

# Penerapan Metode *Fuzzy Logic* Sebagai Pengatur Kecepatan Pada Motor BLDC

<sup>1</sup>Iradiratu D.P.K, <sup>2</sup>Tria Wisnu Saputra

Jurusan Teknik Elektro Fakultas Teknik dan Ilmu Kelautan Universitas Hang Tuah Surabaya  
Jurusan Teknik Elektro Fakultas Teknik dan Ilmu Kelautan Universitas Hang Tuah Surabaya

<sup>1</sup>[iradiratu@hangtuah.ac.id](mailto:iradiratu@hangtuah.ac.id) <sup>2</sup>[yayowisnu77@gmail.com](mailto:yayowisnu77@gmail.com)

Abstrak- Motor *Brushless* DC (BLDC) adalah motor sinkron dengan magnet permanen pada rotor dan kumparan jangkar pada stator. Keuntungan paling jelas dari konfigurasi tanpa sikat adalah penghilangan sikat, yang menghilangkan pemeliharaan sikat dan bunga api. Sistem motor BLDC mengacu pada konsep rangkaian elektromekanik, sistem penggerak yang tanggap dan hemat energi. Sistem tersebut dibangun melalui perpaduan elektromekanik, rangkaian elektronika, sistem sensor dan rangkaian logika. Pada bagian elektromekanik menonjolkan konsep keunggulan motor DC konvensional dalam hal pengendalian dan keunggulan motor sinkron 3 fasa dalam hal efisiensi. Selain keunggulan motor terdapat juga masalah pada motor. Masalah yang ada pada motor adalah dimana pada saat kenaikan beban maka dapat terjadi *overshoot* dan *error steady state*. Untuk mengatasi *overshoot* dan *error* maka diperlukan pengontrol pada Motor BLDC. Pengontrol yang digunakan ada pada *Speed Controller*, dimana *speed controller* berfungsi sebagai pengatur kecepatan pada motor BLDC. Pada *speed controller* terdapat *fuzzy logic* yang berfungsi untuk memproses data yang telah di masukkan. Dengan menggunakan kontrol kecepatan maka diharapkan dapat memperkecil terjadinya *overshoot* pada saat putaran motor mencapai maksimum dan *undershoot* ketika terjadi penurunan kecepatan saat ada penambahan beban, waktu putar motor mencapai *steady state* lebih cepat dibandingkan dengan referensi jurnal sebagai patokan, nilai *steady state* sesuai dengan kecepatan referensi yang telah diberikan sehingga tidak terjadi *error steady state*. Pada system yang telah di jalankan motor dapat mencapai hasil yang baik dimana *overshoot* dan *undershoot* memiliki nilai yang kecil, waktu *rise time* baik dan *error steady state* sangat kecil.

**Kata Kunci :** Motor BLDC, *Overshoot*, Speed Controller, Fuzzy Logic, *Steady state*

**Abstrack-** Brushless DC motor (BLDC) is a synchronous motor with a permanent magnet on the rotor and anchor coil on the stator. The most obvious advantage of the brushless configuration is the removal of the brush, which eliminates the maintenance of brushes and sparks. BLDC motor system refers to the concept of electromechanical circuit, responsive and energy-efficient drive system. The system is built through electromechanical fusion, electronics circuit, sensor system and logic circuit. In the electromechanical section highlight the concept of the benefits of conventional DC motors in terms of control and the advantages of

3 phase sync motor in terms of efficiency. In addition to motor excellence there are also problems on the motor. Problems that exist on the motor is where at the time of increase in load it can occur *overshoot* and *steady state error*. To overcome *overshoot* and *error* then needed controller on BLDC motor. Controller used is on Speed Controller, where speed controller serves as a speed regulator on the BLDC motor. At the speed controller there is fuzzy logic that serves to process the data that has been entered. By using speed control it is expected to minimize *overshoot* when motor rotation reaches maximum and *undershoot* when there is decreasing speed when there is load addition, motor rotation time reach *steady state* faster than reference journal as benchmark, *steady state* value according to reference speed has been given so that no *steady state error* occurs. In the system that has been run on the motor can achieve good results where *overshoot* and *undershoot* has a small value, good *rise time* and *steady state error* is very small.

**Keyword :** Brushless DC Motor, *Overshoot*, Speed Controller, Fuzzy Logic, *Steady state*

## I. Pendahuluan

Penggunaan motor di berbagai bidang industri saat ini sangat di perlukan, dengan adanya motor proses produksi bisa berjalan dengan cepat dan efisien. Ada berbagai macam motor yang digunakan di industri salah satunya adalah motor jenis DC yaitu motor Brushless DC (BLDC) atau motor arus searah tanpa sikat Motor Brushless DC (BLDC) adalah motor sinkron dengan magnet permanen pada rotor dan kumparan jangkar pada stator, magnet permanen menciptakan fluks pada rotor dan mengimbas pada stator sehingga menciptakan kutub elektromagnetik, rotor berputar karena imbas dari fase stator [1]. Keuntungan paling jelas dari konfigurasi tanpa sikat adalah penghilangan sikat, yang menghilangkan pemeliharaan sikat dan bunga api. Penggunaan kumparan

jangkar pada stator membantu konduksi panas pada kumparan. Karena tidak ada kumparan pada rotor, kerugian listrik dalam rotor menjadi minimal. Motor BLDC lebih disukai dibandingkan dengan motor induksi untuk daerah daya sebagian-sebagian [2].

