

RANCANG BANGUN PENGENDALIAN KECEPATAN *BRUSHLESS* DC MOTOR TIPE A2212/10T 1400 KV MENGGUNAKAN KONTROLER PID BERBASIS LABVIEW

Fauzi Yuda Pratama

S1 Teknik Elektro, Fakultas Teknik, Universitas Negeri Surabaya
e-mail : yuda.tama1995.yt@gmail.com

Endryansyah

Dosen Teknik Elektro, Fakultas Teknik, Universitas Negeri Surabaya
e-mail : syahryanend@yahoo.com

Abstrak

Pada dunia industri, motor dc sangat berperan penting dalam menjalankan suatu proses dalam sebuah industri. Salah satu contohnya adalah motor dc pada mesin bubut konvensional ataupun mesin bubut modern yang sudah diterapkan CNC dalam sistemnya. Kekurangan dari motor dc konvensional adalah *brush* yang cepat rusak jika motor dc digunakan secara 24 jam *non-stop*. Untuk memperbaiki masalah tersebut, maka terdapat alternative lain yaitu *brushless* dc motor yang memiliki kecepatan yang tinggi dan efisiensi yang lebih baik jika dibandingkan dengan motor dc konvensional. Tujuan penelitian ini adalah untuk merancang dan mengontrol kecepatan dari *brushless* dc motor menggunakan kontroler PID. Penulis menggunakan pendekatan karakteristik respon sistem orde 2 yang memiliki parameter K , ζ , dan ω_n . Penelitian ini menggunakan *software* Labview 2017 yang digunakan untuk mengontrol parameter seperti K_p , K_i , K_d , Jenis kontroler yang digunakan, beban simulasi yang digunakan dan sebagai monitoring kecepatan dari *brushless* dc motor. Hasil penelitian menunjukkan bahwa respon kecepatan dari *brushless* dc motor menggunakan kontroler PID mampu mencapai *setpoint* yang diinginkan dengan $\tau = 1.089$ detik dan *error steady state* 4.3% (tanpa beban) dan $\tau = 1.20$ detik dan *error steady state* 3.7% (dengan beban). Respon ini menunjukkan perbaikan dari respon sistem tanpa kontroler dengan $\tau = 0.833$ detik dan *error steady state* 223.4%.

Kata Kunci: Motor DC Konvensional, *Brushless* DC Motor, Kontroler PID, Labview.

Abstract

Nowadays, dc motors are very instrumental in running a process within an industry. One example is the dc motor on Conventional lathes or modern lathes that use CNC in its system. The weakness of the conventional dc motor is the brush that prone to damage if used 24 hours non-stop. To fix the problem, brushless dc motor that has high speed and better efficiency compared to the conventional dc motors is created. The purpose of this research is to design and control the speed of a brushless dc motor using PID controllers. For system modeling plant, the author uses system response characteristics of order 2 approach which has parameters of K , ζ , and ω_n . This research uses Labview 2017 software to control parameters such as K_p , K_i , K_d , type of controller is used, the simulation load that used as monitoring the speed of a brushless dc motor. So Labview is the Control Center of the system. The results showed that the response speed of brushless dc motor using PID controller is able to achieve the desired setpoint with $\tau = 1,089$ seconds and steady state error of 4.3% (without load) and $\tau = 1.20$ seconds and steady state error 3.7% (with the load). This response showed an improvement of system response without the controller with $\tau = 0833$ seconds and steady state error 223.4 %.

Keywords: Conventional DC Motors, Brushless DC Motors, PID Controller, Labview.

PENDAHULUAN

Seiring dengan perkembangan zaman, kebutuhan akan motor yang memiliki efisiensi tinggi, torsi yang tinggi, kecepatan yang tinggi dan dapat divariasikan, dan biaya perawatan yang rendah semakin meningkat. Hanya saja motor yang digunakan secara umum saat ini, yakni motor DC dan motor induksi, belum dapat memenuhi kebutuhan tersebut. Motor DC memiliki efisiensi yang tinggi karena penggunaan tegangan DC pada rotor untuk menggerakkan motor tersebut. Hanya saja motor DC memiliki biaya perawatan yang tinggi. Biaya perawatan yang tinggi ini muncul akibat digunakannya *brush* dalam

komutasi motor DC. *Brush* pada motor DC ini cepat mengalami kerusakan (Saputra, 2013)

Hal ini terjadi karena pada saat motor berputar, pada *brush*, akan timbul *arching* akibat proses komutasi. Oleh karena itu untuk memenuhi kebutuhan akan efisiensi tinggi, torsi yang tinggi, kecepatan yang tinggi dan dapat divariasikan, dan biaya perawatan yang rendah maka digunakan motor *brushless* DC motor (BLDC).

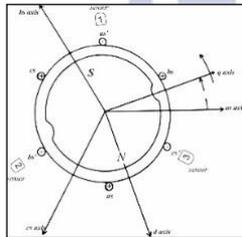
Fokus permasalahan yang ingin dipecahkan adalah pengaturan kecepatan *brushless* DC motor agar tetap konstan saat terjadi pemotongan dengan menggunakan kontroler PID. Harapannya, algoritma kontrol ini akan

dapat dijadikan referensi dalam perancangan kontrol pada *brushless* DC motor..

