



UNIVERSITAS INDONESIA

**PERANCANGAN PENGENDALI PID PADA PROPORTIONAL
VALVE**

SKRIPSI

JEPRY

0606074041

**FAKULTAS TEKNIK UNIVERSITAS INDONESIA
TEKNIK ELEKTRO
DEPOK, JUNI 2010**



UNIVERSITAS INDONESIA

**PERANCANGAN PENGENDALI PID PADA PROPORTIONAL
VALVE**

SKRIPSI

Diajukan sebagai salah satu syarat memperoleh gelar sarjana

JEPRY

0606074041

**FAKULTAS TEKNIK UNIVERSITAS INDONESIA
TEKNIK ELEKTRO
DEPOK, JUNI 2010**

HALAMAN PERNYATAAN ORISINALITAS

Skripsi ini adalah hasil karya saya sendiri,
dan semua sumber baik yang dikutip maupun dirujuk
telah saya nyatakan dengan benar

Nama : Jepry

NPM : 0606074041

Tanda Tangan :



Tanggal : 15 Juni 2010

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :

Nama : Jepry

NPM : 0606074041

Program Studi : Teknik Elektro

Judul Skripsi : Perancangan Pengendali PID pada Proportional Valve

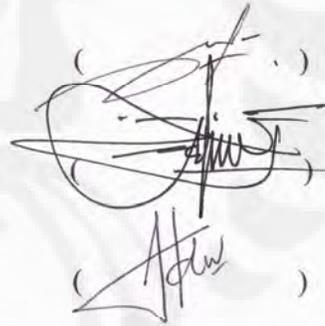
Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Elektro, Fakultas Teknik, Universitas Indonesia

DEWAN PENGUJI

Pembimbing : Dr. Ir Feri Yusivar, M.Eng

Penguji : Dr. Ir. Ridwan Gunawan MT

Penguji : Dr. Ir. Abdul Halim M.Eng



()
()
()

Ditetapkan di : Depok

Tanggal : 15 Juni 2010

UCAPAN TERIMA KASIH

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, saya dapat menyelesaikan skripsi ini. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Teknik Departemen Elektro pada Fakultas Teknik Universitas Indonesia. Saya menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan skripsi ini, sangatlah sulit bagi saya untuk menyelesaikan penulisan skripsi ini. Oleh karena itu, saya mengucapkan terima kasih kepada:

- (1) Tuhan Yang Maha Esa, yang telah memberi kekuatan dan kesempatan kepada saya untuk menyelesaikan penyusunan skripsi ini;
- (2) Dr. Feri Yusivar, ST, M.Eng, selaku dosen pembimbing yang telah menyediakan waktu, tenaga dan pikiran didalam mengarahkan saya dalam penyusunan skripsi ini;
- (3) Orangtua dan keluarga saya yang telah memberikan bantuan dukungan material maupun moril; dan
- (4) Cynthia, Suria, Marcel, Agny, Apriyudi, Deni, Chandra, Ferdi, dan sahabat lainnya yang telah banyak membantu penulis dalam menyelesaikan skripsi ini.

Akhir kata, saya berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu. Saya juga memohon maaf yang sebesar-besarnya bilamana ada kata-kata yang kurang berkenan selama penyusunan laporan skripsi ini. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu pengetahuan di Indonesia dan di dunia.

Depok, Juni 2010



Penulis

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS
AKHIR UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : Jepry
NPM : 0606074041
Program Studi : Teknik Elektro
Departemen : Teknik Elektro
Fakultas : Teknik
Jenis karya : Skripsi

demikian pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif (Non-exclusive Royalty-Free Right)** atas karya ilmiah saya yang berjudul :

Perancangan Pengendali PID pada Proportional Valve

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (database), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada tanggal : 15 Juni 2010

Yang menyatakan



Jepry

ABSTRAK

Nama : Jepry
Program Studi : Teknik Elektro
Judul : Perancangan Pengendali PID pada Proportional Valve

Kebutuhan manusia akan energi semakin meningkat dari waktu ke waktu seiring dengan pesatnya pertumbuhan penduduk. *Fuel cell* merupakan salah satu alternatif *renewable energy* yang sekarang banyak dikembangkan untuk memenuhi tuntutan kebutuhan manusia. *Fuel cell* memanfaatkan energi yang dihasilkan dari reaksi kimia antara hidrogen dan oksigen yang selanjutnya akan diubah menjadi energi listrik yang dapat digunakan sebagai sumber energi. Untuk mendapatkan reaksi kimia yang sesuai, dibutuhkan pengendali untuk mengatur *actuator valve fuel cell* sehingga keluaran gas hidrogen dapat bercampur dengan gas oksigen dalam komposisi yang sesuai dan energi yang dihasilkan optimal. Pada skripsi ini akan dibahas tentang perancangan pengendali PID untuk *proportional valve* pada *fuel cell* dengan menggunakan *DC geared motor* beserta tuning parameter PID dengan korelasi Ciancone. Tuning parameter terbaik dipilih untuk memberikan hasil yang baik pada sistem berupa perbaikan respon transien dan *steady state*. Pengendali ini akan digunakan sebagai sub-sistem dari sistem kendali *fuel cell*.

Kata Kunci: *DC geared motor*, perancangan pengendali PID, korelasi Ciancone

ABSTRACT

Name : Jepry
Study Program : Electrical Engineering
Title : PID Controller Design for Proportional Valve

Human need for energy increases from time to time along with the rapid population growth. Fuel cell is one of the renewable energy alternatives that have been developed nowadays to meet the demands of human needs. Fuel cells use energy produced from the chemical reaction between hydrogen and oxygen which is converted into electrical energy that can be used as an energy source. To obtain a suitable chemical reaction, the control is needed to manage the fuel cell valve actuator so that the output of hydrogen gas can be mixed with oxygen gas in the appropriate composition and optimal energy could be produced. This thesis will discuss about the design using PID control for the proportional valve control in the fuel cell with geared DC motor along with PID tuning parameter using Ciancone correlation. The best tuning parameter will be selected to improve transient and steady state response. This controller will be used as a sub-system of the fuel cell control system.

Keywords: DC geared motor, PID controller design, Ciancone correlation

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PERNYATAAN ORISINALITAS.....	ii
HALAMAN PENGESAHAN	iii
UCAPAN TERIMA KASIH	iv
LEMBAR PERSETUJUAN PUBLIKASI KARYA ILMIAH	v
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR GAMBAR	x
DAFTAR TABEL	xii
DAFTAR GRAFIK	xiii
1. PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Tujuan Penulisan	2
1.3 Pembatasan Masalah	2
1.4 Metode Penelitian.....	2
1.5 Sistematika Penulisan.....	3
2. DASAR TEORI	4
2.1 Motor DC.....	4
2.1.1 Konstruksi Motor DC	4
2.1.2 Rangkaian Ekuivalen Motor DC	6
2.1.3 Prinsip Kerja Motor DC	8
2.1.4 Prinsip Kerja Komutator pada Motor DC	9
2.1.5 DC Geared Motor	11
2.2 Potensiometer	13
2.3 Sistem Gear Box	14
2.4 Pengendali PID	17
2.5 PWM	22
3. PEMODELAN SISTEM DAN DESAIN PENGENDALI.....	24
3.1 Desain Flowchart.....	24

3.2	Pemodelan Sistem secara Empiris	25
3.2.1	Metode <i>Process Reaction Curve</i>	25
3.2.2	Data dan Pemodelan Sistem.....	28
3.3	Pengendali PID dan PD	29
3.4	Desain Perangkat Keras	31
3.4.1	Desain Rangkaian PIC Mikrokontroler.....	31
3.4.2	Desain Rangkaian Driver	32
3.4.3	Interkoneksi Sistem dalam Pengendali	33
3.5	Desain Perangkat Lunak	33
3.6	Tuning Parameter PID.....	36
3.6.1	Tuning Parameter PID untuk Mengendalikan Kecepatan Motor.....	36
3.6.2	Tuning Parameter PD untuk Mengendalikan Posisi Motor....	40
3.6.3	Tuning Parameter PD dan PID untuk Mengendalikan Posisi Motor.....	43
4.	SIMULASI, ANALISA DAN IMPLEMENTASI.....	47
4.1	Simulasi Open Loop	47
4.2	Simulasi Kecepatan PID untuk <i>DC Geared Motor</i>	49
4.2.1	Simulasi Pengendali PID Ideal untuk Kecepatan Motor.....	50
4.2.2	Simulasi Pengendali PID dengan Selisih CV untuk Kecepatan.....	51
4.2.3	Simulasi Perbaikan Parameter PID untuk Kecepatan.....	53
4.3	Simulasi Pengendali PD untuk Posisi <i>DC Geared Motor</i>	54
4.3.1	Simulasi Pengendali PD untuk Mengendalikan Posisi.....	54
4.3.2	Simulasi Pengendali PID dan PD untuk Kecepatan dan Posisi.....	56
4.4	Implementasi Sistem Pengendali.....	57
5.	KESIMPULAN.....	60
	DAFTAR REFERENSI.....	61
	LAMPIRAN.....	62

DAFTAR GAMBAR

Gambar 2.1. Motor DC.....	4
Gambar 2.2. Stator Motor DC.....	4
Gambar 2.3. Rotor Motor DC.....	5
Gambar 2.4. Komutator.....	5
Gambar 2.5. Brush dan Brush Holder.....	6
Gambar 2.6. Rangkaian Ekuivalen Motor DC	6
Gambar 2.7. Prinsip Kerja Motor DC	8
Gambar 2.8. Komutator Motor DC pada Posisi 0^0	9
Gambar 2.9. Komutator Motor DC pada Posisi 90^0	10
Gambar 2.10. Komutator Motor DC pada Posisi 180^0	10
Gambar 2.11. Komutator Motor DC pada Posisi 270^0	11
Gambar 2.12 DC Geared Motor	12
Gambar 2.13 Ukuran DC Geared Motor.....	12
Gambar 2.14. Karakteristik Beban DC Geared Motor	12
Gambar 2.15. Potensiometer	13
Gambar 2.16. Potensiometer sebagai Sensor Posisi	14
Gambar 2.17. Gear Box.....	15
Gambar 2.18. Susunan Gear.....	16
Gambar 2.19 Pengendali PID (a) Kontinu, (b) Ideal, (c) Dengan selisih CV.....	21
Gambar 2.20 Pembentukan PWM.....	22
Gambar 2.21 <i>Duty Cycle</i> PWM.....	23
Gambar 3.1 <i>DC Geared Motor</i> desain <i>Flowchart</i>	24
Gambar 3.2 <i>Process Reaction Curve</i> metode I.....	26
Gambar 3.3 <i>Process Reaction Curve</i> metode II.....	27
Gambar 3.4 Sistem Pengendali PD.....	30
Gambar 3.5 Pengendali Posisi dengan PD.....	30
Gambar 3.6 Pengendali Posisi dan Kecepatan dengan PD dan PID.....	30
Gambar 3.7 Skematik PIC.....	31
Gambar 3.8 Skematik <i>Driver</i> Pengendali	32
Gambar 3.9 Interkoneksi Perangkat Keras pada Sistem.....	33

Gambar 3.10 Flowchart Pengendalian dengan PIC 16F876.....	34
Gambar 3.11 Skema Pembangkitan Sinyal PWM.....	35
Gambar 3.12 Skema Pengendalian Posisi dengan <i>PD</i>	35
Gambar 3.13 Skema Pengendalian Posisi dan Kecepatan dengan <i>PD</i> dan <i>PID</i> ...	35
Gambar 3.14 Tuning Korelasi Ciancone	37
Gambar 3.15 Blok Diagram Pengendali PD untuk Posisi.....	40
Gambar 3.16 Dominan Pole Sistem Spesifikasi PD untuk Posisi.....	42
Gambar 3.17 <i>Angle of Departure</i> dan <i>Angle of Arrival</i> dengan PD.....	42
Gambar 3.18 Blok Diagram Pengendali PID untuk Kecepatan dan Posisi.....	44
Gambar 3.19 <i>Angle of Departure</i> dan <i>Angle of Arrival</i> dengan PD dan PID....	45
Gambar 4.1 Grafik (a)Posisi,(b)Kecepatan <i>DC Geared Motor</i> Loop Terbuka...	48
Gambar 4.2 Grafik Pengendali PID Ideal untuk Kecepatan.....	50
Gambar 4.3 Grafik Pengendali PID dengan Selisih CV untuk Kecepatan.....	52
Gambar 4.4 Grafik Kecepatan dengan Perbaikan Parameter PID.....	53
Gambar 4.5 Pengendali PID untuk Posisi <i>DC Geared Motor</i>	55
Gambar 4.6 Grafik Posisi dengan Pengendali PD dan PID.....	56

DAFTAR TABEL

Tabel 2.1 Karakteristik Parameter Pengendali PID.....	21
Tabel 3.1 Parameter PIC 16F876.....	32
Tabel 4.1 Nilai Tanggapan Waktu Kecepatan pada Percobaan Open Loop.....	48
Tabel 4.2 Nilai Tanggapan Waktu Kecepatan pada PID Ideal.....	51
Tabel 4.3 Nilai Tanggapan Waktu Kecepatan pada PID dengan Selisih CV.....	52
Tabel 4.4 Nilai Tanggapan Waktu Kecepatan dengan Perbaikan Parameter PID.....	53
Tabel 4.5 Nilai Tanggapan Waktu Posisi dengan Pengendali PD	55
Tabel 4.6 Nilai Tanggapan Waktu untuk Posisi dengan Memakai PD dan PID.....	56
Tabel 4.7 Nilai Tanggapan Waktu Pengendali PD untuk Posisi.....	58
Tabel 4.8 Nilai Tanggapan Waktu Pengendali PD dan PID untuk Posisi.....	59

DAFTAR GRAFIK

Grafik 3.1 Data Output Sistem Terhadap Input Step.....	28
Grafik 4.1 Gradien Tanggapan Posisi Sistem dan Simulasi.....	49
Grafik 4.2 Perubahan Output dengan Pengendali PD untuk Posisi.....	57
Grafik 4.3 Perubahan Output dengan Pengendali PD dan PID untuk Posisi.....	58

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Teknologi terus berkembang dengan pesat dari waktu ke waktu. Hampir semua teknologi yang kita gunakan menggunakan sumber energi baik berupa minyak bumi, gas alam, listrik maupun sumber daya lainnya. Sumber energi tersebut secara garis besar dapat dibagi menjadi 2 jenis yaitu sumber energi yang dapat diperbaharui (*renewable energy*) dan sumber energi yang tidak dapat diperbaharui (*unrenewable energy*).

Jumlah penduduk dunia terus bertambah dari tahun ke tahun. Pertumbuhan jumlah penduduk yang pesat ini berimplikasi pada konsumsi energi yang semakin besar pula. Akan tetapi, sumber energi yang kita miliki sekarang mayoritas masih berasal dari *unrenewable energy* yang cepat atau lambat akan segera habis. Oleh karena itu, diperlukan suatu langkah preventif untuk mengembangkan alternatif baru untuk memenuhi kebutuhan energi dari semua penduduk di dunia terutama di bidang *renewable energy*.

Salah satu pengembangan bidang *renewable energy* yang memiliki prospek yang cerah ke depannya adalah *fuel cell*. *Fuel cell* merupakan salah satu sumber energi yang mampu mengubah energi kimia menjadi energi listrik. *Fuel cell* memanfaatkan energi yang dihasilkan dari reaksi kimia antara hidrogen dan oksigen yang selanjutnya akan diubah menjadi energi listrik yang dapat digunakan sebagai sumber energi. Untuk mendapatkan reaksi kimia yang sesuai, diperlukan adanya komposisi yang seimbang antara gas hidrogen dan oksigen. Oleh karena itu, diperlukan adanya suatu pengendali yang mampu mengatur *actuator valve* dari gas hidrogen sehingga keluaran gas hidrogen dapat bercampur dengan gas oksigen dalam komposisi yang sesuai dan energi yang dihasilkan dari reaksi kimia tersebut dapat di konversi menjadi energi listrik secara efisien.

Pada skripsi ini akan dibahas tentang perancangan pengendali untuk *proportional valve* pada *fuel cell* dengan menggunakan pengendali PID (Proportional, Integral dan Derivative) untuk mengendalikan *geared DC*

motor beserta tuning parameter PID dengan korelasi Ciancone. Tuning parameter terbaik dipilih untuk memberikan hasil yang baik pada sistem berupa perbaikan respon transien dan *steady state*. Pengendali ini akan digunakan sebagai sub-sistem dari sistem kendali *fuel cell*.

1.2 Tujuan Penulisan

Tujuan dari skripsi ini adalah sebagai berikut:

1. Membahas dan menganalisa desain pengendali PD dan PID dengan menggunakan *geared DC motor* melalui simulasi.
2. Membahas dan menganalisa perangkat lunak maupun perangkat keras dari sistem pengendali yang digunakan serta pemrograman pengendali PD dan PID menggunakan PIC 16F876.

1.3 Pembatasan Masalah

Pembahasan skripsi akan dibatasi pada:

1. Pembahasan pengendali PD dan PID pada *DC geared motor*.
2. Perancangan perangkat lunak dan perangkat keras dari sistem pengendali yang digunakan dan pemrograman PD dan PID menggunakan PIC 16F876.
3. Interkoneksi dalam sistem pengendali *DC geared motor*
4. Analisa simulasi dan implementasi pengendali pada *DC geared motor*.

1.4 Metode Penelitian

Metode penelitian yang digunakan pada pekerjaan ini meliputi:

1. Tinjauan pustaka melalui studi literatur dari buku - buku pustaka, *datasheet*, atau *manual book* serta *reference book* dari suatu piranti.
2. Pendekatan diskusi dengan pembimbing skripsi yang berkaitan dengan topik bahasan skripsi.
3. Perancangan perangkat keras dan perangkat lunak dari sistem pengendali proporsional pada *valve fuel cell*.

1.5 Sistematika Penulisan

Untuk mempermudah penulisan dan agar pembahasan yang disajikan lebih sistematis, maka skripsi ini dibagi ke dalam lima bab, yaitu pendahuluan, landasan teori, desain pengendali, simulasi dan implementasi, dan kesimpulan.

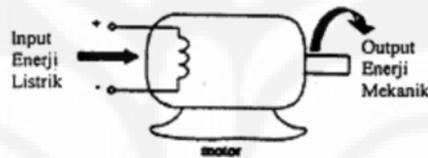
Bab pertama menjelaskan tentang latar belakang masalah, tujuan, batasan masalah, metode penulisan, dan sistematika penulisan skripsi ini. Bab kedua menjelaskan dasar teori dari *fuel cell*, *geared DC motor*, *potensiometer*, sistem *gear box*, PWM, dan pengendali PID. Bab ketiga mencakup tentang pemodelan motor DC dan desain dari sistem pengendali pada *proportional valve fuel cell* yang dibuat. Penjabaran meliputi perangkat lunak dan perangkat keras dari sistem pengendali. Bab keempat akan membahas hasil yang didapat ketika sistem yang didesain pada bab ketiga disimulasikan dan diimplementasikan ke sistem riil. Hasil yang didapat kemudian dianalisa di bab ini. Bab kelima berisi poin-poin penting dari hasil yang didapatkan dari desain sistem, simulasi, dan implementasi riil.

