

KENDALI KECEPATAN PUTAR BRUSHLESS DC MOTOR (BLDC) HALL SENSOR BERBASIS MIKROKONTROLER

ROTATING SPEED CONTROL OF BRUSHLESS DC MOTOR (BLDC) HALL SENSOR BASED ON MICROCONTROLLER

Hari Prasetijo^{*1}, Vita Anjarwati², Priswanto³, Daru Tri Nugroho⁴

*Email: hari.prasetijo@unsoed.ac.id

^{1,2,3,4}Jurusan Teknik Elektro, Fakultas Teknik, Universitas Jenderal Soedirman, Purwokerto

Abstrak— *Brushless DC Motor (BLDC Motor)* merupakan motor AC sinkron 3 fase menggunakan rotor magnet permanen dengan komutasi elektrik bukan mekanis maka BLDC motor memiliki efisiensi, kecepatan, dan torsi yang lebih baik dibanding motor DC. Medan putar stator BLDC motor 3 fase dapat terjadi jika suplai tegangan yang tepat pada masing-masing belitan fasa. Suplai tegangan dapat diperoleh dari *output inverter* sehingga perlu kendali tegangan untuk memicu kerja *gate* pada *inverter* agar terjadi komutasi elektrik yang tepat. Studi ini melakukan rancang bangun kendali putaran BLDC motor sensor *hall*. Sensor *hall* digunakan untuk mendeteksi posisi kutub magnet rotor untuk *input* pengatur tegangan pemacu *gate inverter* sehingga terjadi komutasi yang tepat. Pengaturan kecepatan dilakukan dengan metode *fuzzy*. Perbandingan performa pengaturan tegangan dengan dan tanpa FLC (*Fuzzy Logic Control*) berbasis Arduino dilakukan secara *software* dan *hardware*. Pada pengujian simulasi menggunakan *software*, pada *set point* 1200 rpm *output* kecepatan tanpa FLC berada di kisaran 1400 rpm, sedangkan ketika menggunakan FLC berada di kisaran 1200 rpm. *Rise time* yang dihasilkan tanpa FLC yaitu 28,955 ms dan *overshoot* yang dihasilkan 1,531%, sedangkan ketika menggunakan FLC menghasilkan *rise time* 15,385 ms dan *overshoot* yang dihasilkan 0,471%. Pada pengujian *hardware* tanpa menggunakan FLC pada *set point* 1200 rpm *output* kecepatan yang dihasilkan yaitu berada dikisaran 1341 rpm, sedangkan menggunakan FLC menghasilkan *output* kecepatan berada di 1212 rpm. *Rise time* yang dihasilkan ketika tanpa menggunakan FLC adalah 100 ms dan ketika menggunakan FLC adalah 25 ms. Jadi, dapat disimpulkan sistem kendali kecepatan motor saat menggunakan FLC menghasilkan *output* kecepatan dan respon sistem yang lebih baik dibandingkan dengan yang tidak menggunakan FLC.

Kata kunci — BLDC motor, mikrokontroler, sensor *hall*, kecepatan.

Abstract— *Brushless DC Motor (BLDC Motor)* is a 3-phase synchronous AC motor using a permanent magnet rotor with non-mechanical electrical commutation, so BLDC motors have better efficiency, speed and torque than DC motors. The rotating field of a 3-phase BLDC motor stator can occur if the proper voltage is supplied to each phase winding. The supply voltage can be obtained from the output of the inverter so it is necessary to control the voltage to trigger the gate work on the inverter so that proper electrical commutation occurs. This study designs a hall sensor motor BLDC rotation control. Hall sensor is used to detect the magnetic pole position of the rotor to input the gate inverter trigger voltage regulator so that proper commutation occurs. Speed regulation is done by fuzzy method. Comparison of the performance of voltage regulation with and without FLC (*Fuzzy Logic Control*) based on Arduino is done by software and hardware. In the simulation test using software, at the set point of 1200 rpm the output speed without FLC is in the range of 1400 rpm, while when using FLC it is in the range of 1200 rpm. The resulting rise time without FLC is 28.955 ms and the resulting overshoot is 1.531%, whereas when using FLC it produces a rise time of 15.385 ms and the resulting overshoot is 0.471%. In hardware testing when without using FLC at a set point of 100 rpm the output speed is in the range of 1341 rpm, while when using FLC the output speed is in the range of 1212 rpm. Rise time generated when without using FLC is 100 ms and when using FLC is 25ms. So it can be concluded that the motor speed control system when using FLC produces better speed output and system response compared to those that do not use FLC.

Keywords — BLDC motor, microcontroller, hall sensor, speed.

I. PENDAHULUAN

Seiring perkembangan ilmu pengetahuan dan teknologi, terdapat banyak penemuan baru sehingga hal tersebut dapat mempermudah dalam melakukan sesuatu pekerjaan.. Salah satu dari pengembangan teknologi tersebut adalah berupa peralatan yang sering dibutuhkan dan digunakan dalam kehidupan sehari-hari yaitu motor penggerak. Permasalahan global berupa pemanasan global dan penghematan penggunaan bahan bakar minyak merupakan hal yang menjadi pendongkrak bagi perkembangan teknologi motor listrik. Motor DC konvensional sudah dikenal efisien dan bebas polusi, Namun Motor DC masih memiliki kekurangan yang perlu diperbaiki, sehingga diperlukan pengembangan maupun perancangan motor listrik baru untuk mendapatkan performa motor listrik yang lebih baik [1]

Brushless DC (BLDC) merupakan motor magnet permanen dimana proses *comutator* dan sikatnya digantikan oleh sebuah *solid state switch*. Controller Brushless DC Motor (BLDC) tiga fasa memiliki peran sangat penting karena merupakan penunjang utama beroperasinya motor DC Brushless. Hal tersebut dikarenakan motor ini membutuhkan suatu trigger pulsa ke bagian elektromagnetik (*stator*) dari motor DC Brushless sebagai pengaturan besarnya arus yang mengalir sehingga motor dapat berputar dan diatur secara akurat [2].

Selanjutnya, Motor Brushless DC (BLDC) pada kondisi lain dapat juga disebut sebagai motor sinkron atau Permanent Magnet Synchronous Motor (PMSM). Motor ini memiliki beberapa kelebihan diantaranya efisiensi yang tinggi, keandalan yang tinggi, kepresisian dan kecepatan yang juga tinggi. Sehingga, pada saat ini banyak para pelaku bidang industri sebagai pengguna motor BLDC karena manfaat kelebihannya tersebut [3]. Namun, untuk beberapa kondisi motor BLDC tidak cukup handal, seperti ketika digunakan secara jangka panjang, dan juga motor BLDC mempunyai harga yang lebih mahal dibandingkan motor elektrik lainnya [4]. Motor BLDC pada umumnya terdiri dari konstruksi tiga fasa, dengan tiga magnet kumparan listrik yang terletak di stator, dan dinyalakan sesuai dengan enam transistor MOSFET yang berbeda [5],

Motor BLDC memiliki efisiensi, keandalan, dan rentang kecepatan yang tinggi [6]. Selain itu, motor BLDC membutuhkan perawatan yang lebih sedikit dan dapat beroperasi lebih tenang daripada motor DC. Wang [7] membuat kendali BLDC motor tiga fasa menggunakan inverter berbasis MOSFET. Sensor *hall* digunakan sebagai sinyal input

kecepatan motor untuk mendapatkan output sinyal PWM. Beladjine [8] melakukan studi simulasi perbandingan antara kontrol PI dan kontrol berbasis *Artificial Neural Network (ANN)* dari BLDC motor. Saturasi yang menyebabkan ketidakstabilan sistem pada kontrol PI dapat diperbaiki dengan kontrol berbasis ANN untuk menghasilkan respons yang lebih dinamis dan cepat.

