

## **STUDI PERBANDINGAN KINERJA KONTROLER PROPORSIONAL *INTEGRAL DERIVATIVE* (PID) DAN *FUZZY LOGIC CONTROL* (FLC) PADA PENGONTROLAN KECEPATAN MOTOR DC**

**Anton Viantika**

Program Studi Teknik Elektro, Fakultas Teknik, Universitas Riau Kepulauan Batam

Email : anton@ft.unrika.ac.id

### **ABSTRAK**

Motor *Direct current* (DC) banyak digunakan di dunia industri kecil dan besar, pada pengaplikasiannya motor DC sering mengalami penurunan atau kenaikan kecepatan akibat dari pembebanan pada motor, sehingga diperlukan suatu rancangan kontroler untuk mengatur kecepatan motor agar berjalan dengan kecepatan yang diinginkan. Kontrol *Proportional Integral Derivative* (PID) merupakan suatu sistem kontrol yang digunakan untuk menentukan kontrol presisi suatu sistem instrumentasi dengan karakteristik adanya umpan balik (*Feedback*) pada sistem tersebut. Metode alternatif yang digunakan untuk perbandingan dalam pengaturan kecepatan motor ini adalah kontrol dengan logika *fuzzy* (*Fuzzy Logic Controller* (FLC)). Metode logika *fuzzy* ini digunakan untuk membandingkan kinerja dari kontroler PID dalam aplikasi pengaturan kecepatan motor DC menggunakan simulink matlab. Berdasarkan pengujian yang dilakukan perbandingan kinerja pengontrolan PID dan FLC pada set point 1400 dan 1800 RPM untuk waktu naik ( $T_r$ ) dan waktu penetapan ( $T_s$ ) lebih cepat menggunakan kontrol PID dibanding FLC, pada kontrol PID masih terjadi *overshoot* 12.5 % pada *setpoint* 1400 RPM dan 1800 RPM sedangkan FLC tidak ada *overshoot* dan PID tidak mengalami *steady state error* sedangkan FLC masih mempunyai *steady state error* 0.56 % pada *setpoint* 1800 RPM.

Kata kunci : Motor DC, Kontrol PID, *Tuning*, FLC, Matlab.

### **ABSTRACT**

*Direct current (DC) motors are widely used in small and large industries, on its application, DC motors often experience a decrease or increase in speed due to loading on the motor, so a controller design is needed to regulate the motor speed so that it runs at the desired speed. Proportional Integral Derivative (PID) control is a control system that is used to determine the precision control of an instrumentation system with the characteristics of the presence of feedback on the system. The alternative method used for comparison in regulating motor speed is control with fuzzy logic (Fuzzy Logic Controller (FLC)). This fuzzy logic method is used to compare the performance of PID controllers in DC motor speed regulation applications using simulink matlab. Based on the tests carried out, the comparison of the performance of PID and FLC control at the setpoint of 1400 and 1800 RPM for the rise time ( $T_r$ ) and the setting time ( $T_s$ ) is faster using the PID control than the FLC, the PID control still has 12.5% overshoot at the 1400 RPM and 1800 setpoints. RPM while FLC has no overshoot and PID does not experience steady state error while FLC still has a steady state error of 0.56% at the setpoint of 1800 RPM.*

*Keywords: DC Motor, PID Controller, Tuning, FLC, Matlab*

## 1. PENDAHULUAN

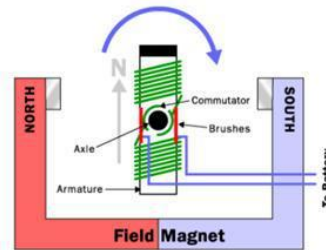
Motor DC telah dikenal sejak teori gaya Lorentz dan induksi elektromagnetik ditemukan. Motor DC sering digunakan karena kemudahannya sehingga dipakai pada berbagai macam keperluan. Kecepatan putar motor DC akan mengalami penurunan akibat dari peningkatan beban sehingga menyebabkan putarannya menjadi lambat dan kecepatannya tidak konstan. Agar kecepatan motor DC berjalan sesuai dengan kecepatan yang diinginkan diperlukan sistem kendali kecepatan motor DC. Kontroler yang sering digunakan untuk mengatasi hal ini yaitu kontroler PID dan FLC.

Konfigurasi standar kontroler PID memiliki parameter  $K_p$ ,  $K_i$  dan  $K_d$  dan FLC memiliki aturan logika *fuzzy* yang dipilih atau ditentukan berdasarkan proses *Tuning* dengan metode *trial and error* agar karakteristik perencanaan sesuai dengan kriteria desain yang diinginkan. Parameter yang diukur pada respon kecepatan adalah *rise time* ( $T_r$ ), *settling time* ( $T_s$ ), *maximum overshoot* ( $MP$ ) dan *error steady state* ( $E_{ss}$ ) terhadap masukan yang diberikan. Kendali kecepatan motor DC diatur oleh kontroler PID dan FLC yang disimulasikan dengan perangkat lunak Matlab versi 7.9, selanjutnya diperagakan melalui simulink matlab yang berfungsi sebagai Human Interface, untuk mendapatkan respon kecepatan yang konstan pada *setpoint* 1400 dan 1800 RPM pada waktu yang cepat.

## 2. LANDASAN TEORI.

### 2.1. Motor DC

Motor DC (*direct current*) merupakan salah satu jenis motor yang paling banyak digunakan dalam dunia elektronika maupun robotika. Motor DC yang digunakan pada robot beroda umumnya adalah motor DC dengan magnet permanen. Motor DC jenis ini memiliki dua buah magnet permanen sehingga timbul medan magnet di antara kedua magnet tersebut. Di dalam medan magnet inilah jangkar/rotor berputar.



Gambar 2.1 Bagan mekanisme kerja motor DC magnet permanen

Kecepatan putar motor DC ( $N$ ) dirumuskan dengan Persamaan berikut.

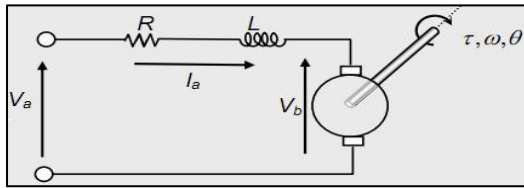
$$N = \frac{V_{TM} - I_A R_A}{K \phi} \dots \dots \dots (2.1)$$

Keterangan :

- $N$  : Kecepatan Putar (RPM)
- $V_{TM}$  : Tegangan terminal (volt)
- $I_A$  : Arus jangkar motor (Ampere)
- $R_A$  : Hambatan jangkar motor (ohm)
- $K$  : Konstanta motor (Nm/A)
- $\phi$  : Fluks magnet (weber)

Pengendalian kecepatan putar motor DC dapat dilakukan dengan mengatur besar tegangan terminal motor  $V_{TM}$ . Untuk memutar motor DC cukup diberi tegangan DC, sedangkan untuk mengatur arah perputaran, tinggal membalik polaritas tegangannya saja. Rangkaian ekivalen

motor DC magnet permanen ditunjukkan pada gambar berikut.