BAB 2

DASAR TEORI

2.1. Motor DC

Motor DC merupakan salah satu tipe aktuator yang sangat sering digunakan dalam implementasi suatu sistem pengendali. Motor DC merubah energi listrik menjadi energi mekanik, baik dalam bentuk pergerakan memutar maupun pergerakan lurus.



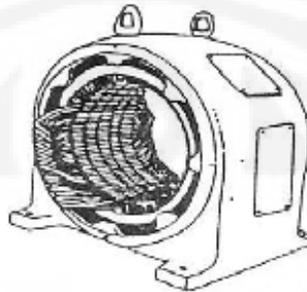
Gambar 2.1 Motor DC

2.1.1 Konstruksi Motor DC

Konstruksi dari motor DC secara umum terdiri dari:

1. Stator

Merupakan bagian yang diam pada motor DC. Pada bagian stator ini diberi catu DC pada kumparan medan(field windings) sehingga muncul medan magnet konstan. Oleh karena itu, stator pada mesin DC disebut sebagai penghasil medan magnet utama.



Gambar 2.2 Stator Motor DC

2. Rotor/jangkar motor DC

Merupakan bagian yang bergerak pada motor DC. Rotor terdiri dari inti besi yang dililitkan kumparan. Kumparan pada rotor ini disebut sebagai kumparan jangkar dimana GGL Induksi dihasilkan.



Gambar 2.3 Rotor Motor DC

3. Air gap/celah udara

Merupakan suatu celah udara yang memisahkan antara rotor dan stator. Adanya celah udara ini penting supaya motor dapat berputar. Tanpa celah udara ini, rotor dan stator akan bersinggungan sehingga perputaran tidak dapat dihasilkan.

4. Komutator

Komutator terdiri dari batangan tembaga yang dikeraskan yang diisolasi dengan sejenis mika. Fungsi komutator ini adalah mengumpulkan arus induksi dari konduktor jangkar dan mengkonversinya menjadi arus searah melalui sikat.

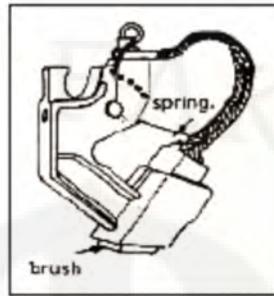


Gambar 2.4 Komutator

5. Sikat (*Brush*)

Sikat (*brushes*) pada umumnya terletak di sisi komutator untuk menyuplai listrik ke motor. Pada umumnya brush terbuat dari karbon, logam *graphite* atau campuran karbon dan grafit yang dilengkapi dengan pegas penekan dan kontak sikatnya.

Universitas Indonesia

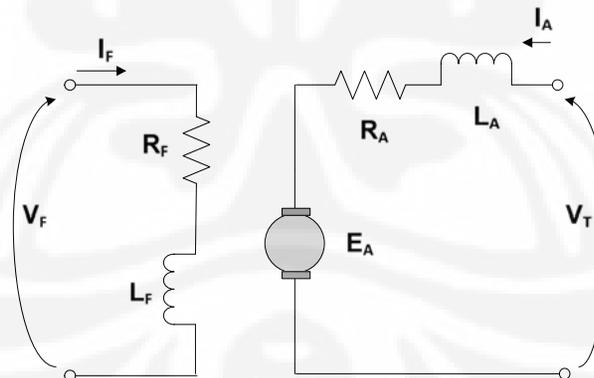


Gambar 2.5 *Brush dan Brush Holder*

Pegas berfungsi sebagai penekan sikat agar terjadi kontak dengan permukaan segmen komutator untuk menyalurkan arus listrik. Jadi, *brush* selalu tertekan ke *segment*. Hal ini berguna untuk menjamin koneksi elektrik yang bagus. Karbon dari *brush* diusahakan memiliki konduktivitas yang tinggi untuk mengurangi rugi-rugi listrik, dan koefisien gesekan yang rendah untuk mengurangi keausan (*excessive wear*).

2.1.2 Rangkaian Ekuivalen Motor DC

Gambar rangkaian ekuivalen motor DC ialah sebagai berikut:



Gambar 2.6 Rangkaian Ekuivalen Motor DC

dimana: I_F = arus medan

V_F = tegangan medan

R_F = resistansi medan

L_F = induktansi medan

I_A = arus jangkar

E_A = back emf

V_T = tegangan terminal

R_A = resistansi jangkar

Pada motor DC baik stator maupun rotornya dicatu sumber DC. Besarnya sumber DC yang dicatu pada stator dapat diubah-ubah dengan menggunakan resistor variabel.

Dengan menggunakan analisa Mesh pada rangkaian rotor maka dapat diperoleh persamaan [1]:

$$E_A = V_T - I_A R_A - I_A L_A \quad (2.1)$$

Nilai L_A pada persamaan diatas memiliki nilai yang sangat kecil apabila dibandingkan dengan nilai R_A sehingga yang diperhitungkan pada rangkaian rotor hanya nilai hambatan riil (R_A) saja sehingga diperoleh persamaan [1]:

$$E_A = V_T - I_A R_A \quad (2.2)$$

Nilai dari E juga dapat dicari dari persamaan:

$$E = kn\phi \quad (2.3)$$

dimana: k = konstanta persamaan motor

n = kecepatan dalam rpm

Φ = fluks medan

Maka didapatkan persamaan:

$$n = \frac{V_T - I_A R_A}{k\phi} \quad (2.4)$$

Torsi elektromagnetik dari motor DC adalah:

$$T = k\phi I_A \quad (2.5)$$

Daya masukan (P_{in}), daya keluaran (P_{out}), dan efisiensi () dari motor DC adalah sebagai berikut:

$$P_{in} = V_F I_F + V_T I_A \quad (2.6)$$

$$P_{out} = \frac{1000}{975} nT \quad (2.7)$$

$$\eta = \frac{P_{out}}{P_{in}} 100\% \quad (2.8)$$

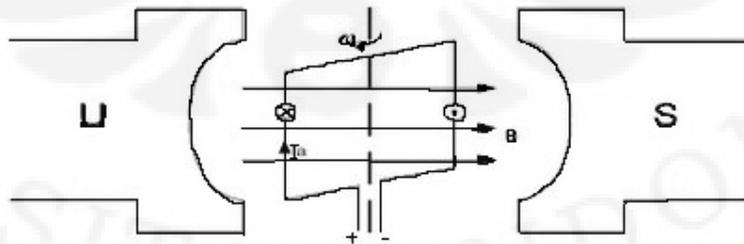
Motor DC yang digunakan pada skripsi ini merupakan jenis motor DC seri. Motor jenis ini memiliki kecepatan yang tinggi dan torsi yang rendah pada beban yang ringan. Kenaikan torsi yang cepat akibat kenaikan beban akan mengakibatkan penguatan fluks pada rangkaian medan dan rangkaian jangkar yang terjadi secara bersamaan karena kedua rangkaian tersebut terhubung secara seri dengan nilai arus yang hampir sama pada kedua rangkaian tersebut. Motor DC seri mampu memberikan perubahan kecepatan yang cukup tinggi dengan arus yang kecil sehingga motor jenis ini banyak digunakan untuk sistem dengan perubahan kecepatan yang bervariasi.

2.1.3 Prinsip Kerja Motor DC

Prinsip Kerja dari Motor DC adalah sebagai berikut:

1. Kumparan stator dicatu dengan sumber DC. Dikarenakan rangkaian stator merupakan rangkaian tertutup maka akan timbul arus I_F pada rangkaian stator. Ketika melewati konduktor, arus akan menimbulkan medan magnet B yang bernilai sebesar $B = \frac{\mu NI}{l}$ dimana stator merupakan penghasil B utama.
2. Rangkaian rotor dihubungkan dengan sumber DC pula. Rangkaian rotor juga merupakan rangkaian tertutup sehingga bisa dialiri arus listrik yang akan menimbulkan medan magnet konstan akibat sumber DC tadi.
3. Adanya arus dan medan magnet yang terjadi pada panjang kawat tertentu akan menimbulkan gaya Loorentz yang besarnya:

$$F = I.(l \times B) \quad (2.9)$$



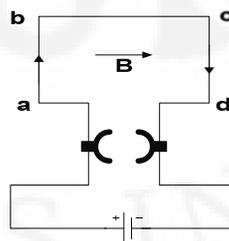
Gambar 2.7 Prinsip Kerja Motor DC

4. Arah gaya Lorentz yang dihasilkan dapat ditentukan dengan kaidah tangan kanan Fleming. Pada sisi kiri rotor arus mengarah ke dalam dan medan magnet yang terjadi arahnya ke kanan sehingga gaya Lorentz yang dihasilkan mengarah ke bawah. Pada sisi kanan rotor, arus mengalir keluar dan B mengarah ke kanan sehingga gaya Lorentz yang dihasilkan mengarah ke atas.
5. Gaya-gaya Lorentz yang bekerja pada jarak tertentu menyebabkan timbulnya torsi sesuai dengan persamaan $\tau = F \times r$ sehingga rotor mulai bergerak.
6. Pada saat perputaran rotor mencapai 90° , rotor berhenti berputar karena nilai torsinya nol yang disebabkan oleh r dan F memiliki arah yang sama sehingga $\tau = Fr \sin \theta = 0$, akan tetapi gayanya tetap ada. Karena masih terdapat torsi sebelumnya, rotor masih tetap berputar sehingga posisi sudut 90° dapat dilewati sebentar saja. Hal ini akan menjadi masalah apabila motor starting awal dengan sudut 90° sehingga perlu digerakkan secara manual atau menggunakan alat bantuan lain (*premotor*).

2.1.4. Prinsip Kerja Komutator pada Motor DC

Komutator diperlukan pada motor DC agar rotor pada motor DC dapat berputar 360° . Komutator terdiri dari 2 bagian besar yaitu cincin belah dan *brush*.

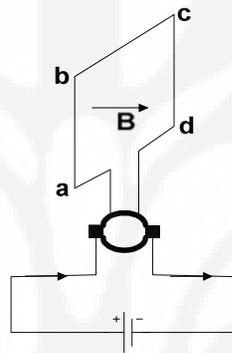
Bagian cincin belah pada komutator dihubungkan melingkari ujung batang rotor sehingga cincin belah akan ikut berputar seiring perputaran rotor. Brush dihubungkan dengan kabel keluar. Oleh karena itu, *brush* tidak ikut berputar akan tetapi terkikis akibat perputaran cincin belah. *Brush* yang digunakan terbuat dari karbon/grafit dikarenakan koefisien gesek karbon kecil.



Gambar 2.8 Komutator Motor DC pada Posisi 0°

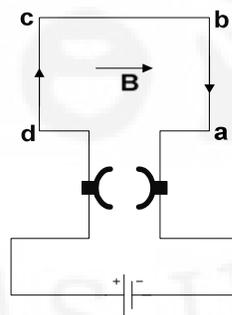
Universitas Indonesia

Pada gambar 2.8, arus mengalir melalui *brush*, lalu ke komutator dan menuju ke kumparan jangkar dan akibatnya akan timbul gaya Loorentz akibat interaksi arus jangkar dan B dari stator. Arus mengalir dari a ke b dengan arah masuk ke dalam sehingga gaya Loorentz akan ke arah bawah. Begitu pula pada arah arus c ke d akan ke atas arah gaya Loorentznya sesuai dengan prinsip kerja motor DC. Adanya gaya Loorentz yang bekerja pada jarak tertentu dari pusat sumbu putar menimbulkan torsi yang selanjutnya akan membuat rotor bergerak ke arah *counter clockwise*.



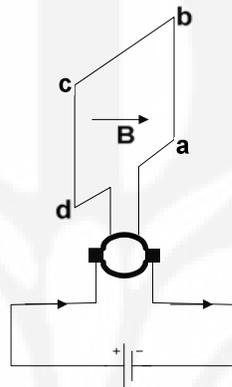
Gambar 2.9 Komutator Motor DC pada Posisi 90^0

Setelah perputaran rotor sebesar 90^0 , posisi cincin belah akan seperti pada gambar 2.9. Pada gambar 2.9 terlihat bahwa cincin belah membentuk hubungan pendek/*short* dengan *brush*. Meskipun begitu, arah arus tetap sama dengan adanya komutator sehingga rotor tetap berputar karena energi rotasi yang ada sebelumnya. Apabila komutator tidak ada maka rotor akan bergerak bolak-balik 180^0 dan -180^0 sehingga tidak akan mencapai 1 putaran penuh (pada kondisi 90^0 tanpa komutator ini nilai torsinya nol walaupun gaya Loorentz tetap ada).



Gambar 2.10 Komutator Motor DC pada Posisi 180^0

Pada gambar 2.10, rotor sudah berputar sebesar 180° sehingga posisi titik a, b, c dan d berubah menjadi kebalikan dari posisi a. Namun, adanya komutator menyebabkan arus tetap mengalir pada arah yang sama sehingga gaya Lorentz ada torsi yang bekerja pada kondisi gambar 2.10 memiliki arah yang sama dengan gambar 2.8. Akibatnya torsi yang terjadi memiliki arah yang sama dengan kondisi gambar 2.8 sehingga rotor akan kembali berputar *counter-clockwise*.



Gambar 2.11 Komutator Motor DC pada Posisi 270°

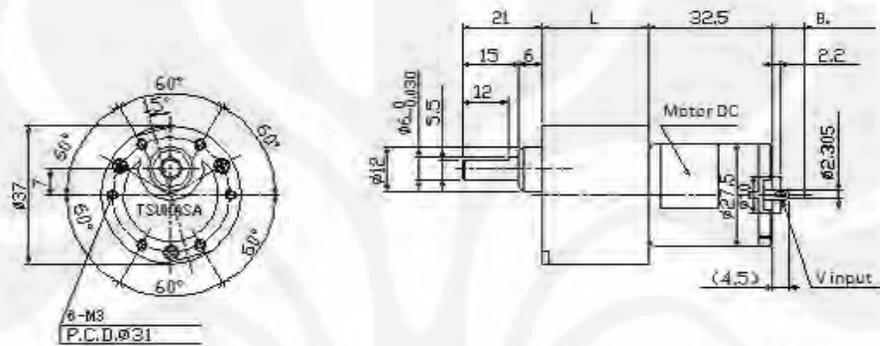
Pada gambar 2.11, posisi komutator mirip seperti pada gambar 2.9, namun posisi a, b, c, dan d telah berubah terbalik dari arah pada kondisi 2.9. Sama seperti pada kondisi 2.9, pada kondisi 2.11 rotor masih dapat berputar karena energi rotasi yang ada sebelumnya. Dengan demikian rotor telah berputar 360° dengan adanya komutator yang dihubungkan pada rotor di motor DC.

2.1.5 DC Geared Motor

DC geared motor merupakan motor DC memiliki sumber tegangan input DC dan kecepatan putar motornya diperlambat dengan menggunakan *gear* yang disusun dengan perbandingan tertentu. Pada skripsi ini digunakan *DC geared motor* tipe TG-47 yang merupakan motor DC yang cukup umum digunakan untuk peralatan rumah tangga. Motor ini memiliki tegangan input maksimum 24 V DC. Motor ini memiliki daya output mendekati 2 watt dan mampu menghasilkan kecepatan sampai 71,7 rpm setelah motor ini diperlambat dengan susunan *gear* tertentu. *DC Geared Motor* ini memiliki *reduction ratio* 1:60 / motor ini diperlambat 60 kali kecepatan ratingnya. Pada skripsi ini *DC geared motor* akan digunakan sebagai aktuator untuk mengendalikan perputaran *valve* dari *fuel cell*.

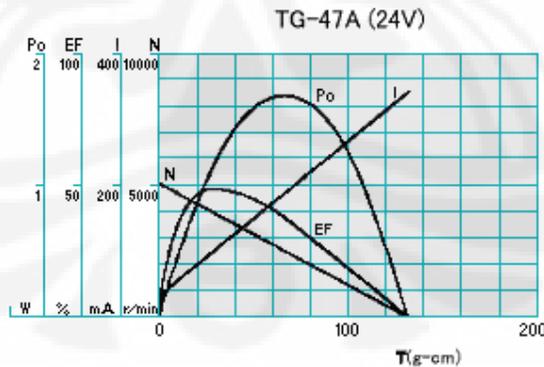


Gambar 2.12 DC Geared Motor



Gambar 2.13 Ukuran dari DC Geared Motor

Berikut ini adalah gambar dari karakteristik pembebanan terhadap daya(P_0), efisiensi(EF), arus(I), dan kecepatan dalam rpm(N) pada DC geared motor:



Gambar 2.14 Karakteristik Beban DC Geared Motor

Pada sistem DC geared motor digunakan beban yang rendah (sekitar 20%) dengan beban berupa sistem gear, proportional valve, dan potensiometer sehingga

efisiensi dari motor mendekati nilai maksimumnya sebesar 50% dan kecepatan motor mendekati 66,7 rpm.

2.2 Potensiometer

Potensiometer(disingkat pot) merupakan resistor tiga terminal dengan *sliding contact* yang berfungsi sebagai voltage divider yang bisa diatur. Apabila hanya 2 terminal saja yang digunakan maka potensiometer berfungsi sebagai resistor variabel/rheostat. Potensiometer pada umumnya digunakan sebagai pengontrol peralatan listrik seperti pengontrol volume pada perangkat audio. Pada skripsi ini potensiometer akan digunakan sebagai sensor posisi untuk *DC geared motor*.

Sebuah potensiometer terdiri dari elemen resistif dengan *sliding contact* (*wiper*). Elemen resistif dari potensiometer, terdiri dari terminal pada satu atau kedua ujung potensiometer, pada umumnya terbuat dari grafit. *Wiper* terhubung melalui *sliding contact* lain menuju ke terminal lain. Pada panel potensiometer, *wiper* biasanya terletak pada bagian tengah dari ketiga terminal potensiometer.

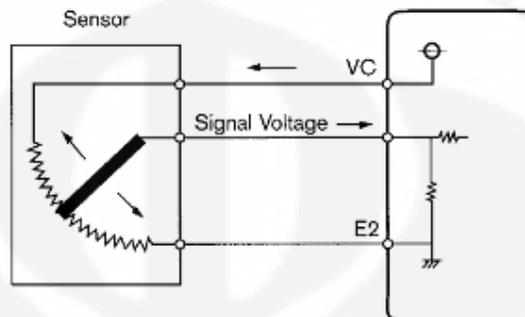


Gambar 2.15 Potensiometer

Terminal *wiper* yang terletak pada sensor secara mekanik terhubung ke bagian yang berputar seperti *valve*(katup) pada *fuel cell*. Ketika *valve* berputar, *wiper* juga mengalami pergeseran. Terminal *wiper* juga terhubung dengan resistor. Ketika *wiper* mengalami pergeseran, terjadi perubahan resistansi yang membuat sinyal tegangan output mengalami perubahan. Pada *contact* yang tersambung

Universitas Indonesia

dengan potensiometer, tegangan yang terbaca merupakan sinyal tegangan yang menunjukkan posisi. Potensiometer yang digunakan dalam skripsi ini merupakan potensiometer linear dimana tegangan dan hambatan yang dibaca pada sistem berubah secara linear (bukan secara eksponensial).

