

IMPLEMENTASI PENGENDALIAN KECEPATAN *BRUSHLESS* MOTOR DC MENGGUNAKAN LOGIKA FUZZY BERBASIS GUI LABVIEW 2014

Jordan Theja Sibuea

S1 Teknik Elektro, Fakultas Teknik, Universitas Negeri Surabaya

e-mail : jordansibuea@mhs.unesa.ac.id

Endryansyah

Teknik Elektro, Fakultas Teknik, Universitas Negeri Surabaya

e-mail : endryansyah@gmail.com

Abstrak

Brushless Motor Direct Current (*Brushless DC*) membutuhkan suatu pengendali untuk menggerakkan dan mengendalikan putaran *Brushless* Motor DC. Juga penggerak daya atau *Driver Brushless* Motor DC agar motor dapat dikendalikan secara akurat. Tujuan dari penelitian ini adalah merancang dan mengimplementasikan logika *fuzzy* untuk pengendali kecepatan *Brushless* Motor DC. Rancang bangun sistem menggunakan perangkat akuisisi data NI Elvis II+ dan perangkat lunak NI LabView 2014 sebagai unit pengendali. Selain itu, *tacho* digunakan sebagai transduser untuk mengukur kecepatan motor. *Brushless* Motor DC digunakan sebagai perangkat aktuasi untuk memanipulasi keadaan sistem. Sistem logika *fuzzy* digunakan untuk mengatur kecepatan *Brushless* Motor DC dan menjaga kestabilan putarannya. Pada perancangan pengendali dengan 5 fungsi keanggotaan diperoleh nilai masukan *error* dan *derror* -5 sampai dengan 5 dan nilai keluaran -7,5 sampai dengan 7,5. Menggunakan 2 representasi yaitu segitiga dan trapesium dalam penyusunannya. Digunakan 25 aturan basis yang berfokus pada titik positif sesuai dengan kebutuhan *plant*. Jenis defuzzifikasi yang digunakan adalah COA/centroid. Implementasi pengendali pada sistem fisik diketahui bahwa logika *fuzzy* yang dirancang memiliki kemampuan menstabilkan sistem dan ketahanan terhadap beban yang baik, berhasil mencapai nilai *setpoint* dan mengurangi *Ess* (*Error Steady State*) pada respon sistem riil menjadi 2,50% (tanpa beban) dari kondisi semula tanpa logika yaitu 224%, serta memiliki respon dengan *tr* (*rise time*) pada logika *fuzzy* 5 fungsi keanggotaan sebesar 0,588 detik dan *Maximum Overshoot* (*MO*) sebesar 13,8%.

Kata Kunci : *Brushless* Motor DC, Logika *Fuzzy*, LabVIEW 2014

Abstract

Brushless Direct Current (*Brushless DC*) motor requires a controller to drive and control the *Brushless* Motor DC rotation. Also a power drive or *Brushless* Motor DC Driver for the motor can be controlled accurately. The purpose of this research is to design and implement *fuzzy* controller for speed control of *Brushless* Motor DC. System design using NI Elvis II + data acquisition device and NI LabView 2014 software as control unit. In addition, the *tacho* is used as a transducer to measure motor speed. *Brushless* Motor DC is used as an actuation device to manipulate the state of the system. The *fuzzy* controller system is used to adjust the *Brushless* Motor DC speed and maintain the stability of the rotation. Designing the controller with 5 membership functions obtained the value of input error and *derror* -5 to 5 and the output value -7,5 up to 7.5. Using 2 representations of triangle and trapezium in the preparation. 25 base rules are used that focus on positive points in accordance with the needs of the plant. The defuzzification type used is COA / centroid. Implementation of the controller on the physical system is known that the designed *fuzzy* controller has the ability to stabilize the system and good load resistance, managed to reach the *setpoint* value and reduce the *Ess* (*Error Steady State*) in the real system response to 2.50% (without load) from the original condition without controller that is 224%, and has a response with *tr* (*rise time*) on the *fuzzy* controller 5 membership function of 0.588 seconds and *Maximum Overshoot* (*MO*) of 13.8%.

Keywords : *Brushless* Motor DC, Fuzzy Controller, LabVIEW 2014

PENDAHULUAN

Pada tahun 2015, Hudaibiy Hibban, Josaphat Pramudijanto, dan Nurlita Gamayanti melakukan

penelitian dengan judul “Desain Logika *Fuzzy* PID *Gain Scheduling* untuk Pengaturan Kecepatan Motor DC Tanpa Sikat” Tujuannya adalah menghasilkan suatu sistem pengendalian kecepatan *Brushless* Motor DC

dengan desain logika *Fuzzy PID Gain Scheduling* yang memiliki *settling time* 3,6 detik dan berfokus pada penggunaan rem ketika diberi beban.

Di tahun yang sama pula, Agung Dwi Yulianta, Sasongko Pramono Hadi, dan Suharyanto melakukan penelitian dengan judul “Pengendalian Kecepatan *Brushless Motor DC (BLDC)* Menggunakan Metode Logika *Fuzzy*”. Metode defuzzifikasi yang mampu mengikuti *setting* kecepatan yang diberikan adalah COA dengan waktu kenaikan 0,25 detik, dan waktu tunak 0,27 detik. Dalam penelitian yang penulis buat ini bertujuan untuk merancang dan membuat pengendalian kecepatan *Brushless Motor DC* berbasis LabView. Dimana LabView digunakan sebagai program dan antar muka dari grafik dan juga media pembuatan *diagram block* dan memunculkan hasil pengendalian kecepatan *Brushless Motor DC* menggunakan logika *fuzzy*.