**Gambar 2.2** Rangkaian ekivalen Motor DC magnet permanen

Keterangan :

- $V_a$  = Tegangan armatur,
- $I_a$  = Arus motor,
- $R$  = Resistensi armatur,
- $L$  = Induktansi lilitan armatur,
- $V_b$  = Tegangan induksi balik,
- $\tau$  = Torsi motor,
- $\omega$  = Kecepatan putar motor,
- $\theta$  = Sudut putaran poros motor.

Dari rangkaian ekivalen motor, nilai  $V_a$  dapat diperoleh dari :

$$V_a(t) = L \frac{dI_a(t)}{dt} + RI_a(t) + V_b(t) \dots (2.2)$$

$$V_b(t) = K_b \omega(t) = K_b \frac{d\theta(t)}{dt} \dots (2.3)$$

$K_b$  adalah konstanta yang diukur dari tegangan yang dihasilkan oleh motor ketika berputar setiap satuan kecepatan (Volt.det/rad). Nilai  $V_a$  dapat dituliskan seperti pada persamaan berikut :

$$V_a(t) = L \frac{dI_a(t)}{dt} + RI_a(t) + K_b \frac{d\theta(t)}{dt} \dots (2.4)$$

Jika diterapkan transformasi Laplace terhadap persamaan (2.2) diperoleh hasil sebagai berikut:

$$V_a(s) = sLI_a(s) + RI_a(s) + sK_b\theta(s) \dots (2.5)$$

$$I_a(s) = \frac{V_a(s) - sK_b\theta(s)}{sL + R} \dots (2.6)$$

Bila Persamaan umum torsi output motor terhasil adalah

$$\tau(t) = K_{tn} * I_a(t) \dots (2.7)$$

Dengan  $K_{tn}$  adalah konstanta proposional torsi motor, maka transformasi laplacenya akan menjadi:

$$\tau(s) = K_{tn} * I_a(s) = K_{tn} \frac{V_a(s) - sK_b\theta(s)}{sL + R} \dots (2.8)$$

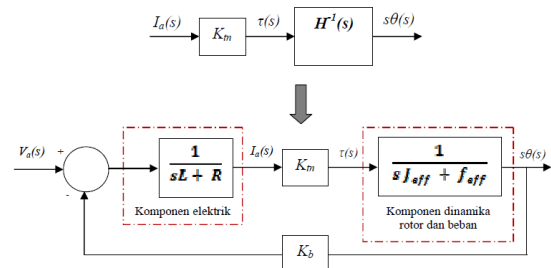
Bila persamaan torsi output motor ditinjau dari pembebanan akan diperoleh :

$$\tau(t) = J_{eff} \theta''(t) + f_{eff} \theta'(t) \dots (2.9)$$

Sehingga transformasi laplace-nya menjadi :

$$\tau(s) = J_{eff} s^2\theta(s) + f_{eff} s\theta(s) \dots (2.10)$$

Dengan asumsi bahwa motor DC dianggap linier maka fungsi transfer motor dapat digambarkan sbb:



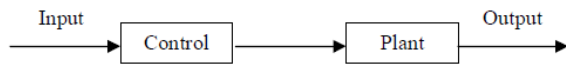
**Gambar 2.3** Fungsi transfer close loop motor DC magnet permanen

## 2.2. Sistem kontrol

Sistem kendali atau sistem kontrol (*Control system*) adalah suatu alat (kumpulan alat) untuk mengendalikan, memerintah dan mengatur keadaan dari suatu sistem. Sistem kendali secara umum terbagi menjadi dua jenis, yaitu sistem kontrol terbuka (*Open Loop Control*) dan sistem kontrol tertutup (*Close Control System*).

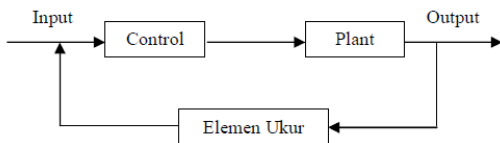
Sistem kontrol terbuka adalah sistem kontrol yang keluarannya tidak berpengaruh pada aksi pengontrolan. Jadi pada sistem kontrol terbuka, keluaran tidak

dibandingkan dengan masukan acuan, sehingga untuk setiap acuan masukan terdapat suatu kondisi operasi yang tetap.



Gambar 2.4 Model sistem kontrol terbuka

Sistem kontrol loop tertutup adalah sistem kontrol yang keluarannya mempunyai pengaruh langsung pada aksi pengontrolan. Jadi sistem kontrol loop tertutup adalah sistem kontrol berumpan balik. Sinyal kesalahan penggerak, yang merupakan selisih antara sinyal masukan dan sinyal umpan balik diumpungkan ke controller untuk memperkecil kesalahan dan membuat agar keluaran sistem mendekati harga yang diinginkan.



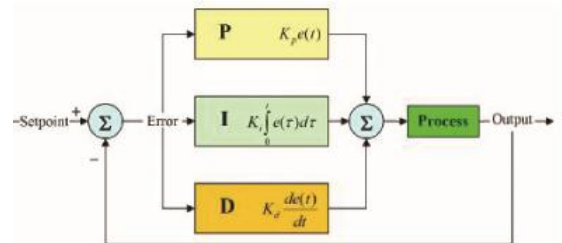
Gambar 2.5 Model sistem kontrol tertutup

Ketelitian sistem kontrol terbuka sangat tergantung pada proses dan hasil kalibrasi agar sistem dapat bekerja dengan baik. Bila terdapat gangguan pada sistem kontrol terbuka, hasil keluaran pada sistem ini tidak dapat bekerja seperti yang diharapkan. Sedangkan pada sistem kontrol tertutup, ketelitian sistem tergantung pada seberapa cepat dan akurat koreksi kesalahan pada hasil keluaran.

