identifikasi sistem orde dua, antara lain: ω_n , ω_d , dan ξ . Parameter-parameter tersebut didapatkan melalui perhitungan berikut ini:



Gambar 3.31 Respon Kecepatan Motor DC

Menentukan besar overshoot maksimum (persen) dengan rumus :

$$M_p = \frac{c(t_p) - c(\infty)}{c(\infty)} \times 100\%$$

dengan nilai $c(t_p) = 1126$, dan, $c(\infty) = 1090$ didapatkan nilai M_p sebagai berikut

$$M_p = \frac{1126 - 1090}{1090} \times 100\% = 5,5\%$$

Nilai ω_d yang merupakan frekuensi alami teredam dapat dicari dengan persamaan berikut

$$t_p = \frac{\pi}{\omega_d}$$

dengan nilai $t_p = 1,287$ maka didapatkan nilai ω_d sebagai berikut

$$\omega_d = \frac{\pi}{t_p}$$

$$\omega_d = \frac{3,14}{1,287} = 2,44$$

Untuk menemukan nilai koefisien peredaman (ξ), digunakan persamaan:

$$M_p = e^{-(\xi/\sqrt{1-\xi^2})\pi}$$

sehingga didapatkan nilai ξ sebagai berikut

$$0,055 = e^{-(\xi/\sqrt{1-\xi^2})\pi}$$

$$\ln 0,055 = \ln e^{-(\xi/\sqrt{1-\xi^2})\pi}$$

$$-2,9 = -(\xi/\sqrt{1-\xi^2}) \times \pi$$

$$8,4 = \frac{\xi^2}{1-\xi^2} \times \pi^2$$

$$\frac{8,4}{\pi^2} = \frac{\xi^2}{1-\xi^2}$$

$$0,85 - 0,85\xi^2 = \xi^2$$

$$\xi^2 + 0,85\xi^2 = 0,85$$

$$\xi^2 = \frac{0,85}{1,85} = 0,46$$

$$\xi = \sqrt{0,46} = 0,678$$

Setelah mendapatkan nilai ξ , maka dapat ditemukan nilai ω_n dengan persamaan :

$$\omega_d = \omega_n \sqrt{1-\xi^2}$$

sehingga dari persamaan di atas didapatkan nilai ω_n sebagai berikut

$$\omega_n = \frac{\omega_d}{\sqrt{1-\xi^2}}$$

$$\omega_n = \frac{2,44}{\sqrt{1-0,678^2}} = 3,318$$

Respon sistem menunjukkan bahwa sistem tergolong redaman kurang atau *under damped* ($0 < \xi < 1$), maka fungsi alih sistem dapat dituliskan sebagai berikut

$$G(s) = \frac{\omega_n^2}{(s + \xi\omega_n + j\omega_d)(s + \xi\omega_n - j\omega_d)}$$

sehingga didapatkan fungsi alih motor DC sebagai berikut :

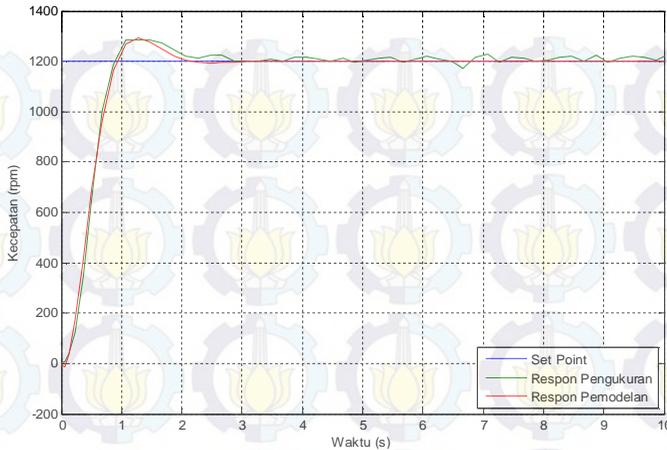
$$G(s) = \frac{3,318^2}{(s + 2,25 + j2,44)(s + 2,25 - j2,44)}$$

$$G(s) = \frac{11,01}{(s^2 + 4,5s + 11,056)}$$

Berdasarkan hasil perhitungan, dilakukan validasi terhadap beberapa fungsi alih. Pengujian respon dapat dilihat pada Tabel 3.10 dan Gambar 3.32.

Tabel 3.10 Pemodelan Dan Validasi *Plant* Motor DC

No	Fungsi Alih <i>Plant</i>	RRMSE(%)
1	$G(s) = \frac{10,98}{(s^2 + 4,8s + 10,975)}$	0,7355
2	$G(s) = \frac{11,09}{(s^2 + 4,52s + 11,087)}$	0,6129
3	$G(s) = \frac{11,2}{(s^2 + 4,49s + 11,17)}$	0,6248
4	$G(s) = \frac{11,01}{(s^2 + 4,5s + 11,056)}$	0,6104
5	$G(s) = \frac{11,35}{(s^2 + 4,54s + 11,29)}$	0,6589



Gambar 3.32 Hasil Identifikasi dan Permodelan Kecepatan Motor DC

3.6.3 Model Referensi

Model referensi merupakan model matematis dengan karakteristik respon *plant* yang diinginkan. Model referensi harus memiliki *rise time* yang lebih cepat daripada *rise time* ICE dan tidak memiliki *error steady state*. Dari hasil identifikasi didapatkan *rise time* motor bakar sebesar 1 s dan nilai *time constant* sebesar 0,3726 s. Model referensi harus memiliki *rise time* lebih cepat dan *time constant* yang lebih kecil sehingga dipilih model referensi dengan fungsi alih ditunjukkan pada Persamaan (3.12).

$$G(s) = \frac{1}{0,2 s + 1} \quad (3.12)$$

3.7 Perancangan Kontroler *Neuro-Fuzzy* Prediktif

Pada Tugas Akhir ini akan digunakan kontroler *Neuro Fuzzy* Prediktif. Kontroler ini akan digunakan untuk mengatur traksi pada simulator HEV. Kontroler *Neuro Fuzzy* Prediktif menggunakan target keluaran prediktif yang diperoleh dari permodelan *invers model* pada *plant*. Salah satu struktur *Neuro Fuzzy* yang dapat digunakan dalam mengontrol suatu *plant* adalah *Fuzzy Modelling Networks (FMN)*. FMN

3.7.1 Perancangan Tahap *Forward Propagation Neuro-Fuzzy*[10]

Pada Gambar 3.33 dapat dilihat bahwa struktur jaringan FMN Tipe II terdiri dari satu lapisan *input*, lima lapisan tersembunyi, dan satu lapisan *output*. Pada subbab ini akan dijelaskan perhitungan *forward propagation* setiap lapisan jaringan.

a. Lapisan A

Lapisan A merupakan lapisan *input* dari struktur jaringan FMN Tipe II. Pada lapisan ini terdapat empat buah *node* yang akan meneruskan sinyal *error* dan sinyal *delta error*. Bobot dari *link* ini adalah 1.

$$O_{1,1} = 1 ; O_{2,1} = X_1 ; O_{3,1} = 1 ; O_{3,1} = X_2 \quad (3.13)$$

b. Lapisan B

Lapisan ini terdiri dari delapan *node* yang merupakan hasil penjumlahan dari nilai *input* yang dikali dengan bobot w_c . Empat *node* awal merupakan kelompok untuk *input* X_1 , sedangkan empat *node* lainnya merupakan kelompok untuk *input* X_2 .

$$\begin{aligned} O_{i,2} &= 1(w_c) + X_1 \\ O_{j,2} &= 1(w_c) + X_2 \end{aligned} \quad (3.14)$$

dimana $i = 1,2,3,4$ dan $j = 5,6,7,8$

c. Lapisan C

Lapisan ini merupakan fungsi aktivasi dari *output* lapisan B dikali dengan bobot w_g . Lapisan ini memiliki delapan buah *node* yang terbagi menjadi dua kelompok. Empat *node* awal merupakan kelompok untuk *input* X_1 , sedangkan empat *node* lainnya merupakan kelompok untuk *input* X_2 . Fungsi aktivasi yang digunakan pada lapisan ini adalah fungsi aktivasi *sigmoid*.