II. TINJAUAN PUSTAKA

Brushless DC motor (BLDC motor) merupakan salah satu usaha pengembangan teknologi motor listrik untuk memenuhi kriteria motor listrik yang memiliki efisiensi tinggi, torsi yang tinggi, kecepatan putar yang tinggi dan dapat divariasikan, dan biaya perawatan yang rendah. Brushless DC (BLDC) motor sebenarnya adalah motor sinkron AC 3 fase dengan medan magnet putar yang dihasilkan stator dan medan magnet yang dihasilkan oleh putaran rotor memiliki frekwensi yang sama [9].

Pengembangan teknologi BLDC motor telah dilakukan oleh beberapa peneliti: [10] melalui simulasi menyatakan BLDC motor memiliki torsi yang tinggi namun *torque ripple* yang terjadi pada proses komutasi akan menyebabkan sedikit vibrasi akibat torque spike and dip. [11] secara analitis membuktikan gelombang AC 3 fase bentuk trapezoid input BLDC motor menghasilkan efisiensi yang lebih tinggi dibanding bentuk square namun memerlukan umpan balik posisi magnet. [12] melakukan simulasi dan eksperimen dengan memodifikasi *slot stator* sehingga mengurangi *peak to peak cogging torque* sebesar 70% dan *torque ripple* 17% sehingga pengoperasian BLDC motor lebih *smooth* pada kecepatan rendah. [13] melalui simulasi dan eksperimen menemukan jika laju perubahan arus komutasi 2 fase dengan mempertahankan arus konstan pada satu fase lain yang tidak berkumutasi dapat mengurangi *torque ripple* pada putaran tinggi. Ref. [14] menggunakan mikrokontroler untuk mengendalikan kecepatan putar dan torsi BLDC motor dengan sinyal PWM melalui modul Electronic Speed Control (ESC) [15] setelah melakukan simulasi menyampaikan bahwa pengaturan kestabilan kecepatan BLDC motor pada beban dinamis memungkinkan motor menghasilkan konsistensi torsi tinggi. [16] melakukan eksperimen kendali kecepatan putar dengan mendeteksi *zero crossing point back EMF* sebagai alternatif yang lebih sederhana dan murah dibanding penggunaan sensor posisi. Ref. [17] melakukan simulasi dan eksperimen menggunakan metode *six-step* dalam

pengendalian BLDC motor. Gelombang yang dihasilkan *inverter* dari metode ini berbentuk *square* atau trapezoid. Pada BLDC motor 3 fase, sensor *hall* dipasang pada *stator* untuk mendeteksi medan magnet rotor. Sinyal dari sensor *hall* digunakan untuk menentukan komutasi dengan tepat pada 2 fasa kumparan stator. Pemasangan sudut sensor yang kurang akurat akan menurunkan performa BLDC motor. [18] melalui simulasi dan eksperimen menunjukkan metode deteksi arus stator untuk mengetahui keakuratan posisi maupun sensitivitas sensor *hall*.

Studi ini melakukan dilakukan rancang bangun kendali putaran BLDC motor sensor *hall* berbasis mikrokontroler Arduino. Penggunaan sensor *hall* diperlukan untuk mendeteksi posisi kutub magnet rotor untuk input pengatur tegangan pemicu *gate inverter* sehingga terjadi komutasi yang tepat. Pengaturan kecepatan dilakukan dengan metode fuzzy menggunakan software Matlab dengan *input* dari *optocoupler* sebagai deteksi kecepatan. Perbandingan performa dilakukan untuk kendali menggunakan mikrokontroler tanpa dan dengan logika *fuzzy*.

III. METODE

Metode penelitian yang akan dilakukan yaitu (1) perancangan sistem, (2) pembuatan *hardware* dan (3) pengujian. Perancangan sistem meliputi perancangan rangkaian, pemrograman Arduino, pemrograman logika fuzzy. Sementara itu pada tahap Pembuatan hardware meliputi mikrokontroler, *driver* dan *sensing* kecepatan. Sedangkan tahapan pengujian dilakukan untuk mengetahui performa *hardware* serta membandingkan dengan hasil simulasi.

A. Perancangan Sistem

Perancangan Rangkaian Simulasi

Perancangan rangkaian menggunakan *software* Proteus yang digunakan untuk menyimulasikan sistem yang terdiri mikrokontroler, *driver*, BLDC motor, DC *power supply*, sensor *hall*, kontroler kecepatan putar motor.

Programming Arduino IDE

Programming Arduino dibuat sebagai *software* untuk mengoperasikan simulasi perancangan sistem yang telah dibuat dalam *software* proteus dengan mengupload program ke mikrokontroler Arduino. Pemrograman meliputi *input-output*, *pulse with modulation* (pwm), dan fuzzy. Program *input-output* dibuat agar mikrokontroler dapat membaca input

sinyal tegangan analog hasil sensing kecepatan BLDC motor, selanjutnya memberikan sinyal digital ke *driver*. Program pwm dibuat agar mikrokontroler dapat memberikan sinyal ke *driver* agar memberikan variasi besar tegangan ke motor.

Programming Fuzzy

Program fuzzy, dibuat untuk mengatur kecepatan putar BLDC motor menggunakan software Matlab Simulink. Prosedur logika fuzzy yang digunakan meliputi proses fuzzyfikasi, evaluasi rule Mamdani dan proses defuzzifikasi. Data diambil berdasarkan perubahan dari variasi setting point dan jenis fuzzyfikasi. Untuk memperoleh tanggapan kecepatan yang sesuai, dilakukan dengan perubahan fungsi keanggotaan dengan rentang nilai tertentu. Data kecepatan dimasukkan ke dalam sistem pengendali agar diperoleh data koreksi kecepatan (*error*) dan selisih koreksi kecepatan (Δ_{error}), sedangkan output berupa *error* yang semakin kecil.

B. Pembuatan Hardware

Mikrokontroler

Mikrokontroler yang digunakan adalah Arduino uno 328. Agar mikrokontroler dapat menjalankan fungsinya, maka program yang telah dibuat dalam laptop menggunakan software Arduino IDE harus di-*upload* ke mikrokontroler. Mikrokontroler dihubungkan melalui kabel usb dengan laptop yang telah terinstall driver Arduino uno, kemudian upload program dilakukan dari laptop ke mikrokontroler.