Dari kedua penelitian yang telah diuraikan tersebut dilakukan dengan menggunakan MatLab sebagai pengendali utamanya. Dalam penelitian yang penulis buat ini bertujuan untuk merancang dan membuat pengendalian kecepatan *Brushless Motor DC* berbasis LabView. Dimana LabView digunakan sebagai program dan antar muka dari grafik dan juga media pembuatan *diagram block* dan memunculkan hasil pengendalian kecepatan *Brushless Motor DC* menggunakan logika *fuzzy*. Dari permasalahan tersebut maka penulis mengambil judul penelitian “Implementasi Pengendalian Kecepatan *Brushless Motor DC* Menggunakan Logika *fuzzy* Berbasis GUI Labview 2014”.

Tujuan penelitian ini untuk menghasilkan desain logika *fuzzy*. Desain logika *fuzzy* yang terbentuk akan diimplementasikan pada *Brushless Motor DC* untuk mengatur kecepatannya melalui Labview 2014.

KAJIAN PUSTAKA

Logika *Fuzzy*

Teori ini merupakan perluasan teori himpunan tegas, didasarkan pada cara manusia mempersepsikan suatu nilai yang tidak pasti seperti mempersepsikan suhu dengan istilah panas atau dingin bukan dengan nilai derajatnya. Teori ini berkembang cukup pesat dengan aplikasi pada berbagai bidang. Sistem kendali *fuzzy* yang merupakan aplikasi teori *fuzzy* pada bidang kendali, mencakup himpunan *fuzzy*, fungsi keanggotaan, aturan dasar *fuzzy*, dan struktur dasarnya.

Brushless Motor DC

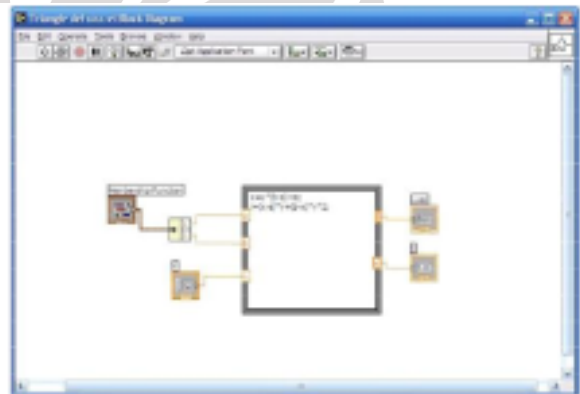
Brushless Motor DC adalah motor sinkron dengan magnet permanen yang menggunakan sebuah rangkaian elektronika (*inverter*) untuk mengendalikan arus yang ada. Motor ini menggunakan suatu komutator mekanik seperti yang ada pada motor dc konvensional, selain itu juga menggunakan komutasi elektronik yang menjadikannya motor yang minim perawatan (Azzumar, 2012).



Gambar 1. *Brushless Motor DC*
(Sumber : inventoautomation,2018)

Labview

LabView adalah sebuah *software* pemrograman yang diproduksi oleh *National Instruments*. Seperti bahasa pemrograman lainnya yaitu C++, matlab atau *visual basic*, LabView juga mempunyai fungsi dan peranan yang sama, perbedaannya adalah LabView menggunakan bahasa pemrograman berbasis grafis atau blok diagram sedangkan bahasa pemrograman lainnya menggunakan *text*. Berfungsi sebagai pusat pengendalian sistem (Rakhman, 2017).



Gambar 2. Labview
(Sumber: Labview, 2018)

NI ELVIS

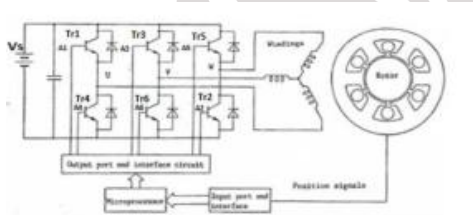
National Instruments Educational laboratory Virtual Instrumentation Suite (NI ELVIS) adalah suatu modul instrumentasi dan *prototyping* berbasis LabView dan komputer. NI ELVIS terdiri dari *prototyping board*, piranti *data acquisition (DAQ)* multifungsi, dan instrumen virtual berbasis LabView. Kombinasi ini menghasilkan suatu *platform* instrumentasi *modular* terintegrasi, yang kemampuan fungsionalnya dapat dibandingkan dengan instrumen pada umumnya seperti multimeter digital, osiloskop, *function generator*, dan *power supply* yang bisa dijumpai di dunia teknik elektro (Wismal, 2014).



Gambar 3. NI ELVIS
(Sumber: ni, 2018)

Driver Motor

Driver Motor merupakan bagian yang berfungsi untuk menggerakkan *Brushless Motor DC* dimana perubahan arah motor bergantung pada nilai tegangan yang dimasukkan. Bertugas untuk menjalankan motor baik mengatur arah putaran motor maupun kecepatan motor. Selain itu, dengan kontrol sinyal PWM dapat digunakan untuk mengatur kecepatan motor dengan memberi pulsa frekwensi tetap ke motor. (Saputro, 2012).



Gambar 4. Skema Driver Motor
(Sumber : Azzumar,2012)

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, imbas elektromagnet dari magnet *permanent*. Dapat apat membangkitkan tegangan *DC* yang langsung dapat menghasilkan informasi kecepatan, sensitivitas *tachogenerator DC* cukup baik terutama pada daerah kecepatan tinggi (Purnama, 2012).



Gambar 5. Tachogenerator
(Sumber : Datasheet Johnson 3C4052, 2018)

Electro break Magnetic U-1638

Digunakan sebagai beban mekanik dimana terdapat 1-10 indikator rem yang semakin besar indikator rem maka semakin besar daya rem magnetic yang diberikan.