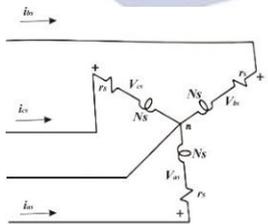
KAJIAN PUSTAKA

Brushless DC Motor

Brushless DC motor adalah motor sinkron dengan magnet permanen yang menggunakan sensor posisi dan sebuah rangkaian elektronika (*inverter*) untuk mengendalikan arus *armature* yang ada. Motor brushless DC kadangkala dimodelkan sebagai suatu motor DC 2 arah karena *armature*-nya ada di stator dan magnet berada pada rotor dan karakteristik operasinya mirip dengan motor DC. Daripada menggunakan suatu komutator mekanik seperti yang ada pada motor dc konvensional, motor brushless DC menggunakan komutasi elektronik yang menjadikannya motor yang minim perawatan. (Azzumar, 2012). Gambar 1 dan Gambar 2 menunjukkan struktur dan rangkaian ekuivalen dari suatu *brushless* DC motor. Sensor posisi biasanya menggunakan 3 sensor efek Hall yang mendeteksi kehadiran magnet kecil yang terpasang pada shaft motor



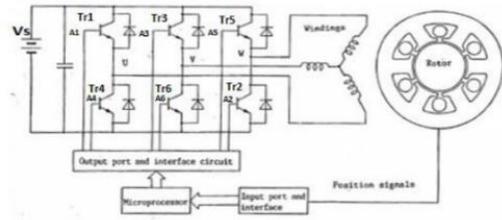
Gambar 1 Struktur *Brushless* DC Motor (Sumber : Krause, 2002)



Gambar 2 Rangkaian Ekuivalen *Brushless* DC Motor (Sumber : Krause, 2002)

Brushless DC Motor Driver (Inverter)

Cara kerja *inverter*, yaitu setiap transistor aktif selama interval 180° terhadap siklus gelombang, hanya waktu pengaktifan transistor masing – masing fasa dibuat berbeda 120°. Hal ini menggambarkan adanya tiga buah fasa dari tegangan yang masing – masing berbeda 120°. Pergeseran fasa ini dihasilkan dalam urutan pengaktifan, dimana transistor diaktifkan pada suatu interval yang teratur dan terus menerus dari 60° secara berurutan, yaitu Tr1, Tr2, Tr3, Tr4, Tr5, dan Tr6, untuk menyelesaikan satu siklus dari gelombang tegangan keluaran (360°). (Azzumar, 2012). Adapun rangkaian *inverter* tiga fasa diberikan oleh Gambar 3



Gambar 3. Skema umum *driver Brushless* DC Motor (Sumber : Azzumar, 2012)

di mana

Tr = transistor

A = pulsa trigger

Tachogenerator

Sensor yang sering digunakan untuk sensor kecepatan angular adalah *tachogenerator*. *Tachogenerator* adalah sebuah generator kecil yang membangkitkan tegangan DC ataupun tegangan AC. Dari segi eksitasi *tachogenerator* dapat dibangkitkan dengan eksitasi dari luar atau imbas elektromagnetik dari magnet permanen. *Tachogenerator* DC dapat membangkitkan tegangan DC yang langsung dapat menghasilkan informasi kecepatan (Ridwan, 2012)

Software Labview

Laboratory virtual instrument engineering workbench (LabVIEW) adalah suatu software yang digunakan untuk pemrosesan dan visualisasi data dalam bidang akuisisi data, kendali dan instrumentasi, serta otomatisasi industri. Kelebihan LabVIEW dibandingkan dengan bahasa pemrograman lainnya adalah bahasanya yang mudah dipahami, pembuatan program mudah, waktu pembuatan program singkat, memiliki integrasi dengan ribuan *hardware* dan ratusan *library* yang siap digunakan, dapat menangani beberapa instruksi sekaligus, bersifat modular, telah terbukti andal, *powerfull*, dan fleksibel.

Analisis Karakteristik Sistem Orde dua

Sistem orde dua mempunyai fungsi alih dengan pangkat s tertinggi dua. Karakteristik respon waktu untuk sistem orde kedua ditunjukkan pada Gambar 4, respon sistem ini didapatkan dengan mengamati respon sistem orde kedua terhadap sinyal uji step. (Shafiudin, 2016). Model sistem orde satu secara matematis dapat dituliskan sebagai berikut :

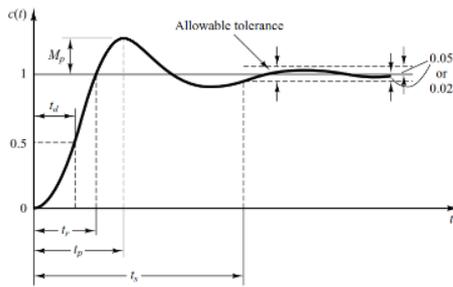
$$\frac{C(s)}{R(s)} = \frac{K\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{1}$$

Keterangan :

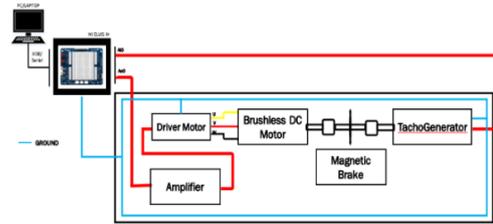
K = Gain Overall (V)

ζ = Rasio Redaman (dB)

ω_n = Frekuensi Alami (rad/s)



Gambar 4. Karakteristik respon sistem orde dua terhadap masukan *unit-step* (Sumber: Ogata, 2010)



Gambar 6. Rancang Bangun *Hardware* Pengendalian Kecepatan *Brushless* DC motor menggunakan Kontroler PID Berbasis Labview (Sumber: Dokumen Pribadi, 2018)

METODE PENELITIAN

Pendekatan Penelitian

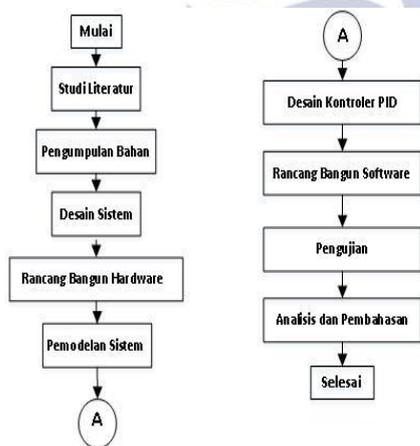
Pada penelitian ini menggunakan *software* Labview 2017 untuk simulasi serta menjalankan sistem PID yang akan digunakan dan merancang bangun *plant* yang akan menghasilkan respon kecepatan pada *brushless* DC motor tipe A2212/10T 1400KV

Tempat dan Waktu Penelitian

Penelitian dilaksanakan di Laboratorium Kendali Jurusan Teknik Elektro Universitas Negeri Surabaya dan waktu pelaksanaannya dilakukan pada semester genap 2017/2018.