Selain motor brushless DC terdapat juga motor DC konvensional, tetapi pada saat ini motor BLDC lebih banyak digunakan untuk menggantikan motor arus searah konvensional. Hal ini dikarenakan motor arus searah tanpa sikat memiliki performa yang lebih baik dibanding motor arus searah konvensional dilihat dari segi efisiensi, *lifetime*, *torque delivery*, dan operasi kecepatan tinggi [3].

Motor BLDC memiliki beberapa keunggulan jika dibandingkan dengan Motor DC konvensional. Keunggulannya antara lain perawatan menjadi lebih mudah karena motor ini tidak memiliki sikat, umur pemakaian lebih lama dan tidak menimbulkan *electrical noise*. Selain itu Motor BLDC memiliki efisiensi yang tinggi, respon dinamik yang tinggi, dan konsumsi energi yang kecil. [4]

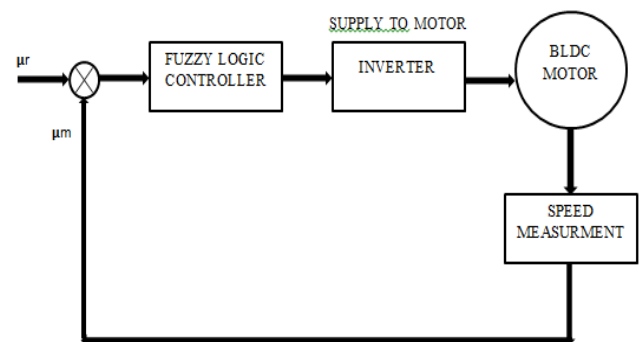
Pada dasarnya penggunaan motor yang utama di tekankan kepada putaran atau torsi, putaran motor harus konstan (tetap) dengan kecepatan yang telah ditentukan, namun masalah pada motor terjadi ketika ada penambahan pada beban. Dimana jika terdapat penambahan beban sedangkan respon kecepatan pada motor kurang baik maka dapat terjadi *undershoot* dan *error steady state*. Hal ini akan merugikan jika motor digunakan pada lingkungan industri. Untuk menangani masalah kurangnya respon kecepatan maka dapat diatur dengan menggunakan kontroler. Penggunaan kontroler dimaksudkan untuk mengatur kecepatan pada motor BLDC agar dapat mencapai *steady state*. Pada motor terdapat *speed* kontroler yang diterapkan metode *Fuzzy logic* untuk mengatur masukan dan keluaran dari kecepatan yang diinginkan. Kontrol Logika *Fuzzy* (*Fuzzy logic Controller*) merupakan salah satu bentuk Kontrol cerdas (intelligent control). Penggunaan kontrol logika *fuzzy* dapat menyelesaikan permasalahan pada sistem yang memiliki perilaku kompleks. Dalam logika *fuzzy*, tidak terdapat batas yang jelas, melainkan terdapat transisi antara set yang satu dengan set yang lainnya.

Transisi ini dikarakterisasi oleh fungsi keanggotaan yang memberikan sebuah *fuzzy set* fleksibilitas dalam memodelkan ekspresi linguistik pada *fuzzy* [5]

Terdapat beberapa penelitian sebelumnya tentang motor BLDC salah satunya oleh Richardus Dimas K.A. (2015). Melakukan penelitian Perancangan Kendali Kecepatan Mesin Arus Searah Tanpa Sikat dengan Menggunakan PID – Algoritma Genetika.[6] Pada penelitian kali ini akan menggunakan kontrol *fuzzy* yang di terapkan pada *speed* kontroler sebagai pengendali kecepatan pada motor BLDC.

## II. Metode Penelitian

### A. Blok Diagram Sistem



Gambar 1. Blok Diagram Sistem

Pada gambar 1 merupakan blok diagram sistem, dimana nilai input pada sistem berupa kecepatan. Metode Fuzzy logic di gunakan sebagai pengolah data dari pembacaan kecepatan real yang kemudian di komparasi dengan nilai input kecepatan yang telah di masukkan.

### B. Perancangan sistem

Pada penelitian ini akan di lakukan perancangan system pada motor BLDC yang terdiri dari dua bagian utama yaitu bagian motor (hardware) dan bagian controller. Pada penelitian nilai yang di tampilkan merupakan nilai dari pengukuran kecepatan motor yang di tampilkan dalam scope.

