



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**TUGAS AKHIR - TE 141599**

**PERANCANGAN PENGGERAK ELEKTRONIK DAN  
KONTROLER PID-FUZZY PADA PENGATURAN  
KECEPATAN *BRUSHLESS* DC (BLDC) MOTOR**

Suwondo Saputra  
NRP 2213105003

Dosen Pembimbing  
Ir. Ali Fatoni, M.T.

JURUSAN TEKNIK ELEKTRO  
Fakultas Teknologi Industri  
Institut Teknologi Sepuluh Nopember  
Surabaya 2015



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**FINAL PROJECT - TE141599**

**DESIGN ELECTRONIC DRIVE AND FUZZY PID  
CONTROLLER IN SPEED CONTROL OF  
BRUSHLESS DC (BLDC) MOTOR**

Suwondo Saputra  
NRP 2213105003

***Supervisor***  
Ir. Ali Fatoni, M.T.

DEPARTMENT OF ELECTRICAL ENGINEERING  
Faculty of Industrial Technology  
SepuluhNopember Instituteof Technology  
Surabaya 2015

**PERANCANGAN PENGGERAK ELEKTRONIK DAN  
KONTROLER PID-FUZZY PADA PENGATURAN KECEPATAN  
BRUSHLESS DC (BLDC) MOTOR**

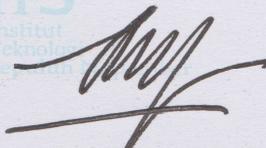
**TUGAS AKHIR**

**Diajukan Guna Memenuhi Sebagian Persyaratan  
Untuk Memperoleh Gelar Sarjana Teknik  
Pada**

**Bidang Studi Teknik Sistem Pengaturan  
Jurusan Teknik Elektro  
Institut Teknologi Sepuluh Nopember**

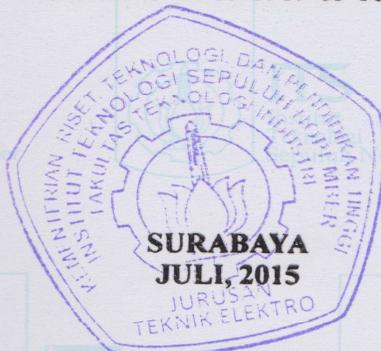
**Menyetujui :**

**Dosen Pembimbing**



**Ir. Ali Fatoni, M.T.**

**NIP. 1962 06 03 1989 03 1002**



# PERANCANGAN PENGGERAK ELEKTRONIK DAN KONTROLER PID-*FUZZY* PADA PENGATURAN KECEPATAN *BRUSHLESS* DC (BLDC) MOTOR

**Nama** : Suwondo Saputra  
**Pembimbing** : Ir. Ali Fatoni, M.T.

## ABSTRAK

Seiring perkembangan jaman, permintaan akan sistem penggerak listrik yang efisien, torsi tinggi, dan biaya perawatan yang murah semakin meningkat. Motor DC dan motor induksi belum mampu memenuhi kebutuhan tersebut. BLDCM mampu memenuhi kebutuhan tersebut. BLDCM yang merupakan motor sinkron 3 fasa namun tetap membutuhkan tegangan DC sebagai sumbernya, dimana tegangan DC dikonversi menjadi tegangan AC 3 fasa menggunakan *inverter*.

Disamping keuntungan dari BLDCM ternyata metode pengaturan kecepatan masih sulit. Menurunnya kecepatan motor yang diakibatkan perubahan beban sudah menjadi hal yang umum. Penggunaan kontroler PID-*Fuzzy* menjadi salah satu metode yang bisa memecahkan masalah tersebut. Algoritma PID yang sederhana ditambah logika *fuzzy* dalam hal penalaan parameter PID dapat menyesuaikan dengan berbagai beban.

Hasil implemmentasi kontroler PID-*Fuzzy* dapat mengembalikan keluaran plant kembali ke kondisi *steady state* bila terjadi perubahan beban. Hasil respon mendekati dengan *set point* 645 rpm, *time constant* 0,520 detik, *rise time* 1,490 detik, *settling time* 1,670 detik, *delay time* 0,370 detik, *error steady state* 0,099% dan *overshoot* sebesar 5,426% saat perubahan beban.

**Kata kunci** : BLDCM, *Inverter*, Logika *Fuzzy*, PID, PID-*Fuzzy*

# PERANCANGAN PENGGERAK ELEKTRONIK DAN KONTROLER PID-*FUZZY* PADA PENGATURAN KECEPATAN *BRUSHLESS* DC (BLDC) MOTOR

**Nama** : Suwondo Saputra  
**Pembimbing** : Ir. Ali Fatoni, M.T.

## ABSTRAK

Seiring perkembangan jaman, permintaan akan sistem penggerak listrik yang efisien, torsi tinggi, dan biaya perawatan yang murah semakin meningkat. Motor DC dan motor induksi belum mampu memenuhi kebutuhan tersebut. BLDCM mampu memenuhi kebutuhan tersebut. BLDCM yang merupakan motor sinkron 3 fasa namun tetap membutuhkan tegangan DC sebagai sumbernya, dimana tegangan DC dikonversi menjadi tegangan AC 3 fasa menggunakan *inverter*.

Disamping keuntungan dari BLDCM ternyata metode pengaturan kecepatan masih sulit. Menurunnya kecepatan motor yang diakibatkan perubahan beban sudah menjadi hal yang umum. Penggunaan kontroler PID-*Fuzzy* menjadi salah satu metode yang bisa memecahkan masalah tersebut. Algoritma PID yang sederhana ditambah logika *fuzzy* dalam hal penalaan parameter PID dapat menyesuaikan dengan berbagai beban.

Hasil implemmentasi kontroler PID-*Fuzzy* dapat mengembalikan keluaran plant kembali ke kondisi *steady state* bila terjadi perubahan beban. Hasil respon mendekati dengan *set point* 645 rpm, *time constant* 0,520 detik, *rise time* 1,490 detik, *settling time* 1,670 detik, *delay time* 0,370 detik, *error steady state* 0,099% dan *overshoot* sebesar 5,426% saat perubahan beban.

**Kata kunci** : BLDCM, *Inverter*, Logika *Fuzzy*, PID, PID-*Fuzzy*



# **DESIGN ELECTRONIC DRIVE AND FUZZY PID CONTROLLER IN SPEED CONTROL OF BRUSHLESS DC (BLDC) MOTOR**

**Name** : Suwondo Saputra  
**Supervisor** : Ir. Ali Fatoni, M.T.

## **ABSTRACT**

*Along with industry development, the need of electric motors that have high efficiency, high torque, and inexpensive treatment is increasing. DC motor and induction motor are not able to meet those need. BLDCM is able to meet those need. BLDCM is a 3 phase synchronous motor but still requires DC voltage as the source where it was converted into 3 phase AC voltage using inverter.*

*Beside the benefit of BLDCM evidently speed control method still difficult. Decreasing speed of motor caused load change has become a common thing. Using Fuzzy PID controller become a method that can solve the problem. A simple PID algorithm and fuzzy logic to tune PID parameters can be adapted with load changes.*

*Result of implementation of Fuzzy PID controller can be back output of plant to steady state condition if load changes was happened. Result of response approach to 645 rpm of set point, 0.520 seconds of time constant, 1.490 seconds of rise time, 1.670 seconds of settling time, 0.370 of delay time, 0.099% of error steady state, and 5.426% of overshoot when load is happened.*

**Keywords** : BLDCM, Fuzzy Logic, Fuzzy PID, Inverter, PID



## KATA PENGANTAR

Segala puji kehadiran Tuhan Yang Maha Esa yang telah melimpahkan rahmat dan hidayah-Nya kepada kami dalam usaha dan kerja kami sehingga dapat menyelesaikan Tugas Akhir dengan judul :

### **PERANCANGAN PENGGERAK ELEKTRONIK DAN KONTROLER PID-*FUZZY* PADA PENGATURAN KECEPATAN *BRUSHLESS* DC (BLDC) MOTOR**

Tugas Akhir ini disusun guna memenuhi persyaratan untuk menyelesaikan studi strata 1 di jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember, Surabaya.

Dalam penyusunan laporan Tugas Akhir ini, kami banyak mendapatkan bantuan dan dukungan dari berbagai pihak. Oleh karena itu, penulis dengan tulus ikhlas menyampaikan banyak terima kasih kepada:

1. Bapak Ir. Ali Fatoni, M.T. selaku dosen pembimbing.
2. Ayah, Ibu, Adek dari penulis yang telah membantu baik dari segi materi, semangat maupun motivasinya.
3. Umi Nurhidayati yang telah banyak memberikan *support* dan motivasi.
4. Saudara/i Kurniawan Khoiruddin, Marika Ayu PR., sebagai partner TA.
5. Teman-teman seperjuangan Teknik Sistem Pengaturan.
6. Dan semua pihak yang telah membantu terselesaikannya buku Tugas Akhir ini.

Penulis menyadari dan memohon maaf atas segala kekurangan pada Tugas Akhir ini dikarenakan keterbatasan kemampuan penulis. Penulis mengharapkan berbagai masukan dan kritik dari berbagai pihak agar lebih baik di masa mendatang. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat dalam ilmu pengetahuan demi kemajuan Indonesia menuju negeri yang berteknologi mandiri.

Surabaya, Juni 2015

Penyusun



# DAFTAR ISI

<b>HALAMAN JUDUL</b> .....	<b>i</b>
<b>PERNYATAAN KEASLIAN TUGAS AKHIR</b> .....	<b>v</b>
<b>HALAMAN PENGESAHAN</b> .....	<b>vii</b>
<b>ABSTRAK</b> .....	<b>ix</b>
<b>ABSTRACT</b> .....	<b>xi</b>
<b>KATA PENGANTAR</b> .....	<b>xiii</b>
<b>DAFTAR ISI</b> .....	<b>xv</b>
<b>DAFTAR GAMBAR</b> .....	<b>xix</b>
<b>DAFTAR TABEL</b> .....	<b>xxiii</b>
<b>BAB I PENDAHULUAN</b> .....	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Perumusan Masalah .....	1
1.3 Batasan Masalah .....	2
1.4 Tujuan Penelitian .....	2
1.5 Metodologi .....	2
1.6 Sistematika Penulisan .....	3
1.7 Relevansi .....	4
<b>BAB II TEORI PENUNJANG</b> .....	<b>5</b>
2.1 <i>Brushless DC Motor</i> (BLDCM) .....	5
2.1.1 Penjelasan BLDCM .....	5
2.1.2 Prinsip Kerja BLDCM .....	6
2.2 Penggerak Elektronik .....	8
2.3 Pengendalian BLDCM .....	9
2.3.1 Metode Six-Step .....	9
2.3.2 Metode Sinyal PWM .....	10
2.4 Rem Magnetik .....	11
2.5 Identifikasi Sistem .....	11
2.5.1 Identifikasi Statis .....	11
2.5.1.1 Identifikasi Statis dengan Metode Viteckova Orde 1 .....	13
2.5.1.2 Identifikasi Statis dengan Metode Viteckova Orde 2 .....	14
2.5.1.3 Identifikasi Statis dengan Metode Sundaresan & Krishnaswamy .....	14

2.5.1.4	Identifikasi Statis dengan Metode Grafis ...	15
2.6.	Validasi Model RMSE .....	16
2.7	Kontroler PID .....	17
2.7.1	Kontroler Proporsional (P) .....	17
2.7.2	Kontroler <i>Integral</i> (I) .....	18
2.7.3	Kontroler Derivatif (D) .....	18
2.8	Logika <i>Fuzzy</i> .....	19
2.8.1	Himpunan <i>Fuzzy</i> .....	19
2.8.2	Fungsi Keanggotaan <i>Fuzzy</i> .....	19
2.8.3	Operasi Himpunan <i>Fuzzy</i> .....	21
2.8.4	Sistem Kontrol <i>Fuzzy</i> .....	22
2.9	Kontroler PID- <i>Fuzzy</i> .....	23
2.10	Sensor <i>Rotary Encoder</i> .....	24
2.11	Arduino .....	25
<b>BAB III</b>	<b>PERANCANGAN SISTEM .....</b>	<b>27</b>
3.1	Arsitektur Sistem .....	27
3.2	Perancangan Perangkat Keras ( <i>Hardware</i> ) .....	28
3.2.1	Perancangan Mekanik <i>Plant</i> .....	29
3.2.2	Perancangan Sensor <i>Rotary Encoder</i> .....	30
3.2.3	Perancangan Rangkaian Penggerak Elektronik .....	32
3.2.3.1	Rangkaian Pembagi Fasa .....	32
3.2.3.2	Rangkaian Pengolah Sinyal Fasa dan PWM .....	33
3.2.3.3	Rangkaian <i>Optoisolator</i> .....	34
3.2.3.4	Rangkaian <i>Switching</i> .....	35
3.2.3.5	Rangkaian <i>Power Supply</i> .....	35
3.3	Perancangan Perangkat Lunak ( <i>Software</i> ) .....	36
3.3.1	Arduino Sebagai Pembangkit Frekuensi dan PWM	36
3.3.2	Arduino Sebagai Pembaca Data Sensor .....	38
3.3.3	Perancangan Metode Identifikasi .....	38
3.4	Identifikasi <i>Plant</i> .....	39
3.4.1	Metode Identifikasi .....	39
3.4.2	Metode Pembebanan <i>Plant</i> .....	43
3.5	Perancangan Kontroler PID- <i>Fuzzy</i> .....	44
3.5.1	Perancangan Kontroler PID .....	45
3.5.2	Perancangan Mekanisme PID- <i>Fuzzy</i> .....	46
3.5.2.1	Fungsi Keanggotaan Masukan .....	47
3.5.2.2	Fungsi Keanggotaan Keluaran .....	47

3.5.2.3	<i>Rule Base</i> .....	48
3.5.2.4	Defuzifikasi .....	49
3.6	Perancangan Simulasi Kontroler .....	49
3.6.1	Blok Model <i>Plant</i> Beban Nominal .....	50
3.6.2	Blok Model Pembebanan <i>Plant</i> .....	50
3.6.3	Blok <i>Fuzzy Tunner</i> .....	51
<b>BAB IV PENGUJIAN DAN ANALISA</b> .....		<b>53</b>
4.1	Pengujian Arduino .....	53
4.1.1	Analisa Pengujian Pembangkit Sinyal PWM .....	53
4.1.2	Analisa Pengujian Pembangkit Sinyal Fasa .....	54
4.2	Pengujian Rangkaian Penggerak Elektronik .....	55
4.2.1	Analisa Pengujian Keluaran Gerbang AND dan NAND .....	55
4.2.2	Analisa Pengujian Keluaran <i>Optocoupler</i> .....	55
4.3	Analisa Pengujian Simulasi tanpa Kontroler .....	58
4.4	Analisa Pengujian Simulasi dengan Kontroler PID- <i>Fuzzy</i> .	60
4.5	Analisa Implementasi Pengujian Kontroler PID- <i>Fuzzy</i> .....	63
<b>BAB V PENUTUP</b> .....		<b>67</b>
5.1	Kesimpulan .....	67
5.2	Saran .....	67
<b>DAFTAR PUSTAKA</b> .....		<b>69</b>
<b>LAMPIRAN</b> .....		<b>71</b>
<b>RIWAYAT PENULIS</b> .....		<b>77</b>



*--Halaman ini sengaja dikosongkan--*

## DAFTAR TABEL

<b>Tabel 2.1</b>	Urutan Pengaturan Saklar BLDCM .....	9
<b>Tabel 3.1</b>	Spesifikasi BLDCM .....	30
<b>Tabel 3.2</b>	Validasi Model Matematika Beban Minimal.....	40
<b>Tabel 3.3</b>	Validasi Model Matematika Beban Nominal .....	40
<b>Tabel 3.4</b>	Validasi Model Matematika Beban Maksimal .....	41
<b>Tabel 3.5</b>	Model Matematika Beban .....	44
<b>Tabel 3.6</b>	Parameter PID pada Kondisi Pembebanan.....	46
<b>Tabel 4.1</b>	Hasil Pengujian Pembangkit Frekuensi Fasa Arduino	54
<b>Tabel 4.2</b>	Hasil Pengujian Kecepatan BLDCM.....	57
<b>Tabel 4.3</b>	Spesifikasi Respon Tiap Beban .....	60
<b>Tabel 4.4</b>	Spesifikasi Respon Tiap Beban dengan Kontroler PID- <i>Fuzzy</i> .....	62
<b>Tabel 4.5</b>	Spesifikasi Implementasi Respon Tiap Beban.....	66



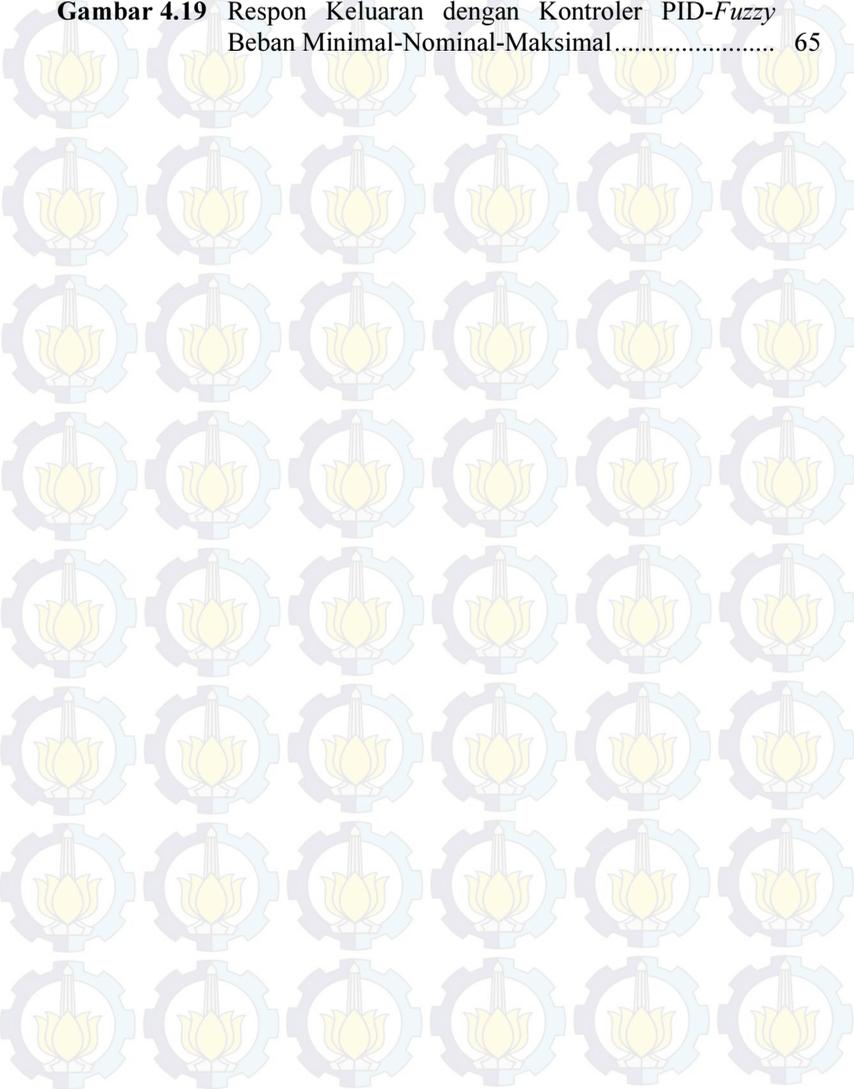
## DAFTAR GAMBAR

<b>Gambar 2.1</b>	Konstruksi BLDCM .....	6
<b>Gambar 2.2</b>	<i>Wiring Diagram</i> BLDCM.....	7
<b>Gambar 2.3</b>	Tegangan Stator BLDCM.....	7
<b>Gambar 2.4</b>	Urutan <i>Switching</i> pada <i>Stator</i> .....	8
<b>Gambar 2.5</b>	Bentuk Gelombang Stator Hasil <i>Switching</i> .....	9
<b>Gambar 2.6</b>	Driver 3 Fasa dengan Modul PWM.....	10
<b>Gambar 2.7</b>	<i>Switching</i> Menggunakan Sinyal PWM.....	10
<b>Gambar 2.8</b>	Karakteristik Respon Orde Satu .....	11
<b>Gambar 2.9</b>	Penentuan Garis Singgung Metode Grafis.....	15
<b>Gambar 2.10</b>	Diagram Blok Kontroler PID.....	17
<b>Gambar 2.11</b>	Fungsi Keanggotaan Segitiga .....	20
<b>Gambar 2.12</b>	Fungsi Keanggotaan Trapesium .....	21
<b>Gambar 2.13</b>	Sistem Kontrol <i>Fuzzy</i> .....	22
<b>Gambar 2.14</b>	Diagram Blok Kontroler PID- <i>Fuzzy</i> .....	24
<b>Gambar 2.15</b>	Keluaran Sinyal dan Susunan dari <i>Rotary Encoder</i> . ..	24
<b>Gambar 2.16</b>	<i>Board</i> Arduino UNO (kiri) dan Arduino Mega2560 (kanan).....	25
<b>Gambar 3.1</b>	Arsitektur Sistem .....	27
<b>Gambar 3.2</b>	Komponen dan Alur Perangkat Keras Sistem.....	29
<b>Gambar 3.3</b>	Konstruksi <i>Plant</i> Secara Keseluruhan .....	30
<b>Gambar 3.4</b>	Skema Rangkaian Sensor <i>Rotary Encoder</i> .....	31
<b>Gambar 3.5</b>	Konstruksi <i>Rotary Encoder</i> dengan <i>Optocoupler</i> ....	31
<b>Gambar 3.6</b>	Bentuk Fisik Rangkaian Penggerak Elektronik .....	32
<b>Gambar 3.7</b>	Skema Rangkaian Pembagi Fasa .....	32
<b>Gambar 3.8</b>	Skema Rangkaian Pengolah Sinyal Fasa dan PWM .....	33
<b>Gambar 3.9</b>	Skema Rangkaian <i>Optoisolator</i> .....	34
<b>Gambar 3.10</b>	Skema Rangkaian <i>Switching</i> .....	35
<b>Gambar 3.11</b>	Skema Rangkaian <i>Power Supply</i> .....	36
<b>Gambar 3.12</b>	<i>Flowchart</i> Pemrogramman Pembangkit PWM dan Frekuensi .....	37
<b>Gambar 3.13</b>	<i>Flowchart</i> Pembacaan Data Sensor Kecepatan .....	38
<b>Gambar 3.14</b>	Blok Diagram Simulink Identifikasi.....	39
<b>Gambar 3.15</b>	Respon Model Matematika Beban Minimal .....	40
<b>Gambar 3.16</b>	Respon Model Matematika Beban Nominal.....	41
<b>Gambar 3.17</b>	Respon Model Matematika Beban Maksimal.....	42
<b>Gambar 3.18</b>	Penarikan Garis Singgung pada Metode Grafis.....	42

<b>Gambar 3.19</b>	Diagram Blok Kontroler PID- <i>Fuzzy</i> .....	44
<b>Gambar 3.20</b>	Blok Diagram Kontroler PID .....	46
<b>Gambar 3.21</b>	Fungsi Keanggotaan <i>Fuzzy</i> untuk <i>Error</i> , <i>Delta Error</i> , dan <i>Integral Error</i> .....	47
<b>Gambar 3.22</b>	Fungsi Keanggotaan $K_p$ .....	47
<b>Gambar 3.23</b>	Fungsi Keanggotaan $K_i$ .....	48
<b>Gambar 3.24</b>	Fungsi Keanggotaan $K_d$ .....	48
<b>Gambar 3.25</b>	Diagram Blok Simulasi PID- <i>Fuzzy</i> pada Simulink .	49
<b>Gambar 3.26</b>	Blok Model Pembebanan <i>Plant</i> .....	50
<b>Gambar 3.27</b>	Diagram Blok <i>Fuzzy Tuner</i> .....	51
<b>Gambar 4.1</b>	Mekanisme Pengujian Sinyal PWM.....	53
<b>Gambar 4.2</b>	Hasil Pengujian Sinyal PWM.....	53
<b>Gambar 4.3</b>	Hasil Pengujian Sinyal Frekuensi Fasa .....	54
<b>Gambar 4.4</b>	Mekanisme Pengujian Sinyal gerbang AND dan NAND .....	55
<b>Gambar 4.5</b>	Sinyal Keluaran AND (kiri) dan NAND (kanan)....	55
<b>Gambar 4.6</b>	Sinyal Keluaran <i>Optocoupler HIGH</i> (kiri) dan <i>LOW</i> (kanan) .....	56
<b>Gambar 4.7</b>	Mekanisme Pengujian Kecepatan BLDCM .....	57
<b>Gambar 4.8</b>	Hasil Pengujian Kecepatan BLDCM.....	58
<b>Gambar 4.9</b>	Hasil Pengujian Tanpa Kontroler untuk Beban Minimal .....	59
<b>Gambar 4.10</b>	Hasil Pengujian Tanpa Kontroler untuk Beban Nominal.....	59
<b>Gambar 4.11</b>	Hasil Pengujian Tanpa Kontroler untuk Beban Maksimal.....	60
<b>Gambar 4.12</b>	Respon Keluaran Beban Minimal dengan Kontroler PID- <i>Fuzzy</i> .....	61
<b>Gambar 4.13</b>	Respon Keluaran Beban Nominal dengan Kontroler PID- <i>Fuzzy</i> .....	61
<b>Gambar 4.14</b>	Respon Keluaran Beban Maksimal dengan Kontroler PID- <i>Fuzzy</i> .....	62
<b>Gambar 4.15</b>	Respon Keluaran dengan Kontroler PID- <i>Fuzzy</i> Beban Berubah .....	63
<b>Gambar 4.16</b>	Respon Keluaran dengan Kontroler PID- <i>Fuzzy</i> Beban Minimal.....	64
<b>Gambar 4.17</b>	Respon Keluaran dengan Kontroler PID- <i>Fuzzy</i> Beban Nominal.....	64

**Gambar 4.18** Respon Keluaran dengan Kontroler PID-*Fuzzy*  
Beban Maksimal..... 65

**Gambar 4.19** Respon Keluaran dengan Kontroler PID-*Fuzzy*  
Beban Minimal-Nominal-Maksimal..... 65





## RIWAYAT PENULIS



Suwondo Saputra lahir di Prabumulih pada tanggal 23 September 1989. Penulis merupakan anak pertama dari pasangan Paino dan Tanzima. Setelah lulus dari SMKN 1 Cimahi pada tahun 2009, penulis melanjutkan studi di Program Studi Diploma III Politeknik Negeri Semarang (Polines), Jurusan Teknik Elektro. Pada tahun 2013 terdaftar sebagai mahasiswa lintas jalur Institut Teknologi Sepuluh Nopember (ITS), Jurusan Teknik Elektro dengan Bidang Studi Teknik Sistem Pengaturan. Selama menjalani studi di Institut Teknologi Sepuluh Nopember (ITS), penulis selalu antusias dalam mengikuti kegiatan perkuliahan. Penulis mengikuti seminar dan ujian Tugas Akhir di Bidang Studi Teknik Sistem Pengaturan, Jurusan Teknik Elektro, ITS Surabaya sebagai salah satu persyaratan untuk memperoleh gelar Sarjana Teknik Elektro. Email penulis adalah [suwondo1989@yahoo.com](mailto:suwondo1989@yahoo.com).



# BAB I

## PENDAHULUAN

Pada bab ini menjelaskan latar belakang, perumusan masalah, dan batasan masalah. Selain itu pada bab ini juga dijelaskan tujuan penelitian, metodologi yang digunakan, sistematika penulisan, dan relevansi.

### 1.1 Latar Belakang

Penggunaan motor listrik didalam kehidupan sehari-hari sudah menjadi kebutuhan yang sangat penting. Berbagai alat banyak penggunaan motor listrik, seperti *air conditioning*, *conveyor*, *vacuum cleaner* dan sebagainya. Motor DC adalah salah satu mesin listrik yang banyak digunakan di dunia industri, akan tetapi penggunaan motor DC menimbulkan masalah yang diakibatkan penggunaan sikat (*brush*). BLDCM merupakan alternatif pengganti motor DC. BLDCM memiliki kelebihan diantaranya adalah efisiensi yang tinggi, memiliki torsi besar, dan perawatan mudah. Penghilangan sikat komutator pada BLDCM dapat mengatasi permasalahan yang berhubungan dengan kontak berupa gesekan antara komutator yang menimbulkan keausan sehingga perlu dilakukan perawatan secara berkala.

Penggantian motor DC oleh BLDCM menjadi sebab dibutuhkannya cara kontrol yang berbeda untuk komutasi fasa dari BLDCM. Rangkaian kontrol pengganti kumutator pada BLDCM ini disebut dengan penggerak elektronik untuk mencatu daya ke kumparan stator untuk memberikan kontrol besarnya arus yang mengalir sehingga putaran rotor bisa dikontrol. Rangkaian kontrol terdiri atas 6 buah MOSFET yang digunakan untuk *switching* tegangan stator.

Kontroler PID (*Proporsional-Integral-Derivatif*) merupakan kontrol yang umum digunakan dalam industri dikarenakan kontrol yang sederhana akan tetapi hanya berlaku untuk satu kondisi, apabila motor diberi beban maka terjadi perubahan pada pemodelannya yang mengakibatkan perlunya *tuning* pada parameter kontrol PID untuk menjaga plant tetap stabil apabila terjadi perubahan beban. Penggunaan logika *fuzzy* dapat melakukan *tuning* parameter kontrol PID secara otomatis, sehingga dapat mempertahankan kecepatan motor dengan berbagai perubahan pada bebannya.

## 1.2 Perumusan Masalah

BLDCM yang merupakan motor 3 fasa akan tetapi membutuhkan tegangan DC sebagai sumbernya. BLDCM yang tidak memiliki sikat (*brush*) menjadi sebab dibutuhkannya rangkaian penggerak elektronik untuk mengubah tegangan DC menjadi tegangan AC 3 fasa agar dapat menggerakkan BLDCM. Tidak adanya sikat sebagai penunjang komutasi pada stator dan menurunnya kecepatan yang diakibatkan perubahan beban membuat pengaturan BLDCM menjadi lebih sulit. Oleh karena itu, pada Tugas Akhir ini dirancang rangkaian penggerak elektronik untuk menggerakkan BLDCM dan kontrol PID-*Fuzzy* pada pengaturan kecepatan BLDCM untuk mempertahankan performansi dari BLDCM tetap *steady state* terhadap nilai *set point* walaupun diberi pembebanan.

## 1.3 Batasan Masalah

Beberapa hal yang perlu dibatasi dalam Tugas Akhir ini agar penelitian yang dilakukan dapat tercapai, yaitu:

1. Efek pembebanan rem magnetik pada *plant* hanya dilihat dari besarnya sudut rem magnetik terhadap piringan rotor BLDCM dan pengaruh secara spesifik tidak diperhitungkan secara detail
2. Sistem kontrol yang dirancang hanya untuk mengatur kecepatan
3. Metode identifikasi dilakukan secara statis menggunakan 4 metode
4. *Range* kerja dari rangkaian penggerak elektronik berkisar antara 10-55 Hz.

## 1.4 Tujuan Penelitian

Pada tugas akhir ini, Penelitian Tugas Akhir ini adalah merancang penggerak elektronik dan kontroler PID-*Fuzzy* untuk mengatur kecepatan BLDCM sesuai dengan nilai *set point* yang diberikan.

## 1.5 Metodologi

Metodologi dalam pelaksanaan penelitian ini dilakukan dengan beberapa tahapan yaitu studi literatur, perancangan dan pembuatan sistem, pengujian sistem, analisa data, dan penulisan buku Tugas Akhir. Tahap studi literatur dilakukan dengan mencari beberapa sumber referensi yang berhubungan dengan penggerak elektronik dan kontrol PID-*Fuzzy*. Sumber referensi yang digunakan dapat berupa buku, jurnal,

*website*, dan sebagainya. Tahap perancangan dan pembuatan sistem dilakukan dengan cara mendesain penggerak elektronik, merancang *plant*, mendesain kontroler, membuat simulasi kontroler serta membuat sistem secara keseluruhan.

Tahap pengujian sistem dilakukan untuk mengetahui kontroler yang dibuat telah sesuai dengan hasil implementasinya. Pengujian ini dilakukan dengan menggunakan *software* Matlab. Tahap analisa data dilakukan berdasarkan hasil simulasi dan pengujian. Tahap ini dilakukan analisa terhadap persentase kesalahan yang terjadi. Hal ini diperlukan untuk mengevaluasi kembali dari hasil desain kontrol. Penulisan buku Tugas Akhir merupakan tahap akhir dalam penelitian. Buku Tugas Akhir ditulis secara intensif bila proses pengujian telah selesai.

## **1.6 Sistematika Penulisan**

Sistematika penulisan yang disusun pada Tugas Akhir ini terdiri dari lima bab, sebagai berikut:

### **BAB I : PENDAHULUAN**

Bab ini menjelaskan tentang latar belakang, perumusan masalah, batasan masalah, tujuan penelitian, sistematika penulisan dan relevansi.

### **BAB II : TEORI PENUNJANG**

Bab ini berisi tentang dasar – dasar pengetahuan dari tiap komponen yang digunakan pada sistem yang akan dirancang seperti: *Brushless DC Motor* (BLDCM), penggerak elektronik, rem magnetik, identifikasi sistem, validasi *Root Mean Square Error* (RMSE), kontroler *PID-Fuzzy*, sensor *rotary encoder*, dan Arduino.

### **BAB III : PERANCANGAN SISTEM**

Bab ini berisi tentang perancangan sistem secara garis besar yang terdiri dari arsitektur sistem, perancangan *hardware*, perancangan *software*, identifikasi BLDCM, dan perancangan dan implementasi kontroler.

### **BAB IV : PENGUJIAN DAN ANALISA**

Bab ini menjelaskan tentang hasil pengujian rangkaian penggerak elektronik, pengujian simulasi dan implementasi sistem secara keseluruhan.

## BAB V : PENUTUP

Bab ini berisi tentang kesimpulan dari penelitian Tugas Akhir ini dan saran – saran mengenai kemungkinan pengembangan penelitian ini.

### 1.7 Relevansi

Hasil yang dicapai dari Tugas Akhir ini diharapkan menjadi salah satu referensi yang dapat digunakan dalam pengaturan kecepatan BLDCM, sehingga dapat membantu dalam melakukan pengaturan BLDCM. Selain itu penelitian ini juga diharapkan dapat menjadi perbandingan dari beberapa metode pengaturan kecepatan BLDCM dimasa sebelumnya maupun yang akan datang.

## BAB II

### TEORI PENUNJANG

BAB II menjelaskan beberapa teori penunjang yang dijadikan sebagai acuan dalam melakukan penelitian. Beberapa hal yang dibahas meliputi *Brushless DC Motor* (BLDCM), penggerak elektronik, rem magnetik, identifikasi sistem, validasi *Root Mean Square Error* (RMSE), kontroler PID-Fuzzy, sensor *rotary encoder*, dan Arduino.

#### 2.1 *Brushless DC Motor* (BLDCM)[1]

*Brushless Direct Current Motor* (BLDCM) adalah salah satu jenis motor yang cepat populer. BLDCM digunakan di dunia industri seperti permobilan, atmosphere, konsumen, otomasi medis, industri dan peralatan instrumentasi. Sesuai dengan namanya, BLDCM tidak menggunakan sikat untuk pergantian medan magnet (komutasi), tetapi dilakukan secara komutasi elektronik.

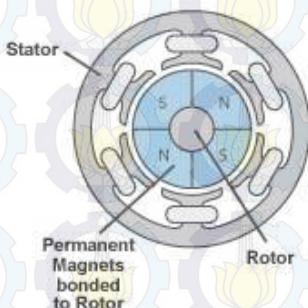
##### 2.1.1 Penjelasan BLDCM

Motor DC banyak digunakan pada aplikasi dunia industri memiliki konstruksi yang menyebabkan timbulnya beberapa kendala pada penggunaannya. Motor DC memiliki kemudahan dalam pengaturan serta perubahan kecepatan mendekati linier dengan perubahan tegangan, akan tetapi memiliki kendala utama yaitu pemakaian komutator *brush* atau sikat arang untuk menghubungkan kumparan rotor. Kendala yang mencakup penggantian atau penyetalan sikat arang, bunga api (*spark*) karena komutasi, keterbatasan arus dan tegangan, dan disipasi panas yang dibangkitkan *rotor*, sehingga diperlukan perawatan secara intensif. BLDCM menjadi solusi untuk mengatasi kendala tersebut. Walaupun BLDCM dapat mengatasi kendala motor DC, akan tetapi metode pengendalian BLDCM menjadi lebih sulit, karena tidak adanya *brush* yang menunjang proses komutasi, selain itu harga untuk BLDCM masih lebih mahal dibanding dengan motor listrik lainnya.

BLDCM adalah motor DC yang dihilangkan sikat dan komutatornya. Secara umum BLDCM terdiri dari dua bagian, yakni, *rotor*, yang terbuat dari permanen magnet dan *stator*, yang terbuat dari kumparan 3 fasa. Walaupun merupakan motor sinkron 3 fasa, motor ini tetap disebut dengan BLDCM karena pada implementasinya BLDCM menggunakan sumber DC sebagai sumber tegangan yang kemudian

diubah menjadi tegangan AC dengan menggunakan *inverter* 3 fasa. Tujuan dari pemberian tegangan AC 3 fasa pada *stator* BLDCM adalah menciptakan medan magnet putar stator agar BLDCM bisa berputar. Rangkaian komutasi elektronik mengeliminasi penggantian sikat arang sehingga kumparan dapat dilewati arus dengan tegangan yang lebih tinggi.

Konstruksi dari BLDCM dapat dilihat pada Gambar 2.1.



Gambar 2.1 Konstruksi BLDCM

### 2.1.2 Prinsip Kerja BLDCM[2]

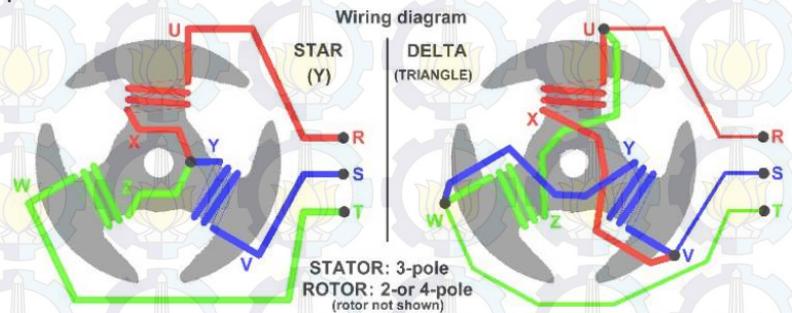
Motor BLDC ini dapat bekerja ketika stator yang terbuat dari kumparan diberikan arus 3 fasa. Akibat arus yang melewati kumparan pada stator timbul medan magnet ( $B$ ):

$$B = \frac{\mu NI}{2l} \quad (2.1)$$

Dimana  $N$  merupakan jumlah lilitan,  $i$  merupakan arus,  $l$  merupakan panjang lilitan, dan  $\mu$  adalah permeabilitas bahan. Arus yang diberikan berupa arus AC, maka nilai medan magnet dan polarisasi setiap kumparan akan berubah-ubah. Medan magnet yang berubah-ubah membuat kutub stator mengikuti perubahan medan magnet, sehingga BLDCM berputar. Kecepatan medan putar magnet dapat dirumuskan seperti Persamaan 2.2.

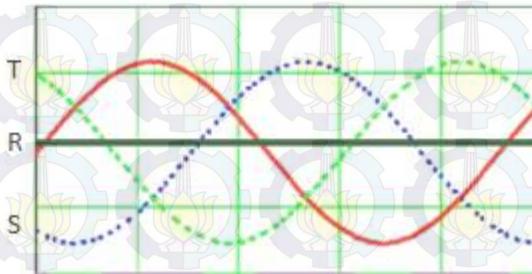
$$N_s = \frac{120f}{p} \quad (2.2)$$

Dimana  $f$  merupakan frekuensi tegangan fasa yang dinyatakan dalam satuan hertz (Hz),  $p$  merupakan jumlah kutub (*pole*) pada rotor dan  $120^\circ$  didapat dalam 1 putaran ( $360^\circ$ ) per 3 fasa motor. Ketika motor berputar permanent magnet pada rotor bergerak melewati kumparan stator dan menginduksi potensial listrik dalam kumparan tersebut, maka terjadinya GGL balik. *Wiring* pada BLDCM dapat dilihat pada Gambar 2.2.



**Gambar 2.2** *Wiring Diagram* BLDCM

Pada Gambar 2.2 dapat dilihat bahwa *wiring diagram* dari BLDCM ada 2 macam. Berdasarkan Gambar 2.2, medan putar magnet stator timbul akibat adanya perubahan polaritas pada stator R, S, dan T. Perubahan polaritas ini terjadi akibat adanya arus yang mengalir pada stator.

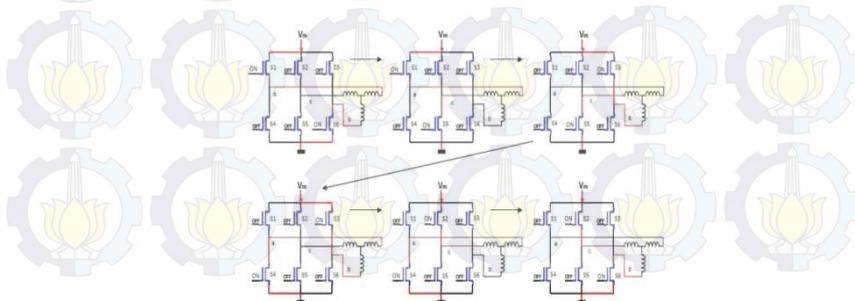


**Gambar 2.3** Tegangan Stator BLDCM

Berdasarkan Gambar 2.3, ketika medan stator R diberikan tegangan negatif maka akan timbul medan magnet dengan polaritas negatif sedangkan medan stator S dan T yang diberikan tegangan positif akan memiliki polaritas positif. Akibat adanya perbedaan polaritas antara medan magnet kumparan stator dan magnet rotor, sisi positif magnet rotor akan berputar mendekati medan magnet stator R, sedangkan sisi negatifnya akan berputar mengikuti medan magnet stator S dan T. Akibat tegangan yang digunakan berupa tegangan AC sinusoidal, medan magnet stator R, S, dan T akan berubah – ubah polaritasnya dan besarnya mengikuti perubahan tegangan AC sinusoidal. Ketika medan stator R dan S memiliki medan magnet negatif akibat mendapatkan tegangan negatif dan medan stator T memiliki medan magnet positif akibat tegangan positif, magnet permanen rotor akan berputar menuju ke polaritas yang bersesuaian yakni bagian negatif akan berputar menuju medan magnet stator T dan sebaliknya bagian positif akan berputar menuju medan magnet stator R dan S. Tegangan AC sinusoidal yang digunakan secara kontinyu, proses perubahan polaritas tegangan pada stator ini akan terjadi secara terus menerus sehingga menciptakan medan putar magnet stator dan magnet permanen rotor akan berputar mengikuti medan putar magnet stator ini. Hal inilah yang menyebabkan rotor pada BLDCM dapat berputar.

## 2.2 Penggerak Elektronik[2]

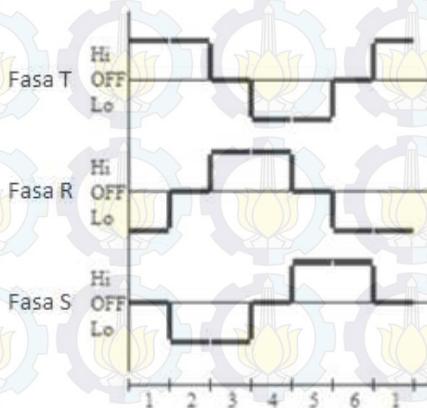
BLDCM membutuhkan enam langkah komutasi yang dilakukan secara kontinyu untuk berputar. *Driver* 3 fasa terdiri dari 6 buah saklar yang akan memberikan tegangan positif (sinyal *high*) dan tegangan 0V (sinyal *low*) secara bergantian. untuk membangkitkan tegangan AC pada BLDCM.