Gambar 6. Electro Break Magnetic U-1638
(Sumber : dokumen pribadi, 2018)

Tachometer Digital model DT-2234C

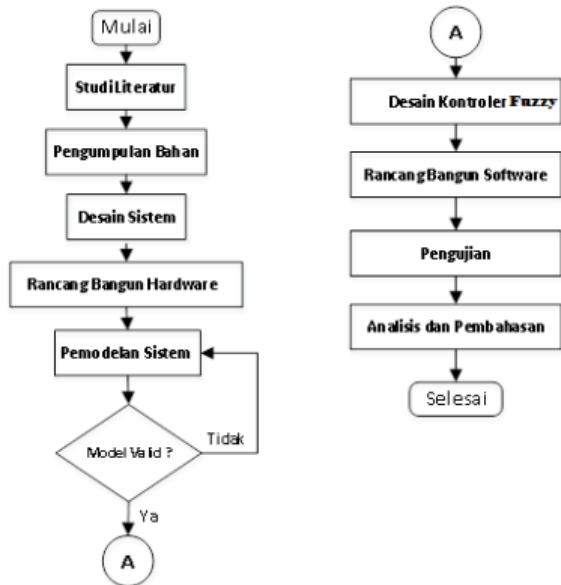
Digunakan untuk menghitung kecepatan dengan rpm secara digital Memiliki fitur utama dengan tampilan: Besar 5 digit, 18mm LCD, rentang: 2,5 hingga 99,999 RPM. resolusi: 0,1 RPM (dari 2,5 hingga 999,9 RPM) 1 RPM (lebih dari 1.000 RPM), akurasi: +/- (0,05% +1 digit), jarak: 50 hingga 500mm / 2 hingga 20 inci, memori: Nilai Terakhir. Nilai Maks. Min RPM danan ukuran: 13x7x2,9cm. Menggunakan daya baterai 9V.



Gambar 7. Tachometer Digital model DT-2234C
(Sumber : dokumen pribadi, 2018)

METODE PENELITIAN

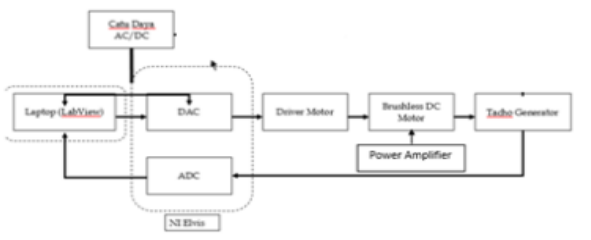
Pada penelitian ini menggunakan *software* LabVIEW 2014. *Software* LabVIEW akan digunakan untuk mendesain kontrolernya, yaitu kontroler *fuzzy* dengan 5 fungsi kenggotaan. serta menampilkan respon sistem dari *hardware* *Brushless Motor DC*. Tahapan perancangan penelitian ini secara garis besar dijelaskan dalam Gambar 8.



Gambar 8. Diagram Alir Tahapan Penelitian (Sumber : data primer, 2018)

Desain Sistem

Desain sistem pengendalian kecepatan *Brushless* Motor DC menggunakan logika *fuzzy* berbasis GUI LABview 2014 ditunjukkan pada Gambar 9.

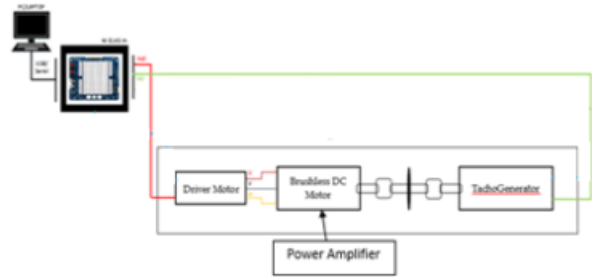


Gambar 9. Desain Sistem Pengendalian (Sumber : data primer, 2018)

Pada model desain sistem Gambar 9 menunjukkan bahwa sistem terdiri dari beberapa bagian. Pengendalian kecepatan *Brushless* Motor DC berbasis logika *fuzzy* yang terintegrasi menjadi satu sistem utuh. Pembagian blok sistem dibagi menjadi blok *hardware* dan blok pengolahan data. Pada blok *hardware* terdiri dari PC/laptop, NI Elvis, *Driver Motor*, *Brushless* Motor DC serta *tachogenerator*. *Tachogenerator* adalah sensor kecepatan yang akan menerima pembacaan data kondisi kecepatan *Brushless* Motor DC. NI Elvis digunakan untuk merancang dan membantu mengurangi persentase kesalahan dalam melakukan perancangan sebuah rangkaian elektronika. Digunakan untuk menyediakan tegangan listrik ke semua rangkaian dengan sumber listrik sebesar 220 VAC, 5 VDC dan 12 VDC. PC/Laptop berfungsi sebagai HMI (*Human Machine Interface*) yang digunakan untuk mengetahui respon keluaran sistem, membuat logika dan juga digunakan untuk pemberian nilai *input* yang digunakan.

Rancang Bangun Hardware

Berikut adalah desain rancang bangun *hardware* pada penelitian ini :



Gambar 10. Skema Rancang Bangun Hardware (Sumber : data primer, 2018)

Identifikasi Plant

Pada penelitian ini, sebelum menggunakan logika *fuzzy*, pertama harus mengetahui karakteristik respon dari *Brushless* Motor DC yang digunakan. Sehingga nantinya kontroler yang digunakan mampu memperbaiki respon pada *Brushless* Motor DC. Untuk melakukan identifikasi *plant*, pada penelitian ini menggunakan sistem *open-loop* di LabView 2014.