Terdapat 7 blok utama dalam perancangan system pada motor BLDC yang terdiri dari lima blok pada hardware dan 2 blok pada controller , yaitu:

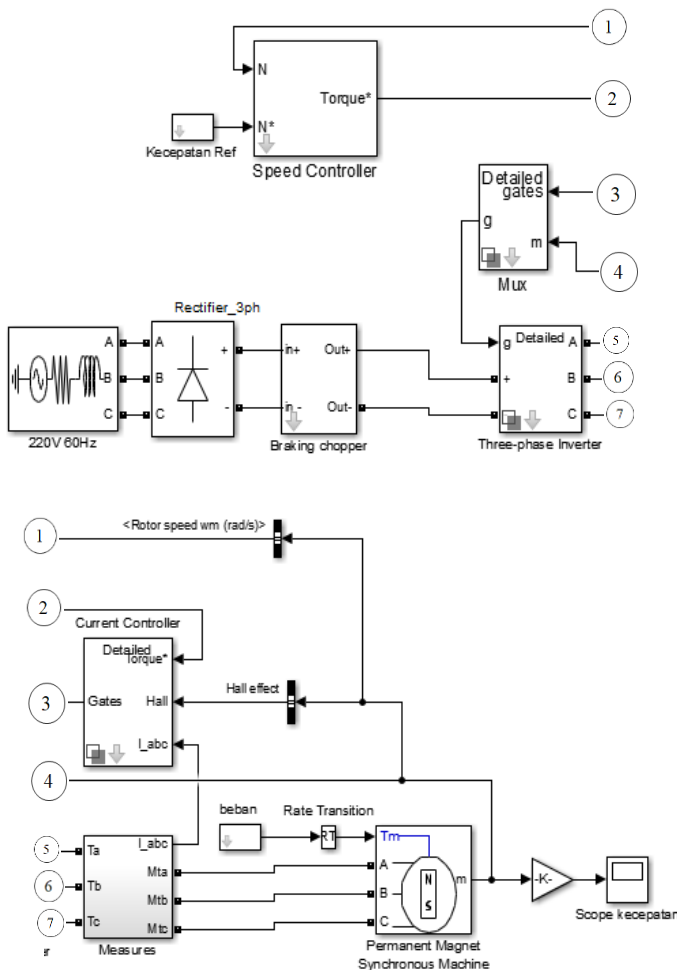
Pada blok hardware meliputi:

1. Sumber tegangan 220 v 60 hz
2. Rectifier 3 ph
3. Braking chooper
4. Inverter 3 ph
5. Motor BLDC

Pada blok controller meliputi:

1. Speed controller
2. Current controller

Dari penjelasan diatas berikut merupakan gambaran system motor BLDC secara keseluruhan.

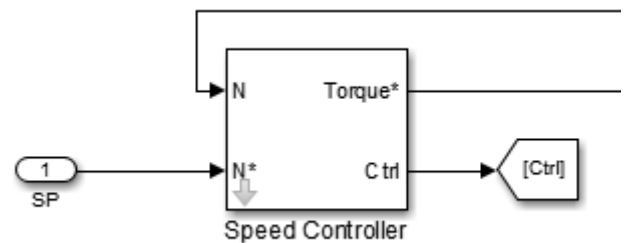


Gambar 2. Sistem motor BLDC

Pada sistem diatas terdapat angka didalam lingkaran, setiap angka yang terdapat dalam lingkaran merupakan garis yang tersambung.

### C. Speed Control

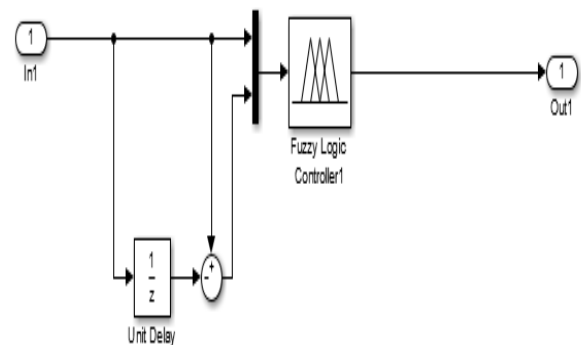
Pada penelitian ini blok *speed control* merupakan objek utama dalam penelitian. Dalam blok *speed controller* digunakan Fuzzy logic sebagai metode untuk pengolah data dari *speed referensi* (kecepatan referensi) yang di masukkan, keluaran dari *speed Controller* merupakan *Torsi referensi* dan sinyal PWM. Pada blok ini terdapat *feed back* dari pembacaan kecepatan aktual putaran motor yang nantinya akan di proses ulang sampai kecepatan putar motor sesuai dengan referensi.



Gambar 3. Block speed controller pada matlab Simulink

### D. Perancangan Fuzzy Logic Controller

Pada penelitian kali ini digunakan metode fuzzy logic untuk memproses dan mengatur nilai masukkan referensi kecepatan yang telah di masukkan.



Gambar 4. Diagram Simulink Controller Fuzzy

*Fuzzy logic controller* menggunakan variabel linguistik bukan variabel numerik. Dalam sistem kontrol, kesalahan antara sinyal referensi dan sinyal *output* [6] kesalahan sinyal dapat dijelaskan sebagai misalnya :

- NB : *Negative Big*
- NM : *Negative Medium*
- NS : *Negative Small*
- PS : *Positive Small*
- PM : *Positive Medium*
- PB : *Positive Big*
- Z : *Zero*

*Rule base* yang akan digunakan di tunjukkan pada tabel kesalahan yang terdiri dari 7 anggota dan memiliki keluaran sebanyak 49 anggota.

