



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - TE 141599

**PENGATURAN KECEPATAN MOTOR DC PADA SIMULATOR
HYBRID ELECTRIC VEHICLE (HEV) MENGGUNAKAN
KONTROLER PID BERDASARKAN ALGORITMA *PARTICLE
SWARM OPTIMIZATION* (PSO)**

Alif Ridwan Maulana
NRP 2211 100 117

Dosen Pembimbing
Ir. Rusdhianto Effendie A.K., M.T.
Ir. Ali Fatoni, M.T.

JURUSAN TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2015



FINAL PROJECT - TE 141599

***DC MOTOR SPEED CONTROL FOR HYBRID ELECTRIC
VEHICLE (HEV) SIMULATOR USING PID CONTROLLER
BASED ON PARTICLE SWARM OPTIMIZATION (PSO)
ALGORITHM***

Alif Ridwan Maulana
NRP 2211 100 117

Supervisors

Ir. Rusdhianto Effendie A.K., M.T.

Ir. Ali Fatoni, M.T.

ELECTRICAL ENGINEERING DEPARTMENT
Faculty of Industrial Technology
Sepuluh November Institute of Technology
Surabaya 2015

**PENGATURAN KECEPATAN MOTOR DC PADA
SIMULATOR *HYBRID ELECTRIC VEHICLE* (HEV)
MENGUNAKAN KONTROLER PID BERDASARKAN
ALGORITMA *PARTICLE SWARM OPTIMIZATION* (PSO)**

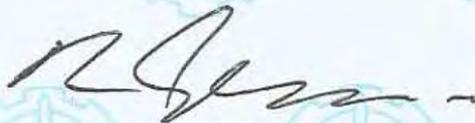
TUGAS AKHIR

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

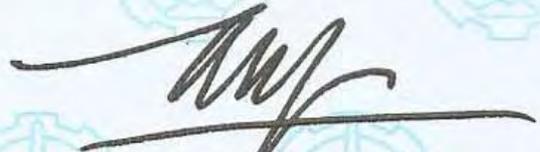
Menyetujui :

Dosen Pembimbing I,

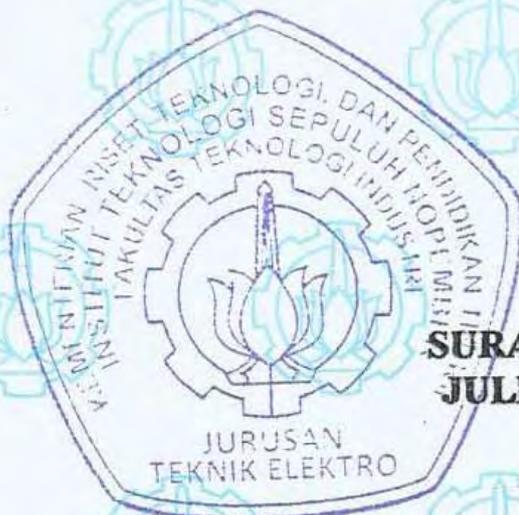
Dosen Pembimbing II,



Ir. Rusdianto Effendie A.K., M.T.
NIP. 1957 04 24 1985 02 1001



Ir. Ali Fatoni, M.T.
NIP. 1962 06/03 1989 03 1002



**SURABAYA
JULI, 2015**

**PENGATURAN KECEPATAN MOTOR DC PADA
SIMULATOR *HYBRID ELECTRIC VEHICLE* (HEV)
MENGUNAKAN KONTROLER PID BERDASARKAN
ALGORITMA *PARTICLE SWARM OPTIMIZATION* (PSO)**

Nama : Alif Ridwan Maulana
Pembimbing : 1. Ir. Rusdhianto Effendie A.K., M.T.
2. Ir. Ali Fatoni, M.T.

ABSTRAK

Salah satu solusi jangka pendek yang sedang dilakukan oleh industri automotif guna mengurangi ketergantungan dunia akan penggunaan bahan bakar minyak dan mengurangi emisi gas buang kendaraan adalah dengan adanya teknologi *Hybrid Electric Vehicle* (HEV). Namun HEV sendiri memiliki kelemahan ketika adanya pembebanan berlebih. Disamping itu, untuk menjaga performansi HEV agar tetap baik, maka digunakan kontroler yang tepat untuk menanggulangi permasalahan tersebut. Algoritma PSO digunakan dalam proses *tuning* parameter kontroler PID untuk pengaturan kecepatan pada simulator HEV. Oleh karena itu, dengan menggunakan algoritma PSO pada kontroler PID diharapkan mampu memperoleh nilai parameter K_p , K_i , dan K_d yang optimal. Kemudian dengan parameter PSO $w = 0,48$ jumlah partikel, $n = 10$ dan jumlah iterasi, $iter = 30$ diperoleh nilai parameter kontroler PID $K_p = 9,798$, $K_i = 20,863$, dan $K_d = 0,153$. Berdasarkan hasil pengujian secara simulasi, motor DC dapat membantu kinerja ICE terhadap adanya pembebanan berupa arus pengereman (0,7 A; 0,8 A; 0,9 A) yang selanjutnya diperoleh data performansi HEV berupa penurunan kecepatan putar sebesar 6 – 11 rpm, rata – rata *settling time* ($t_s = 1,6$ detik) serta $E_{ss} = 0\%$ dan *overshoot* = 0 rpm. Dari data performansi HEV tersebut, maka simulator HEV telah mampu mempertahankan kecepatan putar sesuai dengan *set point* yang diberikan.

Kata Kunci : *Hybrid Electric Vehicle*, Kontroler PID, *Particle Swarm Optimization*,

Halaman ini sengaja dikosongkan

**DC MOTOR SPEED CONTROL FOR HYBRID ELECTRIC
VEHICLE (HEV) SIMULATOR USING PID CONTROLLER
BASED ON PARTICLE SWARM OPTIMIZATION (PSO)
ALGORITHM**

Name : Alif Ridwan Maulana
Supervisors : 1. Ir. Rusdhianto Effendie A.K., M.T.
2. Ir. Ali Fatoni, M.T.

ABSTRACT

One short-term solution that is being conducted by the automotive industry to reduce the world's reliance will the use of fuel oil and reduce vehicle exhaust emissions are with the technology Hybrid Electric Vehicle (HEV). However HEV itself has drawbacks when the excess load. In addition, to maintain the performance of HEV in order to remain good, then use the right controller to overcome these problems. PSO algorithm used in the process PID controller tuning parameters for setting the pace on the simulator HEV. Therefore, using the PSO algorithm in PID controllers are expected to obtain the value of the parameter K_p , K_i , and K_d are optimal. Then the PSO parameter $w = 0.48$ number of particles, $n = 10$ and the number of iterations, $iter = 30$ parameter values obtained PID controller $K_p = 9.798$, $K_i = 20.863$, and $K_d = 0.153$. Based on test results in simulation, the DC motor can help the performance of ICE against the imposition of such braking current (0.7 A; 0.8 A; 0.9 A), hereinafter the data obtained in the form of a decrease in the performance of HEV rotational speed of 6-11 rpm, the average settling time ($t_s = 1.6$ sec) and $E_{ss} = 0\%$ and overshoot = 0 rpm. HEV performance of the data, then the HEV simulator has been able to maintain the rotational speed in accordance with a given set point.

Keywords : Hybrid Electric Vehicle, Particle Swarm Optimization, PID Controller,

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji syukur Kehadirat Alloh SWT yang selalu senantiasa memberikan kasih dan sayang – Nya sehingga penulis dapat menyelesaikan Tugas Akhir dengan judul *Pengaturan Kecepatan Motor DC pada Simulator Hybrid Electric Vehicle (HEV) Menggunakan Kontroler PID Berdasarkan Algoritma Particle Swarm Optimization (PSO)* tepat pada waktunya. Tak lupa Sholawat serta salam selalu tercurahkan kepada junjungan Nabi Muhammad SAW yang telah membimbing umatnya menuju jalan yang terang – benerang. Tugas Akhir yang telah disusun ini merupakan syarat yang harus dipenuhi dalam mencapai gelar Sarjana Teknik di Bidang Studi Teknik Sistem Pengaturan, Jurusan Teknik Elektro Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember (ITS) Surabaya.

Penulis juga menyadari bahwa isi buku Tugas Akhir ini masih sangat jauh dari sempurna. Untuk itu, penulis mohon maaf yang sebesar – besarnya atas kesalahan atau kekurangan dalam menulis buku ini. semoga penelitian dalam Tugas Akhir ini bisa dilanjutkan dan dikembangkan oleh generasi penerus sehingga didapatkan hasil yang lebih baik di kemudian hari. Oleh karena itu, penulis menyampaikan rasa terima kasih yang sebesar – besarnya kepada:

1. Bapak Ir. Rushianto Effendie A.K, MT. Dan Bapak Ir. Ali Fatoni, MT. Selaku pembimbing yang telah dengan arif dan sabar dalam mengarahkan dan membimbing penulis selama proses pengerjaan Tugas Akhir ini.
2. Kedua orang tua penulis yang selalu senantiasa menasehati dan mendo'akan penulis atas keberhasilan Tugas Akhir ini.
3. Hendra Antomy anak Pulung Ponorogo yang telah memberikan masukan dan pemahaman konsep selama masa perkuliahan.
4. Ajib Setiawan N., Sentosa Sondang O., Faniesha H., Mbak Aulia Rahma A., dan Mas Fahrul selaku rekan satu tim penulis dalam pengerjaan Tugas Akhir ini.
5. Tito, Adi, Dino, Dedy, Muji, Engkong, Saf, dan Lung selaku kawan berbagi ilmu di kontrakan.
6. Teman – teman dari Kediri, SP 2011, dan keluarga e-51 yang sudah memberikan cerita indah selama mejalani proses perkuliahan di kampus.

7. Semua pihak yang telah ikut membantu baik secara langsung maupun tidak langsung dalam pengerjaan Tugas Akhir ini.

Akhir kata, penulis sekali lagi menyampaikan mohon maaf atas segala kekurangan yang terdapat dalam buku Tugas Akhir ini. Semoga hasil Tugas Akhir ini dapat memberikan manfaat dalam pengembangan ilmu pengetahuan dan teknologi.

Surabaya, Juli 2015

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
PERNYATAAN KEASLIAN TUGAS AKHIR	v
HALAMAN PENGESAHAN	vii
ABSTRAK	ix
ABSTRACT	xi
KATA PENGANTAR	xiii
DAFTAR ISI	xv
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxi
BAB 1 PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Perumusan Masalah	2
1.3. Batasan Masalah	2
1.4. Tujuan	3
1.5. Sistematika Penulisan	3
1.6. Relevansi	4
BAB 2 DASAR TEORI	5
2.1. <i>Hybrid Electric Vehicle</i> (HEV)	5
2.2. Simulator HEV	7
2.2.1. <i>Internal Combustion Engine</i> (ICE) 2 Tak	7
2.2.2. Motor DC	10
2.2.3. Rem Elektromagnetik Arus Eddy	12
2.2.4. <i>Gear</i>	14
2.2.5. <i>Rotary Encoder</i>	16
2.3. Arduino Uno R3	17
2.4. Identifikasi Sistem	19
2.4.1. Metode Identifikasi Fisik	19
2.4.2. Metode Identifikasi Streg	21
2.4.3. <i>Root Mean Square Error</i> (RMSE)	23
2.5. Kontroler PID	24
2.6. <i>Particle Swarm Optimiation</i> (PSO)	25
BAB 3 PERANCANGAN SISTEM	31
3.1. Gambaran Umum Simulator HEV	31
3.2. Perancangan Perangkat Keras (<i>Hardware</i>).....	32
3.2.1. Perancangan Mekanik.....	32

3.2.2. Perancangan Elektronik.....	39
3.3. Perancangan Perangkat Lunak (<i>Software</i>).....	43
3.3.1. <i>Software</i> Arduino	44
3.3.2. <i>Software</i> LabVIEW	45
3.3.3. <i>Software</i> MATLAB.....	46
3.4. Pengujian Sensor	47
3.4.1. Pengujian Sensor <i>Rotary Encoder</i>	47
3.4.2. Pengujian Sensor Potensiometer	49
3.5. Identifikasi dan Pemodelan Sistem.....	49
3.5.1. Identifikasi Parameter Motor DC	50
3.5.2. Pemodelan Motor DC.....	55
3.5.3. Identifikasi Parameter Beban	56
3.5.4. Identifikasi ICE	58
3.5.5. Pemodelan ICE.....	59
3.6. Pengujian dan Validasi	65
3.7. Desain Kontroler PID – PSO.....	69
3.7.1. Algoritma PSO Untuk Tuning Parameter PID	69
3.7.2. Desain Simulink PID – PSO	72
BAB 4 PENGUJIAN DAN ANALISIS	73
4.1. Gambaran Umum Pengujian Sistem.....	73
4.2. Simulasi Sistem	73
4.2.1. Simulasi dengan <i>Tuning</i> PSO.....	73
4.2.2. Simulasi ICE Berbeban Tanpa Bantuan Motor DC.....	76
4.2.3. Simulasi ICE Berbeban dengan Bantuan Motor DC Menggunakan Kontroler PID - PSO	78
4.3. Implementasi Sistem	81
4.3.1. Implementasi ICE Berbeban Tanpa Bantuan Motor DC	81
4.3.2. Implementasi ICE Beberbean dengan Bantuan Motor DC Menggunakan Kontroler PID - PSO	82
BAB 5 PENUTUP	85
5.1. Kesimpulan.....	85
5.2. Saran	85
DAFTAR PUSTAKA.....	87
LAMPIRAN	89
A. Tampilan <i>Simulink</i> Simulasi HEV	89
B. Tampilan <i>Simulink</i> Motor DC.....	90
C. Program Particle Swarm Optimization.....	91

Main Program	91
Sub Program.....	95
RIWAYAT PENULIS.....	97

Halaman ini sengaja dikosongkan

DAFTAR TABEL

Tabel 2.1	Perbandingan Kapasitas dari Konfigurasi HEV	7
Tabel 2.2	Spesifikasi <i>board</i> Arduino Uno	18
Tabel 2.3	Nilai Konstan Rasio T	22
Tabel 2.4	Taksiran dari Orde ke $-n$ dan Nilai Titik Singgung	23
Tabel 2.5	Efek Tiap Parameter	25
Tabel 3.1	Spesifikasi ICE	34
Tabel 3.2	Spesifikasi Motor DC	35
Tabel 3.3	Data Hasil Pembacaan Kecepatan Putar dari Tachometer Digital dan <i>Rotary Encoder</i>	48
Tabel 3.4	Data Hasil Linearisasi dan Hasil Pengukuran	48
Tabel 3.5	Data Hasil Pengukuran untuk R_a , L_a , R_f , L_f	50
Tabel 3.6	Data Konstanta Balik Motor DC Hasil Percobaan	51
Tabel 3.7	Data Nilai Parameter I_{amax} , I_{fmax} , I_{anom} , dan I_{fnom}	52
Tabel 3.8	Data Nilai Parameter V_{in1} , V_{in2} , I_{a1} , I_{a2} , I_{f1} , dan I_{f2}	54
Tabel 3.9	Data Hasil Pengukuran Rem Elektromagnetik	57
Tabel 3.10	Persamaan Model dan Validasi <i>Plant ICE</i>	66
Tabel 3.11	Persamaan Model dan Validasi <i>Plant ICE</i>	67
Tabel 3.12	Persamaan Model dan Validasi <i>Plant ICE</i>	68
Tabel 3.13	Model <i>Plant ICE</i> yang Digunakan	69
Tabel 3.14	Pemodelan dengan Beberapa Metode	69
Tabel 4.1	Hasil <i>Tuning</i> K_p , K_i , dan K_d dengan PSO	74
Tabel 4.2	Variasi Arus Pengereman pada Rem Elektromagnetik	76
Tabel 4.3	Performansi HEV dengan Kontroler PID - PSO	79

Halaman ini sengaja dikosongkan

DAFTAR GAMBAR

Gambar 2.1	Konfigurasi HEV	6
Gambar 2.2	Planetary <i>Gear</i>	6
Gambar 2.3	Komponen pada Ruang Bakar ICE.....	8
Gambar 2.4	Prinsip Kerja ICE 2 Tak.....	9
Gambar 2.5	Gaya Lorentz pada Motor DC.....	11
Gambar 2.6	Karakteristik Motor DC	12
Gambar 2.7	Rem Elektromagnetik	13
Gambar 2.8	Gaya Pengereman oleh Arus Eddy	14
Gambar 2.9	Bentuk Fisik <i>Gear</i>	14
Gambar 2.10	Konfigurasi <i>Gear</i> Dua Poros	16
Gambar 2.11	Gambar Fisik <i>Rotary Encoder</i>	16
Gambar 2.12	<i>Board</i> Arduino Uno R3.....	17
Gambar 2.13	Tampilan <i>Framework</i> Arduino	19
Gambar 2.14	Konfigurasi Fisik Motor DC Paralel.....	20
Gambar 2.15	Diagram Blok Motor DC Paralel	20
Gambar 2.16	Respon Step Sistem dengan t_1 , t_2 , t_i , T_U , dan T_N	21
Gambar 2.17	Diagram Blok PID	25
Gambar 2.18	Diagram Alir Algoritma PSO	29
Gambar 3.1	Bentuk Fisik Simulator HEV	31
Gambar 3.2	Diagram Blok Pengaturan Simulator HEV	32
Gambar 3.3	Konfigurasi Perangkat Keras Simulator HEV	33
Gambar 3.4	Mesin Pemotong Rumput 2 Tak	33
Gambar 3.5	Bentuk Fisik Motor DC	34
Gambar 3.6	Bentuk Fisik Rem Elektromagnetik.....	35
Gambar 3.7	Konfigurasi <i>Gear</i> pada Simulator HEV	37
Gambar 3.8	Pedal Gas pada Simulator HEV	39
Gambar 3.9	<i>Optocoupler</i> dan Piringan Cacah	40
Gambar 3.10	Skema Rangkaian <i>Optocoupler</i>	41
Gambar 3.11	Sistem Minimum Arduino Uno R3.....	42
Gambar 3.12	Skema Rangkaian Penyearah Gelombang Penuh	42
Gambar 3.13	Bentuk Fisik Rangkaian Penyearah Gelombang Penuh	43
Gambar 3.14	Tampilan Program Arduino	44
Gambar 3.15	Tampilan Diagram Blok LabVIEW	45
Gambar 3.16	Menu Editor pada MATLAB.....	46
Gambar 3.17	Tampilan <i>Simulink</i>	46
Gambar 3.18	<i>Optocoupler</i> dan Piringan Hitam	47

Gambar 3.19	Grafik Kecepatan Putar Hasil Linearisasi dengan Hasil Pengukuran Tachometer Digital	49
Gambar 3.20	Diagram Blok Motor DC.....	55
Gambar 3.21	Respon Keluaran Motor DC saat Diberikan Masukan Sinyal Step	57
Gambar 3.22	Respon Kecepatan Putar ICE dengan Beban Minimal.....	61
Gambar 3.23	Respon Kecepatan Putar ICE dengan Beban Nominal.....	63
Gambar 3.24	Respon Kecepatan Putar ICE dengan Beban Maksimal.....	65
Gambar 3.25	Respon Hasil Identifikasi dan Pemodelan ICE.....	66
Gambar 3.26	Respon Hasil Identifikasi dan Pemodelan ICE.....	67
Gambar 3.27	Respon Hasil Identifikasi dan Pemodelan ICE.....	68
Gambar 3.28	Diagram Alir Algoritma PSO	71
Gambar 3.29	Diagram <i>Simulink</i>	72
Gambar 4.1	Konvergensi PSO saat Bobot Inersia $w = 0,38$	75
Gambar 4.2	Konvergensi PSO saat Bobot Inersia $w = 0,48$	75
Gambar 4.3	Konvergensi PSO saat Bobot Inersia $w = 0,58$	76
Gambar 4.4	Respon Kecepatan Putar ICE dengan 3 Variasi Beban Arus Pengereman	77
Gambar 4.5	Besar Arus Pengereman yang Diberikan.....	77
Gambar 4.6	Sinyal Kesalahan ICE dengan 3 Variasi Beban Arus Pengereman.....	78
Gambar 4.7	Respon Kecepatan Putar HEV dengan 3 Variasi Beban Arus Pengereman	79
Gambar 4.8	Sinyal Kontrol yang Diberikan pada Motor DC	80
Gambar 4.9	Sinyal Kesalahan dari Sistem dengan 3 Variasi Beban Arus Pengereman	80
Gambar 4.10	Respon Kecepatan Putar ICE dengan Beban 0,9 A	81
Gambar 4.11	Respon Kecepatan Putar HEV dengan Kontroler PID – PSO dengan Beban Arus Pengereman 0,9 A	82

BAB 1

PENDAHULUAN

Tugas Akhir merupakan salah satu bentuk penelitian mandiri yang dilakukan oleh mahasiswa pada jenjang S-1. Tugas Akhir sendiri merupakan syarat wajib untuk menyelesaikan studi dalam program sarjana teknik. Untuk Tugas Akhir yang penulis kerjakan berupa perancangan dan penelitian.

Pada bab ini akan dijelaskan mengenai hal-hal yang mengawali pelaksanaan Tugas Akhir. Hal-hal tersebut meliputi latar belakang, perumusan masalah, batasan masalah, tujuan, sistematika penulisan, dan relevansi.

1.1. Latar Belakang

Jumlah populasi penduduk yang semakin meningkat dan diiringi dengan bertambahnya jumlah penggunaan kendaraan bermotor, memaksa ketergantungan terhadap bahan bakar minyak (BBM) pada sistem transportasi juga semakin meningkat, khususnya transportasi di darat. Selain itu, kendaraan yang mengkonsumsi bahan bakar minyak sebagai sumber tenaganya bisa dipastikan gas buangnya menghasilkan emisi gas berbahaya yang menjadi salah satu penyebab polusi udara. Di sisi lain, cadangan minyak bumi yang juga semakin berkurang membuat dunia industriomotif harus memikirkan solusi yang tepat untuk mengatasi keadaan tersebut. Untuk itu, salah satu solusi jangka pendek yang diberikan oleh industriomotif dalam mengurangi kebutuhan akan bahan bakar minyak dan jumlah emisi gas buang kendaraan di masa depan adalah dengan adanya konsep teknologi *Hybrid Electric Vehicle* (HEV) [1].

Kendaraan *Hybrid Electric Vehicle* (HEV) dengan mesin *hybrid* merupakan kendaraan yang menggunakan dua atau lebih tenaga sebagai sumber penggerak, dalam hal ini gabungan antara mesin bakar atau *Internal Combustion Engine* (ICE) yang ada pada kendaraan umumnya dengan motor listrik atau *Electric Motor* (EM). Pada mesin *hybrid*, tenaga yang digunakan untuk menggerakkan kendaraan bisa berasal dari mesin bakar, motor listrik, maupun gabungan diantara keduanya. Dengan adanya mesin *hybrid* maka mesin bakar tidak lagi bekerja sendiri untuk menopang tenaga dalam menggerakkan kendaraan tetapi juga dibantu oleh motor listrik. Di samping itu, kelebihan dari mesin *hybrid* sendiri

yaitu ketika mesin bakar tidak mampu mempertahankan kecepatan maka motor listrik akan membantu memberikan tenaga. Dengan kombinasi kedua sumber tenaga penggerak tersebut, diharapkan jumlah kebutuhan energi minyak dan emisi gas buang kendaraan dapat berkurang.

Untuk menghasilkan performansi yang baik, maka PID digunakan sebagai kontroler dalam membantu mengatur kecepatan motor listrik pada kendaraan HEV, sehingga diharapkan dapat membantu ICE ketika terjadi pembebanan berlebih yang dapat menurunkan kecepatan putar kendaraan HEV. Kemudian untuk memperoleh nilai dari parameter K_p , K_i , dan K_d yang optimal, maka digunakanlah metode komputasi cerdas *Particle Swarm Optimization* (PSO) [2]. Sehingga dari penggunaan metode komputasi cerdas PSO pada kontroler PID tersebut diharapkan mampu mengatur kecepatan motor listrik untuk membantu ICE ketika terjadi pembebanan berlebih yang dapat mengakibatkan penurunan kecepatan kendaraan.

1.2. Perumusan Masalah

Pada saat kendaraan berada pada medan yang berat, seperti harus melalui banyak tanjakan. Kecepatan dari kendaraan secara tidak langsung akan mengalami penurunan yang dikarenakan adanya beban berlebih pada kendaraan tersebut. Kondisi seperti itu juga akan mempengaruhi performansi dari kendaraan tersebut. Di mana akan dibutuhkan tenaga tambahan untuk mampu melakukan akselerasi dengan cepat saat terjadi pembebanan yang berlebih sehingga nantinya kecepatan kendaraan dijaga tetap sesuai dengan laju yang dikehendaki. Dan kendaraan *hybrid* mampu untuk melakukan hal tersebut, karena pada mesin *hybrid* terdapat motor listrik yang mampu membantu mesin bakar dalam memenuhi kebutuhan kecepatan ketika terjadi pembebanan lebih. Pada simulator HEV, motor listrik dan mesin bakar jenis ICE mampu bekerja secara bersama – sama untuk menopang beban.

Oleh karena itu, dibutuhkan suatu kontroler yang mampu menjaga performa dari kecepatan putar HEV. Salah satunya dengan mengatur kecepatan putar motor listrik untuk membantu kinerja ICE. Pada Tugas Akhir ini digunakan kontroler PID untuk mengatur kecepatan motor listrik. Kemudian untuk nilai dari parameter kontroler (K_p , K_i , dan K_d) akan diperoleh dengan menggunakan metode pencarian *Particle Swarm Optimization* (PSO). Selain itu, dalam sistem simulator HEV untuk

merepresentasikan dari adanya pengaruh jalan yang menanjak, maka ditambahkan rem elektromagnetik untuk pembebanan.