Rancangan Penelitian

Rancangan penelitian ini bertujuan untuk menjawab permasalahan dalam rangka merumuskan kesimpulan, seperti dijelaskan pada Gambar 5. diagram alir sebagai berikut:



Gambar 5. Perancangan Penelitian (Sumber : Dokumen Pribadi, 2018)

Rancang Bangun *Hardware*

Rancang bangun *hardware* yang dimaksud adalah pemasangan kabel (*wiring*) dan hubungan antar komponen, sehingga semua komponen yang digunakan dapat terhubung dan dapat beroperasi sesuai dengan desain sistem. Rancang bangun *hardware* pada penelitian ini ditunjukkan pada Gambar 6.

Identifikasi *Plant*

Pada penelitian ini, fungsi alih didapatkan dengan cara pendekatan sistem orde-2. Sehingga untuk mendapatkan parameter-parameter dari karakteristik sistem orde-2 yaitu K , ω_n , dan ζ , dilakukan pengambilan data sebanyak 5 kali dengan *setpoint* berbeda-beda. Percobaan dilakukan dengan *loop* terbuka sehingga didapatkan respon sistem orde-2. Setelah didapatkan 5 fungsi alih dari setiap *setpoint*, selanjutnya fungsi alih tersebut akan disimulasikan dan dibandingkan dengan respon sistem sesungguhnya. Setelah dibandingkan, akan didapatkan *error* antar respon sesungguhnya dan respon simulasi. Fungsi alih dengan *error* rata-rata terkecil akan digunakan sebagai fungsi alih sistem dan menjadi acuan untuk mendesain kontroler PID.

Validasi model yang dimaksud adalah mencari fungsi alih dengan *error* terkecil. *Error* didapatkan dengan membandingkan output rata-rata *steady state* sistem sesungguhnya dengan output rata-rata *steady state* simulasi. *Input* pada proses tersebut adalah nilai tegangan dan outputnya adalah fungsi alih.

Data yang diterima dari tachogenerator akan disimpan dalam bentuk data excel. Setelah itu, data excel yang didapatkan akan diolah dan ditentukan parameter-parameter K , ω_n , dan ζ berdasarkan karakteristik sistem orde-2 dari tiap *setpoint* dengan rumus sebagai berikut :

$$K = \frac{Y_{ss}}{X_{ss}} \quad (2)$$

$$Mp = \frac{Y_p - Y_{ss}}{Y_{ss}} \quad (3)$$

$$\zeta = \frac{|\ln Mp|}{\sqrt{\pi^2 + (\ln Mp)^2}} \quad (4)$$

$$\omega_n = \frac{\pi}{t_p \sqrt{1 - \zeta^2}} \quad (5)$$

Keterangan :

X_{ss} = Input Tegangan (V)

Y_{ss} = Output Tegangan (V)

Y_p = Tegangan Puncak (V)

t_p = Waktu Puncak (s)

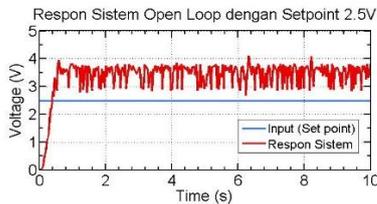
Mp = *Overshoot* Maksimum Tegangan Output (V)

Konfigurasi Kontrol PID

Pemodelan Sistem

Pemodelan sistem dilakukan dengan cara memberikan *setpoint* tegangan bervariasi yaitu 0.5V, 1V, 1.5V, 2V dan 2.5V dengan *sampling time* 0.03 detik dan lamanya pengambilan data 10 detik. Model matematika

yang didapatkan merupakan hubungan antara tegangan yang dibaca oleh *tachogenerator* dengan *setpoint* tegangan yang diinginkan. Pemodelan dilakukan pada kondisi tanpa beban. Berikut adalah respon sistem dengan *setpoint* 2.5V ditunjukkan pada Gambar 7.



Gambar 7. Respon Sistem Open Loop pada *setpoint* 2.5 V

(Sumber : Dokumen Pribadi, 2018)

Berdasarkan Gambar 7, maka dapat dihitung parameter dari karakteristik sistem orde-2 yaitu K , ζ , ω_n menggunakan persamaan (2), (3), (4), dan (5).

Pemilihan Fungsi Alih Sistem

Pemilihan fungsi alih dilakukan dengan menghitung nilai *error* rata-rata dari setiap fungsi alih. Fungsi alih dengan nilai *error* rata-rata terkecil akan digunakan sebagai acuan untuk mendesain kontroler PID. Berikut persamaan yang digunakan untuk menghitung *error* :

$$Error = |Output\ Simulasi - Output\ Real| \quad (6)$$

Output pada detik ke 0.03 simulasi akan dikurangi dengan *output* real pada detik ke 0.03 juga. Begitu seterusnya sampai detik ke-10. Sehingga didapatkan nilai *error* tiap kelipatan 0.03 detik. Setelah itu, nilai-nilai *error* tersebut akan di rata-ratakan dan dibandingkan dengan *error* rata-rata fungsi alih yang lain.

Desain Kontroler PID

Fungsi dari perancangan kontroler adalah memperbaiki respon sistem. Oleh karena itu, karakteristik *output* yang akan diperbaiki ditunjukkan pada Tabel 1.