### 2.3. Kontrol PID

Karakteristik kontroler PID sangat dipengaruhi oleh kontribusi besar dari ketiga parameter konstanta *Proportional* (P), *Integral* (I) dan *Derivative* (D). Penyetelan konstanta  $K_p$ ,  $K_i$ , dan  $K_d$  akan mengakibatkan penonjolan sifat dari masing-masing elemen. Satu atau dua dari

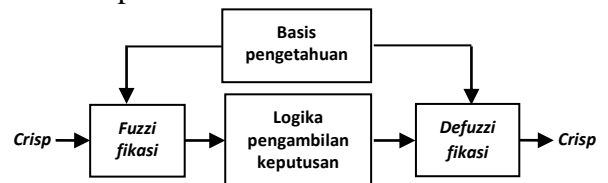
ketiga konstanta tersebut dapat disetel lebih menonjol dibanding yang lain. Konstanta yang menonjol itulah akan memberikan kontribusi pengaruh pada respon sistem. Kontrol PID adalah salah satu jenis sistem kontrol tertutup seperti terlihat pada bagan dibawah ini.



Gambar 2.6 Model sistem kontrol PID

### 2.4. Logika Fuzzy

Konsep teori *fuzzy* yang pertama kali diperkenalkan oleh L.A. Zadeh pada tahun 1965 berupa teori Himpunan *Fuzzy* (*Fuzzy Set*). Pada himpunan klasik, nilai keanggotaan bersifat jelas artinya masuk dalam anggota (bernilai 1) atau tidak (bernilai 0). Suatu elemen dalam sebuah semesta untuk himpunan *fuzzy* sifat keanggotaannya samar. Himpunan *fuzzy* berisi elemen-elemen yang mempunyai nilai keanggotaan yang bervariasi dalam suatu himpunan.



Gambar 2.5 Konfigurasi dasar sistem logika fuzzy

#### 2.4.1. Fuzzifikasi

Fuzzifikasi merupakan pemetaan nilai-nilai masukan nilai *error*(e) dan delta *error*( $\Delta e$ ) dari keluaran sistem ke bentuk himpunan-himpunan *fuzzy* dalam semesta pembicaraan tertentu. Data masukan yang

biasanya tegas (*crisp*) perlu ditransformasikan ke besaran *fuzzy*.

#### 2.4.2. Basis Aturan

Termasuk di dalamnya adalah basis data dan aturan dasar (*rule base fuzzy*). Basis data berfungsi untuk mendefinisikan himpunan-himpunan *fuzzy* dari sinyal input dan sinyal output agar dapat digunakan oleh variabel linguistik pada aturan dasar. Sedangkan aturan dasar terdiri dari aturan kontrol *fuzzy* yang dibutuhkan untuk mencapai tujuan pengontrolan.

#### 2.4.3. Logika Pengambilan Keputusan

Umumnya aturan dasar *fuzzy* dinyatakan dalam bentuk IF –THEN yang juga disebut implikasi *fuzzy*.

#### 2.4.4. Defuzzifikasi

Defuzzifikasi adalah proses pemetaan ruang aksi kontrol *fuzzy* menjadi ruang aksi kontrol non-*fuzzy* (*crisp*). Tujuannya adalah untuk menghasilkan sinyal kontrol yang dapat digunakan plant.

### 2.5. Simulink Matlab

Matlab merupakan suatu *software* pemrograman perhitungan dan analisis yang banyak digunakan dalam semua area penerapan matematika baik bidang pendidikan maupun penelitian pada universitas dan industri. Dengan matlab, maka perhitungan matematis yang rumit dapat diimplementasikan dalam program dengan lebih mudah.

Matlab diciptakan pada akhir tahun 1970-an oleh Cleve Moler untuk memberikan akses bagi mahasiswa dalam memakai LINPACK dan EISPACK (sebuah pustaka software untuk komputasi

numerik berbasis pemrograman bahasa Fortran) tanpa harus mempelajari Fortran. Tahun 1984 Jack Little, Cleve Moler, dan Steve Bangert menulis ulang MATLAB dalam bahasa pemrograman C, kemudian mendirikan The MathWorks untuk melanjutkan pengembangannya. Matlab memungkinkan manipulasi matriks, pemplot-an fungsi dan data, implementasi algoritma, pembuatan antarmuka pengguna, dan peng-antarmuka-an dengan program dalam bahasa lainnya. Meskipun hanya bernuansa numerik, sebuah kotak kakas (*toolbox*) yang menggunakan mesin simbolik MuPAD, memungkinkan akses terhadap kemampuan aljabar komputer. Sebuah paket tambahan, Simulink, menambahkan simulasi grafis multirahang dan Desain Berdasar-Model untuk sistem terlekat dan dinamik.

## 3. METODOLOGI PENELITIAN

### 3.1. Pemodelan Sistem.

Pemodelan dilakukan pada sebuah motor DC menggunakan Logika *Fuzzy* yang dibentuk dalam 2 kontrol pemodelan sistem.

#### 3.1.1. Pemodelan kontrol PID

PID kontroler konvensional hanya dapat mengolah *error* secara apa adanya, artinya *error* diolah secara *deterministic*, misalnya dikalikan secara proporsional di-*Integral*-kan, di-*defferensial*-kan. Disimpan beberapa saat untuk selanjutnya digunakan untuk sinyal kontrol berikutnya. Kontrol PID konvensional hanya tepat pada *plant* yang linier dan tidak terjadi perubahan beban, sedangkan untuk beban berubah perlu merubah parameter  $K_p$ ,  $K_i$  dan  $K_d$ .

### 3.1.2. Pemodelan FLC

Sistem logika *fuzzy* yang di rancang mempunyai 2 input yaitu *error* dan  $\Delta error$  serta 1 output yaitu kecepatan putar motor DC. Dalam sistem ini di rancang fungsi keanggotaan *error* dan  $\Delta error$  dengan jumlah label 3 label. Untuk input *error* yaitu NE (Negatif *Error*), ZE (Zero *Error*), PE (Positif *Error*) dan untuk delta *error* yaitu NDE (Negatif Delta *Error*), ZDE (Zero Delta *Error*), dan Positif Delta *Error* (PDE). Untuk output mempunyai label yaitu lambat, sedang, dan cepat. Fungsi keanggotaan input dan output mempunyai bentuk representasi kurva segitiga serta mempunyai batasan-batasan fungsi keanggotaan yang akan memenuhi persyaratan target keluaran seperti pada tabel berikut:

**Tabel 3.1** Target output pada kontroller logika *fuzzy*

		Masukan Delta <i>Error</i> ( $\Delta E$ )		
		NDE	ZDE	PDE
Masukan <i>Error</i> (E)	NE	Lambat	Lambat	Lambat
	ZE	Sedang	Sedang	Cepat
	PE	Cepat	Cepat	Cepat

### 3.2. Simulasi Pengontrolan Kecepatan Motor

Simulasi merupakan suatu teknik meniru operasi-operasi atau proses-proses yang terjadi dalam suatu sistem dengan bantuan perangkat komputer dan dilandasi oleh beberapa asumsi tertentu sehingga sistem tersebut bisa dipelajari secara ilmiah. Simulasi yang dilakukan pada

perancangan ini menggunakan Matlab dengan versi 7.9. Simulasi akan dibagi pada 2 tahapan pengontrolan yaitu :

1. Simulasi kontrol dengan PID
2. Simulasi kontrol dengan FLC

### 3.3. Analisa Kinerja Sistem

#### 3.3.1. Analisa pengontrolan kecepatan motor dc dengan PID

Pada analisa ini pengujian dilakukan pada jenis motor DC magnet permanen dengan 2 variasi referensi kecepatan motor DC yang berbeda. Penentuan nilai parameter input  $K_p$ ,  $K_i$ , dan  $K_d$  dilakukan dengan metode *trial and error* pada tabel dibawah ini.