**Gambar 2.4** Urutan *Switching* pada *Stator*

**Tabel 2.1** Urutan Pengaturan Saklar BLDCM

Urutan ke-	Saklar Aktif		Fasa T	Fasa R	Fasa S
1	S1	S6	High	Low	Off
2	S1	S5	High	Off	Low
3	S3	S5	Off	High	Low
4	S3	S4	Low	High	Off
5	S2	S4	Low	Off	High
6	S2	S6	Off	Low	High



**Gambar 2.5** Bentuk Gelombang Stator Hasil Switching

## 2.3 Pengendalian BLDCM

Metode yang digunakan dalam pengaturan BLDCM diantaranya adalah metode *six-step*, metode sinyal PWM, dan metode PWM sinusoidal.

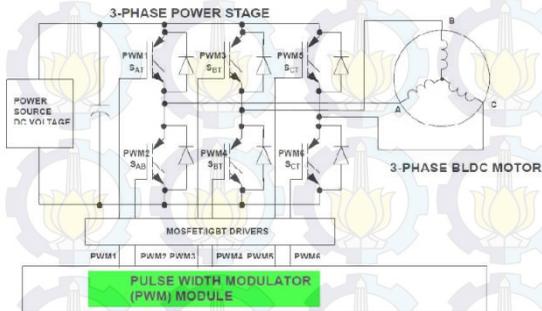
### 2.3.1 Metode *Six-Step*

Metode *six-step* merupakan metode yang umum digunakan dalam pengaturan BLDCM, karena sifatnya yang sederhana, sehingga mudah untuk diaplikasikan. Metode ini disebut metode *six-step*, karena untuk menciptakan gelombang *trapezoid* atau *square* yang menyerupai gelombang sinusoidal, menggunakan 6 step posisi saklar untuk BLDCM. Metode *six-step* ini menggunakan urutan penyalan saklar

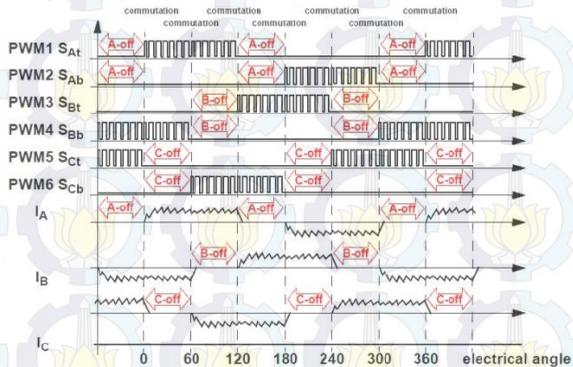
seperti Tabel 2.1. Frekuensi gelombang *square* diatur dengan mengatur waktu tunda (*delay*) pada pergantian tiap urutan penyalan saklar. Penyalan saklar berdasarkan sensor *hall* untuk mengetahui posisi rotor.

### 2.3.2 Metode Sinyal PWM

Metode sinyal PWM menggunakan modul PWM untuk memberikan *trigger* pada 6 buah MOSFET. Sinyal PWM yang diberikan pada MOSFET berbeda-beda seperti. Gambar 2.6, sehingga keluaran yang dihasilkan oleh *driver* tersebut dapat menggerakkan BLDCM terlihat pada Gambar 2.7.



Gambar 2.6 Driver 3 Fasa dengan Modul PWM



Gambar 2.7 Switching Menggunakan Sinyal PWM

## 2.4 Rem Magnetik[3]

Sistem pengereman magnetik menggunakan gaya magnet untuk memperlambat gerakan. Sebuah piringan dengan bahan logam non-magnetik terpasang dengan poros rotor dan diapit oleh sisi yang dipasangi magnet permanen.

Logam piringan non-magnetik berputar memotong medan magnet yang ditimbulkan oleh magnet permanen, sehingga menghasilkan arus *eddy* pada piringan. Arus *eddy* menimbulkan medan magnet yang arahnya berlawanan dengan medan magnet sebelumnya, sehingga memperlambat putaran dari poros tersebut. Rem magnetik digunakan sebagai efek pembebanan, bukan untuk menghentikan gerakan.

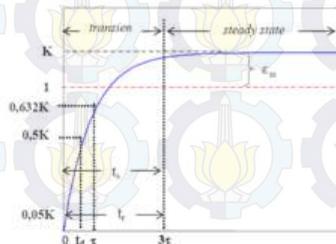
## 2.5 Identifikasi Sistem[4]

Model matematika sistem diperlukan untuk menggambarkan hubungan *input* dengan *output* sistem. Metode identifikasi merupakan suatu metode yang menggunakan hubungan data masukan dan keluaran yang selanjutnya dilakukan pengujian dan analisa dengan metode pendekatan, sehingga dapat ditentukan nilai parameter secara analitik. Beberapa sinyal masukan yang dapat digunakan untuk mendapatkan respon suatu sistem seperti sinyal *step*, *ramp*, *impulse*, dan sinusoidal.

Terdapat dua macam identifikasi sistem yaitu identifikasi statis dan identifikasi dinamis. Pada identifikasi statis, sinyal yang diberikan berupa sinyal *step* yang konstan sampai sistem mencapai keadaan *steady state*. Pada identifikasi dinamis, sinyal yang digunakan berupa sinyal acak (*random*).

### 2.5.1 Identifikasi Statis

Suatu sistem disebut orde satu apabila dilihat secara grafis memiliki bentuk respon seperti yang ditunjukkan pada Gambar 2.8.



Gambar 2.8 Karakteristik Respon Orde Satu

Karakteristik respon sistem orde satu dilihat berdasarkan respon sistem ketika diberi masukan sinyal *step*. Model matematik sistem orde satu dirumuskan dengan:

$$G(s) = \frac{K}{\tau s + 1} \quad (2.3)$$

Terdapat dua parameter yaitu K yang menyatakan *gain overall* dan  $\tau$  yang menyatakan *time constant* (konstanta waktu). Nilai K merupakan hasil perhitungan dari Persamaan 2.4:

$$K = \frac{Y_{ss}}{X_{ss}} \quad (2.4)$$

$Y_{ss}$  adalah keluaran saat *steady state* dan  $X_{ss}$  adalah masukan saat *steady state*. Karakteristik sistem orde satu dibedakan menjadi karakteristik respon transien dan karakteristik respon keadaan tunak atau *steady state*. Karakteristik respon transien pada orde satu terdiri dari:

a. Spesifikasi Teoritis

*Time constant* atau konstanta waktu ( $\tau$ ) merupakan waktu yang dibutuhkan respon mulai  $t = 0$  sampai respon mencapai 63,2% dari respon *steady state*. Konstanta waktu menyatakan kecepatan respon sistem.

b. Spesifikasi Praktis

- *Settling time* atau waktu tunak ( $t_s$ ) merupakan waktu yang menyatakan bahwa respon sistem telah masuk pada daerah *steady state*. Jika dihubungkan dengan konstanta waktu  $\tau$ , maka  $t_s$  dapat dirumuskan dengan:

$$t_s (\pm 5\%) \approx 3\tau \quad (2.5)$$

$$t_s (\pm 2\%) \approx 4\tau \quad (2.6)$$

$$t_s (\pm 0.5\%) \approx 5\tau \quad (2.7)$$

- *Rise time* atau waktu naik ( $t_r$ ) merupakan waktu yang menyatakan bahwa respon sistem telah naik dari 5% - 95% atau 10% - 90% dari nilai respon *steady state*.

$$t_r (5\% - 95\%) = \tau \ln 19 \quad (2.8)$$

$$t_r (10\% - 90\%) = \tau \ln 9 \quad (2.9)$$

- *Delay time* atau waktu tunda ( $t_d$ ) merupakan waktu yang dibutuhkan respon mulai  $t = 0$  sampai respon mencapai 50% dari nilai *steady state*. Waktu tunda menyatakan besarnya faktor keterlambatan respon akibat proses *sampling*.

$$t_d = \tau \ln 2 \quad (2.10)$$

Karakteristik respon *steady state* sistem orde satu diukur berdasarkan kesalahan pada keadaan tunak atau *error steady state* ( $e_{ss}$ ), yaitu:

$$e_{ss} = Y_{ss} - X_{ss} \quad (2.11)$$

### 2.5.1.1 Identifikasi Statis dengan Metode Vitečková Orde 1

Metode ini menggunakan pendekatan orde satu dengan kemungkinan adanya waktu tunda. Fungsi alih untuk metode Vitečková Orde 1 ditunjukkan pada Persamaan 2.12.

$$G_{V1}(s) = \frac{K}{\tau_{V1}s + 1} e^{-T_{dV1}s} \quad (2.12)$$

$T_{dV1}$  adalah waktu tunda (*delay time*)

$$T_{dV1} = 1,498t_{33} - 0,498t_{70} \quad (2.13)$$

$\tau_{V1}$  adalah konstanta waktu

$$\tau_{V1} = 1,245(t_{70} - t_{33}) \quad (2.14)$$

$t_{33}$  dan  $t_{70}$  merupakan waktu saat respon berada pada kondisi 33% dan 70% dari keluaran *steady state*. Apabila  $T_{dV1}$  bernilai negatif, maka sistem dianggap tidak memiliki waktu tunda.

### 2.5.1.2 Identifikasi Statis dengan Metode Vitečková Orde 2

Metode ini menggunakan pendekatan orde dua dengan kemungkinan adanya waktu tunda. Fungsi alih untuk metode Vitečková Orde 2 ditunjukkan pada Persamaan 2.15.

$$G_{V2}(s) = \frac{K}{(\tau_{V2}s + 1)^2} e^{-T_{dV2}s} \quad (2.15)$$

$T_{dV2}$  adalah waktu tunda (*delay time*)

$$T_{dV2} = 1,937t_{33} - 0,937t_{70} \quad (2.16)$$

$\tau_{V2}$  adalah konstanta waktu

$$\tau_{V2} = 0,794(t_{70} - t_{33}) \quad (2.17)$$

$t_{33}$  dan  $t_{70}$  merupakan waktu saat respon berada pada kondisi 33% dan 70% dari keluaran *steady state*. Apabila  $T_{dV2}$  bernilai negatif, maka sistem dianggap tidak memiliki waktu tunda.

### 2.5.1.3 Identifikasi Statis dengan Metode Sundaresan & Krishnaswamy

Metode ini menggunakan pendekatan orde satu dengan kemungkinan adanya waktu tunda. Fungsi alih untuk metode Sundaresan & Krishnaswamy ditunjukkan pada Persamaan 2.18.

$$G_{SK}(s) = \frac{K}{\tau_{SK}s + 1} e^{-T_{dSK}s} \quad (2.18)$$

$T_{dSK}$  adalah waktu tunda

$$T_{dSK} = 1,3t_{35,3} - 0,29t_{85,3} \quad (2.19)$$

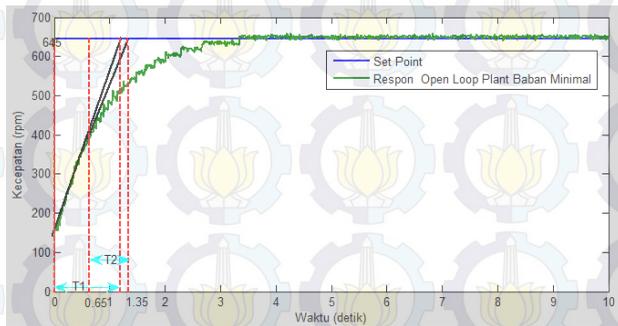
$\tau_{SK}$  adalah konstanta waktu

$$T_{dSK} = 0,67(t_{85,3} - t_{35,3}) \quad (2.20)$$

$t_{35,3}$  dan  $t_{85,3}$  merupakan waktu saat respon berada pada kondisi 35,3% dan 85,3% dari keluaran *steady state*. Apabila  $\tau_{dSK}$  bernilai negatif, maka sistem dianggap tidak memiliki waktu tunda.

#### 2.5.1.4 Identifikasi Statis dengan Metode Grafis

Metode ini merupakan secara grafis dengan menarik garis singgung yang memotong respon *plant* pada Gambar 2.9.



**Gambar 2.9** Penentuan Garis Singgung Metode Grafis

$t_1$  dan  $t_2$  merupakan waktu yang digunakan untuk menentukan akar-akar karakteristik dari sistem nantinya. Pada metode ini untuk mendapatkan  $t_1$  dan  $t_2$  terdapat prosedur tertentu.

Prosedurnya adalah sebagai berikut ;

- 1) Tarik garis singgung dari awal respon sampai garis tersebut mempunyai kemiringan yang sama dengan respon. Garis akan berakhir pada  $X_{ss}$ , pada gambar garis ini ditunjukkan pada garis hitam tebal.
- 2) Tarik garis lurus dari awal respon sampai  $X_{ss}$ , pada gambar garis ini ditunjukkan pada garis putus-putus berwarna merah.
- 3) Selisih antara garis tersebut merupakan nilai  $T_1$
- 4) Tarik lagi garis yang menyinggung dari akhir garis singgung yang pertama sampai menuju  $X_{ss}$ .
- 5) Ulangi lagi langkah 2 pada mulai dari awal garis 2 berpotongan dengan respon.
- 6) Nilai  $t_2$  adalah selisih 2 garis tersebut.
- 7) Setelah nilai  $T_1$  dan  $T_2$  didapatkan maka bentuk didapatkan fungsi transfer sebagai berikut

$$G(s) = \frac{1/T_1}{s + 1/T_1} + \frac{1/T_2}{s + 1/T_2} \quad (2.21)$$

## 2.6 Validasi Model RMSE[4]

Identifikasi parameter sistem didapatkan data untuk mendapatkan model matematik *plant*. Tujuan validasi model secara umum adalah untuk membuktikan bahwa model yang diidentifikasi memenuhi persyaratan permodelan menurut kriteria tujuan (*objective*) dari aproksimasi permodelan yang baik. Model matematika tersebut perlu diuji validasi untuk mengetahui kesamaan dengan *plant* dalam kondisi nyata. Metode validasi model menggugurkan *Root Mean Square Error* (RMSE) untuk mengetahui perbandingan dengan data yang didapat dengan model matematika melalui perhitungan yang disimulasikan.

RMSE adalah pengukuran akurasi pada nilai deret waktu secara statistik, khususnya *trend*. Persamaan 2.22 menyajikan formulasi dasar dari RMSE.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n e_i^2} \quad i = 1,2,3,4,\dots,n \quad (2.22)$$

dimana  $e$  adalah persentase kesalahan data hasil dari permodelan terhadap hasil pengukuran seperti yang dirumuskan pada Persamaan (2.23).

$$e_i = A_i - M_i \quad i = 1,2,3,4,\dots,n \quad (2.23)$$

Keterangan :

$n$  : jumlah data

$i$  : urutan data

$A$  : nilai data hasil pengukuran

$M$  : nilai data hasil permodelan

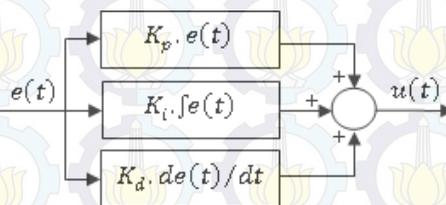
RMSE adalah komponen terpadu dari sebuah model statistika seperti regresi. Hal ini menjadikan RMSE sebagai sebuah pengukuran alami yang digunakan dalam berbagai evaluasi terhadap perkiraan kesalahan yang menggunakan metode statistika. Secara umum, tidak ada kriteria mutlak dari sebuah data untuk dianggap “bagus”. Keuntungan yang

dimiliki RMSE adalah kesamaan skala yang dimiliki oleh data hasil pengukuran dengan data hasil permodelan, sehingga RMSE dapat merepresentasikan ukuran dari kesalahan rata-rata. Nilai mutlak dari penghitungan ini dijumlahkan untuk setiap titik yang dilengkapi atau perkiraan dalam domain waktu dan dibagi kembali dengan jumlah  $n$  pengambilan data. Hal ini membuat kesalahan diukur dalam persen sehingga kesalahan sebuah deret waktu dapat dibandingkan di tingkat yang berbeda.

## 2.7 Kontroler PID[5]

Kontroler PID merupakan kontroler yang banyak digunakan di industri. Hal ini dikarenakan kemudahan penggunaannya terutama pada segi perhitungan. Kontroler PID melakukan mekanisme kontrol melalui selisih antara masukan dengan nilai keluaran yang diumpan balik. Kontroler ini terdiri dari 3 parameter yaitu proporsional, *integral* dan derivatif. Kontrol proporsional melakukan perkalian dengan kesalahan. Kontrol *integral* melakukan penjumlahan kesalahan tiap waktu. Kontrol differensial melakukan aksi kontrol berdasarkan kecepatan perubahan sinyal kesalahan. Ketiga kontroler ini selanjutnya ditambahkan untuk menghasilkan sinyal kontrol  $u(t)$  seperti pada Gambar 2.10.

$$u(t) = K_p \left( e(t) + \frac{1}{\tau_i} \int_0^t e(t) + \tau_d \frac{de(t)}{dt} \right) \quad (2.24)$$



Gambar 2.10 Diagram Blok Kontroler PID

### 2.7.1 Kontroler Proporsional (P)

Kontroler proporsional merupakan kontroler yang aksi kontrolnya proporsional atau sebanding terhadap sinyal kesalahan. Kontroler proporsional berupa konstanta dan secara sederhana keluaran kontroler proporsional merupakan perkalian antara konstanta

proporsional dengan masukannya. Persamaan dari kontrol proporsional dapat dilihat sebagai berikut:

$$u(t) = K_p (e(t)) \quad (2.25)$$

Karakteristik dari kontroler proporsional adalah dapat mempercepat respon sistem untuk mencapai keadaan *steady state*. Semakin besar nilai kontrol proporsional akan memperkecil kesalahan keadaan *steady state*, namun akan membawa sistem ke daerah tidak stabil, karena itu kontroler proporsional tidak dapat menghilangkan nilai *error* ( $e \neq 0$ ), sehingga diperlukan kontroler *integral* agar nilai *error* dapat mendekati nol.

### 2.7.2 Kontroler Integral (I)

Karakteristik kontroler *integral* adalah menghasilkan respon sistem yang memiliki nilai *error steady state* = 0 (*error* mendekati nol), namun kontroler ini cenderung memperlambat respon. Keluaran kontroler *integral* merupakan penjumlahan yang terus menerus dari perubahan masukannya. Jika sinyal kesalahan tidak mengalami perubahan, keluaran akan menjaga keadaan seperti sebelum terjadinya perubahan masukan. Persamaan dari kontrol *integral* dapat dilihat sebagai berikut:

$$u(t) = \frac{K_p}{\tau_i} \left( \int_0^t e(t) \right) \quad (2.26)$$

### 2.7.3 Kontroler Derivatif (D)

Karakteristik keluaran kontroler diferensial memiliki sifat seperti operasi derivatif. Perubahan yang mendadak pada masukan kontroler, akan mengakibatkan perubahan yang sangat besar dan cepat pada keluaran. Persamaan dari kontrol diferensial dapat dilihat sebagai berikut:

$$u(t) = K_p \tau_d \left( \frac{de(t)}{dt} \right) \quad (2.27)$$

Kelebihan kontroler derivatif apabila ditambah dengan kontroler proporsional dapat mengurangi aksi kontrol yang berlebihan sehingga tidak mengalami *overshoot*.

## 2.8 Logika Fuzzy[6]

Konsep logika *fuzzy* pertama kali dikenalkan pada tahun 1965 oleh L.A. Zadeh dari universitas California Barkeley dalam makalah seminarnya yang berjudul “*Fuzzy Set*”. Pada konsep logika *fuzzy* harga kebenaran diberikan dalam variabel bahasa (linguistik) dengan menyertakan sifat kekaburan (*fuzziness*) pada proposisinya. Harga kebenaran dan derajat kekaburan pada variabel linguistik dapat dinyatakan dengan tolok ukur, misalnya agak, cukup, sangat, dan sebagainya. Sehingga dalam logika *fuzzy* memungkinkan nilai keanggotaan antara 0 dan 1. Ini yang membedakan antara logika *fuzzy* dan logika Boolean yang nilai keanggotaannya hanya 0 dan 1.

### 2.8.1 Himpunan Fuzzy

Suatu himpunan *fuzzy*  $F$  dalam semesta  $X$  didefinisikan sebagai kumpulan pasangan elemen  $x$  dan fungsi keanggotaan  $\mu_F(x)$ . Fungsi keanggotaan  $\mu_F(x)$  mempunyai nilai interval  $[0, 1]$  pada tiap  $x$  dalam semesta  $X$ . Nilai fungsi keanggotaan menunjukkan tingkat keanggotaan elemen  $x$  dalam  $F$ . Tingkat keanggotaan 1 menunjukkan keanggotaan penuh (*full membership*), tingkat 0 menyatakan tanpa keanggotaan (*non membership*), tingkat antara 0 dan 1 menyatakan kekaburan antara keanggotaan penuh dan tanpa keanggotaan.

Tingkat keanggotaan inilah yang biasanya disebut tingkat logika *fuzzy*. Himpunan *fuzzy* pada  $F$  pada semesta pembicaraan  $X$  yang anggotanya berhingga,  $X = \{x_1, x_2, x_3, \dots, x_n\}$  dapat dinyatakan dalam bentuk himpunan *fuzzy*. Secara umum himpunan *fuzzy* dinotasikan sebagai :

$$F = \left\{ (x, \mu_F(x)) \mid x \in X \right\} \quad (2.28)$$

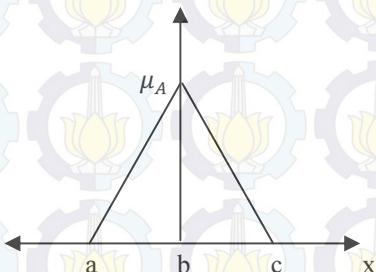
### 2.8.2 Fungsi Keanggotaan Fuzzy

Fungsi keanggotaan adalah suatu fungsi untuk menyatakan hubungan antara sebuah masukan dengan sebuah himpunan *fuzzy*. Untuk menyatakan suatu fungsi keanggotaan dalam pendefinisian himpunan *fuzzy* tergantung pada metode yang akan digunakan. Secara umum ada

dua metode yaitu pendefinisian secara numerik dan bentuk fungsi. Pendefinisian secara numerik diterapkan atas himpunan dengan pendukung diskrit, sedangkan pendefinisian bentuk fungsi diterapkan pada pendukung kontinyu.

a. Fungsi Keanggotaan Segitiga (Triangular)

Fungsi keanggotaan segitiga (tringular) sesuai dengan namanya berbentuk segitiga dengan parameter a, b dan x pada Persamaan 2.29. Fungsi keanggotaan segitiga ditunjukkan oleh Gambar 2.11.



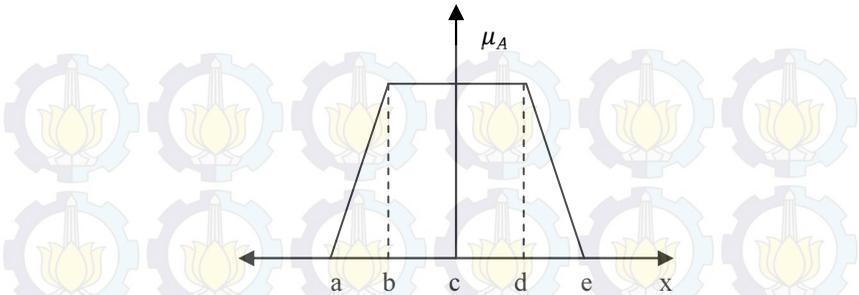
Gambar 2.11 Fungsi Keanggotaan Segitiga

$$\mu_A = \begin{cases} 0, & x \leq a; x \geq c \\ \frac{x-a}{b-a}, & a < x < b \\ \frac{c-x}{c-b}, & b < x < c \end{cases} \quad (2.29)$$

b. Fungsi Keanggotaan Trapesium

Fungsi ini menggunakan parameter a dan b, dengan Persamaan 2.30, serta bentuk fungsi keanggotaan terlihat pada Gambar 2.12.

$$\mu_A(x) = \begin{cases} 0, & x \leq a; x \geq e \\ \frac{x-a}{b-a}, & a < x < b \\ 1, & b \leq x < d \\ \frac{d-x}{d-c}, & d < x < e \end{cases} \quad (2.30)$$



**Gambar 2.12** Fungsi Keanggotaan Trapesium

### 2.8.3 Operasi Himpunan *Fuzzy*

Operasi dasar yang berlaku pada himpunan *fuzzy* A dan B dari semesta X dengan tingkat keanggotaan  $\mu_A(x)$  dan  $\mu_B(x)$  untuk  $x \in X$  adalah sebagai berikut :

- Dua himpunan *fuzzy* A dan B dikatakan sama ( $A = B$ ) apabila keduanya memiliki semesta yang sama dan fungsi keanggotaan yang sama. Hal ini dapat dinyatakan dengan Persamaan 2.31.

$$\frac{\mu(x_A)}{x} = \frac{\mu(x_B)}{x}; x \in X \quad (2.31)$$

- Gabungan dua himpunan *fuzzy* A dan B ( $A \cup B$ ) dinyatakan dengan Persamaan 2.32.

$$\mu_A \cup B = \max(\mu_A(x), \mu_B(x)); x \in X \quad (2.32)$$

- Irisan dari himpunan *fuzzy* A dan B ( $A \cap B$ ) dinyatakan seperti yang ditunjukkan pada Persamaan 2.33.

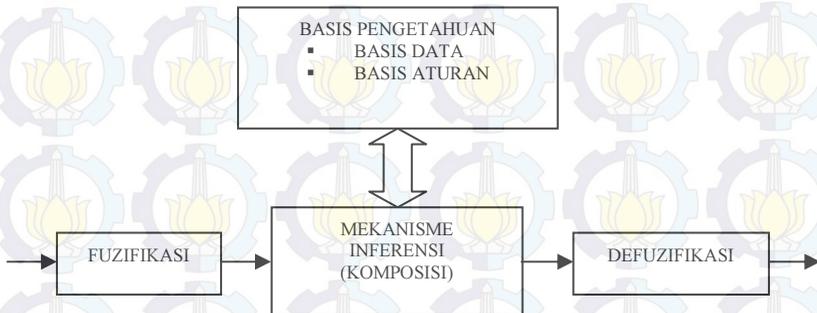
$$\mu_A \cap B = \min(\mu_A(x), \mu_B(x)); x \in X \quad (2.33)$$

- Komplemen (operasi “NOT”) himpunan *fuzzy* A dinotasikan dengan  $\bar{A}$  dapat dinyatakan pada Persamaan 2.34.

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x); x \in X \quad (2.34)$$

### 2.8.4 Sistem Kontrol Fuzzy[6]

Sistem kontrol *fuzzy* merupakan pemetaan dari masukan menjadi keluaran *fuzzy*. Secara umum sistem kontrol *fuzzy* terdiri dari empat bagian dasar yaitu fuzifikasi, aturan basis (*Rule Base*), inferensi *fuzzy*, dan defuzifikasi. Proses pada sistem kontrol *fuzzy* dimulai dari nilai masukan *fuzzy* diubah menjadi himpunan *fuzzy* melalui proses fuzifikasi. Selanjutnya himpunan *fuzzy* masuk ke mekanisme inferensi *fuzzy* untuk menentukan kesimpulan *fuzzy* dengan mengacu pada *rule base*. Proses akhir masuk ke proses defuzifikasi untuk diubah menjadi keluaran *fuzzy*. Sistem kontrol *fuzzy* ditunjukkan pada Gambar 2.13.



Gambar 2.13 Sistem Kontrol Fuzzy

a. Fuzifikasi

Fuzifikasi merupakan proses yang mengubah variabel nyata menjadi variabel *fuzzy*. Hal ini bertujuan agar masukan kontroler *fuzzy* bisa dipetakan menuju jenis yang sesuai dengan himpunan *fuzzy*. Pemetaan dilakukan dengan bantuan model dari fungsi keanggotaan agar dapat diketahui besar masukan tersebut.

b. Aturan Basis (*Rule Base*)

Kaidah dasar yang berisi aturan-aturan secara linguistik yang menunjukkan kepakaran terhadap *plant* atau suatu aturan yang menghubungkan antara masukan *fuzzy* dan aksi kontrol dari kontroler *fuzzy*. Format yang paling umum adalah aturan *IF-THEN*, sebagai berikut:

***IF*** premise ***THEN*** conclusion

c. Inferensi *Fuzzy*

Inferensi *fuzzy* adalah sebuah proses formulasi pemetaan masukan terhadap keluaran dengan menggunakan logika *fuzzy*. Proses dari inferensi *fuzzy* melibatkan fungsi keanggotaan operator logika *fuzzy*, dan aturan *IF-THEN*. Terdapat dua metode inferensi yang paling dikenal yaitu metode inferensi Mamdani dan metode inferensi Takagi-Sugeno. Pada metode Mamdani, menggunakan operasi Max-Min atau *Max-product* dalam memperoleh keluaran, sedangkan pada metode Takagi-Sugeno menggunakan fungsi keanggotaan keluaran yang bersifat linear atau konstan.

- Metode Mamdani

$$\mu_u(k) = \max [\mu_u(k), \min \{\mu_e(j), \mu_{de}(i)\}] \quad (2.35)$$

d. Defuzzifikasi

Defuzzifikasi adalah proses yang digunakan untuk mengubah kembali variabel *fuzzy* menjadi variabel nyata, atau dengan kata lain aksi kontrol *fuzzy* yang masih berupa himpunan, diubah menjadi nilai nyata yang berupa nilai tunggal. Metode yang dapat digunakan untuk defuzzifikasi yaitu *Center of Average* (COA) dengan Persamaan 2.36.

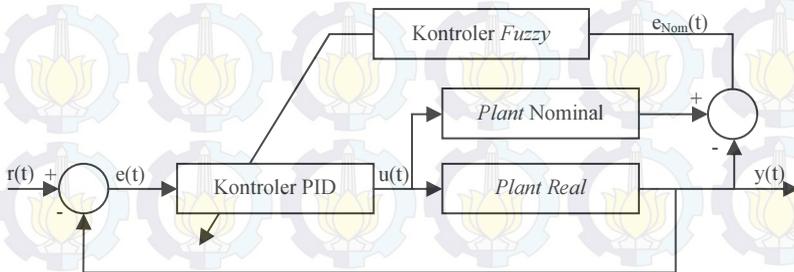
$$U_0 = \frac{\sum_{k=1}^m u_k(T) \cdot \mu_u(u_k(T))}{\sum_{k=1}^m \mu_k(u_k(T))} \quad (2.36)$$

## 2.9 Kontroler PID-*Fuzzy*

Lebih dari 90% kontroler yang digunakan saat ini adalah kontroler PID. Hal ini dikarenakan kontroler PID mudah dimengerti dan mudah diimplementasikan. Namun, kontroler PID hanya dapat digunakan dengan baik untuk parameter *plant* dan kondisi operasi yang tetap. Oleh karena itu, diperlukan mekanisme penalaan parameter kontroler PID yang dapat menyesuaikan dengan perubahan parameter *plant* dan kondisi operasi sistem.

Mekanisme kontroler PID-*Fuzzy* ditunjukkan seperti pada Gambar 2.14, dimana masukan *fuzzy* adalah keluaran sistem yang

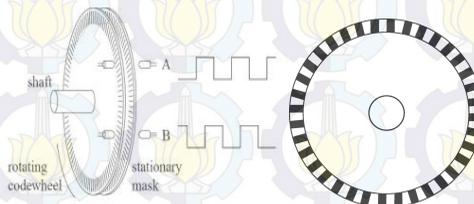
dibandingkan dengan keluaran referensi. Apabila terjadi perbedaan, kontroler *fuzzy* akan melakukan mekanisme penalaan parameter kontroler PID. Parameter PID yang didapat saat beban minimal, nominal, dan maksimal akan dijadikan sebagai fungsi anggota keluaran sinyal hasil dari defuzifikasi. Hasil defusifikasi akan diteruskan pada kontroler PID sebagai parameternya. *Plant* nominal yang berisi respon model yang diinginkan, sehingga *output plant* diharapkan mendekati respon pada saat beban nominal.



Gambar 2.14 Diagram Blok Kontroler PID-Fuzzy

## 2.10 Sensor *Rotary Encoder*[6]

Sensor yang digunakan untuk memonitor kecepatan putaran motor induksi pada Tugas Akhir ini adalah *rotary encoder*. *Rotary encoder* digunakan untuk mengubah gerakan linier atau putaran menjadi sinyal digital, dimana sensor putaran memonitor gerakan putar dari suatu alat. *Rotary encoder* umumnya menggunakan sensor optik untuk menghasilkan pulsa yang dapat diartikan menjadi gerakan, posisi, dan arah.. Struktur *rotary encoder* dapat dilihat pada Gambar 2.15



Gambar 2.15 Keluaran Sinyal dan Susunan dari *Rotary Encoder*

## 2.11 Arduino[7]

Arduino adalah pengendali mikro *single-board* yang bersifat *open-source*, diturunkan dari Wiring Platform, dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang. *Hardware*nya memiliki prosesor Atmel AVR dan *software*nya memiliki bahasa pemrograman sendiri. Saat ini Arduino sangat populer di seluruh dunia. Banyak pemula yang belajar mengenal robotika dan elektronika lewat Arduino karena mudah dipelajari. Bahasa yang dipakai dalam Arduino bukan *assembler* yang relatif sulit, tetapi bahasa C yang disederhanakan dengan bantuan pustaka-pustaka (*library*) Arduino. Arduino juga menyederhanakan proses bekerja dengan mikrokontroler, sekaligus menawarkan berbagai macam kelebihan antara lain:

- Tidak perlu perangkat *chip programmer* karena di dalamnya sudah ada *bootloader* yang akan menangani *upload program* dari komputer.
- Sudah memiliki sarana komunikasi USB, sehingga pengguna *notebook* yang tidak memiliki *port serial/RS232* bisa menggunakannya.
- Bahasa pemrograman relatif mudah karena *software* Arduino dilengkapi dengan kumpulan *library* yang cukup lengkap.
- Memiliki modul siap pakai (*shield*) yang bisa ditancapkan pada board Arduino. Misalnya *shield* GPS, *Ethernet*, *SD Card*, dll



**Gambar 2.16** Board Arduino UNO (kiri) dan Arduino Mega2560 (kanan)



## BAB III PERANCANGAN SISTEM

Bab III menjelaskan tentang perancangan sistem secara keseluruhan yang dibagi menjadi beberapa bagian, yaitu arsitektur sistem, perancangan *hardware*, perancangan *software*, identifikasi BLDCM, dan perancangan simulasi dan implementasi kontroler.

### 3.1 Arsitektur Sistem

Sistem yang dirancang adalah sistem penggerak elektronik dan pengaturan kecepatan dari BLDCM. Sistem yang dirancang ini mempunyai tujuan agar dapat membuat rangkaian penggerak elektronik sebagai komutasi elektrik stator untuk menggerakkan BLDCM dan menghasilkan kecepatan motor yang diinginkan. Prinsip kerja dari sistem yaitu rangkaian penggerak elektronik berfungsi untuk menyuplai tegangan AC tiga fasa ke BLDCM yang akan dioperasikan pada kecepatan tertentu sesuai dengan *set point*. Sistem pengaturan digunakan untuk mengatur kecepatan motor supaya mendekati dengan *set point*. Sinyal *feedback* dijadikan acuan terhadap sinyal referensi yang nantinya digunakan untuk mengevaluasi terhadap kontroler. Penggerak elektronik digunakan sebagai aktuator sebelum diteruskan ke *plant* BLDCM. Arduino mega2560 berfungsi sebagai *data logging*, karena pada arduino UNO sudah difungsikan sebagai pembangkit sinyal frekuensi dan sinyal PWM, akan sangat sulit bila ditambah fungsi sebagai *data logging*.

Arsitektur dari sistem ditunjukkan seperti pada Gambar 3.1.



Gambar 3.1 Arsitektur Sistem

Keterangan:

- : Sistem bagian *software*.
- : Sistem bagian *hardware*.

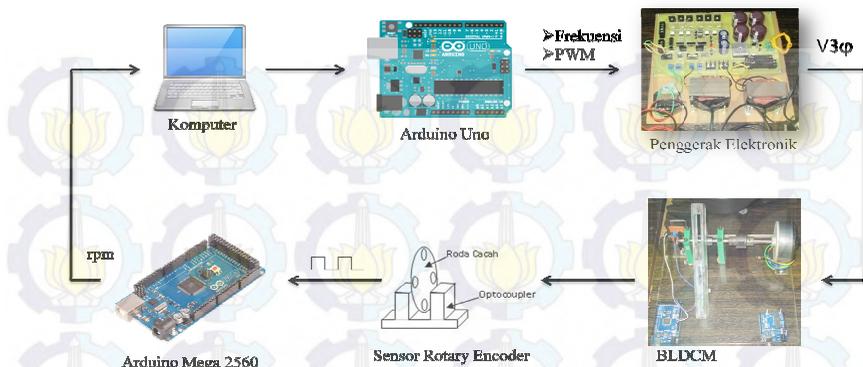
Komponen yang digunakan pada sistem ini, adalah komputer, arduino, penggerak elektronik, BLDCM, dan *rotary encoder*. Pada sistem ini, komputer sebagai HMI (*Human Machine Interface*) yang digunakan untuk mengetahui respon keluaran sistem, membuat kontroler dan juga digunakan untuk memberikan nilai *setpoint* yang diinginkan. Arduino sebagai pembangkit frekuensi fasa, pwm, dan pembacaan data dari sensor *encoder*. Penggerak elektronik sebagai aktuator yang digunakan untuk pemberian aksi dalam pengaturan kecepatan BLDCM yang telah dibuat oleh kontroler. *Rotary encoder* sebagai *feedback* berupa sensor kecepatan yang digunakan untuk mengetahui nilai *error* yang terjadi dalam sistem kontrol terhadap *set point*. Nilai *error* yang akan digunakan oleh kontroler untuk memberikan aksi kontrol hingga nilai *error* semakin kecil.

Perancangan sistem dibagi menjadi empat tahap. Tahap pertama ialah perancangan perangkat keras (*hardware*) yang meliputi perancangan mekanik *plant*, rangkaian penggerak elektronik, rangkaian sensor *rotary encoder*. Tahap kedua berupa perancangan perangkat lunak (*software*) untuk menampilkan respon dari BLDCM, juga sebagai *Human Machine Interface* (HMI) menggunakan Matlab dan pemrograman arduino. Tahap ketiga berupa identifikasi *plant* dan perancangan kontroler PID-Fuzzy.

### **3.2 Perancangan Perangkat Keras (*Hardware*)**

Perangkat keras (*Hardware*) dari sistem terdiri dari beberapa komponen, yaitu arduino, rangkaian penggerak elektronik, BLDCM, rangkaian sensor *rotary encoder*, rem magnetik, dan komputer. Perangkat keras arduino yang digunakan adalah arduino UNO dan arduino MEGA 2560. Komponen dan alur dari perangkat keras sistem secara spesifik ditunjukkan pada Gambar 3.2.

Tahap pertama dalam perancangan sistem adalah perancangan perangkat keras (*hardware*) yang meliputi perancangan mekanik *plant*, perancangan rem magnetik, perancangan sensor *rotary encoder*, dan perancangan penggerak elektronik. Alur dari perangkat keras sistem secara spesifik diperlihatkan pada Gambar 3.2.



**Gambar 3.2** Komponen dan Alur Perangkat Keras Sistem

Dari Gambar 3.2 dapat dilihat bahwa untuk menjalankan BLDCM diperlukan sebuah komputer yang berfungsi sebagai HMI. HMI digunakan untuk memonitor nilai *set point* (SP), *present value* (PV), *error*, dan parameter kontrol. Nilai *set point* diberikan melalui komputer dan selanjutnya dikirim ke arduino melalui *serial port* yang akan memberikan aksi pada penggerak elektronik *inverter*. Arduino memberikan keluaran berupa frekuensi dan PWM ke rangkaian penggerak elektronik. *Penggerak elektronik* akan mengolah frekuensi tegangan masukan dari arduino ini menjadi frekuensi 10-50 Hz tiga fasa dan pwm dengan *duty cycle* 80% dengan frekuensi 4 KHz.

Sensor *rotary encoder* digunakan untuk mengukur kecepatan BLDCM yang berputar diantara *optocoupler* tipe U. Keluaran dari sensor ini berupa pulsa dan dihubungkan ke modul arduino mega. Hasil pembacaan kecepatan dari *rotary encoder* digunakan untuk mengetahui nilai *error* yang terjadi dengan nilai *set point* yang nanti dijadikan parameter untuk mengubah sinyal kontrol.

### 3.2.1 Perancangan Mekanik *Plant*

*Plant* yang digunakan dalam Tugas Akhir ini adalah BLDCM ARW31S8P30AM buatan Malaysia dengan spesifikasi yang tertera pada *name plate* seperti yang ditunjukkan pada Gambar 3.3. Motor ini juga digunakan untuk menggerakkan baling-baling dari AC (*Air Conditioner*) *Inverter* Panasonic.

**Tabel 3.1** Spesifikasi BLDCM

<i>Name Plate</i>	Keterangan
8P	Jumlah kutub yang terpasang pada rotor sebanyak 8 buah atau setara 4 pasang kutub
30W	Daya nominal yang diserap oleh BLDCM
DC280-340V	Tegangan yang diberikan pada BLDCM sebesar 280-340 VDC

*Shaft* dari BLDCM diberi piringan alumunium yang diapit rem magnetik ketika BLDCM berputar. Beban rem magnetik tersebut digunakan untuk mempengaruhi sistem kerja dari BLDCM. Pengaruh yang ditimbulkan oleh rem magnetik berupa penurunan kecepatan dari BLDCM. Bentuk dari *plant* ditunjukkan pada Gambar 3.3.



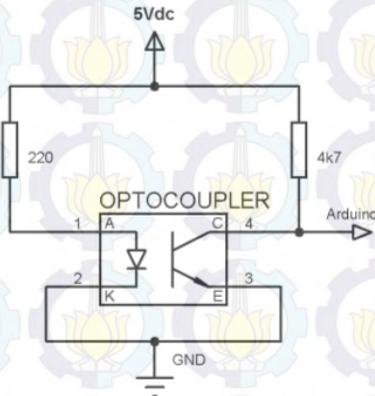
**Gambar 3.3** Kontruksi *Plant* Secara Keseluruhan

Pada Gambar 3.3 tampak bahwa *shaft* BLDCM terhubung dengan *shaft* yang terdapat piringan alumunium. Penghubung antar *shaft* menggunakan *coupling*. *Coupling* selain berguna untuk menggabungkan dua *shaft* juga berguna untuk meredam getaran yang ditimbulkan antar *shaft* saat motor berputar. Rem magnetik pada *plant* terdiri dari dua buah magnet permanen. Magnet diletakkan pada dudukan yang didesain agar bisa diubah posisinya saat motor berputar. Pada ujung *shaft* diletakkan piringan dari sensor *rotary encoder*.

### 3.2.2 Perancangan Sensor *Rotary Encoder*

Sensor kecepatan yang digunakan pada Tugas Akhir ini adalah berupa piringan atau roda cacah dan sebuah *optocoupler* tipe U.

Piringan *rotary encoder* berputar diantara celah pada *optocoupler*. *Optocoupler* merupakan komponen *optoisolator* yang memiliki karakteristik penerima (*phototransistor*) akan mengalami perubahan logika 0 ke 1 atau sebaliknya bila terjadi perubahan intensitas cahaya yang dipancarkan oleh pemancar (LED infra merah) untuk penerima. Skema rangkaian dapat dilihat pada Gambar 3.4.



**Gambar 3.4** Skema Rangkaian Sensor *Rotary Encoder*

Gambar 3.4, merupakan rangkaian untuk *rotary encoder*. *Phototransistor* merupakan jenis transistor yang peka terhadap cahaya infra merah. Piringan akan ditempatkan di tengah dari *optocoupler* tipe U yang berfungsi untuk mempengaruhi intensitas cahaya yang diberikan oleh LED pada *optocoupler* ke *phototransistor* yang akan memberikan perubahan level logika sesuai dengan putaran piringan. Perubahan logika dari *HIGH* ke *LOW* akan dibaca oleh arduino mega2560 sebagai frekuensi pulsa dimana nantinya akan dikonversi menjadi rpm. *Konstruksi* sensor *rotary encoder* dapat dilihat pada Gambar 3.5.