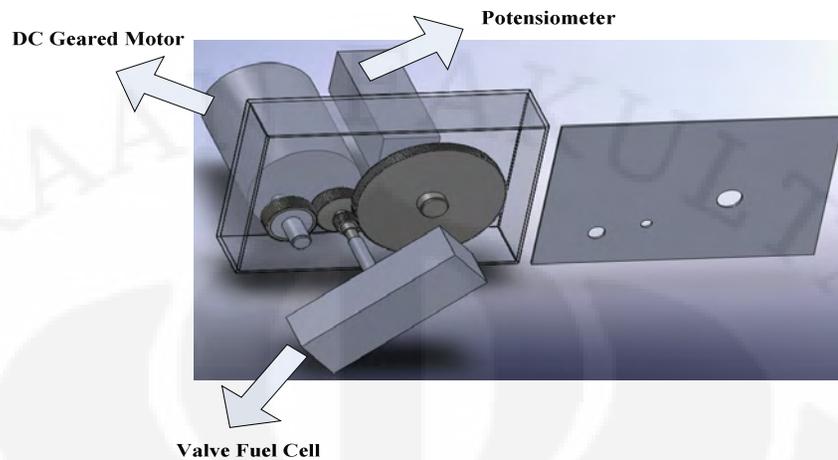


Gambar 2.16 Potensiometer sebagai Sensor Posisi

2.3 Sistem Gear Box

Salah satu cara untuk mengendalikan sistem *fuel cell* adalah dengan melakukan pengendalian terhadap input masukan gas hidrogen sehingga dengan tepat bereaksi dengan oksigen sehingga bisa menghasilkan output yang diinginkan. Untuk mengatur masuknya gas hidrogen ke dalam *fuel cell*, yang harus dikendalikan adalah proses membuka/menutupnya *valve* tempat masuknya gas hidrogen ke *fuel cell*. Oleh karena itu, perlu dirancang suatu sistem *gear box* yang menghubungkan *DC geared motor* sebagai aktuator yang akan memutar membuka/menutupnya *proportional valve* dari *fuel cell*.

Pada sistem *gear box* ini juga ditambahkan potensiometer yang berfungsi sebagai sensor untuk mendeteksi perubahan posisi *proportional valve* yang akan digunakan dalam perancangan pengendali PD dan PID pada *proportional valve fuel cell*. Potensiometer pada rangkaian ini kemudian dihubungkan ke rangkaian mikrokontroler PIC 16F876 dan digunakan sebagai umpan balik pada sistem pengendali PD dan PID. Susunan dari sistem *gear box* beserta *DC geared motor* dan *proportional valve* dapat dilihat dari gambar berikut:



Gambar 2.17 Gear Box

Pada gambar terlihat bahwa antara *DC geared motor*, valve, dan potensiometer dihubungkan oleh *gear-gear* dengan rasio perbandingan tertentu. *Gear ratio* merupakan hubungan antara jumlah roda gigi diantara 2 gear yang terhubung bersama. Secara sederhana, *gear ratio* dapat dinyatakan sebagai:

$$\text{gear ratio}(gr) = \frac{\text{(Jumlah gigi pada gear yang didorong)}}{\text{(Jumlah gigi pada pendorong)}} \quad (2.10)$$

Karena jumlah gigi juga sebanding dengan keliling roda gigi (semakin besar ukuran roda semakin banyak gigi), rasio gear juga dapat dinyatakan sebagai hubungan antara keliling roda (di mana d adalah diameter roda kecil dan D adalah diameter roda yang lebih besar):

$$gr = \frac{\pi d}{\pi D} = \frac{d}{D} \quad (2.11)$$

Karena diameter sama dengan dua kali jari-jari, maka:

$$gr = \frac{d}{D} = \frac{2r}{2R} = \frac{r}{R} \quad (2.12)$$

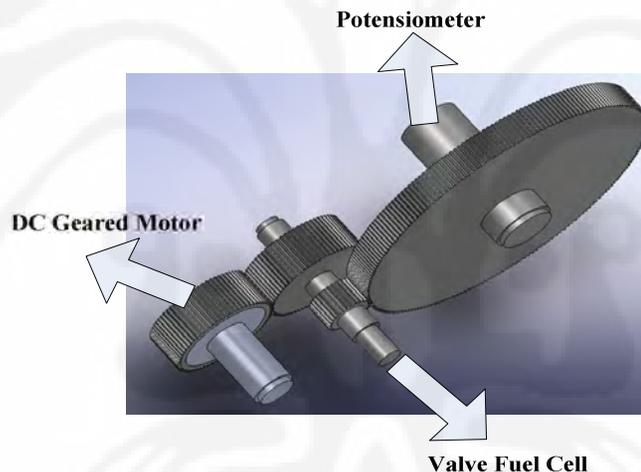
juga:

$$v_d = v_D \rightarrow \omega_d r = \omega_D R \rightarrow \frac{r}{R} = \frac{\omega_D}{\omega_d} \quad (2.13)$$

sehingga

$$gr = \frac{\omega_D}{\omega_d} \quad (2.14)$$

Dengan kata lain, *gear ratio* sebanding dengan rasio diameter dari *gear* dan berbanding terbalik dengan rasio kecepatan *gear*. Semakin banyak jumlah roda gigi dari *gear* akan mengurangi kecepatan perputaran *gear* dan demikian pula sebaliknya. Susunan dari *gear* pada sistem *DC geared motor* yang telah dirancang dan difabrikasi pada skripsi ini dapat dilihat pada gambar berikut:



Gambar 2.18 Susunan Gear

Pada gambar diatas terlihat bahwa perbandingan *gear* yang menghubungkan antara *DC geared motor* dan *valve fuel cell* adalah 1:1. Hal ini berarti bahwa kecepatan putar dari *DC geared motor* sama dengan berputarnya *valve fuel cell* karena kecepatan perputaran dari *DC geared motor* sudah cukup lambat. Sistem ini dirancang supaya apabila *valve fuel cell* telah membuka maksimum maka gear pada *DC geared motor* akan mengalami *loss* untuk menghindari kerusakan pada *DC geared motor* tersebut.

Pada poros/*shaft* dari *valve fuel cell* ditambahkan lagi sebuah gear yang lebih kecil yang dihubungkan dengan *gear* potensiometer dengan perbandingan 1:7 sehingga ukuran gear dapat diminimalisasi. Hal ini berarti bahwa ketika *valve fuel cell* telah berputar 7 kali maka potensiometer baru berputar 1 kali. Hal ini disebabkan bahwa *valve fuel cell* perlu berputar sebanyak 6,5 kali dari kondisi tertutup sampai terbuka penuh. Oleh karena itu, supaya potensiometer mampu mendeteksi posisi *valve* mulai dari 0% sampai 100% maka perlu dirancang gear dengan perbandingan 1:7.

2.4 Pengendali PID

Pengendali PID (*Proportional, Integral, Derivative*) merupakan suatu pengendali yang mampu memperbaiki tingkat akurasi dari suatu sistem *plant* yang memiliki karakteristik umpan balik/*feedback* pada sistem tersebut. Pengendali PID menghitung dan meminimalisasi nilai error/selisih antara output dari proses terhadap input/*setpoint* yang diberikan ke sistem.

Pengendali PID terdiri dari tiga komponen yaitu *proportional, integral, dan derivative* yang dapat dipakai secara bersamaan maupun sendiri-sendiri tergantung dari respon yang diinginkan pada suatu sistem/*plant*. Penjelasan dari masing-masing komponen adalah sebagai berikut:

1. Pengendali *Proportional*

Pengendali *proportional/gain* bertindak sebagai penguat yang mampu mengubah output dari sistem secara proporsional tanpa memberikan efek dinamik pada kinerja pengendali tersebut. Respon dari pengendali proporsional dapat dinyatakan dengan persamaan:

$$P_{out} = K_c E(t) \quad (2.15)$$

dimana: P_{out} = output dari pengendali proporsional

K_c = konstanta gain

$E(t)$ = error yang dinyatakan dalam waktu secara kontinu

Pengendali proporsional yang digunakan pada skripsi ini adalah dalam bentuk digital, sehingga rumus pengendalinya berubah menjadi:

$$P_{out} = K_c E_N \quad (2.16)$$

dimana: P_{out} = output dari pengendali proporsional

K_c = konstanta gain

E_N = error yang dinyatakan secara diskrit

Pengaturan dengan pengendali proporsional ini mampu memperbaiki respon transien dari sistem, khususnya *rise time*. Pengendali ini juga mampu memperbaiki *settling time* dari sistem.

2. Pengendali *Integral*

Pengendali *integral/reset* merupakan pengendali yang berfungsi untuk memperbaiki respon tunak/*steady state* dari sistem sehingga pengendali ini mampu memperkecil error sistem. Respon dari pengendali integral dapat dinyatakan dalam persamaan:

$$I_{out} = K_i \int_0^t E(\tau) d\tau \quad (2.17)$$

dimana: I_{out} = output pengendali integral

K_i = konstanta integral

$E(\tau)$ = error yang dinyatakan secara kontinu

τ = variabel integrasi

Dalam bentuk digital dari pengendali integral adalah sebagai berikut:

$$I_{out} = \frac{K_c \Delta t}{T_I} \sum_{i=1}^N E_i \quad (2.18)$$

dimana: I_{out} = output pengendali integral

T_I = waktu integral

Δt = periode sampling

E = error yang dinyatakan secara diskrit

$$K_I = K_c / T_I.$$

Dengan mengatur nilai dari konstanta integral yang tepat, nilai error *steady state* dapat diperkecil dalam waktu yang lebih cepat sehingga nilai output akan lebih cepat mengikuti nilai *set point*.

3. Pengendali *Derivative*

Pengendali derivative/rate merupakan suatu pengendali yang terutama berfungsi untuk memperbaiki respon transien dari sistem. Respon pengendali *derivative* dapat dinyatakan dalam persamaan berikut:

$$D_{out} = K_d \frac{d}{dt} E(t) \quad (2.19)$$

dimana: D_{out} = output pengendali derivative

K_d = konstanta derivative

$E(t)$ = error yang dinyatakan dalam waktu kontinu

Dalam bentuk digital pengendali derivative dinyatakan sebagai berikut:

$$D_{out} = \frac{K_c T_d}{\Delta t} (E_N - E_{N-1}) \quad (2.20)$$

dimana: T_d = waktu derivative

Δt = waktu sampling

E_N, E_{N-1} = error yang dinyatakan dalam waktu diskrit

$$K_d = K_c T_d.$$

Rumus derivative yang digunakan diatas memiliki 2 kelemahan, yaitu apabila terdapat perubahan besar pada setpoint akan mengakibatkan perubahan besar pada error dan turunan dari perubahan step dapat membesar tak terhingga sehingga dapat membahayakan suatu sistem/plant. Oleh karena itu, perubahan error pada derivative ini diubah menjadi perubahan *controlled variable* (CV) dimana $E_N = SP_N - CV_N$ sehingga:

$$D_{out} = -\frac{K_c T_d}{\Delta t} (CV_N - CV_{N-1}) \quad (2.21)$$

dimana: T = waktu derivative

Δt = waktu sampling

CV = controlled variable

$K_d = K_c T_d$.

Selain mampu memperbaiki respon transien, pengendali *derivative* juga mampu mengurangi nilai overshoot yang timbul akibat penggunaan pengendali integral.

Nilai output dari pengendali PID didapatkan berdasarkan penjumlahan dari masing-masing komponennya sebagai berikut:

$$MV(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t) \quad (2.22)$$

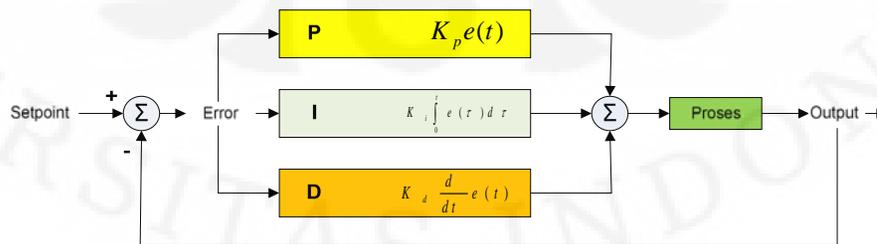
Bentuk pengendali PID digital dari rumus diatas adalah sebagai berikut:

$$MV_N = K_c \left[E_N + \frac{\Delta t}{T_i} \sum_{i=1}^N E_i + \frac{T_d}{\Delta t} (E_N - E_{N-1}) \right] \quad (2.23)$$

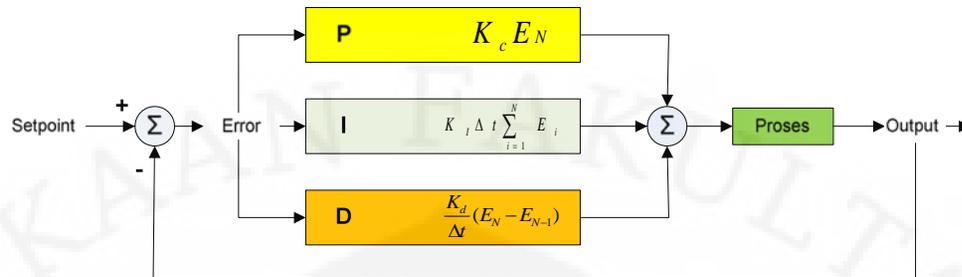
Untuk meminimalisasi error pada saat terjadi perubahan *setpoint* maka digunakan rumus pengendali PID sebagai berikut:

$$MV_N = K_c \left[E_N + \frac{\Delta t}{T_i} \sum_{i=1}^N E_i - \frac{T_d}{\Delta t} (CV_N - CV_{N-1}) \right] \quad (2.24)$$

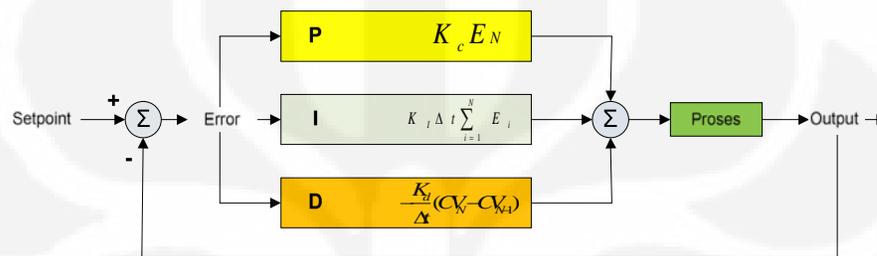
Gambar blok diagram dari pengendali PID diatas adalah sebagai berikut:



(a)



(b)



(c)

Gambar 2.19 Pengendali PID (a) Kontinu, (b) Ideal(diskrit) (c) Dengan selisih CV (diskrit)

Ringkasan dari karakteristik pengendali PID adalah sebagai berikut:

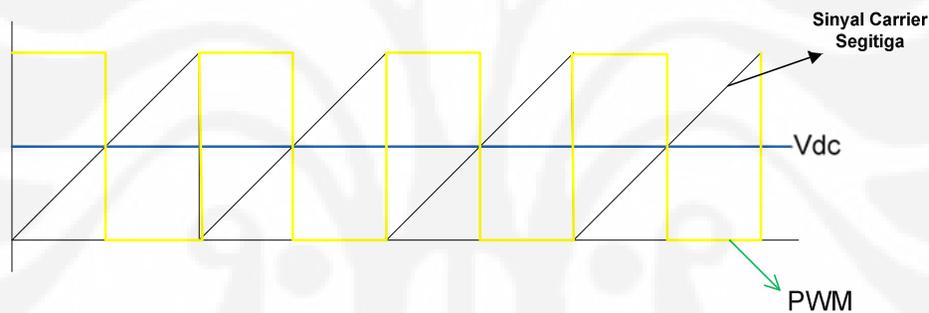
Tabel 2.1 Karakteristik Parameter Pengendali PID

	Rise Time	Overshoot	Settling Time	Error Steady State
Kp	Mengurangi	Menambah	Menimbulkan perubahan kecil	Mengurangi
Ki	Mengurangi	Menambah	Menambah	Meniadakan
Kd	Menimbulkan perubahan kecil	Mengurangi	Mengurangi	Menimbulkan perubahan kecil

2.5 PWM (Pulse Width Modulation)

Pada simulasi nilai tegangan yang masuk ke dalam motor DC merupakan sinyal keluaran PWM yang diberi masukan dari pengendali PID. Sinyal PWM ini akan menghasilkan pulsa yang lebarnya berbeda-beda tergantung dengan nilai masukannya.

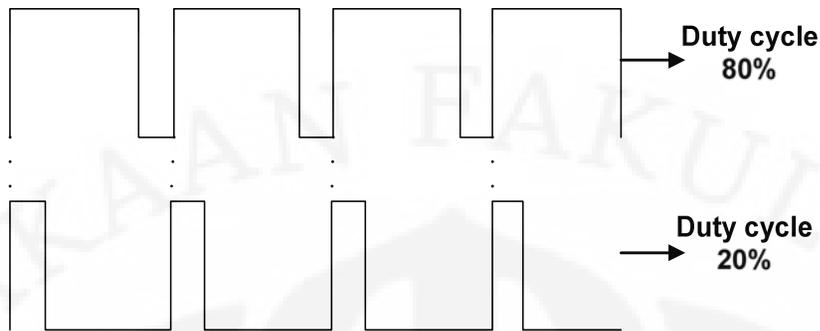
Proses pembentukan pulsa pada PWM merupakan perbandingan antara sinyal input DC dengan sinyal carriernya. Untuk setengah gelombang pertama, ketika sinyal masukan DC lebih besar dari sinyal carriernya, maka pulsa akan bernilai maksimum, sedangkan apabila sinyal masukan lebih kecil daripada sinyal carriernya, maka pulsa akan bernilai nol dan seterusnya.



Gambar 2.20 Pembentukan PWM

PWM dapat digunakan sebagai pengendali tegangan yang masuk ke motor dengan mengatur nilai *duty cycle* dari PWM tersebut. *Duty cycle* merupakan suatu perbandingan antara waktu nyala terhadap interval/periode waktu tertentu. *Duty cycle* yang rendah akan menghasilkan daya yang rendah, karena dayanya kebanyakan bernilai nol selama periode tertentu. *Duty cycle* bernilai 100% ketika sistem menyala terus selama periode tertentu. Pada skripsi ini, nilai *duty cycle* akan dikendalikan dengan menggunakan pengendali PID untuk mendapatkan hasil yang optimal.