1.3. Batasan Masalah

Pada penyelesaian Tugas Akhir ini diberi beberapa batasan masalah sebagai berikut:

1. *Plant* yang digunakan adalah simulator HEV yang terdiri dari ICE sebagai penggerak utama dan motor DC sebagai penggerak pembantu serta rem elektromagnetik sebagai pemberi efek pembebanan.
2. Pada simulator HEV, ICE telah dapat mencapai set point yang diinginkan, dan efek beban dari rem elektromagnetik akan menyebabkan penurunan kecepatan ICE dan motor DC bekerja sebagai penggerak pembantu dalam mengembalikan kecepatan simulator HEV.
3. Penghitungan parameter K_p , K_i , dan K_d dilakukan secara *offline*.

1.4. Tujuan

Tujuan pada penelitian Tugas Akhir ini adalah untuk merancang simulasi dan mengimplementasikan kontroler PID berdasarkan algoritma *Particle Swarm Optimization* (PSO). Selain itu, dari hasil penerapan metode PSO dalam *tuning* parameter PID dapat memberikan hasil yang optimal. Di mana mampu membantu memperbaiki performa sistem yang dihasilkan, terutama dalam permasalahan regulator berupa pengaturan kecepatan putar HEV pada kondisi pembebanan berlebih.

1.5. Sistematika Penulisan

Sistematika penulisan dalam pembahasan Tugas Akhir ini terdiri dari lima bagian, yaitu:

BAB 1 : PENDAHULUAN

Pada bab ini dijelaskan tentang latar belakang, perumusan masalah, batasan masalah, tujuan, sistematika penulisan dan relevansi.

BAB 2 : DASAR TEORI

Berisi tentang teori dasar yang meliputi *Hybrid Electric Vehicle*, kontroler PID, serta *Particle Swarm Optimization* yang akan digunakan sebagai landasan dasar dalam menyusun Tugas Akhir ini.

BAB 3 : PERANCANGAN SISTEM

Pada bab 3 ini berisi penjelasan mengenai perancangan *hardware* (perancangan komponen mekanik dan perancangan komponen elektronik) serta *software* (MATLAB, LABVIEW, dan Arduino) yang digunakan dalam pengerjaan Tugas Akhir.

BAB 4 : PENGUJIAN DAN ANALISIS

Bab ini memuat tentang hasil simulasi dan implementasi untuk penerapan dari kontroler PID – PSO pada *plant* simulator HEV.

BAB 5 : KESIMPULAN DAN SARAN

Berisi pemaparan kesimpulan dan saran dari hasil pembahasan yang telah diperoleh serta sebagai pertimbangan bagi pembaca untuk pengembangan dari Tugas Akhir ini selanjutnya.

1.6. Relevansi

Diharapkan Tugas Akhir ini dapat memberikan manfaat dan referensi untuk penelitian selanjutnya terutama dalam strategi kontrol untuk mengatur kecepatan HEV sehingga performa regulasi yang dihasilkan lebih optimal.

BAB 2

DASAR TEORI

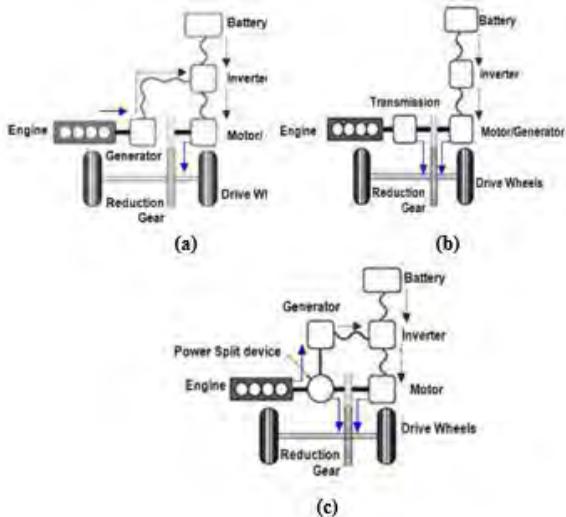
Pada Bab 2 akan dipaparkan mengenai teori-teori dasar yang digunakan sebagai landasan dalam pengerjaan Tugas Akhir ini. Diantaranya teori mengenai *Hybrid Electric Vehicle* (HEV), *Internal Combustion Engine* (ICE), kontroler PID, serta *Particle Swarm Optimization* (PSO) akan dibahas pada bab ini.

2.1. *Hybrid Electric Vehicle (HEV)* [3] [4]

Pada umumnya, perbedaan konfigurasi antara *Hybrid Electric Vehicle* (HEV) dan kendaraan bermotor konvensional salah satunya terletak pada tenaga penggerak yang digunakan. Untuk kendaraan HEV menggunakan mesin bakar dan motor listrik, sedangkan kendaraan bermotor konvensional hanya menggunakan mesin bakar saja. HEV merupakan suatu kendaraan yang menggunakan *Internal Combustion Engine* (ICE) yang menjadi tenaga penggerak utama dan motor listrik sebagai pendukung. Bisa dikatakan jika dalam situasi tertentu HEV bisa menggunakan hanya satu sumber penggerak dan bisa juga menggunakan sumber penggerak pendukung dalam situasi yang berbeda.

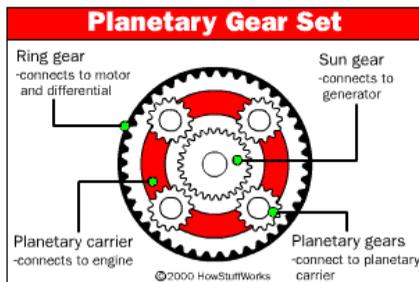
Berdasarkan konfigurasinya HEV dapat dibagi menjadi tiga yaitu konfigurasi seri, konfigurasi paralel, dan konfigurasi seri-paralel. Kemudian untuk konfigurasi seri, sumber tenaga penggerak kendaraan diperoleh dari motor listrik. Arus listrik untuk mendrive motor diperoleh dari pembakaran mesin dengan menggunakan bahan bakar. Energi kinetik hasil pembakaran mesin akan diubah menjadi energi listrik oleh generator yang selanjutnya akan disimpan di dalam baterai untuk digunakan sebagai penggerak motor. Jadi mesin bakar hanya digunakan untuk menghasilkan energi listrik, sedangkan untuk penggerak kendaraan menggunakan motor listrik. Konfigurasi seri ditunjukkan oleh Gambar 2.1(a).

Selanjutnya untuk konfigurasi paralel, tenaga yang digunakan sebagai sumber penggerak bisa menggunakan mesin bakar, motor listrik, atau kombinasi dari keduanya karena mesin bakar dan motor listrik berada pada satu poros yang sama. Dalam konfigurasi jenis ini pengaturan kerja motor listrik penting sehingga tidak memboroskan energi yang tersimpan di dalam baterai. Karena motor listrik pada konfigurasi ini bisa difungsikan sebagai motor maupun generator. Untuk konfigurasi paralel sendiri ditunjukkan pada Gambar 2.1(b).



Gambar 2.1 Konfigurasi HEV

Untuk konfigurasi seri-paralel ini yang ditunjukkan pada Gambar 2.1(c) disebut juga dengan konfigurasi *power split*, di mana pada konfigurasi ini terdapat perbedaan utama dari konfigurasi yang lain yaitu dengan adanya *power split device*. Komponen ini mampu membagi alur energi yang dihasilkan oleh sumber sesuai dengan kebutuhan kendaraan. Seperti pada kendaraan *hybrid* Toyota Prius, komponen tersebut berupa *gear* yang memiliki bentuk menyerupai sistem tata surya yang Toyota menyebutnya *planetary gear*. Komponen dari *planetary gear* dapat dilihat pada Gambar 2.2.



Gambar 2.2 Planetary Gear

Pada Tabel 2.1 dapat dilihat perbandingan kapasitas konfigurasi seri, paralel, dan seri-paralel (*power split device*).

Tabel 2.1 Perbandingan Kapasitas dari Konfigurasi HEV

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

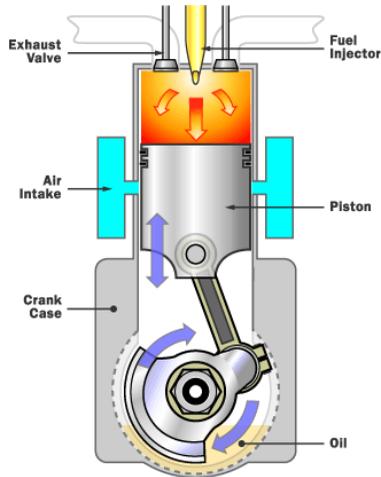
Pada penelitian Tugas Akhir ini simulator HEV yang akan digunakan memiliki konfigurasi paralel. Sumber penggerak pada simulator adalah mesin bakar dengan didukung motor listrik.

2.2. Simulator HEV

Pada penelitian ini simulator HEV yang digunakan terdiri dari *Internal Combustion Engine* (ICE) 2 langkah sebagai penggerak utama, motor DC sebagai elemen penggerak pendukung serta beban berupa rem elektromagnetik.

2.2.1. *Internal Combustion Engine* (ICE) 2 Langkah [5]

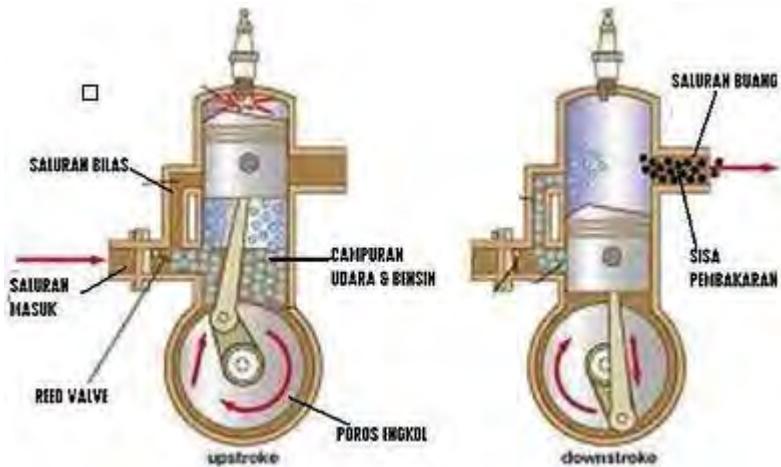
Motor bakar merupakan mesin yang menggunakan energi termal untuk melakukan kerja mekanik, yaitu dengan cara mengubah energi kimia dari bahan bakar menjadi energi panas, dan menggunakan energi tersebut untuk melakukan kerja mekanik. Untuk jenis motor bakar sendiri dibagi menjadi 2 yaitu motor bakar dalam atau *Internal Combustion Engine* (ICE) dan motor bakar luar atau *External Combustion Engine* (ECE). Dan pada Tugas Akhir ini jenis motor bakar yang digunakan adalah ICE. Pada ICE, proses pembakaran bahan bakar terjadi di dalam mesin itu sendiri sehingga panas dari hasil pembakaran langsung bisa diubah menjadi energi mekanik. Untuk komponen ICE ini dapat dilihat pada Gambar 2.3.



Gambar 2.3 Komponen pada Ruang Bakar ICE

Secara umum motor bakar jenis ICE dibagi menjadi 2, yaitu mesin 2 langkah dan mesin 4 langkah. Pada mesin 2 langkah tahap kerjanya yaitu motor bakar memerlukan dua kali langkah torak yang berupa satu kali ke atas (*ascending stroke*) dan satu kali ke bawah (*descending stroke*) untuk mendapatkan satu kali usaha di ruang pembakaran. Piston yang bergerak naik dari titik mati bawah (TMB) ke titik mati atas (TMA) menyebabkan saluran hisap dan saluran buang akan tertutup. Dalam hal ini gas yang berada dalam ruang pembakaran dikompresikan.

Di sisi lain, untuk gas yang baru masuk ke dalam ruang engkol, beberapa derajat sebelum piston mencapai titik mati atas, busi akan melontarkan bunga api sehingga akan terjadi pembakaran bahan bakar. Kemudian untuk tampilan prinsip kerja ICE 2 langkah dapat dilihat pada Gambar 2.4. Kelemahan ICE 2 langkah yaitu pada saat campuran bahan bakar dan udara memasuki ruang bakar, terjadi pencampuran dengan gas buang hasil pembakaran siklus sebelumnya, akibatnya mengurangi tenaga keluaran motor bakar.



Gambar 2.4 Prinsip Kerja ICE 2 langkah

Prinsip kerja dari ICE 2 langkah tersebut secara runtut akan dijelaskan pada penjelasan di bawah ini:

a. Tahap Pengisapan dan Kompresi

Awalnya piston bergerak ke atas sehingga ruang bawah piston menjadi hampa udara, akibatnya udara dan campuran bahan bakar terisap masuk ke dalam ruang di bawah piston. Selanjutnya, dibagian ruang atas piston terjadi langkah kompresi, sehingga udara dan campuran bahan bakar yang sudah berada di ruang atas piston, suhu dan tekanannya menjadi naik. Pada saat 10-5 derajat sebelum TMA, busi akan memercikan bunga api serta mengakibatkan campuran udara dan bahan bakar yang telah naik temperatur dan tekanannya menjadi terbakar dan meletup – letup.

b. Tahap Kerja dan Buang

Dari hasil pembakaran sebelumnya membuat piston bergerak ke bawah. Ketika piston bergerak ke bawah, ruang di bawah piston menjadi dimampatkan / dikompresikan akibatnya campuran udara dan bahan bakar yang berada di ruang bawah piston menjadi terdesak keluar dan naik ke ruang di atas piston melalui saluran bilas. Sedangkan sisa hasil pembakaran sebelumnya akan terdorong keluar dan menuju saluran buang, dan akhirnya menuju knalpot.

2.2.2. Motor DC [6]

Motor listrik merupakan perangkat elektromagnetis yang digunakan untuk mengubah energi listrik menjadi energi gerak. Motor listrik sendiri dibagi menjadi dua jenis, yaitu motor arus bolak-balik atau *Alternating Current* (AC) dan motor arus searah atau *Direct Current* (DC). Dan yang digunakan pada Tugas Akhir ini adalah motor listrik arus searah (DC). Untuk tipe motor DC yang digunakan adalah motor DC *Shunt*. Motor DC memerlukan suplai tegangan yang searah pada kumparan medan untuk diubah menjadi energi mekanik. Kumparan medan pada motor DC terdiri dari *stator* (bagian yang tidak berputar) dan kumparan jangkar yang disebut *rotor* (bagian yang berputar).

Pada motor DC magnet permanen terdiri dari magnet permanen, kumparan jangkar, dan sikat (*brush*). Terdapat dua prinsip dasar yang melatar belakangi kerja dari motor DC. Medan magnet menghasilkan magnet permanen yang nilainya konstan, sedangkan komutator dan sikat berfungsi menyalurkan arus listrik dari sumber di luar motor ke dalam kumparan jangkar. Untuk medan stator menghasilkan ϕ dari kutub *U* ke kutub *S*. Selanjutnya sikat arang menyentuh terminal kumparan *rotor* di bawah kutub. Bila sikat arang dihubungkan pada satu sumber arus searah di luar tegangan *V*, maka arus *I* masuk ke terminal kumparan *rotor* di bawah kutub *U* dan keluar dari terminal di bawah kutub *S*.

Dengan adanya fluks *stator* dan arus *rotor* akan menghasilkan satu gaya *F* bekerja pada kumparan yang dikenal dengan gaya Lorentz. Besar gaya Lorentz yang dihasilkan dapat ditentukan berdasarkan Persamaan (2.1).

$$F = BIL \sin \theta \quad (2.1)$$

Dengan,

F : Gaya Lorent (N)

B : Kerapatan fluks magnet (Weber/m²)

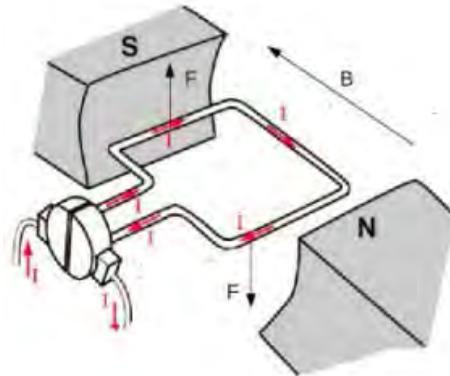
I : Arus yang mengalir (Ampere)

L : Panjang kawat penghantar (m)

θ : Sudut yang dibentuk antara arah arus medan dengan arah arus yang mengalir pada kawat (derajat)

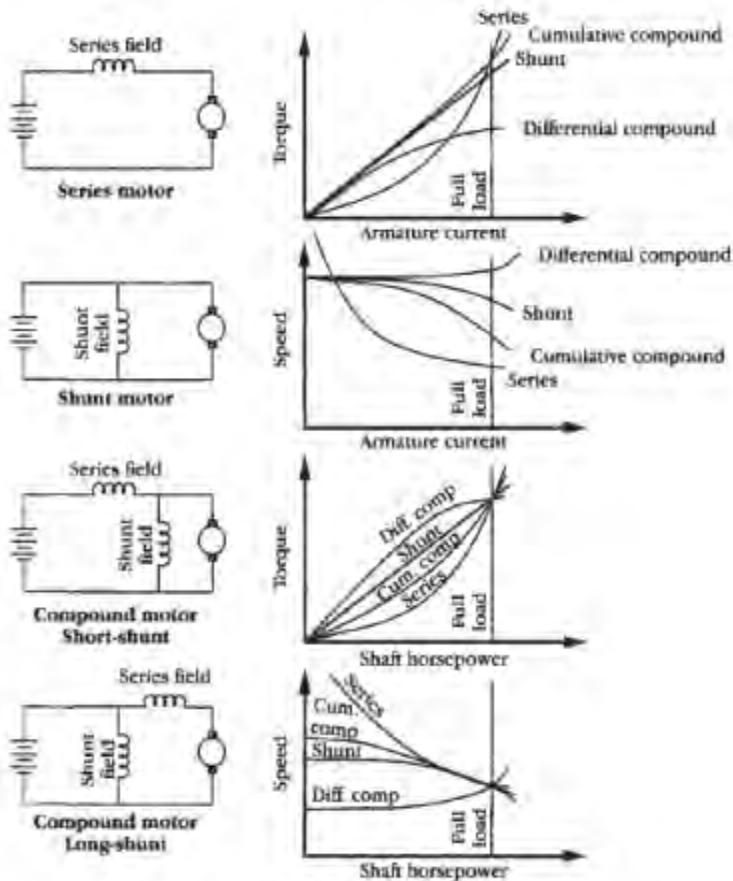
Pada saat arus mengalir melewati kawat yang memotong medan magnet seperti Gambar 2.5, maka akan terjadi gaya Lorentz. Dari pada itu gaya Lorentz inilah yang menimbulkan torsi pada *shaft* motor. Karena

dasarnya motor DC membangkitkan torsi, maka seringkali disebut sebagai pembangkit torsi. Selain itu, terdapat pula komutator yang berfungsi untuk membalik polaritas, yang menyebabkan arus mengalir dengan arah yang berlawanan dari sebelumnya. Komutator secara fisik berupa segmen-segmen, biasanya terbuat dari tembaga dan terletak pada *rotor*.



Gambar 2.5 Gaya Lorentz pada Motor DC

Kemudian untuk konstruksi dari motor DC dapat dibedakan menjadi *brushed*, *brushless*, dan magnet permanen. Motor DC *brushless* tidak menggunakan komutator dan sikat arang untuk memberikan daya listrik dari sumber listrik ke *rotor*, serta motor DC magnet permanen menggunakan magnet permanen di *stator* sebagai penghasil medan magnet. Karena meniadakan kumparan medan, maka memungkinkan motor memiliki bobot yang ringan dan ukuran yang lebih kecil. Motor DC *brushed* karena konstruksinya yang sederhana, menjadikan motor DC *brushed* berharga lebih murah, dan juga kemampuannya yang cukup baik, maka motor tipe ini lebih sering dipakai di industri. Motor DC *brushed* terdiri dari beberapa macam konfigurasi, yaitu *shunt*, *compound*, dan seri. Dan ketiganya memiliki karakteristik yang berbeda-beda. Karakteristik masing-masing konfigurasi dapat dilihat pada Gambar 2.6.



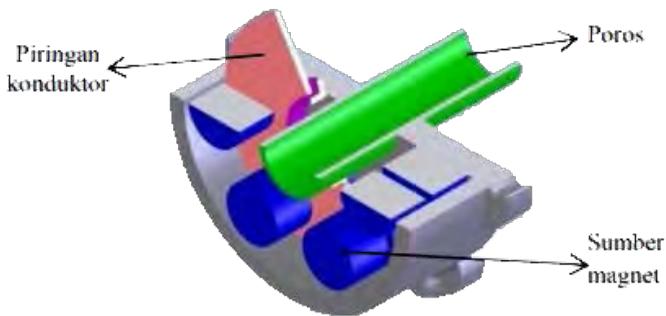
Gambar 2.6 Karakteristik Motor DC

2.2.3. Rem Elektromagnetik Arus Eddy [7]

Rem merupakan suatu komponen yang digunakan untuk melakukan aksi deselerasi yang mampu menurunkan kecepatan dalam selang waktu tertentu. Secara umum, tipe rem yang digunakan adalah rem yang menggunakan gaya gesek untuk memberikan gaya lawan terhadap gaya gerak. Model yang umum digunakan berupa piringan cakram ataupun model drum (tromol). Sebuah piringan dengan bahan logam non-

feromagnetik terpasang dengan poros yang berputar. Piringan tersebut diapit oleh sisi *stator* berupa sistem lilitan elektromagnetik yang dapat membangkitkan medan magnet dari aliran listrik. Arus listrik menimbulkan medan magnet pada lilitan dan logam piringan yang memotong medan magnet tersebut akan menimbulkan arus eddy pada piringan itu sendiri.

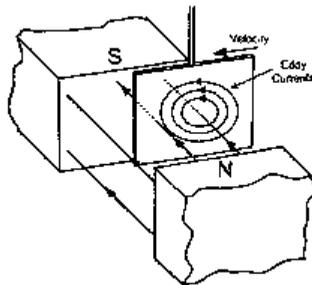
Kemudian arus eddy ini akan menimbulkan medan magnet yang arahnya berlawanan dengan medan magnet sebelumnya, sehingga menghambat gerakan putar dari poros tersebut. Rem elektromagnetik akan optimal untuk memberikan penurunan kecepatan, bukan untuk menghentikan gerak suatu objek. Oleh karena itu, rem ini sering diaplikasikan untuk sistem pengereman pada kereta api, dan juga digunakan pada alat *dinamometer* untuk pengukuran torsi suatu mesin. Komponen dasar dari rem elektromagnetik ditunjukkan oleh Gambar 2.7. Arus eddy yang melingkar menyebabkan medan magnet induksi melawan arah medan magnet mula-mula. Hal ini menyebabkan terjadinya gaya pengereman yang melawan arah kecepatan konduktor yang bergerak memotong medan magnet dari kedua solenoid.



Gambar 2.7 Rem Elektromagnetik

Dan untuk gaya pengereman yang dihasilkan oleh arus eddy yang melingkar ditunjukkan pada Gambar 2.8, di mana terdapat medan magnet yang arahnya menjauhi pengamat. Kemudian sebuah konduktor memotong medan magnet tersebut dengan kecepatan tertentu. Berdasarkan hukum Faraday, apabila terjadi perubahan medan magnet, maka akan timbul GGL pada konduktor. Pada konduktor, bidang yang akan mengalami perubahan fluks magnet hanya pada kedua sisinya, yang

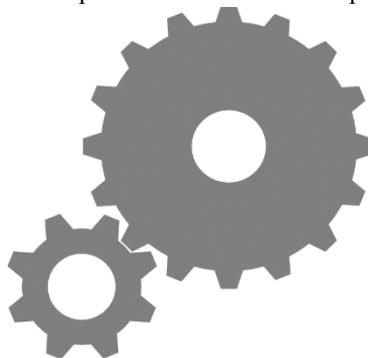
pertama adalah saat keluar dari medan magnet dan yang kedua adalah saat memasuki medan magnet. Di sisi lain, bagian tengah konduktor tidak mengalami perubahan fluks magnet sehingga tidak timbul lagi. Dengan artian, gaya lawan hanya dihasilkan apabila permukaan tersebut memiliki kecepatan. Semakin tinggi kecepatan maka gaya lawan yang dihasilkan juga semakin besar. Tetapi semakin rendah kecepatan, maka gaya lawan akan semakin kecil.



Gambar 2.8 Gaya Pengereman oleh Arus Eddy

2.2.4. Gear [8]

Pada umumnya sering kali kita menginginkan adanya perubahan kecepatan sudut atau torsi, baik itu memperbesar ataupun memperkecil nilai kecepatan sudut atau torsi tersebut. Dengan menggunakan rangkaian *gear*, maka hal tersebut dapat dilakukan. *Gear* merupakan suatu lingkaran



Gambar 2.9 Bentuk Fisik *Gear*

dengan gerigi di lingkaran luarnya. Bentuk fisik *gear* dapat dilihat pada Gambar 2.9.