**Gambar 3.5** Konstruksi *Rotary Encoder* dengan *Optocoupler*

### 3.2.3 Perancangan Rangkaian Penggerak Elektronik

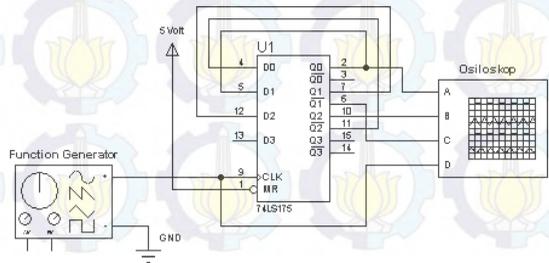
Skema perancangan penggerak elektronik di desain sebaik mungkin dan digabung menjadi satu kesatuan, dimulai dari rangkaian *power supply*, pengolah sinyal fasa dan pwm, *optocoupler*, dan bagian *switching*. Skema rangkaian terlihat pada lampiran. Bentuk fisik rangkaian penggerak elektronik terlihat pada Gambar 3.6. Rangkaian *power supply* menggunakan tegangan sebesar 12 VAC, 18 VAC, dan 64 VAC yang nantinya diubah menjadi tegangan DC sebesar 5 VDC untuk rangkaian pembangkit sinyal frekuensi, 12 VDC untuk rangkaian *optoisolator*, dan 90 VDC untuk rangkaian *switching* yang mensuplai tegangan BLDCM.



Gambar 3.6 Bentuk Fisik Rangkaian Penggerak Elektronik

#### 3.2.3.1 Rangkaian Pembagi Fasa

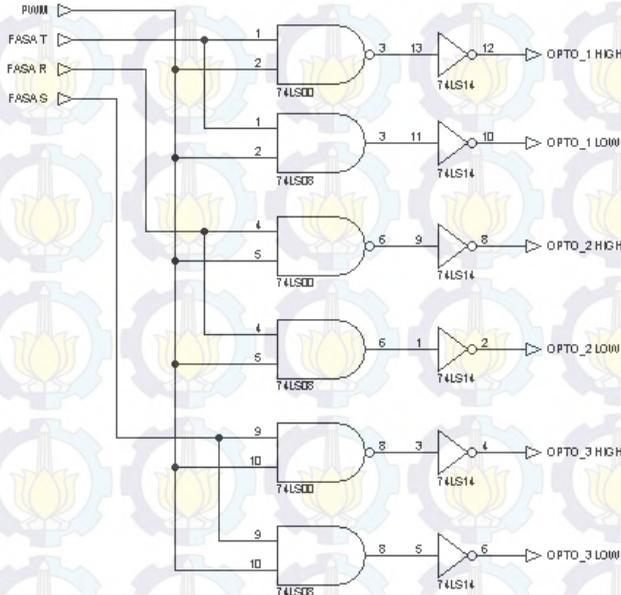
Rangkaian ini merupakan menggunakan IC 74LS175. Masukkan dari rangkaian ini berupa sinyal frekuensi fasa, sedangkan keluaran dari rangkian pembagi fasa berupa sinyal kotak tiga fasa. Skema rangkaian dapat dilihat pada Gambar 3.7.



Gambar 3.7 Skema Rangkaian Pembagi Fasa

### 3.2.3.2 Rangkaian Pengolah Sinyal Fasa dan PWM

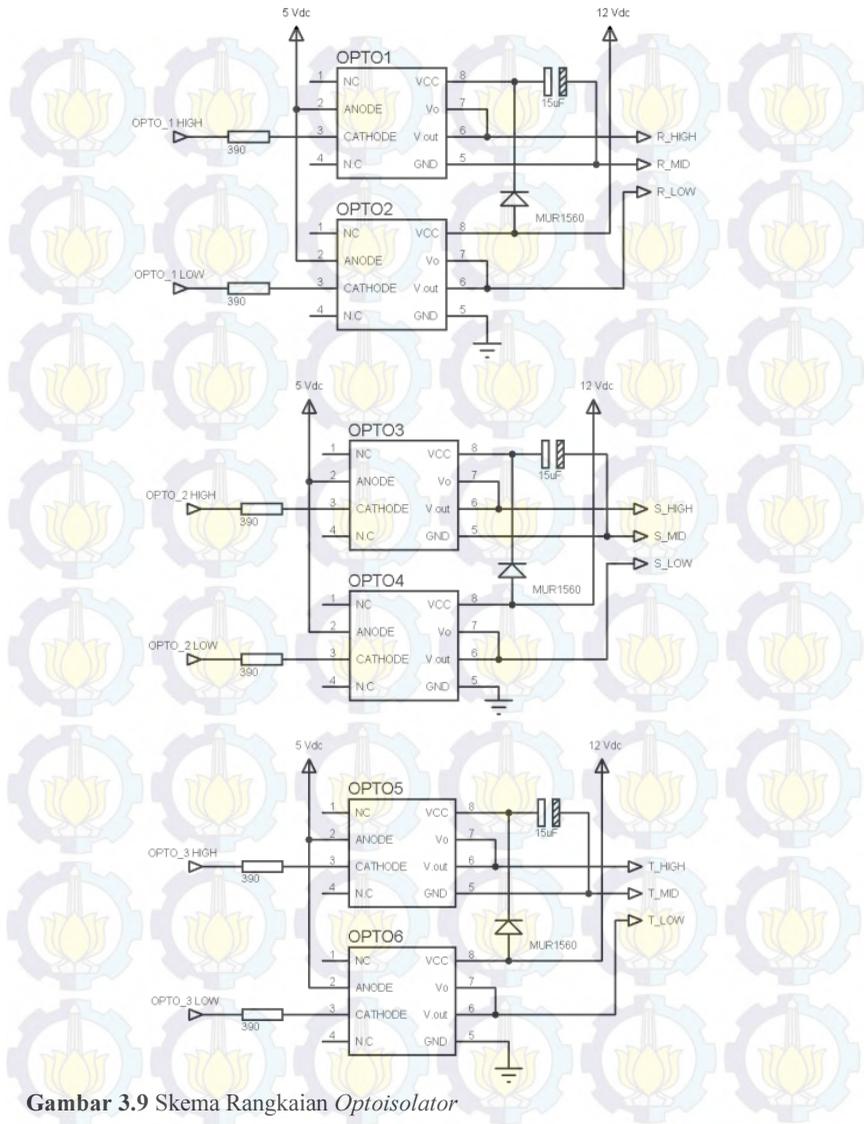
Rangkaian ini terdiri dari rangkaian gerbang logika AND, NAND, dan NOT. Rangkaian ini berfungsi untuk memasukkan sinyal PWM kedalam sinyal frekuensi. Skema rangkaian dapat dilihat pada Gambar 3.8.



Gambar 3.8 Skema Rangkaian Pengolah Sinyal Fasa dan PWM

### 3.2.3.3 Rangkaian Optoisolator

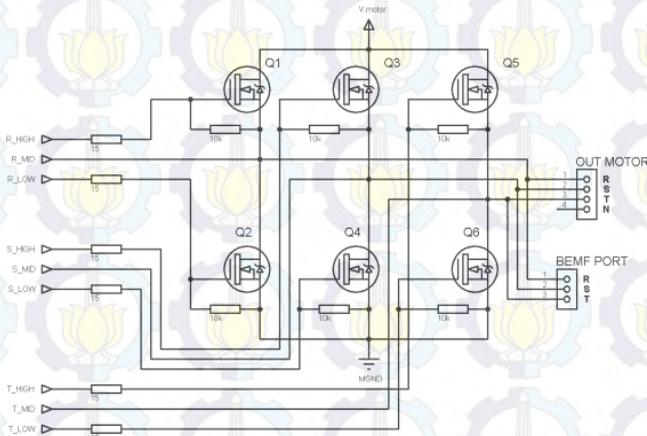
Rangkaian ini terdiri dari *optocoupler* tipe TLP 250, dimana fungsinya sebagai pemisah antara rangkaian pembangkit sinyal dengan rangkaian *switching*. Hal ini diperlukan untuk menghindari terjadinya hubung singkat apabila pada rangkaian *switching* mengalami hubung singkat. Skema rangkaian dapat dilihat pada Gambar 3.9. Pemilihan komponen *optocoupler* harus melihat kemampuan *switching* berdasarkan *datasheet* yang tersedia, hal ini dikarenakan pada rangkaian penggerak elektronik juga menggunakan sinyal masukkan PWM yang berfrekuensi tinggi.



**Gambar 3.9** Skema Rangkaian *Optoisolator*

### 3.2.3.4 Rangkaian Switching

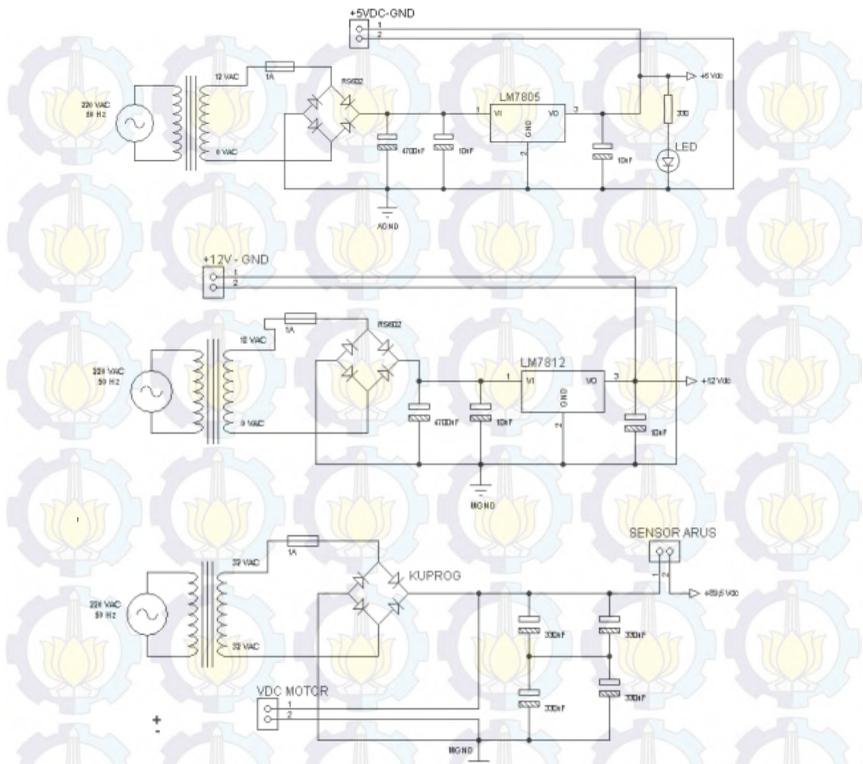
Rangkaian ini terdiri dari enam buah MOSFET yang digunakan untuk *switching* dari tegangan DC menjadi tegangan AC 3 fasa. Penggunaan resistor untuk membatasi arus yang masuk ke kaki *gate* MOSFET, sedangkan resistor *gate-source* MOSFET digunakan untuk mempercepat transisi dari keadaan *HIGH* menjadi keadaan *LOW*. Skema rangkaian *switching* dapat dilihat pada Gambar 3.10.



Gambar 3.10 Skema Rangkaian Switching

### 3.2.3.5 Rangkaian Power Supply

Rangkaian ini merupakan catu daya yang sangat diperlukan oleh rangkaian penggerak elektronik. Rangkaian ini terdiri dari IC *regulator* LM7805 dan LM 7812, kapasitor, dioda *bridge*, dan *fuse*. Selain itu diperlukannya komponen trafo CT (*Center Tap*) dan trafo *non-CT*. Trafo CT digunakan untuk *supply* daya BLDCM, trafo *non-CT* digunakan untuk *supply* 5Vdc dan 12 Vdc. Catu daya yang dibutuhkan yaitu catu daya 5 Vdc untuk rangkaian pembagi fasa, pengolah sinyal fasa dan PWM. Catu daya 12 Vdc digunakan oleh rangkaian *optoisolator*. Catu daya 89,6 Vdc sebagai sumber tegangan MOSFET *HIGH*. Skema rangkaian dapat dilihat pada Gambar 3.11.



Gambar 3.11 Skema Rangkaian Power Supply

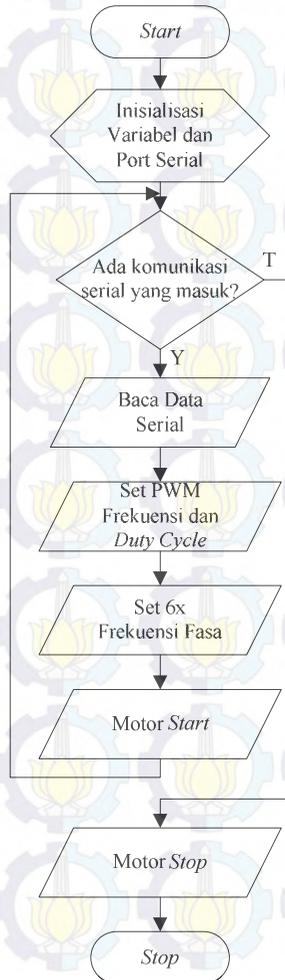
### 3.3 Perangkat Perangkat Lunak (Software)

Perangkat (*software*) yang digunakan dalam Tugas Akhir ini antara lain *program* pembangkit frekuensi dan *program* baca kecepatan yang dilakukan oleh mikrokontroler arduino. Alur pemrograman sistem secara keseluruhan.

#### 3.3.1 Arduino sebagai Pembangkit Frekuensi dan PWM

Arduino UNO digunakan sebagai pembangkit frekuensi dimana perintah untuk membangkitkan frekuensi dilakukan oleh komputer dengan menggunakan *software Matlab*. *Flowchart* pemrograman Arduino dapat dilihat pada Gambar 3.12. Frekuensi yang dibangkitkan

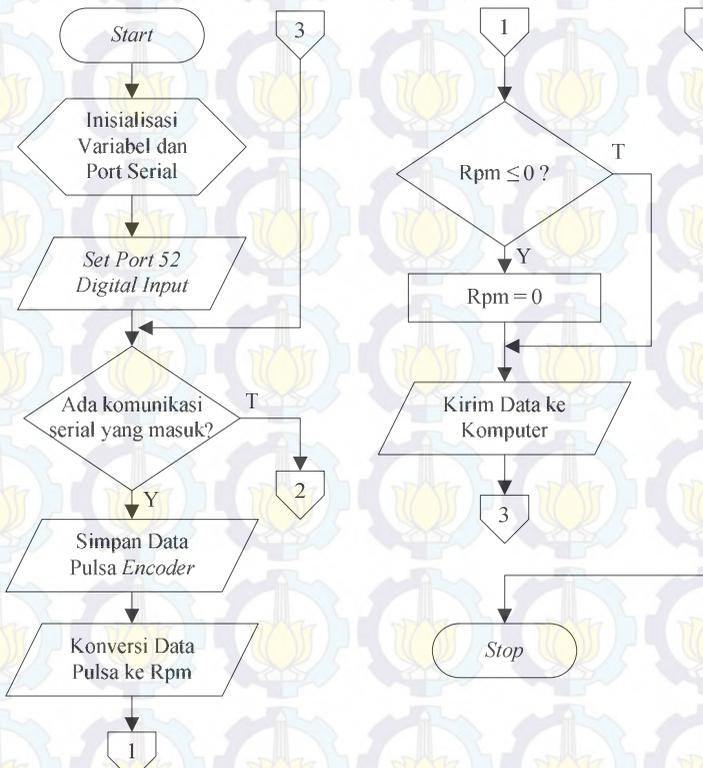
oleh arduino merupakan 6x frekuensi fasa, hal ini diperlukan karena spesifikasi *hardware* pada *driver inverter* menggunakan rangkaian pembagi fasa. *Output IC* pembagi fasa (74HC175) berupa sinyal kotak antar fasa dimana sudut antar fasa sebesar  $120^\circ$ . PWM digunakan untuk memperhalus putaran motor dan mengurangi disipasi daya yang diserap oleh BLDCM.



**Gambar 3.12** Flowchart Pemrograman Pembangkit PWM dan Frekuensi

### 3.3.2 Arduino sebagai Pembaca Data Sensor

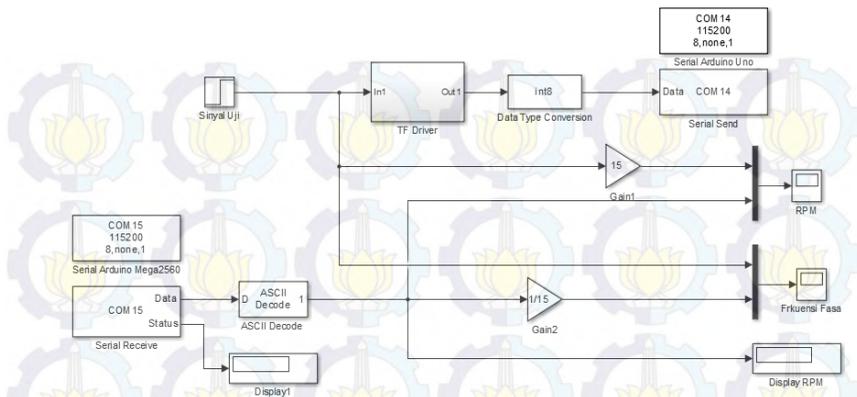
Arduino Mega2560 digunakan sebagai pembaca data sensor kecepatan. Proses pembacaan data akan dikirim ke komputer dengan satuan rpm. Berikut *flowchart* pemrogramannya ada pada Gambar 3.13.



Gambar 3.13 *Flowchart* Pembacaan Data Sensor Kecepatan

### 3.3.3 Perancangan Metode Identifikasi

Perancangan simulink menggunakan Matlab R2014a, perancangan simulink blok diagram ditujukan untuk mengidentifikasi *plant* sebelum dilakukan pengaturan pada *plant*. Koneksi antara simulink dengan arduino menggunakan blok diagram Serial sebagai pengirim dan penerima data dapat dilihat pada Gambar 3.14.



**Gambar 3.14** Blok Diagram Simulink Identifikasi

### 3.4 Identifikasi *Plant*

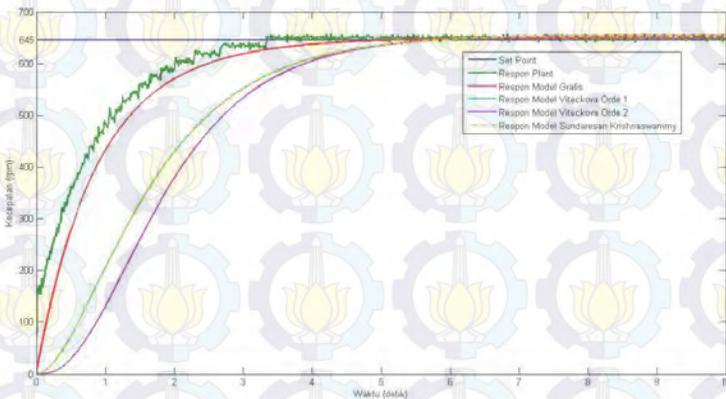
Identifikasi sistem diperlukan untuk mendapatkan model matematika dari BLDCM. Pada Tugas Akhir ini, identifikasi dilakukan adalah identifikasi statis dengan melihat keluaran respon berupa kecepatan terhadap referensi yang diberikan. Sinyal uji *step* diberikan melalui Arduino dengan member nilai *set point* sebesar 43 Hz atau sama dengan 645 rpm. Hasil respon *plant* yang terbaca oleh *rotary encoder* akan ditampilkan di simulink melalui blok diagram *Serial Receive*. Data kecepatan juga dan disimpan dalam bentuk *workspace* dengan *time sampling* 0,001 detik.

#### 3.4.1 Metode Identifikasi

Metode identifikasi diperlukan untuk mencari respon model yang paling mendekati respon *plant*. Terdapat tujuh metode identifikasi yang dapat digunakan untuk sistem tanpa osilasi dengan masukan respon *step*. Dari ketujuh metode tersebut digunakan 4 buah metode, yaitu metode Viteckova 1<sup>st</sup> Order, Viteckova 2<sup>nd</sup> Order, Sundaresan – Krishnaswamy dan metode Grafis. Keempat metode tersebut dicari RMSE yang paling kecil yang akan mendekati dengan model *plant* BLDCM. Validasi model beban minimal dapat dilihat pada Tabel 3.2. Respon model beban minimal dapat dilihat pada Gambar 3.15.

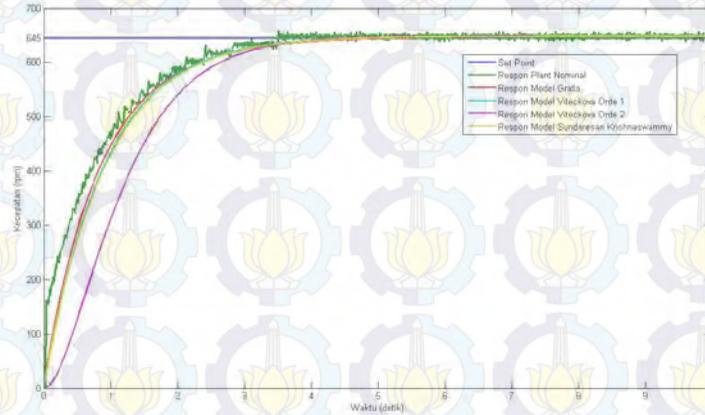
**Tabel 3.2** Validasi Model Matematika Beban Minimal

No	Metode	Model Matematika	RMSE
1.	Viteckova 1 <sup>st</sup> Order	$\frac{1,008}{0,859s + 1}$	116,138
2.	Viteckova 2 <sup>nd</sup> Order	$\frac{1,008}{0,299s^2 + 1,094s + 1}$	143,336
3.	Sundaresan & Krishnaswamy	$\frac{1,008}{0,864s + 1}$	116,527
4.	Grafis	$\frac{1,008 (0,935s + 1)}{0,817s^2 + 1,881s + 1}$	29,587

**Gambar 3.15** Respon Model Matematika Beban Minimal**Tabel 3.3** Validasi Model Matematika Beban Nominal

No	Metode	Model Matematika	RMSE
1.	Viteckova 1 <sup>st</sup> Order	$\frac{1,005}{0,9s + 1}$	24,339
2.	Viteckova 2 <sup>nd</sup> Order	$\frac{1,005}{0,383s^2 + 1,238s + 1}$	63,362
3.	Sundaresan & Krishnaswamy	$\frac{1,005}{0,918s + 1}$	25,660
4.	Grafis	$\frac{1,005 (0,881s + 1)}{0,722s^2 + 1,761s + 1}$	20,880

Pada Gambar 3.15, metode Viteckova 2<sup>nd</sup> Order memiliki simpangan terjauh, sehingga model matematika beban minimal menggunakan metode Grafis. Validasi beban nominal dapat dilihat pada Tabel 3.2 dan respon model beban nominal dapat dilihat pada Gambar 3.16.

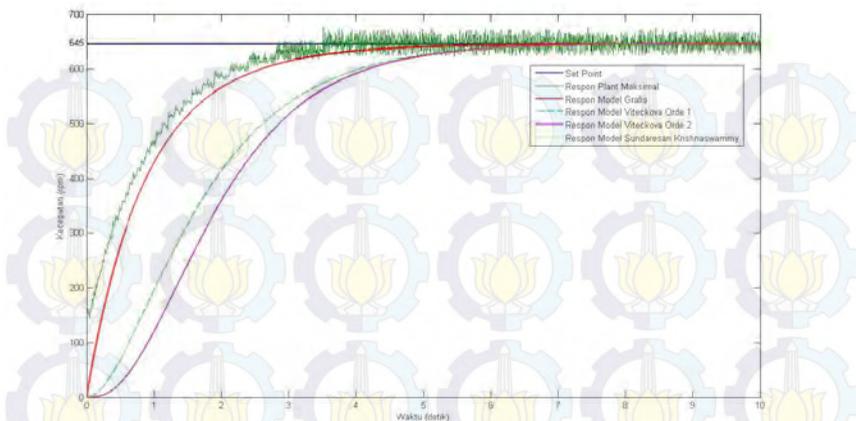


**Gambar 3.16** Respon Model Matematika Beban Nominal

Pada Gambar 3.16, respon model metode Grafis dipilih karena memiliki nilai RMSE paling rendah atau respon paling mendekati respon *plant* nominal. Validasi model matematika beban maksimal dapat dilihat pada Tabel 3.4 dan respon model matematika beban maksimal dapat dilihat pada Gambar 3.17.

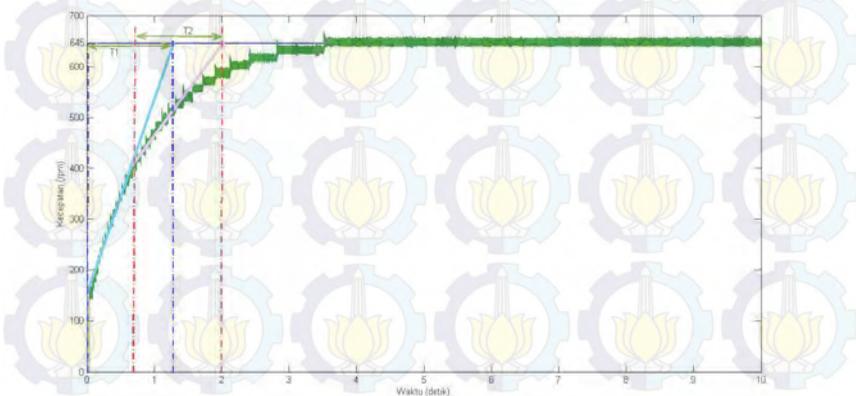
**Tabel 3.4** Validasi Model Matematika Beban Maksimal

No	Metode	Model Matematika	RMSE
1.	Viteckova 1 <sup>st</sup> Order	$\frac{1,002}{0,908s + 1}$	118,098
2.	Viteckova 2 <sup>nd</sup> Order	$\frac{1,002}{0,383s^2 + 1,238s + 1}$	146,069
3.	Sundaresan & Krishnaswamy	$\frac{1,002}{0,911s + 1}$	118,326
4.	Grafis	$\frac{1,002 (0,956s + 1)}{0,836s^2 + 1,91s + 1}$	29,950



**Gambar 3.17** Respon Model Matematika Beban Maksimal

Metode Grafis dipilih karena memiliki validasi nilai RMSE yang terkecil. Hasil respon dari pendekatan Grafis berupa respon orde 2. Hasil dari tiap perhitungan dari beberapa metode yang terdapat dalam tabel, disimulasikan pada *software* Matlab. Hal ini dilakukan untuk dapat melihat perbandingan hasil dari perhitungan dengan data hasil respon sebenarnya. Hasil grafik respon pada Matlab didapatkan bahwa pendekatan model matematika dengan metode Grafis yang paling mendekati data yang sebenarnya.



**Gambar 3.18** Penarikan Garis Singgung pada Metode Grafis

Perhitungan dalam menentukan pendekatan model matematika dengan metode Grafis adalah sebagai berikut:

- Plot hasil respon *plant*
- Tarik garis singgung yang memotong kurva respon *plant*, kemudian hitung  $\tau_{s1}$  dan  $\tau_{s2}$ , seperti Gambar 3.20
- Penurunan rumus model matematika seperti di bawah ini.

$$G(s) = K \left( \frac{1/1,17}{s+1,17} + \frac{1/0,65}{s+0,65} \right) \quad (3.1)$$

Mula-mulai gain K bernilai 1.

- Menentukan nilai gain (K)  
Gain (K) bisa didapat apabila setelah diberi sinyal uji *unit step* sesuai dengan *set point* (SP) dan melihat hasil respon melalui *scoope*. Setelah melihat hasil respon, maka nilai dari K didapat melalui rumus

$$K = Y_{ss} \left( \frac{SP}{Y_{ss}} \right) = 1289,8513 \left( \frac{648,3006}{1289,8513} \right) = 0,502 \quad (3.2)$$

- Fungsi alih keseluruhan

$$G(s) = 1,005 \left( \frac{0,881s + 1}{0,722s^2 + 1,761s + 1} \right) \quad (3.3)$$

### 3.4.2 Metode Pembebanan *Plant*

Pembebanan diperlukan untuk menentukan model matematika dari suatu *plant* dalam kondisi yang berbeda. Perubahan parameter pada *plant* dapat terjadi akibat adanya pembebanan yang diberikan pada sistem. Pada Tugas Akhir ini, rem magnetik digunakan sebagai beban pada BLDCM. Pembebanan dibagi ke dalam 3 kondisi berdasarkan besar posisi pembebanan rem magnetik, yaitu beban minimal, beban nominal atau setengah dari permukaan rem magnetik diberikan terhadap piringan motor, dan beban maksimal dengan posisi rem magnetik diberikan full terhadap piringan motor. Pengambilan data dilakukan ketika BLDCM berputar pada *setpoint* 645 rpm. Penentuan model matematika didasarkan pada tiga kondisi yaitu kondisi beban minimal,

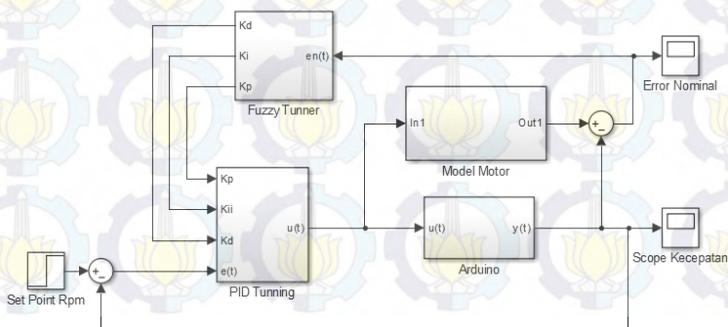
nominal, dan maksimal. Beban minimal dipilih ketika motor diberi beban 0 atau tidak diberi efek pembebanan rem magnetik menghasilkan *output steady state* sebesar 650,22 rpm. Pada kondisi beban nominal, posisi rem magnetik terbenam separuhnya pada piringan alumunium yang menghasilkan *output steady state* sebesar 648,3 rpm. Pada kondisi beban maksimal, posisi rem magnetik terbenam sepenuhnya pada piringan alumunium yang menghasilkan *output steady state* sebesar 646,514 rpm. Model matematika kondisi beban pada *plant* BLDCM dapat dilihat pada Tabel 3.5.

**Tabel 3.5** Model Matematika Beban BLDCM

No	Kondisi Pembebanan	Model Matematika
1.	Beban Minimal	$\frac{1,008 (0,935s+1)}{0,817s^2+1,881s+1}$
2.	Beban Nominal	$\frac{1,005 (0,881s+1)}{0,722s^2+1,761s+1}$
3.	Beban Maksimal	$\frac{1,002 (0,956s+1)}{0,836s^2+1,91s+1}$

### 3.5 Perancangan Kontroler PID-Fuzzy

Pembebanan nominal dianggap sebagai keadaan ketika motor bekerja pada kondisi yang seharusnya atau kerja BLDCM pada kondisi yang normal. Mekanisme kontroler PID-Fuzzy ditunjukkan pada Gambar 3.19.



**Gambar 3.19** Diagram Blok Kontroler PID-Fuzzy

Blok Pemodelan motor merupakan blok pemodelan motor dalam keadaan nominal atau *transfer function* BLDCM yang didapat saat melakukan identifikasi. Blok Arduino yang terdiri dari *Serial Send* dan *Serial Receive* digunakan sebagai *real plant* untuk bias memberikan sinyal kontrol pada arduino dimana yang nantinya dikonversi menjadi frekuensi fasa dan PWM untuk keperluan *driver* BLDCM.

### 3.5.1 Perancangan Kontroler PID

Perancangan kontroler PID dilakukan secara analitik dari parameter model matematika yang diperoleh dari hasil identifikasi. Parameter model matematik tersebut digunakan untuk menentukan nilai  $K_p$ ,  $K_i$  dan  $K_d$ . Nilai  $K_p$ ,  $K_i$ , dan  $K_d$  yang diperoleh memiliki nilai yang berbeda-beda sesuai dengan perubahan pada nilai parameter model matematika. Hasil perancangan kontroler PID pada kondisi pembebanan yang berbeda akan digunakan sebagai keluaran dari *fuzzy* yang dijadikan sebagai parameter pada kontroler PID. Persamaan model matematika *plant* secara umum diberikan sebagai berikut:

$$G(s) = \frac{K(\tau s + 1)}{\frac{1}{\omega n} s^2 + \frac{2\zeta}{\omega n} s + 1} \quad (3.4)$$

Langkah awal dalam menentukan nilai parameter PID adalah menentukan nilai  $\omega n$  dan  $\zeta$  dari Persamaan 3.4. Sebelum itu terlebih dahulu menentukan spesifikasi sistem,  $\tau_s$  (5%) sebesar 2 detik,  $e_{ss}$  sebesar 0%, dan *overshoot*  $\leq 5\%$ . Langkah selanjutnya adalah menentukan nilai  $\tau$  yang diinginkan ( $\tau^*$ ) dan dalam perancangan ini nilai  $\tau^* = 2/3$  detik. Langkah selanjutnya untuk menentukan nilai PID dapat dipilih jika  $\tau_{d2} = \tau$ ,  $\tau_i + \tau_{d2} = \frac{2\zeta}{\omega n}$ , dan  $\tau_i(\tau_{d1} + \tau_{d2}) = \frac{1}{\omega n^2}$ , maka dapat dirumuskan:

$$\tau_{d2} = \tau \quad (3.5)$$

$$\tau_{d1} = \frac{(1/\omega n)^2}{\tau_i} - \tau_{d2} \quad (3.6)$$

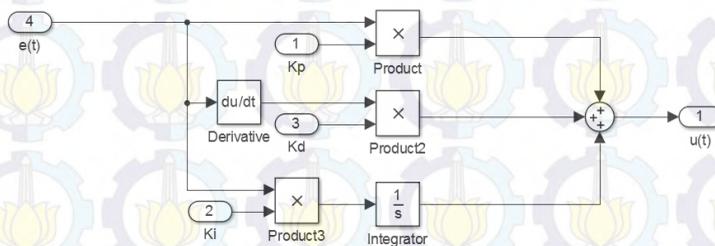
$$\tau_i = \frac{2\zeta}{\omega n} - \tau \quad (3.7)$$

$$K_p = \frac{\tau_i}{\tau * K.} \quad (3.8)$$

$$K_i = \frac{K_p}{\tau_i.} \quad (3.9)$$

$$K_d = K_p * \tau_{d1} \quad (3.10)$$

Perhitungan parameter PID yang telah dilakukan sesuai dengan persamaan di atas, pada setiap kondisi pembebanan diperoleh hasil parameter kontroler seperti yang ditunjukkan pada Tabel 3.6 dengan blok diagram pada Gambar 3.20.



**Gambar 3.20** Blok Diagram Kontroler PID

**Tabel 3.6** Parameter PID pada Kondisi Pembebanan

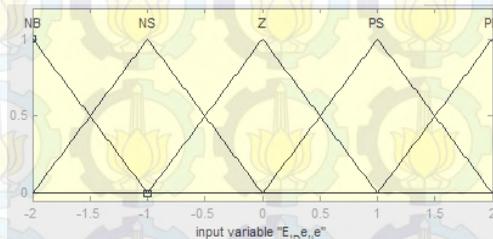
No	Kondisi Pembebanan	Kp	Ki	Kd
1.	Beban Minimal	1,313	1,488	-0,099
2.	Beban Nominal	1,407	1,492	-0,787
3.	Beban Maksimal	1,430	1,499	-0,110

### 3.5.2 Perancangan Mekanisme PID-Fuzzy

Pada dasarnya kontroler PID hanya dapat digunakan pada *plant* dengan satu fungsi alih saja, sehingga tidak dapat digunakan pada keadaan dengan fungsi alih yang berbeda. Hal ini yang menjadi dasar dalam penggunaan *fuzzy*. Data masukan yang digunakan dalam hal ini berupa *error* nominal dan data keluaran berupa nilai  $K_p$ ,  $K_i$ , dan  $K_d$ .

### 3.5.2.1 Fungsi Keanggotaan Masukan

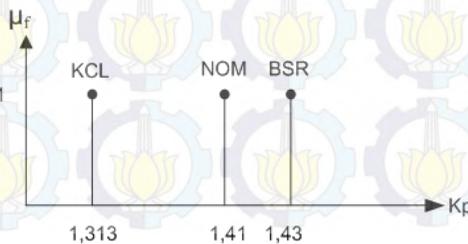
Fungsi keanggotaan masukan *fuzzy* berupa data *error* yang didapatkan dari selisih hasil *output plant* nominal dengan *plant* berbeban. Jumlah fungsi keanggotaan *fuzzy* yang digunakan pada sebanyak lima. Nilai linguistik dari fungsi keanggotaan *fuzzy* didefinisikan dengan angka -2, -1, 0, 1, dan 2. Tiap angka yang ada mewakili beberapa kondisi pembebanan dari keadaan beban 1 sampai kondisi beban 9. Fungsi keanggotaan *fuzzy* yang telah ternormalisasi dengan gain normalisasi  $\Delta error$  sebesar 2/0,13, gain normalisasi *error* sebesar 2/0,128 dan gain normalisasi *integral error* sebesar 2/1,28, sehingga dapat diskala masuk kedalam anggota fuzzifikasi Gambar 3.21.



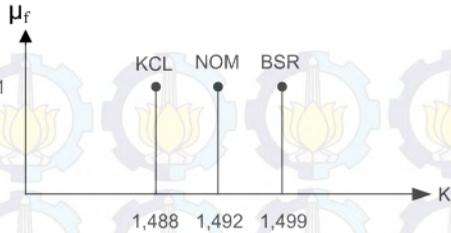
Gambar 3.21 Fungsi Keanggotaan *Fuzzy* untuk *Error*, *Delta error*, dan *Integral Error*

### 3.5.2.2 Fungsi Keanggotaan Keluaran

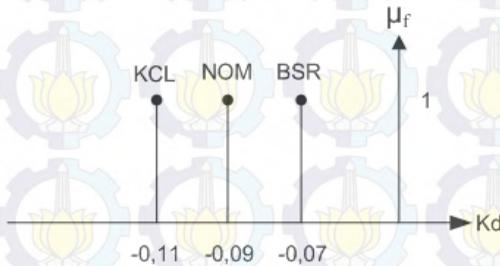
Fungsi keanggotaan keluaran *fuzzy* berupa nilai  $K_p$ ,  $K_i$ , dan  $K_d$  yang diperoleh dari perhitungan pada tiap pembebanan. Pada Tugas Akhir ini *fuzzy* yang digunakan adalah *fuzzy* Mamdani. *Fuzzy* mamdani menggunakan fungsi keanggotaan *triangular* yang terdiri dari tiga fungsi keanggotaan keluaran yaitu  $K_p$ ,  $K_i$ , dan  $K_d$ .



Gambar 3.22 Fungsi Keanggotaan  $K_p$



Gambar 3.23 Fungsi Keanggotaan  $K_i$



Gambar 3.24 Fungsi Keanggotaan  $K_d$

### 3.5.2.3 Rule Base

Perancangan *rule base* untuk kontroler *fuzzy* diperlukan agar kontroler *fuzzy* dapat bekerja pada *plant*. Berdasarkan fungsi keanggotaan masukan berupa data arus pada beban dan keluaran berupa nilai parameter kontroler PID, diperoleh beberapa *rule base* untuk  $K_p$ ,  $K_i$ , dan  $K_d$  dengan aturan linguistik sebagai berikut:

IF (Error is "NB") THEN ( $K_p$  is "KCL")

IF (Error is "NS") THEN ( $K_p$  is "KCL")

IF (Error is "Z") THEN ( $K_p$  is "NOM")

IF (Error is "PS") THEN ( $K_p$  is "BSR")

IF (Error is "PB") THEN ( $K_p$  is "BSR")

IF ( $\Sigma$ error is "NB") THEN ( $K_i$  is "KCL")

IF ( $\Sigma$ error is "NS") THEN ( $K_i$  is "KCL")

IF ( $\Sigma$ error is "Z") THEN ( $K_i$  is "NOM")

IF ( $\Sigma$ error is "PS") THEN ( $K_i$  is "BSR")

IF ( $\Sigma$ error is "PB") THEN ( $K_i$  is "BSR")

IF ( $\Delta error$  is "NB") THEN ( $K_d$  is "KCL")  
 IF ( $\Delta error$  is "NS") THEN ( $K_d$  is "KCL")  
 IF ( $\Delta error$  is "Z") THEN ( $K_d$  is "NOM")  
 IF ( $\Delta error$  is "PS") THEN ( $K_d$  is "BSR")  
 IF ( $\Delta error$  is "PB") THEN ( $K_d$  is "BSR")

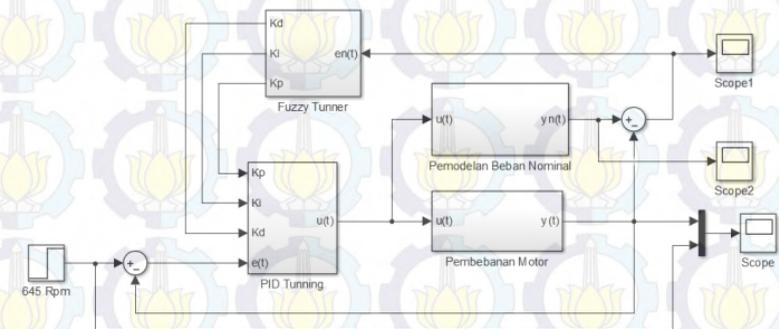
### 3.5.2.4 Defuzifikasi

Defuzifikasi digunakan untuk mengubah hasil dari inferensi *fuzzy* menjadi variabel nyata, atau diubah menjadi nilai nyata yang berupa nilai tunggal. Metode defuzifikasi yang menggunakan defuzifikasi COA (*Centre of Area*). Defuzifikasi untuk mendapatkan nilai  $K_p$ ,  $K_i$ , dan  $K_d$  ditunjukkan pada Persamaan 3.11.

$$U_0 = \frac{\sum_{k=1}^m u_k(T) * \mu_u(u_k(T))}{\sum_{k=1}^m \mu_u(u_k(T))} \quad (3.11)$$

## 3.6 Perancangan Simulasi Kontroler

Simulasi merupakan salah satu tahap yang dapat menentukan keberhasilan dari kontroler yang telah didesain sebelum kontroler diimplimentasikan pada *plant*. Pada Tugas Akhir ini, simulasi dari kontroler menggunakan simulink pada Matlab. Diagram blok simulink dari simulasi dapat dilihat pada Gambar 3.25.



Gambar 3.25 Diagram Blok Simulasi PID-Fuzzy pada Simulink

Berdasarkan Gambar 3.25 di atas, *set point* yang digunakan dalam simulasi berupa kecepatan (rpm). Hasil keluaran dari simulasi di atas berupa kecepatan (rpm).

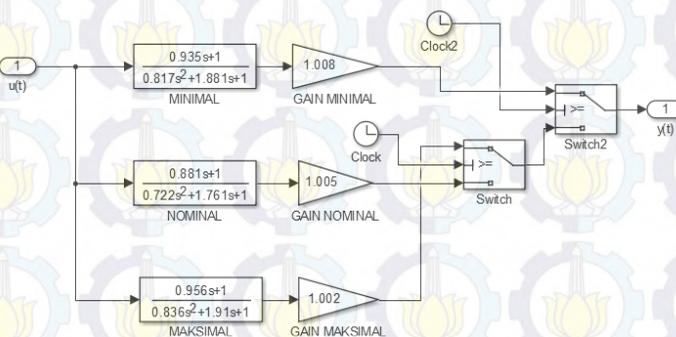
### 3.6.1 Blok Model *Plant* Beban Nominal

Blok ini berisi model pendekatan matematika dari *plant* yang nilainya didapatkan dari perhitungan yang telah dilakukan sebelumnya. Pada blok model *plant* ini berisi model fungsi alih dari BLDCM pada Persamaan 3.12:

$$\frac{1,005(0,881s+1)}{0,722s^2+1,761s+1} \quad (3.12)$$

### 3.6.2 Blok Model Pembebanan *Plant*

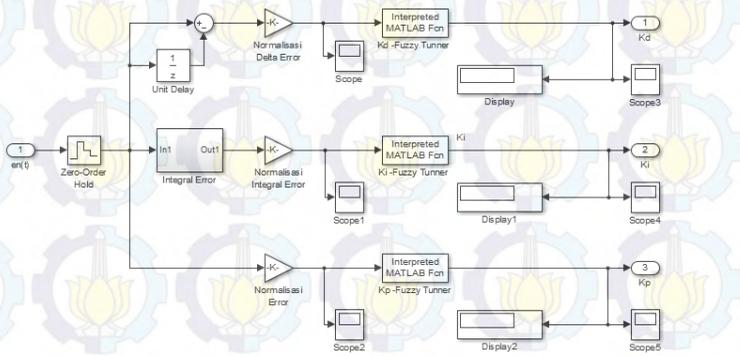
Blok ini berisi model pendekatan matematika dari *plant* yang nilainya didapatkan dari perhitungan yang telah dilakukan sebelumnya. Pada blok model *plant* ini, nilai model pendekatan matematika pada *plant* berubah sesuai dengan adanya perubahan beban. Nilai perubahan parameter model matematika *plant* pada blok ini didapatkan dari hasil perhitungan blok estimasi parameter *plant*. Model matematika pembebanan yang didapat dari hasil identifikasi akan dijadikan sebagai blok model pembebanan pada perancangan simulasi dengan selang waktu antara beban nominal, maksimal dan beban minimal selama 10 detik pada Gambar 3.26.