Beikut merupakan tabel *Rule Base Function*

Tabel 1 *Rule Base Function*

e/de	NB	NM	NS	Z	PS	PM	PB
NB	NB	NB	NB	NB	NM	NS	Z
NM	NB	NB	NB	NM	NS	Z	PS
NS	NB	NB	NM	NS	Z	PS	PM
Z	NB	NM	NS	Z	PS	PM	PB
PS	NM	NS	Z	PS	PM	PB	PB
PM	NS	Z	PS	PM	PB	PB	PB
PB	Z	PS	PM	PB	PB	PB	PB

E. Fungsi keanggotaan fuzzy

Fungsi keanggotaan (*membership function*) adalah suatu kurva yang menunjukkan pemetaan titik-titik input data ke dalam nilai keanggotaannya yang memiliki interval antara 0 sampai 1. Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah melalui pendekatan fungsi. Nilai keanggotaan didapatkan dari beberapa fungsi keanggotaan [7].

Ada beberapa fungsi yang dapat digunakan untuk memperoleh nilai keanggotaan, yaitu:

- a. Representasi Linier (Linier Naik dan Turun)
- b. Representasi Kurva Segitiga

- c. Representasi Kurva Trapesium
- d. Representasi kurva-S (Pertumbuhan dan Penyusutan)
- e. Representasi Kurva Bentuk Lonceng (Bell Curve)
  - 1. Kelas Kurva Phi (M)
  - 2. Kelas Kurva Beta (N)
  - 3. Kurva Gauss (O)

Himpunan fuzzy F dalam semesta X biasanya dinyatakan sebagai pasangan berurutan dari elemen x dan mempunyai derajat keanggotaan :

$$F = \{(x, \mu_F(x)) \mid x \in X\} \dots\dots\dots (1)$$

Dimana :

- F = Notasi himpunan fuzzy
- X = Semesta pembicaraan
- X = Elemen generik dari X
- $\mu_F(x)$  = Derajat keanggotaan dari x (nilai antara 0 dan 1)

Fungsi keanggotaan (*membership function*) dari himpunan fuzzy dapat disajikan dengan dua cara yaitu numeric dan fungsional. Secara numeric himpunan fuzzy disajikan dalam bentuk gabungan derajat keanggotaan tiap – tiap elemen pada semesta pembicaraan yang dinyatakan sebagai:

$$F = \sum \mu_F(u_i) / u_i \dots\dots\dots (2)$$

Secara fungsional himpunan fuzzy disajikan dalam bentuk persamaan matematis sehingga untuk mengetahui derajat keanggotaan dari masing-masing elemen dalam semesta pembicaraan memerlukan perhitungan.

F. Kurva Bentuk Lonceng

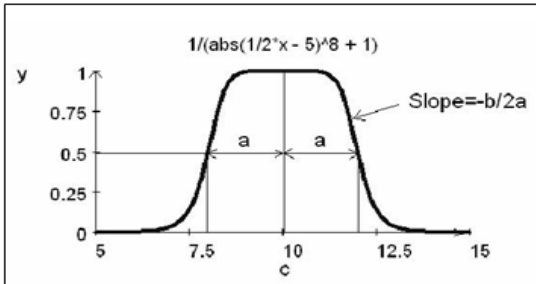
Pada penelitian ini fungsi nilai keanggotaan yang di gunakan adalah dalam bentuk lonceng (*gbellmf*) dimana pada awalnya kurva berbentuk segitiga yang kemudian di ubah kedalam bentuk lonceng. Parameter fungsi keanggotaan tipe GBell sebagai hasil distribusi pendekatan dari tipe segitiga dapat ditentukan sebagai berikut :

$$a = \frac{rtShoulder + lftShoulder}{2} - lftWaist \dots\dots\dots (3)$$

$$b = 2x \frac{a}{(lftShoulder - lftWaist)} \dots\dots\dots (4)$$

$$c = \frac{rtShoulder + lftShoulder}{2} \dots\dots\dots(5)$$

Berikut representasi kurva dalam bentuk lonceng (*generalized bell*) dalam bentuk gambar:



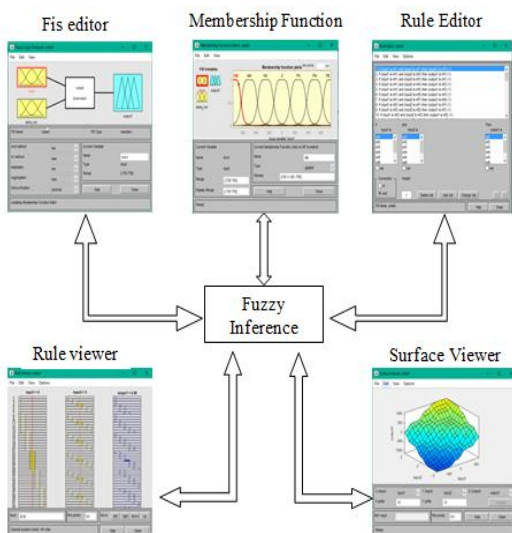
Gambar 5. Representasi fungsi keanggotaan bentuk lonceng