Driver

Driver dibuat dengan membuat rangkaian 6 buah mosfet pada *board*. Setiap saat driver memberikan tegangan DC pada 2 kumparan stator yang sesuai secara bergantian sehingga terjadi komutasi yang tepat.

Sensing kecepatan

Sensor Optocoupler digunakan untuk mendeteksi kecepatan BLDC motor. Cara kerja dari sensor optocoupler adalah bila terhalang maka output akan *open*, dan bila tidak terhalang output akan *short*. Dengan cara kerja tersebut, sinar inframerah akan putus-putus dan menimbulkan pulsa-pulsa listrik. Pulsa pulsa itu kemudian dapat diolah dan ditangkap oleh mikrokontroler.

C. Pengujian

Pengujian dilakukan untuk melihat performa pengendalian sistem *hardware* dan simulasi untuk mengetahui output kecepatan, rise time dan *overshoot* dengan dan tanpa FLC. Klasifikasi kecepatan yang dilakukan dengan fuzzy adalah kecepatan rendah, sedang, dan tinggi.

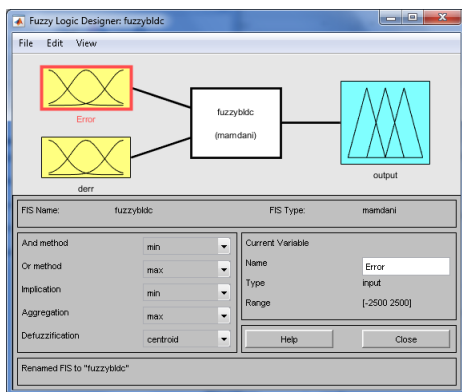
HASIL DAN PEMBAHASAN

Perancangan *Software*

Perancangan Kontroler Fuzzy Logic

Software yang dirancang dengan menggunakan Matlab adalah berupa sistem inferensi fuzzy yang ditampilkan dalam bentuk grafis. Terdapat 2 perancangan kontroler fuzzy yaitu untuk Simulasi pada Simulink MATLAB dan untuk pengujian pada hardware. Adapun hasil perancangan yang telah dibuat dapat dilihat pada gambar di bawah ini.

a Kontroler *Fuzzy* untuk Simulink MATLAB

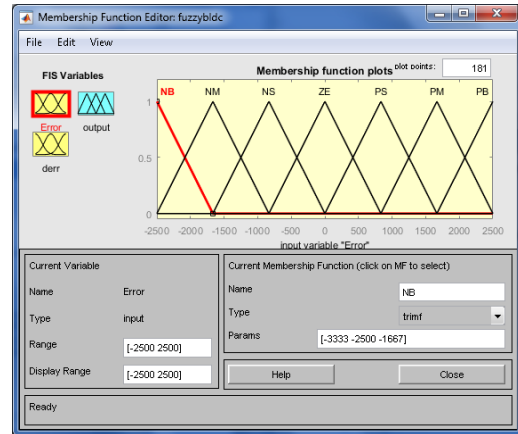


Gambar 1. Perancangan sistem inferensi fuzzy

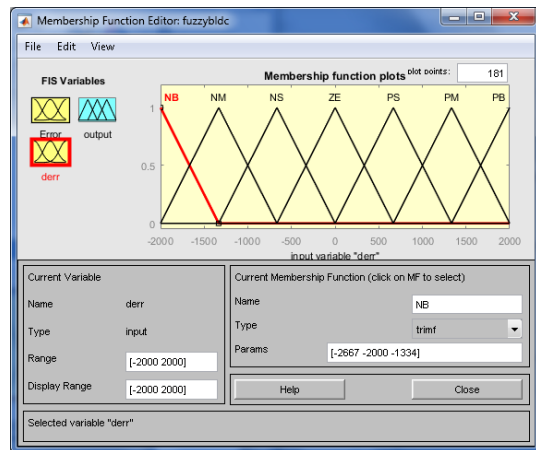
Prosedur logika fuzzy yang dilakukan meliputi proses fuzzifikasi, evaluasi rule/aturan dengan metode Mamdani dan proses defuzzifikasi menggunakan metode Center of Area (CoA). Sistem logika fuzzy yang dirancang mempunyai 2 masukan yaitu error dan Δ error serta 1 keluaran yaitu PWM. Selain itu sistem tersebut juga menggunakan aturan sebanyak 49 basis aturan (rule base).

Dalam sistem ini dirancang fungsi keanggotaan *error* dan Δ error, dan PWM dengan jumlah label 7 label yaitu NB (Negative Big), NM (Negative Medium), NS (Negative Small), ZE (Zero), PS (Positive Small), PM (Positive Medium), PB (Positive Big). Fungsi keanggotaan pada logika fuzzy ini mempunyai bentuk segitiga. Nilai-nilai yang ada pada fungsi keanggotaan masukan dan keluaran fuzzy didapatkan dari trial and error pada

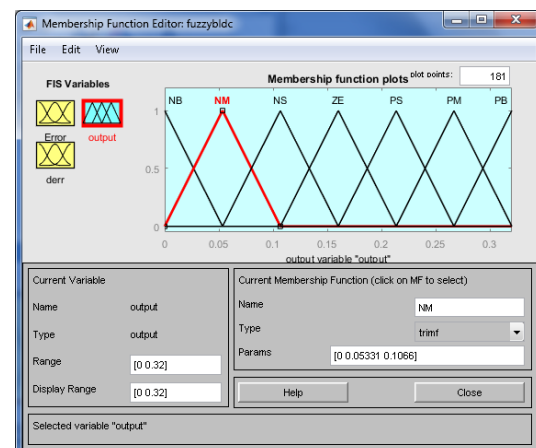
rangkain *simulink*. Adapun hasil perancangan fungsi keanggotaan yang telah dibuat dapat dilihat pada gambar di bawah ini.



Gambar 2. Grafik fungsi keanggotaan variabel error



Gambar 3. Grafik fungsi keanggotaan variabel Δ error



Gambar 4 Grafik fungsi keanggotaan variabel output PWM

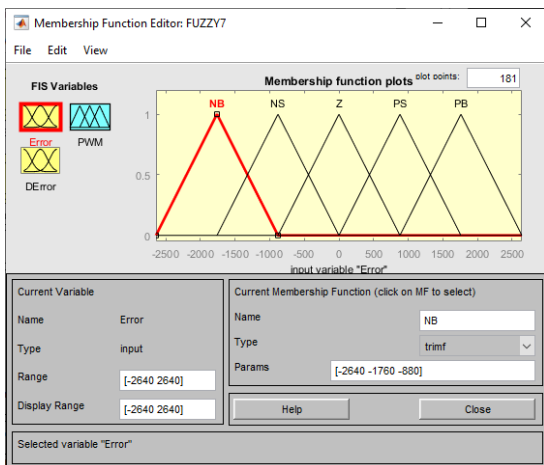
Basis aturan merupakan keseluruhan aturan dari kombinasi dua masukan yang mungkin. Secara lengkap jumlah kombinasi yang mungkin dari dua himpunan masukan fuzzy pada perancangan sistem pengendali kecepatan motor BLDC dengan fungsi

keanggotaan tujuh dikali tujuh adalah empat puluh sembilan aturan seperti pada Tabel 1.