Gambar 3.26 Blok Model Pembebanan *Plant*

### 3.6.3 Blok *Fuzzy Tuner*

Blok *Fuzzy Tuner* digunakan untuk mengubah nilai parameter kontroler PID agar kontroler dapat beradaptasi terhadap setiap perubahan pada beban. Simulasi untuk PID-*Fuzzy* ini juga digunakan untuk mengetahui mekanisme perubahan parameter PID. Masukan dari blok ini merupakan *error nominal*. Diagram blok untuk PID-*Fuzzy* untuk parameter  $K_p$ ,  $K_i$ , dan  $K_d$  dapat dilihat pada Gambar 3.27.



Gambar 3.27 Diagram Blok *Fuzzy Tuner*



## BAB IV PENGUJIAN DAN ANALISA

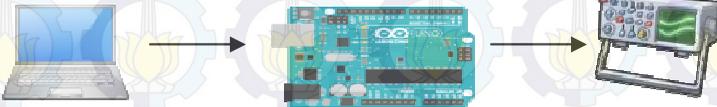
Bab IV menjelaskan tentang hasil pengujian dan analisa dari perancangan sistem yang telah dilakukan dan dijelaskan pada bab sebelumnya. Bab ini terdiri pengujian rangkaian penggerak elektronik, pengujian sensor, pengujian kontroler secara simulasi pada *Software Matlab*, dan implementasi kontroler *PID-Fuzzy*. Pengujian ini dilakukan untuk mengetahui respon keluaran dari sistem dan juga untuk mengetahui proses kerja dari kontroler yang didesain.

### 4.1 Pengujian Arduino

Pengujian pada Arduino meliputi pengujian sinyal PWM dan pengujian pembangkit sinyal fasa.

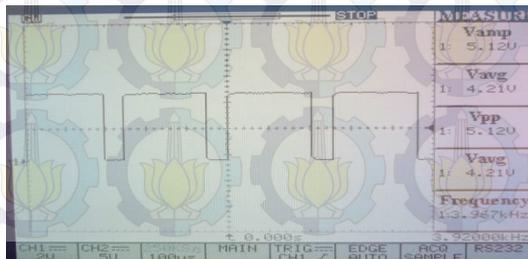
#### 4.1.1 Analisa Pengujian Pembangkit Sinyal PWM

Pengujian pembangkit sinyal PWM pada arduino dengan mekanisme yang terlihat pada Gambar 4.1.



Gambar 4.1 Mekanisme Pengujian Sinyal PWM

Sebelum dimulai pengujian, terlebih dahulu arduino di program untuk membangkitkan sinyal PWM dengan frekuensi 3,9 KHz dengan *duty cycle* 80%. Hasil pengujian dapat dilihat pada Gambar 4.2.



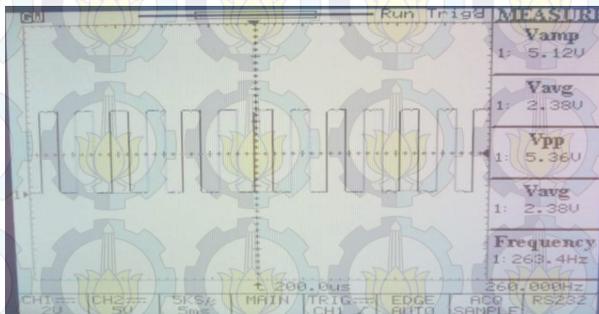
Gambar 4.2 Hasil Pengujian Sinyal PWM

#### 4.1.2 Analisa Pengujian Pembangkit Sinyal Fasa

Arduino diprogram untuk membangkitkan sinyal fasa sebesar enam kali frekuensi fasa. Hal ini dilakukan karena pada rangkaian penggerak elektronik menggunakan IC (*Integrated Circuit*) pembagi fasa 74HC175. Mekanisme pengujian sinyal pembangkit fasa oleh arduino terlihat pada Gambar 4.1. Frekuensi fasa yang dibangkitkan dengan nilai *set point* 43 Hz. Frekuensi yang akan dibangkitkan oleh arduino sebesar enam kali dari nilai frekuensi *set point*. Hasil pengujian pembangkit sinyal fasa dapat dilihat pada Gambar 4.3 dan Tabel 4.1.

**Tabel 4.1** Hasil Pengujian Pembangkit Frekuensi Fasa Arduino

Frekuensi Fasa (Hz)	Perhitungan 6x Frekuensi Fasa (Hz)	Pembangkit Frekuensi Arduino (Hz)	Error (%)
10	60	56	-6,666
20	120	119	-0,833
30	180	181	0,555
40	240	241	0,416
43	260	262	0,769
45	270	273	1,111
50	300	301	0,333
55	330	333	0,909



**Gambar 4.3** Hasil Pengujian Sinyal Frekuensi Fasa

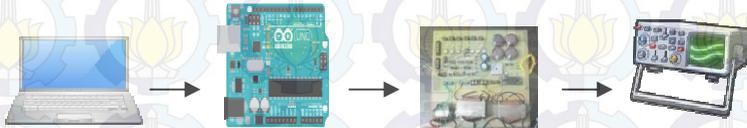
Pada Gambar 4.3, frekuensi yang dibangkitkan sebesar 263,4 Hz. Terjadi *error* sekitar 2,09% dari frekuensi seharusnya, yaitu sebesar 258 Hz.

## 4.2 Pengujian Rangkaian Penggerak Elektronik

Pengujian rangkaian penggerak elektronik meliputi pengujian keluaran gerbang AND dan NAND, pengujian keluaran *optocoupler*.

### 4.2.1 Analisa Pengujian Keluaran Gerbang AND dan NAND

Pengujian keluaran AND dan NAND dilakukan dengan melihat sinyal keluaran. Pengujian ini dilakukan untuk melihat bentuk sinyal keluaran dari gerbang AND dan NAND yang berbentuk kotak dengan PWM. Sinyal ini berfungsi untuk menggerakkan *optocoupler* sebelum masuk ke rangkaian *switching*. Mekanisme pengujian dapat dilihat pada Gambar 4.4 dan hasilnya dapat dilihat pada Gambar 4.5.



Gambar 4.4 Mekanisme Pengujian Sinyal gerbang AND dan NAND



Gambar 4.5 Sinyal Keluaran AND (kiri) dan NAND (kanan)

### 4.2.2 Analisa Pengujian Keluaran *Optocoupler*

Pengujian keluaran *optocoupler* digunakan untuk melihat sinyal yang dihasilkan oleh *optocoupler* yang akan digunakan oleh rangkaian *switching*. Keluaran sinyal *optocoupler* bergantung dari jenis masukannya. *Optocoupler* ini untuk memisahkan antara *ground* kontrol dengan *ground* BLDCM untuk menghindari terjadinya hubung singkat apabila salah satu terjadi gangguan. Mekanisme pengujian sinyal *optocoupler* dapat dilihat pada Gambar 4.4. Hasil sinyal keluaran *optocoupler* dapat dilihat pada Gambar 4.6.



**Gambar 4.6** Sinyal Keluaran *Optocoupler HIGH* (kiri) dan *LOW* (kanan)

Pada Gambar 4.6, sinyal keluaran serupa dengan keluaran gerbang AND dan NAND. Sinyal bagian *HIGH* digunakan untuk menggerakkan MOSFET bagian atas yang menghasilkan sinyal sinus bagian positif, sedangkan sinyal bagian *LOW* digunakan untuk menggerakkan MOSFET bagian bawah yang menghasilkan sinyal sinus bagian negatif. Sinyal tersebut merupakan sinyal frekuensi fasa yang mengandung sinyal PWM di dalamnya.

Pengujian ini dilakukan untuk mengetahui dan mengkalibrasi kecepatan hasil dari pembacaan sensor *rotary encoder* BLDCM dengan kecepatan hasil pembacaan *tachometer*. Pengujian dilakukan dengan menghubungkan BLDCM dengan sumber tiga fasa dari rangkaian penggerak elektronik. Piringan sensor *rotary encoder* dipasang pada *shaft* motor yang nantinya piringan akan dibaca oleh *optocoupler* tipe U. Pulsa hasil pembacaan *rotary encoder* dikirim ke arduino Mega2560 untuk diolah sebelum dikirim ke komputer. Data dari arduino Mega2560 kemudian dikirim dan hasilnya ditampilkan pada *display* yang terdapat pada komputer menggunakan *serial port*. *Tachometer* digunakan untuk melihat perbandingan kecepatan dari piringan dengan hasil pengolah arduino. Mekanisme pengujian kecepatan BLDCM dapat dilihat pada Gambar 4.7. Hasil pengujian dapat dilihat pada Tabel 4.2.



**Gambar 4.7** Mekanisme Pengujian Kecepatan BLDCM

**Tabel 4.2** Hasil Pengujian Kecepatan BLDCM

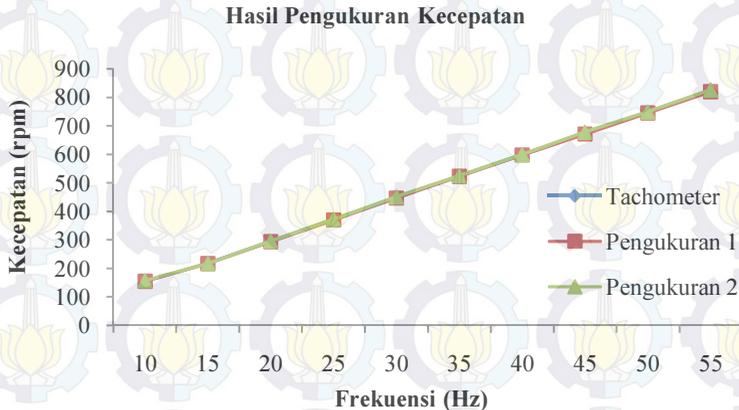
Frekuensi Fasa (Hz)	Kecepatan (Rpm)		
	Tachometer	Pengukuran 1	Pengukuran 2
10	155,5	155	157
15	218,5	217	217
20	295	294	296
25	372,5	370	371,5
30	447	447	451
35	524,5	523	525
40	600	598	599
45	673	671,4	678
50	748,5	745	749
55	826	820	826

Dari Tabel 4.2, dapat dibuat grafik hasil pengukuran yang dapat dilihat pada Gambar 4.8.

Berdasarkan data yang didapatkan pada tabel, hasil pembacaan kecepatan motor dengan *tachometer* dan perhitungan memiliki selisih yang kecil. Untuk mencari presentase *error* rata-rata digunakan rumus sebagai berikut.

$$error = \frac{(pembacaan\ sensor - Pembacaan\ Tachometer)}{Pembacaan\ Tachometer} \times 100\%$$

Sehingga didapatkan eror rata-rata dari semua pembacaan sebesar 0,109%.

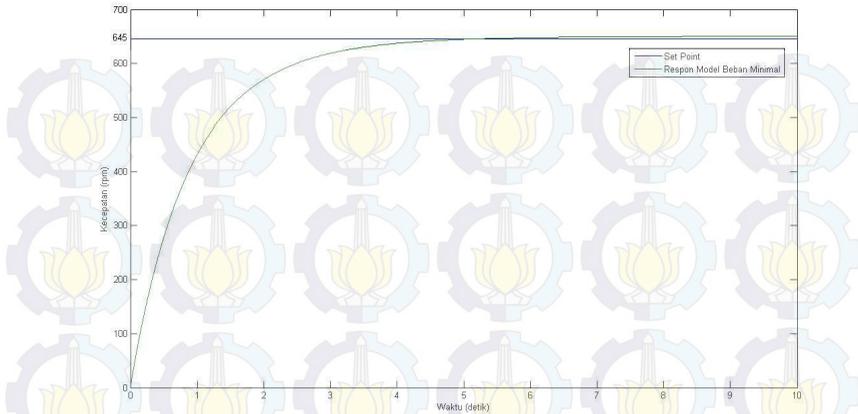


**Gambar 4.8** Hasil Pengujian Kecepatan BLDCM

### 4.3 Analisa Pengujian Simulasi tanpa Kontroler

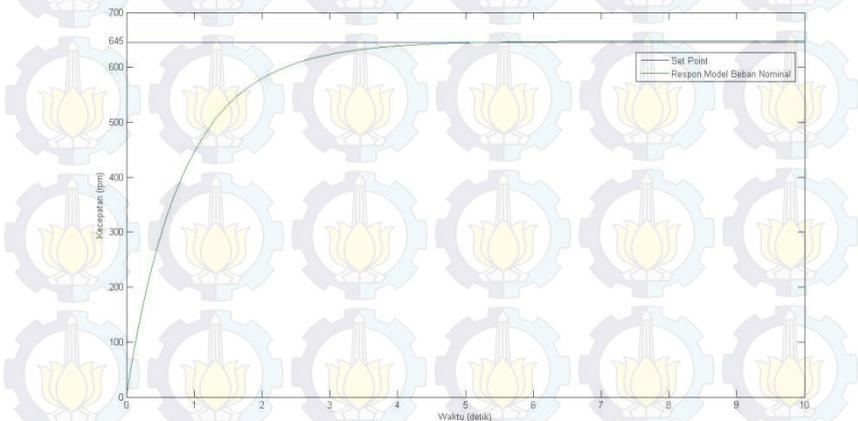
Pada tahap ini pengujian dilakukan untuk melihat respon *plant* secara *open loop* tanpa menggunakan kontroler tiap beban. Sinyal uji berupa sinyal *unit step* dengan *set point* sebesar 645 rpm. Pengujian ini dilakukan untuk melihat respon transien hasil identifikasi yang telah didapat. Pengujian ini berlangsung selama 10 detik. Respon beban minimal dapat dilihat pada Gambar 4.9.

Pada Gambar 4.9 dapat diketahui spesifikasi respon transient dengan nilai *time constant* 0,909 detik, *settling time* 2,950 detik, *rise time* 2,676 detik, *delay time* 0,622 detik, dan *error steady state* sebesar 0,744%.



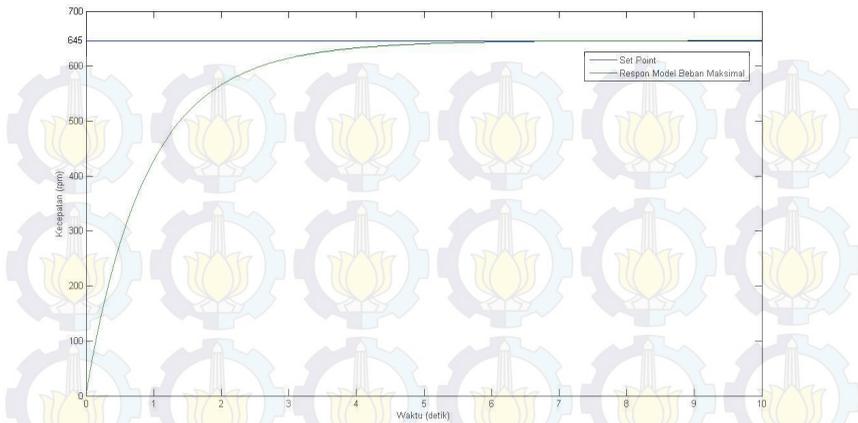
**Gambar 4.9** Hasil Pengujian Tanpa Kontroler untuk Beban Minimal

Respon hasil pengujian beban nominal tanpa menggunakan kontroler dapat dilihat pada Gambar 4.10.



**Gambar 4.10** Hasil Pengujian Tanpa Kontroler untuk Beban Nominal

Pada Gambar 4.10 dapat diketahui spesifikasi respon transient dengan nilai *time constant* 0,849 detik, *settling time* 2,736 detik, *rise time* 2,502 detik, *delay time* 0,582 detik, dan *error steady state* sebesar 0,515%. Respon hasil pengujian beban maksimal terlihat pada Gambar 4.11.



**Gambar 4.11** Hasil Pengujian Tanpa Kontroler untuk Beban Maksimal

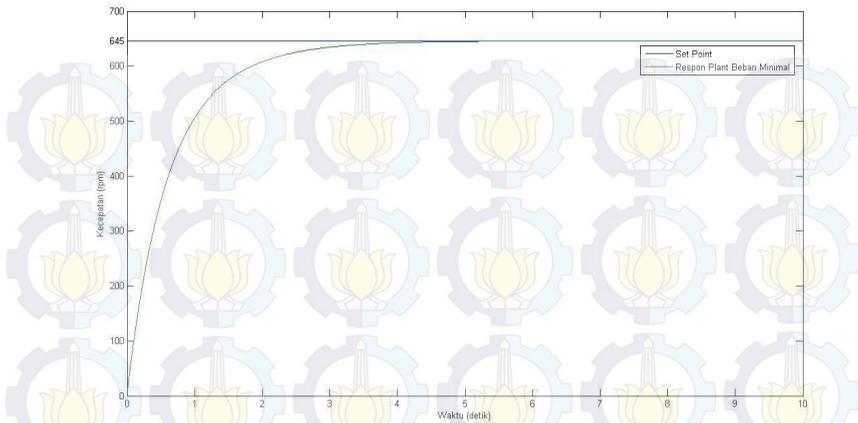
Pada Gambar 4.11 dapat diketahui spesifikasi respon transient dengan nilai *time constant* 0,913 detik, *settling time* 2,987 detik, *rise time* 2,688 detik, *delay time* 0,624 detik, dan *error steady state* sebesar 0,018%.

**Tabel 4.3** Spesifikasi Respon Tiap Beban

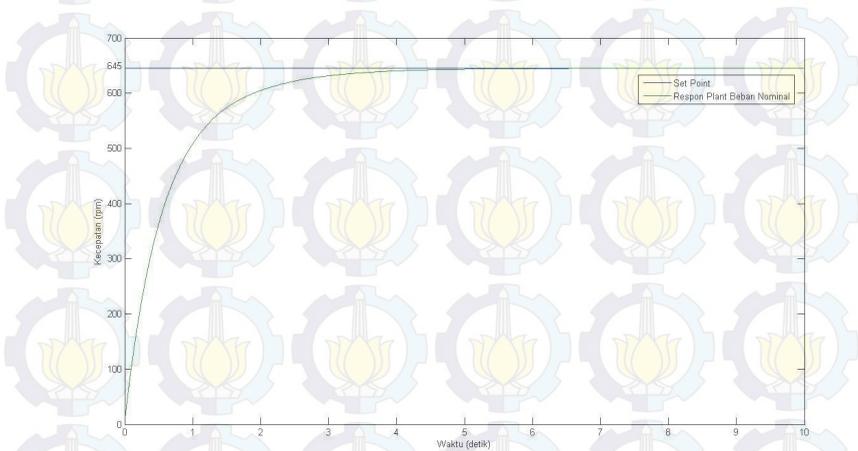
Spesifikasi	Beban Minimal	Beban Nominal	Beban Maksimal	Rata-rata
$\tau$	0,909 detik	0,849 detik	0,913 detik	0,890 detik
$t_s$ 5%	2,950 detik	2,736 detik	2,987 detik	2,891 detik
$t_r$ (5% – 95%)	2,676 detik	2,502 detik	2,688 detik	2,622 detik
$t_d$	0,622 detik	0,582 detik	0,624 detik	0,609 detik
$e_{ss}$	0,744%	0,515%	0,018%	0,425%

#### 4.4 Analisa Pengujian Simulasi dengan Kontroler PID-Fuzzy

Tahap ini dilakukan untuk mengetahui dan menguji proses kerja dari kontroler yang telah didesain. Pengujian dilakukan dengan Simulink yang terdapat dalam *software* Matlab. Pada Tugas Akhir ini dilakukan pengujian untuk PID-Fuzzy. Metode pembebanan diberikan dalam selang waktu 10 detik untuk tiap beban. Respon beban minimal dapat dilihat pada Gambar 4.12. Respon beban nominal dapat dilihat pada Gambar 4.13. Respon beban maksimal dapat dilihat pada Gambar 4.15. Respon beban berubah-ubah dapat dilihat pada Gambar 4.15.



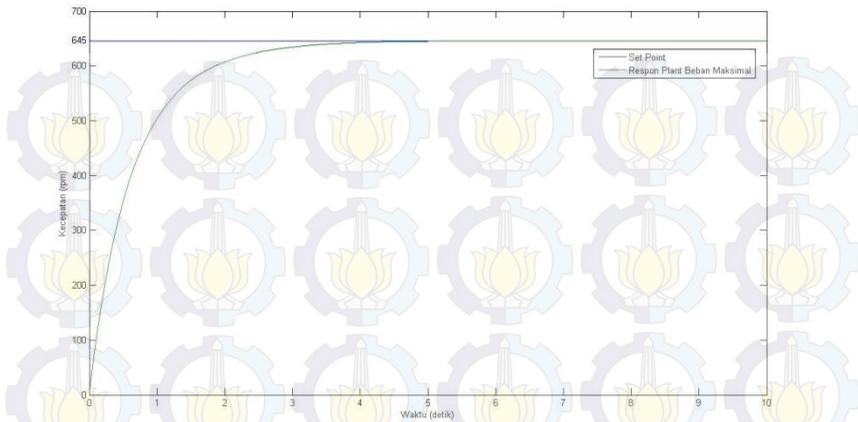
**Gambar 4.12** Respon Keluaran Beban Minimal dengan Kontroler PID-Fuzzy



**Gambar 4.13** Respon Keluaran Beban Nominal dengan Kontroler PID-Fuzzy

Pada Gambar 4.12 dapat diketahui spesifikasi respon transient dengan nilai *time constant* 0,631 detik, *settling time* 2,133 detik, *rise time* 1,866 detik, *delay time* 0,417 detik, dan *error steady state* sebesar 0,011%.

Pada Gambar 4.13 dapat diketahui spesifikasi respon transient dengan nilai *time constant* 0,634 detik, *settling time* 2,194 detik, *rise time* 1,920 detik, *delay time* 0,439 detik, dan *error steady state* sebesar 0,002%.

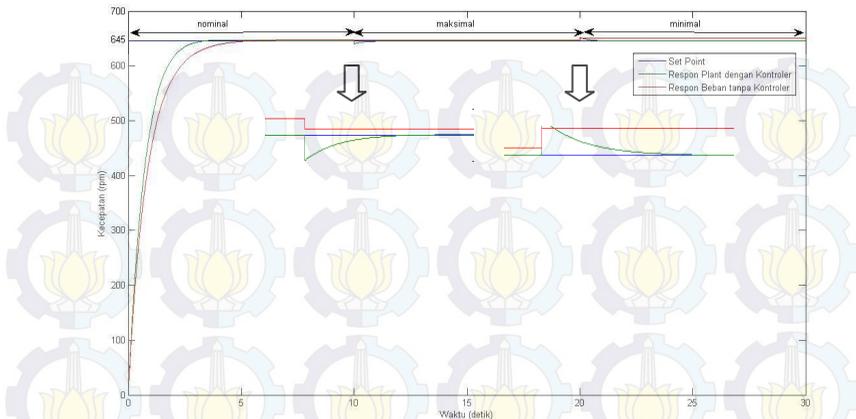


**Gambar 4.14** Respon Keluaran Beban Maksimal dengan Kontroler PID-*Fuzzy*

Pada Gambar 4.14 dapat diketahui spesifikasi respon transient dengan nilai *time constant* 0,639 detik, *settling time* 2,255 detik, *rise time* 1,950 detik, *delay time* 0,436 detik, dan *error steady state* sebesar 0,007%.

**Tabel 4.4** Spesifikasi Respon Tiap Beban dengan Kontroler PID-*Fuzzy*

Spesifikasi	Beban Minimal	Beban Nominal	Beban Maksimal	Rata-rata
$\tau$	0,631 detik	0,634 detik	0,639 detik	0,634 detik
$t_s$ 5%	2,133 detik	2,194 detik	2,255 detik	2,194 detik
$t_r$ (5% – 95%)	1,866 detik	1,920 detik	1,950 detik	1,912 detik
$t_d$	0,417 detik	0,439 detik	0,436 detik	0,430 detik
$e_{ss}$	0,011%	0,002%	0,007%	0,006%



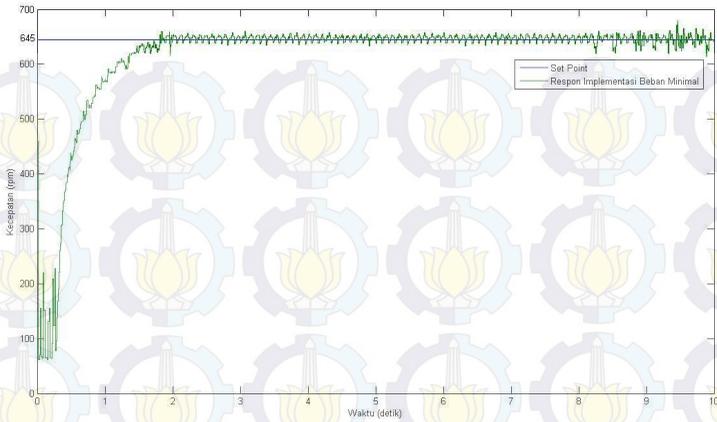
**Gambar 4.15** Respon Keluaran dengan Kontroler PID-*Fuzzy* Beban Berubah

Pada Gambar 4.15 terlihat saat terjadi perubahan beban nominal ke maksimal terjadi penurunan kecepatan kemudian kembali ke keadaan *steady state* dalam waktu 1,5 detik.

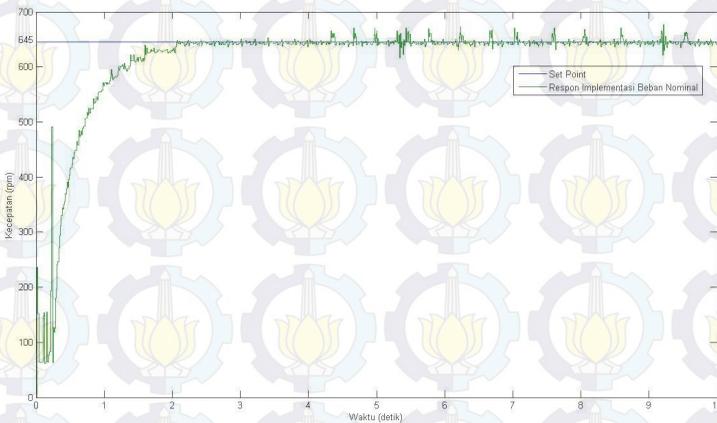
#### 4.5 Analisa Implementasi Pengujian Kontroler PID-*Fuzzy*

Pada implementasi kontroler PID-*Fuzzy* menggunakan beban rem magnetik. Pada beban minimal, piringan rotor tidak diberi beban rem magnetik. Pada beban nominal, piringan rotor diberi beban rem magnetic sebesar  $\frac{1}{2}$  luas rem magnetik. Pada beban maksimal, piringan rotor diberi beban magnetik sepenuhnya dari luas penampang rem magnetik. Implementasi pada beban minimal dapat dilihat pada Gambar 4.16.

Pada Gambar 4.16, hasil respon transien didapat dengan spesifikasi *time constant* sebesar 0,500 detik, *settling time* sebesar 1,640 detik, *delay time* sebesar 0,360 detik, *rise time* sebesar 1,540 detik, dan *error steady state* sebesar 0,254%. *Plant* BLDCM baru berfungsi saat frekuensi sistem yang diberikan mencapai 10 Hz atau setara 150 rpm. Hasil respon implementasi beban nominal dapat dilihat pada Gambar 4.17.

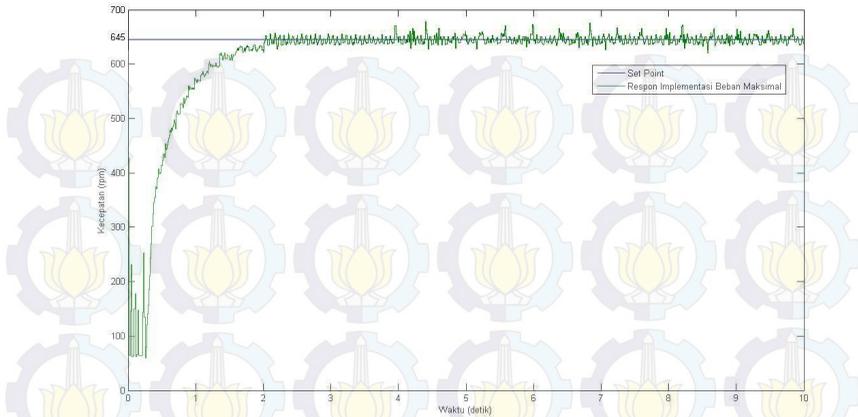


**Gambar 4.16** Respon Keluaran dengan Kontroler PID-Fuzzy Beban Minimal



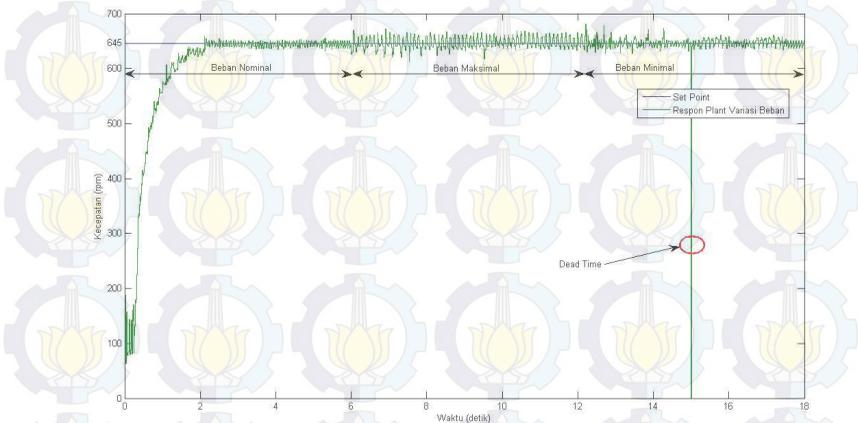
**Gambar 4.17** Respon Keluaran dengan Kontroler PID-Fuzzy Beban Nominal

Pada Gambar 4.17 didapat hasil respon transient dengan spesifikasi Hasil repon transien didapat dengan spesifikasi *time constant* sebesar 0,510 detik, *settling time* sebesar 1,680 detik, *delay time* sebesar 0,370 detik, *rise time* sebesar 1,550 detik, dan *error steady state* -0,200%. Nilai  $e_{ss}$  bernilai negatif dikarenakan nilai *steady state* berada di bawah nilai *set point*.



**Gambar 4.18** Respon Keluaran dengan Kontroler PID-*Fuzzy* Beban Maksimal

Pada Gambar 4.18, *time constant* sebesar 0,550 detik, *settling time* sebesar 1,700 detik, *delay time* sebesar 0,370 detik, *rise time* sebesar 1,570 detik, dan *error steady state* sebesar -0,100%.



**Gambar 4.19** Respon Keluaran dengan Kontroler PID-*Fuzzy* Beban Nominal-Maksimal-Minimal

Pada Gambar 4.19, mula-mula BLDCM dijalankan dalam keadaan beban nominal selama waktu 6 detik. Setelah itu, BLDCM diberi beban

rem magnetik maksimal selama 6 detik kemudian beban rem magnetik dilepas. Terlihat sekitar detik ke-15 terdapat respon *plant* jatuh ke 0 rpm, hal ini disebabkan adanya arduino tidak dapat merespon (*dead time*) atau memberikan nilai ukur ke Matlab. *Dead time* ini terjadi secara acak. Respon transient yang didapat, yaitu *time constant* ( $\tau$ ) sebesar 0,520 detik, *settling time* ( $t_s(5\%)$ ) sebesar 1,670 detik, *delay time* ( $t_d$ ) sebesar 0,370 detik, *rise time* ( $t_r(5\%-95\%)$ ) sebesar 1,490 detik, dan *error steady state* sebesar 0,099%. Terjadi *overshoot* sebesar 5,426% saat terjadi perubahan beban maksimal ke beban minimal. Spesifikasi hasil implementasi tiap beban dapat dilihat pada Tabel 4.5.

**Tabel 4.5** Spesifikasi Implementasi Respon Tiap Beban

Spesifikasi	Beban Minimal	Beban Nominal	Beban Maksimal	Rata-rata
$\tau$	0,500 detik	0,510 detik	0,550 detik	0,520 detik
$t_s$ 5%	1,640 detik	1,680 detik	1,700 detik	1,673 detik
$t_r$ (5% – 95%)	1,540 detik	1,550 detik	1,570 detik	1,553 detik
$t_d$	0,360 detik	0,370 detik	0,370 detik	0,366 detik
$e_{ss}$	0,254%	-0,200%	-0,100%	-0,015%

## DAFTAR PUSTAKA

- [1] Rasyid, M., “*Power Electronics Handbook*”, University Of West Florida, U.S.A.
- [2] Shiddiq Abdul Rachman, A., “Rancang Bangun Prototipe Sistem Aktual Kendali Sirip menggunakan Brushless DC”, *Skripsi*, Teknik Elektro-Universitas Indonesia, Jakarta, 2012.
- [3] Candra Wardianto, Ovi., “Kontrol Fuzzy Adaptif Gain Scheduling untuk Pengaturan Motor Induksi 3 Fasa”, *Tugas Akhir*, JTE-ITS, Surabaya, 2013.
- [4] Putri Suryaditya, Nindita., “Pengaturan Proses Face Milling pada Mesin Computer Numerical Control (CNC) dengan Kontroler Fuzzy-PID”, *Tugas Akhir*, JTE-ITS, Surabaya, 2013.
- [5] Matra Alfian, Rizki., “Perancangan dan Implementasi Pengaturan Kecepatan Motor Induksi Tiga Fasa menggunakan Kontroler PID Fuzzy dengan PLC”, *Tugas Akhir*, JTE-ITS, Surabaya, 2013.
- [6] Yus Peristiaferi, Trio., “Pengaturan Kecepatan Motor Tiga Fasa Pada Mesin Sentrifugal menggunakan Metode PID Fuzzy”, *Tugas Akhir*, JTE-ITS, Surabaya, 2015.
- [7] Fahmizal, “*Arduino Devices – Mikrokontroler Berbasis AVR ATmega Series*” <URL: <https://fahmizaleeits.wordpress.com/tag/arduino-adalah/>>, Februari, 2015.



## BAB V PENUTUP

Bab V menjelaskan tentang kesimpulan dan hasil dari simulasi dan implementasi secara keseluruhan.

### 5.1 Kesimpulan

Dari hasil desain kontrol PID-*Fuzzy* yang telah dikerjakan dan beberapa penerapan dalam tugas akhir ini dapat diambil beberapa kesimpulan sebagai berikut:

1. Pada simulasi kontroler PID-*Fuzzy* mampu memperbaiki performasi sistem pada tiap kondisi pembebanan dan hasil respon mendekati respon beban nominal dengan spesifikasi rata-rata  $\tau$  sebesar 0,634 detik,  $t_s(5\%)$  sebesar 2,194 detik,  $t_d$  sebesar 0,430 detik,  $t_r(5\%-95\%)$  sebesar 1,912 detik, dan  $e_{ss}$  sebesar 0,006%.
2. Pada implementasi kontroler PID-*Fuzzy* mampu memperbaiki performasi sistem pada tiap kondisi pembebanan dan hasil respon mendekati respon beban nominal dengan spesifikasi rata-rata  $\tau$  sebesar 0,520 detik,  $t_s(5\%)$  sebesar 1,673 detik,  $t_d$  sebesar 0,366 detik,  $t_r(5\%-95\%)$  sebesar 1,553 detik, dan  $e_{ss}$  sebesar -0,015%.
3. Pada implementasi beban berubah-ubah setiap 6 detik, kontroler PID-*Fuzzy* mampu mengembalikan respon *plant* ke keadaan *steady state* dengan spesifikasi  $\tau$  sebesar 0,520 detik,  $t_s(5\%)$  sebesar 1,670 detik,  $t_d$  sebesar 0,370 detik,  $t_r(5\%-95\%)$  sebesar 1,490 detik, dan  $e_{ss}$  sebesar 0,099%. Terjadi *overshoot* sebesar 5,426% saat terjadi perubahan beban.

### 5.2 Saran

1. Untuk kelanjutan riset mendatang, diharapkan adanya modifikasi dari *plant* BLDCM.
2. Dalam melakukan identifikasi sebaiknya menggunakan beberapa metode identifikasi dan berbagai macam metode perhitungan *error* hasil identifikasi.
3. Pembuatan rangkaian pembangkit sinyal frekuensi fasa, sehingga rentang frekuensi yang dibangkitkan bisa lebih lebar.





## 2. Program Arduino Pembangkit Frekuensi Fasa dan PWM

```
int bacadata = 0;
```

```
void setup() {  
  // put your setup code here, to run once:  
  // PWM 4 Khz  
  TCCR1B = TCCR1B & B11111000 | B00000010;  
  pinMode(12, OUTPUT); //pin frekuensi 74175  
  Serial.begin(115200); //set baud rate  
}
```

```
void loop() {  
  // put your main code here, to run repeatedly:  
  // cek komunikasi serial  
  if (Serial.available() > 0) {  
    // baca data serial  
    bacadata = (Serial.read());  
    // pembangkit frekuensi fasa  
    digitalWrite(12, HIGH);  
    delayMicroseconds(79000 / bacadata);  
    digitalWrite(12, LOW);  
    delayMicroseconds(79000 / bacadata);  
    //output pwm duty cycle 80%  
    analogWrite(10, 220);  
  }  
  if (Serial.available() < 1) {  
    analogWrite(10, 0);  
  }  
}
```

## 3. Program Arduino sebagai Pembaca Sensor Kecepatan

```
int duration = 0;
```

```
int rpm = 0;
```

```
char bufer[4];
```

```
void setup() {  
  // put your setup code here, to run once:  
  pinMode(52, INPUT); //encoder pin
```

```

Serial.begin(115200);
}

void loop() {
// put your main code here, to run repeatedly:
duration = pulseIn(52, HIGH, 100000);
rpm = (5500000 / 3) / duration;
if (rpm < 0) {
rpm = 0;
}
if (rpm > 1000) {
rpm = 0;
}
//delay(100);
sprintf(bufer, "%4d", rpm);
Serial.print(bufer);
delay(100);
}

```

#### 4. Program m-files fusikp

```

function kp=fusikp(x)
%fuzzifikasi Kp with 5 membership triangular
function
xf=[0 0 0 0 0]';
if x<-2
xf(1)=1;
elseif x<-1
xf(1)=-1-x;
xf(2)=x-(-2);
elseif x<0
xf(2)=0-x;
xf(3)=x-(-1);
elseif x<1
xf(3)=1-x;
xf(4)=x-0;
elseif x<2
xf(4)=2-x;
xf(5)=x-1;
else

```

```

        xf(5)=1;
end

%rule base kp
uf=[0 0 0]';
ukp=[0 0 0]';
uf(1)=max(xf(1),xf(2));
uf(2)=xf(3);
uf(3)=max(xf(4),xf(5));
jkp=uf(1)+uf(2)+uf(3);
ukp(1)=uf(1)/jkp;
ukp(2)=uf(2)/jkp;
ukp(3)=uf(3)/jkp;

%defuzzifikasi kp
%atas=1.3275*uf(1)+1.3635*uf(2)+1.475*uf(3);
%atas=1.313*uf(1)+1.41*uf(2)+1.42*uf(3);%new
ident
atas=1.313*ukp(1)+1.41*ukp(2)+1.42*ukp(3);
bawah=ukp(1)+ukp(2)+ukp(3);
kp=atas/bawah;

```

## 5. Program m-files fusiti

```

function ki=fusiti(x)
%fuzzifikasi Ti with 5 membership triangular
function
xf=[0 0 0 0 0]';
if x<-2
    xf(1)=1;
elseif x<-1
    xf(1)=-1-x;
    xf(2)=x-(-2);
elseif x<0
    xf(2)=0-x;
    xf(3)=x-(-1);
elseif x<1
    xf(3)=1-x;
    xf(4)=x-0;
elseif x<2

```

```

        xf(4)=2-x;
        xf(5)=x-1;
    else
        xf(5)=1;
    end

    %rule base ki
    uf=[0 0 0]';
    uki=[0 0 0]';
    uf(1)=max(xf(1),xf(2));
    uf(2)=xf(3);
    uf(3)=max(xf(4),xf(5));
    jki=uf(1)+uf(2)+uf(3);
    uki(1)=uf(1)/jki;
    uki(2)=uf(2)/jki;
    uki(3)=uf(3)/jki;

    %defuzzifikasi Ti
    %atas=1.25*uf(1)+1.5*uf(2)+2.169*uf(3);
    atas=1.488*uki(1)+1.492*uki(2)+1.499*uki(3);
    %new ident
    %atas=1.488*uf(1)+1.1*uf(2)+1.499*uf(3);
    bawah=uki(1)+uki(2)+uki(3);
    ki=atas/bawah;

```

## 6. Program m-files fusitd

```

function kd=fusitd(x)
%fuzzifikasi Td with 5 membership triangular
function
xf=[0 0 0 0 0]';
if x<-2
    xf(1)=1;
elseif x<-1
    xf(1)=-1-x;
    xf(2)=x-(-2);
elseif x<0
    xf(2)=0-x;
    xf(3)=x-(-1);

```

```

elseif x<1
    xf(3)=1-x;
    xf(4)=x-0;
    elseif x<2
        xf(4)=2-x;
        xf(5)=x-1;
    else
        xf(5)=1;
    end

%rule base Td
uf=[0 0 0]';
ukd=[0 0 0]';
uf(1)=max(xf(1),xf(2));
uf(2)=xf(3);
uf(3)=max(xf(4),xf(5));
jkd=uf(1)+uf(2)+uf(3);
ukd(1)=uf(1)/jkd;
ukd(2)=uf(2)/jkd;
ukd(3)=uf(3)/jkd;

%defuzzifikasi Td
%atas=-0.332*uf(1)+0*uf(2)+0.537*uf(3);
atas=-0.11*ukd(1)+(-0.09*ukd(2))+(-
0.07*ukd(3)); %new ident
bawah=ukd(1)+ukd(2)+ukd(3);
kd=atas/bawah;

```



*--Halaman ini sengaja dikosongkan--*

# **DESIGN ELECTRONIC DRIVE AND FUZZY PID CONTROLLER IN SPEED CONTROL OF BRUSHLESS DC (BLDC) MOTOR**

**Name** : Suwondo Saputra  
**Supervisor** : Ir. Ali Fatoni, M.T.

## **ABSTRACT**

*Along with industry development, the need of electric motors that have high efficiency, high torque, and inexpensive treatment is increasing. DC motor and induction motor are not able to meet those need. BLDCM is able to meet those need. BLDCM is a 3 phase synchronous motor but still requires DC voltage as the source where it was converted into 3 phase AC voltage using inverter.*

*Beside the benefit of BLDCM evidently speed control method still difficult. Decreasing speed of motor caused load change has become a common thing. Using Fuzzy PID controller become a method that can solve the problem. A simple PID algorithm and fuzzy logic to tune PID parameters can be adapted with load changes.*

*Result of implementation of Fuzzy PID controller can be back output of plant to steady state condition if load changes was happened. Result of response approach to 645 rpm of set point, 0.520 seconds of time constant, 1.490 seconds of rise time, 1.670 seconds of settling time, 0.370 of delay time, 0.099% of error steady state, and 5.426% of overshoot when load is happened.*

**Keywords** : BLDCM, Fuzzy Logic, Fuzzy PID, Inverter, PID



## KATA PENGANTAR

Segala puji kehadiran Tuhan Yang Maha Esa yang telah melimpahkan rahmat dan hidayah-Nya kepada kami dalam usaha dan kerja kami sehingga dapat menyelesaikan Tugas Akhir dengan judul :

### **PERANCANGAN PENGGERAK ELEKTRONIK DAN KONTROLER PID-*FUZZY* PADA PENGATURAN KECEPATAN *BRUSHLESS* DC (BLDC) MOTOR**

Tugas Akhir ini disusun guna memenuhi persyaratan untuk menyelesaikan studi strata 1 di jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember, Surabaya.