$$\begin{aligned} O_{i,3} &= \frac{1}{1 + e^{-w_g(o_{i,2})}} \\ O_{j,3} &= \frac{1}{1 + e^{-w_g(o_{j,2})}} \end{aligned} \quad (3.15)$$

dimana $i = 1,2,3,4$ dan $j = 5,6,7,8$

d. Lapisan D

Lapisan ini akan menghasilkan nilai fungsi keanggotaan dari nilai *input* X_1 dan X_2 . Lapisan ini terdiri dari enam *node*, dimana tiga *node* pertama menunjukkan fungsi keanggotaan dari *input* X_1 dan tiga *node* lainnya menunjukkan fungsi keanggotaan dari *input* X_2 . Fungsi keanggotaan dari *input* dibagi menjadi tiga, yaitu: negatif, zero, dan positif.

$$O_{1,4} = O_{1,3}$$

$$O_{2,4} = O_{2,3} + (-1)O_{3,3}$$

$$O_{3,4} = O_{4,3}$$

(3.16)

e. Lapisan E

Lapisan ini terdiri dari sembilan *node* yang merupakan kombinasi aturan-aturan variabel linguistik *input*. Kontroler NF yang dirancang memiliki dua buah *input* (*error* dan *delta error*) sehingga masing-masing *node* pada lapisan ini akan memiliki dua buah *input* yang berasal dari lapisan sebelumnya. Setiap *node* pada lapisan ini akan melakukan operasi AND terhadap kedua *input* tersebut. Operasi yang akan digunakan pada kontroler NF ini adalah *algebraic product*. Persamaan untuk fungsi operasi tersebut dinyatakan dalam Persamaan (3.17).

$$O_{i,5} = O_{m,4} \times O_{n,4}$$

(3.17)

dimana $i = 1,2,3,\dots,9$; $m = 1,2,3$; $n = 1,2,3$

Untuk memperjelas konfigurasi pada lapisan E, Gambar 3.33 menampilkan konfigurasi setiap *node* pada lapisan E. Kombinasi aturan tersebut disajikan dalam bentuk linguistik sebagai berikut:

IF x_1 **is** $O_{m,4}$ **AND** x_2 **is** $O_{n,4}$ **THEN** y **is** *constant*

f. Lapisan F

Lapisan ini merupakan bagian konsekuen dari sistem inferensi *fuzzy*. Berdasarkan aturan linguistik yang sudah dibuat, kemudian *output* dari lapisan sebelumnya dijumlahkan berdasarkan konsekuen yang sesuai. Konsekuen pada kontroler NF ini dibagi menjadi tiga, yaitu: *negative*, *zero*, dan *positive*.

$$O_{i,6} = \sum_{m=1}^9 O_{m,5} \quad (3.18)$$

dimana $i = 1,2,3$

Aturan yang dibuat pada kontroler NF ini direpresentasikan dalam Tabel 3.11.

Tabel 3.11 Basis Aturan Kontroler *Neuro-Fuzzy*

Gaya (u)		error (e)		
		N	Z	P
delta error (Δe)	N	N	P	Z
	Z	N	Z	P
	P	Z	N	P

g. Lapisan G

Lapisan ini merupakan lapisan *output* dari kontroler NF. Pada lapisan ini dilakukan proses defuzzifikasi dengan cara menjumlahkan hasil perkalian dari bobot dengan lapisan sebelumnya. *Output* dari lapisan ini berupa sinyal kontrol tegas. Perhitungan pada lapisan ini dapat dituliskan sebagai Persamaan (3.19).

$$O_{i,7} = \sum_{i=1}^3 O_{i,6} \times w_f \quad (3.19)$$

3.7.2 Perancangan Tahap *Back Propagation Neuro-Fuzzy*[10]

Tahap *back propagation* merupakan tahapan untuk memperbaiki bobot-bobot pada NF agar nilai *output* sesuai dengan target yang diinginkan. Penghitungan *back propagation* dilakukan pada setiap lapisan kontroler NF. Pada sub bab ini akan dibahas penghitungan *back propagation* pada setiap lapisan kontroler NF.

a. Lapisan G

Pada *link* lapisan ini terdapat bobot w_f yang diperbaiki untuk memperoleh nilai *output* sesuai target yang diinginkan. Perambatan *error* juga dihitung dalam lapisan ini. Nilai *error* pada lapisan ini proporsional terhadap bobot yang ada. Perhitungan bobot w_f baru dapat ditulis dalam Persamaan (3.20)

$$w_f(t + 1) = w_f(t) + \alpha (T_i(t) - O_{i,7}(t)) O_{i,6}(t)$$
$$\delta_{i,7}(t) = \frac{O_{i,6}}{O_{i,7}} (T_i(t) - O_{i,7}(t)) \quad (3.20)$$

b. Lapisan F

Pada *link* lapisan ini tidak ada perubahan bobot, akan tetapi tetap dihitung perambatan *error* dari lapisan F menuju lapisan E. Berdasarkan struktur kontroler NF yang digunakan, didapatkan bahwa nilai *error* dari lapisan G menuju lapisan F merupakan penjumlahan dari *error* lapisan F menuju *error* lapisan E. Bobot pada lapisan E menuju lapisan F bernilai satu sehingga perhitungan perambatan *error* dapat ditulis dalam Persamaan (3.21)

$$\delta_{i,6}(t) = \frac{1}{3} \delta_{i,7}(t) \quad (3.21)$$

c. Lapisan E

Pada *link* lapisan ini tidak ada perubahan bobot, akan tetapi tetap dihitung perambatan *error* dari lapisan E menuju lapisan D. Berdasarkan struktur kontroler NF yang digunakan, didapatkan bahwa *output* lapisan E merupakan perkalian dari *output* lapisan D. Perbandingan *error* sama dengan perbandingan *output* pada lapisan D dan lapisan E, sehingga perambatan *error* dapat ditulis dalam Persamaan (3.22)

$$\delta_{i,5}(t) = \sqrt{\delta_{i,6}(t)} \quad (3.22)$$

d. Lapisan D

Pada *link* lapisan ini tidak ada perubahan bobot, akan tetapi tetap dihitung perambatan *error* dari lapisan E menuju lapisan D.

Berdasarkan struktur kontroler NF yang digunakan, didapatkan bahwa *output* lapisan C diteruskan ke lapisan D tanpa ada operasi matematika. Akan tetapi pada *node* kedua dan keempat lapisan C terdapat operasi penjumlahan. Sehingga perambatan *error* dapat ditulis dalam Persamaan (3.23)

$$\begin{aligned}\delta_{i,4}(t) &= \delta_{i,5}(t) \\ \delta_{i,4}(t) &= \frac{1}{2}\delta_{i,5}(t) ; \text{ untuk } i = 2,3,6,7\end{aligned}\quad (3.23)$$

e. Lapisan C

Pada *link* lapisan ini terdapat bobot w_g yang diperbaiki untuk memperoleh nilai *output* sesuai target yang diinginkan. Perambatan *error* juga dihitung dalam lapisan ini. Pada lapisan ini digunakan fungsi aktivasi *sigmoid*. Sehingga perhitungan bobot w_g baru dapat ditulis dalam Persamaan (3.24).

$$\begin{aligned}f(x) &= \left(\frac{1}{1 + (e^{-w_g(x_i.w_c)})} \right) \\ f'(x) &= \left(\frac{1}{1 + (e^{-w_g(x_i.w_c)})} \right) \left(1 - \frac{1}{1 + (e^{-w_g(x_i.w_c)})} \right) \\ w_g(t+1) &= w_g(t) + \alpha(\delta_{i,4}(t)f'(x)x_i.w_c) \\ \delta_{i,3}(t) &= \alpha(\delta_{i,4}(t))\end{aligned}\quad (3.24)$$

f. Lapisan B

Pada *link* lapisan ini terdapat bobot w_c yang diperbaiki untuk memperoleh nilai *output* sesuai target yang diinginkan. Perambatan *error* juga dihitung dalam lapisan ini. Pada lapisan ini digunakan fungsi aktivasi *sigmoid*. Sehingga perhitungan bobot w_c baru dapat ditulis dalam Persamaan (3.25).

$$\begin{aligned}w_c(t+1) &= w_c(t) + \alpha(\delta_{i,3}(t)f'(x)) \\ \delta_{i,2}(t) &= \alpha(\delta_{i,3}(t))\end{aligned}\quad (3.25)$$

g. Lapisan A

Tidak ada parameter yang perlu dilakukan adaptasi pada lapisan ini karena pada lapisan ini hanya berfungsi sebagai penerus dari *input*.