Gambar 11. Identifikasi dengan sistem *open loop* (Sumber : data primer, 2018)

Setelah itu, data yang didapatkan akan diolah dan ditentukan parameter-parameter K , ω_n , dan ζ . Karakteristik sistem orde-2 tiap *setpoint* dengan rumus sebagai berikut :

$$K = \frac{Y_{ss}}{X_{ss}} \tag{1}$$

$$Mp = \frac{Y_p - Y_{ss}}{Y_{ss}} \tag{2}$$

$$\zeta = \frac{|\ln Mp|}{\sqrt{\pi^2 + (\ln Mp)^2}} \tag{3}$$

$$\omega_n = \frac{\pi}{t_p \sqrt{1 - \zeta^2}} \tag{4}$$

Keterangan :

- X_{ss} : Input (V)
- Y_{ss} : Output Steady State (V)
- Y_p : Tegangan Puncak (V)
- t_p : Waktu Puncak (s)
- K : Gain Overall (V)
- π : phi (3,14)
- ζ : Rasio Redaman

ω_n : Frekuensi Alami (Hz)
 M_p : Overshoot Maksimum

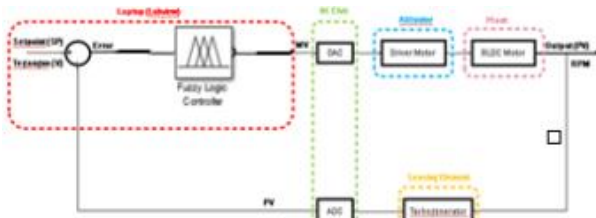
digunakan memiliki dua masukan yaitu *error* dan *delta error*. Dihitung dengan persamaan di bawah ini.

$$e(k) = SP(k) - Y(k) \quad (5)$$

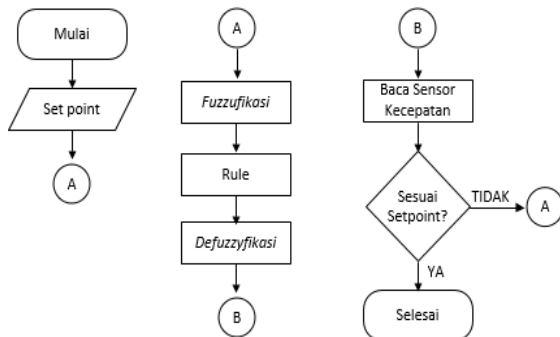
$$de(k) = e(k) - e(k - 1) \quad (6)$$

Rancang Bangun Software Labview

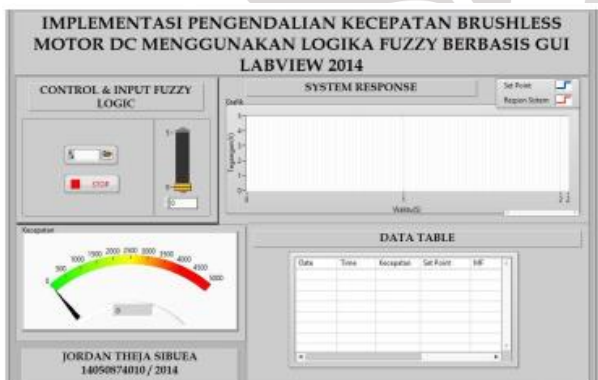
Berikut adalah rancang bangun *software*, diagram alir *software*, dan tampilan *front panel* pada GUI Labview :



Gambar 12. Skema Rancang Bangun *Software* (Sumber : data primer, 2018)



Gambar 13. Diagram Alir Rancang Bangun *Software* (Sumber : data primer, 2018)



Gambar 14. Rancangan Tampilan Front Panel Labview (Sumber : data primer, 2018)

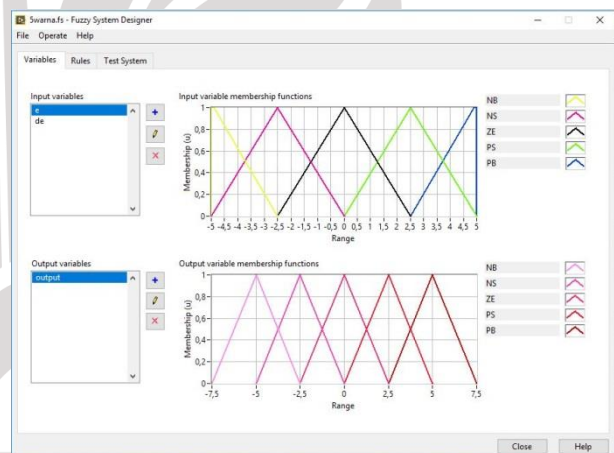
Rancang Bangun Logika Fuzzy

Variabel *input* dapat berupa nilai *error*, sedangkan *output* kontroler *fuzzy* adalah variable *input* yang dimanipulasi untuk menghasilkan nilai keluaran sistem yang diinginkan. Dalam kedua hal tersebut, fungsi keanggotaan akan digunakan untuk merepresentasikan nilai-nilai yang berbeda dalam semesta pembicaraan. Untuk menentukan semesta pembicaraan tersebut, perlu ditentukan terlebih dahulu rentang nilai suatu variabel dalam konteks perancangan sistem. Kontroler *fuzzy* yang

Keterangan :

e : *error* (V)
 SP : *set point* (V)
 k : domain waktu
 de : *delta error* (V)
 Y : nilai *real* (V)

$e(k)$ adalah *error*, $de(k)$ adalah *delta error*, kedua masukan tersebut menjadi variabel *fuzzy* agar memperoleh derajat keanggotaan *fuzzy*, digunakan representasi kurva segitiga (*triangle*) dan trapesium (*trapezoid*). Fungsi keanggotaan untuk variabel *error* dan *delta error* memiliki bentuk yang sama. Untuk *output* digunakan representasi kurva segitiga yang berupa tegangan keluaran hasil defuzzifikasi.



Gambar 15. Tampilan Desain Sistem Fuzzy (Sumber : data primer, 2018)

Rule base yang digunakan yaitu *rule base if.and.then*. Sebagaimana *output plant* yang diharapkan yaitu 0 hingga 5V. Menggunakan metode *min-max* Mamdani. Terbagi atas 5 domain yaitu NB (*Negative Big*), NS (*Negatif Small*), ZE (*Zero*), PS (*Positive Small*), dan PB (*Positive Big*) untuk desain logika *fuzzy* dengan 5 fungsi keanggotaan. Basis aturan digunakan sebagai proses mengevaluasi derajat keanggotaan setiap fungsi keanggotaan masukan ke dalam basis aturan yang telah ditetapkan. Dengan menyusun basis aturan, yaitu aturan-aturan berupa implikasi-implikasi *fuzzy* yang menyatakan relasi Antara variabel input dengan dengan variabel output. Pada penelitian ini menggunakan metode Mamdani, fungsi implikasi yang digunakan adalah MIN.