Dalam penyusunan laporan Tugas Akhir ini, kami banyak mendapatkan bantuan dan dukungan dari berbagai pihak. Oleh karena itu, penulis dengan tulus ikhlas menyampaikan banyak terima kasih kepada:

1. Bapak Ir. Ali Fatoni, M.T. selaku dosen pembimbing.
2. Ayah, Ibu, Adek dari penulis yang telah membantu baik dari segi materi, semangat maupun motivasinya.
3. Umi Nurhidayati yang telah banyak memberikan *support* dan motivasi.
4. Saudara/i Kurniawan Khoiruddin, Marika Ayu PR., sebagai partner TA.
5. Teman-teman seperjuangan Teknik Sistem Pengaturan.
6. Dan semua pihak yang telah membantu terselesaikannya buku Tugas Akhir ini.

Penulis menyadari dan memohon maaf atas segala kekurangan pada Tugas Akhir ini dikarenakan keterbatasan kemampuan penulis. Penulis mengharapkan berbagai masukan dan kritik dari berbagai pihak agar lebih baik di masa mendatang. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat dalam ilmu pengetahuan demi kemajuan Indonesia menuju negeri yang berteknologi mandiri.

Surabaya, Juni 2015

Penyusun



*--Halaman ini sengaja dikosongkan--*

# DAFTAR ISI

<b>HALAMAN JUDUL</b> .....	<b>i</b>
<b>PERNYATAAN KEASLIAN TUGAS AKHIR</b> .....	<b>v</b>
<b>HALAMAN PENGESAHAN</b> .....	<b>vii</b>
<b>ABSTRAK</b> .....	<b>ix</b>
<b>ABSTRACT</b> .....	<b>xi</b>
<b>KATA PENGANTAR</b> .....	<b>xiii</b>
<b>DAFTAR ISI</b> .....	<b>xv</b>
<b>DAFTAR GAMBAR</b> .....	<b>xix</b>
<b>DAFTAR TABEL</b> .....	<b>xxiii</b>
<b>BAB I PENDAHULUAN</b> .....	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Perumusan Masalah .....	1
1.3 Batasan Masalah .....	2
1.4 Tujuan Penelitian .....	2
1.5 Metodologi .....	2
1.6 Sistematika Penulisan .....	3
1.7 Relevansi .....	4
<b>BAB II TEORI PENUNJANG</b> .....	<b>5</b>
2.1 <i>Brushless DC Motor</i> (BLDCM) .....	5
2.1.1 Penjelasan BLDCM .....	5
2.1.2 Prinsip Kerja BLDCM .....	6
2.2 Penggerak Elektronik .....	8
2.3 Pengendalian BLDCM .....	9
2.3.1 Metode Six-Step .....	9
2.3.2 Metode Sinyal PWM .....	10
2.4 Rem Magnetik .....	11
2.5 Identifikasi Sistem .....	11
2.5.1 Identifikasi Statis .....	11
2.5.1.1 Identifikasi Statis dengan Metode Viteckova Orde 1 .....	13
2.5.1.2 Identifikasi Statis dengan Metode Viteckova Orde 2 .....	14
2.5.1.3 Identifikasi Statis dengan Metode Sundaresan & Krishnaswamy .....	14

2.5.1.4	Identifikasi Statis dengan Metode Grafis ...	15
2.6.	Validasi Model RMSE .....	16
2.7	Kontroler PID .....	17
2.7.1	Kontroler Proporsional (P) .....	17
2.7.2	Kontroler <i>Integral</i> (I) .....	18
2.7.3	Kontroler Derivatif (D) .....	18
2.8	Logika <i>Fuzzy</i> .....	19
2.8.1	Himpunan <i>Fuzzy</i> .....	19
2.8.2	Fungsi Keanggotaan <i>Fuzzy</i> .....	19
2.8.3	Operasi Himpunan <i>Fuzzy</i> .....	21
2.8.4	Sistem Kontrol <i>Fuzzy</i> .....	22
2.9	Kontroler PID- <i>Fuzzy</i> .....	23
2.10	Sensor <i>Rotary Encoder</i> .....	24
2.11	Arduino .....	25
<b>BAB III</b>	<b>PERANCANGAN SISTEM .....</b>	<b>27</b>
3.1	Arsitektur Sistem .....	27
3.2	Perancangan Perangkat Keras ( <i>Hardware</i> ) .....	28
3.2.1	Perancangan Mekanik <i>Plant</i> .....	29
3.2.2	Perancangan Sensor <i>Rotary Encoder</i> .....	30
3.2.3	Perancangan Rangkaian Penggerak Elektronik .....	32
3.2.3.1	Rangkaian Pembagi Fasa .....	32
3.2.3.2	Rangkaian Pengolah Sinyal Fasa dan PWM .....	33
3.2.3.3	Rangkaian <i>Optoisolator</i> .....	34
3.2.3.4	Rangkaian <i>Switching</i> .....	35
3.2.3.5	Rangkaian <i>Power Supply</i> .....	35
3.3	Perancangan Perangkat Lunak ( <i>Software</i> ) .....	36
3.3.1	Arduino Sebagai Pembangkit Frekuensi dan PWM	36
3.3.2	Arduino Sebagai Pembaca Data Sensor .....	38
3.3.3	Perancangan Metode Identifikasi .....	38
3.4	Identifikasi <i>Plant</i> .....	39
3.4.1	Metode Identifikasi .....	39
3.4.2	Metode Pembebanan <i>Plant</i> .....	43
3.5	Perancangan Kontroler PID- <i>Fuzzy</i> .....	44
3.5.1	Perancangan Kontroler PID .....	45
3.5.2	Perancangan Mekanisme PID- <i>Fuzzy</i> .....	46
3.5.2.1	Fungsi Keanggotaan Masukan .....	47
3.5.2.2	Fungsi Keanggotaan Keluaran .....	47

3.5.2.3	<i>Rule Base</i> .....	48
3.5.2.4	Defuzifikasi .....	49
3.6	Perancangan Simulasi Kontroler .....	49
3.6.1	Blok Model <i>Plant</i> Beban Nominal .....	50
3.6.2	Blok Model Pembebanan <i>Plant</i> .....	50
3.6.3	Blok <i>Fuzzy Tunner</i> .....	51
<b>BAB IV PENGUJIAN DAN ANALISA</b> .....		<b>53</b>
4.1	Pengujian Arduino .....	53
4.1.1	Analisa Pengujian Pembangkit Sinyal PWM .....	53
4.1.2	Analisa Pengujian Pembangkit Sinyal Fasa .....	54
4.2	Pengujian Rangkaian Penggerak Elektronik .....	55
4.2.1	Analisa Pengujian Keluaran Gerbang AND dan NAND .....	55
4.2.2	Analisa Pengujian Keluaran <i>Optocoupler</i> .....	55
4.3	Analisa Pengujian Simulasi tanpa Kontroler .....	58
4.4	Analisa Pengujian Simulasi dengan Kontroler PID- <i>Fuzzy</i> .	60
4.5	Analisa Implementasi Pengujian Kontroler PID- <i>Fuzzy</i> .....	63
<b>BAB V PENUTUP</b> .....		<b>67</b>
5.1	Kesimpulan .....	67
5.2	Saran .....	67
<b>DAFTAR PUSTAKA</b> .....		<b>69</b>
<b>LAMPIRAN</b> .....		<b>71</b>
<b>RIWAYAT PENULIS</b> .....		<b>77</b>



## DAFTAR TABEL

<b>Tabel 2.1</b>	Urutan Pengaturan Saklar BLDCM .....	9
<b>Tabel 3.1</b>	Spesifikasi BLDCM .....	30
<b>Tabel 3.2</b>	Validasi Model Matematika Beban Minimal.....	40
<b>Tabel 3.3</b>	Validasi Model Matematika Beban Nominal .....	40
<b>Tabel 3.4</b>	Validasi Model Matematika Beban Maksimal .....	41
<b>Tabel 3.5</b>	Model Matematika Beban .....	44
<b>Tabel 3.6</b>	Parameter PID pada Kondisi Pembebanan.....	46
<b>Tabel 4.1</b>	Hasil Pengujian Pembangkit Frekuensi Fasa Arduino	54
<b>Tabel 4.2</b>	Hasil Pengujian Kecepatan BLDCM.....	57
<b>Tabel 4.3</b>	Spesifikasi Respon Tiap Beban .....	60
<b>Tabel 4.4</b>	Spesifikasi Respon Tiap Beban dengan Kontroler PID- <i>Fuzzy</i> .....	62
<b>Tabel 4.5</b>	Spesifikasi Implementasi Respon Tiap Beban.....	66



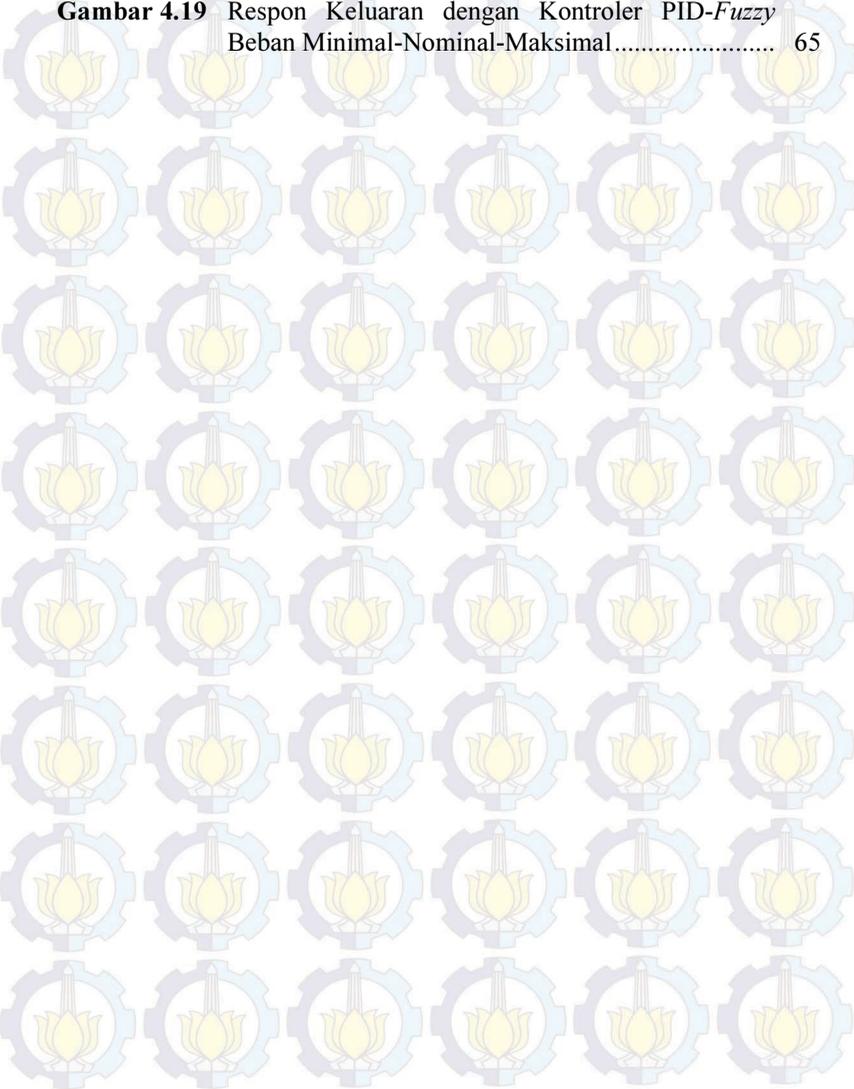
## DAFTAR GAMBAR

<b>Gambar 2.1</b>	Konstruksi BLDCM .....	6
<b>Gambar 2.2</b>	<i>Wiring Diagram</i> BLDCM.....	7
<b>Gambar 2.3</b>	Tegangan Stator BLDCM.....	7
<b>Gambar 2.4</b>	Urutan <i>Switching</i> pada <i>Stator</i> .....	8
<b>Gambar 2.5</b>	Bentuk Gelombang Stator Hasil <i>Switching</i> .....	9
<b>Gambar 2.6</b>	Driver 3 Fasa dengan Modul PWM.....	10
<b>Gambar 2.7</b>	<i>Switching</i> Menggunakan Sinyal PWM.....	10
<b>Gambar 2.8</b>	Karakteristik Respon Orde Satu .....	11
<b>Gambar 2.9</b>	Penentuan Garis Singgung Metode Grafis.....	15
<b>Gambar 2.10</b>	Diagram Blok Kontroler PID.....	17
<b>Gambar 2.11</b>	Fungsi Keanggotaan Segitiga .....	20
<b>Gambar 2.12</b>	Fungsi Keanggotaan Trapesium .....	21
<b>Gambar 2.13</b>	Sistem Kontrol <i>Fuzzy</i> .....	22
<b>Gambar 2.14</b>	Diagram Blok Kontroler PID- <i>Fuzzy</i> .....	24
<b>Gambar 2.15</b>	Keluaran Sinyal dan Susunan dari <i>Rotary Encoder</i> .....	24
<b>Gambar 2.16</b>	<i>Board</i> Arduino UNO (kiri) dan Arduino Mega2560 (kanan) .....	25
<b>Gambar 3.1</b>	Arsitektur Sistem .....	27
<b>Gambar 3.2</b>	Komponen dan Alur Perangkat Keras Sistem.....	29
<b>Gambar 3.3</b>	Konstruksi <i>Plant</i> Secara Keseluruhan .....	30
<b>Gambar 3.4</b>	Skema Rangkaian Sensor <i>Rotary Encoder</i> .....	31
<b>Gambar 3.5</b>	Konstruksi <i>Rotary Encoder</i> dengan <i>Optocoupler</i> ....	31
<b>Gambar 3.6</b>	Bentuk Fisik Rangkaian Penggerak Elektronik .....	32
<b>Gambar 3.7</b>	Skema Rangkaian Pembagi Fasa .....	32
<b>Gambar 3.8</b>	Skema Rangkaian Pengolah Sinyal Fasa dan PWM .....	33
<b>Gambar 3.9</b>	Skema Rangkaian <i>Optoisolator</i> .....	34
<b>Gambar 3.10</b>	Skema Rangkaian <i>Switching</i> .....	35
<b>Gambar 3.11</b>	Skema Rangkaian <i>Power Supply</i> .....	36
<b>Gambar 3.12</b>	<i>Flowchart</i> Pemrogramman Pembangkit PWM dan Frekuensi .....	37
<b>Gambar 3.13</b>	<i>Flowchart</i> Pembacaan Data Sensor Kecepatan .....	38
<b>Gambar 3.14</b>	Blok Diagram Simulink Identifikasi.....	39
<b>Gambar 3.15</b>	Respon Model Matematika Beban Minimal .....	40
<b>Gambar 3.16</b>	Respon Model Matematika Beban Nominal.....	41
<b>Gambar 3.17</b>	Respon Model Matematika Beban Maksimal.....	42
<b>Gambar 3.18</b>	Penarikan Garis Singgung pada Metode Grafis.....	42

<b>Gambar 3.19</b>	Diagram Blok Kontroler PID- <i>Fuzzy</i> .....	44
<b>Gambar 3.20</b>	Blok Diagram Kontroler PID .....	46
<b>Gambar 3.21</b>	Fungsi Keanggotaan <i>Fuzzy</i> untuk <i>Error</i> , <i>Delta Error</i> , dan <i>Integral Error</i> .....	47
<b>Gambar 3.22</b>	Fungsi Keanggotaan $K_p$ .....	47
<b>Gambar 3.23</b>	Fungsi Keanggotaan $K_i$ .....	48
<b>Gambar 3.24</b>	Fungsi Keanggotaan $K_d$ .....	48
<b>Gambar 3.25</b>	Diagram Blok Simulasi PID- <i>Fuzzy</i> pada Simulink .	49
<b>Gambar 3.26</b>	Blok Model Pembebanan <i>Plant</i> .....	50
<b>Gambar 3.27</b>	Diagram Blok <i>Fuzzy Tuner</i> .....	51
<b>Gambar 4.1</b>	Mekanisme Pengujian Sinyal PWM.....	53
<b>Gambar 4.2</b>	Hasil Pengujian Sinyal PWM.....	53
<b>Gambar 4.3</b>	Hasil Pengujian Sinyal Frekuensi Fasa .....	54
<b>Gambar 4.4</b>	Mekanisme Pengujian Sinyal gerbang AND dan NAND .....	55
<b>Gambar 4.5</b>	Sinyal Keluaran AND (kiri) dan NAND (kanan)....	55
<b>Gambar 4.6</b>	Sinyal Keluaran <i>Optocoupler HIGH</i> (kiri) dan <i>LOW</i> (kanan) .....	56
<b>Gambar 4.7</b>	Mekanisme Pengujian Kecepatan BLDCM .....	57
<b>Gambar 4.8</b>	Hasil Pengujian Kecepatan BLDCM.....	58
<b>Gambar 4.9</b>	Hasil Pengujian Tanpa Kontroler untuk Beban Minimal .....	59
<b>Gambar 4.10</b>	Hasil Pengujian Tanpa Kontroler untuk Beban Nominal.....	59
<b>Gambar 4.11</b>	Hasil Pengujian Tanpa Kontroler untuk Beban Maksimal.....	60
<b>Gambar 4.12</b>	Respon Keluaran Beban Minimal dengan Kontroler PID- <i>Fuzzy</i> .....	61
<b>Gambar 4.13</b>	Respon Keluaran Beban Nominal dengan Kontroler PID- <i>Fuzzy</i> .....	61
<b>Gambar 4.14</b>	Respon Keluaran Beban Maksimal dengan Kontroler PID- <i>Fuzzy</i> .....	62
<b>Gambar 4.15</b>	Respon Keluaran dengan Kontroler PID- <i>Fuzzy</i> Beban Berubah .....	63
<b>Gambar 4.16</b>	Respon Keluaran dengan Kontroler PID- <i>Fuzzy</i> Beban Minimal.....	64
<b>Gambar 4.17</b>	Respon Keluaran dengan Kontroler PID- <i>Fuzzy</i> Beban Nominal.....	64

**Gambar 4.18** Respon Keluaran dengan Kontroler PID-*Fuzzy*  
Beban Maksimal..... 65

**Gambar 4.19** Respon Keluaran dengan Kontroler PID-*Fuzzy*  
Beban Minimal-Nominal-Maksimal..... 65



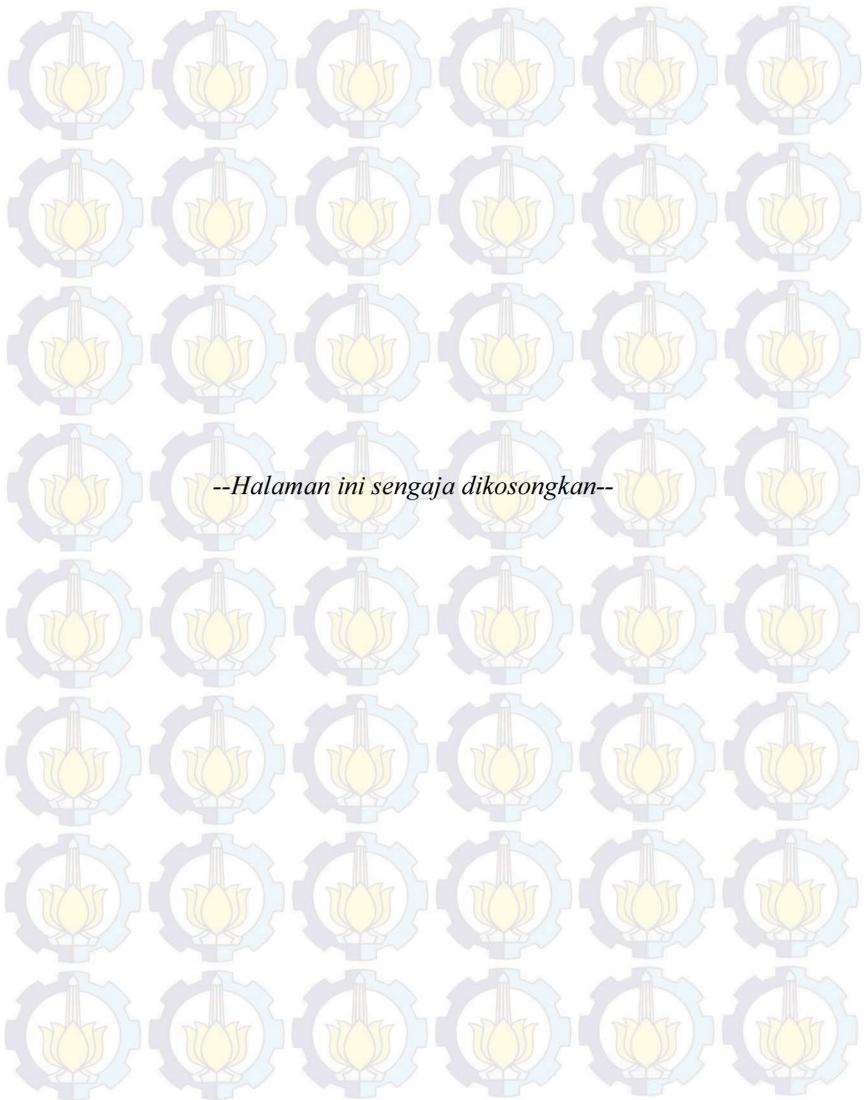


*--Halaman ini sengaja dikosongkan--*

## RIWAYAT PENULIS



Suwondo Saputra lahir di Prabumulih pada tanggal 23 September 1989. Penulis merupakan anak pertama dari pasangan Paino dan Tanzima. Setelah lulus dari SMKN 1 Cimahi pada tahun 2009, penulis melanjutkan studi di Program Studi Diploma III Politeknik Negeri Semarang (Polines), Jurusan Teknik Elektro. Pada tahun 2013 terdaftar sebagai mahasiswa lintas jalur Institut Teknologi Sepuluh Nopember (ITS), Jurusan Teknik Elektro dengan Bidang Studi Teknik Sistem Pengaturan. Selama menjalani studi di Institut Teknologi Sepuluh Nopember (ITS), penulis selalu antusias dalam mengikuti kegiatan perkuliahan. Penulis mengikuti seminar dan ujian Tugas Akhir di Bidang Studi Teknik Sistem Pengaturan, Jurusan Teknik Elektro, ITS Surabaya sebagai salah satu persyaratan untuk memperoleh gelar Sarjana Teknik Elektro. Email penulis adalah [suwondo1989@yahoo.com](mailto:suwondo1989@yahoo.com).



# BAB I

## PENDAHULUAN

Pada bab ini menjelaskan latar belakang, perumusan masalah, dan batasan masalah. Selain itu pada bab ini juga dijelaskan tujuan penelitian, metodologi yang digunakan, sistematika penulisan, dan relevansi.

### 1.1 Latar Belakang

Penggunaan motor listrik didalam kehidupan sehari-hari sudah menjadi kebutuhan yang sangat penting. Berbagai alat banyak penggunaan motor listrik, seperti *air conditioning*, *conveyor*, *vacuum cleaner* dan sebagainya. Motor DC adalah salah satu mesin listrik yang banyak digunakan di dunia industri, akan tetapi penggunaan motor DC menimbulkan masalah yang diakibatkan penggunaan sikat (*brush*). BLDCM merupakan alternatif pengganti motor DC. BLDCM memiliki kelebihan diantaranya adalah efisiensi yang tinggi, memiliki torsi besar, dan perawatan mudah. Penghilangan sikat komutator pada BLDCM dapat mengatasi permasalahan yang berhubungan dengan kontak berupa gesekan antara komutator yang menimbulkan keausan sehingga perlu dilakukan perawatan secara berkala.

Penggantian motor DC oleh BLDCM menjadi sebab dibutuhkannya cara kontrol yang berbeda untuk komutasi fasa dari BLDCM. Rangkaian kontrol pengganti kumutator pada BLDCM ini disebut dengan penggerak elektronik untuk mencatu daya ke kumparan stator untuk memberikan kontrol besarnya arus yang mengalir sehingga putaran rotor bisa dikontrol. Rangkaian kontrol terdiri atas 6 buah MOSFET yang digunakan untuk *switching* tegangan stator.

Kontroler PID (Proporsional-Integral-Derivatif) merupakan kontrol yang umum digunakan dalam industri dikarenakan kontrol yang sederhana akan tetapi hanya berlaku untuk satu kondisi, apabila motor diberi beban maka terjadi perubahan pada pemodelannya yang mengakibatkan perlunya *tuning* pada parameter kontrol PID untuk menjaga plant tetap stabil apabila terjadi perubahan beban. Penggunaan logika *fuzzy* dapat melakukan *tuning* parameter kontrol PID secara otomatis, sehingga dapat mempertahankan kecepatan motor dengan berbagai perubahan pada bebannya.

## 1.2 Perumusan Masalah

BLDCM yang merupakan motor 3 fasa akan tetapi membutuhkan tegangan DC sebagai sumbernya. BLDCM yang tidak memiliki sikat (*brush*) menjadi sebab dibutuhkannya rangkaian penggerak elektronik untuk mengubah tegangan DC menjadi tegangan AC 3 fasa agar dapat menggerakkan BLDCM. Tidak adanya sikat sebagai penunjang komutasi pada stator dan menurunnya kecepatan yang diakibatkan perubahan beban membuat pengaturan BLDCM menjadi lebih sulit. Oleh karena itu, pada Tugas Akhir ini dirancang rangkaian penggerak elektronik untuk menggerakkan BLDCM dan kontrol PID-*Fuzzy* pada pengaturan kecepatan BLDCM untuk mempertahankan performansi dari BLDCM tetap *steady state* terhadap nilai *set point* walaupun diberi pembebanan.

## 1.3 Batasan Masalah

Beberapa hal yang perlu dibatasi dalam Tugas Akhir ini agar penelitian yang dilakukan dapat tercapai, yaitu:

1. Efek pembebanan rem magnetik pada *plant* hanya dilihat dari besarnya sudut rem magnetik terhadap piringan rotor BLDCM dan pengaruh secara spesifik tidak diperhitungkan secara detail
2. Sistem kontrol yang dirancang hanya untuk mengatur kecepatan
3. Metode identifikasi dilakukan secara statis menggunakan 4 metode
4. *Range* kerja dari rangkaian penggerak elektronik berkisar antara 10-55 Hz.

## 1.4 Tujuan Penelitian

Pada tugas akhir ini, Penelitian Tugas Akhir ini adalah merancang penggerak elektronik dan kontroler PID-*Fuzzy* untuk mengatur kecepatan BLDCM sesuai dengan nilai *set point* yang diberikan.

## 1.5 Metodologi

Metodologi dalam pelaksanaan penelitian ini dilakukan dengan beberapa tahapan yaitu studi literatur, perancangan dan pembuatan sistem, pengujian sistem, analisa data, dan penulisan buku Tugas Akhir. Tahap studi literatur dilakukan dengan mencari beberapa sumber referensi yang berhubungan dengan penggerak elektronik dan kontrol PID-*Fuzzy*. Sumber referensi yang digunakan dapat berupa buku, jurnal,

*website*, dan sebagainya. Tahap perancangan dan pembuatan sistem dilakukan dengan cara mendesain penggerak elektronik, merancang *plant*, mendesain kontroler, membuat simulasi kontroler serta membuat sistem secara keseluruhan.

Tahap pengujian sistem dilakukan untuk mengetahui kontroler yang dibuat telah sesuai dengan hasil implementasinya. Pengujian ini dilakukan dengan menggunakan *software* Matlab. Tahap analisa data dilakukan berdasarkan hasil simulasi dan pengujian. Tahap ini dilakukan analisa terhadap persentase kesalahan yang terjadi. Hal ini diperlukan untuk mengevaluasi kembali dari hasil desain kontrol. Penulisan buku Tugas Akhir merupakan tahap akhir dalam penelitian. Buku Tugas Akhir ditulis secara intensif bila proses pengujian telah selesai.

## **1.6 Sistematika Penulisan**

Sistematika penulisan yang disusun pada Tugas Akhir ini terdiri dari lima bab, sebagai berikut:

### **BAB I : PENDAHULUAN**

Bab ini menjelaskan tentang latar belakang, perumusan masalah, batasan masalah, tujuan penelitian, sistematika penulisan dan relevansi.

### **BAB II : TEORI PENUNJANG**

Bab ini berisi tentang dasar – dasar pengetahuan dari tiap komponen yang digunakan pada sistem yang akan dirancang seperti: *Brushless DC Motor* (BLDCM), penggerak elektronik, rem magnetik, identifikasi sistem, validasi *Root Mean Square Error* (RMSE), kontroler *PID-Fuzzy*, sensor *rotary encoder*, dan Arduino.

### **BAB III : PERANCANGAN SISTEM**

Bab ini berisi tentang perancangan sistem secara garis besar yang terdiri dari arsitektur sistem, perancangan *hardware*, perancangan *software*, identifikasi BLDCM, dan perancangan dan implementasi kontroler.

### **BAB IV : PENGUJIAN DAN ANALISA**

Bab ini menjelaskan tentang hasil pengujian rangkaian penggerak elektronik, pengujian simulasi dan implementasi sistem secara keseluruhan.

## BAB V : PENUTUP

Bab ini berisi tentang kesimpulan dari penelitian Tugas Akhir ini dan saran – saran mengenai kemungkinan pengembangan penelitian ini.

### 1.7 Relevansi

Hasil yang dicapai dari Tugas Akhir ini diharapkan menjadi salah satu referensi yang dapat digunakan dalam pengaturan kecepatan BLDCM, sehingga dapat membantu dalam melakukan pengaturan BLDCM. Selain itu penelitian ini juga diharapkan dapat menjadi perbandingan dari beberapa metode pengaturan kecepatan BLDCM dimasa sebelumnya maupun yang akan datang.

## BAB II TEORI PENUNJANG

BAB II menjelaskan beberapa teori penunjang yang dijadikan sebagai acuan dalam melakukan penelitian. Beberapa hal yang dibahas meliputi *Brushless DC Motor* (BLDCM), penggerak elektronik, rem magnetik, identifikasi sistem, validasi *Root Mean Square Error* (RMSE), kontroler PID-Fuzzy, sensor *rotary encoder*, dan Arduino.

### 2.1 *Brushless DC Motor* (BLDCM)[1]

*Brushless Direct Current Motor* (BLDCM) adalah salah satu jenis motor yang cepat populer. BLDCM digunakan di dunia industri seperti permobilan, atmosphere, konsumen, otomasi medis, industri dan peralatan instrumentasi. Sesuai dengan namanya, BLDCM tidak menggunakan sikat untuk pergantian medan magnet (komutasi), tetapi dilakukan secara komutasi elektronik.

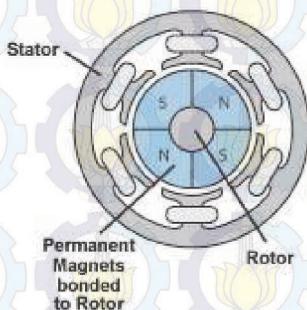
#### 2.1.1 Penjelasan BLDCM

Motor DC banyak digunakan pada aplikasi dunia industri memiliki konstruksi yang menyebabkan timbulnya beberapa kendala pada penggunaannya. Motor DC memiliki kemudahan dalam pengaturan serta perubahan kecepatan mendekati linier dengan perubahan tegangan, akan tetapi memiliki kendala utama yaitu pemakaian komutator *brush* atau sikat arang untuk menghubungkan kumparan rotor. Kendala yang mencakup penggantian atau penyetalan sikat arang, bunga api (*spark*) karena komutasi, keterbatasan arus dan tegangan, dan disipasi panas yang dibangkitkan *rotor*, sehingga diperlukan perawatan secara intensif. BLDCM menjadi solusi untuk mengatasi kendala tersebut. Walaupun BLDCM dapat mengatasi kendala motor DC, akan tetapi metode pengendalian BLDCM menjadi lebih sulit, karena tidak adanya *brush* yang menunjang proses komutasi, selain itu harga untuk BLDCM masih lebih mahal dibanding dengan motor listrik lainnya.

BLDCM adalah motor DC yang dihilangkan sikat dan komutatornya. Secara umum BLDCM terdiri dari dua bagian, yakni, *rotor*, yang terbuat dari permanen magnet dan *stator*, yang terbuat dari kumparan 3 fasa. Walaupun merupakan motor sinkron 3 fasa, motor ini tetap disebut dengan BLDCM karena pada implementasinya BLDCM menggunakan sumber DC sebagai sumber tegangan yang kemudian

diubah menjadi tegangan AC dengan menggunakan *inverter* 3 fasa. Tujuan dari pemberian tegangan AC 3 fasa pada *stator* BLDCM adalah menciptakan medan magnet putar stator agar BLDCM bisa berputar. Rangkaian komutasi elektronik mengeliminasi penggantian sikat arang sehingga kumparan dapat dilewati arus dengan tegangan yang lebih tinggi.

Konstruksi dari BLDCM dapat dilihat pada Gambar 2.1.



Gambar 2.1 Konstruksi BLDCM

### 2.1.2 Prinsip Kerja BLDCM[2]

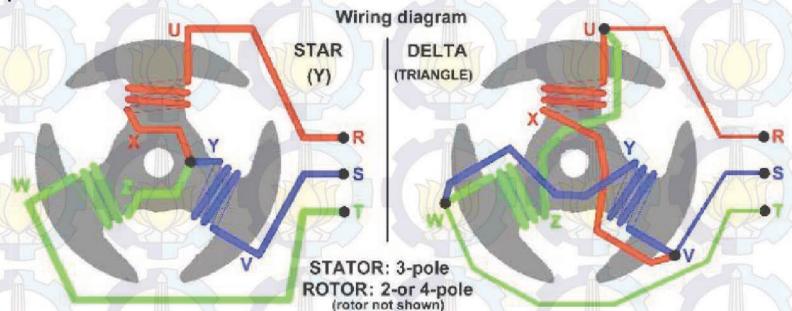
Motor BLDC ini dapat bekerja ketika stator yang terbuat dari kumparan diberikan arus 3 fasa. Akibat arus yang melewati kumparan pada stator timbul medan magnet ( $B$ ):

$$B = \frac{\mu NI}{2l} \quad (2.1)$$

Dimana  $N$  merupakan jumlah lilitan,  $i$  merupakan arus,  $l$  merupakan panjang lilitan, dan  $\mu$  adalah permeabilitas bahan. Arus yang diberikan berupa arus AC, maka nilai medan magnet dan polarisasi setiap kumparan akan berubah-ubah. Medan magnet yang berubah-ubah membuat kutub stator mengikuti perubahan medan magnet, sehingga BLDCM berputar. Kecepatan medan putar magnet dapat dirumuskan seperti Persamaan 2.2.

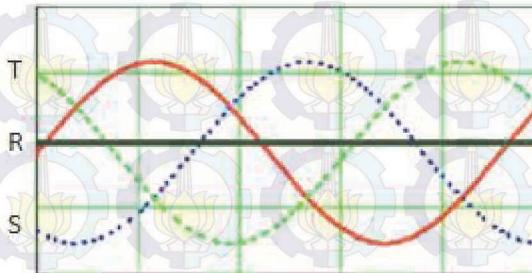
$$N_s = \frac{120f}{p} \quad (2.2)$$

Dimana  $f$  merupakan frekuensi tegangan fasa yang dinyatakan dalam satuan hertz (Hz),  $p$  merupakan jumlah kutub (*pole*) pada rotor dan  $120^\circ$  didapat dalam 1 putaran ( $360^\circ$ ) per 3 fasa motor. Ketika motor berputar permanent magnet pada rotor bergerak melewati kumparan stator dan menginduksi potensial listrik dalam kumparan tersebut, maka terjadinya GGL balik. *Wiring* pada BLDCM dapat dilihat pada Gambar 2.2.



**Gambar 2.2** *Wiring Diagram* BLDCM

Pada Gambar 2.2 dapat dilihat bahwa *wiring diagram* dari BLDCM ada 2 macam. Berdasarkan Gambar 2.2, medan putar magnet stator timbul akibat adanya perubahan polaritas pada stator R, S, dan T. Perubahan polaritas ini terjadi akibat adanya arus yang mengalir pada stator.

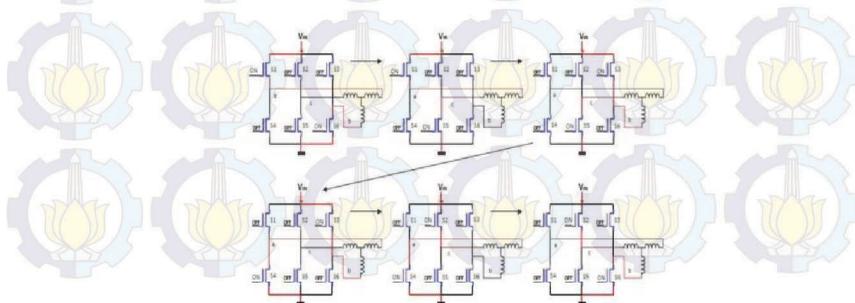


**Gambar 2.3** Tegangan Stator BLDCM

Berdasarkan Gambar 2.3, ketika medan stator R diberikan tegangan negatif maka akan timbul medan magnet dengan polaritas negatif sedangkan medan stator S dan T yang diberikan tegangan positif akan memiliki polaritas positif. Akibat adanya perbedaan polaritas antara medan magnet kumparan stator dan magnet rotor, sisi positif magnet rotor akan berputar mendekati medan magnet stator R, sedangkan sisi negatifnya akan berputar mengikuti medan magnet stator S dan T. Akibat tegangan yang digunakan berupa tegangan AC sinusoidal, medan magnet stator R, S, dan T akan berubah – ubah polaritasnya dan besarnya mengikuti perubahan tegangan AC sinusoidal. Ketika medan stator R dan S memiliki medan magnet negatif akibat mendapatkan tegangan negatif dan medan stator T memiliki medan magnet positif akibat tegangan positif, magnet permanen rotor akan berputar menuju ke polaritas yang bersesuaian yakni bagian negatif akan berputar menuju medan magnet stator T dan sebaliknya bagian positif akan berputar menuju medan magnet stator R dan S. Tegangan AC sinusoidal yang digunakan secara kontinyu, proses perubahan polaritas tegangan pada stator ini akan terjadi secara terus menerus sehingga menciptakan medan putar magnet stator dan magnet permanen rotor akan berputar mengikuti medan putar magnet stator ini. Hal inilah yang menyebabkan rotor pada BLDCM dapat berputar.

## 2.2 Penggerak Elektronik[2]

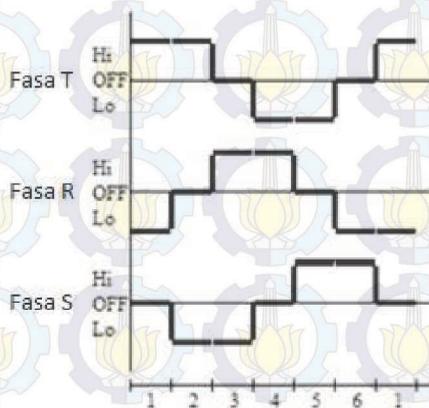
BLDCM membutuhkan enam langkah komutasi yang dilakukan secara kontinyu untuk berputar. *Driver* 3 fasa terdiri dari 6 buah saklar yang akan memberikan tegangan positif (sinyal *high*) dan tegangan 0V (sinyal *low*) secara bergantian. Untuk membangkitkan tegangan AC pada BLDCM.



**Gambar 2.4** Urutan *Switching* pada *Stator*

**Tabel 2.1** Urutan Pengaturan Saklar BLDCM

Urutan ke-	Saklar Aktif		Fasa T	Fasa R	Fasa S
1	S1	S6	High	Low	Off
2	S1	S5	High	Off	Low
3	S3	S5	Off	High	Low
4	S3	S4	Low	High	Off
5	S2	S4	Low	Off	High
6	S2	S6	Off	Low	High



**Gambar 2.5** Bentuk Gelombang Stator Hasil Switching

## 2.3 Pengendalian BLDCM

Metode yang digunakan dalam pengaturan BLDCM diantaranya adalah metode *six-step*, metode sinyal PWM, dan metode PWM sinusoidal.

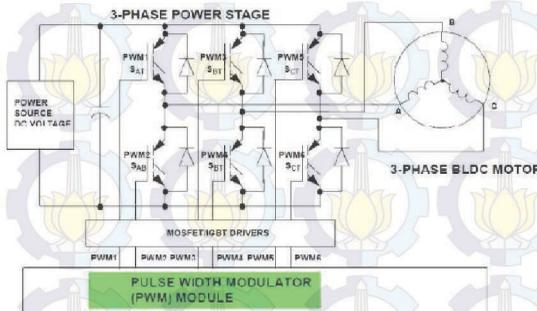
### 2.3.1 Metode *Six-Step*

Metode *six-step* merupakan metode yang umum digunakan dalam pengaturan BLDCM, karena sifatnya yang sederhana, sehingga mudah untuk diaplikasikan. Metode ini disebut metode *six-step*, karena untuk menciptakan gelombang *trapezoid* atau *square* yang menyerupai gelombang sinusoidal, menggunakan 6 step posisi saklar untuk BLDCM. Metode *six-step* ini menggunakan urutan penyalan saklar

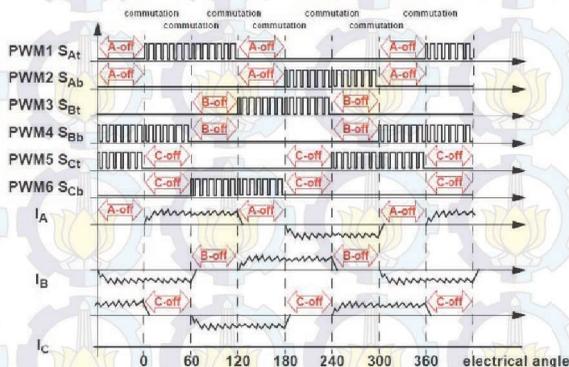
seperti Tabel 2.1. Frekuensi gelombang *square* diatur dengan mengatur waktu tunda (*delay*) pada pergantian tiap urutan penyalan saklar. Penyalan saklar berdasarkan sensor *hall* untuk mengetahui posisi rotor.

### 2.3.2 Metode Sinyal PWM

Metode sinyal PWM menggunakan modul PWM untuk memberikan *trigger* pada 6 buah MOSFET. Sinyal PWM yang diberikan pada MOSFET berbeda-beda seperti. Gambar 2.6, sehingga keluaran yang dihasilkan oleh *driver* tersebut dapat menggerakkan BLDCM terlihat pada Gambar 2.7.



Gambar 2.6 Driver 3 Fasa dengan Modul PWM



Gambar 2.7 Switching Menggunakan Sinyal PWM

## 2.4 Rem Magnetik[3]

Sistem pengereman magnetik menggunakan gaya magnet untuk memperlambat gerakan. Sebuah piringan dengan bahan logam non-magnetik terpasang dengan poros rotor dan diapit oleh sisi yang dipasangi magnet permanen.

Logam piringan non-magnetik berputar memotong medan magnet yang ditimbulkan oleh magnet permanen, sehingga menghasilkan arus *eddy* pada piringan. Arus *eddy* menimbulkan medan magnet yang arahnya berlawanan dengan medan magnet sebelumnya, sehingga memperlambat putaran dari poros tersebut. Rem magnetik digunakan sebagai efek pembebanan, bukan untuk menghentikan gerakan.

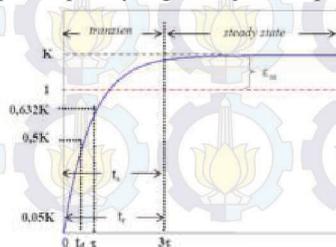
## 2.5 Identifikasi Sistem[4]

Model matematika sistem diperlukan untuk menggambarkan hubungan *input* dengan *output* sistem. Metode identifikasi merupakan suatu metode yang menggunakan hubungan data masukan dan keluaran yang selanjutnya dilakukan pengujian dan analisa dengan metode pendekatan, sehingga dapat ditentukan nilai parameter secara analitik. Beberapa sinyal masukan yang dapat digunakan untuk mendapatkan respon suatu sistem seperti sinyal *step*, *ramp*, *impulse*, dan sinusoidal.

Terdapat dua macam identifikasi sistem yaitu identifikasi statis dan identifikasi dinamis. Pada identifikasi statis, sinyal yang diberikan berupa sinyal *step* yang konstan sampai sistem mencapai keadaan *steady state*. Pada identifikasi dinamis, sinyal yang digunakan berupa sinyal acak (*random*).

### 2.5.1 Identifikasi Statis

Suatu sistem disebut orde satu apabila dilihat secara grafis memiliki bentuk respon seperti yang ditunjukkan pada Gambar 2.8.