Berikut ini merupakan contoh dari *duty cycle* pada keadaan *duty cycle* 80% dan *duty cycle* 20% untuk memperjelas gambaran tentang *duty cycle* pada PWM:



Gambar 2.21 *Duty Cycle* PWM

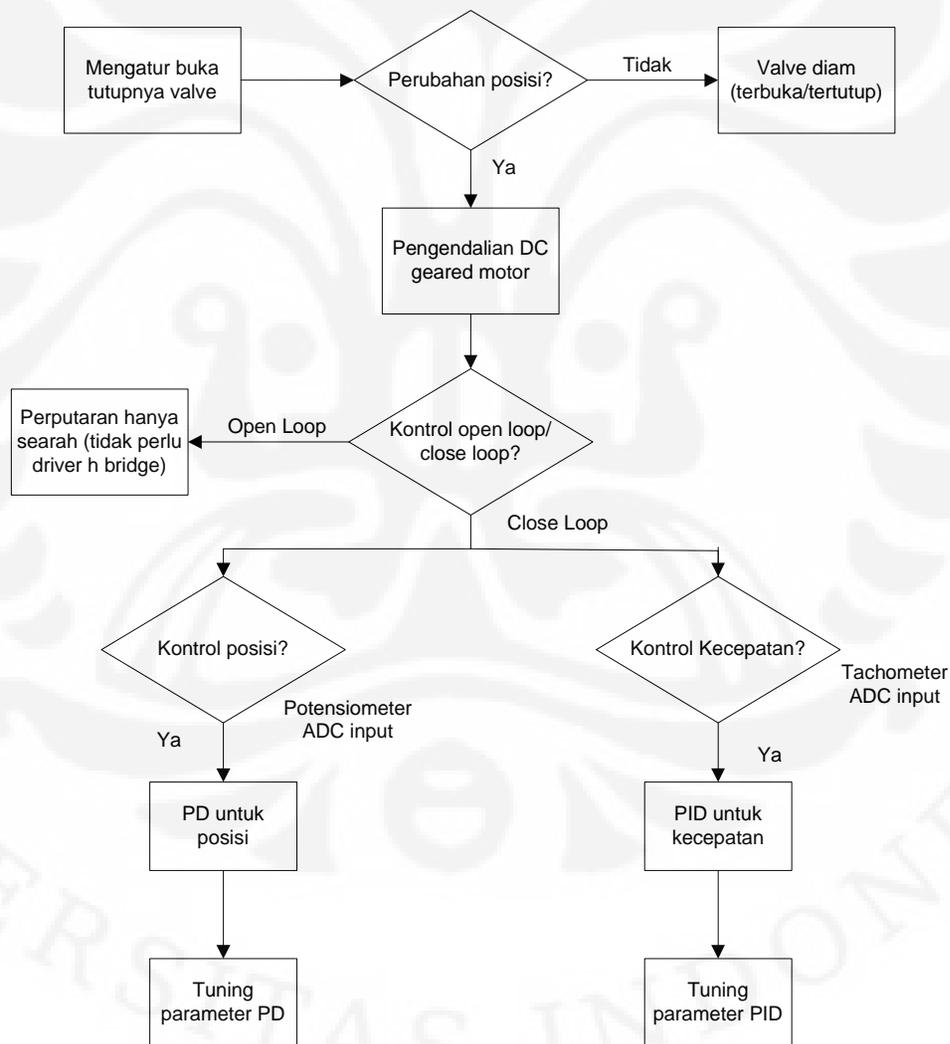
Duty cycle akan menentukan tegangan input rata-rata yang diberikan ke DC *geared motor*. Sebagai contoh, PWM dengan *duty cycle* 80% yang menggunakan tegangan maksimum 24 V akan menghasilkan tegangan rata-rata sebesar 19.2 V sedangkan dengan *duty cycle* 20%, tegangan rata-rata yang dihasilkan adalah sebesar 4.8 V. Dengan demikian, PWM merupakan cara yang sangat efektif untuk mengatur kecepatan perputaran DC *geared motor* tanpa harus mengubah tegangan input DC *geared motor* secara manual.

BAB 3

PEMODELAN SISTEM DAN DESAIN PENGENDALI

3.1 Desain Flowchart

Sebelum membuat sebuah pengendali untuk *DC geared motor*, dibuat desain *flowchart* untuk mengetahui mengenai gambaran umum sistem yang akan dikendalikan. Desain *flowchart* dari *DC geared motor* yang akan dibuat adalah sebagai berikut:



Gambar 3.1 *DC Geared Motor* Desain Flowchart

Berdasarkan desain *flowchart* diatas, akan dirancang sebuah pengendali PID dimana pengendali ini dapat mengendalikan perputaran motor dari 2 arah yaitu untuk mengatur membuka dan menutupnya *valve* fuel cell. Pengendalian motor dilakukan dalam dua cara, yaitu pengendalian motor untuk posisi dan juga untuk kecepatan. Pengendali PD digunakan untuk mengendalikan posisi *DC geared motor* untuk menghindari nilai *steady state* sistem cenderung terus membesar akibat penambahan integrator pada sistem. Hasil dari pengendalian ini kemudian akan dianalisa lebih lanjut untuk melihat kinerja dari pengendali

3.2 Pemodelan Sistem secara Empiris

Pemodelan yang dilakukan dalam skripsi ini adalah pemodelan secara empiris. Pemodelan ini dilakukan dengan melakukan pengambilan data dari sistem yang akan diukur untuk melihat karakteristik input dan output dari sistem yang telah dibuat. Hasil pengambilan data tersebut kemudian dimodelkan menjadi sistem berorde satu dengan dead time. Untuk menentukan parameter dari model yang telah dibuat tersebut, akan dilakukan perhitungan sederhana dari grafik (dengan menggunakan metode *process reaction curve*) yang dihasilkan selama pengambilan data. Model dinamis yang didapatkan ini kemudian akan digunakan sebagai model untuk menentukan parameter tuning pada PID sehingga didapatkan hasil pengendalian yang optimal.

3.2.1 Metode Process Reaction Curve

Metode ini merupakan metode yang umum digunakan untuk mendapatkan model secara empiris dan mampu menghasilkan model yang cukup akurat untuk aplikasi proses kendali. Terdapat 4 hal yang harus diperhatikan dalam pemodelan dinamis dengan menggunakan metode ini, yaitu:

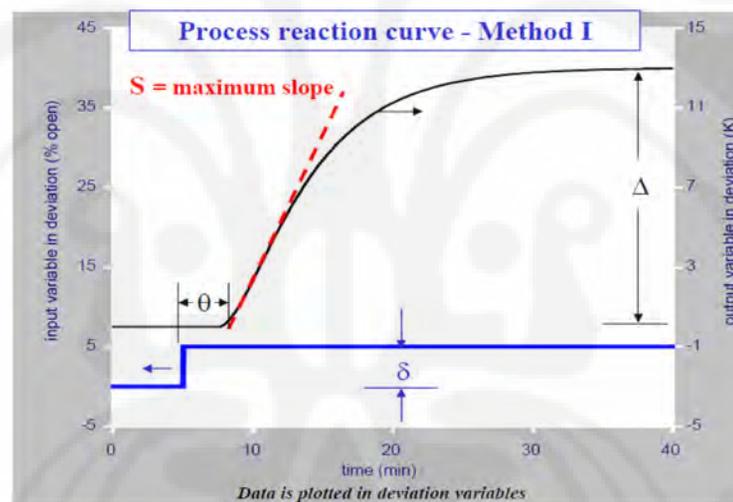
1. Biarkan proses mencapai keadaan *steady state*
2. Berikan masukan sinyal step sebagai variabel input
3. Kumpulkan respons data input dan output sampai proses mencapai keadaan *steady state*

4. Lakukan kalkulasi dari grafik *process reaction curve* yang telah didapat

Hasil dari pengkalkulasian grafik *process reaction curve* tersebut akan digunakan untuk menghitung parameter-parameter model sistem berorde satu dengan dead time. Bentuk dari sistem berorde satu dengan dead time adalah sebagai berikut:

$$G_p(s) = \frac{Y(s)}{X(s)} = \frac{K_p e^{-\theta s}}{\tau s + 1} \quad (3.1)$$

Terdapat 2 teknik dalam menganalisa hasil grafik yang didapatkan. Metode yang pertama diadaptasi dari metode Ziegler Nichols (1942). Metode ini dapat ditunjukkan oleh grafik dibawah ini:



Gambar 3.2 *Process Reaction Curve* Metode I

Nilai parameter dari metode ini dicari dengan menggunakan rumus:

$$K_p = \Delta / \delta \quad (3.2)$$

$$\tau = \Delta / S \quad (3.3)$$

Θ = pertemuan antara slope maksimum dengan nilai awal

Keterangan: K_p = gain

τ = time konstan

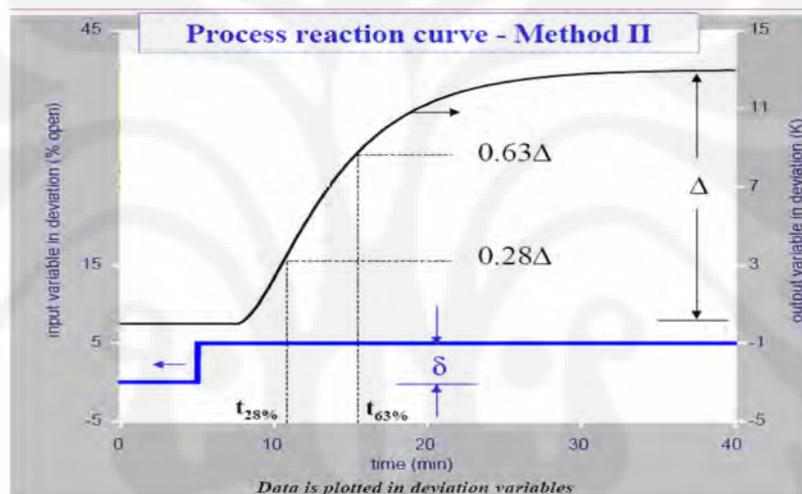
Θ = dead time

Δ = perubahan magnitude steady state pada output

δ = perubahan magnitude steady state pada input

S = kemiringan/slope maksimum pada kurva output vs plot.

Selain cara diatas, terdapat juga metode kedua yang ditunjukkan oleh gambar dibawah ini:



Gambar 3.3 *Process Reaction Curve* Metode II

Langkah identifikasi model dengan metode ini adalah sebagai berikut:

1. Mencari kurva response tanpa sistem pengendali yang dianggap paling ideal dengan mengatur besar input step
2. Menghitung parameter K_p (gain), θ (dead time), dan τ (time constant) yang didapatkan berdasarkan gambar 3.3 diatas,

$$\text{dimana: } K_p = \Delta / \delta \qquad t_{28\%} = \theta + \frac{\tau}{3} \qquad (3.4)$$

$$\tau = 1.5(t_{63\%} - t_{28\%}) \qquad t_{63\%} = \theta + \tau \qquad (3.5)$$

$$\theta = t_{63\%} - \tau \qquad (3.6)$$

Universitas Indonesia

Keterangan: Δ = perubahan magnitude steady state pada output

δ = perubahan magnitude steady state pada input

$t_{63\%}$ = waktu dibutuhkan untuk mencapai 63% dari nilai akhir

$t_{28\%}$ = waktu dibutuhkan untuk mencapai 28% dari nilai akhir
(dapat dilihat pada gambar respons keluaran)

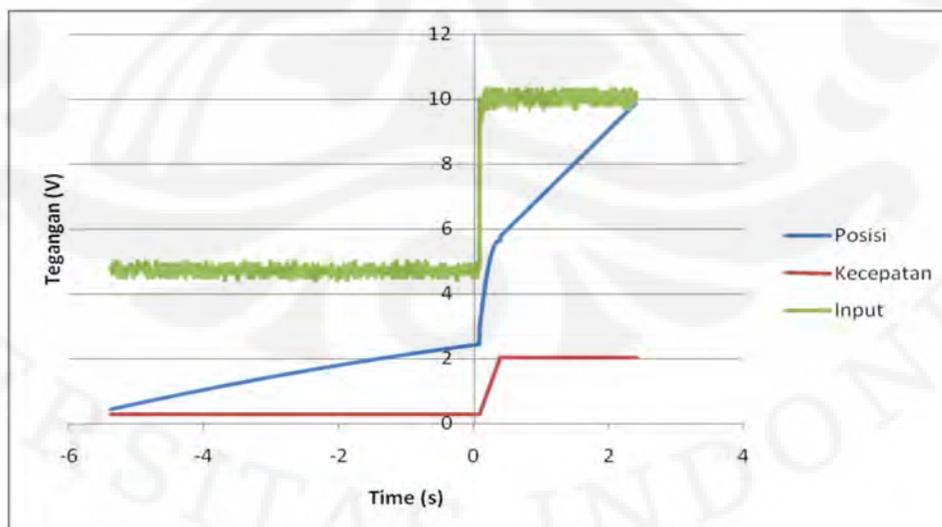
3. Memodelkan sistem $G_p(s)$ sebagai model orde satu dengan dead time sesuai persamaan 3.1:

$$G_p(s) = \frac{Y(s)}{X(s)} = \frac{K_p e^{-\theta s}}{\tau s + 1} \quad (3.7)$$

Di dalam penulisan skripsi ini digunakan metode yang kedua karena pada metode yang pertama terdapat kesulitan dalam menentukan besar slope terutama apabila signalnya memiliki frekuensi tinggi sehingga dapat mengakibatkan error estimasi parameter yang lebih besar.

3.2.2 Data dan Pemodelan Sistem

Berdasarkan pengambilan data yang telah dilakukan, grafik dari karakteristik output terhadap input adalah sebagai berikut:



Grafik 3.1 Data Output Sistem Terhadap Input Step

Pemodelan sistem akan dilakukan antara input tegangan step dan kecepatan sehingga didapatkan hasil sebagai berikut:

$$\Delta = 1.769$$

$$\delta = 5$$

$$K_p = \Delta / \delta = (1.769) / (5) = 0.3538$$

$$0.63\Delta = 1.11447 \quad t_{63\%} = 0.1098 \text{ s}$$

$$0.28\Delta = 0.49632 \quad t_{28\%} = 0.05356 \text{ s}$$

$$\tau = 1.5(t_{63\%} - t_{28\%}) = 1.5(0.1098 - 0.05356) = 1.5(0.05624) = 0.08436 \text{ s}$$

$$\Theta = t_{63\%} - \tau = 0.1098 - 0.08436 = 0.02524 \text{ s}$$

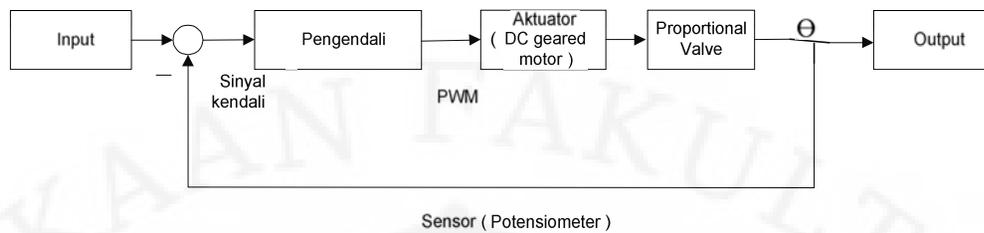
$$G_p(s) = \frac{\omega(s)}{u(s)} = \frac{0.3538 e^{-0.02524 s}}{0.08436 s + 1} \quad (3.8)$$

Jadi, persamaan sistem orde pertama dengan dead time dari *DC geared*

$$\text{motor adalah } G_p(s) = \frac{\omega(s)}{u(s)} = \frac{0.3538 e^{-0.02524 s}}{0.08436 s + 1}.$$

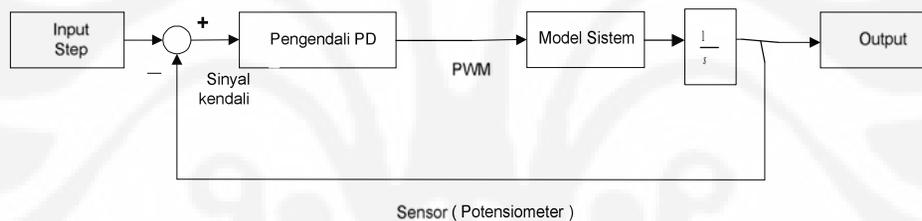
3.3 Pengendali PID dan PD

Perancangan pengendali yang akan digunakan adalah pengendali PID untuk mengendalikan kecepatan dan PD untuk mengendalikan posisi. Pengendali ini akan menentukan keluaran sinyal PWM yang akan diberikan kepada *DC geared motor* yang bertindak sebagai aktuator. Motor tersebut akan memutar *valve* dan datanya akan dibaca oleh sensor posisi yaitu potensiometer yang kemudian akan diumpan balik menjadi masukan sinyal kendali untuk memperbaiki error dari pengendalian motor.

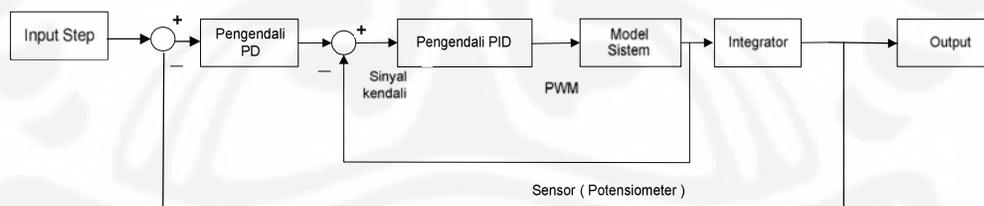


Gambar 3.4 Sistem Pengendali PD

Pengendali PID yang digunakan dalam skripsi ini dibagi menjadi 2 jenis, yaitu pengendali PD untuk posisi motor dan 2 buah pengendali, yaitu PID dan PD untuk mengendalikan kecepatan dan posisi motor. Pengendali PID dan PD digunakan untuk menghasilkan respons tanggapan waktu yang lebih baik pada sistem DC *geared motor* daripada pengendalian dengan PD saja. Skema dari sistem pengendalinya adalah berikut:



Gambar 3.5 Pengendali Posisi dengan PD



Gambar 3.6 Pengendali Posisi dan Kecepatan dengan PD dan PID

Algoritma dari pengendali yang digunakan adalah sebagai berikut:

Algoritma Pengendali
Mulai algoritma
1. Inisialisasi sistem pengendali dan PWM

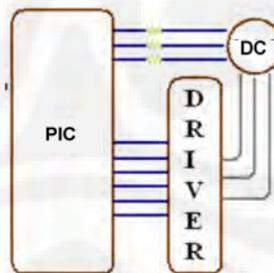
2. Pembacaan data dari input analog (potensiometer) dengan ADC
3. Data tersebut menjadi acuan pengendali untuk menentukan keluaran sinyal PWM untuk memutar motor DC.
4. Perputaran motor akan memutar *valve* dan dibaca oleh potensiometer
5. Hasil pembacaan akan diubah menjadi data digital dengan ADC dan diumpun balik ke pengendali
6. Data yang dihasilkan akan menjadi acuan pengendali untuk membuka/menutup *valve*
7. Atur parameter pengendali untuk mendapatkan respon yang diinginkan

3.4 Desain Perangkat Keras

Desain perangkat keras untuk mengendalikan *DC geared motor* dibagi menjadi 2 bagian, yaitu bagian PIC mikrokontroller dan bagian rangkaian *driver*.