3.7.3 Model Reference Based Invers Model

Permodelan *invers model* digunakan untuk mengetahui nilai suatu *input* jika diketahui nilai *output* dan nilai *input* sebelumnya. Permodelan *invers model* diperoleh dari identifikasi parameter *plant*. Berdasarkan Persamaan (3.26) diperoleh fungsi alih motor DC sebagai fungsi:

$$G(s) = \frac{11,01}{s^2 + 4,5s + 11,056} \quad (3.26)$$

Berdasarkan fungsi alih kontinu pada Persamaan (3.26) kemudian diubah menjadi fungsi alih diskrit dengan *time sampling* sebesar 0.02 s. Fungsi alih diskrit *plant* motor DC dapat dilihat pada Persamaan (3.27)

$$G(z) = \frac{0,002137z + 0,002073}{z^2 - 1,91z + 0,9139} \quad (3.27)$$

Berdasarkan fungsi alih diskrit pada Persamaan (3.27) dapat diperoleh parameter-parameter dari *plant*.

$$y(k) = a_1y(k-1) - a_2y(k-2) + b_0u(k-1) + b_1u(k-2) \quad (3.28)$$

Parameter-parameter a_1 , a_2 , b_0 , dan b_1 diperoleh dari koefisien-koefisien pada Persamaan (3.28) sehingga dapat diperoleh Persamaan (3.29).

$$y(k) = 1,91y(k-1) - 0,9139y(k-2) + 0,002137u(k-1) + 0,002073u(k-2) \quad (3.29)$$

Berdasarkan Persamaan (3.29) dapat diperoleh persamaan prediksi sebagai Persamaan (3.30).

$$y(k+1) = 1,91y(k) - 0,9139y(k-1) + 0,002137u(k) + 0,002073u(k-1) \quad (3.30)$$

Invers Model dari *plant* motor DC direpresentasikan dalam Persamaan (3.31).

$$u(k) = 467,946y(k + 1) + 893,776y(k) - 427,655y(k - 1) + 0,97u(k - 1) \quad (3.31)$$

Nilai $y(k + 1)$ merupakan nilai prediksi yang diinginkan, dimana diasumsikan bahwa $y(k + 1) = y_m(k + 1)$. Nilai $y_m(k + 1)$ diperoleh dari model referensi yang diinginkan. Dimisalkan model yang diinginkan memiliki nilai *time constant* $\tau = 0,01$ s. Maka dengan pendekatan sistem orde satu diperoleh Persamaan (3.32).

$$G(s) = \frac{1}{0,01s + 1} \quad (3.32)$$

Diharapkan nilai $y(k) = y_m(k)$ maka didapatkan Persamaan (3.33).

$$\begin{aligned} \frac{y_m(s)}{y_r(s)} &= \frac{1}{0,01s + 1} \\ \frac{y(s)}{y_r(s)} &= \frac{1}{0,01s + 1} \\ 0,01\dot{y} + y &= y_r \end{aligned} \quad (3.33)$$

Bentuk diskrit dari Persamaan (3.33) direpresentasikan dalam Persamaan (3.34).

$$\begin{aligned} \tau \left(\frac{y(k) - y(k - 1)}{T_s} \right) + y(k) &= y_r(k) \\ \left(\frac{\tau}{T_s} + 1 \right) y(k) - \frac{\tau}{T_s} y(k - 1) &= y_r(k) \\ y(k) &= \frac{\tau}{\tau + T_s} y(k - 1) + \frac{T_s}{\tau + T_s} y_r(k) \end{aligned} \quad (3.34)$$

Nilai $y_m(k + 1)$ dapat diperoleh dari Persamaan (3.35)

$$y_m(k + 1) = \frac{\tau}{\tau + T_s} y(k) + \frac{T_s}{\tau + T_s} y_r(k + 1) \quad (3.35)$$

Diharapkan nilai $y(k+1)=y_m(k+1)$, maka dengan melakukan substitusi Persamaan (3.30) dan (3.35) didapatkan Persamaan (3.36).

$$\begin{aligned} y(k+1) &= y_m(k+1) \\ a_1 y(k) - a_2 y(k-1) + b_0 u(k) + b_1 u(k-1) \\ &= \frac{\tau}{\tau + T_s} y(k) + \frac{T_s}{\tau + T_s} y_r(k+1) \end{aligned} \quad (3.36)$$

Nilai *time constant* $\tau = 0,01$ dan nilai *time sampling* $T_s = 0,02$ s. Berdasarkan Persamaan (3.36) dapat diperoleh Persamaan (3.37) sebagai *invers model* dari *plant* motor DC.

$$\begin{aligned} u(k) &= \frac{1}{b_0} \left\{ \left(\frac{\tau}{\tau + T_s} - a_1 \right) y(k) + a_2 y(k-1) \right. \\ &\quad \left. - b_1 u(k-1) + \frac{T_s}{\tau + T_s} y_r(k+1) \right\} \\ u(k) &= -737,793 y(k) + 427,655 y(k-1) \\ &\quad - 0,97 u(k-1) + 311,964 (y_r(k+1)) \end{aligned} \quad (3.37)$$

3.7.4 Perancangan *Observer* [14]

Pada Tugas Akhir kali ini, simulator HEV yang digunakan hanya memiliki satu *output* berupa *tachogenerator* yang merepresentasikan *output* HEV. Sedangkan pada perancangan kontroler motor DC dibutuhkan *feedback* yang berasal dari motor DC. Akan tetapi pada tahap implementasi *feedback* motor DC tidak bisa didapatkan, oleh karena itu dibuat suatu matriks *observer* yang dapat merepresentasikan *output* motor DC. Matriks *observer* bertujuan untuk mengetahui suatu nilai yang tidak bisa diketahui dari suatu *plant*. Matriks *observer* hanya bisa diterapkan terhadap suatu *plant* yang *observable*.

Perancangan matriks *observer* dimulai dengan membuat suatu pendekatan *state-space* sistem. Berdasarkan model matematika hasil identifikasi ICE dimisalkan

$$\begin{aligned} x_1 &= y_1 \\ \dot{x}_1 &= \dot{y}_1 \\ x_2 &= \dot{y}_1 \end{aligned}$$

$$\dot{x}_2 = \ddot{y}_1 \quad (3.38)$$

Sehingga diperoleh,

$$\begin{aligned} s^2 Y_1(s) + 10,87s Y_1(s) + 29,412 Y_1(s) &= 29,412 U_1(s) \\ \ddot{y}_1 + 10,87 \dot{y}_1 + 29,412 y_1 &= 29,412 u_1 \end{aligned} \quad (3.39)$$

Dengan melakukan substitusi Persamaan (3.38) dan (3.39) didapatkan

$$\begin{aligned} \dot{x}_2 + 10,87 x_2 + 29,412 &= 29,412 u_1 \\ \dot{x}_2 &= -10,87 x_2 - 29,412 + 29,412 u_1 \end{aligned} \quad (3.40)$$

Maka didapatkan representasi *state space* dari ICE dalam Persamaan (3.41)

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ -29,412 & -10,87 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 29,412 \end{bmatrix} u_1 \\ y_1 &= [1 \ 0] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \end{aligned} \quad (3.41)$$

Berdasarkan model matematika hasil identifikasi motor DC dimisalkan

$$\begin{aligned} x_3 &= y_2 \\ \dot{x}_3 &= \dot{y}_2 \\ x_4 &= \ddot{y}_2 \\ \dot{x}_4 &= \ddot{y}_2 \end{aligned} \quad (3.42)$$

Sehingga diperoleh,

$$\begin{aligned} s^2 Y_2(s) + 4,5s Y_2(s) + 11,056 Y_2(s) &= 11,01 U_2(s) \\ \ddot{y}_2 + 4,5 \dot{y}_2 + 11,057 y_2 &= 11,01 u_2 \end{aligned} \quad (3.43)$$

Dengan melakukan substitusi Persamaan (3.42) dan (3.43) didapatkan

$$\begin{aligned} \dot{x}_4 + 4,5 \dot{x}_3 + 11,056 x_3 &= 11,01 u_2 \\ \dot{x}_4 &= -4,5 \dot{x}_3 - 11,056 x_3 + 11,01 u_2 \end{aligned} \quad (3.44)$$

Maka didapatkan representasi *state-space* dari motor DC dalam Persamaan (3.45)

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -11,056 & -4,5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 11,01 \end{bmatrix} u_1$$

$$y_1 = [1 \ 0] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (3.45)$$

Dari representasi *state-space* ICE dan motor DC, dapat dibuat suatu persamaan *state-space* yang merepresentasikan HEV. Persamaan *state-space* dari HEV ditunjukkan pada Persamaan (3.46).