**Tabel 3.2** Parameter Nilai Input  $K_p$ ,  $K_i$ , dan  $K_d$

No	Set Point (Rpm)	Input		
		$K_p$	$K_i$	$K_d$
1				
2				
3				
4				
5				

Hasil dari pengujian ini adalah berupa respon sistem berbentuk grafis dan data yang diambil secara online. Grafis dan data tersebut dapat disimpan dalam bentuk file yang nantinya dapat digunakan untuk menentukan nilai  $t_r$  dan  $t_s$  dari respon sistem.

**Tabel 3.3** Tabel Analisa Respon Kecepatan Motor DC Dengan Kontrol PID

Set point (RPM)	Parameter PID Kontroller			$T_r$ (det)	$T_s$ (det)	MP (%)	Ess (%)
	$K_p$	$K_i$	$K_d$				

### 3.3.2. Analisa pengontrolan kecepatan motor DC dengan FLC

Semua referensi diujikan batasan-batasan dan jumlah label fungsi keanggotaan yang sama seperti pengujian pada kontrol PID diatas, lalu data tersebut diuji dengan metode defuzifikasi COA dengan aturan logika *fuzzy* proses pengontrolan kecepatan motor DC seperti pada tabel dibawah ini.

**Tabel 3.4** Aturan logika *fuzzy* pada kontroler FLC

Set point (RPM)	FIS Editor						Kecepatan Putar (RPM)	Rule Editor (if then)
	Input			Output				
	Error (RPM)			$\Delta$ error (RPM)				
	N E	Z E	P E	N D E	Z D E	P D E		

Hasil dari pengujian ini adalah berupa respon sistem berbentuk grafis dan data yang diambil secara online seperti pada pengujian controller PID. Grafik dan data tersebut dapat disimpan dalam bentuk file yang nantinya dapat digunakan untuk menentukan nilai  $t_r$  dan  $t_s$  dari respon sistem seperti pada tabel 3.3.

## 4. HASIL DAN PEMBAHASAN

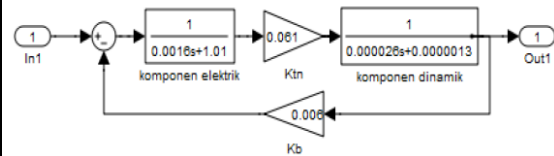
### 4.1. Permodelan matematika motor DC

Pada rancangan ini motor DC yang digunakan adalah type PITTMAN Series GM 14900 dengan spesifikasi motor yang digunakan terlihat tabel dibawah ini.

**Tabel 4.1** Spesifikasi motor DC PITTMAN Series

Notasi	Keterangan	Nilai	Satuan
$K_m$	Konstantan torsi motor	0.061	$Nm/A$
$K_b$	Konstanta tegangan ballik emf	0.06	$V/rpm$
$R_m$	Tahanan motor	1.01	$Ohm$
$L_m$	Induktansi motor	0.0016	$H$
$J_{eff}$	Momen inersia rotor dan beban	0.000026	$Kgm^2$
$f_{eff}$	Koefisien viskous rotor dan beban	0.000012	$Nm/rpm$

Bila fungsi kontrol tertutup pada gambar 2.3 diberikan nilai yang telah diketahui diatas, maka model pengontrolannya pada simulink matlab akan terlihat seperti pada gambar:



**Gambar 4.1** Model matematik motor DC PITTMAN Series pada simulink matlab

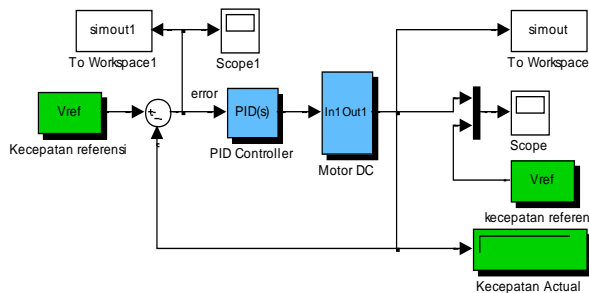
### 4.2. Simulasi kontrol kecepatan motor DC dengan PID

Kendali kecepatan motor DC dengan kontroler PID sangat dipengaruhi oleh kontribusi besar dari Penyetelan konstanta  $K_p$ ,  $K_i$ , dan  $K_d$ . pada penelitian ini, parameter  $K_p$ ,  $K_i$  dan  $K_d$  dicari dengan metode *trial and error* dengan urutan pencarian sbb:

1. Penalaan parameter kontroler dimulai dengan hanya menggunakan kontroler *Proportional*, kemudian baru ditambahkan kontroler *Integral*.
2. Penalaan selanjutnya kontroler *Proportional* hanya ditambahkan kontroler *Derivatif* saja.

3. Penalaan terakhir semua parameter kontroler *Proportional*, *Integral* dan *Derivatif* dimasukan nilainya.
4. Melakukan simulasi penalaan parameter kendali PID untuk aplikasi sistem pengendalian kecepatan motor DC dengan memasukkan nilai model matematika keseluruhan yang telah dihitung sebelumnya.

Pada pembahasan ini akan dilakukan pengujian referensi kecepatan motor DC pada *setpoint* 1400 RPM dan 1800 RPM. Diagram simulasi pengontrolan kecepatan motor dengan kontroler PID pada kondisi dapat dilihat pada gambar berikut:



**Gambar 4.2** Blok diagram simulink kontrol PID pada motor DC magnet permanen

Pengaturan kecepatan motor DC dilakukan dengan memasukkan nilai *setpoint* berupa referensi kecepatan maka sistem kendali ini akan membaca setiap kecepatan aktual keluaran sebagai sinyal umpan balik. Untuk umpan masukan dan umpan balik *error* berupa kecepatan yang masuk kedalam pengendali PID untuk menghitung nilai *error* dengan nilai parameter  $K_p$ ,  $K_i$  dan  $K_d$  yang ideal, setelah itu output dari PID digunakan untuk mengontrol kecepatan motor DC.