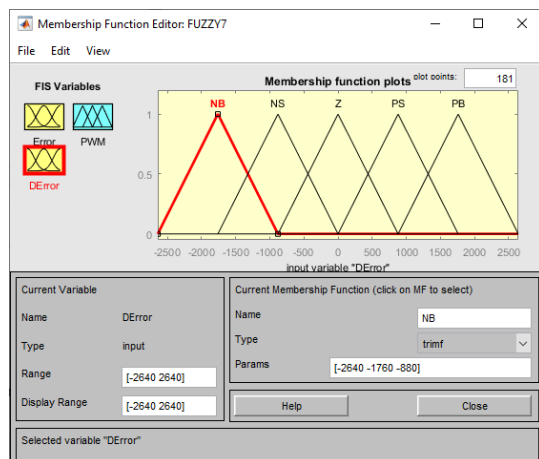
Tabel 1. Basis Aturan Fuzzy

	PWM	Error						
		NB	NM	NS	ZE	PS	PM	PB
Δ error	NB	NB	NB	NB	NB	NM	NS	ZE
	NM	NB	NB	NB	NM	NS	ZE	PS
	NS	NB	NB	NM	NS	ZE	PS	PM
	ZE	NB	NM	NS	ZE	PS	PM	PB
	PS	NM	NS	ZE	PS	PM	PB	PB
	PM	NS	ZE	PS	PM	PB	PB	PB
	PB	ZE	PS	PM	PB	PB	PB	PB

b Kontroler Fuzzy untuk pengujian hardware



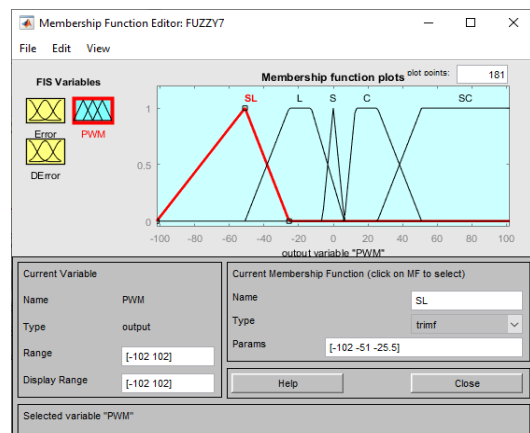
Gambar 5 Grafik fungsi keanggotaan variabel error hardware



Gambar 6 Grafik fungsi keanggotaan delta error hardware

Untuk basis aturan (*rule base*) pada kontroler fuzzy yang digunakan untuk pengujian pada hardware hampir sama dengan rule base yang ada pada rangkaian Simulink seperti pada Tabel 1, yang membedakan hanyalah tidak menggunakan variabel NM dan PM, serta variabel NB, NS, ZE, PS, dan ZE

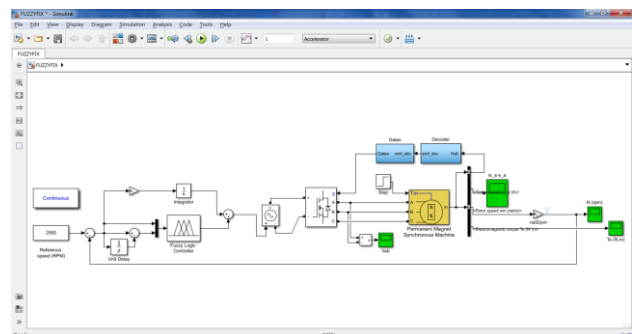
pada output secara berurutan diganti menjadi variabel SL, L, S, C, dan SC.



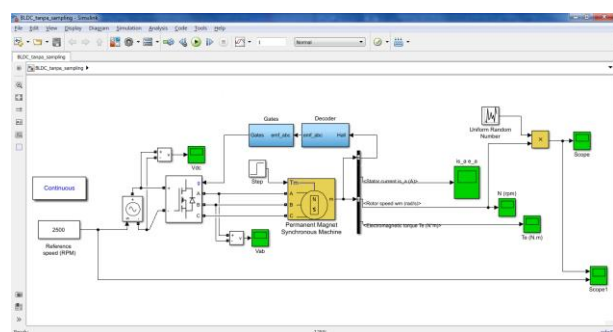
Gambar 7 Grafik fungsi keanggotaan variabel PWM hardware

Perancangan Rangkaian Simulasi Pada Simulink

Perancangan rangkaian simulasi pada simulink MATLAB ini, dilakukan dengan pembuatan rangkaian simulasi sistem pengendalian kecepatan motor brushless DC dengan menghubungkan beberapa parameter blok yang ada pada library simulink menjadi sebuah rangkaian sistem pengendalian kecepatan motor BLDC. Untuk tampilan dari rangkaian sistem pengendalian menggunakan FLC dan tanpa FLC ditunjukkan pada Gambar 8 dan Gambar 9.



Gambar 8 Rangkaian Simulasi Motor BLDC dengan FLC



Gambar 9 Rangkaian Simulasi Motor BLDC Tanpa FLC

Perancangan Program Pengambilan Keputusan Berdasarkan Kontrol Logika Fuzzy

Pada perancangan program pengambil keputusan berdasarkan kontrol Logika Fuzzy dilakukan menggunakan software arduino IDE. Untuk program *Fuzzy Logic Control* yang digunakan dapat dilihat bahwa program yang telah dibuat terdiri dari beberapa proses dijelaskan sebagai berikut.

Inisialisasi

Inisialisasi merupakan sebuah proses yang digunakan untuk melakukan konfigurasi awal terhadap kebutuhan program yang akan dijalankan. Proses inisialisasi berisi pendeklarasian variabel, konfigurasi pin hardware, dan pemanggilan library yang digunakan untuk menjalankan tugas tertentu, seperti pada Gambar 10.

```
#include <Fuzzy.h>

float revolutions = 0;
int rpm = 0;
int pinMotor = 9;
int kecepatan = 0;
long startTime = 0;
long elapsedTime;
int outputPWM;
```

Gambar 10 Program Inisialisasi Kontrol Logika Fuzzy

Fuzzifikasi Data Error, Delta Error, serta data Output PWM

```
Fuzzy *fuzzy = new Fuzzy();

//FuzzyInput error
FuzzySet *eNB = new FuzzySet(-2640, -1759, -1759, -880.1);
FuzzySet *eNS = new FuzzySet(-1759, -880.1, -880.1, 0);
FuzzySet *eZ = new FuzzySet(-880.1, 0, 0, 880.3);
FuzzySet *ePS = new FuzzySet(0, 880.3, 880.3, 1759);
FuzzySet *ePB = new FuzzySet(880.3, 1759, 1759, 2640);

//FuzzyInput derror
FuzzySet *deNB = new FuzzySet(-2640, -1759, -1759, -880.1);
FuzzySet *deNS = new FuzzySet(-1759, -880.1, -880.1, 0);
```

Gambar 11. Program Fuzzifikasi Data Input Error, Delta Error dan Output PWM

Fuzzifikasi data *input error*, *delta error*, serta data output PWM merupakan proses yang digunakan untuk mengubah data yang terbaca oleh pembacaan kecepatan motor oleh modul hall sensor menjadi data fuzzy. Berikut ini merupakan potongan sketch

yang digunakan untuk fuzzifikasi data *input error* dan *delta error*, serta output PWM.