Fungsi keanggotaan kurva bentuk lonceng :

$$\mu[x] = \frac{1}{\left| \frac{x-c}{a} - \frac{c}{a} \right|^{2b} + 1} \dots\dots\dots(6)$$

Pemodelan Fuzzy Logic

Dalam *software* MATLAB *simulink* tersedia *Fuzzy Logic Toolbox* dimana memiliki 5 jenis *Graphic User Interface* (GUI) untuk merancang *fuzzy inference system* (FIS). Berikut merupakan gambar 5 jenis *Graphic User Interface* (GUI):



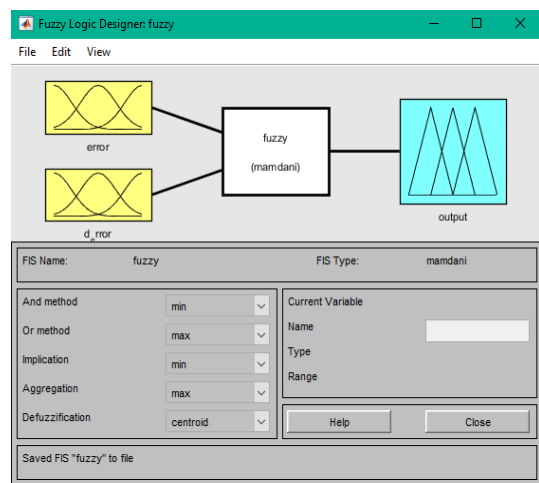
Gambar 6. Fuzzy Inference System pada Matlab

Dari gambar diatas GUI (*Graphic User Interface*) dapat di ketahui sebagai berikut:

1. Fis Editor
2. Membership function
3. Rule editor
4. Rule viewer
5. Surface viewer

D. Fis Editor

Pada Fis designer terdapat 3 variabel yaitu nilai error, delta error dan output. Dengan adanya fitur error dan delta error tersebut dapat digunakan untuk mengatur hasil *output* sesuai dengan kebutuhan. Gambar 7 merupakan jendela pada matlab yang muncul ketika kita menuliskan command “*fuzzy*” pada *command window*. Pada penelitian ini menggunakan sistem *Multi Input Single Output* (MISO) dibuat masukan berupa himpunan berupa variabel *input* tegangan error dan tegangan delta error dengan *output* hasil.

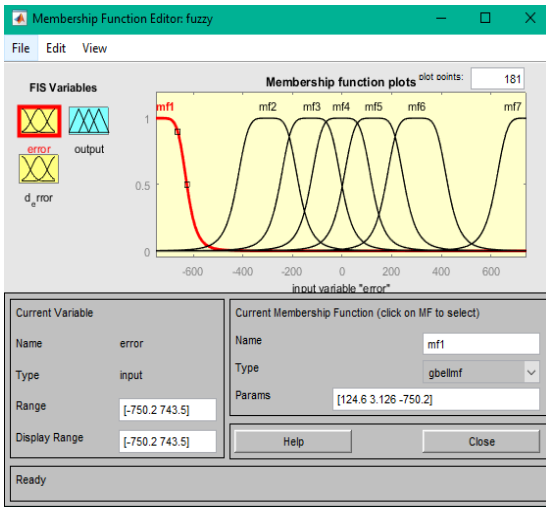


Gambar 7. MATLAB FIS editor

Pada desain fuzzy diatas menggunakan tipe fuzzy mamdani. Variabel linguistik yang dibentuk dalam tujuh fungsi keanggotaan. Dimana masing-masing himpunan pada *error* dan *delta error* dibagi menjadi tujuh fungsi keanggotaan sehingga terdapat 49 fungsi keanggotaan . Fungsi keanggotaan yang digunakan menggunakan metode tak terhingga, dengan menggunakan tipe gbellmf



(*generalized bell*). Sehingga untuk konfigurasiya bisa dilihat pada gambar 8.



Gambar 8. Membership Function FIS Editor

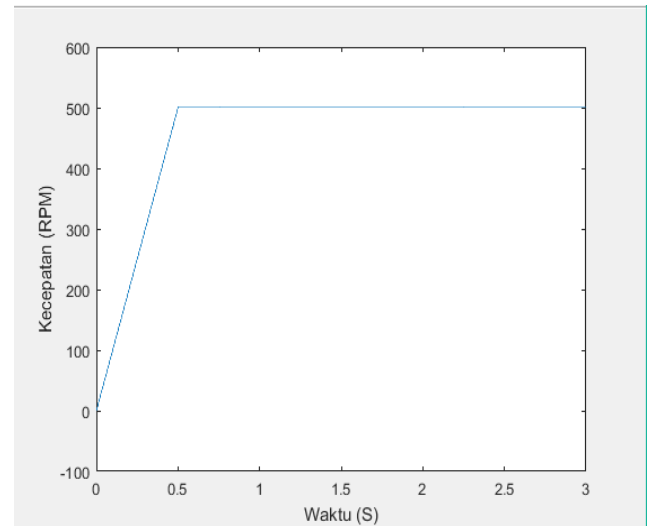
Pada Gambar 8 terlihat fungsi keanggotaannya berbentuk tak terhingga dengan type membership function *gbellmf*. Mengingat dalam algoritma *fuzzy* ini kita melakukan normalisasi terhadap nilai bilangan sehingga didapatkan nilai amplitudo minimal dan maksimal bernilai -750.2 sampai 743.5 pada bagian *error*, -55 sampai 76 pada bagian *delta error* dan -2900 sampai 2900 pada *output*. Dimana nilai tersebut adalah nilai yang terendah dan tertinggi pada masing-masing *membership fuction plot* atau bisa juga disebut fungsi keanggotaan yang bersifat mutlak.