### 4.3. Simulasi kontrol kecepatan motor DC dengan FLC

Sistem logika *fuzzy* yang dirancang dengan 2 input yaitu variabel *error* dan delta *error* serta 1 keluaran yaitu variabel kecepatan. Fungsi keanggotaan *error* dan  $\Delta error$  kecepatan mempunyai 3 label yaitu untuk input *error* yaitu NE (*Negatif Error*), ZE (*Zero Error*), PE (*Positif Error*) dan untuk  $\Delta error$  yaitu NDE (*Negatif Delta Error*), ZDE (*Zero Delta Error*), dan PDE (*Positif Delta Error*).

Untuk variabel output mempunyai label yaitu lambat, sedang dan cepat. Untuk nilai input *error* dan keluaran kecepatan dipetakan dalam semesta pembicaraan yang ditetapkan berdasarkan metode *trial and error*.

Pada pembahasan ini akan dilakukan pengujian referensi kecepatan motor DC pada *setpoint* 1400 RPM dan 1800 RPM. Sebagai input menggunakan variabel *error* dan  $\Delta error$ . Untuk *error* dengan batasan keanggotaan yang dikelompokkan menjadi tiga bagian yaitu :

- NE antara  $[(-5)-1]$  RPM
- ZE antara  $[(-2)-2]$  RPM
- PE antara  $[1-5]$  RPM

Untuk  $\Delta error$  dengan batasan keanggotaan yang dikelompokkan menjadi tiga bagian yaitu :

- NDE antara  $[(-2)-0]$  RPM
- ZDE antara  $[(-1)-1]$  RPM
- PDE antara  $[0-2]$  RPM

Dalam bentuk Tabel batasan keanggotaan diatas direpresentasikan pada tabel 4.2.a & 4.2.b.



**Tabel 4.2.a.** Batasan-batasan fungsi keanggotaan *error* pada representasi kurva segitiga

Set point (RPM)	Label Fungsi keanggotaan Error	A (rpm)	B (rpm)	C (rpm)
1400RPM dan 1800RPM	NE	-5	-2	1
	ZE	-2	0	2
	PE	-1	2	5

**Tabel 4.2.b.** Batasan-batasan fungsi keanggotaan  $\Delta error$  pada representasi kurva segitiga

Set point (RPM)	Label Fungsi keanggotaan $\Delta Error$	A (rpm)	B (rpm)	C (rpm)
1400RPM dan 1800RPM	NDE	-2	-1	0
	ZDE	-1	0	1
	PDE	0	1	2

Untuk *setpoint* 1400 RPM variabel kecepatan sebagai keluaran mempunyai batasan keanggotaan yang dikelompokkan menjadi tiga bagian yaitu :

- Lambat antara  $[(-163)-163]$  RPM
- Sedang antara  $[0-326]$  RPM
- Cepat  $[163-489]$  RPM.

Sedangkan pada *setpoint* 1800 RPM untuk keluaran kecepatan mempunyai batasan keanggotaan yang dikelompokkan menjadi tiga bagian yaitu :

- Lambat antara  $[(-210)-210]$  RPM
- Sedang antara  $[0-415]$  RPM
- Cepat  $[210-630]$  RPM.

Dalam bentuk Tabel batasan keanggotaan diatas direpresentasikan pada tabel 4.3.

**Tabel 4.3.** Batasan-batasan fungsi keanggotaan keluaran pada representasi kurva segitiga

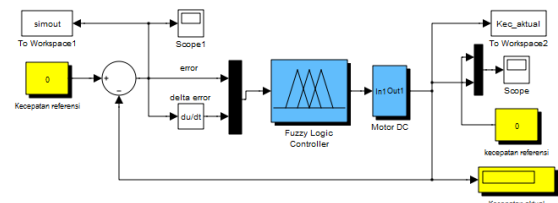
Set point (RPM)	Label Fungsi keanggotaan Keluaran	A (rpm)	B (rpm)	C (rpm)
1400RPM	Lambat	-163	0	163
	Sedang	0	163	326

1800RPM	Cepat	163	326	489
	Lambat	-210	0	210
	Sedang	0	210	415
	Cepat	210	415	630

Jumlah *rule fuzzy* yang digunakan pada sistem ini terdiri dari 9 rule yaitu :

1. If (Error is NE) and (delta\_error is NDE) then (Kecepatan is lambat)
2. If (Error is NE) and (delta\_error is ZDE) then (Kecepatan is lambat)
3. If (Error is NE) and (delta\_error is PDE) then (Kecepatan is lambat)
4. If (Error is ZE) and (delta\_error is NDE) then (Kecepatan is sedang)
5. If (Error is ZE) and (delta\_error is ZDE) then (Kecepatan is sedang)
6. If (Error is ZE) and (delta\_error is PDE) then (Kecepatan is cepat)
7. If (Error is PE) and (delta\_error is NDE) then (Kecepatan is cepat)
8. If (Error is PE) and (delta\_error is ZDE) then (Kecepatan is cepat)
9. If (Error is PE) and (delta\_error is PDE) then (Kecepatan is cepat)

Logika pengambilan keputusan menggunakan metode Mamdani untuk mengkombinasi aturan yang terdapat pada basis suatu pemetaan dari suatu himpunan *fuzzy input* ke himpunan *fuzzy output*. Tahap terakhir adalah proses defuzzifikasi yang dilakukan dengan menggunakan metode *Centre Of Area* (COA). Gambar diagram simulasi pengontrolan kecepatan motor dengan kontroler dapat dilihat pada gambar berikut.



**Gambar 4.3** Blok diagram simulink kontrol FLC pada motor DC magnet permanen

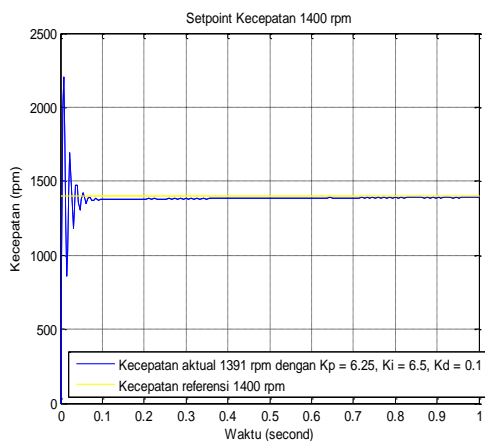
#### 4.4. Analisa kontrol kecepatan motor DC dengan PID

Simulasi pengontrolan kecepatan motor DC dengan kontroler PID bertujuan untuk menentukan nilai  $K_p$ ,  $K_i$ , dan  $K_d$  yang terbaik untuk memperoleh respon terbaik dari sistem. Dengan menggunakan metode *trial and error*, dicoba beberapa kombinasi nilai  $K_p$ ,  $K_i$ , dan  $K_d$  seperti pada tabel dibawah ini.