Gambar 2.8 Karakteristik Respon Orde Satu

Karakteristik respon sistem orde satu dilihat berdasarkan respon sistem ketika diberi masukan sinyal *step*. Model matematik sistem orde satu dirumuskan dengan:

$$G(s) = \frac{K}{\tau s + 1} \quad (2.3)$$

Terdapat dua parameter yaitu K yang menyatakan *gain overall* dan  $\tau$  yang menyatakan *time constant* (konstanta waktu). Nilai K merupakan hasil perhitungan dari Persamaan 2.4:

$$K = \frac{Y_{ss}}{X_{ss}} \quad (2.4)$$

$Y_{ss}$  adalah keluaran saat *steady state* dan  $X_{ss}$  adalah masukan saat *steady state*. Karakteristik sistem orde satu dibedakan menjadi karakteristik respon transien dan karakteristik respon keadaan tunak atau *steady state*. Karakteristik respon transien pada orde satu terdiri dari:

a. Spesifikasi Teoritis

*Time constant* atau konstanta waktu ( $\tau$ ) merupakan waktu yang dibutuhkan respon mulai  $t = 0$  sampai respon mencapai 63,2% dari respon *steady state*. Konstanta waktu menyatakan kecepatan respon sistem.

b. Spesifikasi Praktis

- *Settling time* atau waktu tunak ( $t_s$ ) merupakan waktu yang menyatakan bahwa respon sistem telah masuk pada daerah *steady state*. Jika dihubungkan dengan konstanta waktu  $\tau$ , maka  $t_s$  dapat dirumuskan dengan:

$$t_s (\pm 5\%) \approx 3\tau \quad (2.5)$$

$$t_s (\pm 2\%) \approx 4\tau \quad (2.6)$$

$$t_s (\pm 0.5\%) \approx 5\tau \quad (2.7)$$

- *Rise time* atau waktu naik ( $t_r$ ) merupakan waktu yang menyatakan bahwa respon sistem telah naik dari 5% - 95% atau 10% - 90% dari nilai respon *steady state*.

$$t_r (5\% - 95\%) = \tau \ln 19 \quad (2.8)$$

$$t_r (10\% - 90\%) = \tau \ln 9 \quad (2.9)$$

- *Delay time* atau waktu tunda ( $t_d$ ) merupakan waktu yang dibutuhkan respon mulai  $t = 0$  sampai respon mencapai 50% dari nilai *steady state*. Waktu tunda menyatakan besarnya faktor keterlambatan respon akibat proses *sampling*.

$$t_d = \tau \ln 2 \quad (2.10)$$

Karakteristik respon *steady state* sistem orde satu diukur berdasarkan kesalahan pada keadaan tunak atau *error steady state* ( $e_{ss}$ ), yaitu:

$$e_{ss} = Y_{ss} - X_{ss} \quad (2.11)$$

### 2.5.1.1 Identifikasi Statis dengan Metode Vitečková Orde 1

Metode ini menggunakan pendekatan orde satu dengan kemungkinan adanya waktu tunda. Fungsi alih untuk metode Vitečková Orde 1 ditunjukkan pada Persamaan 2.12.

$$G_{V1}(s) = \frac{K}{\tau_{V1}s + 1} e^{-T_{dV1}s} \quad (2.12)$$

$T_{dV1}$  adalah waktu tunda (*delay time*)

$$T_{dV1} = 1,498t_{33} - 0,498t_{70} \quad (2.13)$$

$\tau_{V1}$  adalah konstanta waktu

$$\tau_{V1} = 1,245(t_{70} - t_{33}) \quad (2.14)$$

$t_{33}$  dan  $t_{70}$  merupakan waktu saat respon berada pada kondisi 33% dan 70% dari keluaran *steady state*. Apabila  $T_{dV1}$  bernilai negatif, maka sistem dianggap tidak memiliki waktu tunda.

### 2.5.1.2 Identifikasi Statis dengan Metode Vitečková Orde 2

Metode ini menggunakan pendekatan orde dua dengan kemungkinan adanya waktu tunda. Fungsi alih untuk metode Vitečková Orde 2 ditunjukkan pada Persamaan 2.15.

$$G_{V2}(s) = \frac{K}{(\tau_{V2}s + 1)^2} e^{-T_{dV2}s} \quad (2.15)$$

$T_{dV2}$  adalah waktu tunda (*delay time*)

$$T_{dV2} = 1,937t_{33} - 0,937t_{70} \quad (2.16)$$

$\tau_{V2}$  adalah konstanta waktu

$$\tau_{V2} = 0,794(t_{70} - t_{33}) \quad (2.17)$$

$t_{33}$  dan  $t_{70}$  merupakan waktu saat respon berada pada kondisi 33% dan 70% dari keluaran *steady state*. Apabila  $T_{dV2}$  bernilai negatif, maka sistem dianggap tidak memiliki waktu tunda.

### 2.5.1.3 Identifikasi Statis dengan Metode Sundaresan & Krishnaswamy

Metode ini menggunakan pendekatan orde satu dengan kemungkinan adanya waktu tunda. Fungsi alih untuk metode Sundaresan & Krishnaswamy ditunjukkan pada Persamaan 2.18.

$$G_{SK}(s) = \frac{K}{\tau_{SK}s + 1} e^{-T_{dSK}s} \quad (2.18)$$

$T_{dSK}$  adalah waktu tunda

$$T_{dSK} = 1,3t_{35,3} - 0,29t_{85,3} \quad (2.19)$$

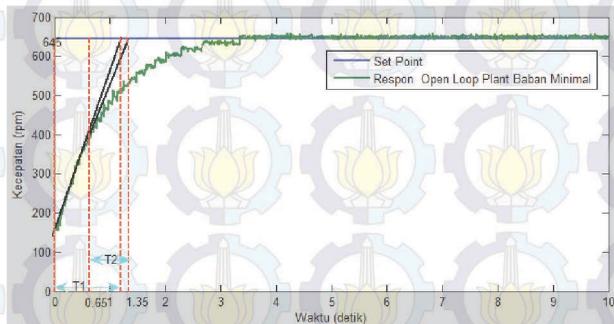
$\tau_{SK}$  adalah konstanta waktu

$$T_{dSK} = 0,67(t_{85,3} - t_{35,3}) \quad (2.20)$$

$t_{35,3}$  dan  $t_{85,3}$  merupakan waktu saat respon berada pada kondisi 35,3% dan 85,3% dari keluaran *steady state*. Apabila  $\tau_{dSK}$  bernilai negatif, maka sistem dianggap tidak memiliki waktu tunda.

#### 2.5.1.4 Identifikasi Statis dengan Metode Grafis

Metode ini merupakan secara grafis dengan menarik garis singgung yang memotong respon *plant* pada Gambar 2.9.



**Gambar 2.9** Penentuan Garis Singgung Metode Grafis

$t_1$  dan  $t_2$  merupakan waktu yang digunakan untuk menentukan akar-akar karakteristik dari sistem nantinya. Pada metode ini untuk mendapatkan  $t_1$  dan  $t_2$  terdapat prosedur tertentu.

Prosedurnya adalah sebagai berikut ;

- 1) Tarik garis singgung dari awal respon sampai garis tersebut mempunyai kemiringan yang sama dengan respon. Garis akan berakhir pada  $X_{ss}$ , pada gambar garis ini ditunjukkan pada garis hitam tebal.
- 2) Tarik garis lurus dari awal repon sampai  $X_{ss}$ , pada gambar garis ini ditunjukkan pada garis putus-putus berwarna merah.
- 3) Selisih antara garis tersebut merupakan nilai  $T_1$
- 4) Tarik lagi garis yang menyinggung dari akhir garis singgung yang pertama sampai menuju  $X_{ss}$ .
- 5) Ulangi lagi langkah 2 pada mulai dari awal garis 2 berpotongan dengan respon.
- 6) Nilai  $t_2$  adalah selisih 2 garis tersebut.
- 7) Setelah nilai  $T_1$  dan  $T_2$  didapatkan maka bentuk didapatkan fungsi transfer sebagai berikut

$$G(s) = \frac{1/T_1}{s + 1/T_1} + \frac{1/T_2}{s + 1/T_2} \quad (2.21)$$

## 2.6 Validasi Model RMSE[4]

Identifikasi parameter sistem didapatkan data untuk mendapatkan model matematik *plant*. Tujuan validasi model secara umum adalah untuk membuktikan bahwa model yang diidentifikasi memenuhi persyaratan permodelan menurut kriteria tujuan (*objective*) dari aproksimasi permodelan yang baik. Model matematika tersebut perlu diuji validasi untuk mengetahui kesamaan dengan *plant* dalam kondisi nyata. Metode validasi model menggugurkan *Root Mean Square Error* (RMSE) untuk mengetahui perbandingan dengan data yang didapat dengan model matematika melalui perhitungan yang disimulasikan.

RMSE adalah pengukuran akurasi pada nilai deret waktu secara statistik, khususnya *trend*. Persamaan 2.22 menyajikan formulasi dasar dari RMSE.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n e_i^2} \quad i = 1,2,3,4,\dots,n \quad (2.22)$$

dimana  $e$  adalah persentase kesalahan data hasil dari permodelan terhadap hasil pengukuran seperti yang dirumuskan pada Persamaan (2.23).

$$e_i = A_i - M_i \quad i = 1,2,3,4,\dots,n \quad (2.23)$$

Keterangan :

$n$  : jumlah data

$i$  : urutan data

$A$  : nilai data hasil pengukuran

$M$  : nilai data hasil permodelan

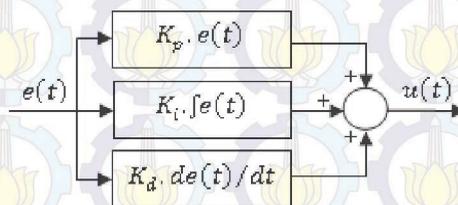
RMSE adalah komponen terpadu dari sebuah model statistika seperti regresi. Hal ini menjadikan RMSE sebagai sebuah pengukuran alami yang digunakan dalam berbagai evaluasi terhadap perkiraan kesalahan yang menggunakan metode statistika. Secara umum, tidak ada kriteria mutlak dari sebuah data untuk dianggap “bagus”. Keuntungan yang

dimiliki RMSE adalah kesamaan skala yang dimiliki oleh data hasil pengukuran dengan data hasil permodelan, sehingga RMSE dapat merepresentasikan ukuran dari kesalahan rata-rata. Nilai mutlak dari penghitungan ini dijumlahkan untuk setiap titik yang dilengkapi atau perkiraan dalam domain waktu dan dibagi kembali dengan jumlah  $n$  pengambilan data. Hal ini membuat kesalahan diukur dalam persen sehingga kesalahan sebuah deret waktu dapat dibandingkan di tingkat yang berbeda.

## 2.7 Kontroler PID[5]

Kontroler PID merupakan kontroler yang banyak digunakan di industri. Hal ini dikarenakan kemudahan penggunaannya terutama pada segi perhitungan. Kontroler PID melakukan mekanisme kontrol melalui selisih antara masukan dengan nilai keluaran yang diumpun balik. Kontroler ini terdiri dari 3 parameter yaitu proporsional, *integral* dan derivatif. Kontrol proporsional melakukan perkalian dengan kesalahan. Kontrol *integral* melakukan penjumlahan kesalahan tiap waktu. Kontrol differensial melakukan aksi kontrol berdasarkan kecepatan perubahan sinyal kesalahan. Ketiga kontroler ini selanjutnya ditambahkan untuk menghasilkan sinyal kontrol  $u(t)$  seperti pada Gambar 2.10.

$$u(t) = K_p \left( e(t) + \frac{1}{\tau_i} \int_0^t e(t) + \tau_d \frac{de(t)}{dt} \right) \quad (2.24)$$



Gambar 2.10 Diagram Blok Kontroler PID

### 2.7.1 Kontroler Proporsional (P)

Kontroler proporsional merupakan kontroler yang aksi kontrolnya proporsional atau sebanding terhadap sinyal kesalahan. Kontroler proporsional berupa konstanta dan secara sederhana keluaran kontroler proporsional merupakan perkalian antara konstanta

proporsional dengan masukannya. Persamaan dari kontrol proporsional dapat dilihat sebagai berikut:

$$u(t) = K_p (e(t)) \quad (2.25)$$

Karakteristik dari kontroler proporsional adalah dapat mempercepat respon sistem untuk mencapai keadaan *steady state*. Semakin besar nilai kontrol proporsional akan memperkecil kesalahan keadaan *steady state*, namun akan membawa sistem ke daerah tidak stabil, karena itu kontroler proporsional tidak dapat menghilangkan nilai *error* ( $e \neq 0$ ), sehingga diperlukan kontroler *integral* agar nilai *error* dapat mendekati nol.

### 2.7.2 Kontroler Integral (I)

Karakteristik kontroler *integral* adalah menghasilkan respon sistem yang memiliki nilai *error steady state* = 0 (*error* mendekati nol), namun kontroler ini cenderung memperlambat respon. Keluaran kontroler *integral* merupakan penjumlahan yang terus menerus dari perubahan masukannya. Jika sinyal kesalahan tidak mengalami perubahan, keluaran akan menjaga keadaan seperti sebelum terjadinya perubahan masukan. Persamaan dari kontrol *integral* dapat dilihat sebagai berikut:

$$u(t) = \frac{K_p}{\tau_i} \left( \int_0^t e(t) \right) \quad (2.26)$$

### 2.7.3 Kontroler Derivatif (D)

Karakteristik keluaran kontroler diferensial memiliki sifat seperti operasi derivatif. Perubahan yang mendadak pada masukan kontroler, akan mengakibatkan perubahan yang sangat besar dan cepat pada keluaran. Persamaan dari kontrol diferensial dapat dilihat sebagai berikut:

$$u(t) = K_p \tau_d \left( \frac{de(t)}{dt} \right) \quad (2.27)$$

Kelebihan kontroler derivatif apabila ditambah dengan kontroler proporsional dapat mengurangi aksi kontrol yang berlebihan sehingga tidak mengalami *overshoot*.

## 2.8 Logika Fuzzy[6]

Konsep logika *fuzzy* pertama kali dikenalkan pada tahun 1965 oleh L.A. Zadeh dari universitas California Barkeley dalam makalah seminarnya yang berjudul “*Fuzzy Set*”. Pada konsep logika *fuzzy* harga kebenaran diberikan dalam variabel bahasa (linguistik) dengan menyertakan sifat kekaburan (*fuzziness*) pada proposisinya. Harga kebenaran dan derajat kekaburan pada variabel linguistik dapat dinyatakan dengan tolok ukur, misalnya agak, cukup, sangat, dan sebagainya. Sehingga dalam logika *fuzzy* memungkinkan nilai keanggotaan antara 0 dan 1. Ini yang membedakan antara logika *fuzzy* dan logika Boolean yang nilai keanggotaannya hanya 0 dan 1.

### 2.8.1 Himpunan Fuzzy

Suatu himpunan *fuzzy*  $F$  dalam semesta  $X$  didefinisikan sebagai kumpulan pasangan elemen  $x$  dan fungsi keanggotaan  $\mu_F(x)$ . Fungsi keanggotaan  $\mu_F(x)$  mempunyai nilai interval  $[0, 1]$  pada tiap  $x$  dalam semesta  $X$ . Nilai fungsi keanggotaan menunjukkan tingkat keanggotaan elemen  $x$  dalam  $F$ . Tingkat keanggotaan 1 menunjukkan keanggotaan penuh (*full membership*), tingkat 0 menyatakan tanpa keanggotaan (*non membership*), tingkat antara 0 dan 1 menyatakan kekaburan antara keanggotaan penuh dan tanpa keanggotaan.

Tingkat keanggotaan inilah yang biasanya disebut tingkat logika *fuzzy*. Himpunan *fuzzy* pada  $F$  pada semesta pembicaraan  $X$  yang anggotanya berhingga,  $X = \{x_1, x_2, x_3, \dots, x_n\}$  dapat dinyatakan dalam bentuk himpunan *fuzzy*. Secara umum himpunan *fuzzy* dinotasikan sebagai :

$$F = \left\{ (x, \mu_F(x)) \mid x \in X \right\} \quad (2.28)$$

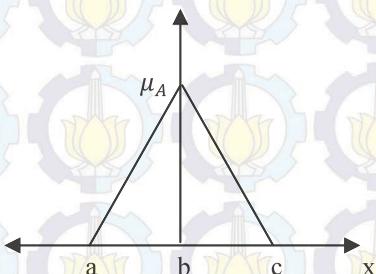
### 2.8.2 Fungsi Keanggotaan Fuzzy

Fungsi keanggotaan adalah suatu fungsi untuk menyatakan hubungan antara sebuah masukan dengan sebuah himpunan *fuzzy*. Untuk menyatakan suatu fungsi keanggotaan dalam pendefinisian himpunan *fuzzy* tergantung pada metode yang akan digunakan. Secara umum ada

dua metode yaitu pendefinisian secara numerik dan bentuk fungsi. Pendefinisian secara numerik diterapkan atas himpunan dengan pendukung diskrit, sedangkan pendefinisian bentuk fungsi diterapkan pada pendukung kontinyu.

a. Fungsi Keanggotaan Segitiga (Triangular)

Fungsi keanggotaan segitiga (tringular) sesuai dengan namanya berbentuk segitiga dengan parameter a, b dan x pada Persamaan 2.29. Fungsi keanggoaan segitiga ditunjukkan oleh Gambar 2.11.



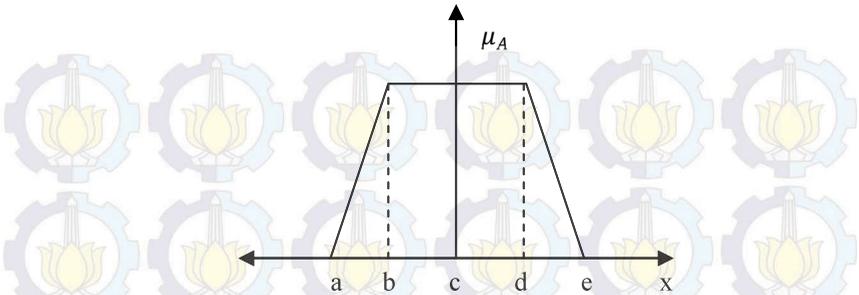
Gambar 2.11 Fungsi Keanggotaan Segitiga

$$\mu_A = \begin{cases} 0, & x \leq a; x \geq c \\ \frac{x-a}{b-a}, & a < x < b \\ \frac{c-x}{c-b}, & b < x < c \end{cases} \quad (2.29)$$

b. Fungsi Keanggotaan Trapesium

Fungsi ini menggunakan parameter a dan b, dengan Persamaan 2.30, serta bentuk fungsi keanggotaan terlihat pada Gambar 2.12.

$$\mu_A(x) = \begin{cases} 0, & x \leq a; x \geq e \\ \frac{x-a}{b-a}, & a < x < b \\ 1, & b \leq x < d \\ \frac{d-x}{d-c}, & d < x < e \end{cases} \quad (2.30)$$



**Gambar 2.12** Fungsi Keanggotaan Trapesium

### 2.8.3 Operasi Himpunan *Fuzzy*

Operasi dasar yang berlaku pada himpunan *fuzzy* A dan B dari semesta X dengan tingkat keanggotaan  $\mu_A(x)$  dan  $\mu_B(x)$  untuk  $x \in X$  adalah sebagai berikut :

- a. Dua himpunan *fuzzy* A dan B dikatakan sama ( $A = B$ ) apabila keduanya memiliki semesta yang sama dan fungsi keanggotaan yang sama. Hal ini dapat dinyatakan dengan Persamaan 2.31.

$$\frac{\mu(x_A)}{x} = \frac{\mu(x_B)}{x}; x \in X \quad (2.31)$$

- b. Gabungan dua himpunan *fuzzy* A dan B ( $A \cup B$ ) dinyatakan dengan Persamaan 2.32.

$$\mu_A \cup B = \max(\mu_A(x), \mu_B(x)); x \in X \quad (2.32)$$

- c. Irisan dari himpunan *fuzzy* A dan B ( $A \cap B$ ) dinyatakan seperti yang ditunjukkan pada Persamaan 2.33.

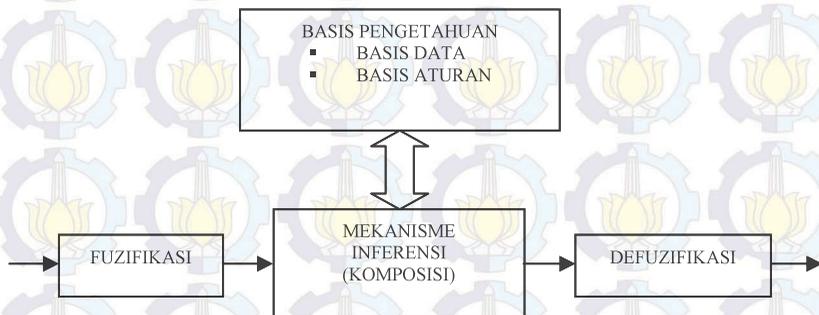
$$\mu_A \cap B = \min(\mu_A(x), \mu_B(x)); x \in X \quad (2.33)$$

- d. Komplemen (operasi “NOT”) himpunan *fuzzy* A dinotasikan dengan  $\bar{A}$  dapat dinyatakan pada Persamaan 2.34.

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x); x \in X \quad (2.34)$$

### 2.8.4 Sistem Kontrol *Fuzzy*[6]

Sistem kontrol *fuzzy* merupakan pemetaan dari masukan menjadi keluaran *fuzzy*. Secara umum sistem kontrol *fuzzy* terdiri dari empat bagian dasar yaitu fuzifikasi, aturan basis (*Rule Base*), inferensi *fuzzy*, dan defuzifikasi. Proses pada sistem kontrol *fuzzy* dimulai dari nilai masukan *fuzzy* diubah menjadi himpunan *fuzzy* melalui proses fuzifikasi. Selanjutnya himpunan *fuzzy* masuk ke mekanisme inferensi *fuzzy* untuk menentukan kesimpulan *fuzzy* dengan mengacu pada *rule base*. Proses akhir masuk ke proses defuzifikasi untuk diubah menjadi keluaran *fuzzy*. Sistem kontrol *fuzzy* ditunjukkan pada Gambar 2.13.



Gambar 2.13 Sistem Kontrol *Fuzzy*

a. Fuzifikasi

Fuzifikasi merupakan proses yang mengubah variabel nyata menjadi variabel *fuzzy*. Hal ini bertujuan agar masukan kontroler *fuzzy* bisa dipetakan menuju jenis yang sesuai dengan himpunan *fuzzy*. Pemetaan dilakukan dengan bantuan model dari fungsi keanggotaan agar dapat diketahui besar masukan tersebut.

b. Aturan Basis (*Rule Base*)

Kaidah dasar yang berisi aturan-aturan secara linguistik yang menunjukkan kepakaran terhadap *plant* atau suatu aturan yang menghubungkan antara masukan *fuzzy* dan aksi kontrol dari kontroler *fuzzy*. Format yang paling umum adalah aturan *IF-THEN*, sebagai berikut:

***IF* premise *THEN* conclusion**

c. Inferensi *Fuzzy*

Inferensi *fuzzy* adalah sebuah proses formulasi pemetaan masukan terhadap keluaran dengan menggunakan logika *fuzzy*. Proses dari inferensi *fuzzy* melibatkan fungsi keanggotaan operator logika *fuzzy*, dan aturan *IF-THEN*. Terdapat dua metode inferensi yang paling dikenal yaitu metode inferensi Mamdani dan metode inferensi Takagi-Sugeno. Pada metode Mamdani, menggunakan operasi Max-Min atau *Max-product* dalam memperoleh keluaran, sedangkan pada metode Takagi-Sugeno menggunakan fungsi keanggotaan keluaran yang bersifat linear atau konstan.

- Metode Mamdani

$$\mu_u(k) = \max [\mu_u(k), \min \{\mu_e(j), \mu_{de}(i)\}] \quad (2.35)$$

d. Defuzzifikasi

Defuzzifikasi adalah proses yang digunakan untuk mengubah kembali variabel *fuzzy* menjadi variabel nyata, atau dengan kata lain aksi kontrol *fuzzy* yang masih berupa himpunan, diubah menjadi nilai nyata yang berupa nilai tunggal. Metode yang dapat digunakan untuk defuzzifikasi yaitu *Center of Average* (COA) dengan Persamaan 2.36.

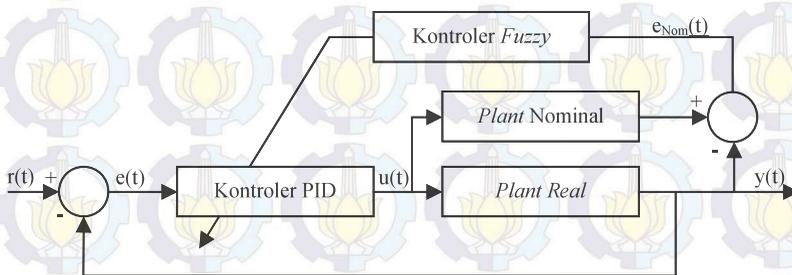
$$U_0 = \frac{\sum_{k=1}^m u_k(T) \cdot \mu_u(u_k(T))}{\sum_{k=1}^m \mu_k(u_k(T))} \quad (2.36)$$

## 2.9 Kontroler PID-Fuzzy

Lebih dari 90% kontroler yang digunakan saat ini adalah kontroler PID. Hal ini dikarenakan kontroler PID mudah dimengerti dan mudah diimplementasikan. Namun, kontroler PID hanya dapat digunakan dengan baik untuk parameter *plant* dan kondisi operasi yang tetap. Oleh karena itu, diperlukan mekanisme penalaan parameter kontroler PID yang dapat menyesuaikan dengan perubahan parameter *plant* dan kondisi operasi sistem.

Mekanisme kontroler PID-Fuzzy ditunjukkan seperti pada Gambar 2.14, dimana masukan *fuzzy* adalah keluaran sistem yang

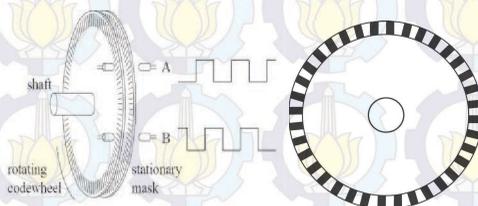
dibandingkan dengan keluaran referensi. Apabila terjadi perbedaan, kontroler *fuzzy* akan melakukan mekanisme penalaan parameter kontroler PID. Parameter PID yang didapat saat beban minimal, nominal, dan maksimal akan dijadikan sebagai fungsi anggota keluaran sinyal hasil dari defuzifikasi. Hasil defusifikasi akan diteruskan pada kontroler PID sebagai parameternya. *Plant* nominal yang berisi respon model yang diinginkan, sehingga *output plant* diharapkan mendekati respon pada saat beban nominal.



Gambar 2.14 Diagram Blok Kontroler PID-Fuzzy

## 2.10 Sensor *Rotary Encoder*[6]

Sensor yang digunakan untuk memonitor kecepatan putaran motor induksi pada Tugas Akhir ini adalah *rotary encoder*. *Rotary encoder* digunakan untuk mengubah gerakan linier atau putaran menjadi sinyal digital, dimana sensor putaran memonitor gerakan putar dari suatu alat. *Rotary encoder* umumnya menggunakan sensor optik untuk menghasilkan pulsa yang dapat diartikan menjadi gerakan, posisi, dan arah.. Struktur *rotary encoder* dapat dilihat pada Gambar 2.15



Gambar 2.15 Keluaran Sinyal dan Susunan dari *Rotary Encoder*

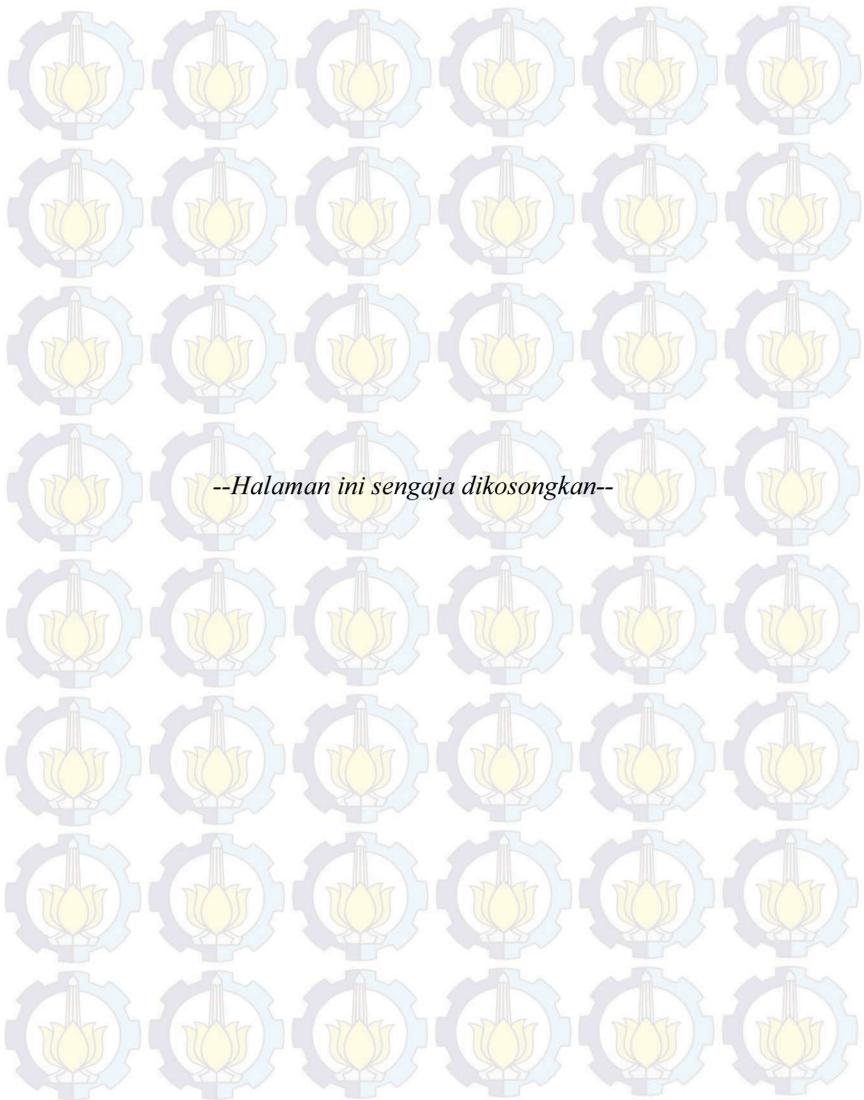
## 2.11 Arduino[7]

Arduino adalah pengendali mikro *single-board* yang bersifat *open-source*, diturunkan dari Wiring Platform, dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang. *Hardware*nya memiliki prosesor Atmel AVR dan *software*nya memiliki bahasa pemrograman sendiri. Saat ini Arduino sangat populer di seluruh dunia. Banyak pemula yang belajar mengenal robotika dan elektronika lewat Arduino karena mudah dipelajari. Bahasa yang dipakai dalam Arduino bukan *assembler* yang relatif sulit, tetapi bahasa C yang disederhanakan dengan bantuan pustaka-pustaka (*library*) Arduino. Arduino juga menyederhanakan proses bekerja dengan mikrokontroler, sekaligus menawarkan berbagai macam kelebihan antara lain:

- a. Tidak perlu perangkat *chip programmer* karena di dalamnya sudah ada *bootloader* yang akan menangani *upload program* dari komputer.
- b. Sudah memiliki sarana komunikasi USB, sehingga pengguna *notebook* yang tidak memiliki *port serial/RS232* bisa menggunakannya.
- c. Bahasa pemrograman relatif mudah karena *software* Arduino dilengkapi dengan kumpulan *library* yang cukup lengkap.
- d. Memiliki modul siap pakai (*shield*) yang bisa ditancapkan pada board Arduino. Misalnya *shield* GPS, *Ethernet*, *SD Card*, dll



**Gambar 2.16** Board Arduino UNO (kiri) dan Arduino Mega2560 (kanan)



## BAB III PERANCANGAN SISTEM

Bab III menjelaskan tentang perancangan sistem secara keseluruhan yang dibagi menjadi beberapa bagian, yaitu arsitektur sistem, perancangan *hardware*, perancangan *software*, identifikasi BLDCM, dan perancangan simulasi dan implementasi kontroler.

### 3.1 Arsitektur Sistem

Sistem yang dirancang adalah sistem penggerak elektronik dan pengaturan kecepatan dari BLDCM. Sistem yang dirancang ini mempunyai tujuan agar dapat membuat rangkaian penggerak elektronik sebagai komutasi elektrik stator untuk menggerakkan BLDCM dan menghasilkan kecepatan motor yang diinginkan. Prinsip kerja dari sistem yaitu rangkaian penggerak elektronik berfungsi untuk menyuplai tegangan AC tiga fasa ke BLDCM yang akan dioperasikan pada kecepatan tertentu sesuai dengan *set point*. Sistem pengaturan digunakan untuk mengatur kecepatan motor supaya mendekati dengan *set point*. Sinyal *feedback* dijadikan acuan terhadap sinyal referensi yang nantinya digunakan untuk mengevaluasi terhadap kontroler. Penggerak elektronik digunakan sebagai aktuator sebelum diteruskan ke *plant* BLDCM. Arduino mega2560 berfungsi sebagai *data logging*, karena pada arduino UNO sudah difungsikan sebagai pembangkit sinyal frekuensi dan sinyal PWM, akan sangat sulit bila ditambah fungsi sebagai *data logging*.

Arsitektur dari sistem ditunjukkan seperti pada Gambar 3.1.



Gambar 3.1 Arsitektur Sistem

Keterangan:

- : Sistem bagian *software*.
- : Sistem bagian *hardware*.

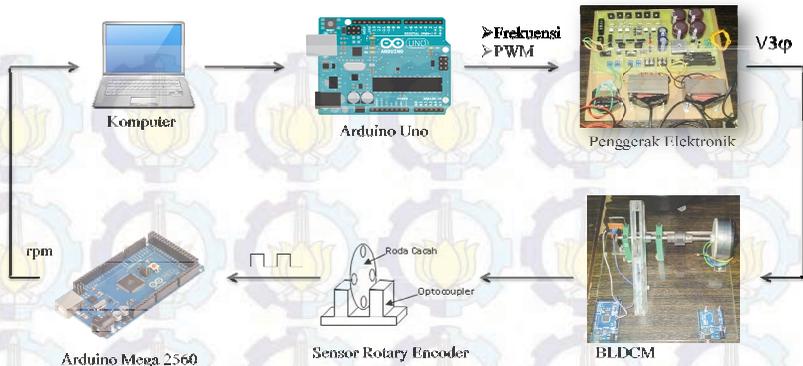
Komponen yang digunakan pada sistem ini, adalah komputer, arduino, penggerak elektronik, BLDCM, dan *rotary encoder*. Pada sistem ini, komputer sebagai HMI (*Human Machine Interface*) yang digunakan untuk mengetahui respon keluaran sistem, membuat kontroler dan juga digunakan untuk memberikan nilai *setpoint* yang diinginkan. Arduino sebagai pembangkit frekuensi fasa, pwm, dan pembacaan data dari sensor *encoder*. Penggerak elektronik sebagai aktuator yang digunakan untuk pemberian aksi dalam pengaturan kecepatan BLDCM yang telah dibuat oleh kontroler. *Rotary encoder* sebagai *feedback* berupa sensor kecepatan yang digunakan untuk mengetahui nilai *error* yang terjadi dalam sistem kontrol terhadap *set point*. Nilai *error* yang akan digunakan oleh kontroler untuk memberikan aksi kontrol hingga nilai *error* semakin kecil.

Perancangan sistem dibagi menjadi empat tahap. Tahap pertama ialah perancangan perangkat keras (*hardware*) yang meliputi perancangan mekanik *plant*, rangkaian penggerak elektronik, rangkaian sensor *rotary encoder*. Tahap kedua berupa perancangan perangkat lunak (*software*) untuk menampilkan respon dari BLDCM, juga sebagai *Human Machine Interface* (HMI) menggunakan Matlab dan pemrograman arduino. Tahap ketiga berupa identifikasi *plant* dan perancangan kontroler PID-Fuzzy.

### **3.2 Perancangan Perangkat Keras (*Hardware*)**

Perangkat keras (*Hardware*) dari sistem terdiri dari beberapa komponen, yaitu arduino, rangkaian penggerak elektronik, BLDCM, rangkaian sensor *rotary encoder*, rem magnetik, dan komputer. Perangkat keras arduino yang digunakan adalah arduino UNO dan arduino MEGA 2560. Komponen dan alur dari perangkat keras sistem secara spesifik ditunjukkan pada Gambar 3.2.

Tahap pertama dalam perancangan sistem adalah perancangan perangkat keras (*hardware*) yang meliputi perancangan mekanik *plant*, perancangan rem magnetik, perancangan sensor *rotary encoder*, dan perancangan penggerak elektronik. Alur dari perangkat keras sistem secara spesifik diperlihatkan pada Gambar 3.2.



**Gambar 3.2** Komponen dan Alur Perangkat Keras Sistem

Dari Gambar 3.2 dapat dilihat bahwa untuk menjalankan BLDCM diperlukan sebuah komputer yang berfungsi sebagai HMI. HMI digunakan untuk memonitor nilai *set point* (SP), *present value* (PV), *error*, dan parameter kontrol. Nilai *set point* diberikan melalui komputer dan selanjutnya dikirim ke arduino melalui *serial port* yang akan memberikan aksi pada penggerak elektronik *inverter*. Arduino memberikan keluaran berupa frekuensi dan PWM ke rangkaian penggerak elektronik. *Penggerak elektronik* akan mengolah frekuensi tegangan masukan dari arduino ini menjadi frekuensi 10-50 Hz tiga fasa dan pwm dengan *duty cycle* 80% dengan frekuensi 4 KHz.

Sensor *rotary encoder* digunakan untuk mengukur kecepatan BLDCM yang berputar diantara *optocoupler* tipe U. Keluaran dari sensor ini berupa pulsa dan dihubungkan ke modul arduino mega. Hasil pembacaan kecepatan dari *rotary encoder* digunakan untuk mengetahui nilai *error* yang terjadi dengan nilai *set point* yang nanti dijadikan parameter untuk mengubah sinyal kontrol.

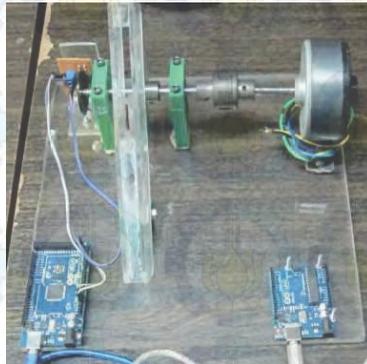
### 3.2.1 Perancangan Mekanik *Plant*

*Plant* yang digunakan dalam Tugas Akhir ini adalah BLDCM ARW31S8P30AM buatan Malaysia dengan spesifikasi yang tertera pada *name plate* seperti yang ditunjukkan pada Gambar 3.3. Motor ini juga digunakan untuk menggerakkan baling-baling dari AC (*Air Conditioner*) *Inverter* Panasonic.

**Tabel 3.1** Spesifikasi BLDCM

<i>Name Plate</i>	Keterangan
8P	Jumlah kutub yang terpasang pada rotor sebanyak 8 buah atau setara 4 pasang kutub
30W	Daya nominal yang diserap oleh BLDCM
DC280-340V	Tegangan yang diberikan pada BLDCM sebesar 280-340 VDC

*Shaft* dari BLDCM diberi piringan alumunium yang diapit rem magnetik ketika BLDCM berputar. Beban rem magnetik tersebut digunakan untuk mempengaruhi sistem kerja dari BLDCM. Pengaruh yang ditimbulkan oleh rem magnetik berupa penurunan kecepatan dari BLDCM. Bentuk dari *plant* ditunjukkan pada Gambar 3.3.



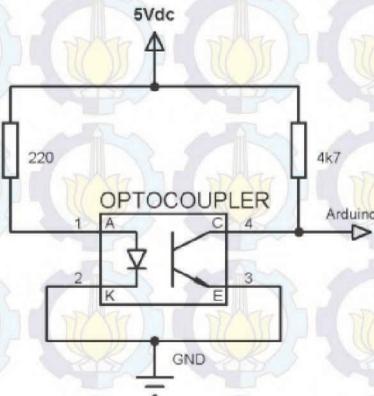
**Gambar 3.3** Kontruksi *Plant* Secara Keseluruhan

Pada Gambar 3.3 tampak bahwa *shaft* BLDCM terhubung dengan *shaft* yang terdapat piringan alumunium. Penghubung antar *shaft* menggunakan *coupling*. *Coupling* selain berguna untuk menggabungkan dua *shaft* juga berguna untuk meredam getaran yang ditimbulkan antar *shaft* saat motor berputar. Rem magnetik pada *plant* terdiri dari dua buah magnet permanen. Magnet diletakkan pada dudukan yang didesain agar bisa diubah posisinya saat motor berputar. Pada ujung *shaft* diletakkan piringan dari sensor *rotary encoder*.

### 3.2.2 Perancangan Sensor *Rotary Encoder*

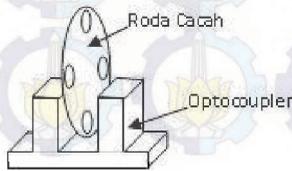
Sensor kecepatan yang digunakan pada Tugas Akhir ini adalah berupa piringan atau roda cacah dan sebuah *optocoupler* tipe U.

Piringan *rotary encoder* berputar diantara celah pada *optocoupler*. *Optocoupler* merupakan komponen *optoisolator* yang memiliki karakteristik penerima (*phototransistor*) akan mengalami perubahan logika 0 ke 1 atau sebaliknya bila terjadi perubahan intensitas cahaya yang dipancarkan oleh pemancar (LED infra merah) untuk penerima. Skema rangkaian dapat dilihat pada Gambar 3.4.



**Gambar 3.4** Skema Rangkaian Sensor *Rotary Encoder*

Gambar 3.4, merupakan rangkaian untuk *rotary encoder*. *Phototransistor* merupakan jenis transistor yang peka terhadap cahaya infra merah. Piringan akan ditempatkan di tengah dari *optocoupler* tipe U yang berfungsi untuk mempengaruhi intensitas cahaya yang diberikan oleh LED pada *optocoupler* ke *phototransistor* yang akan memberikan perubahan level logika sesuai dengan putaran piringan. Perubahan logika dari *HIGH* ke *LOW* akan dibaca oleh arduino mega2560 sebagai frekuensi pulsa dimana nantinya akan dikonversi menjadi rpm. *Konstruksi* sensor *rotary encoder* dapat dilihat pada Gambar 3.5.



**Gambar 3.5** *Konstruksi Rotary Encoder dengan Optocoupler*

### 3.2.3 Perancangan Rangkaian Penggerak Elektronik

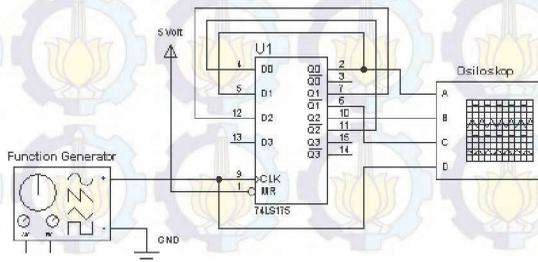
Skema perancangan penggerak elektronik di desain sebaik mungkin dan digabung menjadi satu kesatuan, dimulai dari rangkaian *power supply*, pengolah sinyal fasa dan pwm, *optocoupler*, dan bagian *switching*. Skema rangkaian terlihat pada lampiran. Bentuk fisik rangkaian penggerak elektronik terlihat pada Gambar 3.6. Rangkaian *power supply* menggunakan tegangan sebesar 12 VAC, 18 VAC, dan 64 VAC yang nantinya diubah menjadi tegangan DC sebesar 5 VDC untuk rangkaian pembangkit sinyal frekuensi, 12 VDC untuk rangkaian *optoisolator*, dan 90 VDC untuk rangkaian *switching* yang mensuplai tegangan BLDCM.