Dengan konfigurasi dua buah *gear* seperti ini, maka aturan yang berlaku adalah sebagai berikut:

$$V_a = V_b \quad (2.2)$$

$$\omega_a R_a = \omega_b R_b$$

$$\frac{\omega_a}{\omega_b} = \frac{R_b}{R_a} \quad (2.3)$$

Dimisalkan $R_a = 4R_b$

$$\frac{\omega_a}{\omega_b} = \frac{R_b}{4R_b}$$

$$\omega_a = \frac{1}{4}\omega_b \quad (2.4)$$

Untuk aturan gaya yang bekerja pada *gear* adalah sebagai berikut:

$$F_a = F_b \quad (2.5)$$

$$\frac{\tau_a}{R_a} = \frac{\tau_b}{R_b}$$

$$\frac{\tau_a}{\tau_b} = \frac{R_a}{R_b} \quad (2.6)$$

Misalkan $R_a = 4R_b$

$$\frac{\tau_a}{\tau_b} = \frac{4R_b}{R_b}$$

$$\tau_a = 4\tau_b \quad (2.7)$$

dengan:

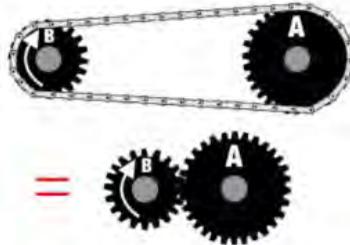
V_a = Kecepatan linier pada *gear* A

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

Untuk konfigurasi dari dua buah gear ini ditunjukkan dengan Gambar 2.10. Kedua buah poros gear ini dihubungkan dengan menggunakan *chain (belt)*.

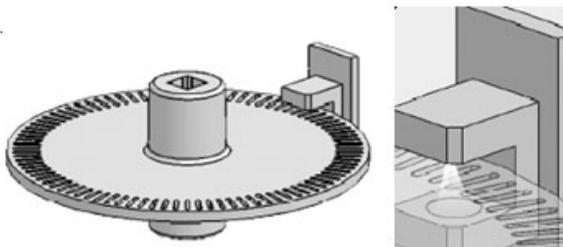


Gambar 2.10 Konfigurasi Gear Dua Poros

2.2.5. Rotary Encoder [9]

Rotary encoder merupakan salah satu jenis sensor kecepatan yang umum digunakan. Dan *rotary encoder* sendiri terdiri dari dua bagian utama yaitu *optocoupler* dan sebuah piringan pencacah tipis. Kemudian pada *optocoupler* terdapat *photo-transistor* dan LED.

Cara kerja dari sensor ini yaitu, ketika cahaya LED menembus lubang pada piringan, maka *photo-transistor* akan mengalami saturasi dan



Gambar 2.11 Gambar Fisik *Rotary Encoder*

akan menghasilkan suatu pulsa gelombang persegi. Gambar 2.11 menunjukkan bentuk fisik dari *rotary encoder* yang berupa bagian *optocoupler* dan piringan pencacah. Semakin banyak deretan pulsa yang dihasilkan pada satu putaran maka dapat menentukan akurasi *rotary encoder* tersebut. Sehingga semakin banyak jumlah lubang yang dapat dibuat pada piringan maka akurasi dari *rotary encoder* akan semakin baik.

2.3. Arduino Uno R3 [10]

Arduino Uno merupakan sebuah *board* mikrokontroler berbasis ATmega 328. Arduino Uno memiliki 14 pin digital *input / output* (dengan 6 pin dapat digunakan sebagai *output* PWM), 6 analog *input*, 16 MHz *ceramic resonator*, konektor USB, *jack power*, *ICSP header*, dan tombol *reset*. Arduino Uno berisi semua yang dibutuhkan untuk mendukung kerja dari mikrokontroler, dapat terhubung ke komputer dengan kabel USB atau menggunakan sumber dari AC de DC *adapter*. Pada Gambar 2.12 ditunjukkan bentuk fisik dari Arduino Uno R3.



Gambar 2.12 *Board* Arduino Uno R3

Untuk Arduino Uno R3 ini merupakan seri dari Arduino Uno revisi ketiga dengan memiliki fitur-fitur baru sebagai berikut:

- a. Pin out 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.

Kemudian untuk spesifikasi dari Arduino Uno ditunjukkan Tabel 2.2.

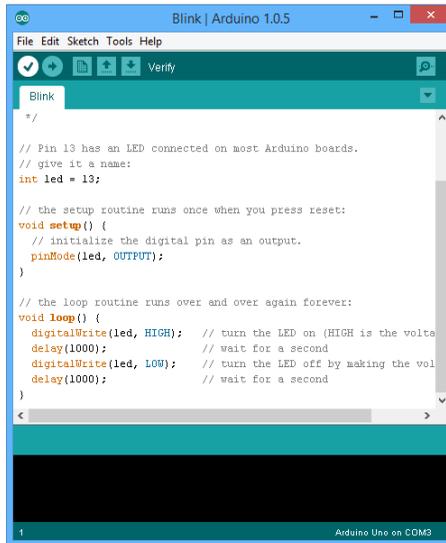
Tabel 2.2 Spesifikasi *board* Arduino Uno

Spesifikasi	Keterangan
Mikrokontroler	ATMega 328
Tegangan Operasi 5 V	5 V
Tegangan Masukan (rekomendasi)	7 – 12 V
Tegangan Masukan (batas)	6 – 20 V
Pin I/O Digital	14 (6 diantaranya 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) dengan 0.5 KB digunakan untuk bootloader
SRAM	2 KB (ATmega 328)
EEPROM 1 KB (Atmega 328)	1 KB (ATmega 328)
Kecepatan Clok	16 MHz

Arduino Uno memiliki beberapa fasilitas untuk berhubungan dengan komputer, Arduino lainnya, dan mikrokontroler lainnya. ATMega 328 menyediakan UART TTL (5V) komunikasi serial, dengan tersedianya 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.

Kemudian arduino dapat diprogram dengan menggunakan *software processing* dalam menuliskan program kedalam arduino. *Processing* sendiri merupakan penggabungan antara C++ dan Java. *Software* arduino ini dapat *di-install* pada berbagai *operating system* (OS) seperti: LINUX, Mac OS, dan Windows. *Software* IDE arduino terdiri dari tiga bagian, yaitu editor program, *compiler*, dan *uploader*. Struktur perintah pada arduino sendiri terdiri dari dua bagian yaitu *void setup* dan *void loop*. *Void setup* berisi perintah yang akan dieksekusi hanya satu kali sejak arduino dihidupkan sedangkan *void loop* berisi perintah yang akan

dieksekusi berulang-ulang selama dihidupkan. Untuk tampilan *framework* dari Arduino Uno dapat dilihat pada Gambar 2.13.



Gambar 2.13 Tampilan *Framework* Arduino

2.4. Identifikasi Sistem

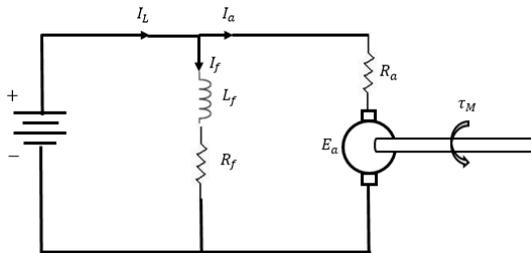
Untuk mendapatkan parameter-parameter dari suatu *plant* maka dilakukan suatu langkah yang dinamakan dengan identifikasi sistem. Dan parameter-parameter *plant* tersebut diperoleh berdasarkan data hasil pengukuran. Identifikasi sistem ini juga dibutuhkan guna memodelkan sistem, karena sebagian besar kontroler berbasis model (*model based controller*) dengan setiap tahap perancangannya dibutuhkan model matematis dari *plant*. Sehingga parameter-parameter yang telah diperoleh tadi dapat digunakan dalam menyusun sebuah model matematis dari suatu *plant*. Pada Tugas Akhir ini identifikasi sistem dilakukan dengan menggunakan Metode Strejc untuk *plant* ICE 2 langkah dan identifikasi fisik untuk *plant* motor DC.

2.4.1. Metode Identifikasi Fisik [11]

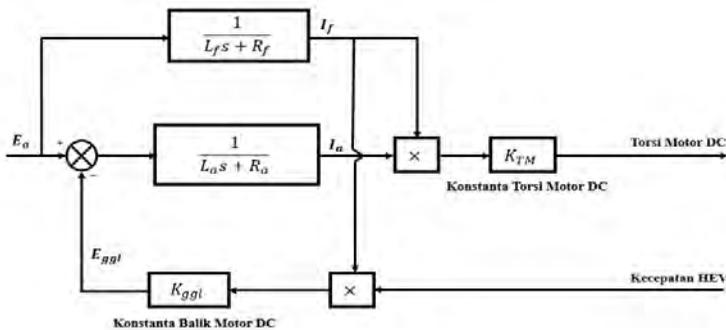
Memperoleh model matematis dari sebuah *plant* dengan identifikasi fisik dapat dilakukan dengan dua cara, yaitu melalui

pendekatan analisis dan eksperimen. Pada Tugas akhir ini digunakan pendekatan analisis di mana sistem yang sesungguhnya diwakili oleh sebuah gabungan elemen-elemen pembentuk yang dianggap ideal. Cara untuk memperoleh model sistem tersebut adalah dengan menggunakan persamaan-persamaan hukum fisika (seperti: Kirchoff, Newton, dll) dan komponen yang terpasang dalam sistem fisik (seperti : resistor, kapasitor, dll). Karena penggunaan dari model matematis yang sesuai sangat penting terhadap keseluruhan analisis dari sistem yang diuji.

Motor DC dengan konfigurasi paralel memiliki karakteristik *nonlinear* terhadap arus jangkar (I_a), arus medan (I_f), dan torsi motor (τ_M) yang dihasilkan. Untuk konfigurasi motor DC secara fisik dapat dilihat pada Gambar 2.14. Dari konfigurasi tersebut nantinya dapat diperoleh persamaan matematika yang digunakan untuk pemodelan motor DC. Selanjutnya dari persamaan matematika yang diperoleh, dapat direpresentasikan kedalam diagram blok pada Gambar 2.15.



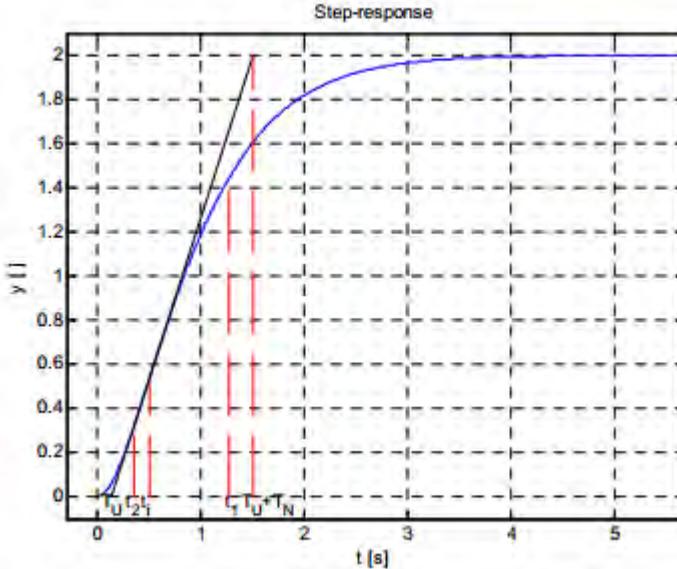
Gambar 2.14 Konfigurasi Fisik Motor DC Paralel



Gambar 2.15 Diagram Blok Motor DC Paralel

2.4.2. Metode Identifikasi Strej [12]

Untuk metode Strejc dilakukan pendekatan pada sistem yang tidak stabil atau *nonlinear*. Dengan tanpa ada waktu tunda, maka untuk perhitungan waktu T_U dan T_N dapat dengan membuat garis singgung yang memotong atau bersinggungan dengan respon sistem. Pada Gambar 2.16 ditunjukkan grafik dari respon sistem yang dihasilkan. Digunakan τ (Persamaan (2.8)) sebagai parameter dari perbandingan waktu.



Gambar 2.16 Respon Step Sistem dengan t_1 , t_2 , t_i , T_U , dan T_N

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

Untuk nilai τ yang diperoleh, perkiraan pendekatan dari pemodelan sistem dapat ditentukan melalui aturan sebagai berikut:

- a. Jika nilai dari $\tau < 0$, maka pada sistem dapat dilakukan pendekatan model orde ke-2 seperti yang ditunjukkan pada Persamaan (2.9).

$$G_{ST_1}(s) = \frac{K}{(\tau_{ST_1}s + 1)(\tau_{ST_2}s + 1)} \quad (2.9)$$

dengan nilai τ_{ST_1} dan τ_{ST_2} dapat ditentukan dengan cara sebagai berikut:

- Memperoleh parameter t_1 dengan mencari waktu pada saat respon sistem mencapai 72% dari y_{SS} (Gambar 2.16), kemudian menentukan jumlah dari τ_{ST_1} dan τ_{ST_2} menurut Persamaan (2.10).

$$\tau_{ST_1} + \tau_{ST_2} = \frac{t_1}{1,2564} \quad (2.10)$$

- Menentukan parameter t_2 dari Persamaan (2.11).

$$t_2 = 0,3574(\tau_{ST_1} + \tau_{ST_2}) \quad (2.11)$$

- Mendapatkan nilai dari $y(t_2)$ dari respon sistem menurut Tabel 2.3 serta menentukan rasio T dari Persamaan (2.12).

$$T = \frac{\tau_{ST_1}}{\tau_{ST_2}} \quad (2.12)$$

- Menentukan nilai dari τ_{ST_1} dan τ_{ST_2} dengan menggunakan Persamaan (2.11) dan Persamaan (2.12). Pada Tabel 2.3 ditampilkan nilai untuk hasil dari rasio T .

Tabel 2.3 Nilai Konstan Rasio 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	-	-

- b. Jika nilai dari $\tau \geq 0$, maka pada sistem dapat dilakukan pendekatan model orde ke- n seperti yang ditunjukkan pada Persamaan (2.13).

$$G_{ST_2}(s) = \frac{K}{(\tau_{ST}s + 1)^n} \quad (2.13)$$

dengan nilai τ_{ST} dapat ditentukan dengan cara berikut:

- Menentukan rasio τ seperti pada Persamaan (2.8).
- Berdasarkan pada nilai τ , dapat diperoleh orde dari pendekatan model yang tepat dan koordinat titik y_i , yang merupakan titik singgung antara respon sistem dengan garis singgung yang tepat berdasar Tabel 2.4.
- Berdasarkan koordinat y_i yang telah diperoleh, selanjutnya mendapatkan t_i yang merupakan waktu saat respon sistem mencapai y_i (ditunjukkan pada Gambar 2.16).
- Menentukan waktu konstan τ_{ST} menurut Persamaan (2.14).

$$\tau_{ST} = \frac{t_i}{n - 1} \quad (2.14)$$

Tabel 2.4 Taksiran dari Orde ke- n dan Dan Nilai Titik Singgung

n	τ	y_i
2	0,014	0,264
3	0,218	0,327
4	0,319	0,359
5	0,410	0,371
6	0,493	0,384
7	0,570	0,394
8	0,642	0,401
9	0,709	0,407
10	0,773	0,413

2.4.3. Root Mean Square Error (RMSE) [13]

RMSE sering digunakan sebagai parameter penting yang menunjukkan akurasi dari suatu analisis. RMSE merupakan komponen yang diperlukan dalam pemodelan statistik (seperti : regresi). RMSE telah banyak digunakan sebagai cara untuk mengevaluasi *error* berdasarkan regresi dan metode statistik perhitungan RMSE ditunjukkan oleh

Persamaan (2.15) dan Persamaan (2.16) menunjukkan formulasi dari nilai *error* dalam bentuk persentase.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n e_i^2}, \quad i = 1, 2, 3, \dots, n \quad (2.15)$$

$$e_i = \frac{A_i - M_i}{A_i} \times 100\%, \quad i = 1, 2, 3, \dots, n \quad (2.16)$$

dengan:

n : Jumlah data

i : Urutan data

e : Nilai *error*

A : Nilai data hasil pengukuran

M : Nilai data hasil pemodelan

2.5. Kontroler PID [13]

Salah satu jenis kontroler yang berdasarkan aksi kontrolnya adalah kontroler *proporsional* ditambah *integral* ditambah *differensial*. Yang merupakan kontroler penggabungan dari ketiga sistem kontrol *proporsional*, *integral*, dan *differensial*. Sama dengan jenis kontroler yang telah dibahas sebelumnya, untuk masukan pada kontroler PID berupa sinyal kesalahan $e(t)$ dan keluaran kontroler adalah sinyal kontrol $u(t)$. Sedangkan hubungan antara masukan kontroler $e(t)$ dan keluaran kontroler $u(t)$ ditulis dalam Persamaan (2.17).

$$u(t) = K_p \left(e(t) + \frac{1}{\tau_i} \int_0^t e(t) dt + \tau_d \frac{de(t)}{dt} \right) \quad (2.17)$$

Jika dalam persamaan *Laplace* Persamaan (2.18) menjadi:

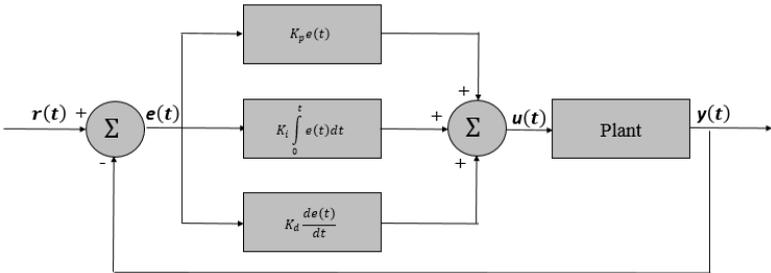
$$U(s) = K_p \left(1 + \frac{1}{\tau_i s} + \tau_d s \right) E(s) \quad (2.18)$$

Dengan K_p adalah penguatan *proporsional* dan τ_i adalah waktu *integral* dan τ_d adalah waktu *differensial*. Dan parameter dari ketiganya dapat

ditentukan. Sehingga untuk fungsi alih kontroler PID ditunjukkan oleh Persamaan (2.19).

$$\frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{\tau_i s} + \tau_d s \right) \quad (2.19)$$

Diagram blok sistem kontrol PID ditunjukkan pada Gambar 2.17.



Gambar 2.17 Diagram Blok PID

Untuk tipe kontroler *proportional* K_p akan memberikan efek mengurangi waktu naik tetapi tidak menghapus kesalahan keadaan tunak. Kontroler *integral* akan memberikan efek menghilangkan *error steady state* tetapi berakibat menurunnya respon *transient*. Sedangkan kontroler *derivative* K_d akan memberikan efek meningkatnya stabilitas sistem dan mengurangi *overshoot*. Efek dari setiap kontroler dalam sistem *closed loop* dapat dilihat pada Tabel 2.5.

Tabel 2.5 Efek Tiap Parameter

Respon Closed Loop	Rise Time	Overshoot	Setting Time	Error Steady State
K_p	Naik	Meningkat	Perubahan kecil	Menurun
K_i	Menurun	Meningkat	Meningkat	Hilang
K_d	Perubahan kecil	Menurun	Menurun	Perubahan kecil

2.6. Particle Swarm Optimization (PSO) [14] [15]

Particle Swarm Optimization, yang disingkat sebagai PSO didasarkan pada perilaku sebuah kawanan serangga, seperti semut, rayap,

lebah atau burung. Algoritma PSO sendiri meniru perilaku sosial organisme ini. Perilaku sosial terdiri dari tindakan individu dan pengaruh dari individu-individu lain dalam suatu kelompok. Kata partikel menunjukkan, misalnya seekor burung dalam kawanan burung. Setiap individu atau partikel berperilaku secara terdistribusi dengan cara menggunakan kecerdasannya (*intelligence*) sendiri dan juga dipengaruhi perilaku kelompok kolektifnya. Dengan demikian, jika satu partikel atau seekor burung menemukan jalan yang tepat atau pendek menuju ke sumber makanan, sisa kelompok yang lain juga akan dapat segera mengikuti jalan tersebut meskipun lokasi mereka jauh dalam kelompok tersebut.

Dalam PSO, kawanan diasumsikan mempunyai ukuran tertentu atau tetap dengan setiap partikel posisi awalna terletak di suatu lokasi yang acak dalam ruang multidimensi. Setiap partikel bergerak dalam ruang atau *space* tertentu dan mengingat posisi terbaik yang pernah dilalui atau ditemukan terhadap sumber makanan atau nilai fungsi objektif. Setiap partikel menyampaikan informasi atau posisi bagusnya kepada partikel yang lain dan menyesuaikan posisi dan kecepatan masing-masing berdasarkan informasi yang diterima mengenai posisi yang bagus tersebut. Sebagai contoh, misalnya perilaku burung-burung dalam kawanan burung. Meskipun setiap burung memiliki ketebatasan dalam hal kecerdasan, biasanya seekor burung akan mengikuti kebiasaan (*rule*) seperti berikut:

- Posisi seekor burung tidak berada terlalu dekat dengan burung yang lain.
- Burung tersebut akan mengarahkan terbangnya ke arah rata-rata ke seluruh kawanan.
- Akan memposisikan diri dengan rata-rata posisi burung yang lain dengan menjaga sehingga jarak antar burung dalam kawanan itu tidak terlalu jauh.

Dengan demikian perilaku kawanan burung akan didasarkan pada kombinasi dari 3 faktor simpel berikut:

- Kohesi : terbang bersama
- Separasi : jarak jangan terlalu dekat
- Penyesuaian : mengikuti arah bersama

Jadi PSO sendiri dikembangkan berdasarkan pada model berikut:

- Ketika seekor burung mendekati target atau makanan (atau bisa minimum atau maksimum suatu fungsi tujuan) secara cepat

mengirim informasi kepada burung-burung yang lain dalam kawanan tertentu.

- Burung yang lain akan mengikuti arah menuju ke makanan tetapi tidak secara langsung.
- Ada komponen yang tergantung pada pikiran setiap burung, yaitu memorinya tentang apa yang sudah dilewati pada waktu sebelumnya.

Pada algoritma PSO ini, pencarian solusi dilakukan oleh suatu populasi yang disebut sebagai kawanan, dengan kawanan yang terdiri dari beberapa individu yang biasa disebut sebagai partikel. Populasi dibangkitkan secara acak dengan batasan nilai terkecil dan terbesar. Setiap partikel merepresentasikan posisi atau solusi dari permasalahan yang dihadapi. Setiap partikel melakukan pencarian solusi yang optimal dengan melintasi ruang pencarian (*search space*). Hal ini dilakukan dengan cara setiap partikel melakukan penyesuaian terhadap posisi terbaik dari partikel tersebut (*local best*) dan penyesuaian terhadap posisi partikel terbaik dari seluruh kawanan (*global best*) selama melintasi ruang pencarian. Jadi, penyebaran pengalaman atau informasi terjadi di dalam partikel itu sendiri dan antara suatu partikel dengan partikel terbaik dari seluruh kawanan selama proses pencarian solusi.

Setelah itu, dilakukan proses pencarian untuk mencari posisi terbaik setiap partikel dalam sejumlah iterasi tertentu sampai diperoleh posisi yang relatif *steady* atau mencapai batas iterasi yang telah ditetapkan. Pada setiap iterasi, setiap solusi yang direpresentasikan oleh posisi partikel, dievaluasi performansinya dengan cara memasukkan solusi tersebut kedalam *fitness function*. Setiap partikel diperlakukan seperti sebuah titik pada suatu dimensi ruang tertentu. Kemudian terdapat dua faktor yang memberikan karakter terhadap status partikel pada ruang pencarian yaitu posisi partikel dan kecepatan partikel. Algoritma PSO orisinal dijelaskan dalam persamaan berikut:

$$v_{i,m}^{(t+1)} = v_{i,m}^{(t)} + c_1 r_1 (pbest_{i,m} - x_{i,m}^{(t)}) + c_2 r_2 (gbest_m - x_{i,m}^{(t)}) \quad (2.20)$$

$$x_{i,m}^{(t+1)} = x_{i,m}^{(t)} + v_{i,m}^{(t+1)} \quad (2.21)$$

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

$$m = 1, 2, \dots, d$$

dengan,

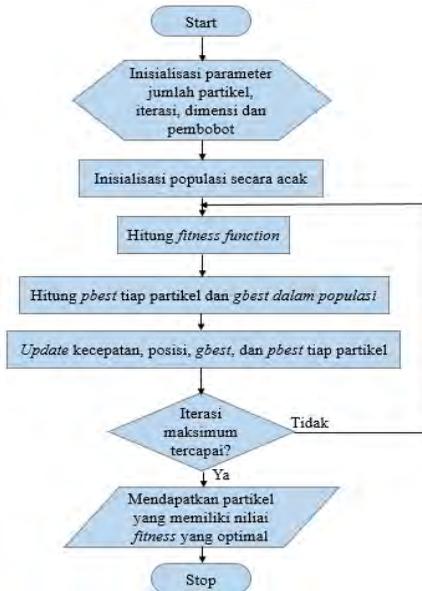
- n : jumlah partikel dalam kelompok
- d : dimensi
- $v_{i,m}^{(t)}$: kecepatan partikel ke- i pada iterasi ke t
- c_1, c_2 : konstanta akselerasi (*learning rate*)
- r_1, r_2 : bilangan random (0 – 1)
- $x_{i,m}^{(t)}$: posisi saat ini dari partikel ke- i pada iterasi ke t
- $pbest_i$: posisi terbaik sebelumnya dari partikel ke- i
- $gbest$: partikel terbaik diantara semua partikel dalam satu kelompok atau populasi

Persamaan dari PSO sendiri memiliki 3 bagian antara lain bagian pertama yang disebut dengan *momentum part*, pada bagian ini partikel tidak dapat berubah secara bebas melainkan mengalami perubahan melalui kecepatan sekarang. Pada bagian kedua yaitu *cognitive part*, merupakan bagian pengalaman partikel, belajar dari pengalaman partikel itu sendiri. Dan pada bagian ketiga adalah *social part*, merupakan kerja sama antar partikel, belajar dari pengalaman partikel-partikel lain.