Setelah proses fuzzifikasi supaya proses fuzzifikasi lebih sempurna dilakukan penambahan sketch program sebelum ke proses kaidah aturan (*rule base*) serta proses ini bertujuan untuk pemanggilan fungsi pada proses defuzzifikasi nantinya.

```
//FuzzyInput error
FuzzyInput *error = new FuzzyInput (1);

error->addFuzzySet (eNB);
error->addFuzzySet (eNS);
error->addFuzzySet (eZ);
error->addFuzzySet (ePS);
error->addFuzzySet (ePB);
fuzzy->addFuzzyInput (error);

//FuzzyInput derror
FuzzyInput *derror = new FuzzyInput (2);

derror->addFuzzySet (deNB);
derror->addFuzzySet (deNS);
derror->addFuzzySet (deZ);
derror->addFuzzySet (dePS);
derror->addFuzzySet (dePB);
fuzzy->addFuzzyInput (derror);
```

Gambar 12 Program pemanggilan fungsi proses defuzzifikasi

Kaidah aturan (*rule base*)

Kaidah aturan (*rule base*) merupakan sekumpulan aturan yang dijadikan sebagai dasar pengambilan keputusan oleh sistem fuzzy. Kaidah aturan yang digunakan pada penelitian ini menggunakan sebanyak 25 aturan. Berikut ini merupakan contoh potongan sketch program kaidah yang telah ditulis dalam bahasa C Arduino.

```
//-----RULE 1=IF eNB & deNB THEN outSL1-----
FuzzyRuleAntecedent *eNB_deNB = new FuzzyRuleAntecedent ();
eNB_deNB->joinWithAND (eNB, deNB);

FuzzyRuleConsequent *out_outSL1 = new FuzzyRuleConsequent ();
out_outSL1->addOutput (outSL1);

FuzzyRule *fuzzyRule1 = new FuzzyRule (1, eNB_deNB, out_outSL1);
fuzzy->addFuzzyRule (fuzzyRule1);

//-----RULE 2=IF eNB & deNS THEN outSL2-----
FuzzyRuleAntecedent *eNB_deNS = new FuzzyRuleAntecedent ();
eNB_deNS->joinWithAND (eNB, deNS);
```

Gambar 13. Program Kaidah Aturan (Rule Base)

Defuzzifikasi Data Output PWM Fuzzy

Defuzzifikasi merupakan proses yang digunakan untuk mengubah nilai output fuzzy yang diperoleh berdasarkan kaidah aturan yang digunakan menjadi nilai tegas (*crisp*). Berikut ini merupakan potongan sketch program yang digunakan untuk proses defuzzifikasi data output fuzzy.

```
fuzzy->setInput(1, error);
fuzzy->setInput(2, derror);
fuzzy->fuzzify();
float output = fuzzy->defuzzify(1);
```

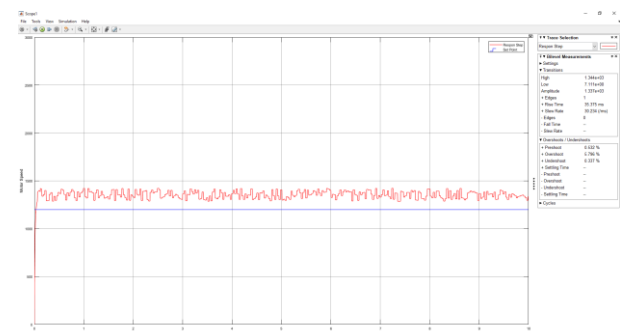
Gambar 14. Program Defuzzifikasi Output PWM Fuzzy

Program di atas merupakan program untuk pengambilan keputusan berdasarkan kontrol logika fuzzy, program pengambilan keputusan dari kaidah aturan logika fuzzy yang memiliki input yaitu nilai error dan delta error dan *feedback* dari modul hall sensor. Dari kedua input tersebut dilakukan kalkulasi dengan kaidah aturan dan dilanjutkan ke proses defuzzifikasi menggunakan metode Center of Area (CoA) yang sudah diatur pada program, kemudian output yang dihasilkan dari kalkulasi tersebut menjadi input bagi driver motor untuk mengatur kecepatan pada motor.

Hasil Simulasi Pada Simulink MATLAB

1) Pengujian Set Point 1200 RPM

Untuk rangkaian pengujian set point 1200 rpm tanpa menggunakan Fuzzy Logic Control (FLC) sebagai berikut.



Gambar 15 Respon Sistem Tanpa FLC dengan sampling time 0.03s Set Point 1200rpm

Dari Gambar 10 terlihat bahwa pada output dari respon sistem ketika diberikan *setpoint* 1200 rpm menunjukkan kecepatan motor yang stabil pada 1400 rpm bukan pada 1200 rpm sesuai *setpoint* yang ditentukan. Hasil respon sistem tersebut dapat dikatakan cukup baik karena *error* yang dihasilkan masih dalam batas toleransi. Pada percobaan ini menghasilkan respon sistem dengan *rise time* 28,955

ms, *overshoot* 1,531%, *time delay* 9,548 ms, *peak time* 85,933 ms dan *settling time* 136,856 ms. Sedangkan untuk rangkaian yang menggunakan FLC adalah sebagai berikut.



Gambar 16 Respon Sistem menggunakan FLC dengan sampling time 0.03s Set Point 1200rpm

Dari Gambar 16 terdapat penambahan kontroler fuzzy dan gain yang dihubungkan pada output rotor speed pada rangkaian sistem. Pengujian dilakukan selama 10 detik dengan *sampling time* 0,03 detik. Respon sistem ketika diberikan *setpoint* 1200 rpm menunjukkan kecepatan motor yang stabil pada 1200 rpm sesuai *setpoint* yang ditentukan. Pada percobaan ini menghasilkan respon sistem dengan *rise time* 15,385 ms, *overshoot* 0,471%, *time delay* 7,339 ms, *peak time* 32,779 ms dan *settling time* 19,68 ms.

Dari kedua hasil pengujian tersebut dilakukan perbandingan untuk parameter respon yang dihasilkan. Hasil perbandingannya ditunjukkan pada Tabel 2.