Tabel 1. Basis Aturan Logika Fuzzy

e/de	NB	NS	ZE	PS	PB
NB	NB	NS	ZE	PS	PB
NS	NS	NS	ZE	PS	PB
ZE	ZE	ZE	ZE	PS	PB
PS	PS	PS	PS	PS	PB
PB	PB	PB	PB	PB	PB

Hasil keluaran proses implikasi akan masuk ke proses defuzifikasi. Tujuan dari defuzifikasi adalah untuk mendapatkan nilai konstanta keluaran dari logika fuzzy dan nilai inilah yang merupakan *manipulated value*. Defuzifikasi merupakan kebalikan dari fuzifikasi. Pada defuzifikasi *input* berupa derajat keanggotaan dalam bentuk luasan dan keluarannya berupa sebuah nilai. Metode defuzzifikasi yang digunakan adalah metode centroid. Dengan adanya defuzzifikasi ini terjadilah suatu proses konversi dari himpunan fuzzy yang dihasilkan dari komposisi ke dalam *crisp value*.

HASIL DAN PEMBAHASAN

Pengujian diagram Vi pada *software* LabVIEW 2014 dilakukan dengan dua tahap yaitu adalah pengujian simulasi dari fungsi alih dari *Brushless Motor DC* dan pengujian terhadap GUI (*Graphic User Interface*) hasil dari pembacaan *tachogenerator* yang dikirim ke Laptop.

1) Kalibrasi Sensor Tachogenerator

Proses kalibrasi sensor *tachogenerator* dilakukan agar didapatkan nilai RPM *brushless* motor DC dalam setiap satuan volt yang dikeluarkan oleh sensor *tachogenerator*. Data 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

Pemodelan Sistem dan Validasi Model

Pemodelan sistem dilakukan menggunakan metode pendekatan karakteristik sistem orde-2

dengan parameter yang dibutuhkan adalah K , ω_n , dan ζ (Ogata, 1985). Metode ini dilakukan bertujuan untuk mendapatkan model atau fungsi alih dari sistem. Fungsi alih tersebut akan digunakan sebagai acuan untuk merancang parameter 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. Dapat dihitung parameter dari karakteristik sistem orde-2 yaitu K , ζ , dan ω_n menggunakan persamaan (2.5), (2.6), (2.7), dan (2.8). Mencari nilai ζ dan nilai ω_n , sebagai berikut :

$$Y_{ss} = 2,524V \quad Y_p = 2,88$$

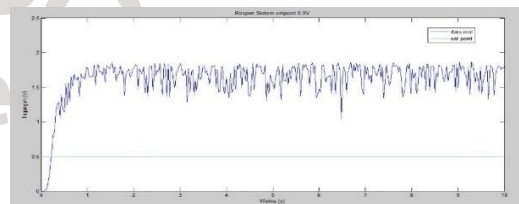
$$X_{ss} = 1V \quad t_p = 1,72$$

Digunakan rumus yang sama saat identifikasi *plant* sehingga didapat fungsi alih ditunjukkan pada Tabel 3.

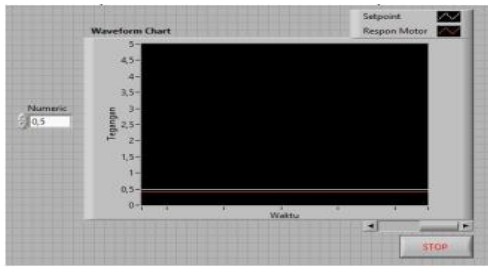
Tabel 3. Fungsi Alih tiap *setpoint*

Setpoint	Fungsi Alih	Error
0,5 V	$G(s) = \frac{3,234}{0,175s^2 + 0,426s + 1}$	0,190
1 V	$G(s) = \frac{2,524}{0,217s^2 + 0,489s + 1}$	0,317
1,5 V	$G(s) = \frac{2,010}{0,102s^2 + 0,259s + 1}$	0,376
2 V	$G(s) = \frac{1,600}{0,124s^2 + 0,368s + 1}$	0,353
2,5 V	$G(s) = \frac{1,343}{0,074s^2 + 0,295s + 1}$	0,347

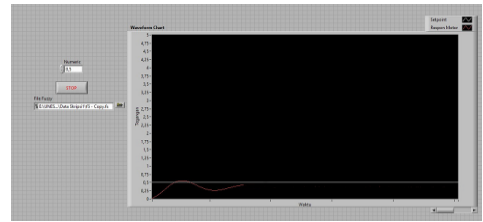
Dari data fungsi alih yang diperoleh dipilih fungsi alih dengan *error* rata-rata terkecil yaitu 0,190. Fungsi alih dengan *setpoint* 0.5 V akan dipilih sebagai acuan untuk mendesain parameter pada kondisi *open loop* riil dan simulasi *close loop*. Berikut perbandingannya :



Gambar 16. Respon *open loop* Riil Fungsi Alih dengan Setpoint 0.5V
(Sumber : data primer, 2018)



Gambar 17. Respon *close loop* Simulasi Fungsi Alih dengan Setpoint 0.5V di LabView 2014 (Sumber : data primer, 2018)



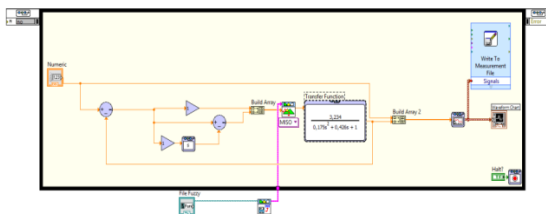
Gambar 18. Respon Sistem dari fungsi Alih 0,5V setelah ditambahkan Logika *Fuzzy* 5 Fungsi Keanggotaan (Sumber : data primer, 2018)