3.4.1 Desain Rangkaian PIC Mikrokontroller

Pada sistem pengendali proporsional *valve*, PIC mikrokontroller memiliki tugas utama untuk menjalankan algoritma pengendali yang akan diterapkan ke dalam sistem pengendali. Skematik umum dari rangkaian PIC adalah sebagai berikut:



Gambar 3.7 Skematik PIC

PIC mikrokontroller yang akan digunakan untuk merancang pengendali PID adalah PIC 16F876. PIC ini memiliki 28 buah pin dan menawarkan semua keuntungan yang dikenal dari PIC mikrokontroller 8 bit dengan jumlah instruction set yang hanya berjumlah 35 buah. PIC ini memiliki operasi tegangan yang luas

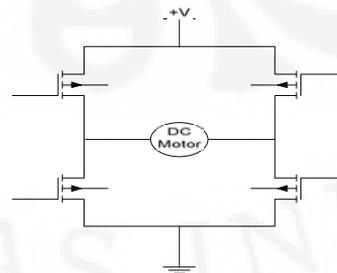
antara 2-5,5 volt, EEPROM *data memory on-board*, perangkat analog 5 channel 10 bit A/D, modul komparator analog dengan dua komparator referensi tegangan yang dapat di program onchip, dan *Enhanced Capture / Compare / PWM (ECCP)*. Berikut ini adalah tabel spesifikasi dari PIC 16F876:

Tabel 3.1 Parameter PIC 16F876

Parameter Name	Value
Program Memory Type	Flash
Operating Frequency	DC-20 MHz
Program Memory (KB)	8
CPU Speed (MIPS)	5
RAM Bytes	368x8
Data EEPROM (bytes)	256x8
Interrupts	13
I/O Ports	Ports A,B,C
Capture/Compare/PWM Peripherals	2 CCP
Timers	2 x 8-bit, 1 x 16-bit
ADC	5 ch, 10-bit
Serial Communications	MSSP, USART
Temperature Range ©	-40 to 85
Operating Voltage Range (V)	2 to 5.5
Instruction Set	35
Pin Count	28

3.4.2 Desain Rangkaian Driver

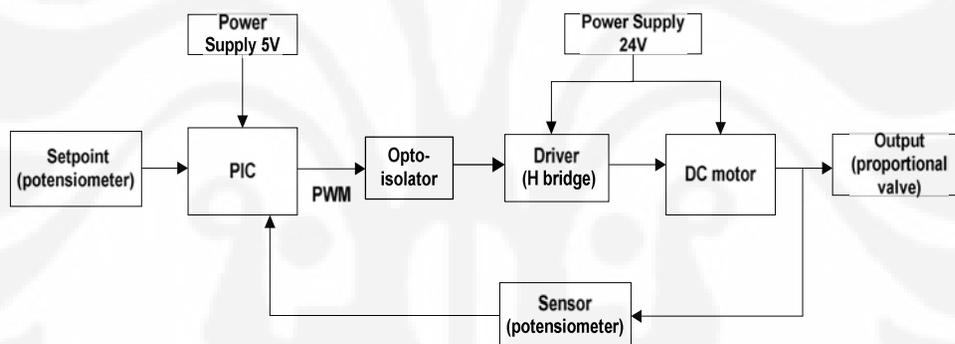
Rangkaian *driver* bertugas untuk menguatkan sinyal PWM dari mikrokontroler menjadi sinyal daya yang menggerakkan motor. Rangkaian driver terdiri dari 4 buah mosfet (rangkaian *H bridge*) yang berfungsi sebagai *switch*. Dari ke 4 mosfet tersebut, hanya 2 mosfet yang menyala pada satu waktu. Gambar skematik dari driver pengendali adalah sebagai berikut:

Gambar 3.8 Skematik *Driver* Pengendali

IC yang digunakan untuk rangkaian MOSFET diatas adalah IRF 9540 dan IRF 540. IRF9540 merupakan *p-channel* MOSFET yang mampu menahan tegangan V_{DS} hingga 100 V dan mampu menahan arus sampai -23 A. IRF 540 merupakan *n-channel* MOSFET yang mampu menahan tegangan V hingga 100 V dan mampu menahan arus sampai 22A. Keduanya memiliki nilai resistansi R_{DS} yang rendah (IRF 9540 sekitar 0.117Ω dan IRF 540 sekitar 0.055Ω) sehingga disipasi panasnya lebih rendah. Kedua jenis IC diatas digunakan sebagai penentu arah pergerakan moto DC (searah atau kebalikan arah jarum jam).

3.4.3 Interkoneksi Sistem dalam Pengendali

Berdasarkan desain dari mikrokontroller PIC dan *driver* yang telah dibuat diatas, sistem tersebut akan dihubungkan seperti gambar dibawah ini:



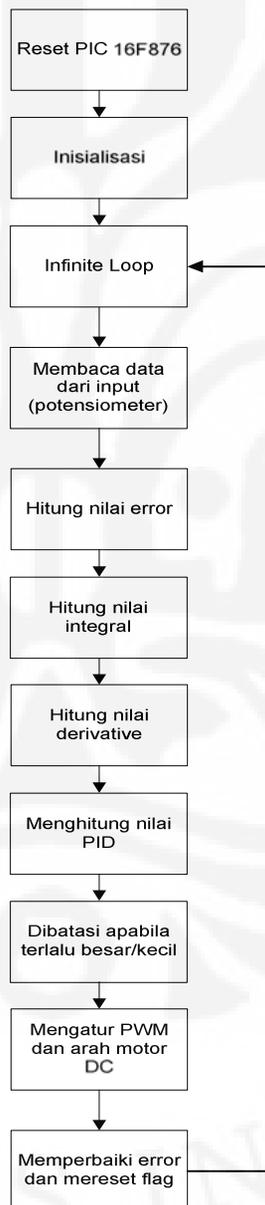
Gambar 3.9 Interkoneksi Perangkat Keras pada Sistem

3.5 Desain Perangkat Lunak

Program pengendali dibuat dengan menggunakan software MPLAB IDE v8.46 yang merupakan *design environment* yang dibuat oleh *Microchip.Inc* untuk pemograman dengan menggunakan PIC. Program yang dibuat menggunakan bahasa C sehingga digunakan compiler bahasa C berupa *HiTech PIC v9.70*.

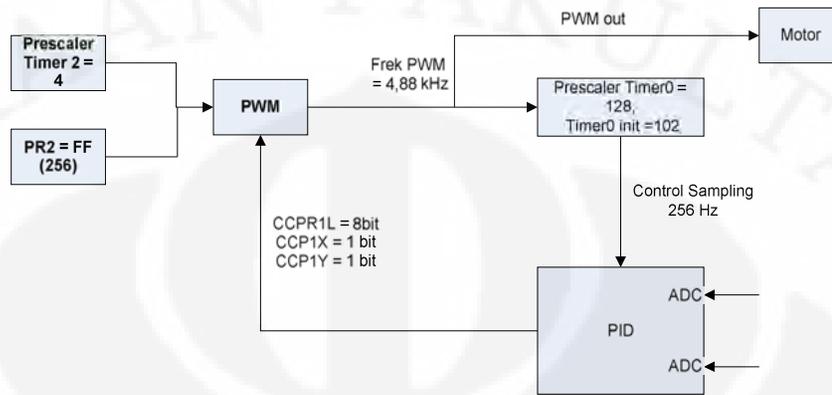
Algoritma pengendali yang telah dibuat kemudian diimplementasikan ke dalam PIC mikrokontroller 16F876. Sebelum pengendali dibuat, terlebih dahulu dilakukan penginisialisasian pada PIC berupa pengaturan port input yang aktif, frekuensi dari mikrokontroller dan timer yang digunakan. Setelah itu dilakukan inisialisasi frekuensi PWM sebesar 4.88 kHz (0.2094 ms) dan frekuensi sampling kendali beserta interrupt yang yang digunakan. Frekuensi sampling yang

digunakan adalah sebesar 256 Hz (3.9 ms). Kemudian dilakukan pembacaan nilai input dari ADC dan nilai tersebut akan digunakan untuk menghitung output dari PID yang akan menjadi nilai PWM untuk mengatur kecepatan perputaran dari *DC geared* motor. Posisi perputaran motor akan dibaca oleh potensiometer yang kemudian akan diumpun balik untuk memperbaiki nilai error pada pengendali. Flowchart pengendalian dengan menggunakan PIC 16F876 adalah sebagai berikut:



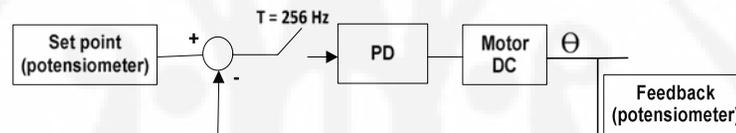
Gambar 3.10 Flowchart Pengendalian dengan PIC 16F876
Universitas Indonesia

Skema pembangkitan sinyal PWM pada PIC mikrokontroler adalah sebagai berikut:

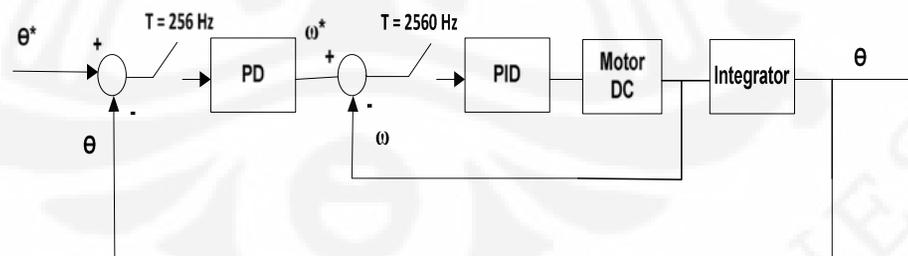


Gambar 3.11 Skema Pembangkitan Sinyal PWM

Pada program yang dibuat dibagi menjadi 2 jenis pengendalian. Skema pengendalian PID-nya adalah sebagai berikut:



Gambar 3.12 Skema Pengendalian Posisi dengan PD



Gambar 3.13 Skema Pengendalian Posisi dan Kecepatan dengan PD dan PID

Pada skema pengendalian posisi dan kecepatan dengan menggunakan 2 buah pengendali, yaitu PD dan PID, terlebih dahulu dilakukan pengendalian kecepatan dengan menggunakan frekuensi sampling 2560Hz. Pengendali ini akan

melakukan koreksi error untuk memperbaiki kecepatan sistem sampai 10 kali, kemudian dilanjutkan koreksi error posisi oleh pengendali PD untuk posisi. Setelah itu koreksi akan dilakukan terus-menerus sampai nilai *setpoint* sama dengan nilai output.

3.6 Tuning Parameter PID

3.6.1 Tuning Parameter PID untuk Mengendalikan Kecepatan Motor

Rumus PID yang digunakan adalah:

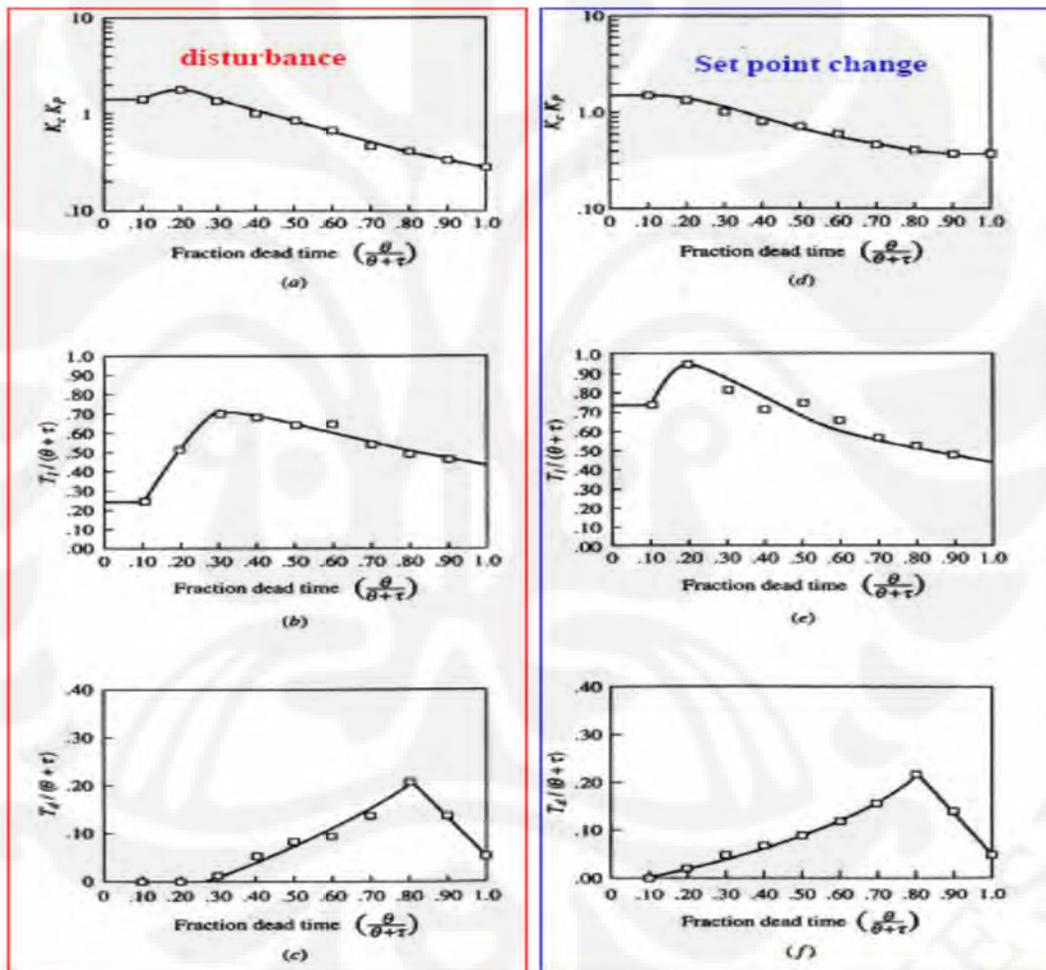
$$MV(t) = K_c \left[E(t) + \frac{1}{T_I} \int_0^t E(t') dt' - T_d \frac{dCV(t)}{dt} \right] \quad (3.9)$$

Dimana nilai yang diukur pada bagian derivative adalah selisih dari *controlled variable* bukan selisih dari error (dimana $E(t) = SP(t) - CV(t)$). Hal ini akan membuat perubahan dari setpoint tidak akan memberikan respons yang berlebihan pada bagian derivative sehingga membuat bagian derivative menjadi lebih stabil terhadap variasi setpoint.

Berdasarkan rumus PID yang digunakan diatas, terlihat bahwa terdapat 3 buah parameter, yaitu K_c , T_I dan T_d yang menentukan besaran keluaran kontrol yang dihasilkan. Agar mendapatkan pengendalian yang optimal pada sistem, maka kita perlu mengoptimalkan nilai dari ketiga parameter PID tersebut dengan cara memberikan nilai konstanta yang sesuai pada ketiga parameter. Salah satu cara yang dapat digunakan untuk menentukan besar parameter PID tersebut adalah dengan menggunakan metode yang dikembangkan oleh Ciancone dan Marlin(1992) yang dinamakan dengan nama korelasi Ciancone. Metode penentuan parameter PID dengan menggunakan korelasi Ciancone terdiri dari beberapa tahapan, yaitu sebagai berikut:

1. Pastikan bahwa tujuan performa dan asumsi yang digunakan sesuai
2. Menentukan model dinamis secara empiris (dengan menggunakan metode process reaction curve) dengan nilai parameter K_p , Θ , dan τ .
3. Menghitung fraksi dari dead time, yaitu $\Theta / (\Theta + \tau)$.

4. Memilih korelasi yang sesuai, yaitu disturbance(gangguan) atau setpoint, menggunakanlah disturbance apabila kita tidak yakin.
5. Menentukan nilai dimensionless tuning dari grafik gain ($K_c K_p$), integral time [$T_I / (\Theta + \tau)$], dan derivative time [$T_D / (\Theta + \tau)$].
6. Menghitung pengendali dimensional tuning, contohnya $K_c = (K_c K_p) / K_p$.
7. Menerapkan tuning yang baik tersebut sesuai dengan keperluan.



Gambar 3.14 Tuning Korelasi Ciancone

Tujuan yang ingin dicapai dari proses tuning parameter PID dengan menggunakan korelasi Ciancone adalah sebagai berikut:

1. Meminimalisasi IAE(Integral Absolute Error) dari variabel yang dikontrol. Nilai dari IAE adalah sebagai berikut:

$$IAE = \int_0^{\infty} |SP(t) - CV(t)| dt \quad (3.10)$$

2. Proses tuning ini telah memperhitungkan perubahan $\pm 25\%$ error pada parameter model proses.
3. Membatasi nilai variasi dari manipulated variable.

Nilai dari korelasi Ciancone akan memberikan nilai K_c , T_I dan T_d berdasarkan nilai parameter model sistem yang digunakan. Pendekatan model yang digunakan dalam menentukan parameter PID adalah dengan menggunakan pendekatan model dinamis orde 1 dengan menggunakan dead time, yaitu

$$G_p(s) = \frac{Y(s)}{X(s)} = \frac{K_p e^{-\theta s}}{\tau s + 1} \quad (3.11)$$

Berdasarkan perhitungan dengan menggunakan korelasi Ciancone akan didapatkan bahwa nilai gain controller yang tinggi bisa didapatkan apabila sistem memiliki fraksi dead time yang kecil karena proses dengan deadtime yang besar lebih sulit dikontrol. Nilai derivative time akan mendekati nol untuk fraksi nilai dead time yang kecil dan meningkat untuk waktu dead time yang lebih lama untuk mengkompensasi nilai gain controller yang mengecil. Nilai integral time juga cenderung mengecil seiring dengan kenaikan nilai dead time.

Selain grafik korelasi perubahan disturbance, terdapat pula grafik korelasi perubahan setpoint. Prosedur untuk melakukan proses tuning parameter PID juga sama seperti yang dilakukan pada grafik korelasi perubahan disturbance dengan asumsi perubahan set point terjadi secara step. Pilihan apakah kita akan memilih korelasi disturbance atau setpoint tergantung pada dominasi variasi input yang dialami oleh sistem kontrol.

Parameter PID yang didapatkan dari hasil tuning ini dapat saja bukan merupakan nilai optimum dari respons yang kita inginkan. Hal ini dapat

dikarenakan ketidaklinearan model, error dari model dinamis sistem yang cukup besar dan lain sebagainya. Apabila respons yang didapatkan dari sistem masih kurang baik, kita dapat mengubah nilai parameter tersebut hingga didapatkan respons yang lebih baik sesuai keinginan kita. Jadi, parameter PID dengan menggunakan korelasi Ciancone ini merupakan tuning awal dari sistem pengendali.