Gambar 3.6 Bentuk Fisik Rangkaian Penggerak Elektronik

#### 3.2.3.1 Rangkaian Pembagi Fasa

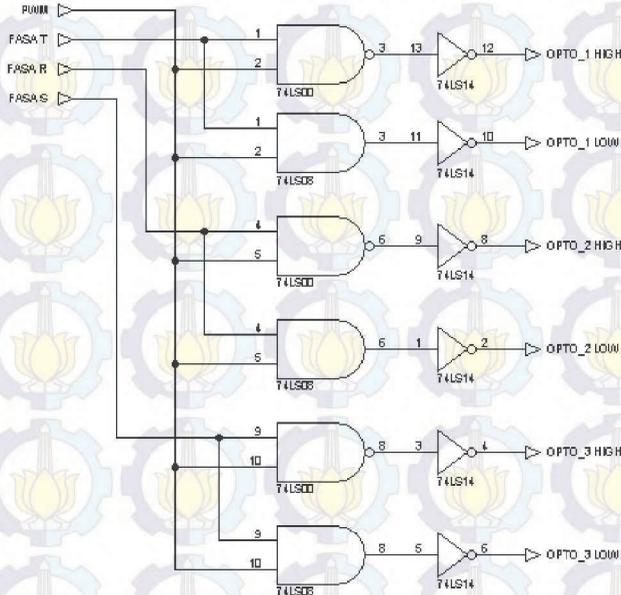
Rangkaian ini merupakan menggunakan IC 74LS175. Masukan dari rangkaian ini berupa sinyal frekuensi fasa, sedangkan keluaran dari rangkian pembagi fasa berupa sinyal kotak tiga fasa. Skema rangkaian dapat dilihat pada Gambar 3.7.



Gambar 3.7 Skema Rangkaian Pembagi Fasa

### 3.2.3.2 Rangkaian Pengolah Sinyal Fasa dan PWM

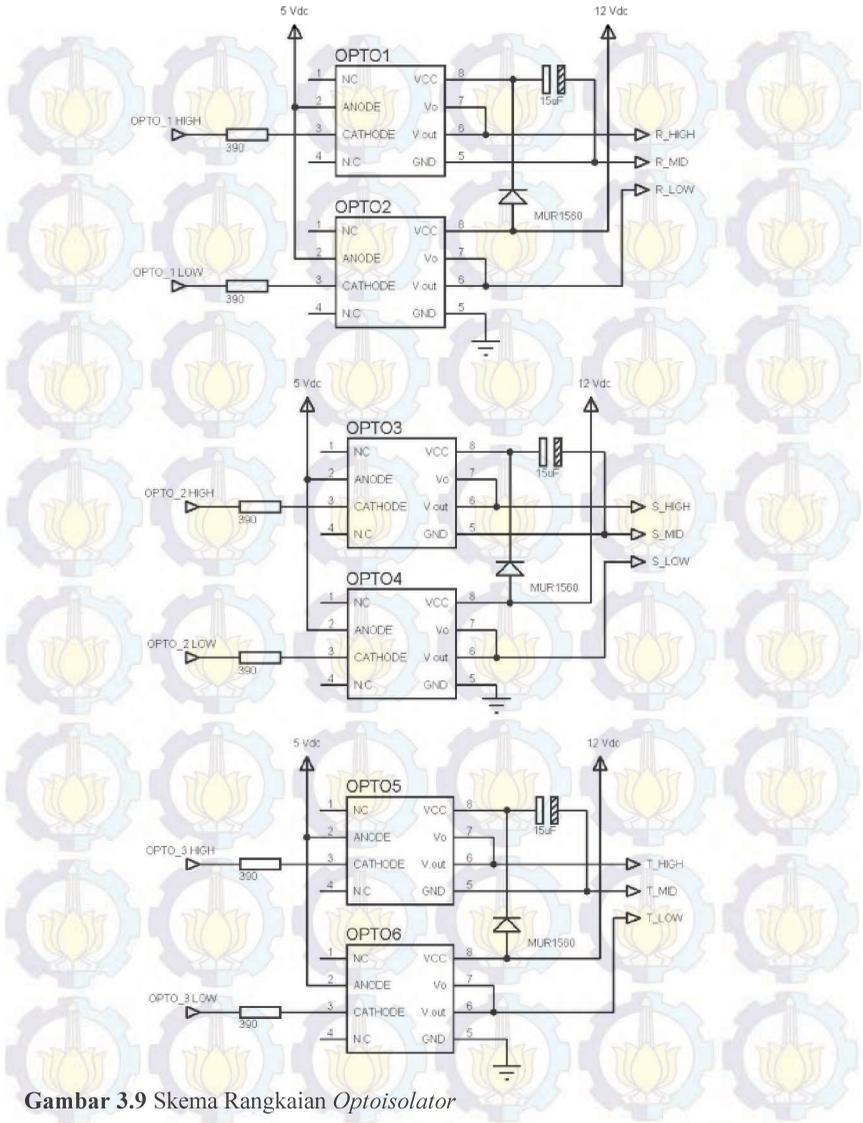
Rangkaian ini terdiri dari rangkaian gerbang logika AND, NAND, dan NOT. Rangkaian ini berfungsi untuk memasukkan sinyal PWM kedalam sinyal frekuensi. Skema rangkaian dapat dilihat pada Gambar 3.8.



Gambar 3.8 Skema Rangkaian Pengolah Sinyal Fasa dan PWM

### 3.2.3.3 Rangkaian Optoisolator

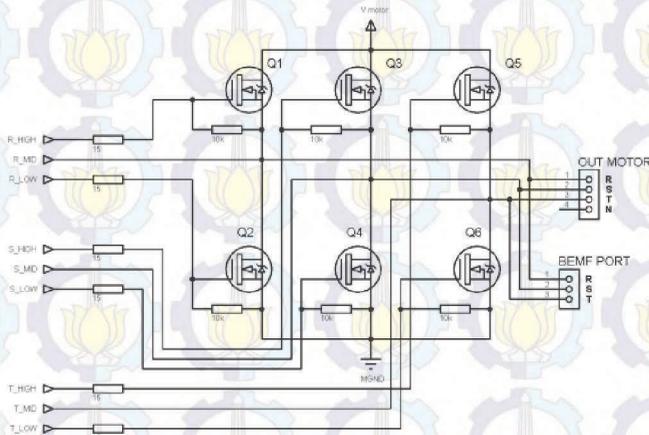
Rangkaian ini terdiri dari *optocoupler* tipe TLP 250, dimana fungsinya sebagai pemisah antara rangkaian pembangkit sinyal dengan rangkaian *switching*. Hal ini diperlukan untuk menghindari terjadinya hubung singkat apabila pada rangkaian *switching* mengalami hubung singkat. Skema rangkaian dapat dilihat pada Gambar 3.9. Pemilihan komponen *optocoupler* harus melihat kemampuan *switching* berdasarkan *datasheet* yang tersedia, hal ini dikarenakan pada rangkaian penggerak elektronik juga menggunakan sinyal masukkan PWM yang berfrekuensi tinggi.



**Gambar 3.9** Skema Rangkaian *Optoisolator*

### 3.2.3.4 Rangkaian Switching

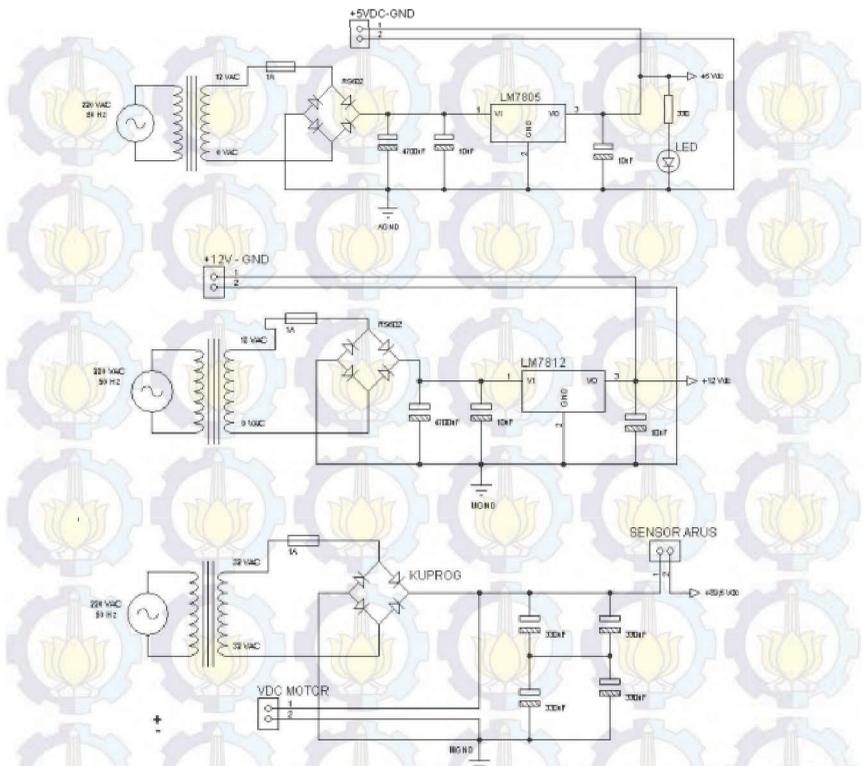
Rangkaian ini terdiri dari enam buah MOSFET yang digunakan untuk *switching* dari tegangan DC menjadi tegangan AC 3 fasa. Penggunaan resistor untuk membatasi arus yang masuk ke kaki *gate* MOSFET, sedangkan resistor *gate-source* MOSFET digunakan untuk mempercepat transisi dari keadaan *HIGH* menjadi keadaan *LOW*. Skema rangkaian *switching* dapat dilihat pada Gambar 3.10.



Gambar 3.10 Skema Rangkaian Switching

### 3.2.3.5 Rangkaian Power Supply

Rangkaian ini merupakan catu daya yang sangat diperlukan oleh rangkaian penggerak elektronik. Rangkaian ini terdiri dari IC *regulator* LM7805 dan LM 7812, kapasitor, dioda *bridge*, dan *fuse*. Selain itu diperlukannya komponen trafo CT (*Center Tap*) dan trafo *non-CT*. Trafo CT digunakan untuk *supply* daya BLDCM, trafo *non-CT* digunakan untuk *supply* 5Vdc dan 12 Vdc. Catu daya yang dibutuhkan yaitu catu daya 5 Vdc untuk rangkaian pembagi fasa, pengolah sinyal fasa dan PWM. Catu daya 12 Vdc digunakan oleh rangkaian *optoisolator*. Catu daya 89,6 Vdc sebagai sumber tegangan MOSFET *HIGH*. Skema rangkaian dapat dilihat pada Gambar 3.11.



Gambar 3.11 Skema Rangkaian Power Supply

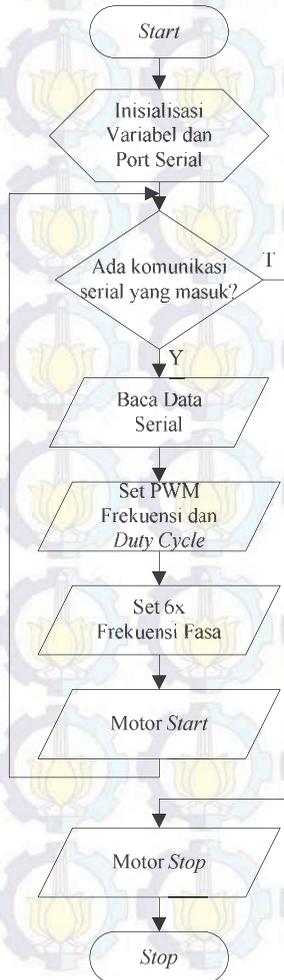
### 3.3 Perangkat Perangkat Lunak (Software)

Perangkat (*software*) yang digunakan dalam Tugas Akhir ini antara lain *program* pembangkit frekuensi dan *program* baca kecepatan yang dilakukan oleh mikrokontroler arduino. Alur pemrograman sistem secara keseluruhan.

#### 3.3.1 Arduino sebagai Pembangkit Frekuensi dan PWM

Arduino UNO digunakan sebagai pembangkit frekuensi dimana perintah untuk membangkitkan frekuensi dilakukan oleh komputer dengan menggunakan *software Matlab*. *Flowchart* pemrograman Arduino dapat dilihat pada Gambar 3.12. Frekuensi yang dibangkitkan

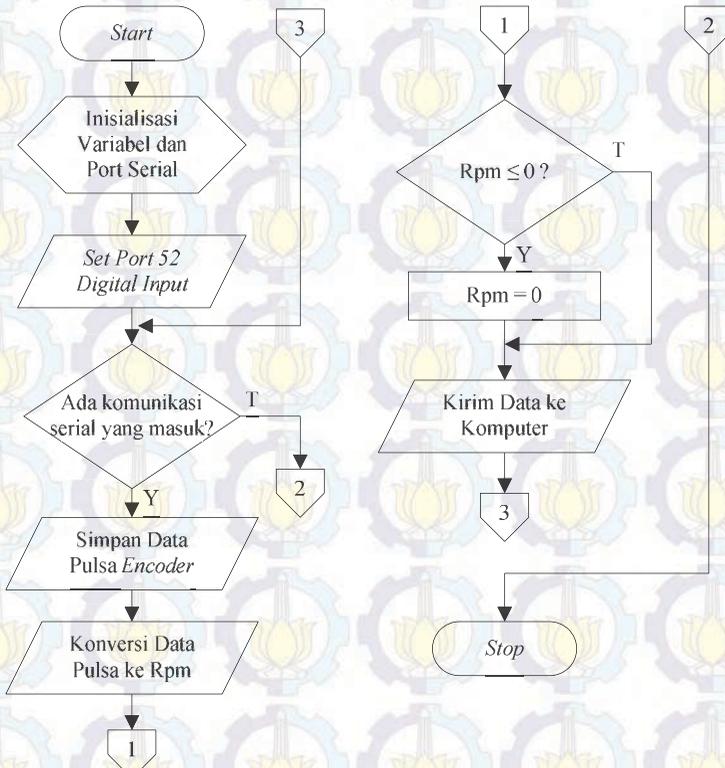
oleh arduino merupakan 6x frekuensi fasa, hal ini diperlukan karena spesifikasi *hardware* pada *driver inverter* menggunakan rangkaian pembagi fasa. *Output* IC pembagi fasa (74HC175) berupa sinyal kotak antar fasa dimana sudut antar fasa sebesar  $120^\circ$ . PWM digunakan untuk memperhalus putaran motor dan mengurangi disipasi daya yang diserap oleh BLDCM.



**Gambar 3.12** *Flowchart* Pemrograman Pembangkit PWM dan Frekuensi

### 3.3.2 Arduino sebagai Pembaca Data Sensor

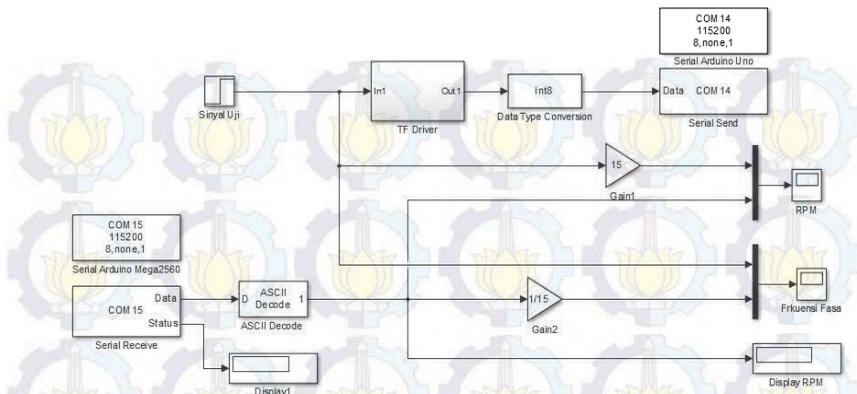
Arduino Mega2560 digunakan sebagai pembaca data sensor kecepatan. Proses pembacaan data akan dikirim ke komputer dengan satuan rpm. Berikut *flowchart* pemrogramannya ada pada Gambar 3.13.



Gambar 3.13 *Flowchart* Pembacaan Data Sensor Kecepatan

### 3.3.3 Perancangan Metode Identifikasi

Perancangan simulink menggunakan Matlab R2014a, perancangan simulink blok diagram ditujukan untuk mengidentifikasi *plant* sebelum dilakukan pengaturan pada *plant*. Koneksi antara simulink dengan arduino menggunakan blok diagram Serial sebagai pengirim dan penerima data dapat dilihat pada Gambar 3.14.



**Gambar 3.14** Blok Diagram Simulink Identifikasi

### 3.4 Identifikasi Plant

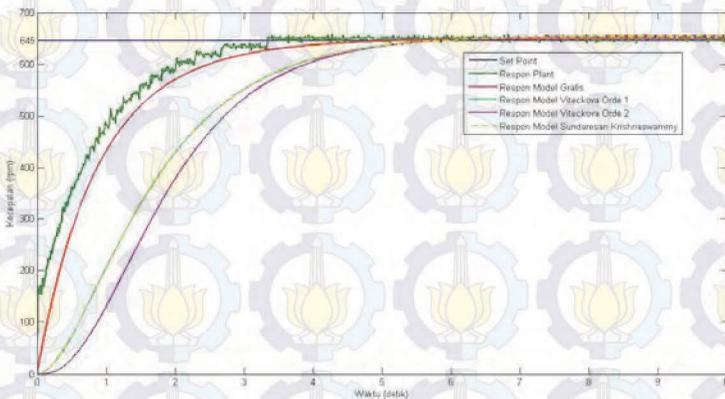
Identifikasi sistem diperlukan untuk mendapatkan model matematika dari BLDCM. Pada Tugas Akhir ini, identifikasi dilakukan adalah identifikasi statis dengan melihat keluaran respon berupa kecepatan terhadap referensi yang diberikan. Sinyal uji *step* diberikan melalui Arduino dengan member nilai *set point* sebesar 43 Hz atau sama dengan 645 rpm. Hasil respon *plant* yang terbaca oleh *rotary encoder* akan ditampilkan di simulink melalui blok diagram *Serial Receive*. Data kecepatan juga dan disimpan dalam bentuk *workspace* dengan *time sampling* 0,001 detik.

#### 3.4.1 Metode Identifikasi

Metode identifikasi diperlukan untuk mencari respon model yang paling mendekati respon *plant*. Terdapat tujuh metode identifikasi yang dapat digunakan untuk sistem tanpa osilasi dengan masukan respon *step*. Dari ketujuh metode tersebut digunakan 4 buah metode, yaitu metode Viteckova 1<sup>st</sup> Order, Viteckova 2<sup>nd</sup> Order, Sundaresan – Krishnaswamy dan metode Grafis. Keempat metode tersebut dicari RMSE yang paling kecil yang akan mendekati dengan model *plant* BLDCM. Validasi model beban minimal dapat dilihat pada Tabel 3.2. Respon model beban minimal dapat dilihat pada Gambar 3.15.

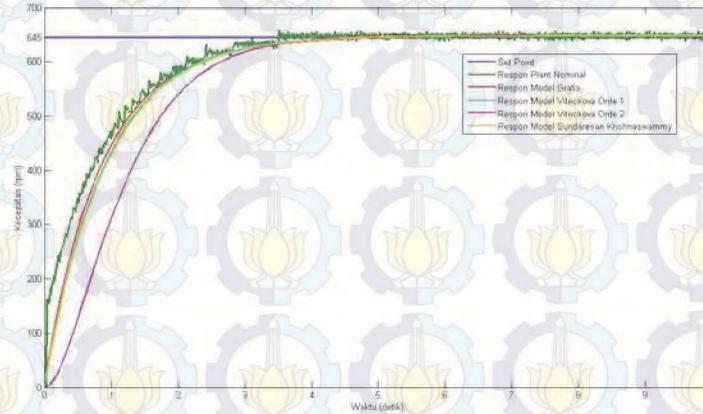
**Tabel 3.2** Validasi Model Matematika Beban Minimal

No	Metode	Model Matematika	RMSE
1.	Viteckova 1 <sup>st</sup> Order	$\frac{1,008}{0,859s + 1}$	116,138
2.	Viteckova 2 <sup>nd</sup> Order	$\frac{1,008}{0,299s^2 + 1,094s + 1}$	143,336
3.	Sundaresan & Krishnaswamy	$\frac{1,008}{0,864s + 1}$	116,527
4.	Grafis	$\frac{1,008 (0,935s + 1)}{0,817s^2 + 1,881s + 1}$	29,587

**Gambar 3.15** Respon Model Matematika Beban Minimal**Tabel 3.3** Validasi Model Matematika Beban Nominal

No	Metode	Model Matematika	RMSE
1.	Viteckova 1 <sup>st</sup> Order	$\frac{1,005}{0,9s + 1}$	24,339
2.	Viteckova 2 <sup>nd</sup> Order	$\frac{1,005}{0,383s^2 + 1,238s + 1}$	63,362
3.	Sundaresan & Krishnaswamy	$\frac{1,005}{0,918s + 1}$	25,660
4.	Grafis	$\frac{1,005 (0,881s + 1)}{0,722s^2 + 1,761s + 1}$	20,880

Pada Gambar 3.15, metode Viteckova 2<sup>nd</sup> Order memiliki simpangan terjauh, sehingga model matematika beban minimal menggunakan metode Grafis. Validasi beban nominal dapat dilihat pada Tabel 3.2 dan respon model beban nominal dapat dilihat pada Gambar 3.16.

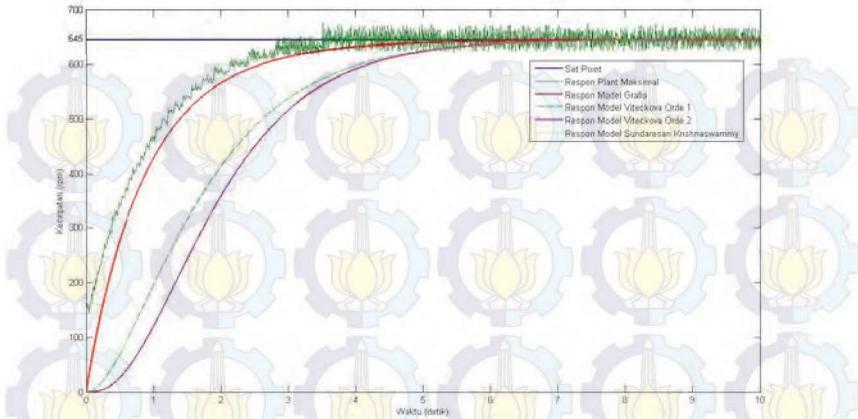


**Gambar 3.16** Respon Model Matematika Beban Nominal

Pada Gambar 3.16, respon model metode Grafis dipilih karena memiliki nilai RMSE paling rendah atau respon paling mendekati respon *plant* nominal. Validasi model matematika beban maksimal dapat dilihat pada Tabel 3.4 dan respon model matematika beban maksimal dapat dilihat pada Gambar 3.17.

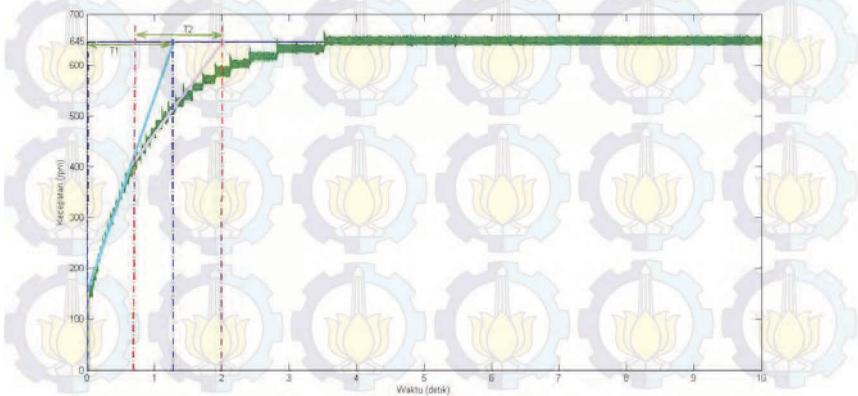
**Tabel 3.4** Validasi Model Matematika Beban Maksimal

No	Metode	Model Matematika	RMSE
1.	Viteckova 1 <sup>st</sup> Order	$\frac{1,002}{0,908s + 1}$	118,098
2.	Viteckova 2 <sup>nd</sup> Order	$\frac{1,002}{0,383s^2 + 1,238s + 1}$	146,069
3.	Sundaresan & Krishnaswamy	$\frac{1,002}{0,911s + 1}$	118,326
4.	Grafis	$\frac{1,002 (0,956s + 1)}{0,836s^2 + 1,91s + 1}$	29,950



**Gambar 3.17** Respon Model Matematika Beban Maksimal

Metode Grafis dipilih karena memiliki validasi nilai RMSE yang terkecil. Hasil respon dari pendekatan Grafis berupa respon orde 2. Hasil dari tiap perhitungan dari beberapa metode yang terdapat dalam tabel, disimulasikan pada *software* Matlab. Hal ini dilakukan untuk dapat melihat perbandingan hasil dari perhitungan dengan data hasil respon sebenarnya. Hasil grafik respon pada Matlab didapatkan bahwa pendekatan model matematika dengan metode Grafis yang paling mendekati data yang sebenarnya.



**Gambar 3.18** Penarikan Garis Singgung pada Metode Grafis

Perhitungan dalam menentukan pendekatan model matematika dengan metode Grafis adalah sebagai berikut:

- Plot hasil respon *plant*
- Tarik garis singgung yang memotong kurva respon *plant*, kemudian hitung  $\tau_{s1}$  dan  $\tau_{s2}$ , seperti Gambar 3.20
- Penurunan rumus model matematika seperti di bawah ini.

$$G(s) = K \left( \frac{1/1,17}{s+1,17} + \frac{1/0,65}{s+0,65} \right) \quad (3.1)$$

Mula-mula gain K bernilai 1.

- Menentukan nilai gain (K)  
Gain (K) bisa didapat apabila setelah diberi sinyal uji *unit step* sesuai dengan *set point* (SP) dan melihat hasil respon melalui *scoope*. Setelah melihat hasil respon, maka nilai dari K didapat melalui rumus

$$K = Y_{ss} \left( \frac{SP}{Y_{ss}} \right) = 1289,8513 \left( \frac{648,3006}{1289,8513} \right) = 0,502 \quad (3.2)$$

- Fungsi alih keseluruhan

$$G(s) = 1,005 \left( \frac{0,881s + 1}{0,722s^2 + 1,761s + 1} \right) \quad (3.3)$$

### 3.4.2 Metode Pembebanan *Plant*

Pembebanan diperlukan untuk menentukan model matematika dari suatu *plant* dalam kondisi yang berbeda. Perubahan parameter pada *plant* dapat terjadi akibat adanya pembebanan yang diberikan pada sistem. Pada Tugas Akhir ini, rem magnetik digunakan sebagai beban pada BLDCM. Pembebanan dibagi ke dalam 3 kondisi berdasarkan besar posisi pembebanan rem magnetik, yaitu beban minimal, beban nominal atau setengah dari permukaan rem magnetik diberikan terhadap piringan motor, dan beban maksimal dengan posisi rem magnetik diberikan full terhadap piringan motor. Pengambilan data dilakukan ketika BLDCM berputar pada *setpoint* 645 rpm. Penentuan model matematika didasarkan pada tiga kondisi yaitu kondisi beban minimal,

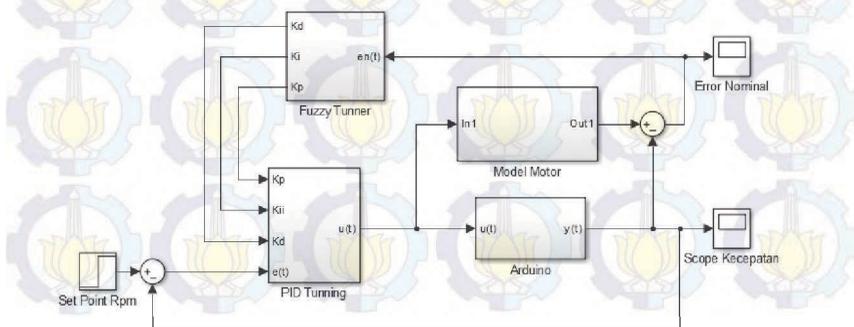
nominal, dan maksimal. Beban minimal dipilih ketika motor diberi beban 0 atau tidak diberi efek pembebanan rem magnetik menghasilkan *output steady state* sebesar 650,22 rpm. Pada kondisi beban nominal, posisi rem magnetik terbenam separuhnya pada piringan alumunium yang menghasilkan *output steady state* sebesar 648,3 rpm. Pada kondisi beban maksimal, posisi rem magnetik terbenam sepenuhnya pada piringan alumunium yang menghasilkan *output steady state* sebesar 646,514 rpm. Model matematika kondisi beban pada *plant* BLDCM dapat dilihat pada Tabel 3.5.

**Tabel 3.5** Model Matematika Beban BLDCM

No	Kondisi Pembebanan	Model Matematika
1.	Beban Minimal	$\frac{1,008 (0,935s+1)}{0,817s^2+1,881s+1}$
2.	Beban Nominal	$\frac{1,005 (0,881s+1)}{0,722s^2+1,761s+1}$
3.	Beban Maksimal	$\frac{1,002 (0,956s+1)}{0,836s^2+1,91s+1}$

### 3.5 Perancangan Kontroler PID-Fuzzy

Pembebanan nominal dianggap sebagai keadaan ketika motor bekerja pada kondisi yang seharusnya atau kerja BLDCM pada kondisi yang normal. Mekanisme kontroler PID-Fuzzy ditunjukkan pada Gambar 3.19.



**Gambar 3.19** Diagram Blok Kontroler PID-Fuzzy

Blok Pemodelan motor merupakan blok pemodelan motor dalam keadaan nominal atau *transfer function* BLDCM yang didapat saat melakukan identifikasi. Blok Arduino yang terdiri dari *Serial Send* dan *Serial Receive* digunakan sebagai *real plant* untuk bias memberikan sinyal kontrol pada arduino dimana yang nantinya dikonversi menjadi frekuensi fasa dan PWM untuk keperluan *driver* BLDCM.

### 3.5.1 Perancangan Kontroler PID

Perancangan kontroler PID dilakukan secara analitik dari parameter model matematika yang diperoleh dari hasil identifikasi. Parameter model matematik tersebut digunakan untuk menentukan nilai  $K_p$ ,  $K_i$  dan  $K_d$ . Nilai  $K_p$ ,  $K_i$ , dan  $K_d$  yang diperoleh memiliki nilai yang berbeda-beda sesuai dengan perubahan pada nilai parameter model matematika. Hasil perancangan kontroler PID pada kondisi pembebanan yang berbeda akan digunakan sebagai keluaran dari *fuzzy* yang dijadikan sebagai parameter pada kontroler PID. Persamaan model matematika *plant* secara umum diberikan sebagai berikut:

$$G(s) = \frac{K(\tau s + 1)}{\frac{1}{\omega n} s^2 + \frac{2\zeta}{\omega n} s + 1} \quad (3.4)$$

Langkah awal dalam menentukan nilai parameter PID adalah menentukan nilai  $\omega n$  dan  $\zeta$  dari Persamaan 3.4. Sebelum itu terlebih dahulu menentukan spesifikasi sistem,  $\tau_s$  (5%) sebesar 2 detik,  $e_{ss}$  sebesar 0%, dan *overshoot*  $\leq 5\%$ . Langkah selanjutnya adalah menentukan nilai  $\tau$  yang diinginkan ( $\tau^*$ ) dan dalam perancangan ini nilai  $\tau^* = 2/3$  detik. Langkah selanjutnya untuk menentukan nilai PID dapat dipilih jika  $\tau_{d2} = \tau$ ,  $\tau_i + \tau_{d2} = \frac{2\zeta}{\omega n}$ , dan  $\tau_i(\tau_{d1} + \tau_{d2}) = \frac{1}{\omega n^2}$ , maka dapat dirumuskan:

$$\tau_{d2} = \tau \quad (3.5)$$

$$\tau_{d1} = \frac{(1/\omega n)^2}{\tau_i} - \tau_{d2} \quad (3.6)$$

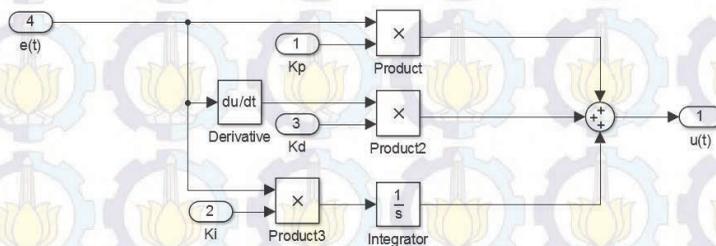
$$\tau_i = \frac{2\zeta}{\omega n} - \tau \quad (3.7)$$

$$K_p = \frac{\tau_i}{\tau * K.} \quad (3.8)$$

$$K_i = \frac{K_p}{\tau_i.} \quad (3.9)$$

$$K_d = K_p * \tau_{d1} \quad (3.10)$$

Perhitungan parameter PID yang telah dilakukan sesuai dengan persamaan di atas, pada setiap kondisi pembebanan diperoleh hasil parameter kontroler seperti yang ditunjukkan pada Tabel 3.6 dengan blok diagram pada Gambar 3.20.



**Gambar 3.20** Blok Diagram Kontroler PID

**Tabel 3.6** Parameter PID pada Kondisi Pembebanan

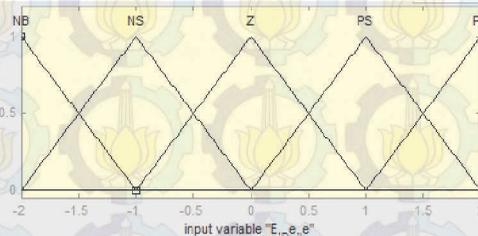
No	Kondisi Pembebanan	Kp	Ki	Kd
1.	Beban Minimal	1,313	1,488	-0,099
2.	Beban Nominal	1,407	1,492	-0,787
3.	Beban Maksimal	1,430	1,499	-0,110

### 3.5.2 Perancangan Mekanisme PID-Fuzzy

Pada dasarnya kontroler PID hanya dapat digunakan pada *plant* dengan satu fungsi alih saja, sehingga tidak dapat digunakan pada keadaan dengan fungsi alih yang berbeda. Hal ini yang menjadi dasar dalam penggunaan *fuzzy*. Data masukan yang digunakan dalam hal ini berupa *error* nominal dan data keluaran berupa nilai  $K_p$ ,  $K_i$ , dan  $K_d$ .

### 3.5.2.1 Fungsi Keanggotaan Masukan

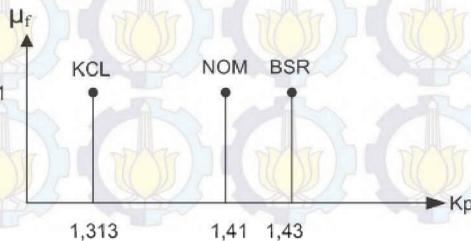
Fungsi keanggotaan masukan *fuzzy* berupa data *error* yang didapatkan dari selisih hasil *output plant* nominal dengan *plant* berbeban. Jumlah fungsi keanggotaan *fuzzy* yang digunakan pada sebanyak lima. Nilai linguistik dari fungsi keanggotaan *fuzzy* didefinisikan dengan angka -2, -1, 0, 1, dan 2. Tiap angka yang ada mewakili beberapa kondisi pembebanan dari keadaan beban 1 sampai kondisi beban 9. Fungsi keanggotaan *fuzzy* yang telah ternormalisasi dengan gain normalisasi  $\Delta error$  sebesar 2/0,13, gain normalisasi *error* sebesar 2/0,128 dan gain normalisasi *integral error* sebesar 2/1,28, sehingga dapat diskala masuk kedalam anggota fuzzifikasi Gambar 3.21.



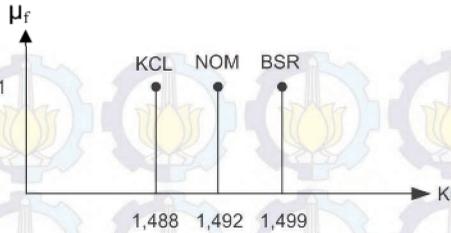
Gambar 3.21 Fungsi Keanggotaan *Fuzzy* untuk *Error*, *Delta error*, dan *Integral Error*

### 3.5.2.2 Fungsi Keanggotaan Keluaran

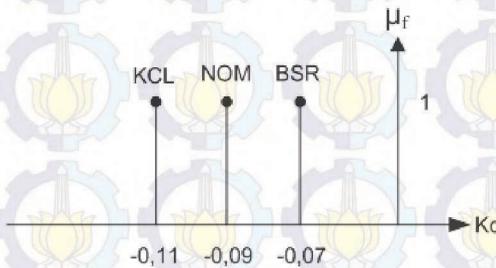
Fungsi keanggotaan keluaran *fuzzy* berupa nilai  $K_p$ ,  $K_i$ , dan  $K_d$  yang diperoleh dari perhitungan pada tiap pembebanan. Pada Tugas Akhir ini *fuzzy* yang digunakan adalah *fuzzy* Mamdani. *Fuzzy* mamdani menggunakan fungsi keanggotaan *triangular* yang terdiri dari tiga fungsi keanggotaan keluaran yaitu  $K_p$ ,  $K_i$ , dan  $K_d$ .



Gambar 3.22 Fungsi Keanggotaan  $K_p$



Gambar 3.23 Fungsi Keanggotaan  $K_i$



Gambar 3.24 Fungsi Keanggotaan  $K_d$

### 3.5.2.3 Rule Base

Perancangan *rule base* untuk kontroler *fuzzy* diperlukan agar kontroler *fuzzy* dapat bekerja pada *plant*. Berdasarkan fungsi keanggotaan masukan berupa data arus pada beban dan keluaran berupa nilai parameter kontroler PID, diperoleh beberapa *rule base* untuk  $K_p$ ,  $K_i$ , dan  $K_d$  dengan aturan linguistik sebagai berikut:

IF (Error is "NB") THEN ( $K_p$  is "KCL")

IF (Error is "NS") THEN ( $K_p$  is "KCL")

IF (Error is "Z") THEN ( $K_p$  is "NOM")

IF (Error is "PS") THEN ( $K_p$  is "BSR")

IF (Error is "PB") THEN ( $K_p$  is "BSR")

IF ( $\Sigma$ error is "NB") THEN ( $K_i$  is "KCL")

IF ( $\Sigma$ error is "NS") THEN ( $K_i$  is "KCL")

IF ( $\Sigma$ error is "Z") THEN ( $K_i$  is "NOM")

IF ( $\Sigma$ error is "PS") THEN ( $K_i$  is "BSR")

IF ( $\Sigma$ error is "PB") THEN ( $K_i$  is "BSR")

IF ( $\Delta error$  is "NB") THEN ( $K_d$  is "KCL")  
 IF ( $\Delta error$  is "NS") THEN ( $K_d$  is "KCL")  
 IF ( $\Delta error$  is "Z") THEN ( $K_d$  is "NOM")  
 IF ( $\Delta error$  is "PS") THEN ( $K_d$  is "BSR")  
 IF ( $\Delta error$  is "PB") THEN ( $K_d$  is "BSR")

### 3.5.2.4 Defuzifikasi

Defuzifikasi digunakan untuk mengubah hasil dari inferensi *fuzzy* menjadi variabel nyata, atau diubah menjadi nilai nyata yang berupa nilai tunggal. Metode defuzifikasi yang menggunakan defuzifikasi COA (*Centre of Area*). Defuzifikasi untuk mendapatkan nilai  $K_p$ ,  $K_i$ , dan  $K_d$  ditunjukkan pada Persamaan 3.11.

$$U_0 = \frac{\sum_{k=1}^m u_k(T) * \mu_u(u_k(T))}{\sum_{k=1}^m \mu_u(u_k(T))} \quad (3.11)$$

## 3.6 Perancangan Simulasi Kontroler

Simulasi merupakan salah satu tahap yang dapat menentukan keberhasilan dari kontroler yang telah didesain sebelum kontroler diimplimentasikan pada *plant*. Pada Tugas Akhir ini, simulasi dari kontroler menggunakan simulink pada Matlab. Diagram blok simulink dari simulasi dapat dilihat pada Gambar 3.25.



Gambar 3.25 Diagram Blok Simulasi PID-Fuzzy pada Simulink

Berdasarkan Gambar 3.25 di atas, *set point* yang digunakan dalam simulasi berupa kecepatan (rpm). Hasil keluaran dari simulasi di atas berupa kecepatan (rpm).

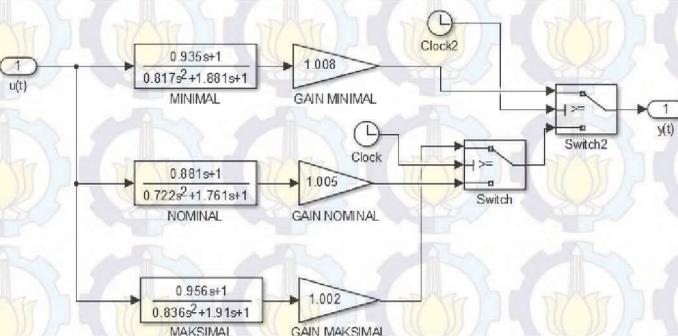
### 3.6.1 Blok Model *Plant* Beban Nominal

Blok ini berisi model pendekatan matematika dari *plant* yang nilainya didapatkan dari perhitungan yang telah dilakukan sebelumnya. Pada blok model *plant* ini berisi model fungsi alih dari BLDCM pada Persamaan 3.12:

$$\frac{1,005(0,881s+1)}{0,722s^2+1,761s+1} \quad (3.12)$$

### 3.6.2 Blok Model Pembebanan *Plant*

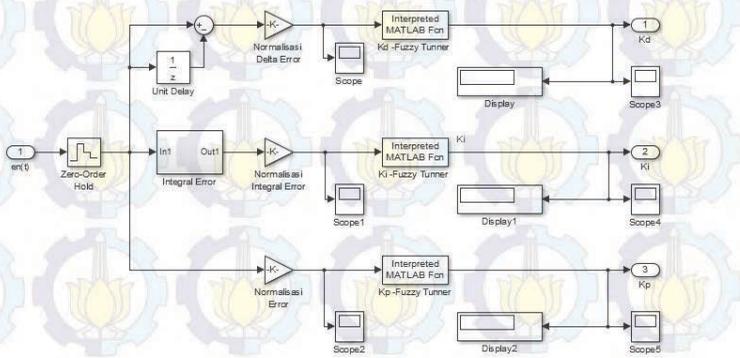
Blok ini berisi model pendekatan matematika dari *plant* yang nilainya didapatkan dari perhitungan yang telah dilakukan sebelumnya. Pada blok model *plant* ini, nilai model pendekatan matematika pada *plant* berubah sesuai dengan adanya perubahan beban. Nilai perubahan parameter model matematika *plant* pada blok ini didapatkan dari hasil perhitungan blok estimasi parameter *plant*. Model matematika pembebanan yang didapat dari hasil identifikasi akan dijadikan sebagai blok model pembebanan pada perancangan simulasi dengan selang waktu antara beban nominal, maksimal dan beban minimal selama 10 detik pada Gambar 3.26.



Gambar 3.26 Blok Model Pembebanan *Plant*

### 3.6.3 Blok *Fuzzy Tuner*

Blok *Fuzzy Tuner* digunakan untuk mengubah nilai parameter kontroler PID agar kontroler dapat beradaptasi terhadap setiap perubahan pada beban. Simulasi untuk PID-*Fuzzy* ini juga digunakan untuk mengetahui mekanisme perubahan parameter PID. Masukan dari blok ini merupakan *error nominal*. Diagram blok untuk PID-*Fuzzy* untuk parameter  $K_p$ ,  $K_i$ , dan  $K_d$  dapat dilihat pada Gambar 3.27.



Gambar 3.27 Diagram Blok *Fuzzy Tuner*



## BAB IV PENGUJIAN DAN ANALISA

Bab IV menjelaskan tentang hasil pengujian dan analisa dari perancangan sistem yang telah dilakukan dan dijelaskan pada bab sebelumnya. Bab ini terdiri pengujian rangkaian penggerak elektronik, pengujian sensor, pengujian kontroler secara simulasi pada *Software Matlab*, dan implementasi kontroler *PID-Fuzzy*. Pengujian ini dilakukan untuk mengetahui respon keluaran dari sistem dan juga untuk mengetahui proses kerja dari kontroler yang didesain.

### 4.1 Pengujian Arduino

Pengujian pada Arduino meliputi pengujian sinyal PWM dan pengujian pembangkit sinyal fasa.