Tabel 4. Parameter Respon *Plant* Riil dengan *setpoint* 0,5 V (*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
Error <i>SteadyState</i> (Ess)	224%
<i>Maximum Overshoot</i> (MO)	—

Implementasi dan Pengujian menggunakan Fungsi Alih

Fungsi alih yang digunakan adalah fungsi alih dengan *setpoint* 0,5V yang didapatkan dari proses pemodelan sistem sebelumnya. Selanjutnya akan dilakukan simulasi untuk melihat pengaruh logika *fuzzy* pada fungsi alih dengan *setpoint* 0,5V. Diagram blok pada *LabView* dan respon sistem berturut-turut dengan ditambahkan logika *fuzzy* ditunjukkan pada gambar berikut :



Gambar 17. Diagram blok pada LabView dari fungsi Alih 0,5V ditambahkan Kontroler *Fuzzy* (Sumber : data primer, 2018)

Parameter respon simulasi dengan *setpoint* 0,5V menggunakan logika *fuzzy* ditunjukkan oleh Tabel 5.

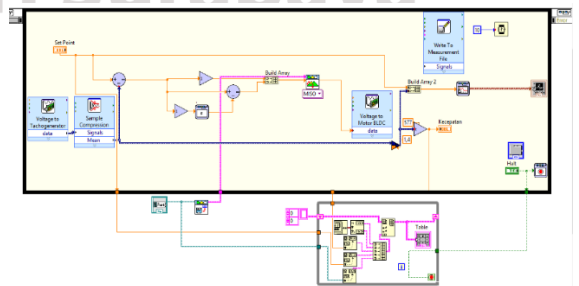
Tabel 5. Parameter Respon simulasi dengan *setpoint* 0,5V menggunakan logika *fuzzy* 5 Fungsi Keanggotaan

Parameter Respon	Nilai
Waktu Tunda (τ_d)	0,159 detik
Waktu Naik ($\tau_r = 5\%-95\%$)	0,677 detik
Waktu Naik ($\tau_r = 10\%-90\%$)	0,505 detik
Waktu Tunak ($\tau_s = 5\%$)	0,69 detik
Waktu Tunak ($\tau_s = 2\%$)	0,92 detik
Waktu Tunak ($\tau_s = 0.5\%$)	1,15 detik
Error <i>SteadyState</i> (Ess) 5 Fungsi Keanggotaan	0%
<i>Maximum Overshoot</i> (MO) 5 Fungsi Keanggotaan	12%

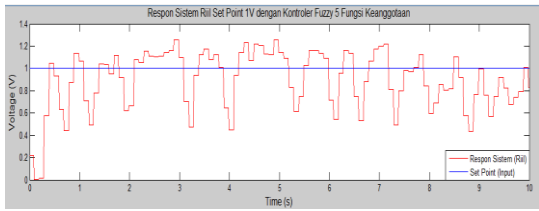
Implementasi dan Pengujian pada Labview

a) Menggunakan logika *fuzzy* dengan *setpoint* tetap tanpa beban

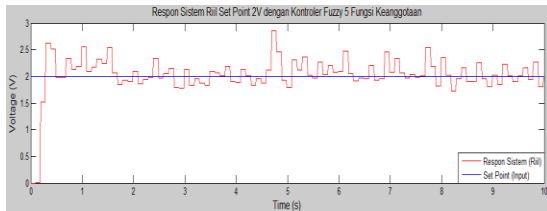
Setelah melakukan simulasi, maka selanjutnya adalah mengimplementasikan logika *fuzzy* pada *plant* riil dan melakukan pengujian respon sistem. Diagram blok pada *LabView* ditunjukkan pada gambar 19.



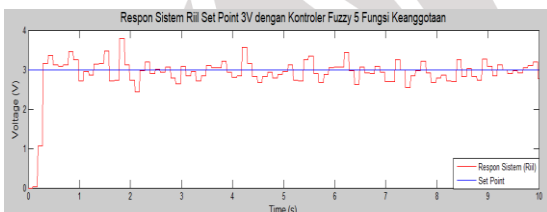
Gambar 19. Diagram blok pada LabView dengan logika *Fuzzy* (Sumber : data primer, 2018)



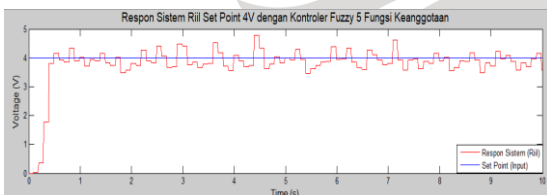
Gambar 20. Respon tegangan sistem riil *setpoint* tetap (tanpa beban) 1V menggunakan Logika Fuzzy
(Sumber : data primer, 2018)



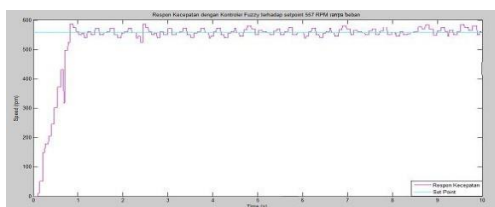
Gambar 21. Respon tegangan sistem riil *setpoint* tetap (tanpa beban) 2V menggunakan Logika Fuzzy
(Sumber : data primer, 2018)



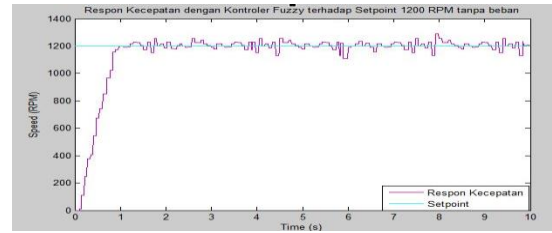
Gambar 22. Respon tegangan sistem riil *setpoint* tetap (tanpa beban) 3V menggunakan Logika Fuzzy
(Sumber : data primer, 2018)



Gambar 23. Respon tegangan sistem riil *setpoint* tetap (tanpa beban) 4V menggunakan Logika Fuzzy
(Sumber : data primer, 2018)



Gambar 24. Respon kecepatan sistem riil *setpoint* tetap (tanpa beban) 557 rpm menggunakan Logika Fuzzy
(Sumber : data primer, 2018)



Gambar 25. Respon kecepatan sistem riil *setpoint* tetap (tanpa beban) 1200 rpm menggunakan Logika Fuzzy
(Sumber : data primer, 2018)

Respon *plant setpoint* 3V logika Fuzzy di Tabel 6.