Penerapan sistem digital pada pengendali PID juga mempengaruhi proses tuning parameter PID. Hal ini disebabkan karena sampling mengakibatkan delay pada loop feedback dengan delay sebesar $\Delta t/2$ (mirip seperti dead time). Oleh karena itu, pada proses tuning PID nilai dead time awal ditambahkan dengan setengah periode sampling, yaitu:

$$\theta' = \theta + \Delta t / 2 \quad (3.12)$$

Perhitungan metode korelasi Ciancone adalah sebagai berikut:

$$G_p(s) = \frac{0.3538 e^{-0.02524 s}}{0.08436 s + 1}$$

$$K_p = \Delta/\delta = (1.769)/(5) = 0.3538$$

$$\tau = 1.5(t_{63\%} - t_{28\%}) = 1.5(0.1098 - 0.05356) = 1.5(0.05624) = 0.08436 \text{ s}$$

$$\Theta = t_{63\%} - \tau = 0.1098 - 0.08436 = 0.02524 \text{ s}$$

$$\Delta t/2 = (0.0039/2) = 0.00195 \text{ s}$$

$$\theta' = \theta + \Delta t / 2 = 0.02719$$

$$\text{Fraksi dead time} = \frac{\theta'}{\theta' + \tau} = \frac{0.02719}{0.02719 + 0.08436} = \frac{0.02719}{0.11155} = 0.2437 \quad (3.13)$$

Berdasarkan gambar 3.14 maka didapat:

$$K_c K_p = 1.65 \quad K_c = 1.65/0.3538 = 4.66$$

$$T_I / (\Theta' + \tau) = 0.62 \quad T_I = 0.62(0.11155) = 0.069 \text{ s}$$

$$T_d / (\Theta' + \tau) = 0.0037 \quad T_d = 0.0037(0.11155) = 0.00041 \text{ s}$$

Berdasarkan parameter diatas, maka didapatkan nilai K_p , K_I , dan K_d sebagai berikut:

$$K_p = K_c = 4.66 \quad (3.14)$$

$$K_I = (K_c / T_I) = 4.66/0.069 = 67.536 \quad (3.15)$$

$$K_d = K_c T_d = 4.66(0.00041) = 0.0019 \quad (3.16)$$

Persamaan PID dari parameter diatas adalah sebagai berikut:

$$MV = K_p + \frac{K_I}{s} + K_d s = \frac{4.66s + 67.536 + 0.0019s^2}{s}$$

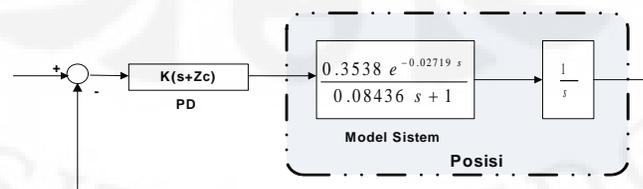
$$MV = \frac{0.0019(s^2 + 2452.63s + 35545.26)}{s} \quad (3.17)$$

3.6.2 Tuning Parameter PD untuk Mengendalikan Posisi Motor

Model sistem yang dinyatakan kedalam fungsi alih posisi adalah sebagai berikut:

$$G_p(s) = \frac{\theta(s)}{u(s)} = \frac{0.3538 e^{-0.02719 s}}{0.08436 s + 1} \left(\frac{1}{s} \right) = \frac{0.3538 e^{-0.02719 s}}{0.08436 s^2 + s} \quad (3.18)$$

Berdasarkan model tersebut akan dirancang pengendali PD untuk mengendalikan posisi dari sistem *DC geared motor*. Pengendali PD digunakan untuk mengendalikan posisi *DC geared motor* untuk menghindari nilai *steady state* sistem menjadi tak terhingga akibat penambahan integrator pada sistem. Blok diagram sistem dengan pengendali PD adalah sebagai berikut:



Gambar 3.15 Blok Diagram Pengendali PD untuk Posisi

Parameter PD ditentukan dengan menggunakan metode *root locus*. Pada blok diagram diatas, dapat terlihat bahwa terdapat nilai *deadtime* sebesar -0.02719 s pada model sistem. Nilai *deadtime* ini pada umumnya tidak dapat digambarkan secara langsung dengan menggunakan *root locus* sehingga digunakan pendekatan untuk memodelkan *deadtime* tersebut ke dalam fungsi alih orde pertama dengan lag sesuai dengan rumus [5] :

$$e^{-Ts} \approx \frac{1}{Ts+1} \quad (3.19)$$

dimana T = waktu delay/ *dead time* dengan *delay time* yang kecil

Jadi, *dead time* dari model sistem dimodelkan menjadi

$$e^{-Ts} \approx \frac{1}{Ts+1} \approx \frac{1}{0.02719s+1} \quad (3.20)$$

Fungsi alih open loop dari adalah sebagai berikut:

$$G_{OLpos1} = \frac{0.3538}{0.08436s^2 + s} \cdot \frac{1}{0.02719s+1}$$

$$G_{OLpos1} = \frac{0.3538}{0.00229s^3 + 0.11155s^2 + s} \quad (3.21)$$

Fungsi alih close loop diatas adalah sebagai berikut:

$$G_{CLpos1} = \frac{0.3538K}{0.00229s^3 + 0.11155s^2 + s + 0.3538}$$

$$G_{CLpos1} = \frac{0.3538K}{(s+37.0327)(s+11.3103)(s+0.3689)} \quad (3.22)$$

Langkah perancangan PD dengan menggunakan *root locus* adalah sebagai berikut:

1. Menentukan kedudukan *pole* dominan sesuai spesifikasi pengendali yang diinginkan.

Sistem DC Geared Motor yang diinginkan memiliki overshoot sebesar 2 % dengan settling time 0.2 s.

$$\%OS = e^{-\zeta\pi/\sqrt{1-\zeta^2}} \cdot 100 \quad (3.23)$$

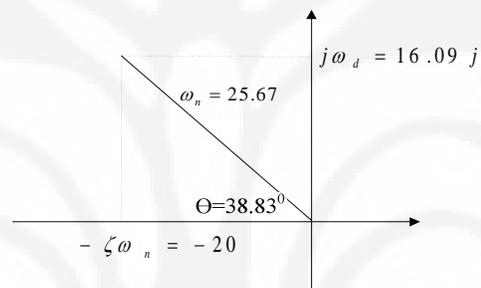
$$\zeta = \frac{-\ln(\%OS/100)}{\sqrt{\pi^2 + \ln^2(\%OS/100)}} = \frac{3.91}{5.02} = 0.779 \quad (3.24)$$

$$\zeta = \cos\theta = 0.779 \quad (\theta = 38.83^\circ)$$

$$T_s = \frac{4}{\zeta\omega_n} = 0.2 \quad (3.25)$$

$$\zeta\omega_n = 20 \quad (\omega_n = 25.67)$$

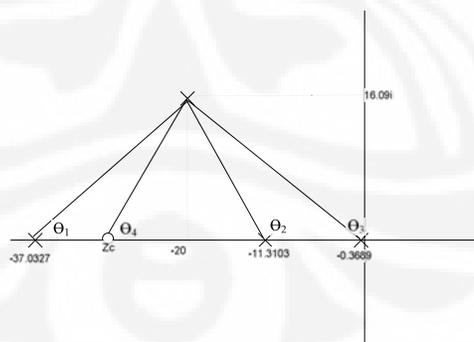
Gambar dari dominan pole dapat ditunjukkan oleh gambar berikut ini:



Gambar 3.16 Dominan Pole Spesifikasi PD untuk Posisi

Jadi, dominan pole dari sistem adalah $-20 \pm 16.09j$

2. Merancang pengendali PD



Gambar 3.17 Angle of Departure dan Angle of Arrival dengan PD

$$\theta_1 = \tan^{-1} \frac{16.09}{17.0327} = 43.37^\circ$$

$$\theta_3 = 180^\circ - \tan^{-1} \frac{16.09}{19.6311} = 140.66^\circ$$

$$\theta_2 = 180^\circ - \tan^{-1} \frac{16.09}{8.6897} = 118.37^\circ$$

$$\Sigma_{\theta_{zero}} - \Sigma_{\theta_{pole}} = \pm(2k+1)180^\circ$$

$$\theta_4 - (\theta_1 + \theta_2 + \theta_3) = \pm(2k+1)180^\circ \quad (3.26)$$

$$\theta_4 = 122.4^\circ$$

$$\tan \theta_4 = \frac{16.09}{20 - Z_c}$$

$$Z_c = 9.789$$

$$\text{Jadi, pengendali PD-nya adalah PD} = K(s+9.789) \quad (3.27)$$

3. Menentukan parameter PD

Bentuk fungsi alih pengendali PD yang digabungkan dengan fungsi alih close loop sistem posisi *DC geared motor* menjadi:

$$G_{CLpos1} = \frac{K0.3538(s+9.789)}{(s+37.0327)(s+11.3103)(s+0.3689)} \Big|_{s=-20+16.09j} = 1$$

$$K = 0.369 \quad (3.28)$$

Persamaan PD berdasarkan perancangan diatas adalah sebagai berikut:

$$MV = 0.369(s+9.789) \quad (3.29)$$

$$MV = 0.369s + 3.612$$

Parameter PD untuk mengendalikan posisi pada *DC geared motor* adalah sebagai berikut:

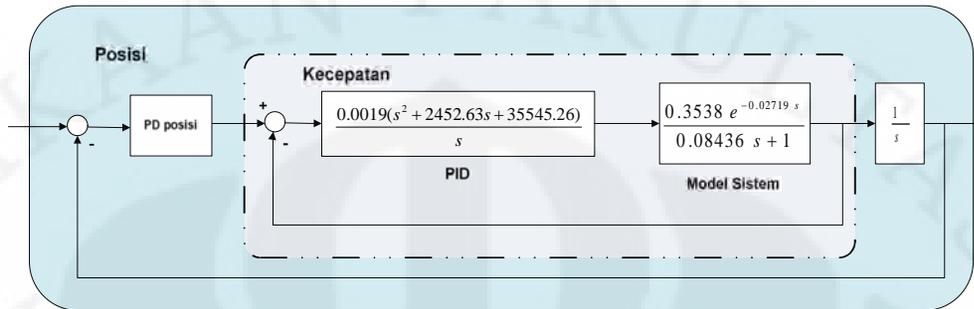
$$K_p = 3.612 \quad (3.30)$$

$$K_d = 0.369 \quad (3.31)$$

3.6.3 Tuning Parameter PID dan PD untuk Mengendalikan Posisi Motor

Berdasarkan sistem pengendalian kecepatan yang telah dirancang sebelumnya, akan dibuat sebuah pengendali PD untuk mengendalikan posisi perputaran dari *DC geared motor* dengan menggunakan metode *root locus*. Blok

diagram sistem dengan pengendali PID untuk kecepatan dan PD untuk posisi adalah sebagai berikut:



Gambar 3.18 Blok Diagram Pengendali PD dan PID untuk Kecepatan dan Posisi

Fungsi alih *open loop* model sistem dan pengendali PID untuk kecepatan adalah sebagai berikut:

$$G_{OLkec} = \frac{0.00067222 (s^2 + 2452.63s + 35545.26)}{0.08436 s^2 + s} \cdot \frac{1}{0.02719 s + 1}$$

$$G_{OLkec} = \frac{0.00067222 (s^2 + 2452.63s + 35545.26)}{0.0022937 s^3 + 0.11155 s^2 + s} \quad (3.32)$$

Fungsi alih *closed loop* dari fungsi alih diatas adalah sebagai berikut:

$$G_{CLkec} = \frac{\frac{0.00067222 (s^2 + 2452.63s + 35545.26)}{0.0022937 s^3 + 0.11155 s^2 + s}}{1 + \frac{0.00067222 (s^2 + 2452.63s + 35545.26)}{0.0022937 s^3 + 0.11155 s^2 + s}}$$

$$G_{CLkec} = \frac{0.00067222 (s^2 + 2452.63s + 35545.26)}{0.0022937 s^3 + 0.11155 s^2 + s + 0.00067222 (s^2 + 2452.63s + 35545.26)}$$

$$G_{CLkec} = \frac{0.00067222 (s^2 + 2452.63s + 35545.26)}{0.0022937 s^3 + 0.1122222 s^2 + 2.6487 s + 23.894} \quad (3.33)$$

Fungsi alih diatas merupakan fungsi alih untuk kecepatan *DC geared motor*. Selanjutnya fungsi tersebut akan diubah menjadi fungsi alih posisi dan

kemudian dilakukan perancangan pengendali PD dengan menggunakan metode *root locus*. Fungsi alih posisi adalah sebagai berikut:

$$G_{OLpos} = \frac{0.00067222 (s^2 + 2452.63s + 35545.26)}{0.0022937 s^4 + 0.1122222 s^3 + 2.6487 s^2 + 23.894 s} \quad (3.34)$$

Fungsi alih closed loop dari persamaan diatas adalah sebagai berikut:

$$G_{CLpos} = \frac{0.00067222 (s^2 + 2452.63s + 35545.26)}{0.0022937 s^4 + 0.1122222 s^3 + 2.649372 s^2 + 25.5427 s + 23.894}$$

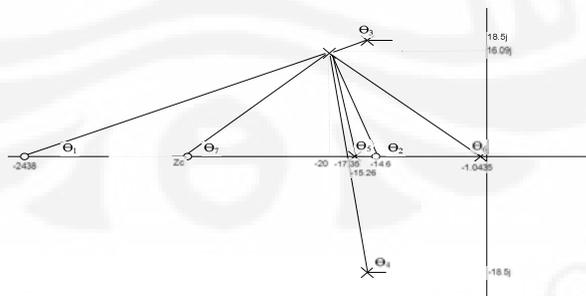
$$G_{CLpos} = \frac{K(s + 2438)(s + 14.6)}{(s + 15.2658 - 18.5012j)(s + 15.2658 + 18.5012j)(s + 17.3512)(s + 1.0435)}$$

Langkah perancangan pengendali PD dengan menggunakan *root locus* adalah sebagai berikut:

1. Menentukan kedudukan *pole* dominan sesuai spesifikasi pengendali yang diinginkan.

Sistem DC Geared Motor yang diinginkan memiliki overshoot sebesar 2 % dengan settling time 0.2 s. Spesifikasi ini sesuai dengan parameter pengendali PD sebelumnya pada bagian 3.62 sehingga didapatkan *pole* dominan dengan nilai $-20 \pm 16.09j$

2. Merancang pengendali PD



Gambar 3.19 Angle of Departure dan Angle of Arrival dengan PD dan PID

$$\theta_1 = \tan^{-1} \frac{16.09}{2418} = 0.38^\circ$$

$$\theta_4 = 180^\circ - \tan^{-1} \frac{34.59}{4.74} = 97.80^\circ$$

Universitas Indonesia

$$\theta_2 = 180^\circ - \tan^{-1} \frac{16.09}{5.4} = 108.55^\circ \quad \theta_5 = 180^\circ - \tan^{-1} \frac{16.09}{2.65} = 99.35^\circ$$

$$\theta_3 = 180^\circ + \tan^{-1} \frac{2.41}{4.74} = 206.95^\circ \quad \theta_6 = 180^\circ - \tan^{-1} \frac{16.09}{18.9565} = 139.67^\circ$$

$$\Sigma_{\theta_{zero}} - \Sigma_{\theta_{pole}} = \pm(2k+1)180^\circ$$

$$(\theta_1 + \theta_2 + \theta_7) - (\theta_3 + \theta_4 + \theta_5 + \theta_6) = \pm(2k+1)180^\circ \quad (3.35)$$

$$\theta_2 = 254.84^\circ$$

$$\tan \theta_2 = \frac{16.09}{20 - Z_c}$$

$$Z_c = 15.65$$

$$\text{Jadi, pengendali PD-nya adalah PD} = (s+15.65) \quad (3.36)$$

3. Menentukan parameter PD

Bentuk fungsi alih pengendali PID yang digabungkan dengan fungsi alih close loop sistem posisi *DC geared motor* menjadi:

$$G_{CLpos} = \frac{K(s+15.65)(s+2438)(s+14.6)}{(s+15.2658-18.5012j)(s+15.2658+18.5012j)(s+17.3512)(s+1.0435)} \Big|_{s=-20+1609j} = 1$$

$$K = 0.375 \quad (3.37)$$

Persamaan PD berdasarkan perancangan diatas adalah sebagai berikut:

$$MV = 0.375(s+15.65) \quad (3.38)$$

$$MV = 0.375s + 5.868$$

Parameter PD untuk mengendalikan posisi pada *DC geared motor* adalah sebagai berikut:

$$K_p = 5.868 \quad (3.39)$$

$$K_d = 0.375 \quad (3.40)$$

BAB 4

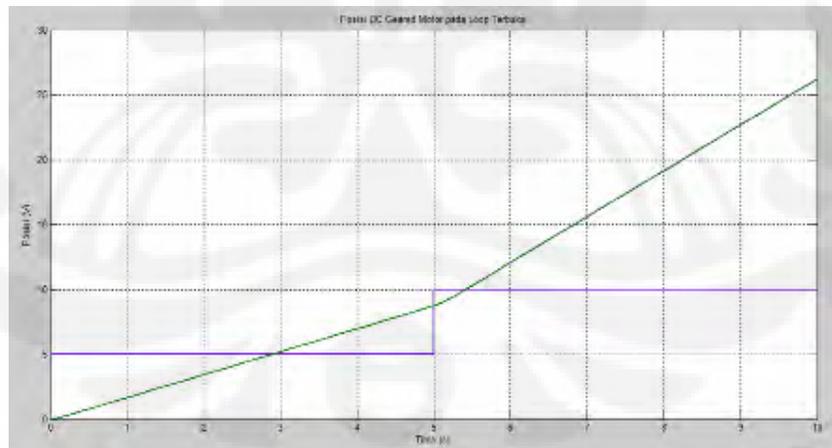
SIMULASI, ANALISA, DAN IMPLEMENTASI

Pada bab ini akan dibahas mengenai hasil simulasi dari pengendalian DC geared motor dengan menggunakan PID untuk mengatur perputaran *proportional valve* pada *fuel cell*. Simulasi dilakukan dengan menggunakan program Simulink yang terdapat pada software MATLAB R2009a. Setelah itu juga akan dibahas mengenai implementasinya pada sistem DC geared motor yang telah dibuat.

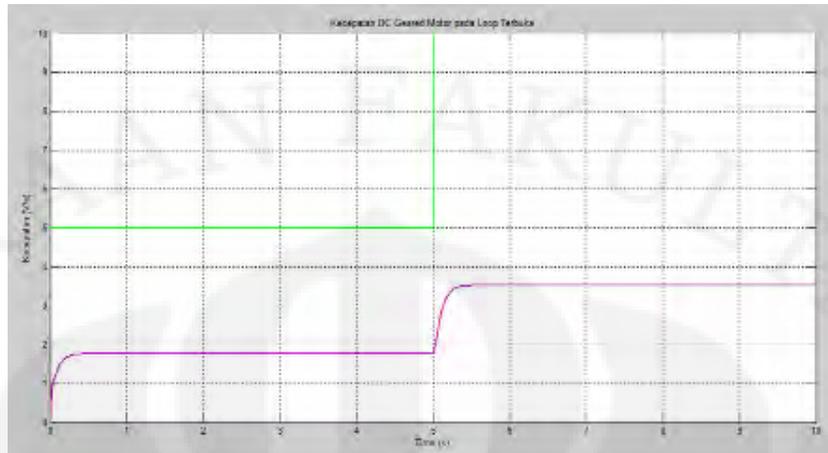
4.1 Simulasi Open Loop

Simulasi *open loop* merupakan proses pemberian masukan kepada DC geared motor berupa tegangan inputan motor ke model yang telah dibuat yaitu berupa step tegangan dari 5V ke 10V. Pada simulasi ini, rangkaian percobaan diberikan nilai delay sebesar 0.02524 s.