Kemudian untuk mekanisme dari PSO adalah sebagai berikut:

1. Menentukan parameter awal, yang meliputi jumlah partikel, jumlah iterasi, dimensi dan pembobot.
2. Membangkitkan populasi awal secara acak.
3. Mengevaluasi fungsi *fitness*.
4. Hasil dari *fitness* kita evaluasi dengan cara membandingkan nilai *fitness* dengan *pbest*. Apabila nilai sekarang lebih baik dari *pbest*, maka set nilai *pbest* sama dengan nilai sekarang dan $pbest_{i,m}(t)$ sama dengan nilai posisi sekarang pada dimensi m .
5. Mengidentifikasi partikel-partikel lain yang memiliki *pbest*, jika nilai *pbest* lebih besar dari *gbest*, maka set *gbest* sama dengan *pbest*.
6. Perbarui nilai kecepatan dari posisi suatu partikel.
7. Mengurangi langkah (3) sampai kriteria telah dipenuhi, dengan mempunyai nilai *fitness* yang optimal atau jumlah iterasi telah tercapai.

Sebelumnya pada Persamaan (2.20), apabila kecepatan partikel melebihi $\pm v_{\max}$ maka kecepatan akan menjadi $\pm v_{\max}$. Hal ini untuk membatasi kecepatan partikel yang terlalu tinggi, sehingga dapat menyebabkan osilasi. Untuk diagram alir dari algoritma PSO dapat dilihat pada Gambar 2.18.



Gambar 2.18 Diagram Alir Algoritma PSO

2.5 Modifikasi PSO

Dalam penerapannya, ditemukan bahwa kecepatan partikel dalam PSO *standard* diperbarui terlalu cepat dan nilai minimum fungsi tujuan yang dicari sering terlewat. Oleh karena itu, seiring berjalannya waktu dilakukan modifikasi atau perbaikan terhadap algoritma PSO *standard*. Perbaikan itu berupa penambahan suatu *inertia weighting* (w) untuk mengurangi kecepatan pada formula untuk pembaruan kecepatan. Biasanya w dibuat sedemikian hingga semakin besar iterasi yang dilalui, semakin mengecil kecepatan partikel. Untuk persamaan modifikasi PSO menjadi:

$$v_{i,m}^{(t+1)} = wv_{i,m}^{(t)} + c_1r_1(pbest_{i,m} - x_{i,m}^{(t)}) + c_2r_2(gbest_m - x_{i,m}^{(t)}) \quad (2.22)$$

dengan w adalah *inertia weighting*.

Persamaan (2.22) merupakan modifikasi terhadap Persamaan (2.20). Nilai dari bobot inersia yang tinggi menambah porsi pencarian global, sedangkan nilai yang rendah lebih menekankan pencarian lokal. Untuk tidak terlalu menitikberatkan pada salah satu bagian dan tetap mencari area pencarian yang baru dalam ruang berdimensi tertentu, maka perlu dicari nilai bobot inersia (w) yang secara imbang menjaga pencarian global dan lokal. Untuk mencapai itu dan mempercepat konvergensi, suatu bobot inersia yang mengecil nilainya dengan bertambahnya itersai digunakan Persamaan (2.23).

$$w = w_{max} - \left(\frac{w_{max} - w_{min}}{iter_{max}} \right) iter \quad (2.23)$$

Di mana,

- w_{max} : *initial inertia weight*
- w_{min} : *final inertia weight*
- $iter$: iterasi berjalan
- $iter_{max}$: jumlah iterasi maksimal

BAB 3 PERANCANGAN SISTEM

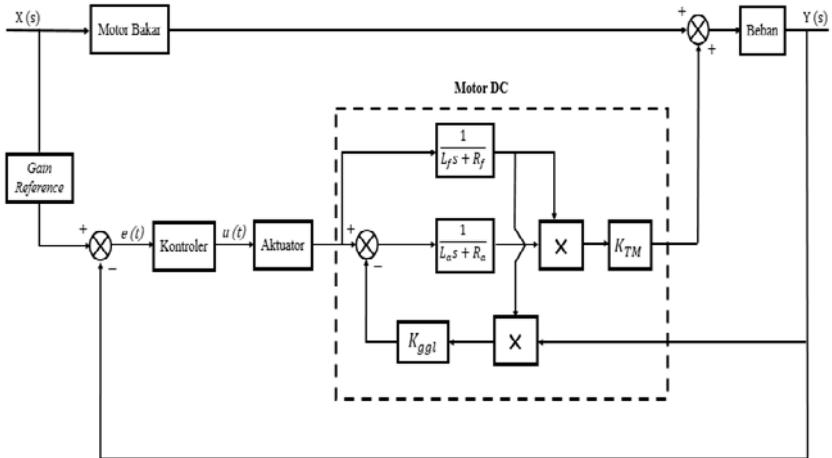
3.1. Gambaran Umum Simulator HEV

Pada Tugas Akhir ini simulator HEV yang digunakan tidak sepenuhnya mengalami perombakan yang signifikan. Hal ini dikarenakan simulator HEV masih bekerja dengan baik dan hanya sedikit mengalami penggantian komponen yang sudah tidak berfungsi lagi. Simulator HEV sendiri menggunakan penggerak yang terdiri dari ICE dan motor DC, dengan ICE sebagai penggerak utama dan motor DC digunakan untuk membantu ICE untuk mencapai kecepatan putar yang diinginkan. Selain itu digunakan rem elektromagnetik untuk memberikan efek pembebanan. Oleh karena itu, dapat diketahui fungsi dari motor DC untuk membantu kinerja ICE dalam mencapai kecepatan putara yang diinginkan. Bentuk fisik simulator HEV yang telah dihasilkan pada penelitian sebelumnya ditunjukkan pada Gambar 3.1.



Gambar 3.1 Bentuk Fisik Simulator HEV

Kontroler yang digunakan dalam pengaturan kecepatan pada simulator HEV ini adalah kontroler tipe *Proportional Integral Derivative* yang dimaksudkan untuk mengatasi permasalahan regulator dengan tujuan menjaga agar nilai dari kecepatan putar tetap pada nilai yang diinginkan dengan melakukan pemilihan parameter kontroler PID berupa K_p , K_i dan K_d yang tepat. Pengujian dari metode ini dengan diberikan gangguan *state* berupa beban rem elektromagnetik. Diagram blok dari sistem pengaturan HEV dapat dilihat pada Gambar 3.2.



Gambar 3.2 Diagram Blok Pengaturan Simulator HEV

3.2. Perancangan Perangkat Keras (*Hardware*)

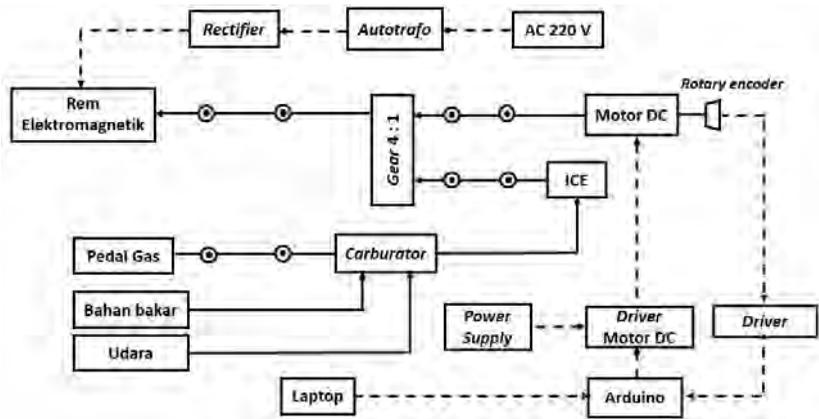
Pada proses perancangan dalam Tugas Akhir ini terlebih dahulu dilakukan pengecekan, penambahan maupun penggantian komponen dari simulator HEV. Perancangan perangkat keras pada simulator HEV ini sendiri secara umum terbagi menjadi dua, yaitu perancangan mekanik dan perancangan elektronik.

3.2.1. Perancangan Mekanik

Konfigurasi sistem untuk simulator HEV sendiri pada dasarnya menggabungkan tenaga penggerak dari ICE dan motor listrik. Kecepatan putar dari ICE jauh lebih tinggi dari pada motor DC, oleh sebab itu digunakan transmisi *gear* untuk membuat kecepatan putar diantara keduanya seimbang. Dengan adanya transmisi *gear* ini, kecepatan putar ICE dapat tereduksi secara proporsional agar sesuai dengan range dari motor DC.

Ditambahkan pula beban berupa rem elektromagnetik arus eddy yang ditempatkan pada poros motor DC. Di samping itu, beban rem elektromagnetik ini akan menghasilkan torsi lawan yang akan mengurangi torsi yang dihasilkan oleh ICE dan motor DC. Sehingga akan menghasilkan penurunan kecepatan putar pada simulator HEV. Secara

keseluruhan untuk konfigurasi perangkat keras dari simulator HEV ditunjukkan oleh Gambar 3.3.



Gambar 3.3 Konfigurasi Perangkat Keras Simulator HEV

3.2.1.1. Internal Combustion Engine (ICE)

Mesin bakar jenis ICE pada simulator HEV berperan sebagai tenaga penggerak utama. Untuk mesin bakar yang digunakan pada simulator HEV ini, dipilih mesin pemotong rumput 2 langkah sebagai penggerak utama. Karena mesin pemotong rumput 2 langkah diasumsikan



Gambar 3.4 Mesin Pemotong Rumput 2 langkah

sebagai ICE pada HEV yang sesungguhnya. transmisi. Bentuk fisik dari mesin pemotong rumput 2 langkah ditunjukkan oleh Gambar 3.4.

Tabel spesifikasi mesin pemotong rumput yang digunakan ditunjukkan pada Tabel 3.1

Tabel 3.1 Spesifikasi ICE

No.	Klasifikasi	Spesifikasi
1	Tipe	2 langkah
2	Bahan bakar	Bensin dan oli (25:1)
3	Kapasitas silinder	33,8 cc
4	Pemasukan karburator	Karburator
5	Sistem starter	<i>Recoil Starter</i>
6	Sistem pengapian	<i>Electric Ignition</i>
7	Sistem kopling	Kering, sentrifugal
8	Kecepatan putar maksimal	8000 rpm
9	Daya keluaran	1,8 HP

3.2.1.2. Motor DC

Pada simulator HEV jenis motor listrik yang digunakan untuk membantu mesin bakar adalah jenis motor DC (*Direct Current*). Motor DC merupakan motor yang menggunakan sumber arus searah dalam kerjanya dan berfungsi untuk mengubah energi listrik DC menjadi energi gerak. Untuk pemilihan motor DC sendiri dipilih motor DC yang memiliki kecepatan putar maksimum yang lebih rendah dari mesin bakar, hal ini dikarenakan motor DC tersebut hanya digunakan sebagai tenaga pembantu untuk mesin bakar. Bentuk fisik motor DC ditunjukkan pada Gambar 3.5.



Gambar 3.5 Bentuk Fisik Motor DC

Kemudian untuk spesifikasi motor DC yang digunakan dalam Tugas Akhir ini ditunjukkan oleh Tabel 3.2.

Tabel 3.2 Spesifikasi Motor DC

No.	Spesifikasi	Keterangan
1	Tipe	Parallel
2	Kecepatan putar maksimum	2000 rpm
3	Tegangan nominal	150 V
4	Daya keluaran	0,5 HP

3.2.1.3. Rem Elektromagnetik

Pada simulator HEV beban yang digunakan berupa rem elektromagnetik. Hal ini dimaksudkan agar dapat menyerupai kondisi sesungguhnya dari HEV dengan memperhatikan beberapa parameter yang ada. Beban berupa rem elektromagnetik tersebut ditempatkan pada poros motor DC. Untuk rem elektromagnetik yang digunakan terdiri dari 12 kumparan yang disusun secara seri dan sebuah piringan besi. Kemudian piringan besi diletakkan diantara kedua kumparan yang dililit koil yang dapat menghasilkan medan magnet jika dialiri arus listrik.

Ketika kecepatan putar tinggi, prinsip kerja dari rem elektromagnetik adalah dengan adanya medan magnet yang dihasilkan akan menimbulkan arus eddy yang menyebabkan torsi yang berlawanan arah dengan arah putar piringan besi. Akibatnya kecepatan putar dari mesin bakar akan menurun dengan adanya efek pembebana tersebut. Bentuk fisik dari rem elektromagnetik ditunjukkan oleh Gambar 3.6.



Gambar 3.6 Bentuk Fisik Rem Elektromagnetik

Digunakan sumber tegangan dari jala-jala PLN 220 V yang dimasukkan pada *auto transformer* dan ditambahkan dengan *full wave rectifier* pada keluaran *auto transformer*. Penggunaan dari *auto transformer* ini adalah untuk mengatur tegangan yang *display* pada rem magnet. Semakin besar tegangan yang masuk maka medan magnet yang dihasilkan semakin besar dan meningkatkan efek pengereman.

3.2.1.4. Transmisi Gear sebagai Penyesuai Kecepatan

Berdasarkan spesifikasi yang dimiliki oleh ICE dan motor DC yang digunakan pada simulator HEV, diketahui bahwa kecepatan putar dari ICE lebih besar dibandingkan dengan kecepatan putar motor DC. Sehingga dibutuhkan *gear* dan *timing belt* untuk mereduksi kecepatan putar ICE agar sesuai dengan rentang kecepatan dari motor DC.

Perbandingan roda gigi diperlukan sebagai penyesuai rentang kecepatan putar antara keduanya. Dengan *range* kerja dari ICE sekitar 0-8000 rpm dan motor DC dengan *range* kerja 0-2000 rpm. Maka dibutuhkan *gear* dan *timing belt* untuk menghubungkan ICE dengan motor DC sebagai media untuk menyatukan poros antara keduanya. Karena *range* dari kecepatan putar keduanya berbeda, maka digunakan perbandingan *gear* dengan rumus:

$$\omega_a R_a = \omega_b R_b \quad (3.1)$$

$$\frac{\omega_a}{\omega_b} = \frac{R_a}{R_b}$$

$$\frac{2000}{8000} = \frac{R_a}{R_b}$$

$$\frac{R_a}{R_b} = \frac{4}{1}$$

Dengan,

ω_a = kecepatan putar motor listrik

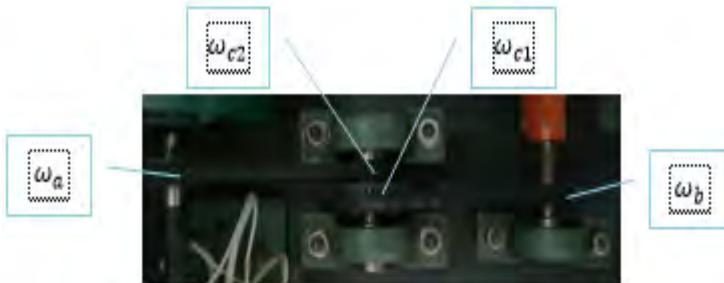
ω_b = kecepatan putar mesin bakar

R_a = jari-jari pada poros motor listrik

R_b = jari-jari pada poros mesin bakar

Karena secara fisik perbandingan *gear* 1:4 memerlukan tempat yang lebih, maka perlu dilakukan penambahan *gear* diantara keduanya

untuk efisiensi tempat . Sehingga konfigurasinya seperti yang ditunjukkan oleh Gambar 3.7, menjadi *gear* pada poros mesin bakar digandeng dengan *gear* 1 poros tengah dengan perbandingan $R_b : R_c = 1 : 2$. *Gear* 1 dan *gear* 2 berada di poros yang sama, yaitu pada poros tengah dengan perbandingan $R_{c1} : R_{c2} = 2 : 1$. Di sisi lain, *gear* 2 poros tengah digandeng dengan *gear* poros motor DC dengan perbandingan $R_{c2} : R_a = 1 : 2$.



Gambar 3.7 Konfigurasi *Gear* pada Simulator HEV

dengan,

$$\begin{aligned} \omega_{c1} &= \text{kecepatan putar poros tengah 1} \\ \omega_{c2} &= \text{kecepatan putar poros tengah 2} \\ R_{c1} &= \text{jari - jari gear pada poros tengah 1} \\ R_{c2} &= \text{jari - jari gear pada poros tengah 2} \\ R_a &= \text{jari - jari gear pada poros motor DC} \\ R_b &= \text{jari - jari gear pada poros mesin bakar} \end{aligned}$$

Sesuai dengan persamaan:

$$\omega_b R_b = \omega_{c1} R_{c1} \quad (3.2)$$

$$\begin{aligned} \omega_{c1} &= \omega_{c2} \\ \omega_a R_a &= \omega_{c2} R_{c2} \end{aligned} \quad (3.3)$$

Maka diperoleh:

$$\frac{\omega_a R_a}{R_{c2}} = \omega_{c2} \quad (3.4)$$

$$\frac{\omega_b R_b}{R_{c1}} = \omega_{c1} \quad (3.5)$$

Karena ω_{c2} memiliki nilai yang sama dengan ω_{c1} maka,

$$\frac{\omega_a R_a}{R_{c2}} = \frac{\omega_b R_b}{R_{c1}} \quad (3.6)$$

$$\frac{\omega_a}{\omega_b} = \frac{R_b}{R_{c1}} \times \frac{R_{c2}}{R_a}$$

$$\frac{\omega_a}{\omega_b} = \frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$$

Dari penggunaan *gear* ini selain berpengaruh pada kecepatan putar kedua mesin juga berpengaruh pada torsi yang dihasilkan kedua mesin. Hal ini dapat terlihat dari Persamaan (3.7).

$$\frac{\tau_a}{\omega_a} = \frac{\tau_b}{\omega_b} \quad (3.7)$$

$$\frac{\tau_a}{\tau_b} = \frac{\omega_a}{\omega_b}$$

$$\frac{\tau_a}{\tau_b} = \frac{8000}{2000}$$

$$\tau_a = 4\tau_b \quad (3.8)$$

dengan,

τ_a = torsi pada poros motor DC

τ_b = torsi pada poros ICE

Dapat diketahui dari Persamaan (3.8) bahwa poros motor DC memiliki torsi sebesar empat kali torsi yang dihasilkan oleh ICE.

3.2.1.5. Pedal gas

Untuk masukan yang digunakan oleh ICE pada simulator HEV yaitu berupa pijakan pedal gas. Kemudian pedal gas ini terhubung langsung dengan *throttle* ICE menggunakan kawat olor gas. Oleh karena itu dibutuhkan sebuah sensor untuk mengetahui seberapa besar bukaan *throttle* pada ICE. Potensiometer dipilih sebagai sensor posisi dalam mengetahui perbandingan perubahan sudut pedal gas dengan bukaan

throttle ICE. Di sisi lain, dengan adanya potensiometer, maka dapat direpresentasikan untuk perubahan nilai resistansi pada potensiometer berbanding lurus dengan bukaan *throttle* dalam presentase.

Hubungan antara nilai resistansi pada potensiometer dengan presentase bukaan *throttle* dapat dilihat pada Persamaan (3.9).

$$f(x) = \frac{100x - 1010}{-8,17} \quad (3.9)$$

dengan,

$$\begin{aligned} f(x) &= \text{presentase bukaan } \textit{throttle} \\ x &= \text{resistansi dari potensiometer} \end{aligned}$$

Berikut ini merupakan bentuk fisik dari pedal gas (ditunjukkan oleh Gambar 3.8) yang digunakan pada simulator HEV dengan dilengkapi potensiometer.



Gambar 3.8 Pedal Gas pada Simulator HEV

3.2.2. Perancangan Elektronik

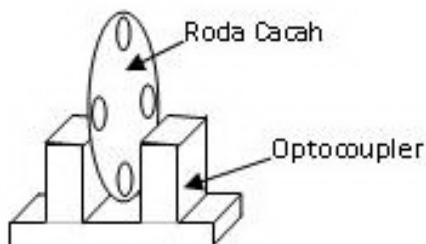
Perancangan perangkat keras yang selanjutnya yaitu meliputi perancangan elektronik. Dalam perancangan elektronik terdapat pembaruan desain rangkaian berupa desain rangkaian elektronik untuk rangkaian pembacaan sensor. Selibhnya untuk rangkaian penyearah gelombang penuh masih menggunakan rangkaian yang terdahulu.

Pada motor DC diperlukan suplai masukan berupa tegangan DC dengan tegangan maksimal 190 V. Masukan tegangan ini diperlukan

untuk mengoperasikan motor DC yang diperoleh dari tegangan jala-jala PLN (AC 220 V 50 Hz). Kemudian disearahkan dengan rangkaian *full wave rectifier* dengan filter kapasitor sehingga tegangan DC yang diperoleh memiliki sedikit *ripple*. Selain itu keluaran dari motor DC yang berupa kecepatan putar diukur melalui pembacaan sensor *rotary encoder*, sedangkan pembacaan masukan diperoleh dari sensor posisi berupa potensiometer yang dipasang pada pedal gas.

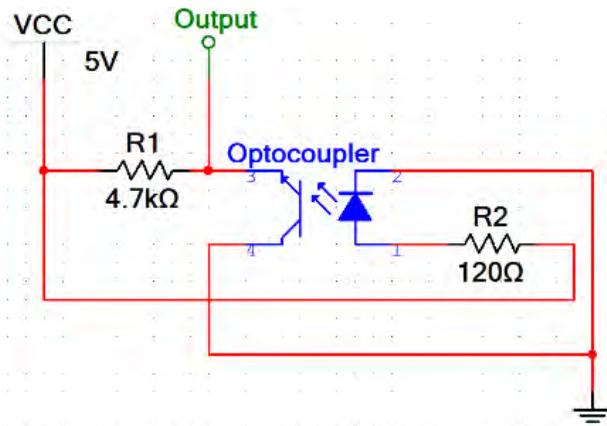
3.2.2.1. Rangkaian Sensor Kecepatan

Pada bagian pembacaan kecepatan putar dari motor DC, digunakan sensor kecepatan yang dipasang pada bagian poros motor DC. Sensor yang digunakan adalah *rotary encoder* yang pada Tugas Akhir sebelumnya menggunakan sensor Tachogenerator. Perlu diketahui bahwa *rotary encoder* sendiri terdiri dari bagian utama yang berupa *optocoupler* dan *encoder disk*. Digunakan *optocoupler* dan *encoder disk* dikarenakan pemasangan *rotary encoder* tidak dimungkinkan ditinjau dari bentuk fisiknya, maka dari itu digunakan *optocoupler* dan tambahan piringan hitam saja sebagai representasi dari kerja *rotary encoder*. Selanjutnya dibuat rangkaian dari *optocoupler* dengann tujuan agar *optocoupler* tersebut dapat bekerja dengan baik dalam pengambilan data kecepatan dari motor DC.



Gambar 3.9 *Optocoupler* dan Piringan Cacah

Untuk pembuatan rangkaian tersebut berdasarkan *datasheet optocoupler* yang diperoleh. Gambaran untuk posisi dari *optocoupler* dan piringan hitam yang digunakan ditunjukkan oleh Gambar 3.9. Sedangkan untuk skema dari rangkaian *optocoupler* yang telah dirancang ditunjukkan oleh Gambar 3.10.

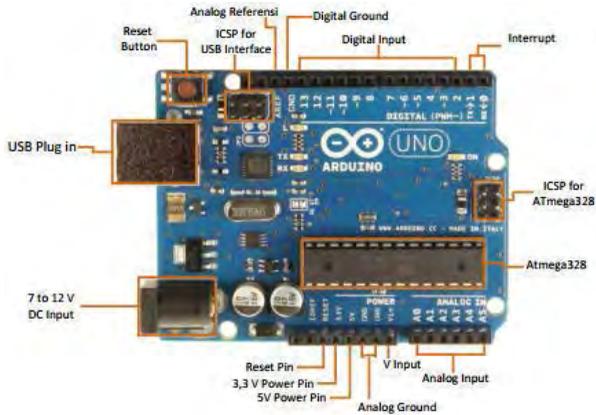


Gambar 3.10 Skema Rangkaian *Optocoupler*

3.2.2.2. *Sistem Minimum Arduino*

Pada simulator HEV sendiri untuk mengatur kecepatan putarnya dibutuhkan serangkaian komponen yang terbentuk dalam sebuah sistem minimum mikrokontroler. Untuk sistem minimum mikrokontroler yang digunakan pada Tugas Akhir ini adalah sistem minimum Arduino Uno R3 berbasis ATmega 328. Kemudian pemrograman pada mikrokontroler Arduino berdasar pada *open source* Arduino. Sistem minimum Arduino Uno R3 sendiri memiliki 6 pin input analog, 14 pin input digital, *port* USB untuk koneksi dengan PC, *port* untuk input 7-12 VDC, *In Circuit Serial Programmer* (ICSP) untuk ATmega 328 dan untuk *interface* USB, tombol reset, dan lain – lain.

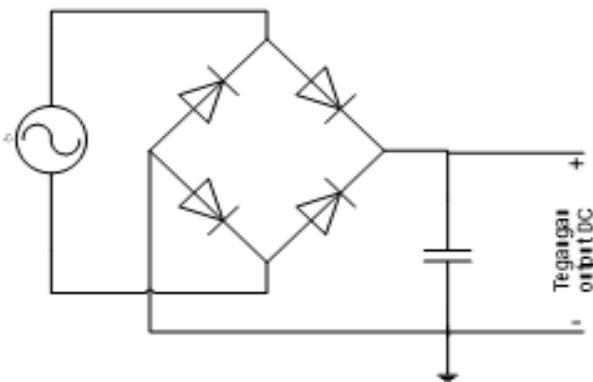
Selain itu, sistem minimum Arduino juga digunakan untuk mengeluarkan sinyal *output* PWM sebagai sinyal kontrol pada motor DC. pengaturan *duty cycle* pada PWM disesuaikan dengan sinyal kontrol yang dibutuhkan oleh motor DC. Bentuk fisik dari sistem minimum Arduino sendiri dapat dilihat pada Gambar 3.11. Di mana pada gambar tersebut telah diberi keterangan untuk masing-masing bagian dari sistem minimum Arduino Uno R3.