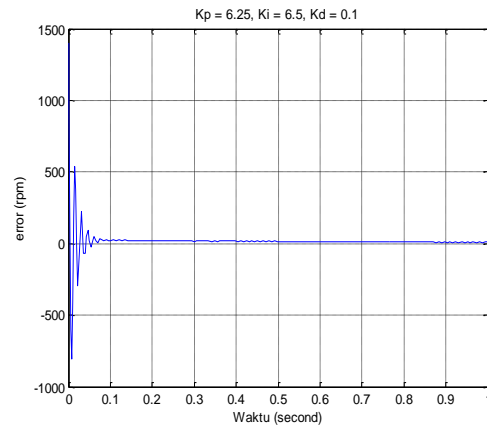
**Tabel 4.4.** Parameter nilai input  $K_p$ ,  $K_i$ , dan  $K_d$  pada *setpoint* 1400 dan 1800 RPM

Set point (RPM)	Parameter PID		
	$K_p$	$K_i$	$K_d$
1400RPM	6.25	6.5	0.1
	1	5	0.05
	0.2	5.56	0.001
1800RPM	6.25	6.5	0.1
	1	5	0.05
	0.2	5.56	0.001

Respon penalaan parameter kontrol PID sistem pengendali kecepatan motor DC dengan  $K_p = 6.25$ ,  $K_i = 6.5$ , dan  $K_d = 0.1$  pada *setpoint* 1400 RPM terlihat pada gambar berikut.



(a)



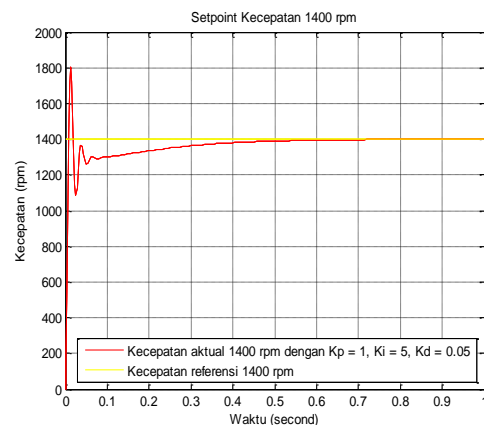
(b)

**Gambar 4.3. (a) & (b)** Grafik kecepatan aktual dan perubahan kecepatan error motor terhadap waktu dengan nilai parameter  $K_p = 6.25$ ,  $K_i = 6.5$  dan  $K_d = 0.1$  pada *setpoint* 1400 RPM

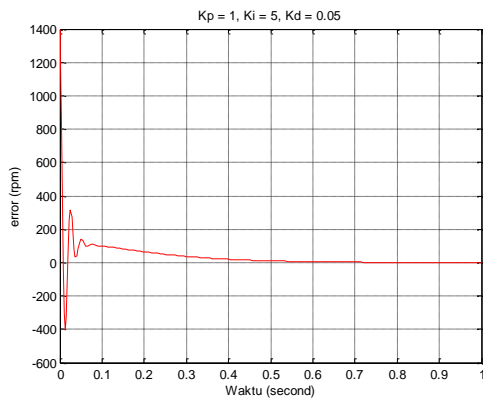
Dari gambar terlihat hasil dari kinerja respon motor DC pengendali PID dengan nilai parameter  $K_p = 6.25$ ,  $K_i = 6.5$  dan  $K_d = 0.1$  pada *setpoint* 1400 RPM yang menghasilkan kecepatan aktual 1391 RPM sampai waktu 1 detik dengan respon grafik sistem sebagai berikut :

- *Rise time* ( $T_r$ ) = 0.00287 det
- *Settling time* ( $T_s$ ) = 0.0792 det
- *Maximum overshoot* ( $MP$ ) = 59 %
- *Error steady state* ( $ess$ ) = 0.64 %

Respon penalaan parameter kontrol PID sistem pengendali kecepatan motor DC dengan  $K_p = 1$ ,  $K_i = 5$ , dan  $K_d = 0.05$  pada *setpoint* 1400 RPM



(a)



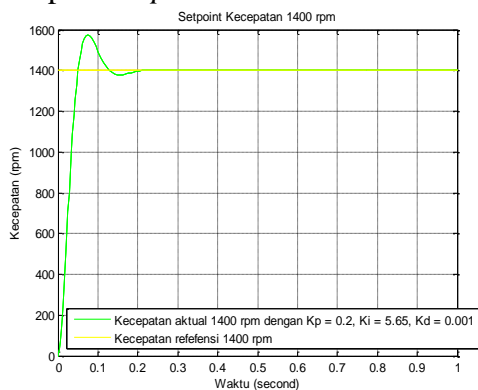
(b)

**Gambar 4.4 (a) & (b)** Grafik kecepatan aktual dan perubahan kecepatan error motor terhadap waktu dengan nilai parameter  $K_p = 1$ ,  $K_i = 5$  dan  $K_d = 0.05$  pada *setpoint* 1400 RPM

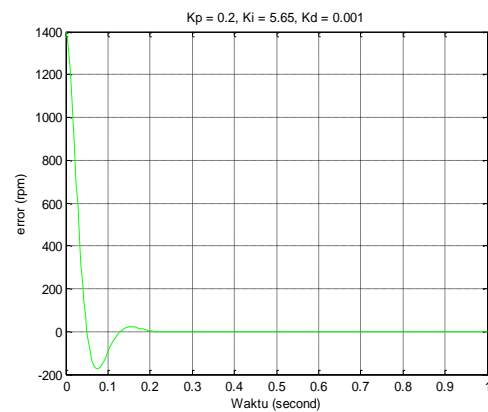
Kinerja respon motor DC pengendali PID dengan nilai parameter  $K_p = 1$ ,  $K_i = 5$  dan  $K_d = 0.05$  pada *setpoint* 1400 RPM yang menghasilkan kecepatan aktual 1400 RPM sampai waktu 1 detik dengan respon grafik sistem sebagai berikut :

- Rise time* ( $T_r$ ) = 0.00545 det
- Settling time* ( $T_s$ ) = 0.347 det
- Maximum overshoot* ( $MP$ ) = 28.7 %
- Error steady state* ( $ess$ ) = 0 %

Respon penalaan parameter kontrol PID sistem pengendali kecepatan motor DC dengan  $K_p = 0.2$ ,  $K_i = 5.65$ , dan  $K_d = 0.001$  pada *setpoint* 1400 RPM



(a)



(b)