### III. Hasil dan Pembahasan

Pengujian pada sistem meliputi pengujian kecepatan motor pada range kecepatan rendah, kecepatan menengah dan kecepatan tinggi serta perubahan kecepatan dan pemberian beban. Pada masing-masing pengujian akan dilakukan pencatatan hasil yang meliputi *rise time* (waktu puncak), *overshoot*, *undershoot* dan waktu mencapai *steady state*. Pengujian akan dilakukan dengan menggunakan beberapa tingkat kecepatan yaitu kecepatan rendah (500rpm), Kecepatan menengah (1000rpm) dan kecepatan tinggi (1500 rpm). Pada setiap

pengujian akan diberikan penjelasan tentang hasil dan tabel untuk mempermudah pembacaan hasil running.

- A. Pengujian pada kecepatan rendah (500rpm)  
 Pada pengujian ini menggunakan kecepatan rendah dimana kecepatan yang digunakan adalah sebesar 500 rpm.



Gambar 9. Pengujian Sistem Pada Kecepatan 500 rpm

Pada pengujian dengan kecepatan 500 rpm waktu yang di perlukan oleh system untuk mencapai *rise time* (waktu puncak) adalah pada 0,5 detik, *overshoot* sebesar 0.53 rpm (0.106 %), *Undershoot* 0.125 rpm (0.025 %), sistem mencapai *steady state* pada waktu 0.505 detik.

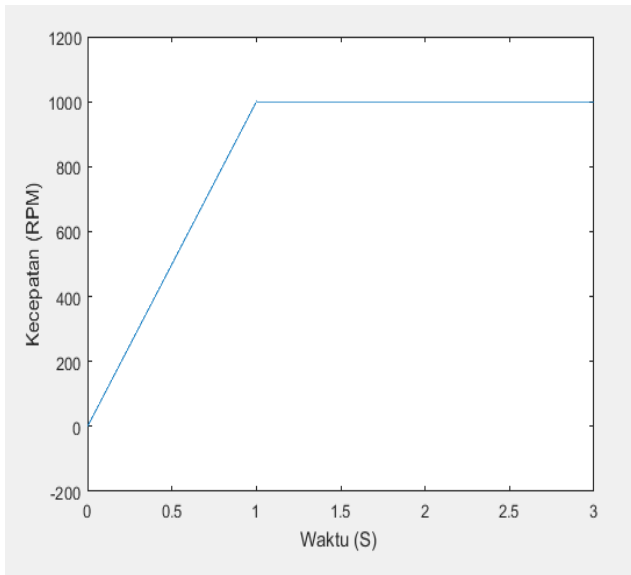
Berikut adalah hasil pengujian yang di ubah dalam bentuk tabel :

Tabel 2. Pengujian Pada Kecepatan 500 rpm

NO	Parameter	Value
1	Rise time	0.5 detik
2	Overshoot	0.53 rpm (0.106%)
3	Undershoot	0.125 rpm (0.025 %)
4	Steady state	0.505 Detik

**B. Pengujian Pada kecepatan Menengah (1000rpm)**

Pada pengujian ini menggunakan kecepatan menengah dimana kecepatan yang digunakan adalah sebesar 1000 rpm.



Gambar 10. Pengujian Sistem Pada Kecepatan 1000 rpm

Pada pengujian dengan kecepatan 1000 rpm waktu yang di perlukan sistem untuk mencapai *rise time* (waktu puncak) adalah pada 1 detik, *overshoot* sebesar 0.45 rpm (0.045 %), *Undershoot* 0.15 rpm (0.015 %), sistem mencapai *steady state* pada waktu 1.008 detik.

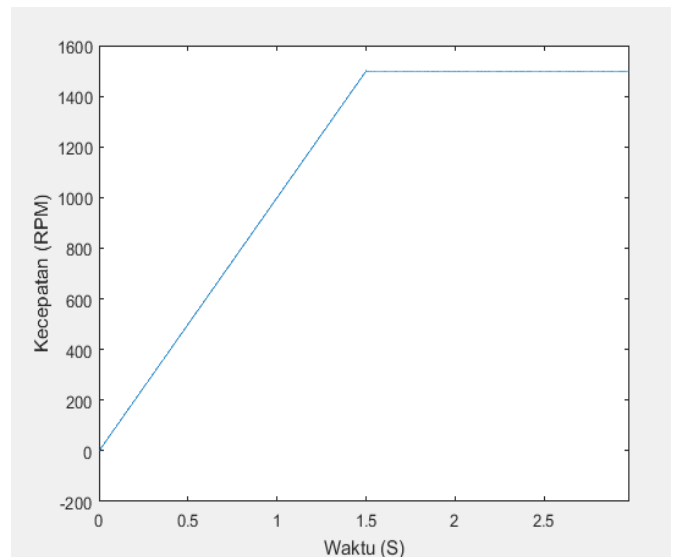
Berikut adalah hasil pengujian yang di ubah dalam bentuk tabel :

Tabel 3. Pengujian Pada Kecepatan 1000 rpm

No	Parameter	Value
1	Rise time	1 detik
2	Overshoot	0.45 rpm (0.045%)
3	Undershoot	0.15 rpm (0.015 %)
4	Steady state	1.08 detik

**C. Pengujian Pada kecepatan Tinggi (1500rpm)**

Pada pengujian ini menggunakan kecepatan tinggi dimana kecepatan referensi yang digunakan adalah sebesar 1500 rpm.