Tabel 1. Karakteristik *Output* yang akan diperbaiki

Karakteristik yang akan diperbaiki
Waktu Tunak (<i>Settling Time</i>)
Waktu Naik (<i>Rise Time</i>)
Simpangan Maksimum (<i>Overshoot Maximum</i>)
Error <i>Steady State</i>

Sedangkan untuk desain parameter kontroler PID didapatkan melalui metode analitik untuk mencari nilai parameter τ_i , τ_d dan K_p sebagai berikut :

$$\tau_i = \frac{2\zeta}{\omega_n} \quad (7)$$

$$\tau_d = \frac{1}{\omega_n^2 \cdot \tau_i} \quad (8)$$

$$\tau^* = \frac{\tau_i}{K_p \cdot K} \quad (9)$$

Untuk mendapatkan nilai K_i dan K_d , digunakan persamaan sebagai berikut

$$K_i = \frac{K_p}{\tau_i} \quad (10)$$

$$K_d = K_p \times \tau_d \quad (11)$$

Keterangan :

τ_i = Waktu Integral (s)

τ_d = Waktu Derivatif (s)

τ^* = Waktu yang diinginkan (s)

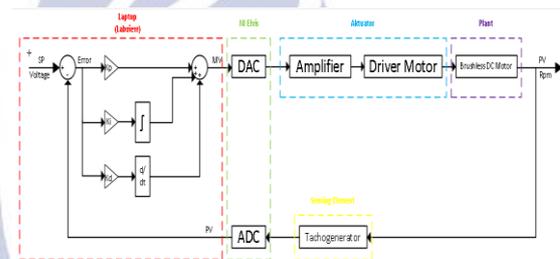
K_p = Konstanta Proporsional

K_i = Konstanta Integral

K_d = Konstanta Derivatif

Rancang Bangun Software

Software yang digunakan pada penelitian ini adalah Labview 2017. Labview 2017 berfungsi sebagai pusat pengendali dan juga berfungsi untuk menyimulasikan fungsi alih yang telah didapatkan sebelumnya. Hubungan antara *software* dan *hardware* dapat dilihat pada Gambar 20. Dan Diagram alir dari rancang bangun *software* ditunjukkan pada Gambar 8.

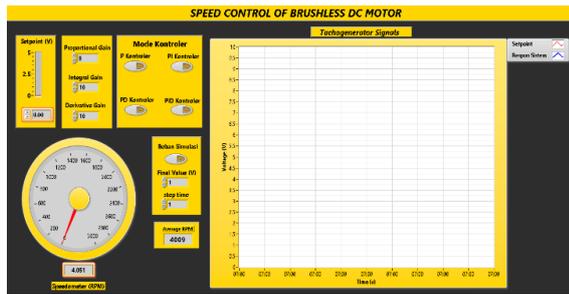


Gambar 8. Hubungan antara *software* dan *hardware* pada pengendalian kecepatan *Brushless DC Motor* menggunakan kontroler PID berbasis Labview (Sumber : Dokumen Pribadi, 2018)

Berdasarkan Gambar 8, secara umum dapat dijelaskan bahwa parameter kontroler PID yaitu K_p , K_i dan K_d sudah dihitung pada proses desain kontroler PID. Sehingga pada penelitian ini, parameter kontroler PID ditentukan oleh penulis dan tidak berubah saat proses berjalan. Nilai *setpoint* juga ditentukan oleh penulis dengan rentang tegangan sebesar 0V sampai dengan 5V. Kecepatan yang dibaca oleh *tachogenerator* akan diubah menjadi nilai tegangan. Nilai tegangan akan dikirim ke NI Elvis dan langsung diterima Labview setelah proses ADC. Nilai tersebut akan dibandingkan dengan nilai *setpoint* dan didapatkan nilai *error*. Nilai *error* tersebut akan diolah oleh kontroler PID. Jika *error* ≈ 0 maka proses selesai. Sedangkan jika *error* tidak sama dengan 0, maka proses akan berulang sampai *error* ≈ 0 .

Perancangan Tampilan pada Labview

Tampilan labview yang akan digunakan ditunjukkan pada Gambar 9.



Gambar 9. Rancangan Tampilan *Front Panel* pada Labview
(Sumber : Labview, 2017)

Nilai *setpoint* berupa nilai tegangan yang akan ditampilkan pada panel tachogenerator. Untuk nilai RPM, didapatkan setelah dilakukan kalibrasi tachogenerator.

HASIL DAN PEMBAHASAN

Pada bagian ini akan dibahas proses kalibrasi sensor *Tachogenerator*, pemodelan sistem, validasi model, perancangan kontroler PID, dan pengujian respon sistem.

1) Kalibrasi sensor *Tachogenerator*

Proses kalibrasi sensor tachogenerator dilakukan agar didapatkan nilai RPM brushless DC motor dalam setiap satuan volt yang dikeluarkan oleh sensor *Tachogenerator*. Alat bantu kalibrasi menggunakan tachometer digital tipe DT-2234C milik lab kendali Universitas Negeri Surabaya. Tabel kalibrasi ditunjukkan pada Tabel 2.

Tabel 2. Data kalibrasi *Tachogenerator*

No	Tachometer Digital (RPM)	Tachogenerator (V)
1.	275	0.5
2.	557	1
3.	850	1.5
4.	1200	2
5.	1375	2.5
6.	1670	3
7.	1925	3.5
8.	2150	4
9.	2450	4.5
10.	2700	5

Setelah mendapatkan data kalibrasi, dapat ditentukan nilai RPM setiap kenaikan tegangan pada tachogenerator yang telah dihitung yaitu rata-rata 557 RPM per 1V tegangan pada tachogenerator.

2) Pemodelan Sistem

Pemodelan sistem dilakukan menggunakan metode pendekatan karakteristik sistem orde-2 dengan parameter yang dibutuhkan adalah K, ω_n , dan

ζ . Metode ini dilakukan bertujuan untuk mendapatkan model atau fungsi alih dari sistem. Fungsi alih tersebut akan digunakan sebagai acuan untuk merancang parameter dari kontroler PID dan simulasi untuk mengetahui respon sistem. Pada skripsi ini dilakukan pemodelan dengan 5 setpoint berbeda, yaitu 0.5V, 1V, 1.5V, 2V dan 2.5V. Berdasarkan gambar 7, dan menggunakan persamaan (2), (3), (4), (5) maka dapat dihitung parameter dari karakteristik sistem orde-2 yaitu $K = 1.343$ V, $\zeta = 0.540$ dB, $M_p = 0.133$ V dan $\omega_n = 3.659$ rad/s.

