ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - TE 141599

PERANCANGAN DAN IMPLEMENTASI SISTEM PENGATURAN KECEPATAN MOTOR BLDC MENGGUNAKAN KONTROLER PI BERBASISKAN *NEURAL-FUZZY* HIBRIDA ADAPTIF

Agung Setyadi Wicaksono
NRP 2212 100 177

Dosen Pembimbing
Ir. Rusdhianto Effendie A.K, M.T.
Eka Iskandar S.T, M.T.

JURUSAN TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2016

ITS
Institut
Teknologi
Sepuluh Nopember

FINAL PROJECT - TE 141599

DESIGN AND IMPLEMENTATION OF BLDC MOTOR SPEEDS CONTROL SYSTEM USING CONTROLLER PI BASED HYBRID ADAPTIVE NEURAL-FUZZY

Agung Setyadi Wicaksono
NRP 2212 100 177

Supervisor
Ir. Rusdhianto Effendie A.K, M.T.
Eka Iskandar S.T, M.T.

JURUSAN TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2016

PERANCANGAN DAN IMPLEMENTASI SISTEM PENGATURAN KECEPATAN MOTOR BLDC MENGGUNAKAN KONTROLER PI BERBASISKAN *NEURAL-FUZZY* HIBRIDA ADAPTIF

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan Untuk
Memperoleh Gelar Sarjana Teknik Elektro
Pada
Bidang Studi Teknik Sistem Pengaturan
Jurusan Teknik Elektro
Institut Teknologi Sepuluh Nopember

Menyetujui

Dosen Pembimbing I,	Dosen Pembimbing II,
Ir. Rusdhianto Effendie A.K, M.T.	**Eka Iskandar S.T, M.T.**
NIP. 195704241985021001	**NIP. 198005282008121001**

SURABAYA
JUNI, 2016

LEMBAR PERNYATAAN
PERSETUJUAN PUBLIKASI KARYA ILMIAH
UNTUK KEPENTINGAN AKADEMIS

Sebagai mahasiswa Institut Teknologi Sepuluh Nopember Surabaya, yang bertanda tangan di bawah ini saya :

Nama : Agung Satyadi W

Nrp. : 2212100177

Jurusan / Fak. : Teknik Elektro / Fakultas Teknologi Industri

Alamat kontak : Taman Bumi Marina Emas Blok I/76 Kapasari Sukolilo Surabaya

a. Email : agungsatyadiw@gmail.com

b. Telp/HP : +6285728731441

Menyatakan bahwa semua data yang saya *upload* di Digital Library ITS merupakan hasil final (revisi terakhir) dari karya ilmiah saya yang sudah disahkan oleh dosen penguji. Apabila dikemudian hari ditemukan ada ketidaksesuaian dengan kenyataan, maka saya bersedia menerima sanksi.

Demi perkembangan ilmu pengetahuan, saya menyetujui untuk memberikan **Hak Bebas Royalti Non-Ekslusif** ***(Non-Exclusive Royalti-Free Right)*** kepada Institut Teknologi Sepuluh Nopember Surabaya atas karya ilmiah saya yang berjudul :

Perancangan dan Implementasi Sistem Pengaturan Kecepatan Motor BLDC Menggunakan Kontroler PI Berbasiskan Neural Fuzzy Hibrida Adaptif

Dengan Hak Bebas Royalti Non-Ekslusif ini, Institut Teknologi Sepuluh Nopember Surabaya berhak menyimpan, mengalih-media/format-kan, mengelolanya dalam bentuk pangkalan data *(database)*, mendistribusikannya, dan menampilkan/mempublikasikannya di internet atau media lain untuk kepentingan akademis tanpa meminta ijin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta. Saya bersedia menanggung secara pribadi, segala bentuk tuntutan hukum yang timbul atas pelanggaran Hak Cipta dalam karya Ilmiah saya ini tanpa melibatkan pihak Institut Teknologi Sepuluh Nopember Surabaya.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Surabaya

Pada tanggal : 29 Juni 2016

Yang menyatakan,

Agung Satyadi Wicaksono

Nrp. 2212100177

Dosen Pembimbing 1

Rusdhianto Eka

NIP. 195704241985021001

KETERANGAN :

Tanda tangan pembimbing wajib dibubuhi stempel jurusan.

Form dicetak dan diserahkan di bagian Pengadaan saat mengumpulkan hard copy TA/Tesis/Disertasi

PERANCANGAN DAN IMPLEMENTASI SISTEM PENGATURAN KECEPATAN MOTOR BLDC MENGGUNAKAN KONTROLER PI BERBASISKAN *NEURAL-FUZZY* HIBRIDA ADAPTIF

Agung Setyadi Wicaksono
2212 100 177

Dosen Pembimbing I : Ir. Rusdhianto Effendie A.K, M.T.
NIP : 195704241985021001
Dosen Pembimbing II : Eka Iskandar S.T, M.T.
NIP : 198005282008121001

ABSTRAK

Mobil listrik menjadi inovasi terbaru dengan tujuan utama untuk melepaskan ketergantungan pada bahan bakar minyak. Penelitian yang telah ada memaparkan bahwa motor listrik yang sesuai untuk menggerakkan mobil listrik adalah motor *Brushless Direct Current* (BLDC). Beberapa keunggulan motor BLDC antara lain adalah suara halus, ukuran kompak, torsi besar, efisiensi tinggi, memiliki umur pakai yang panjang, dan mudah dikontrol. Performa dan kecepatan motor BLDC dapat terganggu apabila bekerja pada kondisi berbeban. Oleh karena itu, dibutuhkan pengaturan kecepatan menggunakan sebuah kontroler yang dapat menjaga kecepatan motor BLDC sesuai *set-point* meskipun sedang beroperasi pada kondisi berbeban.