Gambar 11. Pengujian Sistem Pada Kecepatan 1500 rpm

Pada pengujian dengan kecepatan 1500 rpm waktu yang di perlukan system untuk mencapai *rise time* (waktu puncak) adalah pada 1,5 detik, *overshoot* sebesar 0.45 rpm (0.03 %), *Undershoot* 0.15 rpm (0.01 %), sistem mencapai *steady state* pada waktu 1.509 detik.

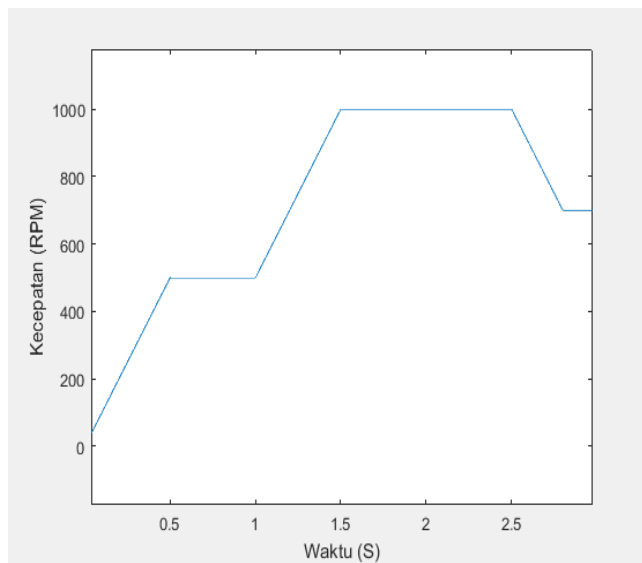
Berikut adalah hasil pengujian yang di ubah dalam bentuk tabel :

Tabel 4. Pengujian pada kecepatan 1500 rpm

NO	Parameter	value
1	Rise time	1,5 detik
2	Overshoot	0.45 rpm (0.045%)
3	Undershoot	0.15 rpm (0.015 %)
4	Steady state	1,509 detik

**D. Pengujian pada kecepatan bertingkat**

Pada pengujian ini akan digunakan kecepatan yang berbeda dimana kecepatan referensi yang diberikan akan dilakukan peningkatan dan penurunan kecepatan yaitu pada kecepatan 500 rpm naik ke 1000 rpm dan turun ke 700 rpm hal ini dilakukan untuk mengetahui respon sistem pada saat kecepatan dinaikkan atau di turunkan pada waktu tertentu.



Gambar 12. Pengujian dengan Kecepatan Berubah

Pada gambar 10 kecepatan awal adalah 500rpm, kemudian kecepatan dinaikkan menjadi 1000 rpm pada detik ke 1 dan diturunkan menjadi 700 rpm pada detik ke 2,5 jadi pada pengujian ini akan dilihat respon sistem pada saat kecepatan dinaikkan dan di turunkan..

Pada kecepatan 500 rpm waktu yang di perlukan untuk mencapai *rise time* (waktu puncak) adalah pada 0,5 detik, *overshoot* sebesar 0.53 rpm (0.106 %), *Undershoot* 0.125 rpm (0.025 %), sistem mencapai *steady state* pada waktu 0.505 detik.

Pada detik ke 1 dilakuakn penaikan kecepatan dari 500 rpm ke kecepatan 1000 rpm waktu yang di perlukan untuk mencapai *rise time* (waktu puncak) dari kecepatan 500 rpm ke kecepatan 1000 rpm adalah 0,5 detik pada detik ke 1,5, *overshoot* sebesar 0.45 rpm (0.045 %), *Undershoot* 0.15 rpm (0.015 %), sistem mencapai *steady state* pada waktu 1.508 detik.

Pada detik ke 2.5 dilakukan pengurangan kecepatan dari 1000 rpm ke kecepatan 700 rpm. Waktu yang di perlukan sistem untuk mencapai kecepatan 700 rpm adalah 0.3 detik, *Undershoot* sebesar 0.5 rpm (0.071 %), *overshoot* sebesar 0.2 rpm (0.028 %), sistem mencapai *steady state* pada waktu 2.808 detik.