Tabel 6. Parameter Respon *plant* riil dengan *setpoint* 3V (Tanpa beban) dengan logika Fuzzy

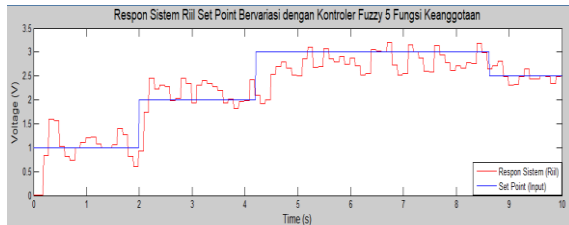
Parameter Respon	Nilai
Waktu Tunda (τ_d)	0,138 detik
Waktu Naik ($\tau_r = 5\%-95\%$)	0,588 detik
Waktu Naik ($\tau_r = 10\%-90\%$)	0,439 detik
Waktu Tunak ($\tau_s = 5\%$)	0,6 detik
Waktu Tunak ($\tau_s = 2\%$)	0,8 detik
Waktu Tunak ($\tau_s = 0.5\%$)	1 detik
Error SteadyState (Ess)	2,50%
Maximum Overshoot (MO)	13,8%

Parameter respon *plant* pada *setpoint* 3V dengan logika Fuzzy ditunjukkan oleh Tabel 7.

Tabel 7. Parameter Respon *plant* riil dengan *setpoint* 557 rpm (Tanpa beban) menggunakan logika Fuzzy

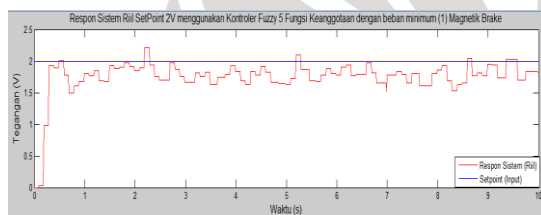
Parameter Respon	Nilai
Waktu Tunda (τ_d)	0,277 detik
Waktu Naik ($\tau_r = 5\%-95\%$)	1,17 detik
Waktu Naik ($\tau_r = 10\%-90\%$)	0,878 detik
Waktu Tunak ($\tau_s = 5\%$)	1,2 detik
Waktu Tunak ($\tau_s = 2\%$)	1,6 detik
Waktu Tunak ($\tau_s = 0.5\%$)	2 detik
Error SteadyState (Ess)	4,30%
Maximum Overshoot (MO)	10,13%

- b) Pengujian Respon Sistem Riil dengan *Setpoint* (Tanpa beban) Bervariasi menggunakan Logika *Fuzzy 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 berikut.

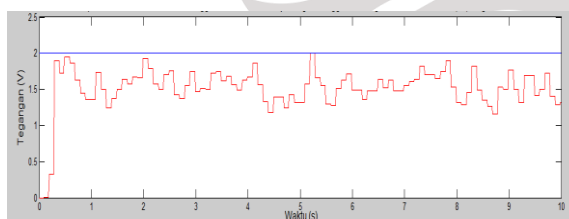


Gambar 26. Respon sistem riil *setpoint* bervariasi (tanpa beban) menggunakan Logika *Fuzzy* (Sumber : data primer, 2018)

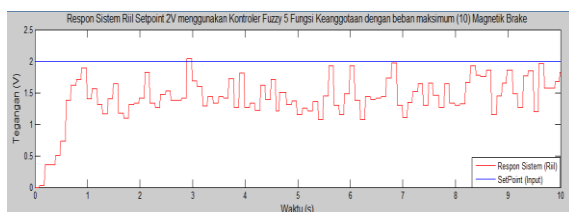
- c) Pengujian Respon Sistem Riil dengan *Setpoint* Tetap dengan beban (*Magnetic Brake*). Bebannya adalah nilai *minimum*, *average*, dan *maximum*.



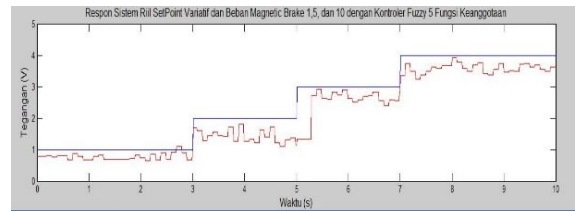
Gambar 27. Respon sistem riil *setpoint* 2V beban *break magnetic* skala *minimum* (1) menggunakan Logika *Fuzzy* (Sumber : data primer, 2018)



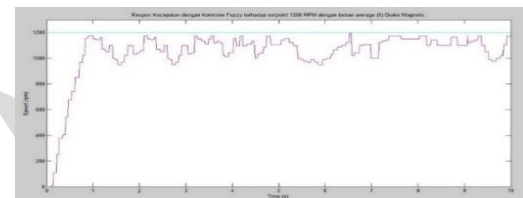
Gambar 28. Respon sistem riil *setpoint* 2V beban *break magnetic* skala *average* (5) menggunakan Logika *Fuzzy* (Sumber : data primer, 2018)



Gambar 29. Respon sistem riil *setpoint* 2V beban *break magnetic* skala *maximum* (10) menggunakan Logika *Fuzzy* (Sumber : data primer, 2018)