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -29,412 & -10,87 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -11,056 & -4,5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 29,412 & 0 \\ 0 & 0 \\ 0 & 11,01 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$y = [1 \ 0 \ 1 \ 0] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \quad (3.46)$$

di mana:

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ -29,412 & -10,87 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -11,056 & -4,5 \end{bmatrix} = \mathbf{A}$$

$$\begin{bmatrix} 0 & 0 \\ 29,412 & 0 \\ 0 & 0 \\ 0 & 11,01 \end{bmatrix} = \mathbf{B}$$

$$[1 \ 0 \ 1 \ 0] = \mathbf{C}$$

Setelah didapatkan matriks **A**, **B**, dan **C** dilakukan pengujian *observability*. Pengujian dilakukan dengan melakukan perhitungan melalui Persamaan (3.47).

$$[C^T \quad A^T C^T \quad (A^T)^2 C^T \quad (A^T)^3 C^T] \quad (3.47)$$

Dari perhitungan menggunakan Persamaan (3.47) didapatkan hasil $rank = 4$, sehingga dapat disimpulkan bahwa sistem tersebut *observable*.

Tahap selanjutnya adalah perancangan *full-order state observer*. Pada tahapan ini dibutuhkan perhitungan nilai K_e sebagai matriks *gain observer*. Perhitungan K_e dapat dilakukan dengan beberapa metode. Pada Tugas Akhir ini dilakukan perhitungan menggunakan metode Ackermann. Sebelum melakukan perhitungan nilai K_e , dilakukan perhitungan *eigenvalue* dari sistem melalui Persamaan (3.48).

$$|sI - A| = 0 \quad (3.48)$$

$$(s + 5,0783)(s + 5,7917)(s + 2,05 - j2,63711)(s + 2,05 + j2,63711) = 0$$

Berdasarkan nilai *pole* dari sistem, kemudian dicari nilai *pole* baru agar didapatkan suatu respon yang sama dengan sistem. Berdasarkan hasil *trial and error* didapatkan nilai *pole* yang baru dalam Persamaan (3.49).

$$\begin{aligned} \phi(s) &= (s + 5 - j2)(s + 5 + j2)(s + 2 - j2)(s + 2 + j2) \quad (3.49) \\ &= (s + \mu_1)(s + \mu_2)(s + \mu_3)(s + \mu_4) \\ &= s^4 + \alpha_1 s^3 + \alpha_2 s^2 + \alpha_3 s + \alpha_4 \\ \phi(A) &= A^4 + \alpha_1 A^3 + \alpha_2 A^2 + \alpha_3 A + \alpha_4 \end{aligned}$$

Nilai matriks *gain observer* K_e didapatkan dari Persamaan (3.50)

$$K_e = \phi(A) \begin{bmatrix} C \\ CA \\ CA^2 \\ CA^3 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (3.50)$$

3.7.5 Perancangan Prediktor

Prediktor merupakan suatu model matematika yang memiliki respon lebih cepat daripada sistem yang sebenarnya. Pada Tugas Akhir

ini, digunakan metode *Neuro-Fuzzy* Prediktif sehingga dibutuhkan suatu prediktor untuk memberikan informasi mengenai *error* yang akan datang.

Pada Tugas Akhir ini, motor DC ditempatkan sebagai tenaga pembantu ketika terjadi proses akselerasi sehingga motor DC akan bekerja ketika ICE tidak bisa mencapai referensi yang diinginkan. Prediktor yang digunakan adalah prediktor model referensi dan ICE, ketika kedua nilai tersebut dikurangkan maka akan terlihat *error* prediksi. Error prediksi ini digunakan sebagai referensi bagi kontroler *Neuro-Fuzzy*. Pada Tugas Akhir ini digunakan prediksi nilai satu langkah ke depan.

Model referensi yang diinginkan memiliki fungsi alih seperti pada Persamaan (3.51)

$$G(s) = \frac{1}{0,2s + 1} \quad (3.51)$$

Berdasarkan Persamaan (3.51) fungsi alih kontinu diubah menjadi fungsi alih diskrit dengan *time sampling* sebesar 0,05 s.

$$G(z) = \frac{0,39346}{z - 0,60654}$$

Sehingga didapatkan

$$\begin{aligned} y_r(k) &= b_r y_r(k-1) + a_r x(k-1) \\ y_r(k) &= 0,60654 y_r(k-1) + 0,39346 x(k-1) \\ y_r(k+1) &= 0,60654 y_r(k) + 0,39346 x(k) \end{aligned} \quad (3.52)$$

ICE dengan beban nominal yang digunakan pada simulator HEV memiliki fungsi alih seperti pada Persamaan (3.53)

$$G(s) = \frac{1}{0,034s^2 + 0,3696s + 1} \quad (3.53)$$

Berdasarkan Persamaan (3.53) fungsi alih kontinu diubah menjadi fungsi alih diskrit dengan *time sampling* sebesar 0,05 s.

$$G(z) = \frac{0,10328z + 0,071766}{z^2 - 1,1622z + 0,33277}$$

Sehingga didapatkan persamaan

$$\begin{aligned}y_b(k) &= b_{b1}y(k-1) + b_{b2}y(k-2) \\ &\quad + a_{b1}x(k-1) + a_{b2}x(k-2) \\ &= 1,1622y(k-1) - 0,33722y(k-2) \\ &\quad + 0,10328x(k-1) + 0,071766x(k-2) \\ y_b(k+1) &= 1,1622y(k) - 0,33722y(k-1) \\ &\quad + 0,10328x(k) + 0,071766x(k-1)\end{aligned}\tag{3.54}$$



BAB 4

PENGUJIAN DAN ANALISIS SISTEM

4.1 Gambaran Umum Pengujian Sistem

Pada Tugas Akhir kali ini dilakukan pengujian sistem melalui simulasi dan implementasi. Pengujian sistem dilakukan untuk mengetahui performansi kontroler yang sudah dirancang dalam menyelesaikan masalah pada sistem. Dalam pengujian berupa simulasi, terdapat beberapa aspek pengujian yang diberikan terhadap sistem, antara lain:

- a. Simulasi pengujian ICE berbeban nominal tanpa bantuan motor DC terhadap *set point* berupa sinyal *step*.
- b. Simulasi pengujian ICE berbeban nominal tanpa bantuan motor DC terhadap *set point* dengan perubahan amplitudo.
- c. Simulasi pengujian ICE berbeban nominal dengan bantuan motor DC yang telah dikontrol terhadap *set point* berupa sinyal *step*.
- d. Simulasi pengujian ICE berbeban nominal dengan bantuan motor DC yang telah dikontrol terhadap *set point* dengan perubahan amplitudo.

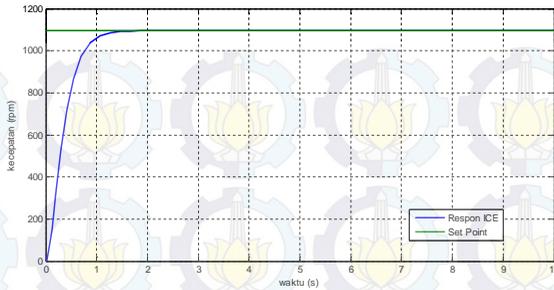
4.2 Simulasi Sistem

Simulasi sistem dilakukan melalui perangkat lunak Matlab. Simulasi sistem bertujuan untuk mengetahui performansi dari kontroler yang telah dirancang.