Kontroler yang digunakan untuk mengatur kecepatan motor BLDC adalah kontroler Proposional Integral (PI) berbasiskan *Neural-Fuzzy* Hibrida Adaptif. Kontroler PI dipilih karena dapat mengeliminasi *steady-state error*. Sedangkan *Neural-Fuzzy* Hibrida Adaptif merupakan kombinasi antara *Fuzzy* dan *Neural-Network*. *Fuzzy* digunakan untuk penentuan parameter kontroler PI. Parameter kontroler PI didapatkan dari *Neural-Network*. Karakteristik respon terhadap hasil implementasi memiliki *settling time* 20 detik, *overshoot* sebesar 1,1%, dan *time constant* 7,7 detik.

Kata Kunci : *Brushless DC, Hybrid Adaptive Neural-Fuzzy, Propotional and Integral* (PI) *Controller.*

DESIGN AND IMPLEMENTATION OF BLDC MOTOR SPEEDS CONTROL SYSTEM USING CONTROLLER PI BASED HYBRID ADAPTIVE NEURAL-FUZZY

Agung Setyadi Wicaksono
2212 100 177

Supervisor I : Ir. Rusdhianto Effendie A.K, M.T.
ID Number : 195704241985021001
Supervisor II : Eka Iskandar S.T, M.T.
ID Number : 198005282008121001

ABSTRACT

Electric cars become the latest innovations with the main objective to release the dependence on fossil fuels. Research that has been there explained that the electric motor is suitable to drive an electric car is a Brushless Direct Current (BLDC) motor. Some of the advantages of BLDC motor is smooth sound, compact size, large torque, high efficiency, has a long lifespan, and easy to control. Performance and speed of the BLDC motor can be disturbed when working on load condition. Therefore, it takes the speed setting using a controller that can keep BLDC motor speed suit to set-point even when operating at load condition.

The controller used to control the speed of the BLDC motor is a Proportional Integral (PI) *controller based Hybrid Adaptive Neural-Fuzzy. PI controller is chosen because it can eliminate the steady-state error. While Hybrid Adaptive Neural-Fuzzy is a combination of Fuzzy Logic and Neural-Network. Fuzzy Logic is used to determine parameters PI controller. Parameters PI Controller obtained from Neural-Network. The response characteristics of the results of the implementation have 20 seconds settling time, overshoot of 1.1%, and the time constant of 7.7 seconds.*

Keywords **:** *Brushless DC, Hybrid Adaptive Neural-Fuzzy, Propotional and Integral* (PI) *Controller.*

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

(halaman ini sengaja dikosongkan)

BAB 1
PENDAHULUAN

1.1 Latar Belakang [1][2]

Perkembangan di bidang otomotif mengantarkan kita pada kendaraan tanpa menggunakan bahan bakar minyak. Mobil listrik menjadi inovasi terbaru dengan tujuan utama untuk melepaskan ketergantungan pada bahan bakar minyak.

Penelitian yang telah ada memaparkan bahwa motor listrik yang sesuai untuk menggerakkan mobil listrik adalah *Brushless Direct Current* (BLDC) motor. Motor BLDC dipilih karena memiliki beberapa keunggulan apabila dibandingkan dengan motor listrik yang lain. Beberapa keunggulan yang menjadi pertimbangan pada pemilihan motor BLDC antara lain adalah suara halus, ukuran kompak, torsi besar, efisiensi tinggi, memiliki umur pakai yang panjang, mudah dikontrol, dan biaya perawatan yang rendah. Motor BLDC beroperasi tanpa menggunakan sikat sehingga rugi gesekan pada sikat dapat dihilangkan.

Performa dan kecepatan motor BLDC dapat terganggu apabila bekerja pada kondisi berbeban. Oleh karena itu, dibutuhkan pengaturan kecepatan menggunakan sebuah kontroler yang dapat menjaga kecepatan motor BLDC sesuai *set-point* meskipun sedang beroperasi pada kondisi berbeban.

Kontroler yang digunakan untuk mengatur kecepatan motor BLDC adalah kontroler Proposional Integral (PI) berbasiskan *Neural-Fuzzy* Hibrida Adaptif. Kontroler PI dipilih karena dapat mengeliminasi *steady-state error*. Sedangkan *Neural-Fuzzy* Hibrida Adaptif merupakan kombinasi antara *Fuzzy Logic* dan *Neural Network*. Sedangkan *Neural-Fuzzy* Hibrida Adaptif merupakan kombinasi antara *Fuzzy* dan *Neural Network*. *Fuzzy* digunakan untuk penentuan parameter kontroler PI. Parameter kontroler PI didapatkan dari *Neural-Network*.