Gambar 30. Respon sistem riil *setpoint* variatif beban *break magnetic* skala variatif (1,5, dan 10) menggunakan Logika *Fuzzy* (Sumber : data primer, 2018)



Gambar 31. Respon kecepatan sistem riil *setpoint* 1200 rpm beban *break magnetic* skala *average* (5) menggunakan Logika *Fuzzy* (Sumber : data primer, 2018)

Tabel 8. Parameter Respon *plant* riil dengan *setpoint* 1200 rpm (beban *Break Magnetik average*) menggunakan Kontroler logika *Fuzzy*

Parameter Respon	Nilai
Waktu Tunda (τ_d)	0,346 detik
Waktu Naik ($\tau_r = 5\%-95\%$)	1,47 detik
Waktu Naik ($\tau_r = 10\%-90\%$)	1,09 detik
Waktu Tunak ($\tau_s = 5\%$)	1,5 detik
Waktu Tunak ($\tau_s = 2\%$)	2 detik
Waktu Tunak ($\tau_s = 0.5\%$)	2,5 detik
<i>Error SteadyState</i> (<i>Ess</i>)	13,41%
<i>Maximum Overshoot</i> (<i>MO</i>)	-

PENUTUP Simpulan

Berdasarkan penelitian dan pengujian yang dilakukan, maka didapatkan simpulan Sistem pada pengendalian kecepatan *Brushless Motor* DC berhasil dibuat berbasis kontroler logika *fuzzy*, Untuk mengaplikasikan kontroler logika *fuzzy* pada *plant Brushless Motor* DC adalah dengan metode penalaran *fuzzy* Mamdani dimana nilai masukan berupa *error* dan *derror* dan nilai keluaran berupa tegangan 0-5V. Dengan 5 fungsi keanggotaan digunakan 25 *rules base* dan metode defuzzifikasi *Center of Area* yang mampu menyesuaikan kecepatan *Brushless Motor* DC.

Pada pengujian *plant Brushless Motor DC* didapatkan fungsi alih dengan *error* terkecil adalah fungsi alih dengan *setpoint* 0.5V dengan *error* rata-rata sebesar 0,190 V seperti ditunjukkan pada tabel 3. Logika *fuzzy* yang dirancang untuk mengendalikan respon kecepatan putar *Brushless Motor DC* berhasil mencapai nilai *setpoint* dan mengurangi *Ess (Error Steady State)* pada respon sistem riil menjadi 2,50% (tanpa beban) seperti ditunjukkan pada tabel 6. Respon kecepatan *Brushless Motor DC* dengan logika *fuzzy* 5 fungsi keanggotaan memiliki respon dengan *tr (rise time)* pada logika *fuzzy* 5 fungsi keanggotaan sebesar 0,588 detik dan *Maximum Overshoot (MO)* sebesar 13,8% seperti ditunjukkan oleh tabel 6. Untuk respon kecepatan, pada kondisi 557 rpm tanpa beban mengalami *tr (rise time)* 1,17 detik dan *Maximum Overshoot (MO)* 10,13% seperti ditunjukkan oleh tabel 8. Sedangkan saat diberi beban *break magnetic* mengalami *tr (rise time)* 1,47 detik seperti ditunjukkan oleh tabel 8.

Saran

Berdasarkan hasil penelitian yang telah dilakukan terdapat beberapa aspek yang dapat diperbaiki dan dikembangkan agar memberikan hasil yang lebih maksimal, 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, dapat diubahnya *setpoint* melalui penelitian di masa mendatang serta menggunakan *software* pengendali yang lain seperti *MatLab*, *Arduino*, dan lain-lain. Menggunakan sensor *tachogenerator* yang lebih akurat dan presisi agar respon kecepatan yang didapat lebih baik. Sistem ini masih dapat dikembangkan dengan menggunakan metode kontroler yang lain seperti *Proportional Integral Derivative controller (PID)*, *Model Predictive Control (MPC)*, *Linear Quadratic Regulator control (LQR)* dan lain-lain untuk mendapatkan hasil respon kecepatan yang lebih baik.

DAFTAR PUSTAKA

- Hibban, Hudaibiy. 2015. "Desain Kontroler Fuzzy PID Gain Scheduling untuk Pengaturan Kecepatan Motor DC tanpa Sikat". Surabaya: Institut Teknologi Sepuluh November (ITS).
- Yulianta, Agung Dwi dkk. 2015. "Pengendalian Kecepatan Brushless Motor DC (BLDC) menggunakan Metode Logika Fuzzy". Yogyakarta: Universitas Gadjah Mada (UGM).
- Azzumar, Muhammad. 2012. "Permodelan dan Simulasi BLDC Motor Kecil untuk Aplikasi Aktuator Sirip Roket". Depok: Universitas Indonesia (UI).
- Wismal, Anggy Kusuma Dewi dkk. "Implementasi Encoder dan Decoder Hamming pada DSK TMS320C6416T". Jurnal TEKNIK POMITS (2014), p. 40.
- Saputro, Dwi Gaguk dkk. 2009. "Membangun e-learning di SMP Negeri 3 Karang Trenggalek dengan menggunakan moodle. Malang: Universitas Negeri Malang (UNM).
- Ogata, Katsuhiko. 1985. *Teknik Kontrol Automatik jilid 1*. Terjemahan Edi Laksono. Jakarta: Erlangga
- Silva, Monique. 2016. *Parameters Identification and Analysis of Bruhless Direct Current Motors*. Mossoro: Universidade Federal do Semi-Arido.
- Rakhman, Alief, 2011, Labview dan perangkatnya (Online), (<https://rakhman.net/electrical-id/labview-software/> diakses 16 Juli 2018)
- Purnama, A., 2012, Definisi dan Konstruksi Tacho Generator (Online), (<http://elektronikadasar.web.id/artikel-elektronika/definisi-dan-konstruksi-tacho-generator/> diakses 20 April 2018)