**Gambar 4.5 (a) & (b)** Grafik kecepatan aktual dan perubahan kecepatan error motor terhadap waktu dengan nilai parameter  $K_p = 0.2$ ,  $K_i = 5.65$  dan  $K_d = 0.001$  pada *setpoint* 1400 RPM

Kinerja respon motor DC pengendali PID dengan nilai parameter  $K_p = 0.2$ ,  $K_i = 5.65$  dan  $K_d = 0.001$  pada *setpoint* 1400 RPM yang menghasilkan kecepatan aktual 1400RPM sampai waktu 1 detik dengan respon grafik sistem sebagai berikut :

- Rise time* ( $T_r$ ) = 0.0342 detik
- Settling time* ( $T_s$ ) = 0.118 detik
- Maximum overshoot* ( $MP$ ) = 12.5 %
- Error steady state* ( $ess$ ) = 0 %

Cara yang sama dengan metode diatas kita lakukan pada *setpoint* 1800 RPM, maka kita akan dapatkan data Respon *Rise time*, *Settling time*, *Overshoot*, dan *Error* untuk kedua *setpoint* adalah sebagai berikut:

**Tabel 4.5.** Hasil analisa respon kecepatan motor DC dengan PID pada *setpoint* 1400 dan 1800 RPM

Setpoint (RPM)	Parameter PID			$T_r$ (det)	$T_s$ (det)	$Mp$ (%)	$Ess$ (%)
	$K_p$	$K_i$	$K_d$				
1400	6.25	6.5	0.1	0.00287	0.0792	59	0.64
	1	5	0.05	0.00545	0.347	28.7	0

	0.2	5.65	0.001	0.0342	0.118	12.5	0
1800	6.25	6.5	0.1	0.00287	0.0792	59	0.72
	1	5	0.05	0.00545	0.347	28.7	0
	0.2	5.65	0.001	0.0342	0.118	12.5	0

Tabel diatas menunjukkan bahwa untuk nilai parameter  $K_p = 0.2$ ,  $K_i = 5.6$ , dan  $K_d = 0.001$  merupakan konstanta nilai parameter PID yang efektif untuk mendapatkan respon sistem yang terbaik dalam pengontrolan kecepatan motor DC pada *setpoint* 1400 RPM dengan menggunakan kontroler PID yang menghasilkan respon sistem untuk waktu naik ( $T_r$ ) = 0.0342 detik, waktu penetapan ( $T_s$ ) = 0.118 detik, *Overshoot* ( $M_p$ ) = 12.5 % dan *Error steady state* = 0 %, sedangkan pada *setpoint* 1800 RPM dengan menggunakan kontroler PID menghasilkan respon sistem untuk waktu naik ( $T_r$ ) = 0.0342 detik, waktu penetapan ( $T_s$ ) = 0.118 detik, *Overshoot* ( $M_p$ ) = 12.5 % dan *Error steady state* = 0 %.

#### 4.5. Analisa kontrol kecepatan motor DC dengan FLC

Simulasi pengontrolan kecepatan motor DC dengan kontroler FLC bertujuan untuk menentukan aturan logika *fuzzy* sehingga diperoleh respon sistem yang paling baik. Untuk memperoleh respon yang terbaik dari sistem dengan mencoba beberapa kombinasi aturan logika *fuzzy*. Pada simulasi digunakan aturan logika *fuzzy* seperti pada tabel dibawah ini.

**Tabel 4.6.** Aturan logika *fuzzy* pada kontroler FLC pada *setpoint* 1400 dan 1800 RPM

Setpoint (RPM)	FIS Editor									Rule Editor (if-then)
	Input						Output			
	Error (RPM)			$\Delta$ Error (RPM)			Kecepatan (RPM)			
	NE	ZE	PE	NDE	ZDE	PDE	Lambat	Sedang	Cepat	
1400	-5 ~ 1	-2 ~ 2	1 ~ 5	-2 ~ 0	-1 ~ 1	0 ~ 2	-163 ~ 163	0 ~ 326	163 ~ 489	1. If (Error is NE) and (delta_error is NDE) then (Kecepatan is lambat) 2. If (Error is NE) and (delta_error is ZDE) then (Kecepatan is lambat) 3. If (Error is NE) and (delta_error is PDE) then (Kecepatan is lambat) 4. If (Error is ZE) and (delta_error is NDE) then (Kecepatan is sedang) 5. If (Error is ZE) and (delta_error is ZDE) then (Kecepatan is sedang) 6. If (Error is ZE) and (delta_error is PDE) then (Kecepatan is cepat) 7. If (Error is PE) and (delta_error is NDE) then (Kecepatan is cepat) 8. If (Error is PE) and (delta_error is ZDE) then (Kecepatan is cepat) 9. If (Error is PE) and (delta_error is PDE) then (Kecepatan is cepat)
1800	-5 ~ 1	-2 ~ 2	1 ~ 5	-2 ~ 0	-1 ~ 1	0 ~ 2	-210 ~ 210	0 ~ 415	210 ~ 630	

Untuk mendapatkan analisa respon sistem pada *setpoint* 1400 RPM dan 1800 RPM dari data pada tabel diatas, maka data tersebut dimasukkan dan disimulasikan dengan *simulink* matlab. Respon sistem terhadap kontroler FLC dengan data diatas dengan metode defuzzifikasi *Centre Of Area* (COA) adalah sebagai berikut:

**Tabel 4.7.** Aturan logika *fuzzy* pada kontroler FLC pada *setpoint* 1400 dan 1800 RPM

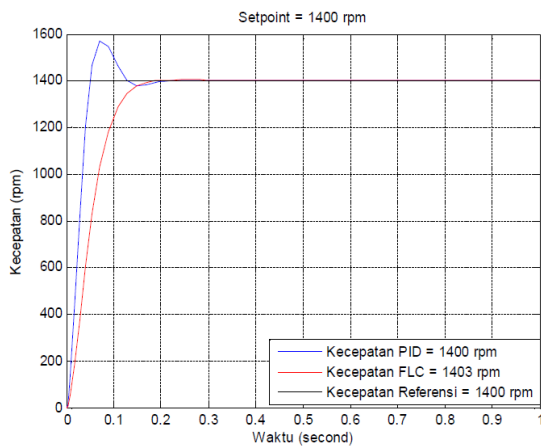
Setpoint (RPM)	Metode Defuzzifikasi			
	COA			
	$T_r$ (det)	$T_s$ (det)	$M_p$ (%)	Ess (%)
1400	0.06	0.17	0	0.21
1800	0.05	0.18	0	0.56