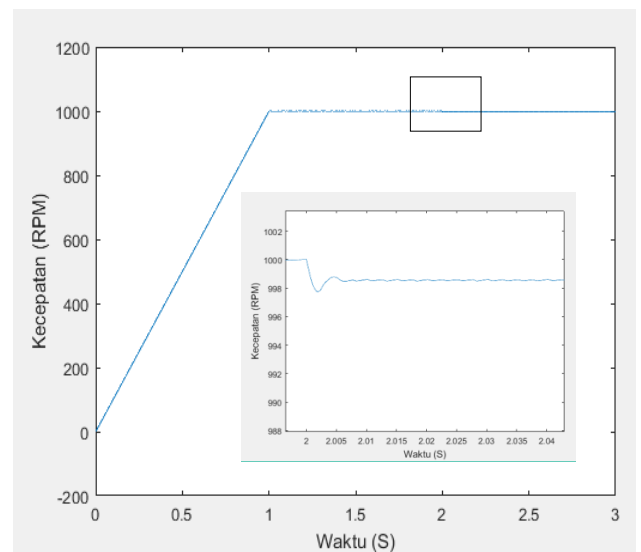
Berikut adalah hasil pengujian yang di ubah dalam bentuk tabel :

Tabel 5. Pengujian kecepatan bertingkat

No	Kecepatan	Parameter	Value
1	500 rpm	Rise time	0.5 detik
		Overshoot	0.53 rpm (0.106%)
		Undershoot	0.125 rpm (0.025 %)
		Steady state	0.506 Detik
2	1000 rpm	Rise time	1,5 detik
		Overshoot	0.45 rpm (0.045%)
		Undershoot	0.15 rpm (0.015 %)
		Steady state	1.508 Detik
3	700 rpm	Overshoot	0.45 rpm (0.045%)
		Undershoot	0.15 rpm (0.015 %)
		Steady state	2.808detik

E. Pengujian berbeban pada kecepatan 1000rpm

Pada pengujian ini diberikan beban maksimal yang mampu di tanggung oleh sistem sebesar 17 N.M dengan kecepatan referensi sebesar 1000 rpm.



Gambar 13. Pengujian Dengan Beban

Pada gambar 11 kecepatan referensi yang diberikan sebesar 1000 rpm waktu yang di perlukan untuk mencapai *rise time* (waktu puncak) adalah pada detik ke 1, *overshoot* sebesar 0.45 rpm (0.045 %), *Undershoot* 0.15 rpm (0.015 %), sistem mencapai *steady state* pada waktu 1.008 detik. Pada detik ke 2 di berikan beban sebesar 17 NM terjadi penurunan kecepatan sebesar



2.25rpm sehingga kecepatan motor menjadi 997.75rpm sistem merespon dan kecepatan naik menjadi 998.5 dan terjadi error *steady state* sebesar 0.15 %.

Tabel 6. Pengujian berbeban

NO	Beban	Parameter	value
1	0 nm	Rise time	1 detik
		Overshoot	0.45 rpm (0.045%)
		Undershoot	0.15 rpm (0.015 %)
		Steady state	1.08 detik
2	17 nm	undershoot	2.25 rpm (0.225%)
		Error steady state	1.5 rpm (0.15%)

#### IV. Kesimpulan

Kesimpulan yang di dapat pada penelitian dan percobaan yang telah di lakukan di atas adalah sebagai berikut

1. Motor dapat bekerja dengan baik dimana *overshoot* dan *undershoot* sangat kecil yaitu kurang dari 1 rpm.
2. Pada setiap pengujian dengan kecepatan yang berbeda seperti 500,1000 dan 1500 rpm maka *overshoot*, *undershoot* dan *rise time* memiliki nilai yang berbeda.
3. Pada saat di berikan beban maksimal pada sistem penurunan kecepatan kecil yaitu 2.25 rpm dan kecepatan naik kembali namun terjadi error *steady state*.
4. Waktu dari *rise time* menuju ke *steady state* memiliki rentang waktu yang singkat dengan rata-rata waktu 0.008 detik.

#### Daftar Pustaka

- [1] B. Akin, M. Bhardwaj. 2011. Trapezoidal Control of BLDC Motors Using Hall Effect Sensors. Texas Instrumen.
- [2] H. A. Toliyat dan T. Gopalarathnam. 2002. The Power Electronics Handbook, CRC Press LLC.

- [3] L. K. Baxter, 1996. Capacitive Sensors: Design and Applications, John Wiley & Sons.
- [4] Baldursson, Stefán. BLDC Motor Modelling and Control-A Matlab®/Simulink® Implementation, Master Thesis Institutionen för Energi och Miljö Chalmers Tekniska Hogskola. 2005.
- [5] J. S. R. Jang, C. T. Sun dan E. Mizutani. 1997. Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence, Upper Saddle River: Prentice Hall.
- [6] R. Dhimas, K. Agustika, and P. H. Sasongko, “dengan Menggunakan PID – Algoritma Genetika,” vol. XIX, no. November, pp. 2–5, 2015.
- [7] Hosein Zadeh, H. Kalam, H. “A Rule-Based Fuzzy Power System Stabilizer Turned by Neural Network. IEEE Transactions on Energy Conversion, vol. 14, No. 3, September 1999.
- [8] Kusumadewi, Sri dan Purnomo, Hari. 2004. Aplikasi Logika Fuzzy untuk Pendukung Keputusan. Yogyakarta: Graha Ilmu.