Gambar 3.11 Sistem Minimum Arduino Uno R3

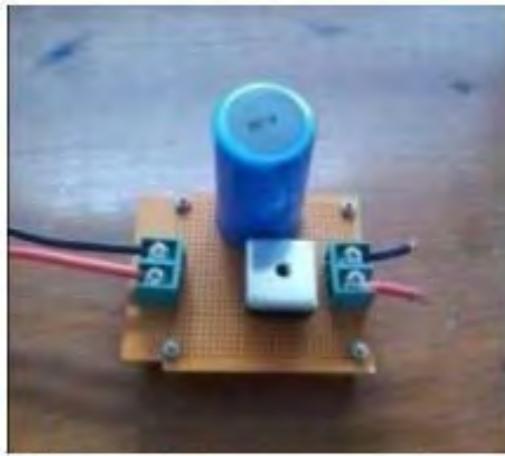
3.2.2.3. Rangkaian Penyearah Gelombang Penuh

Pada simulator HEV terdapat motor DC dan rem elektromagnetik yang merupakan komponen penyusun dalam sistem *hybrid electric vehicle*. Motor DC dan rem elektromagnetik tersebut membutuhkan sumber tegangan berupa tegangan DC. Dan tegangan tersebut diperoleh dari tegangan jala-jala PLN yang dilewatkan ke rangkaian penyearah gelombang penuh. Dengan adanya rangkaian ini maka tegangan AC yang lewat akan berubah menjadi tegangan DC.



Gambar 3.12 Skema Rangkaian Penyearah Gelombang Penuh

Untuk rangkaian penyearah gelombang penuh ini terdiri dari empat buah dioda yang disusun menjadi rangkaian jembatan dan satu buah kapasitor. Kemudian dua dioda aktif untuk menghantarkan tegangan AC sedangkan dua dioda lainnya sebagai isolator pada siklus yang sama. Supaya tegangan yang dihasilkan menyerupai tegangan DC maka digunakan kapasitor untuk mengurangi *ripple* yang dihasilkan pada proses gelombang AC. Skema dari rangkaian penyearah gelombang sebelumnya ditunjukkan oleh Gambar 3.12. Sedangkan bentuk fisik dari rangkaian penyearah gelombang penuh dengan filter kapasitor ditunjukkan oleh Gambar 3.13.



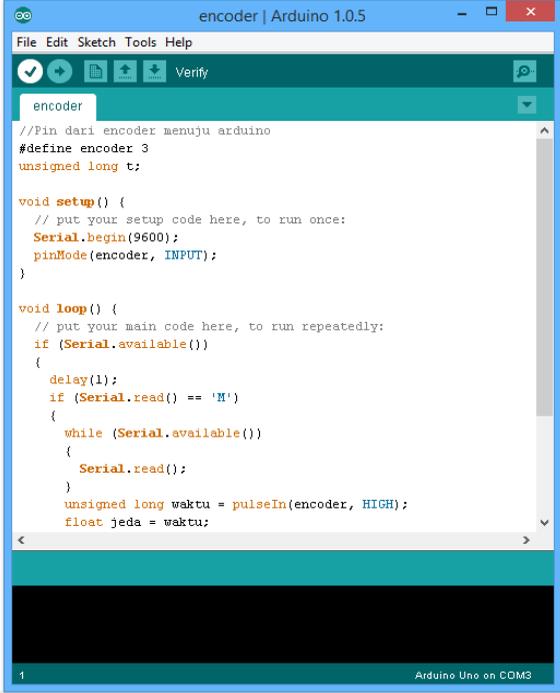
Gambar 3.13 Bentuk Fisik Rangkaian Penyearah Gelombang Penuh

3.3. Perancangan Perangkat Lunak (*Software*)

Pada Tugas Akhir ini perancangan perangkat lunak digunakan pada saat proses pengambilan data, proses identifikasi *plant* dan proses pengolahan data serta pemodelan sistem. Selain itu, dari perancangan perangkat lunak ini juga untuk membantu dalam proses implementasi hasil desain kontroler. Untuk perancangan dari perangkat lunak ini meliputi:

3.3.1. Software Arduino

Pada saat pengambilan data kecepatan motor DC digunakan Arduino sebagai jembatan penghubung antara komputer dengan sensor kecepatan. Oleh sebab itu, selain *project board* Arduino juga dibutuhkan *software* Arduino untuk membaca dan mengirim data dari *rotary encoder* yang diteruskan ke komputer. Program dituliskan pada *software* Arduino, melalui serial USB yang dihubungkan pada komputer. Tampilan dari *software* Arduino ditunjukkan oleh Gambar 3.14.



```
encoder | Arduino 1.0.5
File Edit Sketch Tools Help
[Icons] Verify
encoder
//Pin dari encoder menuju arduino
#define encoder 3
unsigned long t;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  pinMode(encoder, INPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  if (Serial.available())
  {
    delay(1);
    if (Serial.read() == 'H')
    {
      while (Serial.available())
      {
        Serial.read();
      }
      unsigned long waktu = pulseIn(encoder, HIGH);
      float jeda = waktu;
    }
  }
}
```

1 Arduino Uno on COM3

Gambar 3.14 Tampilan Program Arduino

Kemudian pada program Arduino yang dibuat dilakukan linearisasi untuk kecepatan motor DC hasil pembacaan dari *rotary encoder* dengan pembacaan *tachometer digital* yang dapat dilihat pada Persamaan (3.10). Perlunya dilakukan linearisasi yaitu agar data

kecepatan yang akan diolah Labview hanya memiliki sedikit perbedaan dengan data kecepatan asli.

$$f(x) = 229.9536927 + 0,8205613x \quad (3.10)$$

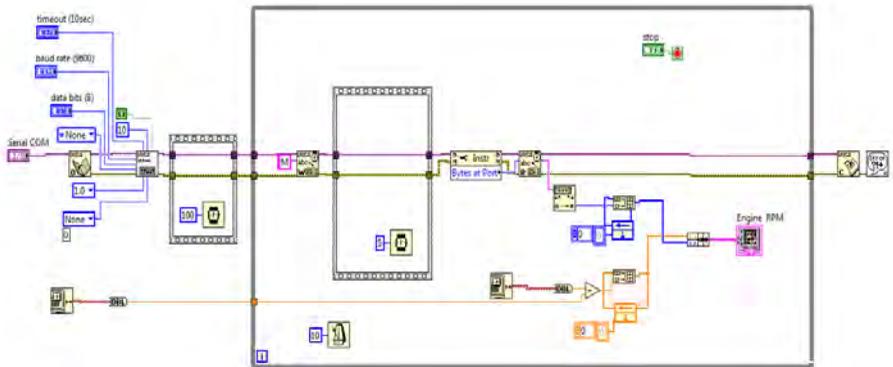
Dengan,

$f(x)$ = kecepatan pada keluaran Arduino

x = kecepatan pada keluaran *rotary encoder*

3.3.2. Software LabVIEW

Saat pengambilan data kecepatan pada mesin bakar, digunakan *software* LabVIEW untuk menampilkan data kecepatan mesin bakar dalam bentuk grafik. LabVIEW sendiri merupakan sebuah *software* yang digunakan untuk mengotomatisasi pengujian dan pengumpulan data. Selain itu dengan menggunakan LabVIEW, data grafik kecepatan juga dapat disimpan. Pada umumnya bahasa pemrograman berbentuk *text based* dengan mengeksekusi instruksi sesuai dengan urutan yang ditulis, sedangkan LabVIEW menggunakan metode *data flow programming* dengan alur data melalui berbagai ikon akan menentukan urutan eksekusi dari setiap instruksi.



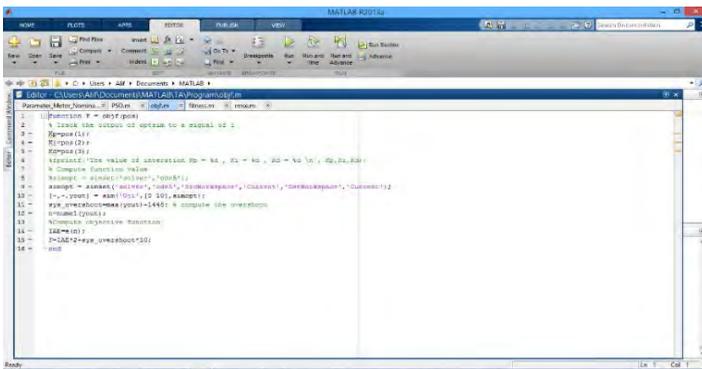
Gambar 3.15 Tampilan Diagram Blok LabVIEW

Untuk tampilan diagram blok dari *software* LabVIEW itu sendiri seperti yang ditunjukkan oleh Gambar 3.15.

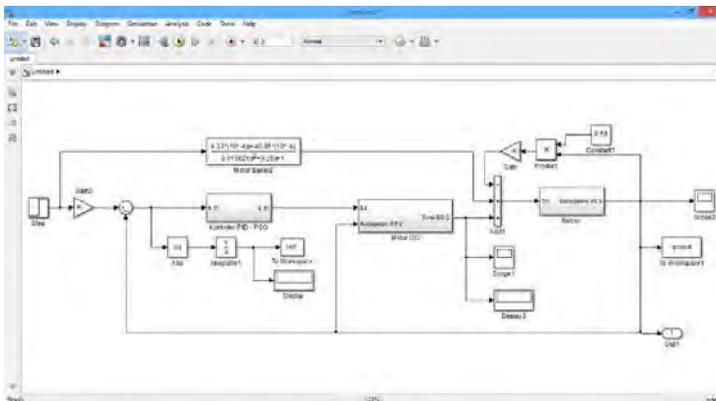
3.3.3. Software MATLAB

Software Matlab pada Tugas Akhir ini digunakan sebagai media dalam merancang simulasi sistem menggunakan menu Editor dan *simulink*. Selain itu juga dapat digunakan sebagai *interface* pada saat proses implementasi. Pada menu Editor digunakan untuk merancang *program file* PSO untuk *tunning* parameter kontroler sedangkan untuk diagram blok dari simulator HEV yang digunakan untuk desain kontroler PID dirancang pada *simulink*.

Berikut merupakan tampilan untuk menu Editor dan Simulink pada Matlab yang ditunjukkan oleh Gambar 3.16 dan Gambar 3.17.



Gambar 3.16 Menu Editor pada MATLAB



Gambar 3.17 Tampilan Simulink

3.4. Pengujian Sensor

3.4.1. Pengujian Sensor *Rotary Encoder*

Sensor yang digunakan untuk menghitung kecepatan putar dari motor DC pada Tugas Akhir sebelumnya yaitu tacho generator. Namun dalam Tugas Akhir kali ini dipilih *rotary encoder* dalam menghitung kecepatan putar dari motor DC. *Rotary encoder* sendiri tersusun atas bagian utama yang bernama *optocoupler* dan sebuah piringan tipis. Dan *optocoupler* inilah yang dipakai dalam memperoleh pembacaan kecepatan putar dari motor DC. Hal tersebut dikarenakan pemasangan dari *rotary encoder* yang tidak memungkinkan untuk dipasang pada *shaft* motor DC, sehingga hanya digunakan bagian *optocoupler* dan piringan tipis berwarna hitam seperti yang ditunjukkan oleh Gambar 3.18.



Gambar 3.18 *Optocoupler* dan Piringan Hitam

Untuk pengujian pada pembacaan kecepatan putar menggunakan *optocoupler* ini yaitu dengan membandingkan kecepatan putar hasil pembacaan tachometer digital. Berdasarkan hasil pengujian diperoleh data pembacaan kecepatan putar dari *optocoupler* dan data dari tachometer digital. Data hasil pembacaan kecepatan putar dari keduanya ditunjukkan pada Tabel 3.3.

Tabel 3.3 Data Hasil Pembacaan Kecepatan Putar dari Tachometer Digital dan *Rotary Encoder*

No.	V _{in} (Volt)	Kecepatan Putar Hasil Pembacaan Tachometer (rpm)	Kecepatan Putar Hasil Pembacaan <i>Rotary Encoder</i> (rpm)
1	25	595	570
2	50	981	1070
3	80	1136	1320
4	90	1179	1400
5	100	1226	1550
6	120	1339	1602
7	150	1539	1881
8	160	1606	1911
9	170	1676	2164
10	180	1743	2209
11	190	1806	2314

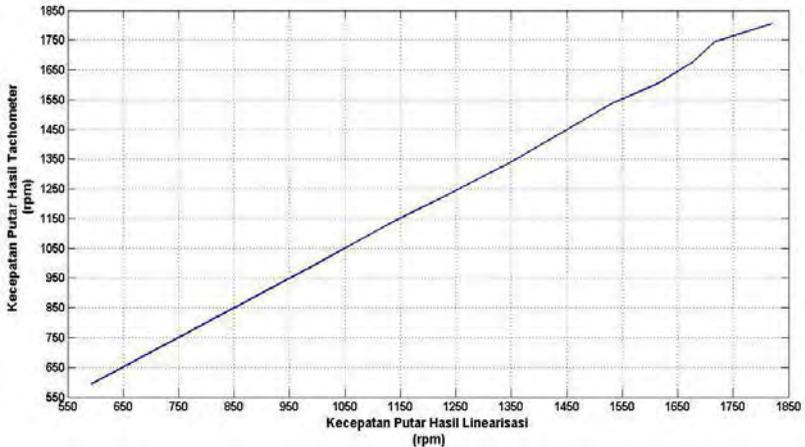
Dari hasil pengujian tersebut dapat diketahui perbandingan perubahan antara kecepatan putar motor DC dengan *rotary encoder* dengan mencari linearisasi menggunakan regresi *linear*. Kemudian untuk hasil linearisasi menggunakan regresi *linear* dapat dilihat pada Persamaan 3.10 sebelumnya. Berikut merupakan perbandingan data kecepatan putar hasil linearisasi dengan kecepatan asli pengukuran menggunakan tachometer digital yang ditampilkan pada Tabel 3.4.

Tabel 3.4 Data Hasil Linearisasi dan Hasil Pengukuran

Kecepatan Putar Hasil Linearisasi (rpm)	Kecepatan Putar Hasil Pengukuran Tachometer Digital (rpm)
592,3796066	595
981,3256628	981
1133,129503	1136
1179,901497	1179
1229,955737	1226
1348,937125	1339
1531,922295	1539
1614,798986	1606
1677,982207	1676
1716,548588	1743

Kecepatan Putar Hasil Linearisasi (rpm)	Kecepatan Putar Hasil Pengukuran Tachometer Digital (rpm)
1819,11875	1806

Selain itu, untuk grafik antara kecepatan putar hasil linearisasi dengan kecepatan putar hasil pengukuran menggunakan tachometer digital ditunjukkan oleh Gambar 3.19.



Gambar 3.19 Grafik Kecepatan Putar Hasil Linearisasi dengan Hasil Pengukuran Tachometer Digital

3.4.2. Pengujian Sensor Potensiometer

Sensor yang digunakan untuk merepresentasikan bukaan *throttle* pada simulator HEV adalah sensor potensiometer. Sensor tersebut dipasang untuk mengetahui derajat bukaan *throttle*. Disisi lain, dengan adanya potensiometer dapat dibuat suatu hubungan antara hambatan pada potensiometer yaitu berbanding lurus dengan presentase bukaan *throttle* ICE. Untuk hubungan antara hambatan dari potensiometer dengan presentase bukaan *throttle* telah dijabarkan sebelumnya pada Persamaan (3.9).

3.5. Identifikasi dan Pemodelan Sistem

Dalam proses perancangan kontroler tentu dibutuhkan suatu model matematika suatu *plant*, sehingga dibutuhkan suatu langkah yang

dinamakan dengan identifikasi. Pada Tugas Akhir ini identifikasi dilakukan untuk mendapatkan parameter yang ada pada simulator HEV. Dari hasil identifikasi tersebut, nantinya diperoleh parameter-parameter yang digunakan dalam membuat model matematika *plant* simulator HEV. Karena model matematika yang diperoleh nantinya dapat digunakan sebagai representasi dari *plant* yang sesungguhnya. Selain itu, model matematika tersebut dapat digunakan dalam perancangan kontroler.

Identifikasi yang dilakukan pada Tugas Akhir ini terdiri atas 3 bagian, yaitu identifikasi parameter pada motor DC, identifikasi parameter beban, dan identifikasi statis *Internal Combustion Engine* (ICE).

3.5.1. Identifikasi Parameter Motor DC

Pada motor DC sendiri parameter yang digunakan untuk pemodelan diperoleh dari besaran-besaran yang ada pada motor DC. Diantaranya berupa resistansi dan induktansi kumparan jangkar (R_a, L_a), resistansi dan induktansi kumparan medan (R_f, L_f), konstanta ballik motor DC (K_{ggl}), konstanta torsi motor DC (K_{TM}), arus jangkar (I_a), dan arus medan (I_f).

3.5.1.1. Resistansi dan Induktansi

Untuk pengukuran parameter yang berupa nilai resistansi dan induktansi dari kumparan motor, baik kumparan jangkar maupun kumparan medan (R_a, L_a, R_f, L_f) dilakukan dengan menggunakan alat yang bernama LCR meter. Dan diperoleh hasil pengukuran yang ditampilkan pada Tabel 3.5.

Tabel 3.5 Data Hasil Pengukuran untuk R_a, L_a, R_f, L_f

Parameter	Nilai
R_a	19,748 Ω
L_a	0,09116 H
R_f	2377,12 Ω
L_f	23,34 H

3.5.1.2. Konstanta Balik Motor DC

Pada pemodelan motor DC, parameter yang dicari selanjutnya adalah konstanta balik motor DC. Untuk mendapatkan nilai dari konstanta balik motor DC yaitu dengan menggerakkan motor DC menggunakan ICE. Dan motor DC sendiri diberi tegangan masukan pada kumparan

medannya saja. Dalam mendapatkan konstanta balik motor DC ini, dilakukan percobaan pada 5 kondisi tegangan masukan ke kumparan medan. Mulai dari 150 V, 160 V, 170 V, 180 V, dan 190 V.

Selanjutnya perubahan tegangan masukan ini dilakukan pada 3 kondisi kecepatan yang berbeda, yaitu 1000 rpm, 910 rpm, dan 830 rpm. Setelah percobaan dilakukan dan semua data telah diperoleh, langkah selanjutnya yaitu memasukkan data tersebut kedalam Persamaan (3.11) untuk memperoleh nilai dari konstanta balik motor DC. selanjutnya data yang telah diperoleh dari hasil percobaan dapat dilihat pada Tabel 3.6, dari data tersebut dapat diperoleh nilai rata-rata konstanta balik motor DC dengan menggunakan Persamaan (3.11). Besar nilai rata-rata dari konstanta balik motor DC adalah sebesar 0.71546.

$$K_{ggl} = \frac{E_a}{I_f \omega} \quad (3.11)$$

Tabel 3.6 Data Kontstanta Balik Motor DC Hasil Percobaan

No.	V_{in} (V)	Kecepatan (rpm)	I_f (A)	E_a (V)	K_{ggl}
1	150	1000	0,0729	68	0,9327
2	160	1000	0,0795	70	0,8805
3	170	1000	0,0835	72	0,8622
4	180	1000	0,0945	73	0,7724
5	190	1000	0,1006	75	0,7455
6	150	910	0,0778	54	0,6512
7	160	910	0,0816	59	0,6550
8	170	910	0,0866	59	0,6812
9	180	910	0,0916	60	0,7230
10	190	910	0,0952	62	0,6940
11	150	830	0,0767	49	0,6388
12	160	830	0,0822	52	0,6236
13	170	830	0,0864	56	0,6481
14	180	830	0,0924	57	0,6168
15	190	830	0,0972	59	0,6069
Rata – rata K_{ggl}					0,7155

3.5.1.3. Konstanta Torsi Motor DC

Parameter motor DC yang lainnya adalah konstanta motor DC. Konstanta motor DC diperoleh dengan melakukan beberapa percobaan. Untuk percobaan tersebut dilakukan pada kondisi tegangan masukan

maksimum dan tegangan masukan nominal. Percobaan pada kondisi tegangan masukan maksimum, motor DC dijalankan dengan mendapatkan tegangan masukan sebesar 190 V dan beban rem elektromagnetik sebesar 80 V. Untuk percobaan dengan kondisi tegangan masukan nominal langkahnya sama dengan percobaan dengan tegangan masukan maksimum, hanya saja nilai pada tegangan masukan nominal sebesar 150 V dan beban rem elektromagnetik sebesar 40 V.

Kemudian konstanta torsi motor DC diperoleh berdasarkan Persamaan (3.12)

$$K_{TM} = \frac{\Delta\tau_{MDC}}{(I_{amax} I_{fmax}) - (I_{anom} I_{fnom})} \quad (3.12)$$

Dengan,

K_{TM}	= konstanta torsi motor DC
τ_{MDC}	= torsi motor DC
I_{amax}	= arus jangkar pada kondisi maksimal
I_{fmax}	= arus medan pada kondisi maksimal
I_{anom}	= arus jangkar pada kondisi nominal
I_{fnom}	= arus medan pada kondisi nominal

Untuk data nilai dari parameter I_{amax} , I_{fmax} , I_{anom} , dan I_{fnom} dapat dilihat pada Tabel 3.7.

Tabel 3.7 Data Nilai Parameter I_{amax} , I_{fmax} , I_{anom} , dan I_{fnom}

Parameter	Nilai (A)
I_{amax}	6,42
I_{fmax}	4,28
I_{anom}	0,0763
I_{fnom}	0,0631

Sebelum mendapatkan nilai dari konstanta torsi motor DC pada Persamaan (3.12), untuk mengetahui nilai dari torsi motor DC dapat dilakukan dengan mencarinya melalui Persamaan (3.17).

$$P_{in} = P_{rugi-rugi} + \tau_{MDC}\omega \quad (3.13)$$

Kemudian $P_{in} = V_{in}(I_a + I_f)$, sehingga persamaannya menjadi,

$$V_{in}(I_a + I_f) = P_{rugi-rugi} + \tau_{MDC}\omega \quad (3.14)$$

Dengan,

P_{in}	= daya masukan motor DC
V_{in}	= tegangan masukan motor DC
I_a	= arus jangkar pada motor DC
I_f	= arus medan pada motor DC
$P_{rugi-rugi}$	= rugi-rugi daya motor DC
τ_{MDC}	= torsi motor DC
ω	= kecepatan putar motor DC

Selanjutnya untuk mendapatkan nilai dari parameter pada Persamaan (3.14), maka dilakukan percobaan dengan dua kondisi. Percobaan pertama dengan kondisi motor DC diberi beban berupa rem elektromagnetik dengan tegangan masukan pada rem sebesar 80 V (maksimal) dan motor DC dijalankan hingga kecepatannya 1450 rpm. Kemudian pada percobaan kedua motor DC diberikan beban berupa rem elektromagnetik dengan tegangan masukan rem sebesar 40 V (nominal) dan motor DC dijalankan dengan kecepatan yang sama, sebesar 1450 rpm. Karena percobaan yang dilakukan memiliki kecepatan yang sama, maka nilai dari $P_{rugi-rugi}$ relatif sama. Sehingga persamaannya menjadi:

Percobaan 1

$$V_{in_1}(I_{a_1} + I_{f_1}) = P_{rugi-rugi} + \tau_{MDC_1}\omega \quad (3.15)$$

Percobaan 2

$$V_{in_2}(I_{a_2} + I_{f_2}) = P_{rugi-rugi} + \tau_{MDC_2}\omega \quad (3.16)$$

Setelah diperoleh dua persamaan, maka eliminasi Persamaan (3.15) dan (3.16). Dan diperoleh Persamaan (3.17).

$$\omega(\tau_{MDC_1} - \tau_{MDC_2}) = V_{in_1}(I_{a_1} + I_{f_1}) - V_{in_2}(I_{a_2} + I_{f_2})$$

$$\Delta\tau_{MDC} = \frac{V_{in_1}(I_{a_1} + I_{f_1}) - V_{in_2}(I_{a_2} + I_{f_2})}{\omega} \quad (3.17)$$

Dengan,

- V_{in_1} = tegangan masukan motor DC saat beban 80 V
- V_{in_2} = tegangan masukan motor DC saat beban 40 V
- I_{a_1} = arus jangkar motor DC saat beban 80 V
- I_{a_2} = arus jangkar motor DC saat beban 40 V
- I_{f_1} = arus medan motor DC saat beban 80 V
- I_{f_2} = arus medan motor DC saat beban 40 V

Untuk nilai dari parameter pada Persamaan (3.17) dapat dilihat pada Tabel 3.8.