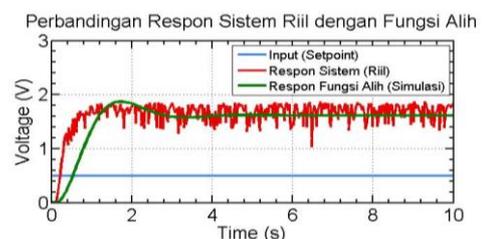
3) Validasi Sistem

Validasi model dilakukan untuk mendapatkan model atau fungsi alih dari sistem yang memiliki respon dengan tingkat error. Validasi model juga berfungsi untuk menentukan apakah fungsi alih yang telah didapatkan benar dan dapat digunakan sebagai model dari *plant* riil. Fungsi alih tiap setpoint beserta error rata-rata ditunjukkan pada Tabel 3.

Tabel 3. Fungsi Alih tiap *Setpoint*

Setpoint	Fungsi Alih	Error Rata-Rata
0.5 V	$G_s = \frac{3.234}{0.175 s^2 + 0.426 s + 1}$	0.19
1 V	$G_s = \frac{2.524}{0.217 s^2 + 0.489 s + 1}$	0.31
1.5 V	$G_s = \frac{2.01}{0.102 s^2 + 0.259 s + 1}$	0.37
2 V	$G_s = \frac{1.6}{0.124 s^2 + 0.368 s + 1}$	0.35
2.5 V	$G_s = \frac{1.343}{0.074 s^2 + 0.295 s + 1}$	0.34

Dari data fungsi alih yang diperoleh dipilih fungsi alih dengan error rata-rata terkecil yaitu 0.19. Fungsi alih dengan setpoint 0.5 V akan dipilih sebagai acuan untuk mendesain parameter dari kontroler PID. Perbandingan respon dari fungsi alih dengan respon sistem (riil) ditunjukkan pada Gambar 10. :



Gambar 10. Perbandingan Respon Sistem (Riil) dengan Respon Fungsi Alih (Setpoint 0.5 V)
(Sumber : Dokumen Pribadi, 2018)

Sebelum merancang parameter dari kontroler PID, akan dihitung parameter dari respon sistem riil *plant* secara *open loop* pada *setpoint* 0.5 V dengan nilai $\tau = 0.3$ detik. Parameter respon *plant* riil dengan *setpoint* 0.5V secara *open loop* ditunjukkan pada Tabel 4.

Tabel 4. Parameter Respon *Plant* Riil dengan *setpoint* 0.5V (*Open Loop*)

Parameter Respon	Nilai
Waktu Tunda (τ_d)	0.207 detik
Waktu Naik ($\tau_r = 5\%-95\%$)	0.833 detik
Waktu Naik ($\tau_r = 10\%-90\%$)	0.659 detik
Waktu Tunak ($\tau_s = 5\%$)	0.9 detik
Waktu Tunak ($\tau_s = 2\%$)	1.3 detik
Waktu Tunak ($\tau_s = 0.5\%$)	1.5 detik
<i>Error SteadyState</i> (Ess)	223.4%
<i>Maximum Overshoot</i> (MO)	—

4) Perancangan Kontroler PID

Perancangan kontroler PID dilakukan untuk mendapatkan parameter dari kontroler PID yaitu K_p , K_i , dan K_d . Pada pembahasan sebelumnya telah ditentukan bahwa fungsi alih dengan *setpoint* 0.5 V memiliki *error* rata-rata terkecil dan dijadikan acuan untuk menentukan parameter dari kontroler PID. Berdasarkan fungsi alih *setpoint* 0.5V, maka dapat dihitung parameter kontroler PID menggunakan persamaan (7), (8), (9), (10), dan (11) sebagai berikut :

Tabel 5. Parameter Kontroler PID

Parameter Kontroler PID	Nilai
K_p	0.790
K_i	1.954
K_d	0.320

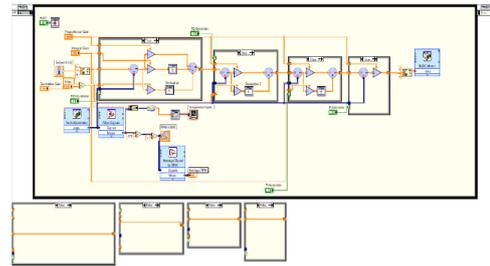
5) Implementasi Kontroler PID dan Pengujian Respon Sistem

Setelah mendapatkan nilai K_p , K_i dan K_d , selanjutnya adalah mengimplementasikan kontroler PID. Sebelum mengimplementasikan kontroler PID langsung ke *plant* riil, akan disimulasikan terlebih dahulu dengan *software Labview* 2017. Dan selanjutnya akan dilakukan pengujian pada respon sistem riil dengan beberapa metode yaitu :

Implementasi dan Pengujian pada *Plant* Riil

Setelah melakukan simulasi, maka selanjutnya adalah mengimplementasikan kontroler PID pada *plant* riil dan melakukan pengujian respon sistem.

Diagram blok pada labview ditunjukkan pada Gambar 11.

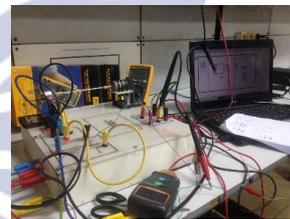


Gambar 11. Diagram blok pada labview dengan kontroler PID
(Sumber : Labview, 2017)

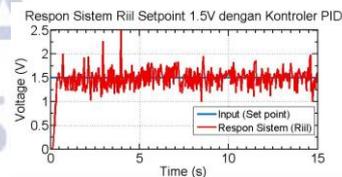
Untuk mengetahui perbedaan respon sistem sebelum dan sesudah ditambahkan kontroler PID, maka dilakukan beberapa metode pengujian sebagai berikut :

a. Pengujian Respon Sistem Riil dengan *Setpoint* Tetap (Tanpa Beban)

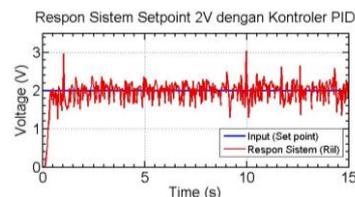
Dilakukan pengujian dengan *setpoint* tetap (tanpa beban) yaitu 1.5V, 2V, 2.5V, 3V, dan 3.5V selama 15 detik dengan *sampling time* 0.03 detik. Bentuk fisik dari pengujian ditunjukkan pada Gambar 12. Respon sistem riil dengan *setpoint* 1.5V, 2V, 2.5V, 3V, dan 3.5V berturut-turut ditunjukkan pada Gambar 13, 14, 15, 16, dan 17.