4.2.1 Pengujian ICE Berbeban Nominal Tanpa Bantuan Motor DC

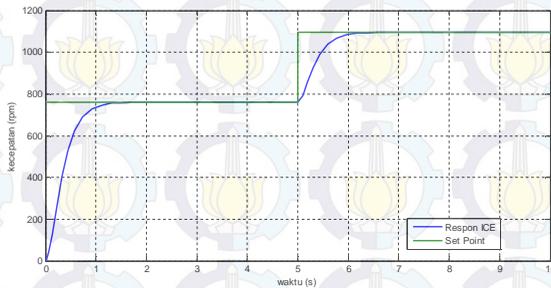
Simulasi sistem dilakukan terhadap ICE tanpa bantuan motor DC. Sinyal *input* yang diberikan berupa sinyal *step* dan sinyal dengan perubahan amplitudo.

Simulasi pertama dilakukan pengujian terhadap *set point* berupa sinyal *step*. *Set point* yang diberikan terhadap sistem sebesar 1095 rpm. Pada Gambar 4.1 dapat dilihat bahwa ICE dapat mencapai *set point* yang diberikan dengan *rise time* $t_r = 0,8155$ s.



Gambar 4.1 Respon ICE Berbeban Tanpa Bantuan Motor DC terhadap Sinyal *Step*

Simulasi kedua dilakukan pengujian terhadap *set point* dengan perubahan amplitudo. Amplitudo pertama memiliki nilai sebesar 760,6 rpm, sedangkan amplitudo kedua memiliki nilai sebesar 1095 rpm. Pada Gambar 4.2 dapat dilihat bahwa respon ICE dapat mencapai kedua amplitudo yang diberikan. Pada saat amplitudo *set point* sebesar 760,6 rpm diperoleh *rise time* $t_r = 0,8147 s$, sedangkan pada amplitudo *set point* sebesar 1095 rpm diperoleh *rise time* $t_r = 0,8495 s$.



Gambar 4.2 Respon ICE Berbeban Tanpa Bantuan Motor DC terhadap *Set Point* dengan Perubahan Amplitudo

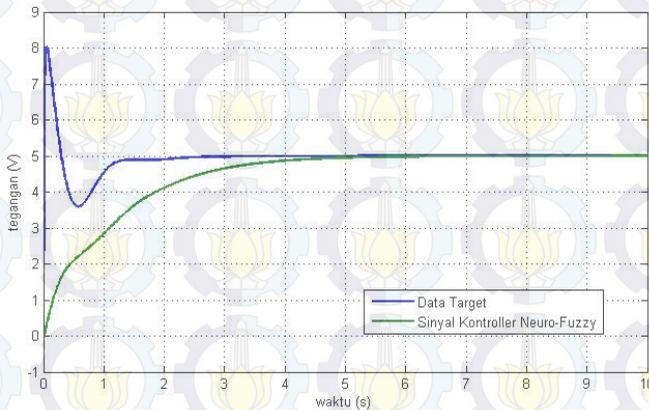
4.2.2 Pengujian Kontroler *Neuro-Fuzzy* Terhadap Sinyal Target

Kontroler *Neuro-Fuzzy* memiliki kemampuan untuk melakukan *learning*, sehingga akan diperoleh data *output* yang sesuai dengan data target. Kemampuan *learning* dari kontroler *Neuro-Fuzzy* dipengaruhi oleh *learning rate* (α) yang dimiliki. Pada Tugas Akhir ini dibutuhkan waktu *learning* kontroler *Neuro-Fuzzy* yang cepat karena *plant* HEV

memiliki respon yang cepat. Pada subbab ini akan dilakukan pengujian terhadap beberapa nilai *learning rate*.

a. *Learning rate* ($\alpha=0,0001$)

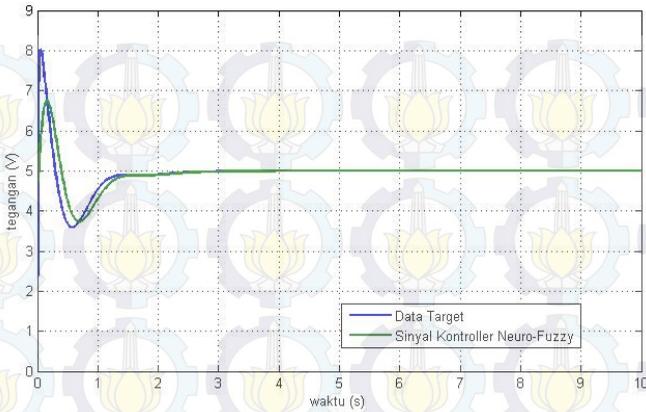
Pada pengujian ini akan diberikan nilai $\alpha=0,0001$. Data target yang akan menjadi sinyal masukan adalah sinyal random. Berdasarkan Gambar 4.3 dapat dilihat bahwa kontroler *Neuro-Fuzzy* memiliki kemampuan untuk *learning* terhadap data target. Sinyal Kontroler *Neuro-Fuzzy* mampu beradaptasi dengan data target pada $t = \pm 4,2$ s.



Gambar 4.3 Output Kontroler *Neuro-Fuzzy* dengan $\alpha=0,0001$

b. *Learning rate* ($\alpha=0,001$)

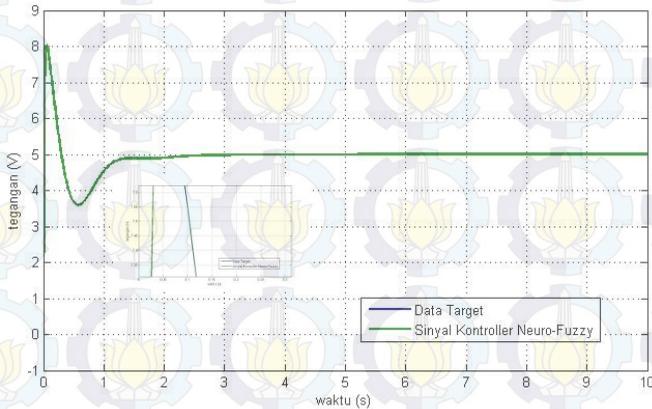
Pada pengujian ini akan diberikan nilai $\alpha=0,001$. Data target yang akan menjadi sinyal masukan adalah sinyal random. Berdasarkan Gambar 4.4 dapat dilihat bahwa kontroler *Neuro-Fuzzy* memiliki kemampuan untuk *learning* terhadap data target. Sinyal Kontroler *Neuro-Fuzzy* mampu beradaptasi dengan data target pada $t = \pm 1,18$ s.



Gambar 4.4 Output Kontroler Neuro-Fuzzy dengan $\alpha=0,001$

c. *Learning rate ($\alpha=0,01$)*

Pada pengujian ini akan diberikan nilai $\alpha=0,01$. Data target yang akan menjadi sinyal masukan adalah sinyal random. Berdasarkan Gambar 4.5 dapat dilihat bahwa kontroler Neuro-Fuzzy memiliki kemampuan untuk *learning* terhadap data target. Sinyal Kontroler Neuro-Fuzzy mampu beradaptasi dengan data target pada $t = \pm 0,025$ s.



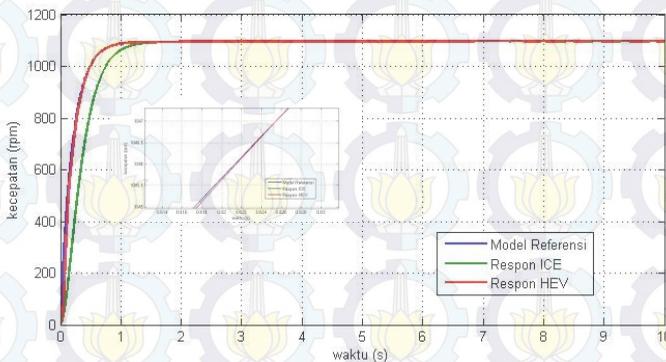
Gambar 4.5 Output Kontroler Neuro-Fuzzy dengan $\alpha=0,01$

Berdasarkan pengujian antara data target dengan data *output* kontroler *Neuro-Fuzzy* diperoleh nilai *learning rate* ($\alpha=0,01$) memiliki waktu adaptasi paling cepat. Sehingga nilai *learning rate* yang terbaik untuk kontroler *Neuro-Fuzzy* adalah $\alpha=0,01$.