Tabel 3.8 Data Nilai Parameter V_{in_1} , V_{in_2} , I_{a_1} , I_{a_2} , I_{f_1} , dan I_{f_2}

Parameter	Nilai
V_{in_1}	63,5 V
V_{in_2}	49 V
I_{a_1}	6,42 A
I_{a_2}	4,28 A
I_{f_1}	0,0763 A
I_{f_2}	0,0631 A

Dari data nilai parameter pada Tabel 3.8 dapat kita gunakan dalam menyelesaikan Persamaan (3.17) untuk memperoleh nilai $\Delta\tau_{MDC}$. Dengan mensubstitusikan nilai dari parameter yang diperoleh kedalam Persamaan (3.17), diperoleh:

$$\Delta\tau_{MDC} = \frac{V_{in_1}(I_{a_1} + I_{f_1}) - V_{in_2}(I_{a_2} + I_{f_2})}{\omega}$$

$$\Delta\tau_{MDC} = \frac{63,5(6,42 + 0,0763) - 49(4,28 + 0,0631)}{1450}$$

$$\Delta\tau_{MDC} = 0,138$$

Setelah memperoleh nilai $\Delta\tau_{MDC}$, maka selanjutnya dapat digunakan untuk memperoleh nilai dari konstanta torsi motor (K_{TM}) dari Persamaan (3.12). Sehingga nilai dari K_{TM} diperoleh sebesar:

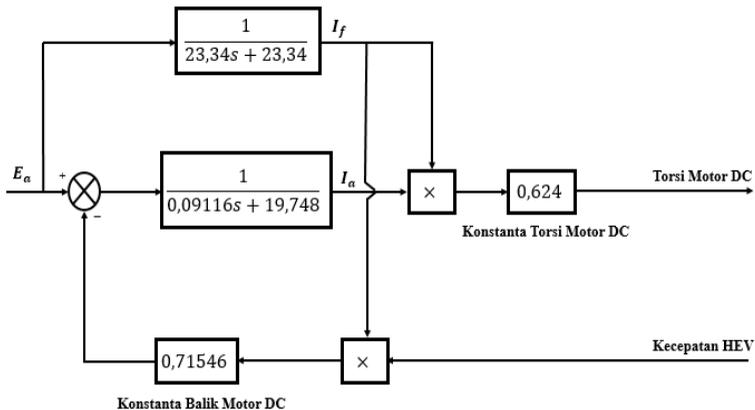
$$K_{TM} = \frac{\Delta\tau_{MDC}}{(I_{a_{max}} I_{f_{max}}) - (I_{a_{nom}} I_{f_{nom}})}$$

$$K_{TM} = \frac{0,138}{(6,42 \cdot 0,0763) - (4,28 \cdot 0,0631)}$$

$$K_{TM} = 0,624$$

3.5.2. Pemodelan Motor DC

Setelah diperoleh parameter untuk motor DC, maka selanjutnya parameter tersebut dapat digunakan untuk menyusun model dari motor DC. Pada Tugas Akhir ini digunakan motor DC dengan konfigurasi paralel yang memiliki karakteristik *nonlinear* terhadap arus jangkar, arus medan, dan torsi motor. Kemudian diperoleh model untuk motor DC seperti yang ditunjukkan oleh Gambar 3.20.



Gambar 3.20 Diagram Blok Motor DC

3.5.3. Identifikasi Parameter Beban

Pada simulator HEV terdapat beban yang terbagai menjadi beban berupa rem elektromagnetik dan beban yang merupakan elemen penghubung antara motor DC dengan ICE. Parameter beban yang digunakan terdiri dari konstanta *dampner disk* atau piringan rem (K_d), momen inersia beban (J_m), koefisien gesek viskos beban (B_m), serta beban pengereman (B_l). Untuk nilai dari parameter konstanta *dampner disk* atau piringan rem (K_d) dan koefisien gesek viskos beban (B_m) dapat diperoleh melalui Persamaan (3.18).

$$\omega B_m - \omega I_l K_d = K_{TM} I_a I_f \quad (3.18)$$

Dengan,

ω	= kecepatan putar motor DC
B_m	= koefisien gesek viskos beban
I_l	= arus yang mengalir pada rem elektromagnetik
K_d	= konstanta <i>dampner disk</i> atau piringan rem
K_{TM}	= konstanta torsi motor DC
I_a	= arus jangkar
I_f	= arus medan

Kemudian untuk nilai parameter beban pengereman (B_l) dapat diperoleh melalui Persamaan (3.19).

$$B_l = -K_d I_l \quad (3.19)$$

Percobaan yang dilakukan untuk memperoleh parameter beban sama seperti saat percobaan untuk memperoleh torsi motor DC, hanya saja terdapat parameter beban pengereman. Kemudian percobaan dilakukan sebanyak dua kali dengan dua kondisi, pertama dengan beban berupa rem elektromagnetik mendapat tegangan masukan sebesar 80 V (maksimal) dan yang kedua dengan tegangan masukan sebesar 40 V (nominal).

Sehingga dari pemberian tegangan masukan pada rem elektromagnetik ini, nantinya akan dapat diukur arus pengereman pada dua kondisi percobaan tersebut. Untuk data hasil pengukuran arus pengereman yang mengalir pada rem elektromagnetik dapat ditunjukkan oleh Tabel 3.9.

Tabel 3.9 Data Hasil Pengukuran Rem Elektromagnetik

Tegangan Masukan Rem (V)	I_l (A)	ω (rpm)
80 V	1,14	1450
40 V	0,56	1450

Nilai parameter yang telah diperoleh kemudian disubstitusikan kedalam Persamaan (3.19), sehingga didapat dua persamaan untuk dua kondisi percobaan.

Percobaan dengan kondisi tegangan masukan rem 80 V ($I_l = 1,14$ A)

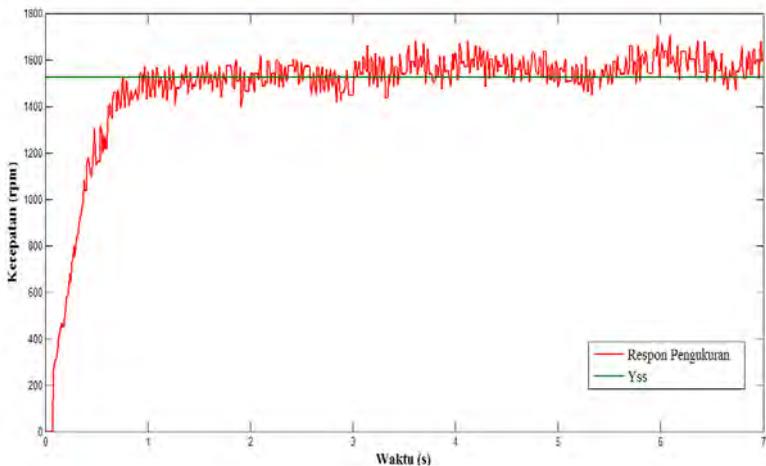
$$1450B_m - 1450 \cdot 1,14K_d = 0,624 \cdot 6,42 \cdot 0,0763$$

$$1450B_m - 1653K_d = 0,3057 \quad (3.20)$$

Percobaan dengan kondisi tegangan masukan rem 40 V ($I_l = 0,56$ A)

$$1450B_m - 1450 \cdot 0,56K_d = 0,624 \cdot 4,28 \cdot 0,0631$$

$$1450B_m - 812K_d = 0,1681 \quad (3.21)$$



Gambar 3.21 Respon Keluaran Motor DC saat Diberikan Masukan Sinyal Step

Selanjutnya eliminasi Persamaan (3.20) dan (3.21), sehingga diperoleh nilai K_d sebesar $1,636 \cdot 10^{-4}$ dan B_m sebesar $2,462 \cdot 10^{-5}$. Di samping itu, untuk B_l diperoleh nilai sebesar $9,162 \cdot 10^{-5}$. Kemudian parameter momen inersia beban (J_m) diperoleh dari respon sistem motor DC dengan mengambil data respon kecepatan motor DC saat diberikan masukan berupa sinyal *step* yang ditunjukkan oleh Gambar 3.21. Sinyal *step* sendiri merupakan representasi dari tegangan masukan motor DC dengan perubahan dari 0 V hingga 150 V dalam waktu yang singkat.

Dari respon motor DC yang dihasilkan dapat diketahui fungsi alihnya berdasarkan Persamaan (3.23).

$$\frac{1}{J_m s + B_m} = \frac{1/B_m}{J_m/B_m s + 1} \quad (3.22)$$

$$\frac{1}{J_m s + B_m} = \frac{1}{\tau s + 1} \quad (3.23)$$

maka diperoleh,

$$\tau = \frac{J_m}{B_m} \quad (3.24)$$

$$J_m = \tau B_m \quad (3.25)$$

Dari respon keluaran motor DC diperoleh untuk nilai τ sebesar 0,5 detik. Selain itu nilai dari B_m juga telah diketahui sebesar $2,462 \cdot 10^{-5}$, sehingga nilai J_m dapat diperoleh dengan memasukan nilai τ dan B_m kedalam Persamaan (3.25).

$$J_m = 0,5 \cdot 2,462 \cdot 10^{-5}$$

$$J_m = 1,231 \cdot 10^{-5}$$

Dan diperoleh model matematika untuk beban adalah sebagai berikut:

$$\frac{1}{J_m s + B_m + B_l} = \frac{1}{1,231 \cdot 10^{-5} s + 11,623 \cdot 10^{-5}} \quad (3.26)$$

3.5.4. Identifikasi ICE

Pada ICE dilakukan jenis identifikasi statis dengan menggunakan metode Strejc. Untuk metode Strejc sendiri menggunakan grafik respon keluaan dari ICE saat diberikan masukan berupa sinyal *step*. Sehingga

nanti akan diperoleh sebuah model matematika dari ICE melalui pengolahan nilai yang terdapat pada grafik respon keluaran tersebut.

3.5.5. Pemodelan ICE

Pada pemodelan ICE sendiri terbagi menjadi 3 jenis pemodelan, yaitu pemodelan saat pembebanan minimal, pembebanan nominal, dan pembebanan maksimal.

3.5.5.1. Pembebanan Minimal

Pada pemodelan ICE dengan pembebanan minimal dilakukan dengan kondisi beban berupa rem elektromagnetik tanpa diberikan tegangan masukan. Oleh karena itu, pengukuran kecepatan putar ICE dilakukan dengan anggapan beban minimal berupa motor DC karena telah terhubung melalui transmisi *gear* yang menghubungkan keduanya. Untuk pengujian dan validasi respon *open loop* ICE pada pembebanan minimal ditunjukkan oleh Gambar 3.22. Di mana identifikasi pada ICE ini menggunakan metode Strejc dengan perhitungan sebagai berikut:

$$Y_{ss} = 1523$$

$$X_{ss} = 41,2$$

$$K = \frac{Y_{ss}}{X_{ss}} = \frac{1523}{41,2} = 36,97 \quad (3.27)$$

Kemudian dengan menarik garis singgung pada grafik respon kecepatan putar ICE, maka diperoleh nilai T_u dan T_N sebagai berikut:

$$T_U = 0,01$$

$$T_U + T_N = 0,41$$

$$T_N = 0,4$$

selanjutnya nilai τ diperoleh melalui Persamaan (3.28),

$$\tau = \frac{T_U}{T_N} = 0,025 \quad (3.28)$$

Untuk pendekatan model orde ke-n dapat dilakukan berdasarkan nilai τ dengan taksiran orde pada Tabel 2.4. Kemudian diperoleh orde untuk pendekatan model adalah orde ke - 2 dengan $y_i = 0,264$ yang merupakan titik singgung antara respon sistem dengan garis singgung. Selanjutnya t_i dapat diperoleh saat respon sistem mencapai 0,264 dari Y_{ss} .

$$y_i = 0,264Y_{ss}$$

$$y_i = 402$$

$$t_i = 0,135$$

Di samping itu, waktu konstan τ_{ST} dapat diperoleh dengan menggunakan Persamaan (2.14).

$$\tau_{ST} = \frac{t_i}{n - 1} = \frac{0,135}{2 - 1}$$

$$\tau_{ST} = 0,135$$

Selanjutnya diperoleh pendekatan dengan orde ke-2 berdasarkan Persamaan (2.13).

$$G_{ST}(s) = \frac{K}{(\tau_{ST}s + 1)^2}$$

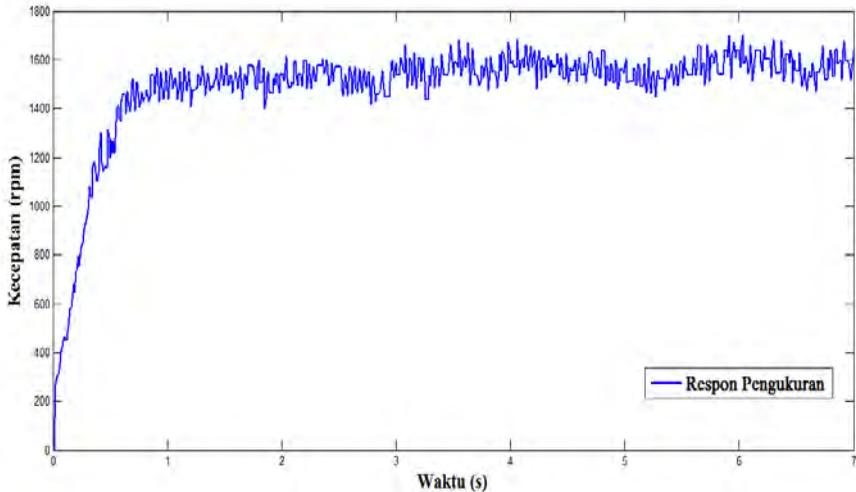
$$G_{ST}(s) = \frac{36,97}{(0,135s + 1)^2}$$

Karena $G_{ST}(s) = G(s)$, maka berdasarkan Persamaan (3.29) pemodelan ICE adalah sebagai berikut:

$$G_{MB}(s) = \frac{G(s)}{G_L(s)} \tag{3.29}$$

$$G_{MB}(s) = \frac{36,97}{(0,135s + 1)^2} (1,231 \cdot 10^{-5}s + 2,642 \cdot 10^{-5})$$

$$G_{MB}(s) = \frac{4,55 \cdot 10^{-4}s + 9,77 \cdot 10^{-4}}{(0,018225s^2 + 0,27s + 1)} \tag{3.30}$$



Gambar 3.22 Respon Kecepatan Putar ICE dengan Beban Minimal

3.5.5.2. Pembebanan Nominal

Untuk pemodelan ICE pada pembebanan nominal dilakukan dengan memberkan tegangan masukan yang telah ditentukan pada beban rem elektromagnetik. Sebelumnya untuk menentukan nilai dari tegangan masukan untuk beban nominal telah dilakukan beberapa kali percobaan. Kemudian diperoleh nilai untuk tegangan masukan pada rem elektromagnetik adalah sebesar 40 V. Untuk pengujian dan validasi respon *open loop* ICE pada pembebanan nominal ditunjukkan oleh Gambar 3.23. Dimana identifikasi pada ICE ini menggunakan metode Strejc dengan perhitungan sebagai berikut:

$$Y_{ss} = 1448$$

$$X_{ss} = 41,2$$

$$K = \frac{Y_{ss}}{X_{ss}} = \frac{1448}{41,2} = 35,14 \quad (3.31)$$

Kemudian dengan menarik garis singgung pada grafik respon kecepatan putar ICE, maka diperoleh nilai T_u dan T_u sebagai berikut:

$$T_u = 0,007$$

$$T_U + T_N = 0,407$$

$$T_N = 0,4$$

Selanjutnya untuk nilai τ diperoleh melalui Persamaan (3.32),

$$\tau = \frac{T_U}{T_N} = 0,0175 \quad (3.32)$$

Untuk pendekatan model orde ke-n dapat dilakukan berdasarkan nilai τ dengan taksiran orde yang ada pada Tabel 2.4. Kemudian diperoleh orde untuk pendekatan model adalah orde ke-2 dengan $y_i = 0,264$ yang merupakan titik singgung antara respon sistem dengan garis singgung. Selanjutnya untuk t_i dapat diperoleh saat respon sistem mencapai 0,264 dari Y_{ss} .

$$y_i = 0,264Y_{ss}$$

$$y_i = 376$$

$$t_i = 0,125$$

Sedangkan waktu konstan τ_{ST} dapat diperoleh dengan menggunakan Persamaan (2.14).

$$\tau_{ST} = \frac{t_i}{n-1} = \frac{0,125}{2-1}$$

$$\tau_{ST} = 0,125$$

Selanjutnya diperoleh pendekatan dengan orde ke-2 berdasarkan Persamaan (2.13).

$$G_{ST}(s) = \frac{K}{(\tau_{ST}s + 1)^2}$$

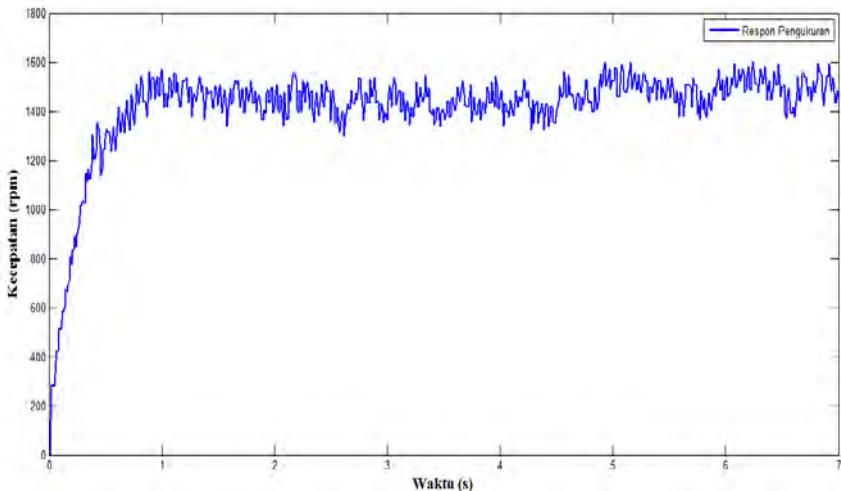
$$G_{ST}(s) = \frac{35,14}{(0,125s + 1)^2}$$

Karena $G_{ST}(s) = G(s)$, maka berdasarkan Persamaan (3.33) pemodelan ICE adalah sebagai berikut:

$$G_{MB}(s) = \frac{G(s)}{G_L(s)} \quad (3.33)$$

$$G_{MB}(s) = \frac{35,14}{(0,125s + 1)^2} (1,231 \cdot 10^{-5}s + 11,623 \cdot 10^{-5})$$

$$G_{MB}(s) = \frac{4,33 \cdot 10^{-4}s + 40,84 \cdot 10^{-4}}{(0,015625s^2 + 0,25s + 1)} \quad (3.34)$$



Gambar 3.23 Respon Kecepatan Putar ICE dengan Beban Nominal

3.5.5.3. Pembebanan Maksimal

Untuk pemodelan ICE pada pembebanan nominal dilakukan dengan memberikan tegangan masukan yang telah ditentukan pada beban rem elektromagnetik. Sebelumnya untuk menentukan nilai dari tegangan masukan untuk beban nominal telah dilakukan beberapa kali percobaan. Kemudian diperoleh nilai untuk tegangan masukan pada rem elektromagnetik adalah sebesar 80 V. Untuk pengujian dan validasi

respon *open loop* ICE pada pembebanan nominal ditunjukkan oleh Gambar 3.24. Di mana identifikasi pada ICE ini menggunakan metode Strejc dengan perhitungan sebagai berikut:

$$Y_{ss} = 1523$$

$$X_{ss} = 41,25$$

$$K = \frac{Y_{ss}}{X_{ss}} = \frac{1523}{41,25} = 36,69 \quad (3.35)$$

Kemudian dengan menarik garis singgung pada grafik respon kecepatan putar ICE, maka diperoleh nilai T_u dan T_N sebagai berikut:

$$T_u = 0,001$$

$$T_u + T_N = 0,45$$

$$T_N = 0,449$$

Selanjutnya untuk nilai τ diperoleh melalui Persamaan (3.36),

$$\tau = \frac{T_u}{T_N} = 0,000225 \quad (3.36)$$

Untuk pendekatan model orde ke-n dapat dilakukan berdasarkan nilai τ dengan taksiran orde yang ada pada Tabel 2.4. kemudian diperoleh orde untuk pendekatan model adalah orde ke-2 dengan $y_i = 0,264$ yang merupakan titik singgung antara respon sistem dengan garis singgung. Selanjutnya untuk t_i dapat diperoleh saat respon sistem mencapai 0,264 dari Y_{ss} .

$$y_i = 0,264Y_{ss}$$

$$y_i = 350$$

$$t_i = 0,130$$

Sedangkan waktu konstan τ_{ST} dapat diperoleh dengan menggunakan Persamaan (2.14).

$$\tau_{ST} = \frac{t_i}{n-1} = \frac{0,130}{2-1}$$

$$\tau_{ST} = 0,130$$

Selanjutnya diperoleh pendekatan dengan orde ke-2 berdasarkan Persamaan (2.13).

$$G_{ST}(s) = \frac{K}{(\tau_{ST}s + 1)^2}$$

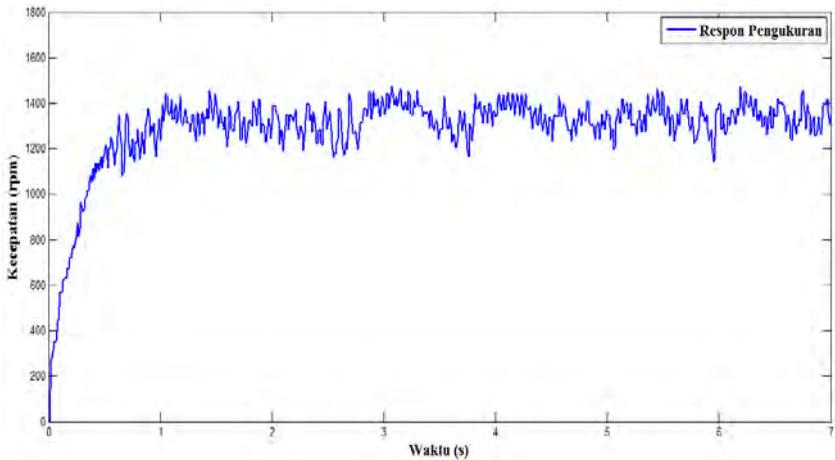
$$G_{ST}(s) = \frac{32,11}{(0,130s + 1)^2}$$

Karena $G_{ST}(s) = G(s)$, maka berdasarkan Persamaan (3.38) pemodelan ICE adalah sebagai berikut:

$$G_{MB}(s) = \frac{G(s)}{G_L(s)} \quad (3.37)$$

$$G_{MB}(s) = \frac{32,11}{(0,130s + 1)^2} (1,231 \cdot 10^{-5}s + 11,623 \cdot 10^{-5})$$

$$G_{MB}(s) = \frac{3,95 \cdot 10^{-4}s + 37,32 \cdot 10^{-4}}{(0,0169s^2 + 0,26s + 1)} \quad (3.38)$$



Gambar 3.24 Respon Kecepatan Putar ICE dengan Beban Maksimal

3.6. Pengujian dan Validasi

Proses pengujian dilakukan pada model matematika yang mendekati model *plant* yang sebenarnya, dengan menggunakan metode Strejc. Pengujian kemudian dilakukan dengan membandingkan respon

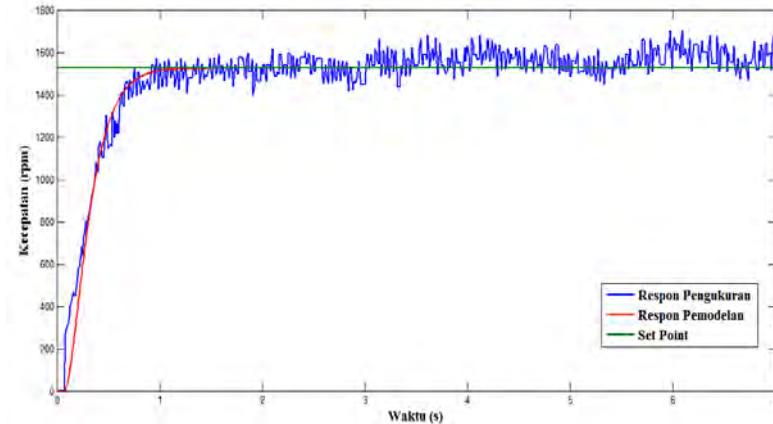
model hasil pendekatan dengan respon *plant* nyata. Dari hasil perbandingan tersebut kemudian akan dilakukan penghitungan *error* dengan menggunakan RMSE. Pengujian ini disebut dengan validasi model. Identifikasi dilakukan sebanyak 5 kali kemudian dilakukan validasi pada semua model pendekatan yang diperoleh.

Hasil pemodelan dan validasi untuk *plant* ICE dengan beban minimal dapat dilihat pada Tabel 3.10.

Tabel 3.10 Persamaan Model dan Validasi *Plant* ICE

No.	Persamaan Model <i>Plant</i>	RMSE
1	$G(s) = \frac{36,60}{(0,132s + 1)^2}$	10,25 %
2	$G(s) = \frac{36,99}{(0,137s + 1)^2}$	10,29 %
3	$G(s) = \frac{36,00}{(0,130s + 1)^2}$	10,64 %
4	$G(s) = \frac{37,00}{(0,139s + 1)^2}$	10,39 %
5	$G(s) = \frac{36,97}{(0,135s + 1)^2}$	10,20 %

Sedangkan respon hasil identifikasi dan pemodelan ICE dengan beban minimal ditunjukkan oleh Gambar 3.25.



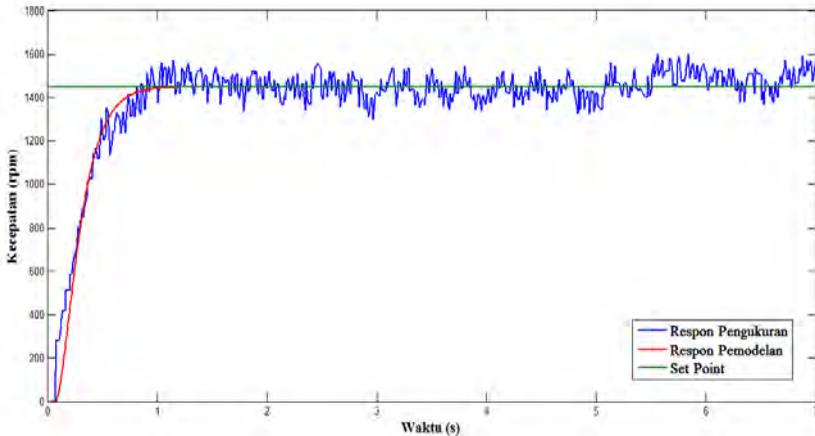
Gambar 3.25 Respon Hasil Identifikasi dan Pemodelan ICE

Kemudian untuk pemodelan dan validasi *plant* ICE dengan beban nominal dapat dilihat pada Tabel 3.11.