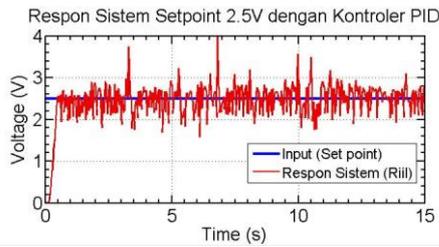
Gambar 12. Pengujian Respon Sistem Riil dengan *Setpoint* Tetap (tanpa beban)
(Sumber : Dokumen Pribadi, 2018)



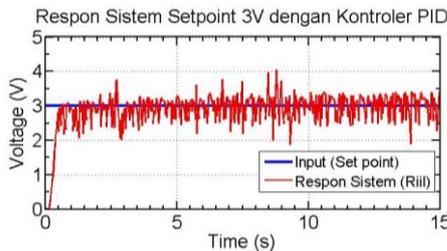
Gambar 13. Respon sistem riil *setpoint* tetap (tanpa beban) 1.5V menggunakan Kontroler PID
(Sumber : Dokumen Pribadi, 2018)



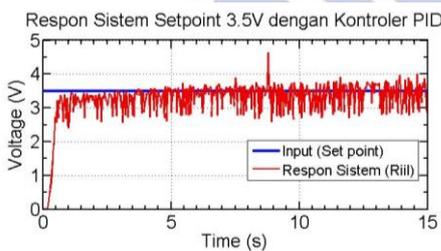
Gambar 14. Respon sistem riil *setpoint* tetap (tanpa beban) 2V menggunakan Kontroler PID
(Sumber : Dokumen Pribadi, 2018)



Gambar 15. Respon sistem riil *setpoint* tetap (tanpa beban) 2.5V menggunakan Kontroler PID (Sumber : Dokumen Pribadi, 2018)



Gambar 16. Respon sistem riil *setpoint* tetap (tanpa beban) 3V menggunakan Kontroler PID (Sumber : Dokumen Pribadi, 2018)



Gambar 17. Respon sistem riil *setpoint* tetap (tanpa beban) 3.5V menggunakan Kontroler PID (Sumber : Dokumen Pribadi, 2018)

Setelah ditambahkan kontroler PID, maka respon sistem riil akan mengalami perubahan. Untuk mengetahui perubahan tersebut, maka dilakukan perhitungan parameter dari salah satu respon sistem riil yaitu pada saat *setpoint* 3V. Parameter yang akan dihitung adalah τ_d (*Time Delay*), τ_r (*Rise Time*), τ_s (*Settling Time*), *Ess* (*Error Steady State*) dengan nilai $\tau = 0.37$ detik. Parameter respon *plant* riil dengan *setpoint* 3V menggunakan kontroler PID (tanpa beban) ditunjukkan pada Tabel 7.

Tabel 7. Parameter Respon *plant* riil dengan *setpoint* 3V menggunakan kontroler PID (Tanpa beban)

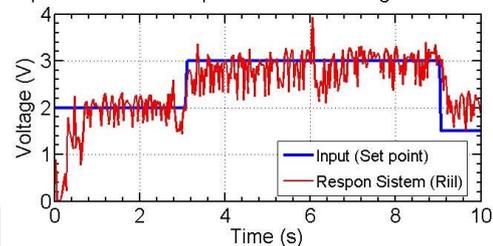
Parameter Respon	Nilai
Waktu Tunda (τ_d)	0.256 detik
Waktu Naik ($\tau_r = 5\%$ - 95%)	1.089 detik
Waktu Naik ($\tau_r = 10\%$ - 90%)	0.812 detik
Waktu Tunak ($\tau_s =$ 5%)	1.1 detik

Parameter Respon	Nilai
Waktu Tunak ($\tau_s =$ 2%)	1.48 detik
Waktu Tunak ($\tau_s =$ 0.5%)	1.85 detik
<i>Error SteadyState</i> (<i>Ess</i>)	4.3%
<i>Maximum Overshoot</i> (<i>MO</i>)	—

b. Pengujian Respon Sistem Riil dengan *Setpoint* Bervariasi (Tanpa Beban)

Dilakukan pengujian dengan memberikan *setpoint* bervariasi yaitu 1.5V, 2V, dan 3V selama 10 detik dengan *sampling time* 0.03 detik. Respon sistem riil dengan *setpoint* bervariasi ditunjukkan pada Gambar 18.