Tabel 1 Parameter Respon Perbandingan Hasil pada Set Point 1200 rpm dengan dan tanpa FLC

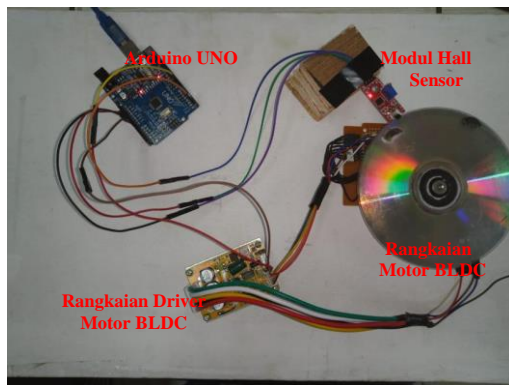
Parameter Respon	Tanpa FLC	Dengan FLC
Rise Time (t_r)	28,955 ms	15,385 ms
Waktu Tunda (t_d)	9,548 ms	7,339 ms
Waktu Tunak (t_s)	136,856 ms	19,68 ms
Maximum Overshoot	1,531 %	0,471 %
Waktu Puncak (t_p)	85,933 ms	32,779 ms

Berdasarkan Tabel 2 dapat disimpulkan bahwa pada *setpoint* 1200 rpm hasil pengujian menggunakan Fuzzy Logic Control mendapatkan respon sistem yang lebih baik dibandingkan dengan tanpa menggunakan kontroler. Ketika menggunakan FLC semua parameter yang dihasilkan lebih cepat untuk mencapai *steady state*. Berbeda dengan tanpa menggunakan FLC parameter yang dihasilkan lebih lama dari yang menggunakan kontroler dan *overshoot* yang dihasilkan sangat tinggi dikarenakan

output yang dihasilkan ketika tidak menggunakan kontroler yaitu menghasilkan kecepatan 1400rpm sedangkan set point yang diujikan adalah 1200rpm. Oleh karena itu, dapat disimpulkan bahwa pengujian menggunakan *Fuzzy Logic Control* lebih baik dibandingkan dengan yang tidak menggunakan kontroler.

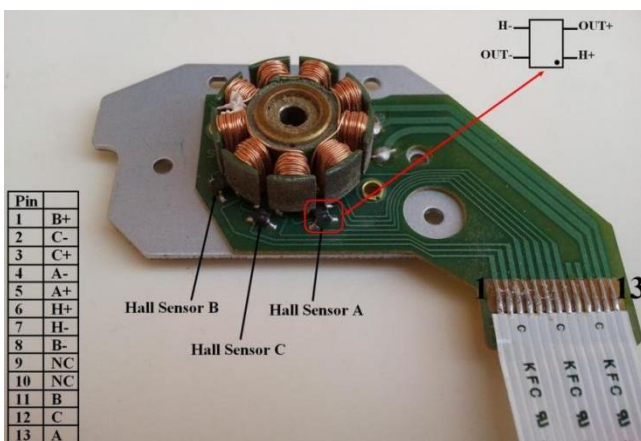
Perancangan Hardware

Pada perancangan hardware ini terdiri dari perancangan masing-masing komponen atau bagian-bagian yang digunakan untuk menyusun atau membentuk keseluruhan sistem. Bagian-bagian yang dirancang yaitu meliputi perancangan rangkaian input/output motor, pembacaan hall effect sensor, driver motor, dan disambungkan juga arduino UNO serta baterai 12V. Untuk tampilan dari rangkaian hardware ditunjukkan pada Gambar 17.



Gambar 17 Rangkaian Hardware

Perancangan Rangkaian input/output Motor BLDC



Gambar 18 Pin Pada Motor BLDC

Perancangan hardware yang pertama yaitu perancangan rangkaian input/output pada motor BLDC. Pada perancangan ini dibuat terminal I/O ke motor BLDC untuk memudahkan pengecekan yang

terhubung pada pin header female. Untuk tampilan dari rangkaian input/output motor ditunjukkan pada Gambar 18 dan Gambar 19.

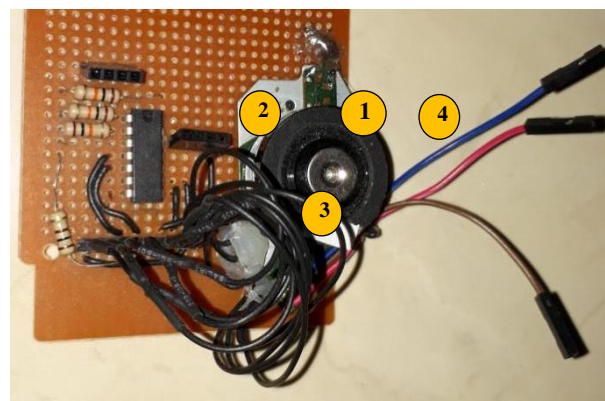


Gambar 19 Rangkaian I/O pada Motor

Perancangan Rangkaian Pembacaan Hall Effect Sensor

Perancangan selanjutnya yaitu perancangan rangkaian pembacaan halleffect sensor. Pada rangkaian ini bertujuan untuk mengolah hasil pembacaan hall sensor yang diterima dari rangkain motor BLDC, kemudian hasil dari rangkaian pembaca hall sensor ini akan dikirimkan ke rangkaian driver motor untuk memberi informasi belitan mana yang harus diberi tegangan agar motor dapat berputar dengan optimal.

Rangkaian ini terdiri dari ada IC LM339, resistor, jumper dan konektor PCB. Semua peralatan tersebut dihubungkan di papan PCB. Untuk tampilan dari rangkaian pembacaan hall effect sensor ditunjukkan pada Gambar 20.



Gambar 20 Rangkaian Pembacaan Hall Effect Sensor

Terdapat fungsi dari tiap-tiap komponen penyusun rangkaian adalah sebagai berikut.

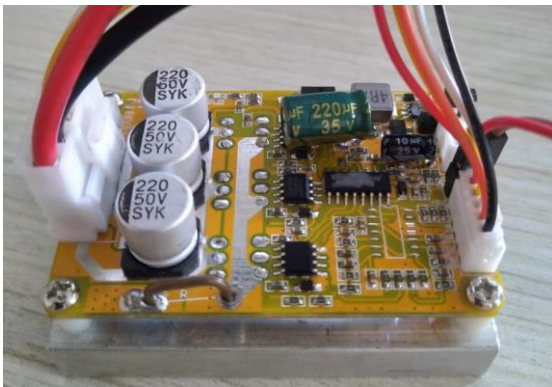
1. IC LM339
IC LM339 sebagai komparator (pembanding tegangan) yang fungsinya membandingkan

input yang ada di *input* + dan *input* - dari motor serta hasilnya perbandingannya akan di tampilkan pada *output*.

2. Resistor
Resistor di sini untuk membatasi atau mengatur arus dalam suatu rangkaian yang akan masuk ke mikrokontroler.
3. Jumper
Jumper berfungsi untuk menghubungkan komponen satu dengan yang lainnya.
4. Pin Header
Pin header yang dipakai yaitu pin header female yang digunakan sebagai penerima *input* dari hasil pembacaan *hall sensor* dari motor, menghubungkan rangkaian *driver* motor.

Perancangan Rangkaian Driver Motor

Pada driver motor BLDC ini terdiri dari sepuluh port input dan lima port output, akan tetapi port input yang digunakan hanya delapan port input yaitu 2 port VCC, 2 port ground, hall A, hall B, hall C, PWM. Dan untuk port output yang digunakan yaitu port VCC, port ground, port MA, port MB, dan port MC. Untuk tampilan dari driver motor BLDC ditunjukkan pada Gambar 22.