Tabel 3.11 Persamaan Model dan Validasi *Plant* ICE

No.	Persamaan Model Plant	RMSE
1	$G(s) = \frac{35,00}{(0,128s + 1)^2}$	10,08 %
2	$G(s) = \frac{35,22}{(0,128s + 1)^2}$	10,04 %
3	$G(s) = \frac{35,14}{(0,125s + 1)^2}$	9,90 %
4	$G(s) = \frac{34,98}{(0,129s + 1)^2}$	10,14 %
5	$G(s) = \frac{34,80}{(0,124s + 1)^2}$	9,94 %

Dan respon hasil identifikasi dan pemodelan ICE dengan beban nominal ditunjukkan oleh Gambar 3.26.



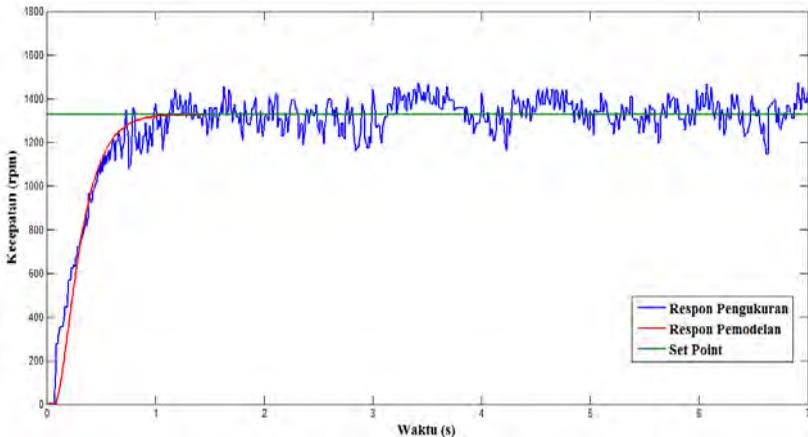
Gambar 3.26 Respon Hasil Identifikasi dan Pemodelan ICE

Selanjutnya untuk pemodelan dan validasi *plant* ICE dengan beban maksimal dapat dilihat pada Tabel 3.12.

Tabel 3.12 Persamaan Model dan Validasi *Plant* ICE

No.	Persamaan Model Plant	RMSE
1	$G(s) = \frac{32,14}{(0,133s + 1)^2}$	10,82 %
2	$G(s) = \frac{31,50}{(0,127s + 1)^2}$	10,75 %
3	$G(s) = \frac{32,11}{(0,130s + 1)^2}$	10,69 %
4	$G(s) = \frac{32,50}{(0,135s + 1)^2}$	10,94 %
5	$G(s) = \frac{32,80}{(0,128s + 1)^2}$	10,77 %

Untuk respon hasil identifikasi dan pemodelan ICE dengan beban maksimal ditunjukkan oleh Gambar 3.27.



Gambar 3.27 Respon Hasil Identifikasi dan Pemodelan ICE

Setelah diperoleh hasil validasi untuk model ICE. Maka dari hasil pengujian dan validasi yang dimuat dalam Tabel 3.10, Tabel 3.11, dan Tabel 3.12. Selanjutnya dapat ditentukan model ICE berdasarkan nilai dari RMSE yang terkecil. Sehingga diperoleh model ICE untuk setiap variasi beban yang ditunjukkan oleh Tabel 3.13.

Tabel 3.13 Model *Plant* ICE yang Digunakan

No.	Jenis Pembebanan	Persamaan Model <i>Plant</i> ICE	RMSE
1	Minimal	$G(s) = \frac{36,97}{(0,135 + 1)^2}$	10,20 %
2	Nominal	$G(s) = \frac{35,14}{(0,125 + 1)^2}$	9,90 %
3	Maksimal	$G(s) = \frac{32,11}{(0,130 + 1)^2}$	10,69 %

Dari tiga tabel sebelumnya telah ditunjukkan data hasil dari pengujian dan validas yang telah dilakukan. Di mana model yang dipakai berdasarkan nilai RMSE yang paling kecil. Pemodelan dengan beberapa metode dapat dilihat pada Tabel 3.14 yang menunjukkan bahwa RMSE paling kecil dapat diapai dengan menggunakan metode Strejc sehingga hasil fungsi alihnya ICE lebih mirip terhadap respon yang sesungguhnya.

Tabel 3.14 Pemodelan dengan Beberapa Metode

Metode	Persamaan Model <i>Plant</i>	RMSE
Latzel's	$G(s) = \frac{35,14}{(0,2151s + 1)^2}$	15,81%
Strejc	$G(s) = \frac{35,14}{(0,125s + 1)^2}$	9,90 %
Viteckova 2 nd Order	$G(s) = \frac{35,14}{(0,158s + 1)^2}$	11,94%

3.7. Desain Kontroler PID – PSO

Dalam mendesain kontroler PID – PSO, maka perlu diperhatikan beberapa hal berikut ini :

3.7.1. Algoritma PSO Untuk Tuning Parameter PID

Pada proses *tuning* parameter PID yang menggunakan algoritma PSO terdapat suatu bagian penting yang berfungsi dalam memandu proses optimalisasi untuk mendapatkan nilai dari parameter kontroler. Bagian yang dimaksud adalah *fitness function*, karena *fitness function* berperan dalam menentukan nilai yang optimal dari nilai parameter yang didapat. Pada Tugas Akhir ini, *fitness function* ditentukan oleh dua faktor, yaitu

nilai *error* yang dihitung dari persamaan *Integral Absolute Error* (IAE) dan nilai *overshoot*.

Untuk merepresentasikan nilai *fitness*, maka digunakan suatu persamaan dalam program untuk mencari nilai dari kedua faktor tersebut. *Fitness function* yang digunakan dalam program PSO ini dinyatakan dalam persamaan berikut:

$$F(i) = \alpha \cdot IAE(i) + \beta \cdot O(i) \quad (3.39)$$

Dengan,

IAE : *Integral Absolute Error*
O : *Overshoot*
 α, β : *improvement weight*

Perancangan kontroler PID – PSO dengan menggunakan algoritma PSO dalam menentukan nilai parameter K_p, K_i, K_d dapat dilakukan dengan langkah – langkah seperti berikut:

Langkah 1

Terlebih dahulu menentukan jumlah partikel, jumlah iterasi, dimensi, konstanta kognitif, konstanta sosial, bobot inersia, serta inisialisasi partikel secara acak dengan menggunakan program utama MATLAB.

Langkah 2

Menjalankan sistem pada tiap parameter (K_p, K_i, dan K_d) yang diperoleh setiap iterasi dan langsung dimasukkan ke kontroler PID yang dijalankan melalui *simulink*.

Langkah 3

Menghitung nilai dari *Integral Absolute Error* (IAE) dan *overshoot* dari langkah 2. Nilai IAE dan *overshoot* langsung dihitung menggunakan program PSO pada MATLAB.

Langkah 4

Menghitung *fitness function* dari PSO. Karena nilai dari *fitness function* akan menentukan nilai dari parameter K_p, K_i, dan K_d yang optimal.

Langkah 5

Menghitung dan menentukan p_{best} serta g_{best} . kemudian dari nilai $fitness$ setiap partikel, akan ditentukan p_{best} dan g_{best} dalam satu populasi.

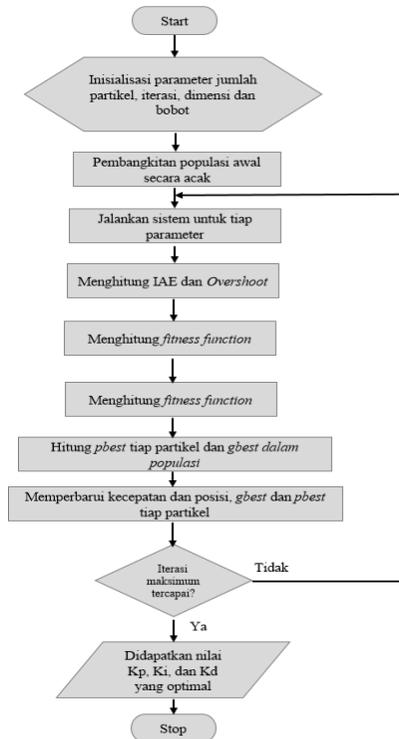
Langkah 6

Memperbarui kecepatan dan posisi dari tiap partikel p_{best} dan g_{best} .

Langkah 7

Mengulang kembali langkah kedua sampai terakhir hingga jumlah itersi terpenuhi, dan kemudian diperoleh nilai dari parameter K_p , K_i , dan K_d yang optimal.

Untuk diagram alir dari algoritma PSO ditunjukkan oleh Gambar 3.28.

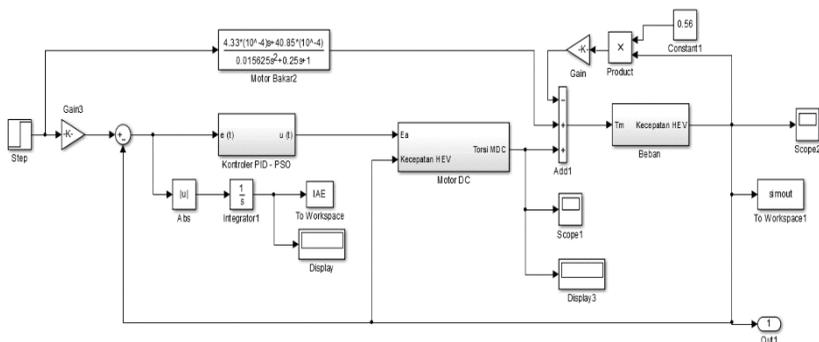


Gambar 3.28 Diagram Alir Algoritma PSO

Untuk program PSO sendiri terdiri dari dua macam program, program pertama merupakan program utama yang berisi mengenai jumlah partikel, jumlah iterasi, dimensi, konstanta kognitif, konstanta sosial, bobot inersia, serta inialisasi kondisi awal. Sedangkan untuk program yang kedua merupakan subprogram mengenai simulasi dari sistem yang menggunakan *simulink* dan perolehan nilai *fitness*.

3.7.2. Desain Simulink PID – PSO

Untuk desain dari kontroler PID – PSO dengan menggunakan *simulink* ditunjukkan oleh Gambar 3.29. Selain itu desain *simulink* juga dirancang untuk langsung menghitung nilai IAE, yang kemudian langsung dimasukkan ke dalam program MATLAB. Dengan demikian, hasil perhitungan untuk parameter kontroler PID dari algoritma PSO di program MATLAB dapat langsung dijalankan di *simulink* untuk diketahui hasilnya.



Gambar 3.29 Diagram *Simulink*

BAB 4

PENGUJIAN DAN ANALISIS

Pada Bab 3 telah dijelaskan tentang perancangan sistem pengatuan simulator HEV. Bab ini akan menjelaskan hasil simulasi sistem dengan berbagai kondisi pembebanan dan menampilkan hasil dari implemenatasi sistem serta hasil kerja dari kontroler yang digunakan.

4.1. Gambaran Umum Pengujian Sistem

Proses pengujian dilakukan dengan tujuan untuk melihat bagaimana kinerja dari motor DC yang digunakan sebagai penggerak pendukung dari mesin bakar pada sistem *Hybrid Electric Vehicle*. Selain itu juga dilakukan simulasi untuk desain kontroler yang telah dibuat pada sistem. Oleh karena itu, dengan adanya simulasi dan implementasi terhadap sistem, dapat diketahui performansi dari *Hybrid Electric Vehicle* dengan adanya pembeban berlebih berupa rem elektromagnetik.

Pengujian sendiri dilakukan pada dua kondisi, pertama pada saat ICE ditambahkan beban rem elektromagnetik tanpa bantuan dari motor DC. Dan yang kedua pada saat ICE ditambahkan beban rem elektromagnetik yang dibantu dengan motor DC menggunakan kontroler *Proportional Integral Derivative – Particle Swarm Optimization*.

4.2. Simulasi Sistem

Sebelum melakukan implementasi kontroler pada *plant*, maka terlebih dahulu perlu dilakukan simulasi. Karena dengan adanya simulasi yang sesuai, maka akan memudahkan dalam penerepan kontroler tersebut pada *plant* yang diuji. Simulasi sendiri dilakukan dengan menggunakan *software* MATLAB R2013a.

4.2.1. Simulasi dengan *Tuning* PSO

Pada simulator HEV kontroler PID diterapkan pada motor DC untuk membantu ICE dalam mengatasi permasalahan regulasi dengan mempertahankan kecepatan putar simulator HEV. Kemudian parameter K_p , K_i , dan K_d yang dijadikan salah satu penentu keberhasilan dari kontroler PID. Oleh karena itu, pada Tugas Akhir ini digunakan algoritma cerdas *Particle Swarm Optimization* untuk proses *tuning* parameter K_p , K_i , dan K_d . Diperoleh parameter PSO yang digunakan untuk proses *tuning* adalah sebagai berikut:

- a. Jumlah partikel, $n = 10$
- b. Jumlah iterasi, $iter = 30$
- c. Bobot inersia untuk PSO modifikasi, $w_1 = 0,75$ dan $w_2 = 0,25$
- d. Konstanta akselerasi kognitif, $c_1 = 1,5$
- e. Konstanta akselerasi sosial, $c_2 = 1,5$

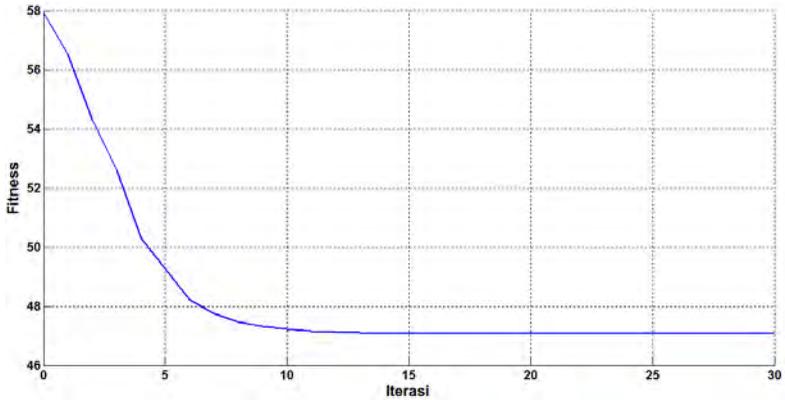
Pada proses *tuning* digunakan salah satu parameter PSO yang berbeda – beda, khususnya pada bobot inersia. Bobot inersia yang digunakan adalah 0,38; 0,48; 0,58. Hal ini dilakukan untuk memperoleh hasil *tuning* parameter Kp, Ki, dan Kd yang terbaik dengan memperhatikan nilai IAE dan *overshoot* yang dihasilkan. Hal tersebut dilakukan untuk memperoleh performansi terbaik sebelum diterapkan pada *plant* yang sesungguhnya. Selain itu, pada PSO proses pencarian nilai parameter memerlukan waktu yang cukup lama, hal ini dikarenakan dengan jumlah partikel dan jumlah iterasi yang dipilih. Oleh sebab itu, ditentukan jumlah partikel sebanyak $n = 10$ dan jumlah iterasi sebanyak $iter = 30$. Karena *fitness function* sudah mendapatkan titik konvergensi pada titik tersebut. Berikut hasil yang diperoleh dari program PSO :

Tabel 4.1 Hasil *Tuning* Kp, Ki, dan Kd dengan PSO

Item		Bobot Inersia		
		w = 0,38	w = 0,48	w = 0,58
Parameter Kontroler	Kp	7,712	9,798	3,698
	Ki	10,008	20,863	13,115
	Kd	0,645	0,153	0,224
Fungsi Objektif	IAE	20,99	20,37	22,36
	Overshoot	0	0	0
Performansi PSO	Nilai <i>fitness</i>	47,0780	37,8468	42,9821

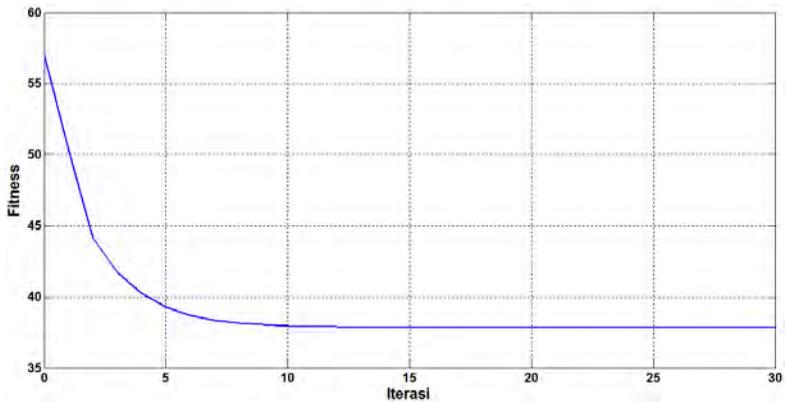
Dari Tabel 4.1 dapat diketahui untuk nilai dari parameter Kp, Ki, dan Kd yang telah diperoleh. Dan berikut ini merupakan grafik dari konvergensi hasil *tuning* parameter Kp, Ki, dan Kd dengan algoritma PSO ditunjukkan oleh untuk $w = 0,38$; untuk $w = 0,48$; dan untuk $w = 0,58$.

- a. Untuk $w = 0,38$ diperoleh nilai dari parameter kontroler PID sebagai berikut:
Kp = 7,712; Ki = 10,008; dan Kd = 0,645



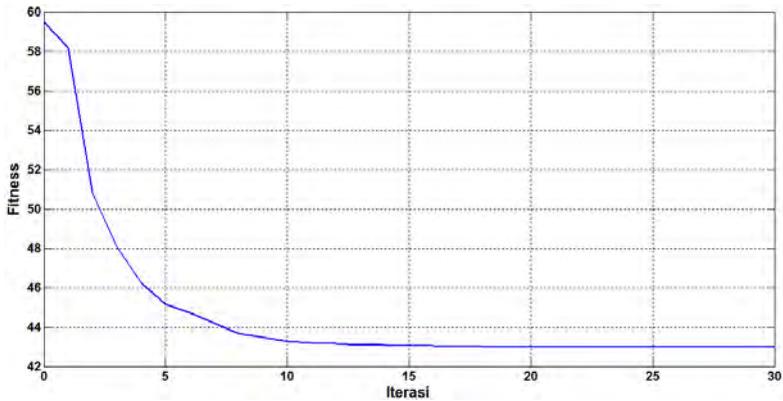
Gambar 4.1 Konvergensi PSO saat Bobot Inersia $w = 0,38$

- b. Untuk $w = 0,48$ diperoleh nilai dari parameter kontroler PID sebagai berikut:
 $K_p = 9,798$; $K_i = 20,863$; dan $K_d = 0,153$



Gambar 4.2 Konvergensi PSO saat Bobot Inersia $w = 0,48$

- c. Untuk $w = 0,58$ diperoleh nilai dari parameter kontroler PID sebagai berikut:
 $K_p = 3,698$; $K_i = 13,115$; dan $K_d = 0,224$



Gambar 4.3 Konvergensi PSO saat Bobot Inersia $w = 0,58$

4.2.2. Simulasi ICE Berbeban Tanpa Bantuan Motor DC

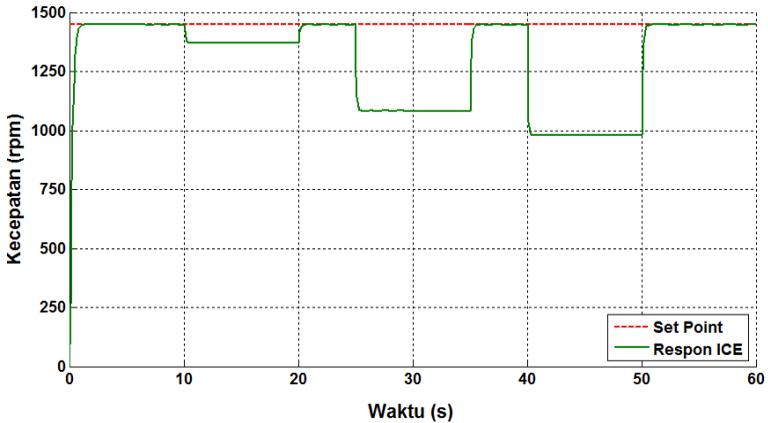
Tujuan dari dilakukannya simulasi ICE yang ditambahkan beban tanpa ada bantuan dari motor DC adalah untuk mengetahui respon dari ICE saja terhadap efek pembebanan yang diberikan. Dengan cara menambahkan tiga beban berupa tegangan masukan rem elektromagnetik yang dapat menghasilkan arus pengereman dengan besar 0,7 A; 0,8 A; dan 0,9 A. Efek dari pemberian beban terhadap penurunan kecepatan putar dapat dilihat pada Tabel.

Untuk pengujian sendiri dilakukan dengan memberi sinyal masukan berupa sinyal step yang merepresentasikan dari bukaan *throttle* pada ICE sebesar 41,2%. Selain itu dengan memperhatikan spek yang ada pada rem, maka untuk pemberian tegangan masukan maksimal pada rem elektromagnetik dibatasi pada tegangan masukan sebesar 80 VDC yang menghasilkan arus pengereman dengan nilai sebesar 1,14 A. Jadi untuk pemberian arus pengereman untuk pembebanan di range nominal 0,56 – 1,14 A.

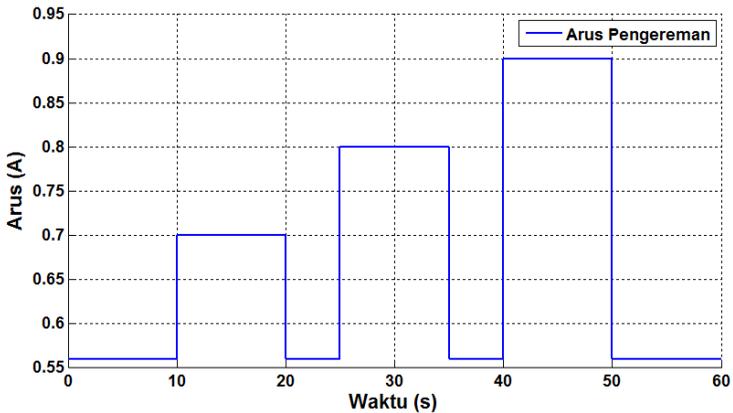
Tabel 4.2 Variasi Arus Pengereman pada Rem Elektromagnetik

Tegangan Beban (Volt)	Arus Pengereman (A)	Penurunan Kecepatan Putar (rpm)
50	0,7	76,8094
57,14	0,8	365
64,28	0,9	468,3

Dari simulasi dengan penambahan beban dalam selang beberapa waktu, diperoleh grafik respon penurunan kecepatan dari ICE yang ditunjukkan oleh Gambar 4.4. Dari grafik tersebut terlihat bahwa respon ICE mengalami penurunan kecepatan ketika adanya beban yang diberikan dalam rentang waktu tertentu. Dan respon ICE juga dapat kembali lagi menuju nilai *steady state* setelah tidak adanya beban.

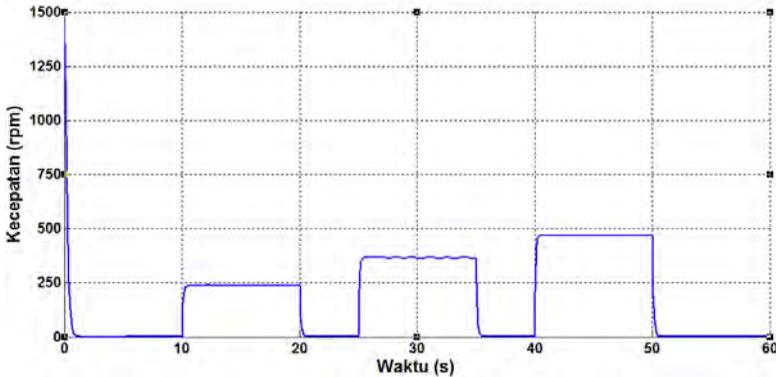


Gambar 4.4 Respon Kecepatan Putar ICE dengan 3 Variasi Beban Arus Pengereman



Gambar 4.5 Besar Arus Pengereman yang Diberikan

Karena pada simulasi ini tidak menggunakan kontroler, maka untuk sinyal kesalahan yang dihasilkan dari respon keluaran ICE tersebut, sama dengan respon penurunan kecepatan putar ICE seperti yang ditunjukkan oleh Gambar 4.6.



Gambar 4.6 Sinyal Kesalahan ICE dengan 3 Variasi Beban Arus Pengereman

4.2.3. Simulasi ICE Berbeban dengan Bantuan Motor DC Menggunakan Kontroler PID-PSO

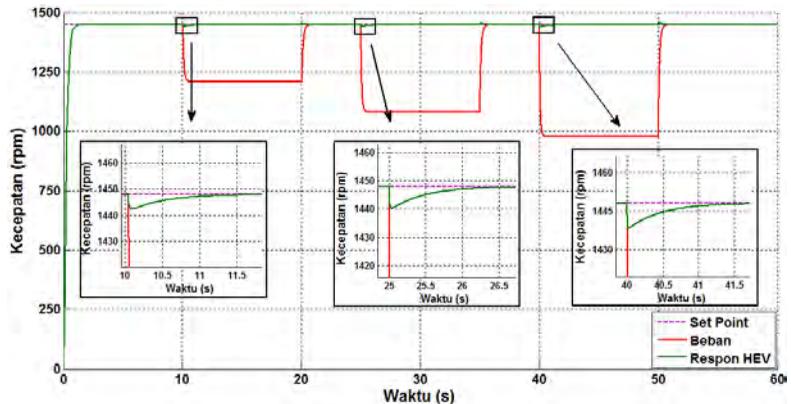
Selanjutnya dilakukan simulasi ICE yang ditambahkan beban dengan bantuan motor DC menggunakan kontroler PID berdasarkan algoritma PSO. Sebelumnya telah dilakukan proses *tuning* parameter pada kontroler PID dan didapatkan untuk nilai K_p , K_i , dan K_d seperti yang ditunjukkan pada Tabel 4.1. Tujuan dari digunakannya kontroler PID-PSO pada Tugas Akhir ini adalah untuk menyelesaikan permasalahan regulasi dengan mempertahankan kecepatan putar ketika terdapat pembebanan pada simulator HEV.