Percobaan ini akan digunakan untuk menguji hasil keluaran dari sistem baik untuk posisi maupun kecepatan DC geared motor. Grafik respons posisi dan kecepatan dari DC geared motor adalah sebagai berikut:



(a)



(b)

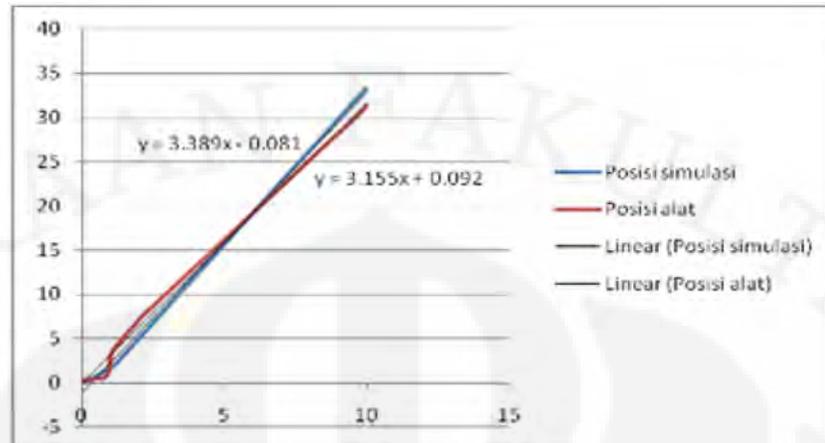
Gambar 4.1 Grafik Posisi (a), Kecepatan (b) *DC Geared Motor* pada Loop Terbuka

Analisa tanggapan waktu untuk kecepatan dari sistem ini ditunjukkan oleh tabel 4.1 berikut ini:

Tabel 4.1 Nilai Tanggapan Waktu Kecepatan pada Percobaan *Open Loop*

Parameter	Nilai
Rise Time	0.2105 s
Peak Time	-
Settling Time	0.54 s
Overshoot	-
Final Value	3.538
Error Steady State	0 %

Grafik tanggapan posisi diatas akan dibandingkan dengan grafik tanggapan posisi yang didapatkan dari pengambilan data pada sistim *DC geared motor*. Hasil perbandingan tersebut akan ditampilkan pada grafik 4.1 untuk melihat kesesuaian antara model dan sistem.



Grafik 4.1 Gradien Tanggapan Posisi Sistem dan Simulasi

Pada grafik tanggapan posisi akan dilihat kesesuaiannya dengan data percobaan yang telah dilakukan. Setelah dilakukan penggambaran gradien tanggapan posisi dari *DC geared motor*, maka didapatkan bahwa terdapat error sebesar 7.42 % terhadap model simulasi alat. Hal ini disebabkan karena adanya faktor pembulatan dan pengabaian overshoot yang dilakukan selama proses perhitungan. Namun error ini masih dapat ditoleransi karena model yang digunakan untuk menentukan parameter PID dengan menggunakan korelasi Ciancone mampu mentoleransi error sebesar 25% [3].

Oleh karena itu, akan dirancang pengendali PID dengan menggunakan parameter tuning berdasarkan korelasi Ciancone untuk memperbaiki respon transien tersebut. Pada simulasi berikutnya akan digunakan perubahan *setpoint* dari 1.769 V/s ke 3.538 V/s untuk pengendalian kecepatan menggunakan PID dan perubahan *setpoint* dari 5 V ke 10 V untuk perubahan posisi dengan pengendali PID.

4.2 Simulasi Pengendali PID untuk Kecepatan *DC Geared Motor*

Pengendalian PID untuk kecepatan *DC geared motor* ditujukan untuk memperbaiki error kecepatan dari sistem yang telah dibuat. Pada sistem ini digunakan time delay sebesar 0.02719 s disebabkan karena efek diskritisasi sistem dengan waktu sampling sebesar 3.9 ms yang digunakan dalam sistem pengendali PID ($\theta' = \theta + \Delta t / 2 = 0.02719s$). Simulasi pengendali yang digunakan disini

menggunakan C-Mex, yang merupakan suatu program yang menggunakan bahasa C yang terintegrasi dengan MATLAB. Dengan menggunakan C-Mex, kita dapat membuat blok diagram Simulink dengan algoritma yang kita inginkan. Kelebihan pemrograman dengan C-Mex dibandingkan dengan bahasa m-file adalah waktu pemrosesan yang lebih cepat.

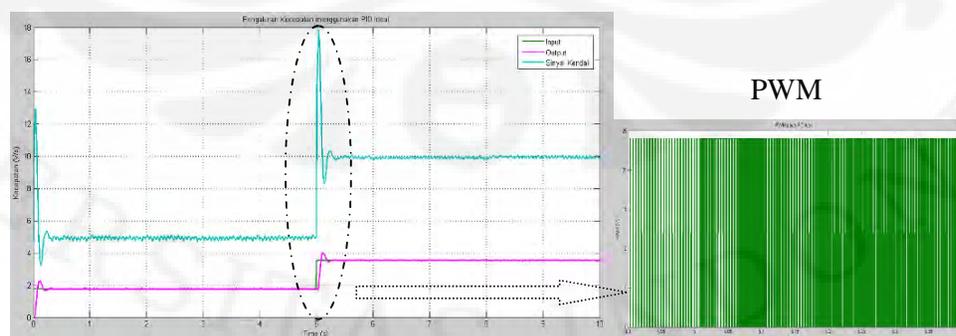
Simulasi pengendalian kecepatan dengan menggunakan PID ini akan dibagi menjadi 2 bagian, yaitu dengan menggunakan rumus PID diskrit ideal (dengan menggunakan selisih error) dan dengan menggunakan rumus PID dengan menggunakan selisih *Controlled Variable* (CV). Keduanya menggunakan parameter PID sesuai dengan korelasi Ciancone, yaitu dengan $K_p = 4.66$, $K_I = 67.536$, dan $K_d = 0.0019$. Berdasarkan hasil simulasi ini kemudian akan dibandingkan keduanya untuk mendapatkan respons pengendali yang terbaik. Tuning PID dengan menggunakan korelasi Ciancone ini kemudian akan dikoreksi dengan menggunakan metode *trial and error*.

4.2.1 Simulasi Pengendali PID Ideal untuk Kecepatan Motor

Pada simulasi ini akan digunakan rumus pengendali PID ideal dengan menggunakan selisih error yaitu sebagai berikut:

$$MV_N = K_c \left[E_N + \frac{\Delta t}{T_I} \sum_{i=1}^N E_i + \frac{T_d}{\Delta t} (E_N - E_{N-1}) \right] \quad (4.1)$$

Hasil dari simulasi rangkaian pengendali PID ideal beserta tanggapan PWM untuk kecepatan adalah sebagai berikut:



Gambar 4.2 Grafik Pengendali PID Ideal untuk Kecepatan

Analisa tanggapan waktu untuk kecepatan dengan menggunakan PID ideal adalah sebagai berikut:

Tabel 4.2 Nilai Tanggapan Waktu Kecepatan pada PID Ideal

Parameter	Nilai
Rise Time	0.035 s
Peak Time	0.12 s
Settling Time	0.35 s
Overshoot	11.64 %
Final Value	3.538
Error Steady State	2 %

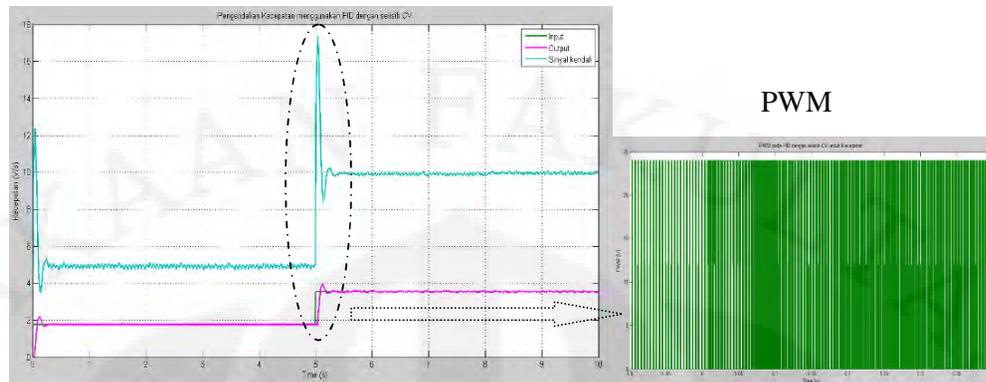
Terlihat bahwa pengendali PID ideal dengan menggunakan parameter dari korelasi Ciancone sudah menunjukkan hasil pengendalian yang cukup baik. Respons transien dari sistem sudah menjadi lebih baik walaupun terdapat overshoot sebesar 11.64 %.

4.2.2 Simulasi Pengendali PID dengan Selisih CV untuk Kecepatan

Pada simulasi ini akan digunakan rumus pengendali PID dengan menggunakan selisih *controlled variable* (CV) yaitu sebagai berikut:

$$MV_N = K_c \left[E_N + \frac{\Delta t}{T_i} \sum_{i=1}^N E_i - \frac{T_d}{\Delta t} (CV_N - CV_{N-1}) \right] \quad (4.2)$$

Hasil simulasi rangkaian pengendali PID dengan selisih CV beserta tanggapan PWM untuk kecepatan adalah sebagai berikut:



Gambar 4.3 Grafik Pengendali PID dengan Selisih CV untuk Kecepatan

Analisa tanggapan waktu untuk kecepatan dengan menggunakan PID dengan selisih CV adalah sebagai berikut:

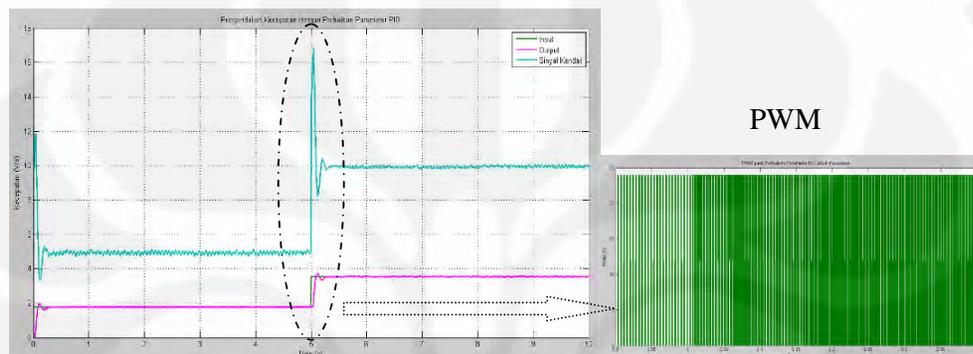
Tabel 4.3 Nilai Tanggapan Waktu Kecepatan pada PID dengan Selisih CV

Parameter	Nilai
Rise Time	0.035 s
Peak Time	0.12 s
Settling Time	0.3 s
Overshoot	9.38 %
Final Value	3.538
Error Steady State	0 %

Berdasarkan tabel diatas dapat dilihat bahwa pengendali PID dengan menggunakan selisih CV memberikan nilai overshoot dan settling time yang lebih baik dibandingkan PID ideal. Hal ini berlaku apabila sistem mengalami perubahan *setpoint* sehingga error yang ditimbulkan oleh perubahan *setpoint* tersebut tidak mempengaruhi nilai derivative pada pengendali PID. Pada bagian 4.2.3 akan dilakukan tuning parameter PID secara trial dan error berdasarkan PID dengan selisih CV untuk memperbaiki overshoot dari sistem.

4.2.3 Simulasi Perbaikan Parameter PID untuk Kecepatan

Pada bagian ini akan dilakukan perbaikan parameter PID yang telah dibuat pada bagian sebelumnya dengan metode *trial and error*. Setelah dilakukan beberapa kali iterasi, didapatkan parameter PID dengan nilai $K_p = 5$, $K_I = 50$, dan $K_d = 0.002$. Hasil simulasi dengan menggunakan parameter PID ini beserta tanggapan PWM-nya adalah sebagai berikut:



Gambar 4.4 Grafik Kecepatan dengan Perbaikan Parameter PID

Analisa tanggapan waktu untuk kecepatan dengan perbaikan parameter PID adalah sebagai berikut:

Tabel 4.4 Nilai Tanggapan Waktu Kecepatan dengan Perbaikan Parameter PID

Parameter	Nilai
Rise Time	0.035 s
Peak Time	0.115 s
Settling Time	0.4 s
Overshoot	3.73 %
Final Value	3.538
Error Steady State	0 %

Berdasarkan tabel diatas dapat dilihat bahwa terdapat sedikit perbaikan pada peak time serta perbaikan yang cukup signifikan pada nilai overshoot,

namun settling time dari sistem menjadi lebih lambat 0.1 s dibandingkan dengan simulasi sebelumnya. Hal ini menunjukkan bahwa tuning parameter PID dengan menggunakan korelasi Ciancone sudah menunjukkan hasil yang cukup baik. Penyesuaian parameter PID yang lebih lanjut dapat disesuaikan dengan jenis respons yang diinginkan (untuk memperbaiki respons *transien* atau respons *steady state*).

4.3 Simulasi Pengendali PD untuk Posisi *DC Geared Motor*

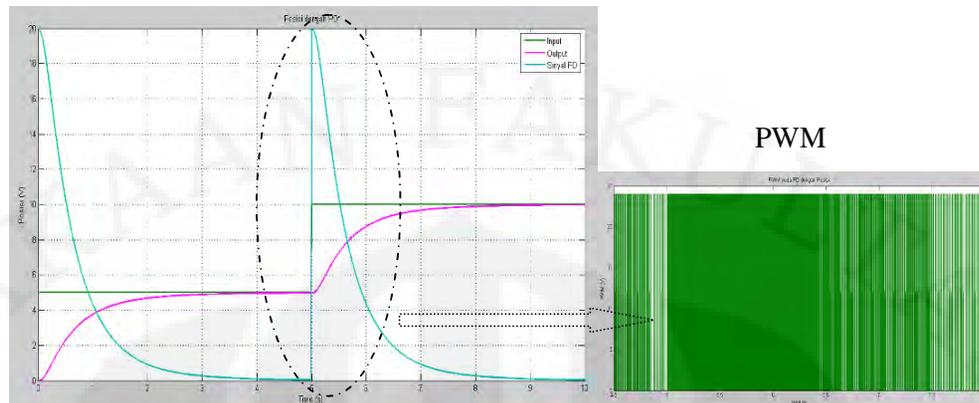
Pada bagian ini akan dibahas tentang pengendali PID yang digunakan untuk mengendalikan posisi dari *DC geared motor*. Pengendali PID yang digunakan disini akan dibagi menjadi 2 bagian, yaitu pengendali PD tunggal untuk mengendalikan posisi serta pengendali PD dan PID untuk mengendalikan posisi dan kecepatan.

4.3.1 Simulasi Pengendali PD untuk Mengendalikan Posisi

Pengendalian posisi pada *DC geared motor* dilakukan dengan menambahkan integrator pada model sistem kecepatan yang telah dibuat sebelumnya sehingga model sistem berubah menjadi:

$$G_p(s) = \frac{\theta(s)}{u(s)} = \frac{0.3538 e^{-0.02719 s}}{0.08436 s^2 + s} \quad (4.3)$$

Selanjutnya akan ditambahkan parameter pengendali PD untuk mengatur posisi motor dengan parameter $K_p=3.612$ dan $K_d=0.369$ yang telah dibuat untuk melihat respon dari motor. Hasil simulasi dari Pengendali PD beserta tanggapan PWM untuk posisi adalah sebagai berikut:



Gambar 4.5 Pengendali PD untuk Posisi *DC Geared Motor*

Analisa tanggapan waktu dari pengendali PD untuk posisi adalah sebagai berikut:

Tabel 4.5 Nilai Tanggapan untuk Waktu Posisi dengan Pengendali PD

Parameter	Nilai
Rise Time	1.49 s
Peak Time	-
Settling Time	3.1 s
Overshoot	-
Final Value	10
Error Steady State	0 %

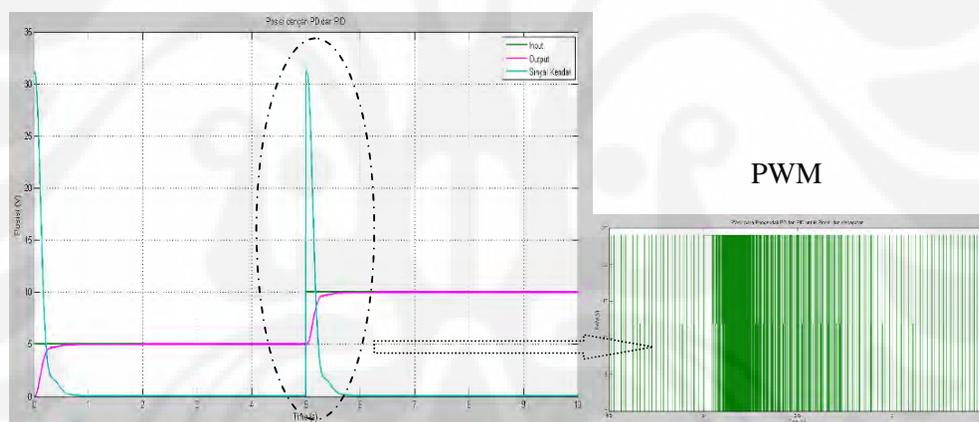
Berdasarkan hasil analisa tanggapan waktu diatas, dapat dilihat bahwa respons transien dari sistem *DC geared motor* masih cukup lambat. Oleh karena itu, pada bagian berikutnya akan dibahas tentang pengendali PID dan PD untuk mengendalikan kecepatan dan posisi untuk memperbaiki respons tanggapan waktu sistem *DC geared motor* dan kemudian dibandingkan hasilnya dengan hasil simulasi pada bagian ini.