Gambar 22 Rangkaian Driver Motor BLDC

Pengujian Rangkaian Hardware

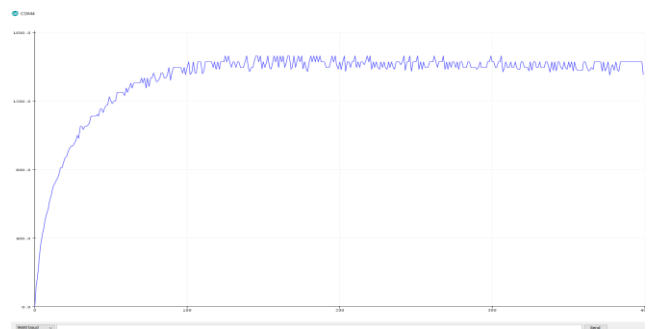
Pada pengujian rangkaian *hardware* dilakukan dengan menggunakan menggunakan kontroler logika fuzzy dan tanpa menggunakan kontroler fuzzy pada set point yang sama seperti pada pengujian menggunakan Simulink MATLAB yaitu 1200 rpm, 1600 rpm, 2000 rpm, dan 2500 rpm. Program pengendalian logika fuzzy yang telah dirancang di-upload ke arduino UNO untuk menjalankan rangkaian ini.

Rangkaian yang sudah dirancang pada tahap perancangan diuji untuk mengetahui output yang

dihasilkan dari motor, apakah output dari motor sudah mendekati set point yang ditentukan dan menghasilkan respon sistem yang bagus atau belum. Untuk pengukuran respon step pada pengujian rangkaian hardware ini dilakukan menggunakan serial plotter bawaan dari Arduino IDE sendiri.

Pengujian Pada Set Point 1200 RPM

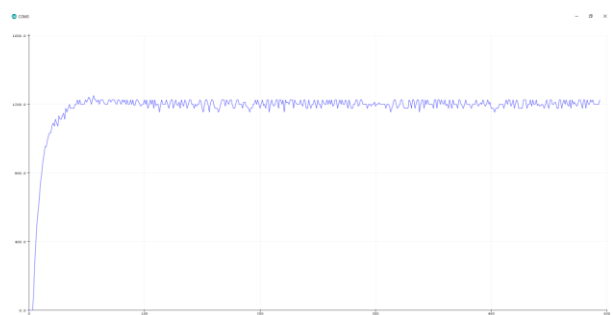
Pada pengujian set point 1200rpm ini dilakukan 2 pengujian yaitu pengujian menggunakan Fuzzy Logic Control (FLC) dan tidak menggunakan FLC. Untuk pengujian menggunakan FLC menghasilkan output kecepatan dan *rise time* yang cepat, sedangkan ketika tidak menggunakan kontroler *rise time* yang dihasilkan lama serta nilai error antara set point dan pembacaan kecepatan lebih besar. Hasil dari respon sistem dan output kecepatan yang tidak menggunakan FLC ditunjukkan pada Gambar 23 dan Gambar 24.



Gambar 23 Respon Tanpa FLC Set Point 1200rpm

Pada Gambar 17 terlihat bahwa respon sistem yang dihasilkan pada pengujian tanpa menggunakan FLC mendapatkan hasil yang kurang tepat dari set point yang telah ditentukan karena pada gambar 24 *set point* yang dihasilkan yaitu 1.341 rpm yang berarti nilai *error* antara set point dan pembacaan kecepatan lebih besar, serta *rise time* 100 ms.

Kemudian untuk hasil respon sistem dan output kecepatan yang menggunakan kontrol FLC ditunjukkan pada Gambar 24.



Gambar 24 Respon Menggunakan FLC Set Point 1200rpm

Pada Gambar 24 terlihat bahwa output kecepatan yang dihasilkan berada pada set point yang ditentukan yaitu 1212 rpm, dan juga respon sistem yang dihasilkan cukup baik dengan menghasilkan rise time dikisaran 8 ms.

Tabel 3 menunjukkan perbandingan respon kecepatan tanpa dan dengan FLC pada set poin 1200 rpm.

Tabel 3. Respon kecepatan dengan dan tanpa FLC set pont 1200 rpm

Tanpa FLC	Dengan FLC
1355	1188
1395	1188
1395	1200
1428	1208
1355	1200
1351	1194
1343	1195
1395	1200
1363	1200
1395	1224
1379	1208
1379	1206
1341	1212

Jika dibandingkan dengan keluaran tanpa menggunakan kontroler, *rise time* saat menggunakan FLC pada percobaan *set point* 1200 rpm ini lebih baik saat menggunakan kontroler FLC dan nilai error kecepatan yang dihasilkan sangat kecil, meski belum sepenuhnya stabil namun tetap berada pada *set point*. Oleh karena itu dapat disimpulkan bahwa hasil keluaran dari motor, baik itu *output* kecepatan maupun respon sistem yang dihasilkan menggunakan FLC lebih baik dibandingkan dengan yang tidak menggunakan FLC.

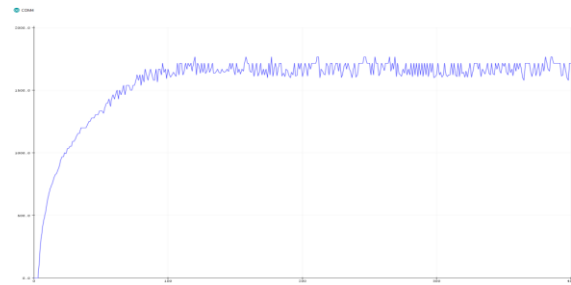
Pengujian Pada Set Point 1600 RPM

Pada pengujian set point 1600rpm ini dilakukan 2 pengujian yaitu pengujian menggunakan Fuzzy Logic Control (FLC) dan tidak menggunakan FLC. Untuk pengujian menggunakan FLC menghasilkan output kecepatan dan rise time yang cepat, sedangkan ketika tidak menggunakan kontroler rise time yang dihasilkan lama serta nilai error antara set point dan pembacaan kecepatan lebih besar. Hasil dari respon sistem dan output kecepatan yang tidak menggunakan FLC ditunjukkan pada gambar 19.

Pada Gambar 19 terlihat respon sistem tanpa menggunakan FLC mendapatkan hasil yang kurang tepat dari set point yaitu 1.760 rpm yang berarti nilai

error antara set point dan pembacaan kecepatan lebih besar, serta rise time yang dihasilkan lama.

Respon sistem dan output kecepatan yang menggunakan kontrol FLC ditunjukkan pada Gambar 25.



Gambar 25 Respon Sistem Tanpa FLC Set Point 1600rpm



Gambar 26 Respon Sistem Menggunakan FLC Set Point 1600rpm

Tabel 4. Respon kecepatan dengan dan tanpa FLC set pont 1600 rpm

Tanpa FLC	Dengan FLC
1605	1603
1621	1531
1714	1618
1714	1620
1714	1678
1764	1620
1714	1558
1714	1600
1714	1618
1714	1680
1714	1632
1714	1589
1760	1620

Pada Gambar 26 terlihat bahwa output kecepatan yang dihasilkan berada pada set point yang ditentukan yaitu 1600 rpm, dan juga respon sistem yang dihasilkan cukup baik dengan menghasilkan rise time di kisaran 10 ms, dan nilai kecepatan yang

dihasilkan berada pada set point, meski belum sepenuhnya stabil namun tetap berada pada *set point*. Tabel 4 menunjukkan perbandingan respon kecepatan tanpa dan dengan FLC pada set poin 1600 rpm.