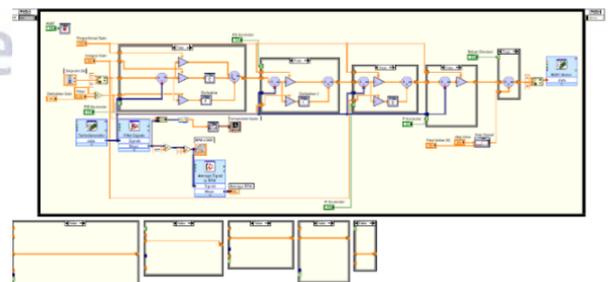
Respon Sistem Riil *Setpoint* Bervariasi dengan Kontroler PID



Gambar 18. Respon sistem riil *setpoint* bervariasi (tanpa beban) menggunakan Kontroler PID (Sumber : Dokumen Pribadi, 2018)

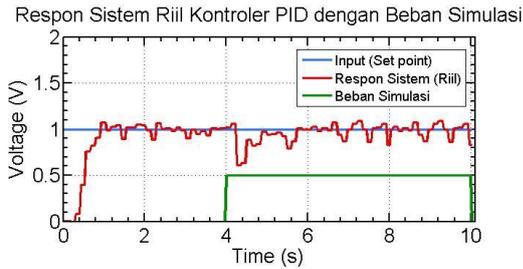
c. Pengujian Respon Sistem Riil dengan *Setpoint* Tetap dengan beban (Simulasi)

Untuk menguji ketahanan sistem dengan ditambahkannya kontroler PID, maka dilakukan proses pembebanan. Proses pembebanan dilakukan secara simulasi pada labview dengan menambahkan sinyal step sebesar 0.5V pada detik ke-4 sampai detik ke-10. Diagram blok labview dan respon sistem dari pengujian dengan beban simulasi berturut-turut ditunjukkan pada Gambar 19. dan Gambar 20.



Gambar 19. Diagram blok labview pada pengujian dengan beban (simulasi) menggunakan Kontroler PID

(Sumber : Labview, 2017)



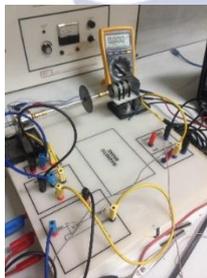
Gambar 20. Respon sistem riil *setpoint* tetap dengan beban (simulasi) menggunakan Kontroler PID

(Sumber : Dokumen Pribadi, 2018)

Respon sistem diberikan input sebesar 1 V. Pada saat beban aktif, terjadi penurunan nilai *output* dari respon sebesar 0.5V. Hal ini sejalan dengan nilai beban yang diberikan. Tetapi respon sistem dengan ditambahkan kontroler PID mampu menyetabilkan nilai dari *output* kembali kepada *setpoint* kurang dari 1 detik.

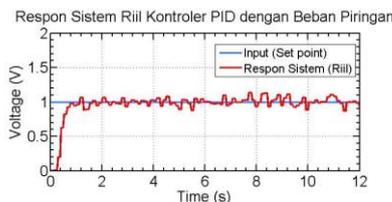
d. Pengujian Respon Sistem Riil dengan *Setpoint* Tetap dengan beban (Piringan Besi)

Setelah memberikan beban dengan simulasi, selanjutnya akan dilakukan pengujian dengan ditambahkan piringan besi pada *shaft* antara *brushless* DC motor dan *tachogenerator*. Pada pengujian ini, diberikan nilai input sebesar 1V selama 12 detik. Bentuk fisik dari pengujian ditunjukkan pada Gambar 21. Untuk respon sistem riil dengan beban piringan besi ditunjukkan pada Gambar 22.



Gambar 21. Pengujian Respon Sistem Riil dengan *Setpoint* Tetap (Beban Piringan Besi)

(Sumber : Dokumen Pribadi, 2018)



Gambar 22. Respon sistem riil dengan beban (piringan besi) menggunakan Kontroler PID

(Sumber : Dokumen Pribadi, 2018)

Berdasarkan gambar 22, dapat dihitung parameter respon t_d (*Time Delay*), t_r (*Rise Time*), t_s (*Settling Time*), E_{ss} (*Error Steady State*)

dengan nilai $\tau = 0.41$ detik. Parameter respon *plant* riil dengan *setpoint* 1V menggunakan kontroler PID (beban piringan besi) ditunjukkan pada Tabel 8.

Tabel 8. Parameter Respon *plant* riil dengan *setpoint* 3V menggunakan kontroler PID (Beban piringan besi)

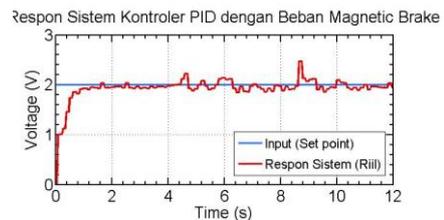
Parameter Respon	Nilai
Waktu Tunda (τ_d)	0.284 detik
Waktu Naik ($\tau_r = 5\%-95\%$)	1.20 detik
Waktu Naik ($\tau_r = 10\%-90\%$)	0.90 detik
Waktu Tunak ($\tau_s = 5\%$)	1.23 detik
Waktu Tunak ($\tau_s = 2\%$)	1.64 detik
Waktu Tunak ($\tau_s = 0.5\%$)	2.05 detik
<i>Error SteadyState</i> (E_{ss})	3.7%
<i>Maximum Overshoot</i> (MO)	—

Selanjutnya, ditambahkan modul *magnetic brake* model U-163B yang memiliki skala kekuatan *magnetic* 1-10. Sehingga pengujian dibagi menjadi 3 bagian, yaitu nilai *minimum*, *average*, dan *maximum*. Dengan nilai *minimum* bernilai 1, *average* bernilai 5 dan *maximum* bernilai 10. Bentuk fisik dari pengujian ditunjukkan pada Gambar 23. Respon sistem riil dengan nilai *magnetic minimum*, *average*, dan *maximum* berturut-turut ditunjukkan pada Gambar 24, Gambar 25, dan Gambar 26.