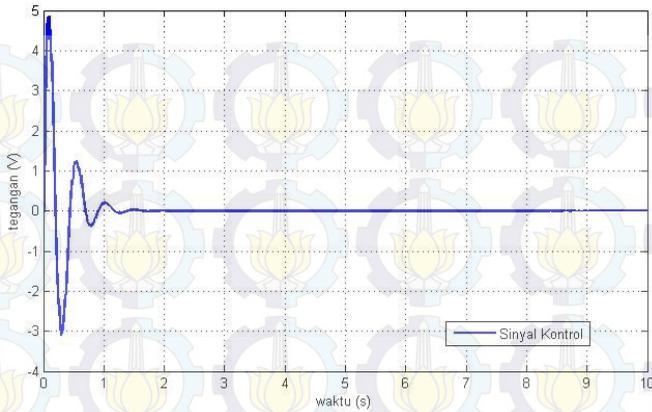
4.2.3 Pengujian ICE Berbeban Nominal Dengan Bantuan Motor DC Menggunakan Kontroler *Neuro-Fuzzy* Prediktif

4.2.3.1 Pengujian Menggunakan Kontroler *Neuro-Fuzzy* Tanpa Model Prediksi

Pengujian ini dilakukan untuk mengetahui pengaruh model prediksi terhadap aksi kontrol yang diberikan. Pengujian dilakukan dengan memberikan *set point* berupa sinyal *step* dan *set point* dengan perubahan amplitudo. Pada simulasi pertama diberikan sinyal *step* sebesar 1095 rpm. Nilai referensi untuk kontroler *Neuro-Fuzzy* didapatkan dari *error* antara model referensi dan mesin bakar. Pada Gambar 4.6 dapat dilihat bahwa respon HEV tidak bisa mencapai model referensi. Pada Gambar 4.6 terlihat respon HEV mencapai model referensi pada $t=\pm 0,622$ s. Sinyal *output* kontroler *Neuro-Fuzzy* ditunjukkan pada Gambar 4.7.

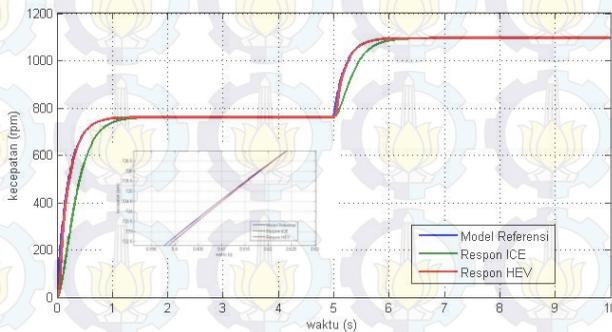


Gambar 4.6 Respon ICE Berbeban dengan Bantuan Motor DC Tanpa Model Prediksi terhadap Sinyal *Step*

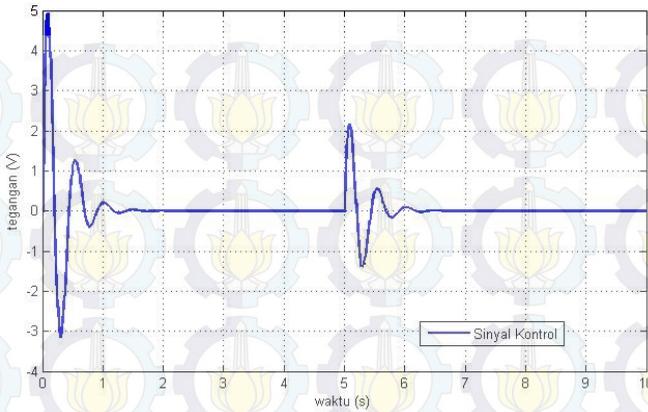


Gambar 4.7 Sinyal Kontrol Motor DC Tanpa Model Prediksi terhadap Sinyal Step

Pengujian kedua dilakukan dengan memberikan *set point* dengan perubahan amplitudo sebesar 760,6 rpm dan 1095 rpm. Pada Gambar 4.8 dapat dilihat bahwa ketika diberi *set point* sebesar 760,6 rpm respon HEV mampu mengikuti model referensi pada $t = \pm 0,6175$ s pada saat proses akselerasi, sedangkan ketika diberi *set point* sebesar 1095 rpm respon HEV mampu mengikuti model referensi pada $t = \pm 0,6$ s. Sinyal *output* kontroler *Neuro-Fuzzy* dapat dilihat pada Gambar 4.9.



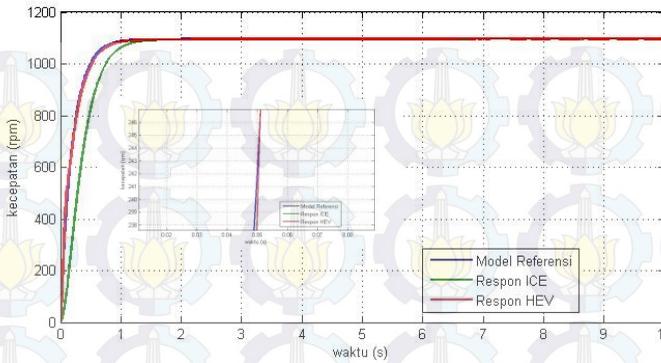
Gambar 4.8 Respon ICE Berbeban dengan Bantuan Motor DC Tanpa Model Prediksi terhadap *Set Point* dengan Perubahan Amplitudo



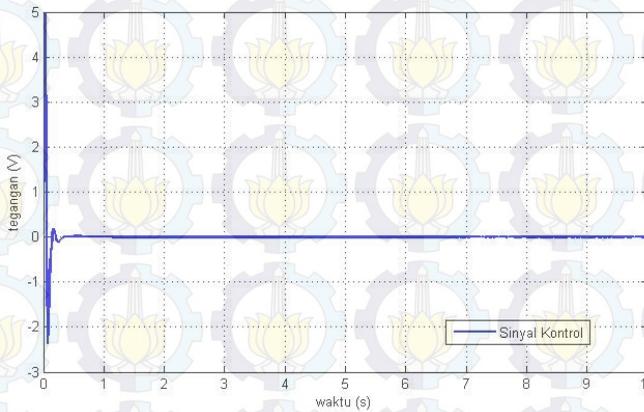
Gambar 4.9 Sinyal Kontrol Motor DC Tanpa Model Prediksi terhadap Sinyal *Input* dengan Perubahan Amplitudo

4.2.3.2 Pengujian Menggunakan Kontroler *Neuro-Fuzzy* Dengan Model Prediksi

Pengujian ini dilakukan untuk mengetahui pengaruh model prediksi terhadap aksi kontrol yang diberikan. Pengujian dilakukan dengan memberikan *set point* berupa sinyal *step* dan *set point* dengan perubahan amplitudo. Pada simulasi pertama diberikan sinyal *step* sebesar 1095 rpm. Nilai referensi untuk kontroler *Neuro-Fuzzy* didapatkan dari prediksi *error* antara model referensi dan mesin bakar. Pada Gambar 4.10 dapat dilihat bahwa HEV mulai mencapai model referensi pada $t = 0,051$ s. Sinyal *output* kontroler *Neuro-Fuzzy* dapat dilihat pada Gambar 4.11.