Pada simulasi ini dilakukan pengujian seperti sebelumnya, yaitu dengan memberikan tiga beban yang berbeda nilainya. Dengan digunakannya kontroler PID pada motor DC sebagai penggerak pendukung, maka nantinya ketika terjadi penurunan respon kecepatan ICE. Motor DC dapat membantu ICE dalam memperbaiki respon kecepataannya. Pada pengujian pertama diberikan arus pengereman sebesar 0,7 A dan terjadi penurunan kecepatan sebesar 76,8094 rpm. Saat ICE mengalami penurunan kecepatan, motor DC akan bekerja membantu

ICE dengan menggunakan kontroler PID-PSO sampai kecepatan dari simulator HEV kembali sesuai dengan *set point*.

Sedangkan pada pengujian yang kedua dengan beban yang lebih tinggi dengan tegangan masukan rem sebesar 57,14 V (arus pengereman 0,8A), dan menyebabkan penurunan kecepatan menjadi lebih besar sekitar 365 rpm, tetapi simulator HEV masih dapat bekerja dengan baik.

Pengujian ketiga dilakukan dengan memberikan tegangan masukan pada rem sebesar 64,28 V yang menghasilkan arus pengereman sebesar 0,9 A. Dengan adanya arus pengereman sebesar 0,9 A menyebabkan terjadinya penurunan kecepatan putar sebesar 468,3 rpm. Untuk respon kecepatan putar ICE dengan bantuan motor DC menggunakan kontroler PID-PSO dengan 3 variasi beban dapat dilihat pada Gambar 4.7.



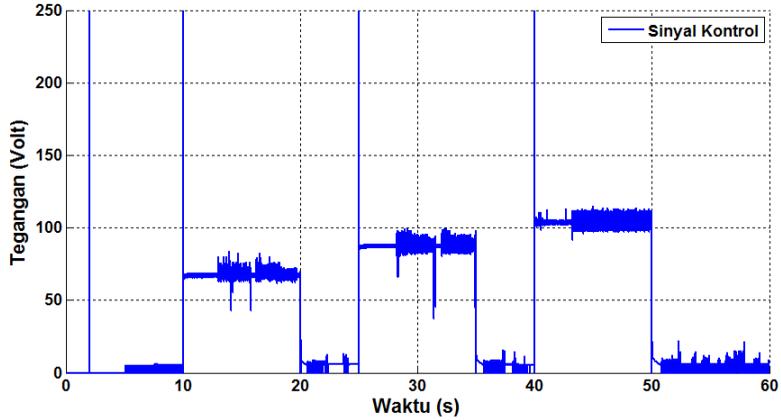
Gambar 4.7 Respon Kecepatan Putar HEV dengan 3 Variasi Beban Arus Pengereman

Dari respon HEV yang dihasilkan diperoleh data performansi HEV yang ditunjukkan pada

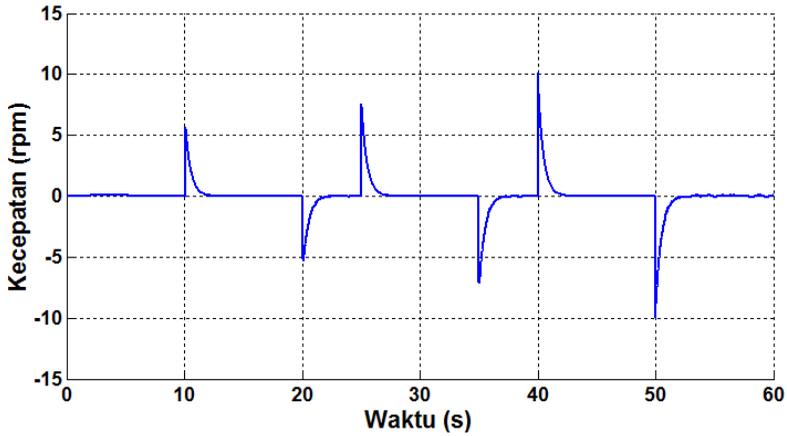
Tabel 4.3 Performansi HEV dengan Kontroler PID - PSO

	Arus Pengereman		
	0,7 A	0,8 A	0,9 A
Penurunan Kecepatan	6 rpm	8 rpm	11 rpm
t_s	1,6 detik	1,7 detik	1,7 detik

	Arus Pengereman		
	0,7 A	0,8 A	0,9 A
E_{ss}	0 rpm	0 rpm	0 rpm
Overshoot	0	0	0



Gambar 4.8 Sinyal Kontrol yang Diberikan pada Motor DC



Gambar 4.9 Sinyal Kesalahan dari Sistem dengan 3 Variasi Beban Arus Pengereman

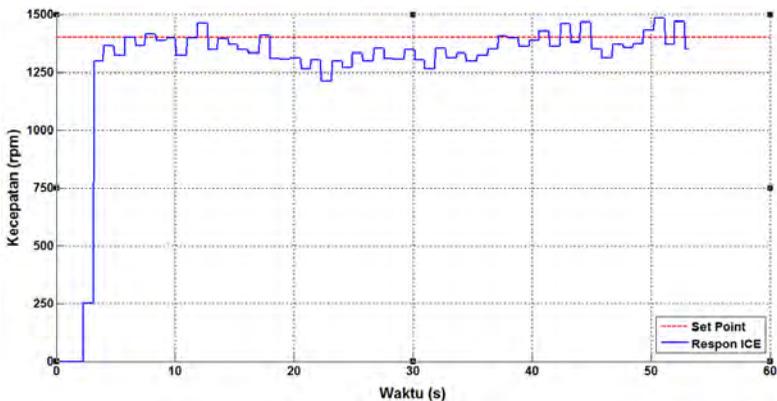
Kemudian dari pengujian yang telah dilakukan diperoleh grafik untuk sinyal kontrol yang diberikan pada motor DC serta grafik sinyal kesalahan yang dihasilkan oleh sistem yang ditunjukkan oleh Gambar 4.8 dan Gambar 4.9. Selain itu, pada pengujian sistem HEV kontroler PID – PSO digunakan dalam mengatur kecepatan motor DC. Karena pada saat ICE mengalami penurunan kecepatan putar, maka motor DC dibutuhkan untuk membantu kinerja ICE agar simulator HEV dapat mempertahankan nilai *set point* saat adanya pembebanan.

4.3. Implementasi Sistem

Untuk implementasi dilakukan pada *plant* secara langsung. Pengujian dilakukan dengan menghubungkan *plant* HEV terhadap kontroler yang telah dirancang pada MATLAB R2013a melalui Arduino *board* dan *driver* motor DC. Kemudian untuk kriteria pengujian sistem pada *plant* berdasarkan beban dari rem elektromagnetik. Selain itu, keluaran kontroler berupa sinyal kontrol akan dihubungkan ke driver motor dengan pengiriman data secara serial untuk menjalankan motor DC sesuai dengan pengaturan PWM dari Arduino.

4.3.1. Implementasi ICE Berbeban Tanpa Bantuan Motor DC

Pada tahap implementasi ini pengaruh pembebanan dari rem elektromagnetik terhadap *plant* simulator HEV dapat dilihat dari respon kecepatan putar ICE tanpa bantuan dari motor DC hasil pembacaan *rotary encoder*.

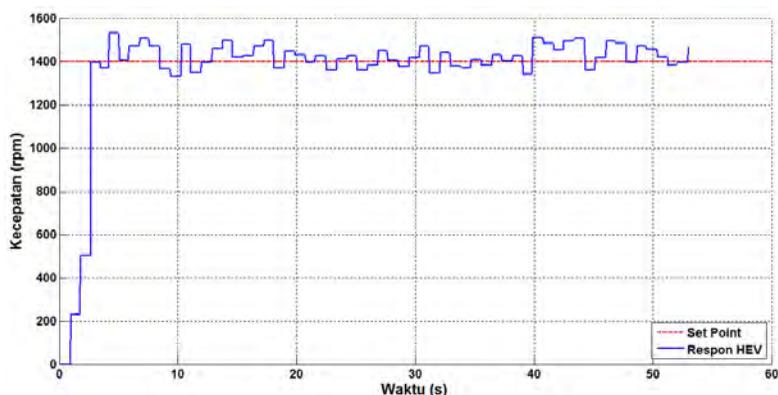


Gambar 4.10 Respon Kecepatan Putar ICE dengan Beban 0,9 A

Untuk beban yang diberikan hanya 1 beban arus pengereman sebesar 0,9 A yang setara dengan tegangan masukan 64,28 V. Ketika ICE diberikan beban arus pengereman, kecepatan putar ICE mengalami penurunan kecepatan putar sebesar 189 rpm.

4.3.2. Implementasi ICE Beberbean dengan Bantuan Motor DC Menggunakan Kontroler PID - PSO

Pada ICE tanpa bantuan motor listrik telah diketahui efek pembebanan yang diberikan oleh rem elektromagnetik. Setelah mengetahui efek pembebanan tersebut maka digunakan kontroler PID - PSO untuk memperbaiki performansi dari simulator HEV dalam permasalahan regulator, dengan adanya kombinasi kerja dari motor DC dan ICE.



Gambar 4.11 Respon Kecepatan Putar HEV dengan Kontroler PID – PSO dengan Beban Arus Pengereman 0,9 A

Penggunaan kontroler PID – PSO, diharapkan mampu mempertahankan kecepatan ketika terjadi pembebanan, sehingga nilai dari respons sistem tetap mendekati nilai *steady state* di sekitar *set point*. Pada Gambar 4.11 terlihat bahwa ketika ICE dibantu dengan motor DC menggunakan kontroler PID - PSO, respon kecepatan putar yang dihasilkan oleh simulator HEV menjadi lebih baik. Hal tersebut ditunjukkan dengan nilai dari respon kecepatan putar yang relatif terjaga di sekitar *set point*. Maka dalam hal ini untuk penerapan kontroler PID –

PSO dalam menanggulangi permasalahan regulator, ternyata mampu membuat motor DC membantu kerja ICE ketika mendapatkan pembebanan berlebih untuk menjaga nilai kecepatan putar HEV tetap di sekitar *set point*.

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

- [1] Paramitha, Nyoman P., "Desain Kontrol Fuzzy untuk Distribusi Torsi pada Hybrid Electric Vehicle", *Tugas Akhir*, Jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember, Surabaya, 2014.
- [2] A. Jalilvand, A. Kimiyaghalam, A. Ahouri and H. Kord, "Optimal Tuning of PID Controller Parameters on DC Motor Based on Advanced Particle Swarm Optimization Algorithm", *International Journal on Technical and Physical Problems of Engineering*, vol. III, no. 9, pp. 10 -17, 2011.
- [3] Momoh, Omonowo D. and Omonigui, Michael O., "An Overview of Hybrid Electric Vehicle Technology," *IEEE*, 978-1-4244-2601, September, 2009.
- [4] Ramadhani, Nugroho Adi, "Perancangan dan Implementasi Kontroler Linear Quadratic Tracking (LQT) untuk Pengaturan Akselerasi pada Simulator Hybrid Electric Vehicle (HEV)", *Tugas Akhir*, Jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember, Surabaya, 2013.
- [5] Sholihah, Suci E., "Desain dan Implementasi Kontroler Sliding Mode untuk Pengaturan Akselerasi pada Simulator Hybrid Electric Vehicle", *Tugas Akhir*, Jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember, Surabaya, 2013.
- [6] Zuhail, *Dasar Teknik Tenaga Listrik dan Elektronika Daya*, Jakarta, PT Gramedia, 1990.
- [7] Mohammad, A. P., "Desain dan Implementasi Kontroler Fuzzy PID Sliding Mode untuk Pengendalian Simulator Hybrid Electric Vehicle", *Tesis*, Jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember, Surabaya, 2012.
- [8] Wibowo, Bayu P., "Traction Control pada Parallel Hybrid Electric Vehicle (HEV) dengan Menggunakan Metode Kontrol Neuro-Fuzzy Prediktif", *Tugas Akhir*, Jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember, Surabaya, 2014.

- [9] Galih. S.A.W, “Desain dan Implementasi Kontroler Optimal Berbasis Neuro Fuzzy untuk Pengendalian Simulator Hybrid Electric Vehicle”, *Tugas Akhir*, Jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember, Surabaya, 2010.
- [10] _____, *Index of Arduino Boards* <URL: <http://www.arduino.cc/en/Main/ArduinoBoardUno>>, Maret 2015.
- [11] Subroto, Ramadhani K., "Desain dan Implementasi Kontroler Optimal Sliding Mode untuk Tracking Posisi Motor DC MS150", *Tugas Akhir*, Jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember, Surabaya, 2011.
- [12] Ing. Pavel Jakoubek, "Experimental Identification of Stable Nonoscillatory Systems from Step-Responses by Selected Methods", in *Konference Studenske Tvuurci-Cinnosti*, 2009.
- [13] Ogata, K., *Modern Control Engineering Third Edition*, Prentice-Hall, 1997.
- [14] Alrijadjis, "Optimasi Parameter Kontroler PID Berbasis Algoritma Particle Swarm Optimization (PSO) untuk Sistem dengan Waktu Tunda", *Tesis*, Jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember, Surabaya, 2010.
- [15] Talukder, Satyobroto, "Mathematical Modelling and Applications of Particle Swarm Optimization", *Thesis*, Department of Mathematics and Science, Blekinge Institute of Technology, 2010.

BAB 5

PENUTUP

5.1. Kesimpulan

Dari hasil pengujian dan analisis yang telah dilakukan diperoleh kesimpulan sebagai berikut:

- a. Parameter kontroler PID berupa K_p , K_i , dan K_d yang optimal dapat dicari dengan menggunakan algoritma *Particle Swarm Optimization* berdasarkan parameter PSO yang telah ditetapkan. Dan diperoleh nilai parameter kontroler PID sebesar $K_p = 9,798$; $K_i = 20,863$; dan $K_d = 0,153$.
- b. Jumlah partikel dan iterasi dalam algoritma PSO akan menentukan waktu komputasi dan solusi optimal. Semakin banyak partikel dan iterasi akan menghasilkan solusi yang lebih baik, namun membutuhkan waktu komputasi yang semakin lama. Dalam sistem *off – line*, waktu komputasi didesain sesingkat mungkin dengan hasil yang masih dapat ditoleransi. Dengan demikian dibutuhkan pembatasan dan pengaturan jumlah partikel dan iterasi.
- c. Berdasarkan hasil pengujian secara simulasi, motor DC dapat membantu kinerja ICE terhadap adanya pembebanan berupa arus pengereman (0,7 A; 0,8 A; 0,9 A) yang kemudian diperoleh data performansi HEV berupa penurunan kecepatan putar sebesar 6 -11 rpm, rata – rata *settling time* ($t_s = 1,6$ detik) serta $E_{ss} = 0\%$ dan *overshoot* = 0 rpm. Dari data performansi HEV tersebut, maka simulator HEV telah mampu mempertahankan kecepatan putar sesuai dengan *set point* yang diberikan.

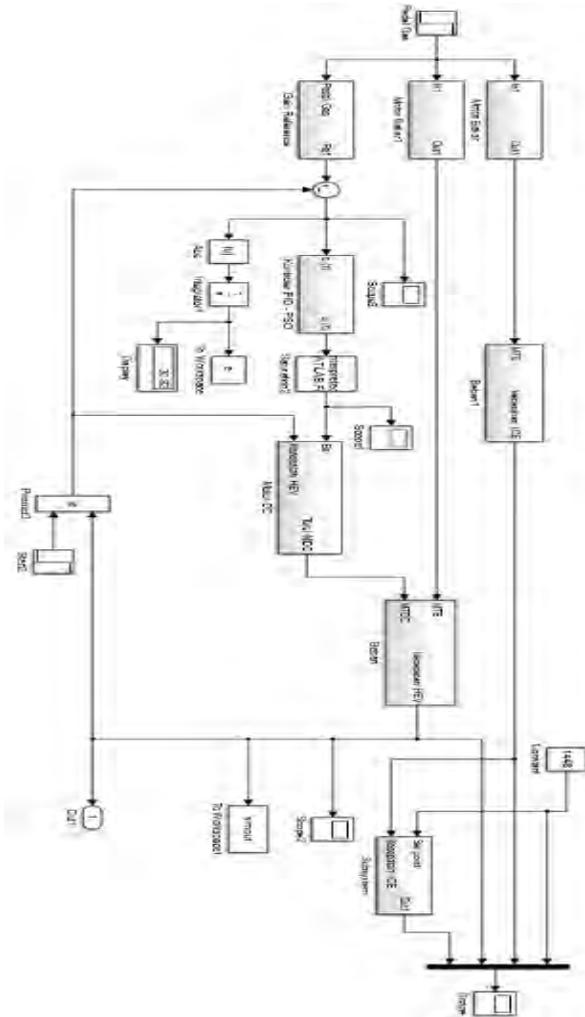
5.2. Saran

Karena *plant* yang digunakan bersifat *nonlinear* maka diharapkan untuk mengetahui terlebih dahulu tentang karakteristik *plant* tersebut serta metode identifikasi yang digunakan. Selain itu pemilihan metode kontrol yang digunakan juga harus sesuai. Pada PSO sendiri pemilihan parameter untuk proses *tuning* sangat berpengaruh, sehingga nilai hasil *tuning* yang diperoleh juga bergantung dari pemilihan parameter PSO yang digunakan.

Halaman ini sengaja dikosongkan

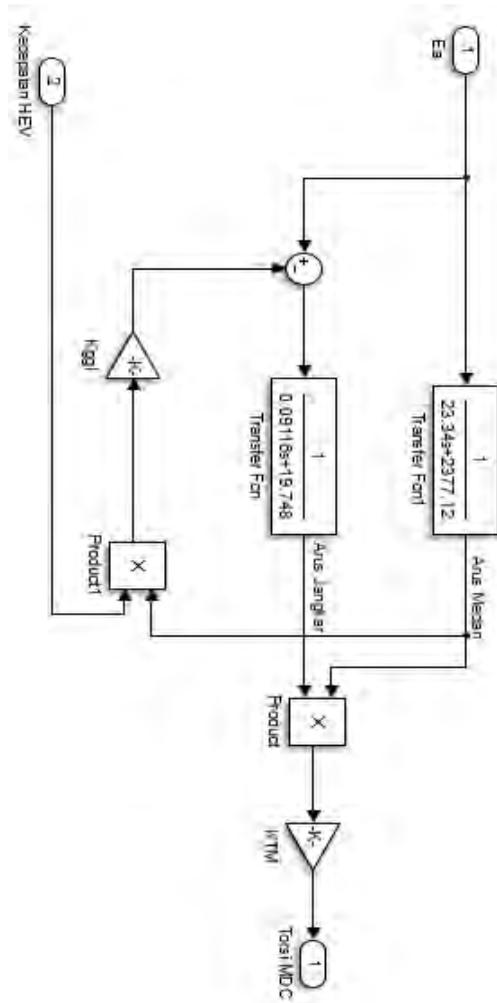
LAMPIRAN

A. Tampilan *Simulink* Simulasi HEV



Gambar A. Blok *Simulink* Sistem HEV

B. Tampilan *Simulink* Motor DC



Gambar B. Blok *Simulink* Motor DC

C. Program *Particle Swarm Optimization*

Main Program

```
% Judul Tugas Akhir
% Pengaturan Kecepatan Motor DC pada Simulator
Hybrid Electric Vehicle (HEV) Menggunakan
Kontroler PID Berdasarkan Algoritma Particle
Swarm Optimization (PSO)
%
% Author:
% Alif Ridwan Maulana
% 2211 100 117
% Jurusan Teknik Elektro
% Bidang Studi Teknik Sistem Pengaturan
% Fakultas Teknologi Industri
% Institut Teknologi Sepuluh Nopember Surabaya

clear
clc

%% Inisialisasi
n = 10;           % Jumlah Partikel
bird_step = 30;  % Iterasi Maksimum
dim = 3;         % Dimension problem
c2 = 1.5;        % Konstanta Sosial
c1 = 1.5;        % Konstanta Kognitif
w = 0.48;        % PSO Bobot Inersia Orisinil
w2 = 0.5;        % PSO Bobot Inersia Akhir
w1 = 0.1;        % PSO Bobot Inersia Awal

fitness=zeros(n,bird_step); % Nilai Fitness
Awal

% Bilangan random (0-1) untuk pengaruh kognitif
dan sosial
R1 = rand(dim, n);
R2 = rand(dim, n);
```

```

current_fitness =zeros(n,1);

% Setting daerah pencarian awal
current_position1 = 10*(rand(1, n)); %0.5 10 2
current_position2 = 7*(rand(1, n)); %0.05 7 3
current_position3 = 0.5*(rand(1, n)); %0.5 0.5
0.3
current_position =
[current_position1;current_position2;current_pos
ition3];
velocity1 = 0.1*rand(1, n) ; %0.05 0.1 0.2
velocity2 = 0.07*rand(1, n) ; %0.005 0.07 0.3
velocity3 = 0.05*rand(1, n) ; %0.05 0.05 0.03
velocity = [velocity1;velocity2;velocity3];
local_best_position = current_position ;

% Evaluasi fitness tiap partikel

for i = 1:n;
    pos=(current_position(:,i))';
    Kp=pos(1);
    Ki=pos(2);
    Kd=pos(3);
    current_fitness(i) =
objf((current_position(:,i))');
end
%% Penentuan pbest dan gbest
local_best_fitness = current_fitness ;
[global_best_fitness,g] =
min(local_best_fitness);

    global_best_position=zeros(dim,n);
for i=1:n;
    global_best_position(:,i) =
local_best_position(:,g) ;
end

```

```

% Perhitungan kecepatan dan posisi awal partikel

% Kecepatan
velocity = w *velocity +
c1*(R1.*(local_best_position-current_position))
+ c2*(R2.*(global_best_position-
current_position));

% Posisi
current_position = current_position + velocity;

%% Main Loop berfungsi untuk update partikel
iter = 0 ;           % Iterasi awal
zz=global_best_fitness;
z1=iter;
while ( iter < bird_step )
    iter = iter + 1;
    for i = 1:n,
        pos=(current_position(:,i))';
        Kp=pos(1);
        Ki=pos(2);
        Kd=pos(3);
        current_fitness(i) =
objf(current_position(:,i)) ;
    end
    % Mean Value
    kk(iter,1) = iter;
    jj(iter,1) = current_fitness(i)/n;

    % Standar Deviasi
    sd(iter,1) = (((current_fitness(i)-
jj(iter,1))^2)/n)^0.5;

    for i = 1 : n
        if current_fitness(i) <
local_best_fitness(i)
            local_best_fitness(i) =
current_fitness(i);

```

```

        local_best_position(:,i) =
current_position(:,i) ;
        end
    end

    [current_global_best_fitness,g] =
min(local_best_fitness);

    if current_global_best_fitness <
global_best_fitness
        global_best_fitness =
current_global_best_fitness;
        for i=1:n
            global_best_position(:,i) =
local_best_position(:,g);
        end
    end

    % Setting bobot inersia
    velocity = (w2-((w2-
w1)*(iter/bird_step)))*velocity+
c1*(R1.*(local_best_position-current_position))
+ c2*(R2.*(global_best_position-
current_position));
    current_position = current_position +
velocity;

    fprintf('Nilai fitness iterasi %d \n', iter
);
    fitness=local_best_fitness;
    disp(current_global_best_fitness);
    zz=[zz current_global_best_fitness];
    zz1=[zz1 iter];
end % iterasi berakhir partikel berhenti
bergerak

%plot(zz1,zz);

```

```

plot(zz1,zz,'-
b');grid('on');xlabel('Iterasi');ylabel('Fitness
')
[Y,I] = min(fitness);
Kp=current_position(1,I);
Ki=current_position(2,I);
Kd=current_position(3,I);
fprintf('Kp = %d , Ki = %d , Kd = %d
\n',Kp,Ki,Kd)

```

Sub Program

```

function F = objf(pos)
% Track the output of optsim to a signal of 1
Kp=pos(1);
Ki=pos(2);
Kd=pos(3);
% simopt = simset('solver','ode5');
[~,~,yout] = sim('Simulator_HEV3',[0 10]);
sys_overshoot=max(yout)-1448; % compute the
overshoot
n=numel(yout);
%Compute objective function
IAE=e(n);
F=IAE*5+sys_overshoot*10;
end

```

Halaman ini sengaja dikosongkan

RIWAYAT PENULIS



Alif Ridwan Maulana merupakan nama lengkap pemberian dari kedua orang tua penulis. Lahir di Kediri pada tanggal 11 Juni 1992 dari pasangan Nuryono dan Eny Susilowati serta memiliki adik perempuan bernama Markona panggilannya. Untuk jenjang akademik penulis menempuh pendidikan SD di SDN Sukorame 1 Kediri selama 6 tahun dari tahun 1999 – 2005. Kemudian melanjutkan jenjang pendidikan di SMPN 1 Kediri selama 3 tahun. Pada tahun 2008 - 2011 penulis menjalani masa – masa SMA di SMAN 2 Kediri. Setelah itu penulis melanjutkan pendidikannya di Jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember (ITS) Surabaya pada tahun 2011. Selama masa perkuliahan penulis pernah aktif dibidang organisasi kemahasiswaan dengan bergabung BEM FTI ITS pada tahun 2012/2013. Serta pernah menjadi panitia SITIA pada tahun 2013. Pada tahun ke empat ini (2015), penulis mengikuti seminar dan sidang Tugas Akhir sebagai salah satu persyaratan untuk memperoleh gelar Sarjana Teknik Elektro.

Halaman ini sengaja dikosongkan