Tabel diatas menunjukkan hasil analisa untuk *setpoint* 1400 dan 1800 RPM menghasilkan aturan logika *fuzzy* yang efektif dan ideal. *Setpoint* 1400 RPM menghasilkan respon sistem dengan nilai  $T_r = 0.06$  detik,  $T_s = 0.17$  detik, *Overshoot* ( $M_p$ ) = 0 % dan *Error steady state* (ess) = 0.21 %. Sedangkan untuk *setpoint* 1800 RPM menghasilkan respon sistem dengan

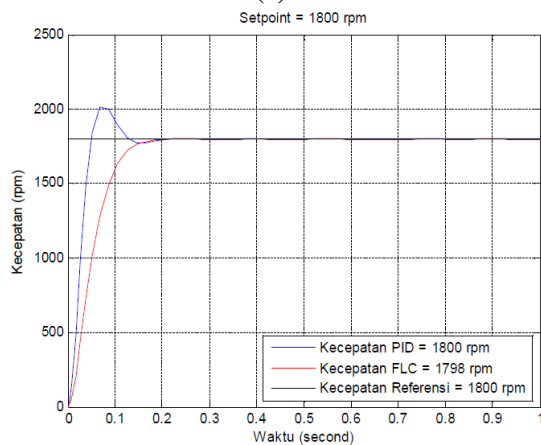
nilai  $T_r = 0.05$  detik,  $T_s = 0.18$  detik,  $Overhoot (M_p) = 0\%$  dan  $Error\ steady\ state (ess) = 0.56\%$ .

#### 4.6. Perbandingan Kinerja Kontroler PID dan FLC

Perbandingan hasil respon sistem yang terbaik dari masing-masing pengontrolan untuk kontroler PID dengan kontroler FLC terlihat pada grafik berikut:



(a)



(b)

**Gambar 4.6 (a) & (b)** Grafik perbandingan kontroler PID dan FLC pada  $setpoint$  1400 rpm dan 1800 rpm

**Tabel 4.8.** Analisa perbandingan respon sistem kontroler PID dan FLC

Setpoint (RPM)	Actual Speed (RPM)	PID Controller				Parameter PID		
		$T_r$ (det)	$T_s$ (det)	$M_p$ (%)	$Ess$ (%)	$K_p$	$K_i$	$K_d$
1400	1400	0.0342	0.118	12.5	0	0.2	5.65	0.001

1800	1800	0.0342	0.118	12.5	0	0.2	5.65	0.001
		FLC Controller						
1400	1403	0.06	0.17	0	0.21			
1800	1798	0.05	0.18	0	0.56			

Terlihat pada grafik dan table diatas bahwa kontroler PID masih tetap memiliki *overshoot* pada permulaan sistem namun dikoreksi dengan cepat pada detik berikutnya dan stabil pada kecepatan yang dikehendaki sedangkan pada kontroler FLC respon tidak didapati *overshoot* namun kecepatan yang didapat tidak 100% sama dengan *setpoint*.

## 5. KESIMPULAN DAN SARAN

### 5.1. Kesimpulan

- Hasil perancangan simulasi pengontrolan kecepatan motor DC dengan PID pada *setpoint* 1400 dan 1800 RPM mendapatkan nilai parameter control PID yang efektif untuk  $K_p = 0.2$ ,  $K_d = 5.65$  dan  $K_i = 0.001$  dengan respon transien untuk *rise time* ( $T_r$ ) = 0.03 detik, *settling time* ( $T_s$ ) = 0.12 detik, *maximum overshoot* ( $M_p$ ) = 12.5 %, dan *error steady state* ( $ess$ ) = 0 %.
- Hasil perancangan simulasi pengontrolan kecepatan motor DC dengan FLC pada *setpoint* 1400 dan 1800 RPM mendapatkan aturan logika *fuzzy* seperti yang ditunjukkan tabel 5.3 pada *setpoint* 1400 RPM dengan respon transien untuk nilai  $T_r = 0.06$  detik,  $T_s = 0.17$  detik,  $Overhoot (M_p) = 0\%$  dan  $Error\ steady\ state (ess) = 0.21\%$ . Pada *setpoint* 1800 RPM respon transien untuk  $T_r = 0.05$  detik,  $T_s = 0.18$  detik,  $Overhoot (M_p) = 0\%$  dan  $Error\ steady\ state (ess) = 0.56\%$ .

3. Perbandingan kinerja pengontrolan PID dan FLC pada *setpoint* 1400 dan 1800 RPM untuk waktu naik ( $T_r$ ) dan waktu penetapan ( $T_s$ ) lebih cepat menggunakan kontrol PID dibanding FLC, pada kontrol PID masih terjadi *overshoot* 12.5 % pada *setpoint* 1400 RPM dan 1800 RPM sedangkan FLC tidak ada *overshoot* dan PID tidak mengalami *steady state error* sedangkan FLC masih mempunyai *steady state error* 0.21 % pada *setpoint* 1400 RPM dan 0.56 % pada *setpoint* 1800 RPM.

## 5.2. Saran

1. Dalam pemodelan sistem tingkat kerumitan dan kespesifikan dari aspek fisik yang didefinisikan sangat menentukan akurasi dari hasil yang diperoleh. Dalam hal ini menyarankan untuk dapat meningkatkan tingkat kerumitan pemodelan sistemnya, sehingga fungsi alih yang dihasilkan lebih akurat dan mewakili gambaran sistem sesungguhnya.
2. Pendekatan dasar desain sistem kontrol praktis adalah prosedur coba-coba, maka tentunya terdapat kekurangan dalam menghasilkan desain yang lebih tepat,

tergantung pada intensitas dan lama waktu proses coba-coba dilakukan. Sehingga semakin sering intensitas coba-coba dilakukan, maka performa sistem yang diperoleh dari desain akan lebih baik.

## DAFTAR PUSTAKA

- [1]. Charles L. Phillips & Royced D. Harbor. Alih bahasa Prof. R. J. Widodo, Dasar-Dasar Sistem Kontrol. Jilid 1. Jakarta : Prenhallindo. 1996
- [2]. Ogata, Katsuhiko. Modern Control Engineering. 3rd ed. Prentice Hall International. 1997
- [3]. Ogata, Katsuhiko. Solving Control Engineering Problems with MatLab. Englewood Cliffs, New Jersey : Prentice Hall Inc. 1994
- [4]. Astrom, K. & Hagglund, T., PID Controllers: Theory, Design, and Tuning, 2ed, Instrument Society of America (ISA), USA, 1995.
- [5]. Sutejo, T. , Mulyanto, E. & Suhartono, V. 2011. Kecerdasan Buatan. Yogyakarta: Andi Offset.
- [6]. Jun Y., Michail R., James P., Using Fuzzy Logic, 1993.