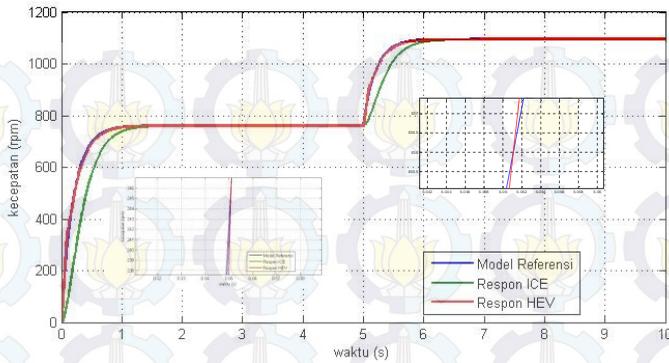


Gambar 4.10 Respon ICE Berbeban dengan Bantuan Motor DC Menggunakan Model Prediksi terhadap Sinyal Step

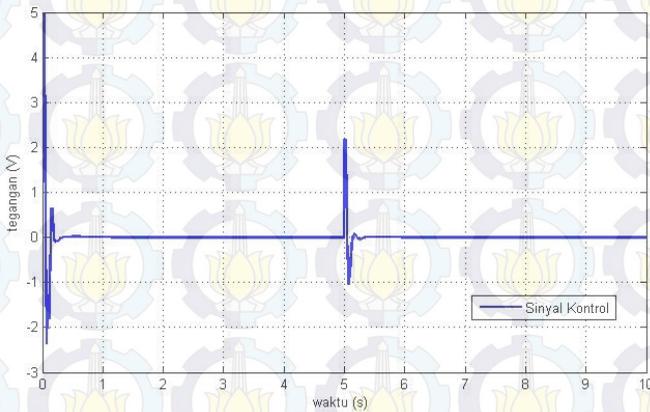


Gambar 4.11 Sinyal Kontrol Motor DC Menggunakan Model Prediksi terhadap Sinyal Step

Pengujian kedua dilakukan dengan memberikan *set point* dengan perubahan amplitudo sebesar 760,6 rpm dan 1095 rpm. Pada Gambar 4.12 dapat dilihat bahwa ketika diberi *set point* sebesar 760,6 rpm, respon HEV mulai mencapai model referensi pada $t = 0,051$ s. Sedangkan pada saat diberi *set point* sebesar 1095 rpm, respon HEV mencapai model referensi pada $t = 0,051$ s. Sinyal *output* kontroler *Neuro-Fuzzy* dapat dilihat pada Gambar 4.13.



Gambar 4.12 Respon ICE Berbeban dengan Bantuan Motor DC Menggunakan Model Prediksi terhadap *Set Point* dengan Perubahan Amplitudo



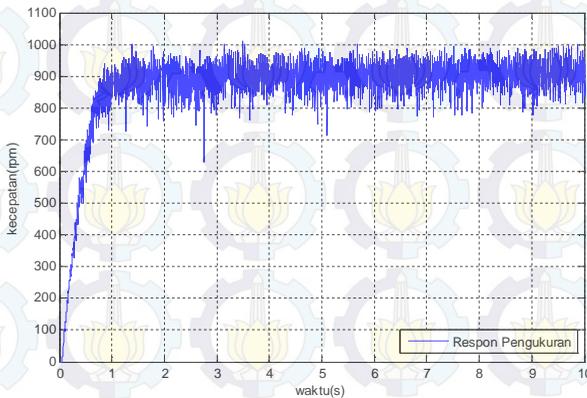
Gambar 4.13 Sinyal Kontrol Motor DC Menggunakan Model Prediksi terhadap Sinyal Masukan dengan Perubahan Amplitudo

4.3 Implementasi Sistem

Implementasi sistem dilakukan untuk menguji performa kontroler ketika diaplikasikan pada *plant* sesungguhnya. Sinyal *input* berupa tegangan dari potensiometer yang merepresentasikan bukaan *throttle*, sedangkan *feedback* dari sistem HEV diperoleh dari tegangan *tachogenerator* yang merepresentasikan respon *output* sistem HEV. *Output* sinyal kontrol berasal dari *output* PWM sistem minimum Arduino dengan *range* tegangan 0 - 5 V. Tegangan *output* dari sistem minimum Arduino kemudian dikuatkan menggunakan *driver* agar sesuai dengan tegangan kerja motor DC.

4.3.1 Pengujian ICE Berbeban Nominal Tanpa Bantuan Motor DC

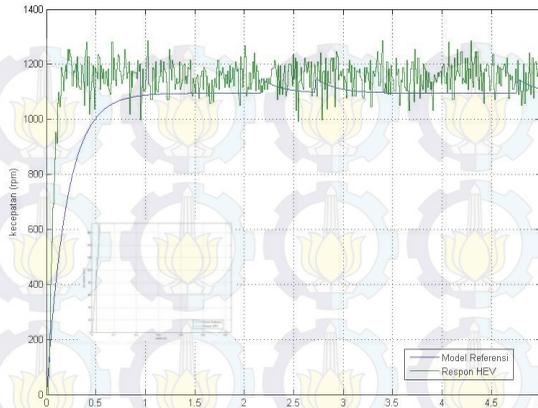
Berdasarkan hasil identifikasi kecepatan ICE berbeban nominal, diperoleh nilai *rise time* $\pm 0,85$ s dengan *time constant* sebesar $\pm 0,3726$ s. Respon kecepatan ICE dapat dilihat pada Gambar 4.14.



Gambar 4.14 Respon ICE Berbeban Nominal Tanpa Bantuan Motor DC

4.3.2 Pengujian Menggunakan Kontroler *Neuro-Fuzzy* Tanpa Model Prediksi

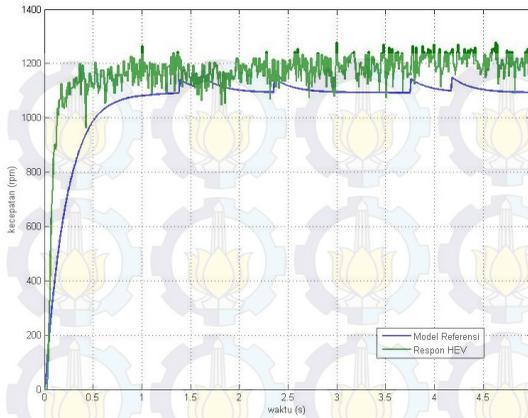
Berdasarkan Gambar 4.15, respon HEV mencapai model referensi pada $t = \pm 0,06$ s. Respon HEV mendahului model referensi dan memiliki *rise time* sebesar 0,129 s. Hal ini dikarenakan adanya pembatasan nilai tegangan motor DC sebesar 0 - 80 V.



Gambar 4.15 Respon Hasil Implementasi Kontroler *Neuro-Fuzzy* Tanpa Model Prediksi

4.3.3 Pengujian Menggunakan Kontroler *Neuro-Fuzzy* Menggunakan Model Prediksi

Berdasarkan Gambar 4.16, respon HEV mencapai model referensi pada $t = \pm 0,05$ s. Respon HEV mendahului model referensi dan memiliki *rise time* sebesar $\pm 0,12$ s. Hal ini dikarenakan adanya pembatasan nilai tegangan motor DC sebesar 0 - 80 V. Pada tahap simulasi, motor DC hanya bekerja pada kondisi transien. Sedangkan pada saat kondisi *steady* motor DC akan berhenti bekerja dikarenakan *error* antara model referensi dan ICE bernilai nol. Akan tetapi, pada tahap implementasi motor DC bekerja secara terus menerus dikarenakan nilai *error* antara model referensi dan ICE tidak bernilai nol pada kondisi *steady*.



Gambar 4.16 Respon Hasil Implementasi Kontroler *Neuro-Fuzzy* Menggunakan Model Prediksi

BAB 5 PENUTUP

5.1 Kesimpulan

Kontroler *Neuro-Fuzzy* dengan *learning rate* 0,01 s mampu melakukan adaptasi terhadap data target pada $t = \pm 0,025$ s. Sehingga kontroler *Neuro-Fuzzy* bisa diaplikasikan pada *plant* yang memiliki perubahan respon cepat.

Kontroler *Neuro-Fuzzy* Prediktif dapat membantu respon HEV mencapai model referensi pada $t = 0,051$ s. Dengan demikian, akselerasi HEV yang diharapkan dapat tercapai. Dalam kondisi tanpa bantuan motor DC, pada respon transien $t = 0,051$ s terdapat *error* sebesar 19,72%. Setelah ditambahkan motor DC yang dikontrol menggunakan *Neuro-Fuzzy* Prediktif *error* yang terjadi direduksi menjadi $\pm 1\%$.

Pada kontroler yang bersifat prediktif, kemampuan mencapai model referensi ditentukan oleh *time sampling* dan jumlah *prediction horizon*. Pada Tugas Akhir ini digunakan prediksi satu langkah ke depan dengan *time sampling* sebesar 0,05 s sehingga respon mulai mencapai model referensi pada $t = 0,051$ s.