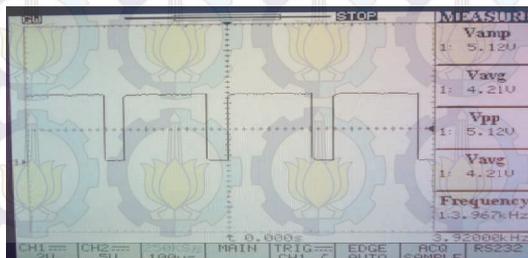
#### 4.1.1 Analisa Pengujian Pembangkit Sinyal PWM

Pengujian pembangkit sinyal PWM pada arduino dengan mekanisme yang terlihat pada Gambar 4.1.



**Gambar 4.1** Mekanisme Pengujian Sinyal PWM

Sebelum dimulai pengujian, terlebih dahulu arduino di program untuk membangkitkan sinyal PWM dengan frekuensi 3,9 KHz dengan *duty cycle* 80%. Hasil pengujian dapat dilihat pada Gambar 4.2.



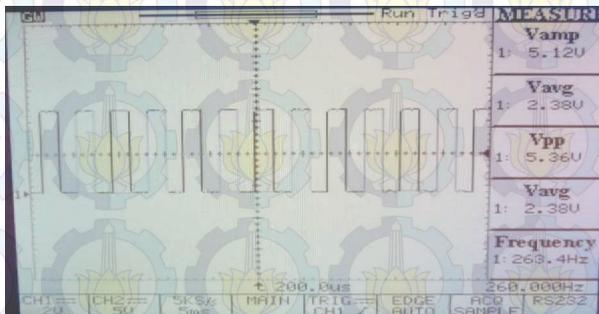
**Gambar 4.2** Hasil Pengujian Sinyal PWM

#### 4.1.2 Analisa Pengujian Pembangkit Sinyal Fasa

Arduino diprogram untuk membangkitkan sinyal fasa sebesar enam kali frekuensi fasa. Hal ini dilakukan karena pada rangkaian penggerak elektronik menggunakan IC (*Integrated Circuit*) pembagi fasa 74HC175. Mekanisme pengujian sinyal pembangkit fasa oleh arduino terlihat pada Gambar 4.1. Frekuensi fasa yang dibangkitkan dengan nilai *set point* 43 Hz. Frekuensi yang akan dibangkitkan oleh arduino sebesar enam kali dari nilai frekuensi *set point*. Hasil pengujian pembangkit sinyal fasa dapat dilihat pada Gambar 4.3 dan Tabel 4.1.

**Tabel 4.1** Hasil Pengujian Pembangkit Frekuensi Fasa Arduino

Frekuensi Fasa (Hz)	Perhitungan 6x Frekuensi Fasa (Hz)	Pembangkit Frekuensi Arduino (Hz)	Error (%)
10	60	56	-6,666
20	120	119	-0,833
30	180	181	0,555
40	240	241	0,416
43	260	262	0,769
45	270	273	1,111
50	300	301	0,333
55	330	333	0,909



**Gambar 4.3** Hasil Pengujian Sinyal Frekuensi Fasa

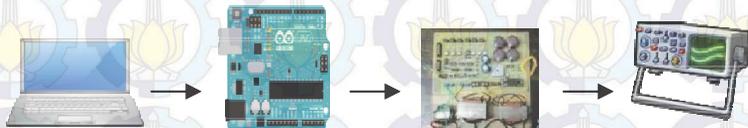
Pada Gambar 4.3, frekuensi yang dibangkitkan sebesar 263,4 Hz. Terjadi *error* sekitar 2,09% dari frekuensi seharusnya, yaitu sebesar 258 Hz.

## 4.2 Pengujian Rangkaian Penggerak Elektronik

Pengujian rangkaian penggerak elektronik meliputi pengujian keluaran gerbang AND dan NAND, pengujian keluaran *optocoupler*.

### 4.2.1 Analisa Pengujian Keluaran Gerbang AND dan NAND

Pengujian keluaran AND dan NAND dilakukan dengan melihat sinyal keluaran. Pengujian ini dilakukan untuk melihat bentuk sinyal keluaran dari gerbang AND dan NAND yang berbentuk kotak dengan PWM. Sinyal ini berfungsi untuk menggerakkan *optocoupler* sebelum masuk ke rangkaian *switching*. Mekanisme pengujian dapat dilihat pada Gambar 4.4 dan hasilnya dapat dilihat pada Gambar 4.5.



Gambar 4.4 Mekanisme Pengujian Sinyal gerbang AND dan NAND



Gambar 4.5 Sinyal Keluaran AND (kiri) dan NAND (kanan)

### 4.2.2 Analisa Pengujian Keluaran *Optocoupler*

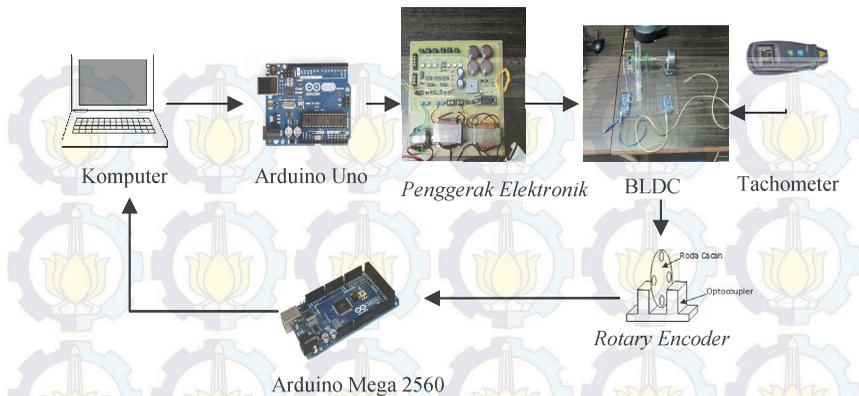
Pengujian keluaran *optocoupler* digunakan untuk melihat sinyal yang dihasilkan oleh *optocoupler* yang akan digunakan oleh rangkaian *switching*. Keluaran sinyal *optocoupler* bergantung dari jenis masukannya. *Optocoupler* ini untuk memisahkan antara *ground* kontrol dengan *ground* BLDCM untuk menghindari terjadinya hubung singkat apabila salah satu terjadi gangguan. Mekanisme pengujian sinyal *optocoupler* dapat dilihat pada Gambar 4.4. Hasil sinyal keluaran *optocoupler* dapat dilihat pada Gambar 4.6.



**Gambar 4.6** Sinyal Keluaran *Optocoupler HIGH* (kiri) dan *LOW* (kanan)

Pada Gambar 4.6, sinyal keluaran serupa dengan keluaran gerbang AND dan NAND. Sinyal bagian *HIGH* digunakan untuk menggerakkan MOSFET bagian atas yang menghasilkan sinyal sinus bagian positif, sedangkan sinyal bagian *LOW* digunakan untuk menggerakkan MOSFET bagian bawah yang menghasilkan sinyal sinus bagian negatif. Sinyal tersebut merupakan sinyal frekuensi fasa yang mengandung sinyal PWM di dalamnya.

Pengujian ini dilakukan untuk mengetahui dan mengkalibrasi kecepatan hasil dari pembacaan sensor *rotary encoder* BLDCM dengan kecepatan hasil pembacaan *tachometer*. Pengujian dilakukan dengan menghubungkan BLDCM dengan sumber tiga fasa dari rangkaian penggerak elektronik. Piringan sensor *rotary encoder* dipasang pada *shaft* motor yang nantinya piringan akan dibaca oleh *optocoupler* tipe U. Pulsa hasil pembacaan *rotary encoder* dikirim ke arduino Mega2560 untuk diolah sebelum dikirim ke komputer. Data dari arduino Mega2560 kemudian dikirim dan hasilnya ditampilkan pada *display* yang terdapat pada komputer menggunakan *serial port*. *Tachometer* digunakan untuk melihat perbandingan kecepatan dari piringan dengan hasil pengolah arduino. Mekanisme pengujian kecepatan BLDCM dapat dilihat pada Gambar 4.7. Hasil pengujian dapat dilihat pada Tabel 4.2.



**Gambar 4.7** Mekanisme Pengujian Kecepatan BLDCM

**Tabel 4.2** Hasil Pengujian Kecepatan BLDCM

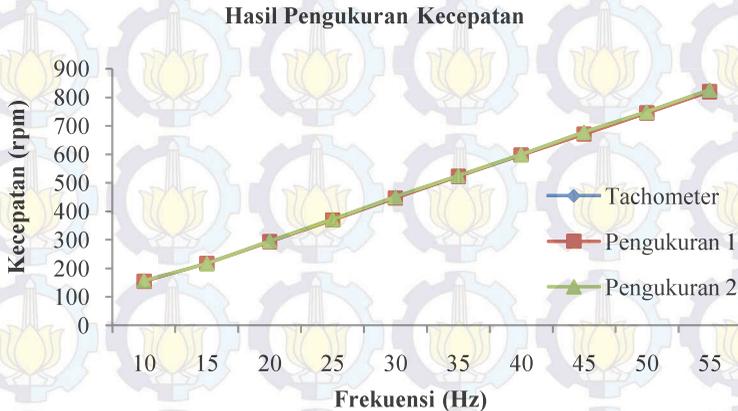
Frekuensi Fasa (Hz)	Kecepatan (Rpm)		
	Tachometer	Pengukuran 1	Pengukuran 2
10	155,5	155	157
15	218,5	217	217
20	295	294	296
25	372,5	370	371,5
30	447	447	451
35	524,5	523	525
40	600	598	599
45	673	671,4	678
50	748,5	745	749
55	826	820	826

Dari Tabel 4.2, dapat dibuat grafik hasil pengukuran yang dapat dilihat pada Gambar 4.8.

Berdasarkan data yang didapatkan pada tabel, hasil pembacaan kecepatan motor dengan *tachometer* dan perhitungan memiliki selisih yang kecil. Untuk mencari presentase *error* rata-rata digunakan rumus sebagai berikut.

$$error = \frac{(pembacaan\ sensor - Pembacaan\ Tachometer)}{Pembacaan\ Tachometer} \times 100\%$$

Sehingga didapatkan eror rata-rata dari semua pembacaan sebesar 0,109%.

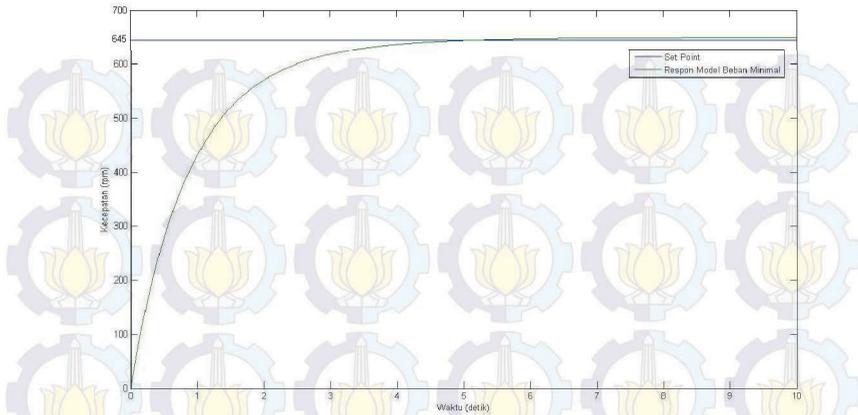


**Gambar 4.8** Hasil Pengujian Kecepatan BLDCM

### 4.3 Analisa Pengujian Simulasi tanpa Kontroler

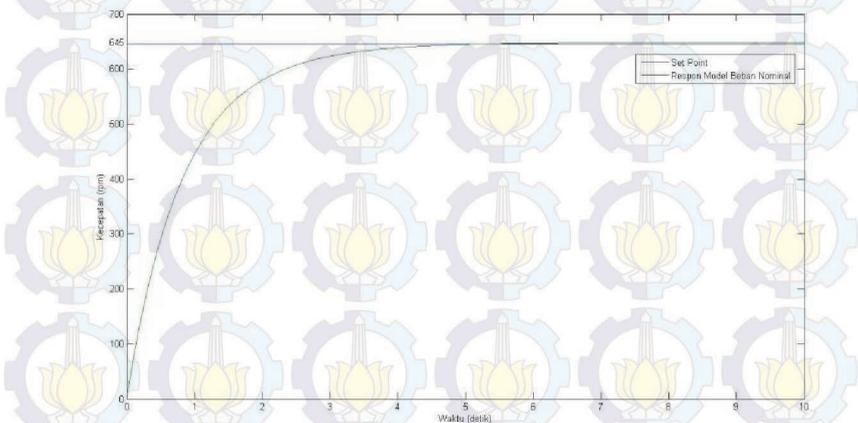
Pada tahap ini pengujian dilakukan untuk melihat respon *plant* secara *open loop* tanpa menggunakan kontroler tiap beban. Sinyal uji berupa sinyal *unit step* dengan *set point* sebesar 645 rpm. Pengujian ini dilakukan untuk melihat respon transien hasil identifikasi yang telah didapat. Pengujian ini berlangsung selama 10 detik. Respon beban minimal dapat dilihat pada Gambar 4.9.

Pada Gambar 4.9 dapat diketahui spesifikasi respon transient dengan nilai *time constant* 0,909 detik, *settling time* 2,950 detik, *rise time* 2,676 detik, *delay time* 0,622 detik, dan *error steady state* sebesar 0,744%.



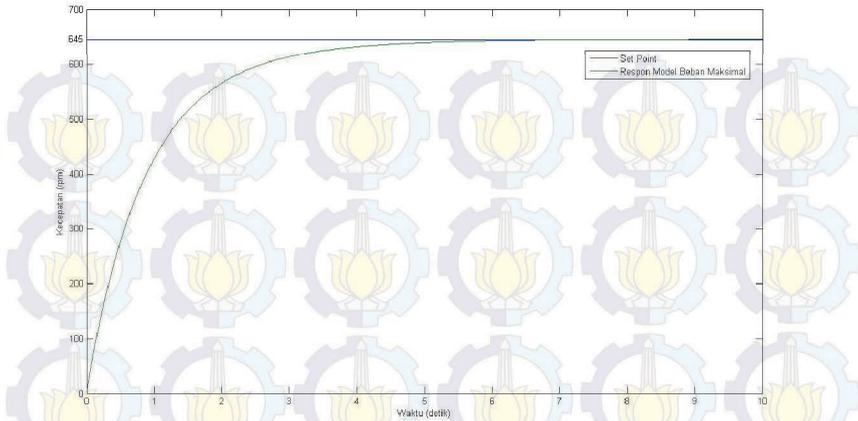
**Gambar 4.9** Hasil Pengujian Tanpa Kontroler untuk Beban Minimal

Respon hasil pengujian beban nominal tanpa menggunakan kontroler dapat dilihat pada Gambar 4.10.



**Gambar 4.10** Hasil Pengujian Tanpa Kontroler untuk Beban Nominal

Pada Gambar 4.10 dapat diketahui spesifikasi respon transient dengan nilai *time constant* 0,849 detik, *settling time* 2,736 detik, *rise time* 2,502 detik, *delay time* 0,582 detik, dan *error steady state* sebesar 0,515%. Respon hasil pengujian beban maksimal terlihat pada Gambar 4.11.



**Gambar 4.11** Hasil Pengujian Tanpa Kontroler untuk Beban Maksimal

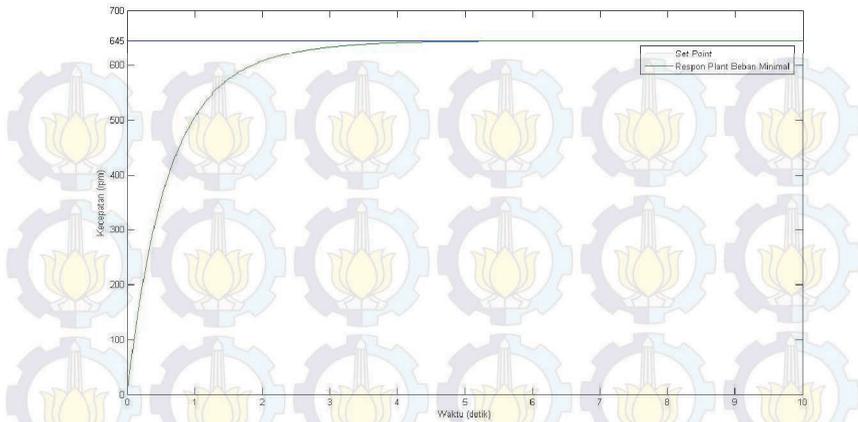
Pada Gambar 4.11 dapat diketahui spesifikasi respon transient dengan nilai *time constant* 0,913 detik, *settling time* 2,987 detik, *rise time* 2,688 detik, *delay time* 0,624 detik, dan *error steady state* sebesar 0,018%.

**Tabel 4.3** Spesifikasi Respon Tiap Beban

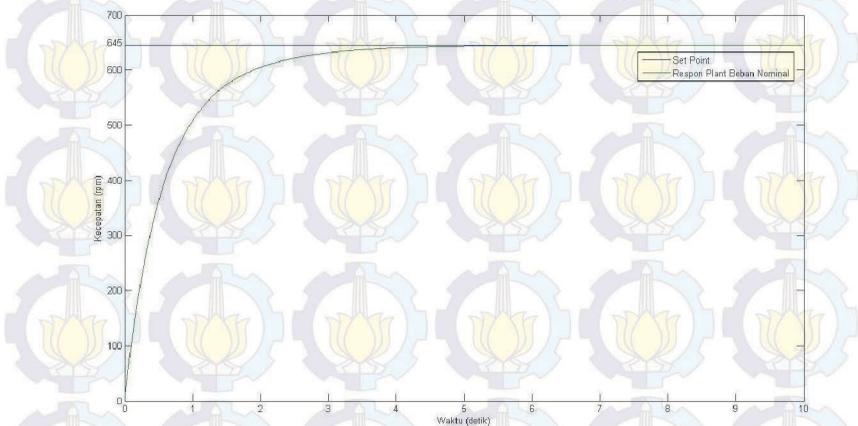
Spesifikasi	Beban Minimal	Beban Nominal	Beban Maksimal	Rata-rata
$\tau$	0,909 detik	0,849 detik	0,913 detik	0,890 detik
$t_s$ 5%	2,950 detik	2,736 detik	2,987 detik	2,891 detik
$t_r$ (5% – 95%)	2,676 detik	2,502 detik	2,688 detik	2,622 detik
$t_d$	0,622 detik	0,582 detik	0,624 detik	0,609 detik
$e_{ss}$	0,744%	0,515%	0,018%	0,425%

#### 4.4 Analisa Pengujian Simulasi dengan Kontroler PID-Fuzzy

Tahap ini dilakukan untuk mengetahui dan menguji proses kerja dari kontroler yang telah didesain. Pengujian dilakukan dengan Simulink yang terdapat dalam *software* Matlab. Pada Tugas Akhir ini dilakukan pengujian untuk PID-Fuzzy. Metode pembebanan diberikan dalam selang waktu 10 detik untuk tiap beban. Respon beban minimal dapat dilihat pada Gambar 4.12. Respon beban nominal dapat dilihat pada Gambar 4.13. Respon beban maksimal dapat dilihat pada Gambar 4.15. Respon beban berubah-ubah dapat dilihat pada Gambar 4.15.



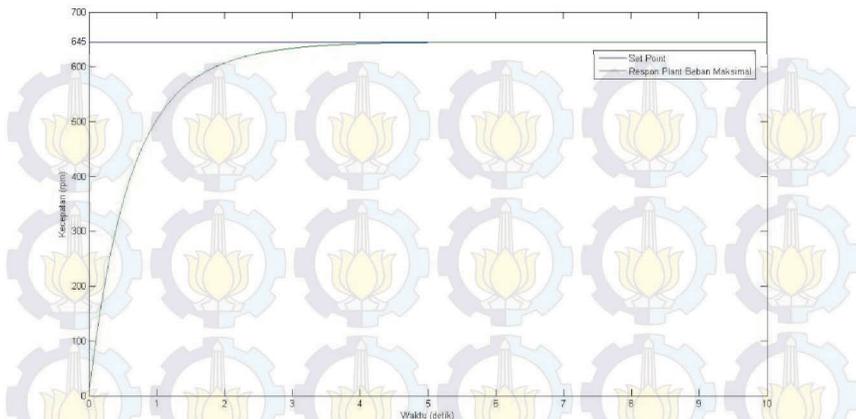
**Gambar 4.12** Respon Keluaran Beban Minimal dengan Kontroler PID-Fuzzy



**Gambar 4.13** Respon Keluaran Beban Nominal dengan Kontroler PID-Fuzzy

Pada Gambar 4.12 dapat diketahui spesifikasi respon transient dengan nilai *time constant* 0,631 detik, *settling time* 2,133 detik, *rise time* 1,866 detik, *delay time* 0,417 detik, dan *error steady state* sebesar 0,011%.

Pada Gambar 4.13 dapat diketahui spesifikasi respon transient dengan nilai *time constant* 0,634 detik, *settling time* 2,194 detik, *rise time* 1,920 detik, *delay time* 0,439 detik, dan *error steady state* sebesar 0,002%.

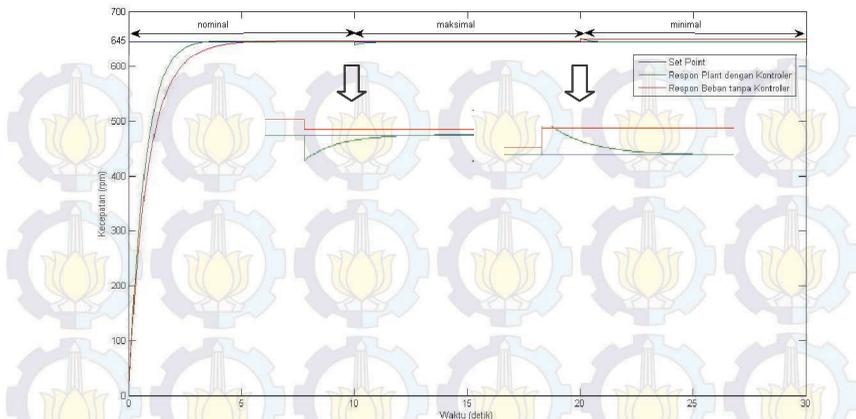


**Gambar 4.14** Respon Keluaran Beban Maksimal dengan Kontroler PID-*Fuzzy*

Pada Gambar 4.14 dapat diketahui spesifikasi respon transient dengan nilai *time constant* 0,639 detik, *settling time* 2,255 detik, *rise time* 1,950 detik, *delay time* 0,436 detik, dan *error steady state* sebesar 0,007%.

**Tabel 4.4** Spesifikasi Respon Tiap Beban dengan Kontroler PID-*Fuzzy*

Spesifikasi	Beban Minimal	Beban Nominal	Beban Maksimal	Rata-rata
$\tau$	0,631 detik	0,634 detik	0,639 detik	0,634 detik
$t_s$ 5%	2,133 detik	2,194 detik	2,255 detik	2,194 detik
$t_r$ (5% – 95%)	1,866 detik	1,920 detik	1,950 detik	1,912 detik
$t_d$	0,417 detik	0,439 detik	0,436 detik	0,430 detik
$e_{ss}$	0,011%	0,002%	0,007%	0,006%



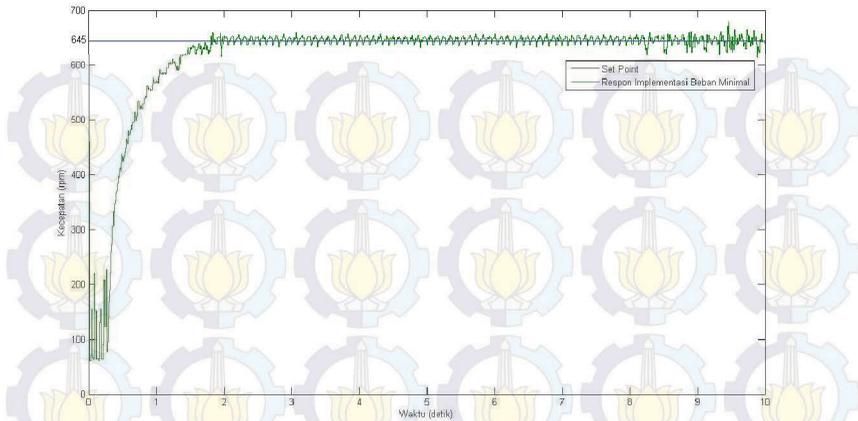
**Gambar 4.15** Respon Keluaran dengan Kontroler PID-*Fuzzy* Beban Berubah

Pada Gambar 4.15 terlihat saat terjadi perubahan beban nominal ke maksimal terjadi penurunan kecepatan kemudian kembali ke keadaan *steady state* dalam waktu 1,5 detik.

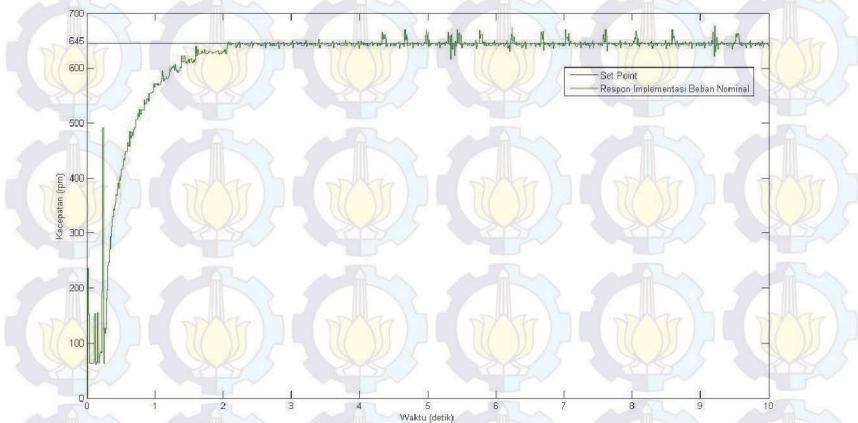
#### 4.5 Analisa Implementasi Pengujian Kontroler PID-*Fuzzy*

Pada implementasi kontroler PID-*Fuzzy* menggunakan beban rem magnetik. Pada beban minimal, piringan rotor tidak diberi beban rem magnetik. Pada beban nominal, piringan rotor diberi beban rem magnetic sebesar  $\frac{1}{2}$  luas rem magnetik. Pada beban maksimal, piringan rotor diberi beban magnetik sepenuhnya dari luas penampang rem magnetik. Implementasi pada beban minimal dapat dilihat pada Gambar 4.16.

Pada Gambar 4.16, hasil respon transien didapat dengan spesifikasi *time constant* sebesar 0,500 detik, *settling time* sebesar 1,640 detik, *delay time* sebesar 0,360 detik, *rise time* sebesar 1,540 detik, dan *error steady state* sebesar 0,254%. *Plant* BLDCM baru berfungsi saat frekuensi sistem yang diberikan mencapai 10 Hz atau setara 150 rpm. Hasil respon implementasi beban nominal dapat dilihat pada Gambar 4.17.

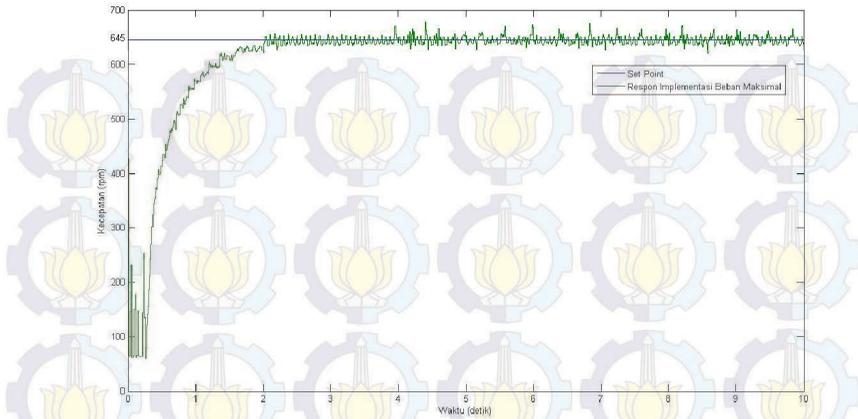


**Gambar 4.16** Respon Keluaran dengan Kontroler PID-Fuzzy Beban Minimal



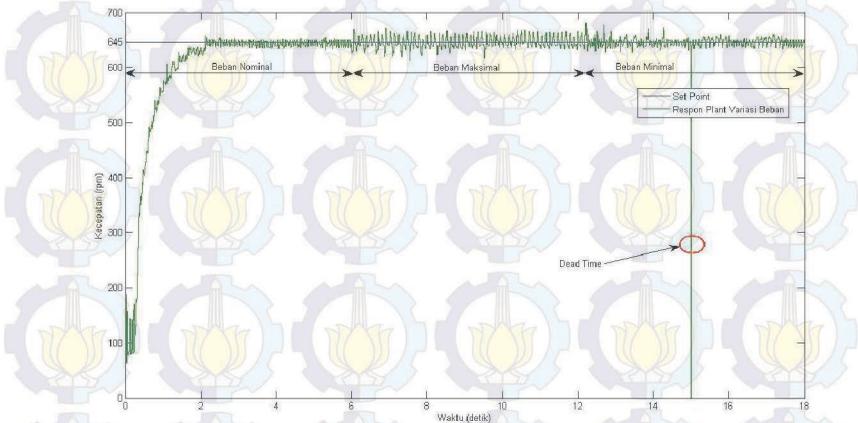
**Gambar 4.17** Respon Keluaran dengan Kontroler PID-Fuzzy Beban Nominal

Pada Gambar 4.17 didapat hasil respon transient dengan spesifikasi Hasil repon transien didapat dengan spesifikasi *time constant* sebesar 0,510 detik, *settling time* sebesar 1,680 detik, *delay time* sebesar 0,370 detik, *rise time* sebesar 1,550 detik, dan *error steady state* -0,200%. Nilai  $e_{ss}$  bernilai negatif dikarenakan nilai *steady state* berada di bawah nilai *set point*.



**Gambar 4.18** Respon Keluaran dengan Kontroler PID-*Fuzzy* Beban Maksimal

Pada Gambar 4.18, *time constant* sebesar 0,550 detik, *settling time* sebesar 1,700 detik, *delay time* sebesar 0,370 detik, *rise time* sebesar 1,570 detik, dan *error steady state* sebesar -0,100%.



**Gambar 4.19** Respon Keluaran dengan Kontroler PID-*Fuzzy* Beban Nominal-Maksimal-Minimal

Pada Gambar 4.19, mula-mula BLDCM dijalankan dalam keadaan beban nominal selama waktu 6 detik. Setelah itu, BLDCM diberi beban

rem magnetik maksimal selama 6 detik kemudian beban rem magnetik dilepas. Terlihat sekitar detik ke-15 terdapat respon *plant* jatuh ke 0 rpm, hal ini disebabkan adanya arduino tidak dapat merespon (*dead time*) atau memberikan nilai ukur ke Matlab. *Dead time* ini terjadi secara acak. Respon transient yang didapat, yaitu *time constant* ( $\tau$ ) sebesar 0,520 detik, *settling time* ( $t_s(5\%)$ ) sebesar 1,670 detik, *delay time* ( $t_d$ ) sebesar 0,370 detik, *rise time* ( $t_r(5\%-95\%)$ ) sebesar 1,490 detik, dan *error steady state* sebesar 0,099%. Terjadi *overshoot* sebesar 5,426% saat terjadi perubahan beban maksimal ke beban minimal. Spesifikasi hasil implementasi tiap beban dapat dilihat pada Tabel 4.5.

**Tabel 4.5** Spesifikasi Implementasi Respon Tiap Beban

Spesifikasi	Beban Minimal	Beban Nominal	Beban Maksimal	Rata-rata
$\tau$	0,500 detik	0,510 detik	0,550 detik	0,520 detik
$t_s$ 5%	1,640 detik	1,680 detik	1,700 detik	1,673 detik
$t_r$ (5% – 95%)	1,540 detik	1,550 detik	1,570 detik	1,553 detik
$t_d$	0,360 detik	0,370 detik	0,370 detik	0,366 detik
$e_{ss}$	0,254%	-0,200%	-0,100%	-0,015%

## DAFTAR PUSTAKA

- [1] Rasyid, M., “*Power Electronics Handbook*”, University Of West Florida, U.S.A.
- [2] Shiddiq Abdul Rachman, A., “Rancang Bangun Prototipe Sistem Aktual Kendali Sirip menggunakan Brushless DC”, *Skripsi*, Teknik Elektro-Universitas Indonesia, Jakarta, 2012.
- [3] Candra Wardianto, Ovi., “Kontrol Fuzzy Adaptif Gain Scheduling untuk Pengaturan Motor Induksi 3 Fasa”, *Tugas Akhir*, JTE-ITS, Surabaya, 2013.
- [4] Putri Suryaditya, Nindita., “Pengaturan Proses Face Miling pada Mesin Computer Numerical Control (CNC) dengan Kontroler Fuzzy-PID”, *Tugas Akhir*, JTE-ITS, Surabaya, 2013.
- [5] Matra Alfian, Rizki., “Perancangan dan Implementasi Pengaturan Kecepatan Motor Induksi Tiga Fasa menggunakan Kontroler PID Fuzzy dengan PLC”, *Tugas Akhir*, JTE-ITS, Surabaya, 2013.
- [6] Yus Peristiaferi, Trio., “Pengaturan Kecepatan Motor Tiga Fasa Pada Mesin Sentrifugal menggunakan Metode PID Fuzzy”, *Tugas Akhir*, JTE-ITS, Surabaya, 2015.
- [7] Fahmizal, “*Arduino Devices – Mikrokontroler Berbasis AVR ATmega Series*” <URL: <https://fahmizaleeits.wordpress.com/tag/arduino-adalah/>>, Februari, 2015.



## BAB V PENUTUP

Bab V menjelaskan tentang kesimpulan dan hasil dari simulasi dan implementasi secara keseluruhan.

### 5.1 Kesimpulan

Dari hasil desain kontrol PID-*Fuzzy* yang telah dikerjakan dan beberapa penerapan dalam tugas akhir ini dapat diambil beberapa kesimpulan sebagai berikut:

1. Pada simulasi kontroler PID-*Fuzzy* mampu memperbaiki performasi sistem pada tiap kondisi pembebanan dan hasil respon mendekati respon beban nominal dengan spesifikasi rata-rata  $\tau$  sebesar 0,634 detik,  $t_s(5\%)$  sebesar 2,194 detik,  $t_d$  sebesar 0,430 detik,  $t_r(5\%-95\%)$  sebesar 1,912 detik, dan  $e_{ss}$  sebesar 0,006%.
2. Pada implementasi kontroler PID-*Fuzzy* mampu memperbaiki performasi sistem pada tiap kondisi pembebanan dan hasil respon mendekati respon beban nominal dengan spesifikasi rata-rata  $\tau$  sebesar 0,520 detik,  $t_s(5\%)$  sebesar 1,673 detik,  $t_d$  sebesar 0,366 detik,  $t_r(5\%-95\%)$  sebesar 1,553 detik, dan  $e_{ss}$  sebesar -0,015%.
3. Pada implementasi beban berubah-ubah setiap 6 detik, kontroler PID-*Fuzzy* mampu mengembalikan respon *plant* ke keadaan *steady state* dengan spesifikasi  $\tau$  sebesar 0,520 detik,  $t_s(5\%)$  sebesar 1,670 detik,  $t_d$  sebesar 0,370 detik,  $t_r(5\%-95\%)$  sebesar 1,490 detik, dan  $e_{ss}$  sebesar 0,099%. Terjadi *overshoot* sebesar 5,426% saat terjadi perubahan beban.

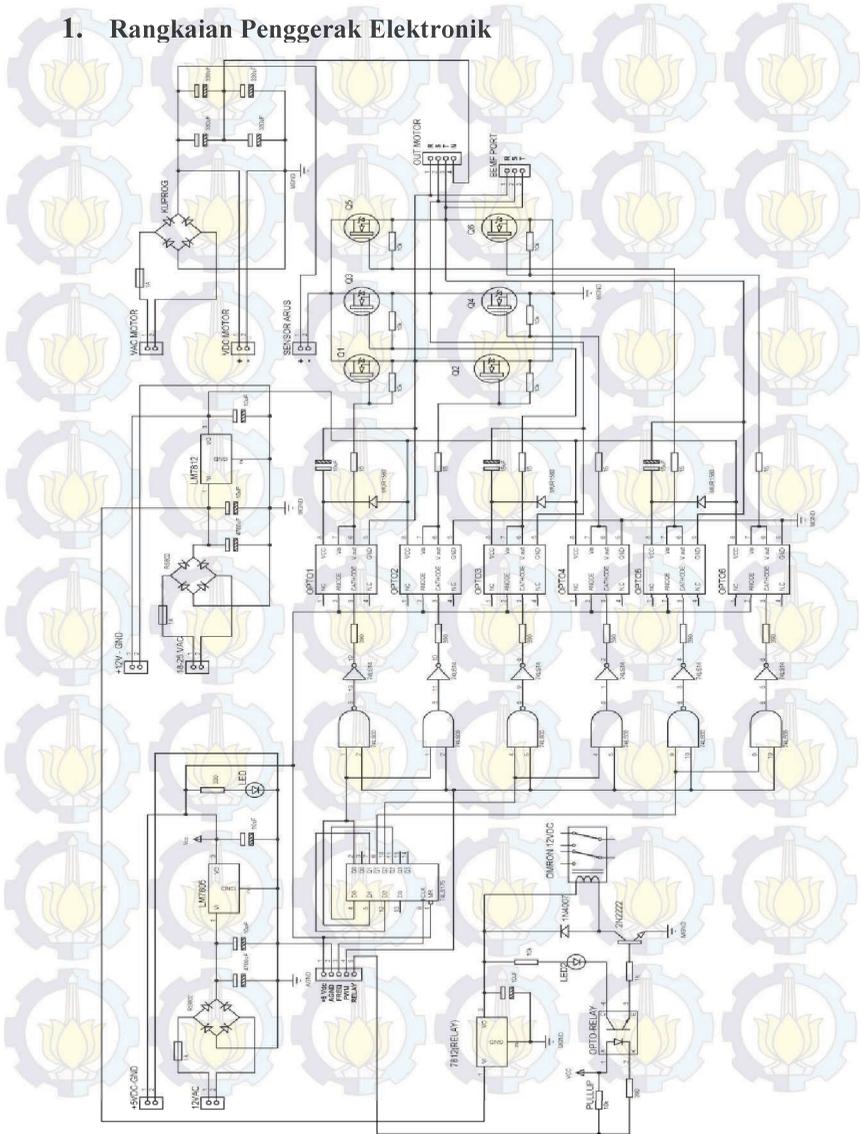
### 5.2 Saran

1. Untuk kelanjutan riset mendatang, diharapkan adanya modifikasi dari *plant* BLDCM.
2. Dalam melakukan identifikasi sebaiknya menggunakan beberapa metode identifikasi dan berbagai macam metode perhitungan *error* hasil identifikasi.
3. Pembuatan rangkaian pembangkit sinyal frekuensi fasa, sehingga rentang frekuensi yang dibangkitkan bisa lebih lebar.



# LAMPIRAN

## 1. Rangkaian Penggerak Elektronik



## 2. Program Arduino Pembangkit Frekuensi Fasa dan PWM

```
int bacadata = 0;
```

```
void setup() {  
  // put your setup code here, to run once:  
  // PWM 4 Khz  
  TCCR1B = TCCR1B & B11111000 | B00000010;  
  pinMode(12, OUTPUT); //pin frekuensi 74175  
  Serial.begin(115200); //set baud rate  
}
```

```
void loop() {  
  // put your main code here, to run repeatedly:  
  // cek komunikasi serial  
  if (Serial.available() > 0) {  
    // baca data serial  
    bacadata = (Serial.read());  
    // pembangkit frekuensi fasa  
    digitalWrite(12, HIGH);  
    delayMicroseconds(79000 / bacadata);  
    digitalWrite(12, LOW);  
    delayMicroseconds(79000 / bacadata);  
    //output pwm duty cycle 80%  
    analogWrite(10, 220);  
  }  
  if (Serial.available() < 1) {  
    analogWrite(10, 0);  
  }  
}
```

## 3. Program Arduino sebagai Pembaca Sensor Kecepatan

```
int duration = 0;
```

```
int rpm = 0;
```

```
char bufer[4];
```

```
void setup() {  
  // put your setup code here, to run once:  
  pinMode(52, INPUT); //encoder pin
```

```

Serial.begin(115200);
}

void loop() {
// put your main code here, to run repeatedly:
duration = pulseIn(52, HIGH, 100000);
rpm = (5500000 / 3) / duration;
if (rpm < 0) {
rpm = 0;
}
if (rpm > 1000) {
rpm = 0;
}
//delay(100);
sprintf(bufer, "%4d", rpm);
Serial.print(bufer);
delay(100);
}

```

#### 4. Program m-files fusikp

```

function kp=fusikp(x)
%fuzzifikasi Kp with 5 membership triangular
function
xf=[0 0 0 0 0]';
if x<-2
xf(1)=1;
elseif x<-1
xf(1)=-1-x;
xf(2)=x-(-2);
elseif x<0
xf(2)=0-x;
xf(3)=x-(-1);
elseif x<1
xf(3)=1-x;
xf(4)=x-0;
elseif x<2
xf(4)=2-x;
xf(5)=x-1;
else

```

```

        xf(5)=1;
    end

    %rule base kp
    uf=[0 0 0]';
    ukp=[0 0 0]';
    uf(1)=max(xf(1),xf(2));
    uf(2)=xf(3);
    uf(3)=max(xf(4),xf(5));
    jkp=uf(1)+uf(2)+uf(3);
    ukp(1)=uf(1)/jkp;
    ukp(2)=uf(2)/jkp;
    ukp(3)=uf(3)/jkp;

    %defuzzifikasi kp
    %atas=1.3275*uf(1)+1.3635*uf(2)+1.475*uf(3);
    %atas=1.313*uf(1)+1.41*uf(2)+1.42*uf(3);%new
    ident
    atas=1.313*ukp(1)+1.41*ukp(2)+1.42*ukp(3);
    bawah=ukp(1)+ukp(2)+ukp(3);
    kp=atas/bawah;

```

## 5. Program m-files fusiti

```

function ki=fusiti(x)
%fuzzifikasi Ti with 5 membership triangular
function
xf=[0 0 0 0 0]';
if x<-2
    xf(1)=1;
elseif x<-1
    xf(1)=-1-x;
    xf(2)=x-(-2);
elseif x<0
    xf(2)=0-x;
    xf(3)=x-(-1);
elseif x<1
    xf(3)=1-x;
    xf(4)=x-0;
elseif x<2

```

```

        xf(4)=2-x;
        xf(5)=x-1;
    else
        xf(5)=1;
    end

    %rule base ki
    uf=[0 0 0]';
    uki=[0 0 0]';
    uf(1)=max(xf(1),xf(2));
    uf(2)=xf(3);
    uf(3)=max(xf(4),xf(5));
    jki=uf(1)+uf(2)+uf(3);
    uki(1)=uf(1)/jki;
    uki(2)=uf(2)/jki;
    uki(3)=uf(3)/jki;

    %defuzzifikasi Ti
    %atas=1.25*uf(1)+1.5*uf(2)+2.169*uf(3);
    atas=1.488*uki(1)+1.492*uki(2)+1.499*uki(3);
    %new ident
    %atas=1.488*uf(1)+1.1*uf(2)+1.499*uf(3);
    bawah=uki(1)+uki(2)+uki(3);
    ki=atas/bawah;

```

## 6. Program m-files fusitd

```

function kd=fusitd(x)
%fuzzifikasi Td with 5 membership triangular
function
xf=[0 0 0 0 0]';
if x<-2
    xf(1)=1;
elseif x<-1
    xf(1)=-1-x;
    xf(2)=x-(-2);
elseif x<0
    xf(2)=0-x;
    xf(3)=x-(-1);

```

```

elseif x<1
    xf(3)=1-x;
    xf(4)=x-0;
    elseif x<2
        xf(4)=2-x;
        xf(5)=x-1;
    else
        xf(5)=1;
    end

%rule base Td
uf=[0 0 0]';
ukd=[0 0 0]';
uf(1)=max(xf(1),xf(2));
uf(2)=xf(3);
uf(3)=max(xf(4),xf(5));
jkd=uf(1)+uf(2)+uf(3);
ukd(1)=uf(1)/jkd;
ukd(2)=uf(2)/jkd;
ukd(3)=uf(3)/jkd;

%defuzzifikasi Td
%atas=-0.332*uf(1)+0*uf(2)+0.537*uf(3);
atas=-0.11*ukd(1)+(-0.09*ukd(2))+(-
0.07*ukd(3)); %new ident
bawah=ukd(1)+ukd(2)+ukd(3);
kd=atas/bawah;

```