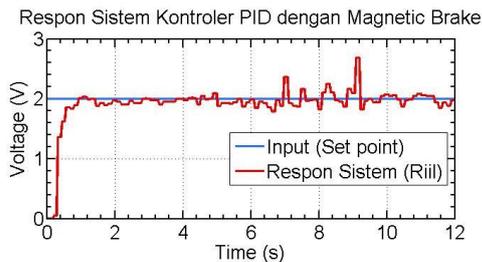
Gambar 23. Pengujian Respon Sistem Riil dengan *Setpoint* Tetap (Beban Piringan Besi dan *Magnetic Brake*)

(Sumber : Dokumen Pribadi, 2018)

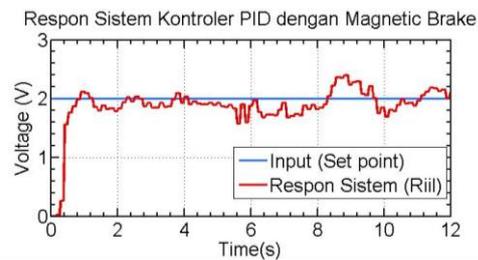


Gambar 24. Respon sistem riil dengan beban (piringan besi) dan *magnetic brake* skala *minimum* menggunakan Kontroler PID

(Sumber : Dokumen Pribadi, 2018)



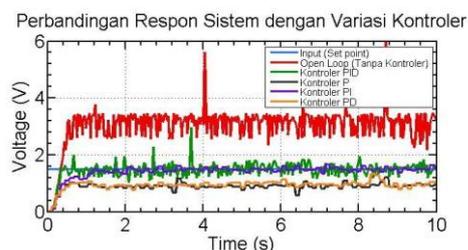
Gambar 25. Respon sistem riil dengan beban (piringan besi) dan *magnetic brake* skala *average* menggunakan Kontroler PID (Sumber : Dokumen Pribadi, 2018)



Gambar 26. Respon sistem riil dengan beban (piringan besi) dan *magnetic brake* skala *maximum* menggunakan Kontroler PID (Sumber : Dokumen Pribadi, 2018)

Pada saat nilai *minimum* (Gambar 24), respon sistem mampu mencapai *setpoint* dengan sedikit gangguan pada detik ke-4. Pada saat nilai *average* (Gambar 25), respon sistem mampu mencapai *setpoint* dengan gangguan yang mencapai 0.5 V pada detik ke-6. Dan pada saat nilai *maximum* (Gambar 26), respon sistem mampu mencapai *setpoint* dan terdapat gangguan yang besar pada detik ke-2.

Pengujian terakhir yang dilakukan adalah mengimplementasikan kontroler P, PI, dan PD sebagai pembanding respon sistem dengan variasi kontroler yang ditunjukkan pada Gambar 27.



Gambar 27. Perbandingan Respon sistem riil dengan beban (piringan besi) menggunakan kontroler PID, PI, PD, P dan Tanpa Kontroler. (Sumber : Dokumen Pribadi, 2018)

Respon sistem dengan kontroler PID dan kontroler PI mampu mencapai *setpoint*, sedangkan kontroler P dan kontroler PD tidak mampu mencapai *setpoint*. Pengujian tersebut dilakukan dengan *setpoint* 1.5V dan dengan beban piringan besi tanpa *magnetic brake*.

PENUTUP

Simpulan

Berdasarkan penelitian yang dilakukan, maka didapatkan simpulan bahwa pengujian yang dilakukan menggunakan *software* Labview mampu mengolah data respon sistem riil dengan *real time*. Dengan pendekatan karakteristik respon sistem orde dua, didapatkan parameter kontroler PID yaitu $K_p = 0.790$, $K_i = 1.954$, dan $K_d = 0.320$. Kontroler PID yang dirancang untuk mengendalikan respon kecepatan putar *brushless* DC motor berhasil memperbaiki respon sistem sebelum ditambahkan kontroler PID. Respon setelah ditambahkan kontroler PID mampu mencapai nilai *setpoint* dan mengurangi *Ess (Error Steady State)* pada respon sistem riil menjadi 4.3% (tanpa beban) dan 3.7% (Piringan Besi).

Saran

Berdasarkan simpulan diatas ada beberapa saran yang dapat dilakukan untuk pengembangan sistem agar hasil lebih maksimal yaitu model atau fungsi alih yang didapatkan masih kurang baik, sehingga dapat dilakukan metode pendekatan yang lain agar mendapatkan fungsi alih yang lebih baik dari penelitian ini, menggunakan sensor *tachogenerator* yang lebih akurat dan presisi agar respon kecepatan yang didapat lebih baik dan sistem ini masih dapat dikembangkan dengan menggunakan metode kontroler yang lain seperti *fuzzy logic*, *model predictive control* (MPC), dan *Linear Quadratic Regulator control* (LQR) untuk mendapatkan hasil respon kecepatan yang lebih baik.

DAFTAR PUSTAKA

- Al-Mashakbel, Atef Saleh Othman. 2009. *Proportional Integral and Derivative Control of Brushless DC Motor*. Europe Journal of Scientific Research Vol.35 No.02, PP : 198-203.
- Azzumar, Muhammad. 2012. *Pemodelan dan Desain Kendali Sistem Aktuator Kendali Sirip Berbasis Brushless DC Motor*. Skripsi. Universitas Indonesia : Depok.
- Krause, Paul. 2002. *Analysis of Electric Machinery and Drive Systems*. Second Edition. Purdue University. ISBN 0-471-14326-X
- Kumar, Brajesh dkk. 2017. *Controller Design for Closed Loop Speed Control of BLDC Motor*. International Journal of Electrical Engineering and Informatics Vol. 09, No. 01, PP: 146-160.
- Nise, Norman S. 2015. *Control System Engineering Seventh Edition*. Pomona: John Willey & Sons.
- Ogata, Katsuhiko. 2010. *Modern Control Engineering. Fifth Edition..* New Jersey.
- Oguntoyinbo, Oludayo John. 2009. *PID Control of Brushless DC Motor and Robot Trajectory Planning and Simulation With Matlab /Simulink*. Vasa Yrkeshogskoa : University of Applied Sciences.

- Ridwan. 2012. *Sensor dan Transduser*. Universitas Negeri Padang.
- Saputra, Edi. 2013. *Analisis Efisiensi Motor DC Seri Akibat Pergeseran Sikat*. Vol 2 No. 3. Universitas Sumatera Utara
- Shafiudin, Sofyan. 2016. *Sistem Monitoring dan Pengontrolan Temperature Pada Inkubator Penetas Telur Berbasis PID*. PPs Universitas Negeri Surabaya.
- Tim Penulis. 2014. *Buku Pedoman Penulisan dan Ujian Skripsi Unesa*. Surabaya: Unesa.