Pada tahap implementasi, respon *output* HEV menggunakan kontroler *Neuro-Fuzzy* tanpa model prediksi dan respon *output* HEV menggunakan kontroler *Neuro-Fuzzy* dengan model prediksi memiliki nilai yang hampir sama. Hal ini dikarenakan terdapat pembatasan nilai tegangan yang masuk pada motor DC sebesar 80 V.

5.2 Saran

Untuk penelitian lebih lanjut, penentuan *invers model* dapat dilakukan dengan menggunakan *Neuro-Fuzzy*. Dengan menggunakan *Neuro-Fuzzy*, *invers model* dapat beradaptasi dengan perubahan parameter yang berubah pada saat implementasi.

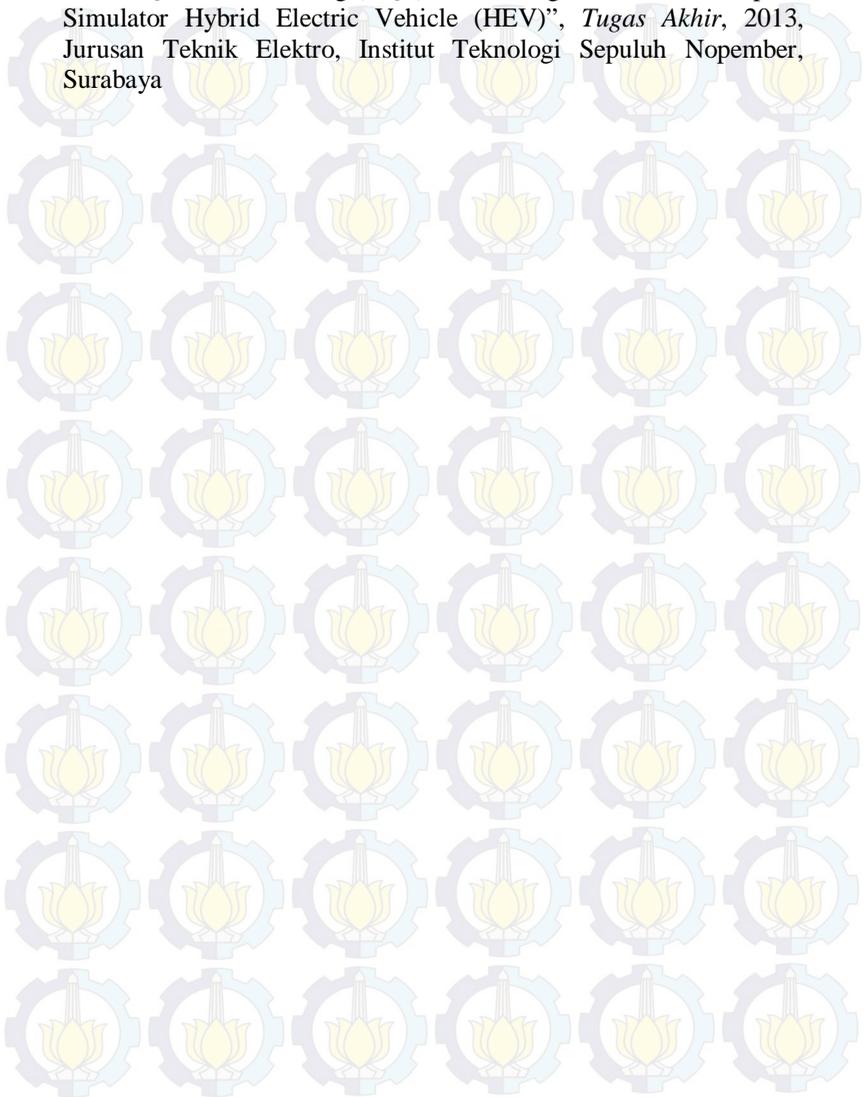


Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

- [1] Satrio Aji W., Galih, “Desain dan Implementasi Kontroler Optimal Berbasis Neuro Fuzzy untuk Pengendalian Simulator Hybrid Electric Vehicle”, *Tugas Akhir*, 2010, Jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember, Surabaya
- [2] Sriwidodo, Wahyu, “Perancangan dan Implementasi Pengendalian Traksi Berbasis Neuro Fuzzy Untuk Simulator Kendaraan Hybrid Pada Mode Akselerasi ”, *Tugas Akhir*, 2010, Jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember, Surabaya
- [3] Heywood, John B., *Internal Combustion Engine Fundamentals*, McGraw-Hill, Inc., 1998, New York
- [4] Austin Hughes and Bill Drury, *Electric Motors And Drives, Fundamentals, Types, And Applications Fourth Edition*, Elsevier, 2013, Oxford
- [5] Baharom, MZ , Nuawi, MZ, “Eddy Current Braking Experiment Using Brake Disc From Aluminium Series Of Al6061 And Al7075 ”, *1st International Conference on Mechanical Engineering Research*, 2011
- [6] Bandyopadhyay, M., N., *Electrical Power System*, PHI Learning Ltd., 2006
- [7] _____, *Index of Arduino Boards* <URL: <http://www.arduino.cc/en/Main/arduinoBoardUno>>, November 2013
- [8] Ing. Jakoubek, Pavel, “ Experimental Identification of Stable Nonoscillatory Systems from Step-Responses by Selected Methods”, *Konference studentské tvůrčí činnosti*, 2009
- [9] Ramadhan, R, “ Pengaturan Proses Tekanan Pada Sistem Pengaturan Berjaringan Menggunakan Kontroller Neuro-Fuzzy”, *Tugas Akhir*, 2012, Jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember, Surabaya
- [10] Kusumadewi, Sri, *Neuro-Fuzzy Edisi 2*, Graha Ilmu, 2010, Yogyakarta
- [11] Passino, Kevin M., *Fuzzy Control* , Addison Wesley, 1998, California
- [12] Herry, Mauridhy, *Supervised Neural Networks dan Aplikasinya*, Graha Ilmu, 2006, Yogyakarta
- [13] Ogata, K. *Modern Control Engineering 4th Edition*, Prentice Hall, 2002, New Jersey, USA

[14] Ramadhani, Nugroho A. “Perancangan dan Implementasi Kontroler Linear Quadratic Tracking (LQR) Untuk Pengaturan Akselerasi pada Simulator Hybrid Electric Vehicle (HEV)”, *Tugas Akhir*, 2013, Jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember, Surabaya



RIWAYAT PENULIS



Bayu Prasetyo Wibowo atau yang biasa dipanggil Bayu, lahir di Malang pada tanggal 2 Maret 1992. Ia merupakan anak kedua dari dua orang bersaudara. Pada tahun 1996, ia memulai pendidikannya di TK Laboratorium Universitas Negeri Malang. Dua tahun berikutnya, ia melanjutkan pendidikannya di SDN Percobaan I Malang. Pada tahun 2004, ia melanjutkan pendidikannya di SMP Negeri 3 Malang dan jenjang sekolah menengah atas dihabiskan di SMA Negeri 3 Malang. Setelah lulus sekolah menengah atas pada tahun 2010, ia melanjutkan pendidikannya di Institut Teknologi Sepuluh Nopember Surabaya (ITS) jurusan S1 Teknik Elektro. Saat ini, ia sedang menempuh tahun terakhir di bidang Teknik Sistem Pengaturan

Selain aktif di bidang akademis, ia juga aktif dalam kegiatan non-akademis. Dimulai dari tingkat sekolah dasar ia mulai aktif di kegiatan pramuka dan drumband. Berbagai perlombaan diikutinya mulai dari tingkat kecamatan hingga tingkat propinsi. Salah satu prestasi yang ditorehkan adalah menjadi juara II Kejuaraan Daerah Drum Band Propinsi Jawa Timur pada tahun 2004. Pada saat melanjutkan pendidikan di tingkat sekolah menengah pertama, ia lebih aktif di kegiatan keolahragaan. Ketika itu, ia aktif dalam ekstrakurikuler bulu tangkis. Ketika menempuh studi di jenjang sekolah menengah atas, kegiatan non-akademis yang diikuti juga tidak jauh dari olahraga yang menjadi kegemarannya. Di tahun pertama dan kedua dia aktif dalam ekstrakurikuler sepak bola dan juga kegiatan tennis lapangan di luar sekolah.