Jika dibandingkan dengan keluaran tanpa menggunakan kontroler, rise time saat menggunakan FLC pada percobaan set point 1600 rpm ini lebih baik saat menggunakan kontroler FLC dan nilai error kecepatan yang dihasilkan sangat kecil, meski belum sepenuhnya stabil namun tetap berada pada set point. Oleh karena itu dapat disimpulkan bahwa hasil keluaran dari motor, baik itu output kecepatan maupun respon sistem yang dihasilkan menggunakan FLC lebih baik dibandingkan dengan yang tidak menggunakan FLC.

IV. PENUTUP

A. Kesimpulan

Berdasarkan penelitian yang telah dilakukan ditemukan bahwa pada pengujian simulasi menggunakan *software*, pada *set point* 1200 rpm *output* kecepatan tanpa FLC berada di kisaran 1400 rpm, sedangkan ketika menggubakan FLC berada dikisaran 1200 rpm. *Rise time* yang dihasilkan ketika tanpa FLC yaitu 28,955 ms dan *overshoot* yang dihasilkan 1,531 %, sedangkan ketika menggunakan FLC menghasilkan *rise time* 15,385 ms dan *overshoot* yang dihasilkan 0,471%. Pada pengujian hardware ketika tanpa menggunakan FLC pada *set point* 1200 rpm *output* kecepatan yang dihasilkan yaitu berada dikisaran 1341 rpm, sedangkan ketika menggunakan FLC menghasilkan *output* kecepatan berada dikisaran 1212 rpm. *Rise time* yang dihasilkan ketika tanpa menggunakan FLC adalah 100 ms dan ketika menggunakan FLC adalah 8ms. Jadi dapat disimpulkan, sistem kendali kecepatan motor saat menggunakan FLC menghasilkan *output* kecepatan dan respon sistem yang lebih baik dibandingkan dengan yang tidak menggunakan FLC.

DAFTAR PUSTAKA

- [1] Rencana, H., Sakti, P., & Yuniarto, M. N. 2014. Perancangan dan Uji Performa Axial Flux Permanent Magnet Coreless Brushless Direct Current (DC) Motor, 1(1), 1–7.
- [2] Indradewa, E. (2013). Brushless Motor DC. Jurnal Elektromagnetika.
- [3] Liang, D., Li, J., & Qu, R. (2017). Sensorless Control of Permanent Magnet Synchronous Machine Based on Second-Order Sliding-Mode Observer With Online Resistance Estimation. IEEE Transactions on Industry Applications, 53(4), 3672–3682. <https://doi.org/10.1109/TIA.2017.2690218>
- [4] Topal, M., Iskender, I., & Genc, N. (2019). Sensorless speed control of a BLDC motor using improved sliding mode observer technique. International Journal on Technical and Physical Problems of Engineering, 11(1), 1–9.
- [5] Chang, P. I. T., Lin, X. Y., & Yu, I. J. (2019). Sensorless BLDC motor sliding mode controller design for interference recovery. 2019 6th International Conference on Control, Decision and Information Technologies, CoDIT 2019, 1780–1785. <https://doi.org/10.1109/CoDIT.2019.8820383>
- [6] C. Hermanu, B. Apribowo, M. Ahmad, H. Maghfiroh : “Fuzzy Logic Controller and Its Application in Brushless DC Motor (BLDC) in Electric Vehicle - A Review”, 2021, Journal of Electrical, Electronic, Information, and Communication Technology (JEEICT) , 3(1), Pages 35-43.
- [7] Z. Wang, H. Jin, J. Guo, J. Su, M. Wang, 2017, “Research on Control System of Three - phase Brushless DC Motor for Electric Vehicle”. IOP Conf. Series: Materials Science and Engineering
- [8] D.E. Beladjine, D. Boudana, A. Moualdia, 2021 “A Comparative Study of BLDC Motor Speed Control Using PI and ANN Regulator”. 18th IEEE International Multi-Conference on Systems, Signals and Devices (SSD 2021), Monastir, Tunisia, March 22-25
- [9] H. Lu, L. Zhang and W. Qu, 2008 “A New Torque Control Method for Torque Ripple Minimization of BLDC Motors With Un-Ideal Back EMF,” in IEEE Transactions on Power Electronics, Vol. 23, No. 2, pp. 950-958.
- [10] K. Tabarraee, J. Iyer, S. Chiniforoosh and J. Jatskevich : “Comparison of brushless DC motors with trapezoidal and sinusoidal back-EMF”, 2011 24th Canadian Conference on Electrical and Computer Engineering (CCECE), Niagara Falls, ON (2011)
- [11] Kshirsagar, P., Krishnan, R., 2012 ‘High-efficiency current excitation strategy for variable-speed nonsinusoidal’, IEEE Trans. Ind. 48, (6), pp.1875–1889
- [12] Stefan Leitner, 2019 “Cogging Torque Minimization and Performance of the Sub-Fractional HP BLDC Claw-Pole Motor”, IEEE Transactions on Industry Applications, Vol. 55, No. 5, September/October, pp. 4653-4664
- [13] Zicheng Li, Qingyao Kong, Shanmei Cheng, Jiang Liu, 2020, “Torque ripple suppression of brushless DC motor drives using an alternating two-phase and three-phase conduction mode”, IET Power Electronics, Vol. 13 Iss. 8, pp. 1622-1629.

- [14] Xia, Chang-liang, 2012, “Permanent Magnet Brushless DC Motor Drives and Controls”. John Wiley and Sons. pp. 18–19
- [15] B. N. Kommula and V. R. Kota, 2015, “Performance evaluation of hybrid fuzzy pi speed controller for brushless dc motor for electric vehicle application,” in Power, Control, Communication and Computational Technologies for Sustainable Growth (PCCCTSG), 2015 Conference on. IEEE, pp. 266–270.
- [16] Y. Zhou, D. Zhang, X. Chen, and Q. Lin, 2016 “Sensorless direct torque control for saliency permanent magnet brushless dc motors,” IEEE Trans. Energy Convers., vol. 31, no. 2, pp. 446–454, Jun.
- [17] Jaya A, Wahjono E, Rusli M.R, Purwanto E, Murdianto F.D, FauziahM.B, 2017, “Design and Simulation of Sensorless BLDC Motor Drive Using Flux Linkage Increment Based on the line-to-line BEMF for Electric Vehicles” (IES-ETA) International Electronics Symposium on Engineering Technology and Applications, Surabaya.
- [18] S. Gowtham, I. Keerthana, M. Balaji, 2019, ”Characterization and Classification of Hall Sensor Faults using S-Transform Analysis on BLDC Motor Drive”, IEEE 1st International Conference on Energy, Systems and Information Processing (ICESIP)