4.3.2 Simulasi Pengendali PID dan PD untuk Kecepatan dan Posisi

Untuk mengendalikan perubahan posisi dari *DC geared motor* sekaligus mengendalikan kecepatannya, maka dirancanglah pengendali PID untuk posisi. Pada sistem ini digunakan model yang telah diturunkan pada bagian 3.5 untuk sistem open loop yaitu:

$$G_{OLpos} = \frac{0.00067222 (s^2 + 2452.63s + 35545.26)}{0.0022937 s^4 + 0.1122222 s^3 + 2.6487 s^2 + 23.894 s} \quad (4.4)$$

Pada sistem ini akan digunakan parameter PD yang telah didapatkan berdasarkan perancangan pengendali dengan menggunakan *root locus*, yaitu dengan $K_p = 5.868$ dan $K_d = 0.375$. Hasil simulasi dengan menggunakan parameter PD dan PID beserta tanggapan PWM untuk mengendalikan posisi dan kecepatan adalah sebagai berikut:



Gambar 4.6 Grafik Posisi dengan Pengendali PD dan PID

Analisa tanggapan waktu untuk pengendalian posisi dengan menggunakan PID adalah sebagai berikut:

Tabel 4.6 Nilai Tanggapan Waktu untuk Posisi dengan Memakai PD dan PID

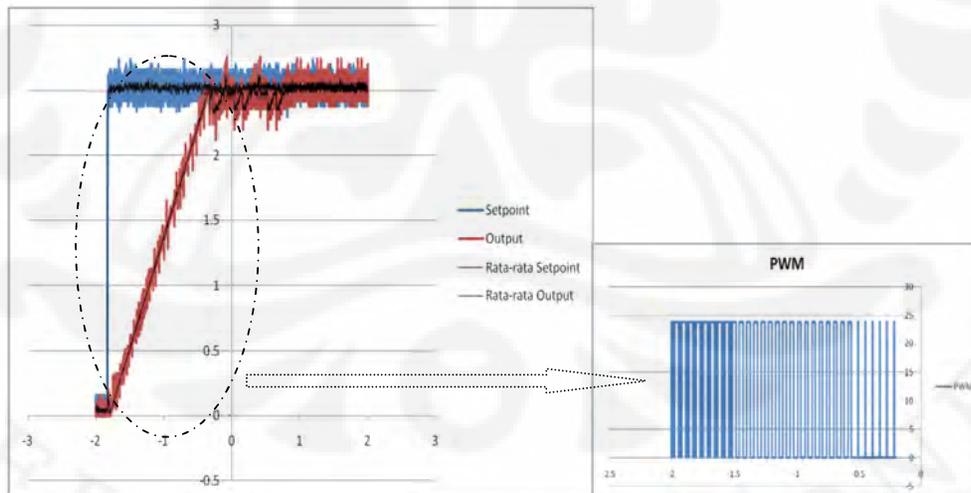
Parameter	Nilai
Rise Time	0.21 s
Peak Time	-
Settling Time	1.1 s

Overshoot	-
Final Value	10
Error Steady State	0%

Berdasarkan hasil analisa yang ditunjukkan pada tabel 4.6 dan tabel 4.5, dapat dilihat bahwa sistem yang dikendalikan dengan menggunakan pengendali PD dan PID untuk posisi dan kecepatan menunjukkan hasil tanggapan waktu yang lebih baik dibandingkan dengan pengendali PD saja untuk mengendalikan posisi sistem. Hal ini dapat terlihat dengan adanya respon tanggapan waktu berupa perbaikan rise time sebesar 1.28 s dan settling time sebesar 2 s.

4.4 Implementasi Sistem Pengendali

Sistem pengendali kecepatan yang dibuat kemudian diimplementasikan ke *DC geared motor* untuk dilihat respons sistemnya. Parameter PD yang digunakan pada *DC geared motor* merupakan parameter PD hasil tuning yang telah diperbaiki responsnya untuk mengurangi penggunaan bilangan *real/floating*. Berdasarkan hasil percobaan didapatkan respons sebagai berikut:



Grafik 4.2 Perubahan Output dengan Pengendali PD untuk Posisi

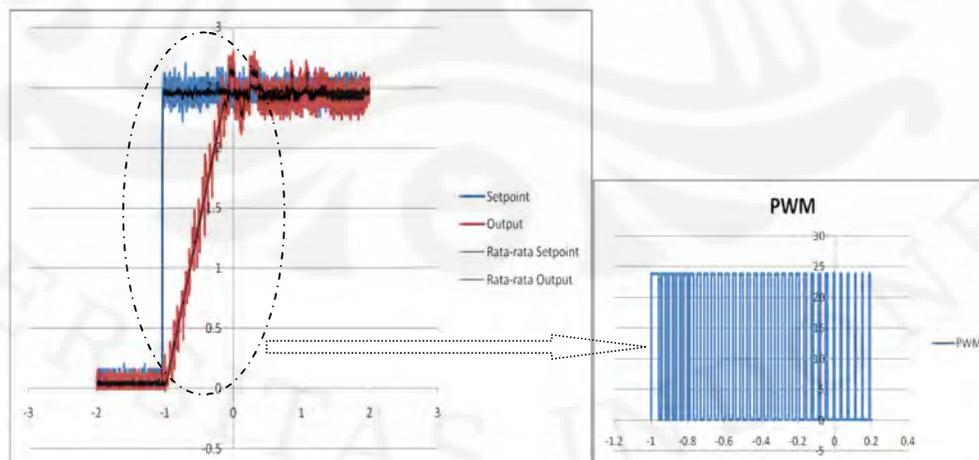
Analisa tanggapan waktu dari pengendali PD untuk posisi adalah sebagai berikut:

Tabel 4.7 Nilai Tanggapan Waktu Pengendali PD untuk Posisi

Parameter	Nilai
Rise Time	1.6 s
Peak Time	1.85 s
Settling Time	1.9 s
Overshoot	2 %
Final Value	2.5
Error Steady State	2 %

Berdasarkan tabel diatas dapat dilihat bahwa nilai tanggapan waktu sistem *DC Geared Motor* menjadi lebih lambat dibandingkan simulasinya dikarenakan faktor gesekan pada *sistim gear motor* dan faktor ketidakidealan alat lainnya.

Setelah itu, akan dilakukan pengukuran respons tanggapan waktu pada pengendali PD dan PID untuk mengendalikan posisi dan kecepatan pada sistem *DC geared motor*. Parameter PD dan PID yang digunakan pada *DC geared motor* merupakan parameter PID hasil tuning yang telah dilakukan pembulatan ke bilangan bulat(integer) dikarenakan keterbatasan memory dari mikrokontroler 16F876. Sistem pengendali PD dan PID untuk posisi dan kecepatan menghasilkan respons sebagai berikut:



Grafik 4.3 Perubahan Output dengan Pengendali PD dan PID untuk Posisi

Universitas Indonesia

Analisa tanggapan waktunya adalah sebagai berikut:

Tabel 4.8 Nilai Tanggapan Waktu Pengendali PD dan PID untuk Posisi

Parameter	Nilai
Rise Time	0.76 s
Peak Time	1.18 s
Settling Time	1.22 s
Overshoot	3.8 %
Final Value	2.5
Error Steady State	3 %

Berdasarkan tabel tanggapan waktu baik pada pengendali posisi dengan PD maupun pengendali untuk posisi dengan PD dan PID, respons transien yang ditunjukkan DC motor lebih lambat daripada hasil simulasi yang telah dilakukan. Hal ini bisa disebabkan karena pembulatan dalam menggunakan parameter PID, maupun ketidakidealan sistem. Hasil implementasi dari sistem pengendali baik pada posisi dan kecepatan sudah memberikan hasil yang memuaskan dengan respons tanggapan waktu mendekati simulasi yang dilakukan.

BAB 5

KESIMPULAN

1. Pemodelan DC *geared motor* secara empiris didapatkan fungsi transfer

$$G_p(s) = \frac{\omega(s)}{u(s)} = \frac{0.3538 e^{-0.02524 s}}{0.08436 s + 1}$$

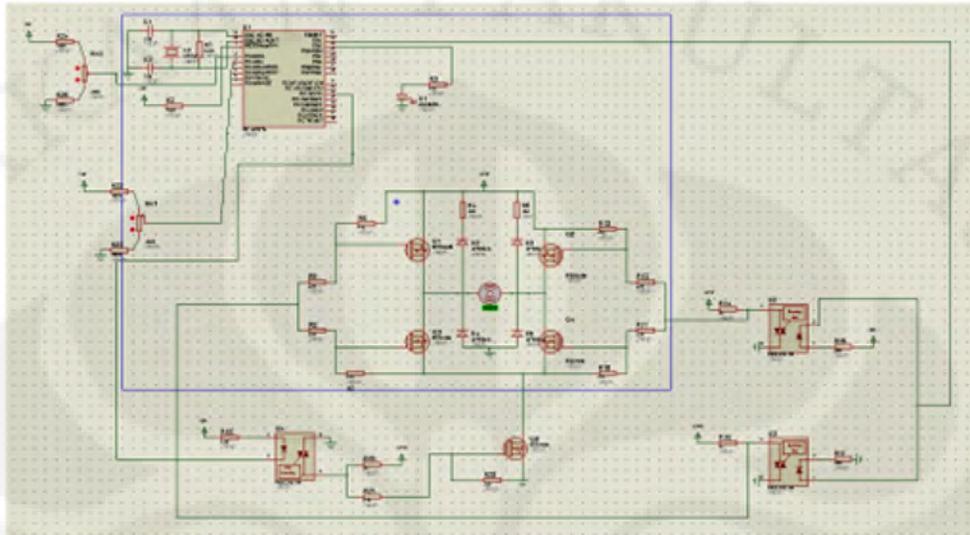
2. Korelasi Ciancone sebagai pengendali kecepatan sistim dapat dijadikan sebagai parameter tuning PID awal yang akurat dengan $K_p= 4.66$, $K_i=67.53$, dan $K_d = 0.0019$ dengan nilai rise time 0.035s, peak time 0.12s, settling time 0.3 s dan overshoot sebesar 9.38%.
3. Pengendalian kecepatan DC *geared motor* didapatkan parameter PID hasil optimasi dengan $K_p = 5$, $K_i = 50$ dan $K_d = 0.002$ dengan nilai rise time 0.035 s, peak time 0.12 s, settling time 0.4 s dan overshoot 3.73%.
4. Hasil pengendalian dengan menggunakan PD dan PID untuk kecepatan dan posisi menunjukkan hasil yang lebih baik dibandingkan pengendalian PD untuk posisi dengan selisih rise time 1.28 s dan settling time 2 s.
5. Implementasi dari sistem pengendali PD dan PID untuk mengendalikan posisi dan kecepatan sistem sudah memberikan hasil pengendalian sesuai yang diharapkan dengan parameter rise time 0.76 s, peak time 1.18 s, settling time 1.22s, overshoot 3.8 %, dan error steady state sebesar 3 %

DAFTAR REFERENSI

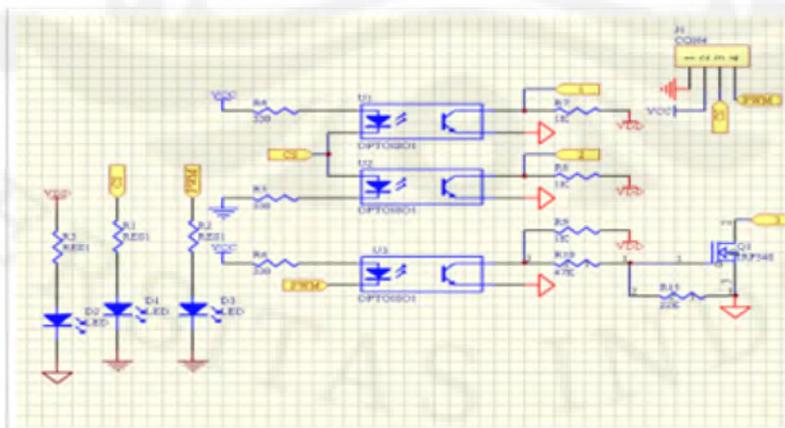
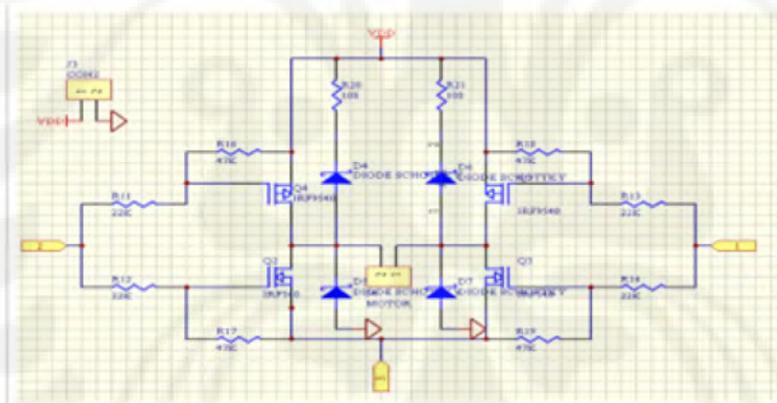
- [1] Chapman, Stephen J. "Electric Machinery and Power System Fundamentals". BAE Systems Australia. 2002.
- [2] Dewantoro, Rahardian. "Simulasi dan Analisa Pengendalian Kecepatan Putar Motor Arus Searah Penguatan Terpisah dengan Pengendali PI dan Logika Fuzzy pada Beban Bervariasi". Skripsi, 2001.
- [3] Marlin, Thomas E. "Process Control, Designing Processes and Control Systems for Dynamic Performance". McGraw-Hill International Editions. 1999.
- [4] Nise, Norman S., "Control Systems Engineering. International Student Edition", Addison Wesley, 1995
- [5] Edrington, Chris S. "Continuous-Time Control Systems Lecture: Root Locus Analysis". Arkansas State University. 2006. <<http://users.ece.utexas.edu/~friedric/clas11.pdf>>. 14 Juni 2010.
- [6] Rylee, Mike. Low Cost Bi-directional Brushed DC Motor Control using PIC16F684. Microchip Technology Inc. 2003
- [7] Charais, John. "AN 964 Implementing a PID Controller on PIC16F684". Microchip Application Notes. October 20th, 2004. <www.microchip.com/downloads/en/AppNotes/00964A.pdf>
- [8] "Dual DC Motor Speed Control with Reverse". Lashwhip Project Online. <<http://lashwhip.com/spdctrl.html>>. 20 Desember 2009
- [9] "PIC PID Micro-Controller". Shawn Lankton Online. March 14th, 2005. <<http://www.shawnlankton.com/2005/03/pic-pid-controller/>>. 30 Mei 2010.

LAMPIRAN

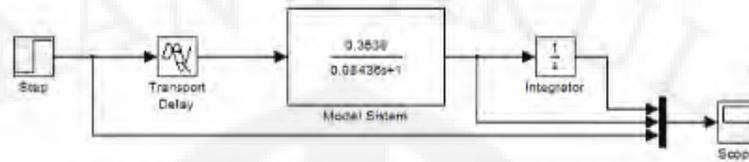
Skematik Rangkaian PIC 16F876



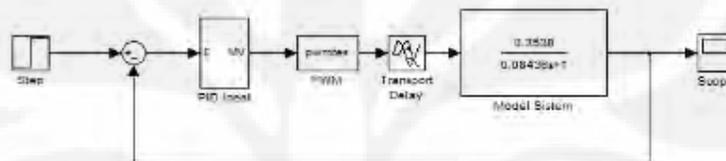
Skematik H-bridge



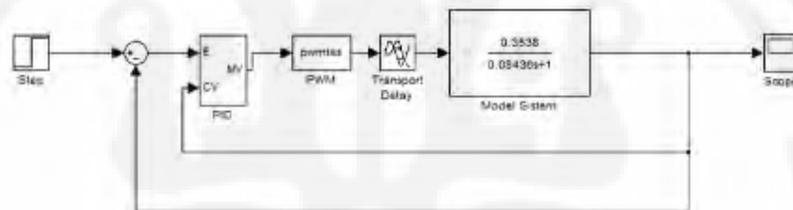
(lanjutan)

Rangkaian *open loop DC geared motor*

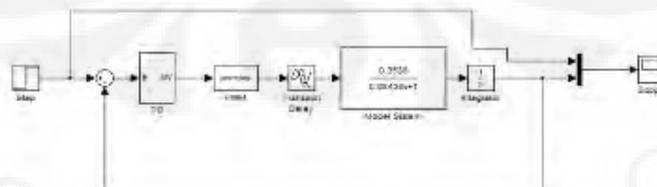
Rangkaian Pengendali PID Ideal untuk Kecepatan



Rangkaian Pengendali PID dengan Selisih CV untuk Kecepatan

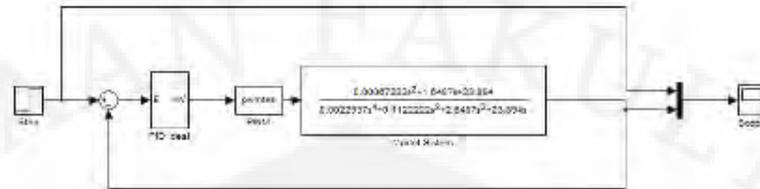


Rangkaian PD untuk Posisi



(lanjutan)

Rangkaian PD dan PID untuk posisi dan kecepatan



Gambar Sistem DC geared motor



Instruction set PIC 14 bit mikrokontroler

Opcode (binary)	Mnemonic	Description
00 0000 0000 0000	NOP	No operation
00 0000 0000 1000	RETURN	Return from subroutine, W unchanged
00 0000 0000 1001	RETFIE	Return from interrupt
00 0000 0110 0010	OPTION	Write W to OPTION register
00 0000 0110 0011	SLEEP	Go into standby mode
00 0000 0110 0100	CLRWDT	Reset watchdog timer
00 0000 0110 01ff	TRIS f	Write W to tristate register f
00 0000 1 fffffff	MOVWF f	Move W to f
00 0001 0 xxxxxxx	CLRW	Clear W to 0 (W = 0)
00 0001 1 fffffff	CLRF f	Clear f to 0 (f = 0)
00 0010 d fffffff	SUBWF f,d	Subtract W from f (d = f - W)
00 0011 d fffffff	DECF f,d	Decrement f (d = f - 1)
00 0100 d fffffff	IORWF f,d	Inclusive OR W with F (d = f OR W)
00 0101 d fffffff	ANDWF f,d	AND W with F (d = f AND W)
00 0110 d fffffff	XORWF f,d	Exclusive OR W with F (d = f XOR W)
00 0111 d fffffff	ADDWF f,d	Add W with F (d = f + W)
00 1000 d fffffff	MOVF f,d	Move F (d = f)
00 1001 d fffffff	COMF f,d	Complement f (d = NOT f)
00 1010 d fffffff	INCF f,d	Increment f (d = f + 1)
00 1011 d fffffff	DECFSZ f,d	Decrement f (d = f - 1) and skip if zero
00 1100 d fffffff	RRF f,d	Rotate right F (rotate right through carry)
00 1101 d fffffff	RLF f,d	Rotate left F (rotate left through carry)
00 1110 d fffffff	SWAPF f,d	Swap 4-bit halves of f (d = f<<4 f>>4)
00 1111 d fffffff	INCFSZ f,d	Increment f (d = f + 1) and skip if zero
01 00 bbb fffffff	BCF f,b	Bit clear f (Clear bit b of f)
01 01 bbb fffffff	BSF f,b	Bit set f (Set bit b of f)
01 10 bbb fffffff	BTFSC f,b	Bit test f, skip if clear (Test bit b of f)
01 11 bbb fffffff	BTFSS f,b	Bit test f, skip if set (Test bit b of f)
10 0 kkkkkkkkkkk	CALL k	Save return address, load PC with k
10 1 kkkkkkkkkkk	GOTO k	Jump to address k (11 bits)
11 00xx kkkkkkkkk	MOVLW k	Move literal to W (W = k)
11 01xx kkkkkkkkk	RETLW k	Set W to k and return
11 1000 kkkkkkkkk	IORLW k	Inclusive or literal with W (W = k OR W)
11 1001 kkkkkkkkk	ANDLW k	AND literal with W (W = k AND W)
11 1010 kkkkkkkkk	XORLW k	Exclusive or literal with W (W = k XOR W)
11 110x kkkkkkkkk	SUBLW k	Subtract W from literal (W = k - W)
11 111x kkkkkkkkk	ADDLW k	Add literal to W (W = k + W)