1.2 Perumusan Masalah

Pada keadaan berbeban, kecepatan motor BLDC tidak bergerak sesuai *set-point*. Pada kondisi berbeban, kecepatan motor bergantung pada intensitas pembebanan yang diberikan. Pembebanan merupakan representasi dari kondisi riil yang ada di lapangan, dimana efek pembebanan sendiri diibaratkan sebagai gangguan eksternal yang dapat menghambat laju kecepatan motor seperti laju pada kendaraan yang sering mendapat gangguan dari kondisi medan yang tidak bersahabat

seperti jalan menanjak, jalan berlubang ataupun bergelombang yang menyebabkan kecepatan motor yang dihasilkan tidak lagi sesuai *setpoint*. Pada Tugas Akhir ini dirancang suatu kontroler yang dapat menjaga kecepatan motor BLDC sesuai *set-point* pada kondisi berbeban. Kontroler yang digunakan untuk mengatur kecepatan motor BLDC adalah kontroler PI berbasiskan *Neural-Fuzzy* Hibrida Adaptif.

1.3 Batasan Masalah

Berdasarkan perumusan masalah, penulis akan membatasi permasalahan yang akan diteliti sehingga tujuan dari penelitian dapat dicapai. Batasan dari penelitian tersebut adalah sebagai berikut:

- *Range* kerja kecepatan dari motor BLDC yang akan diteliti berada pada *range* kecepatan 1000 hingga 1600 rpm.
- Kontroler yang digunakan pada sistem pengaturan kecepatan motor BLDC adalah kontroler PI berbasiskan *Neural-Fuzzy* Hibrida Adaptif menggunakan *interface* berupa mikrokontroler Arduino Uno dan *software* MATLAB R2013a
- Penambahan efek pembebanan menggunakan rem elektromagnetik pada motor BLDC terbagi menjadi 3 variabel beban, yaitu kondisi beban minimal, beban nominal, dan beban maksimal.

1.4 Tujuan Penelitian

Tujuan pelaksanaan Tugas Akhir ini adalah merancang kontroler untuk menjaga kecepatan motor BLDC dari efek pembebanan menggunakan kontroler PI berbasiskan *Neural-Fuzzy* Hibrida Adaptif.

1.5 Sistematika Penulisan

Buku tugas akhir ini terdiri dari lima bab dan disusun menurut sistematika penulisan berikut ini:

BAB I: PENDAHULUAN

Pendahuluan berisi tentang latar belakang, rumusan masalah, tujuan, batasan masalah, sistematika penulisan, dan relevansi.

BAB 2: TINJAUAN PUSTAKA

Tinjauan pustaka berisi tentang teori yang menunjang penelitian, berupa teori tentang BLDC, serta metode yang digunakan untuk pengaturan kecepatan motor BLDC.

BAB 3: PERANCANGAN SISTEM

Perancangan sistem berisi tentang perancangan perangkat keras, perangkat lunak, dan perancangan kontroler.

BAB 4: PENGUJIAN DAN ANALISA

Pengujian dan analisa berisi tentang hasil simulasi kontroler dan analisanya. Selain itu berisi tentang hasil implementasi kontroler pada Simulator BLDC beserta analisa hasil implementasi.

BAB 5: PENUTUP

Penutup berisi kesimpulan dan saran yang dapat dijadikan pertimbangan pengembangan berdasarkan hasil pengerjaan tugas akhir ini.

1.6 Relevansi

Hasil yang diperoleh dari pelaksanaan Tugas Akhir ini diharapkan dapat menjadi referensi perancangan kontroler untuk pengaturan kecepatan motor BLDC. Selain itu, hasil Tugas Akhir ini juga diharapkan dapat menjadi referensi terkait perancangan kontroler berbasis kontroler PI berbasiskan *Neural-Fuzzy* Hibrida Adaptif untuk tujuan penelitian terkait pengaturan kecepatan pada jenis motor lainnya.

(halaman ini sengaja dikosongkan)

BAB 2
TEORI PENUNJANG

2.1 Motor BLDC [3]

Motor BLDC merupakan pilihan tepat untuk aplikasi yang membutuhkan keandalan tinggi, efisiensi tinggi, dan rasio *power-to-volume* tinggi. Secara umum, motor BLDC dianggap sebagai motor dengan performa tinggi yang mampu menghasilkan torsi yang besar pada *range* kecepatan yang besar. Motor BLDC adalah turunan dari motor DC yang paling umum digunakan, yaitu motor DC dengan sikat dan mereka memiliki kurva karakteristik torsi dan kecepatan yang sama. Perbedaan utama motor BLDC dan DC adalah penggunaan sikat. Motor BLDC tidak memiliki sikat dan harus terkomutasi secara elektronik.

Komutasi merupakan perubahan fase arus motor pada waktu yang tepat untuk menghasilkan torsi rotasional. Dalam motor DC dengan sikat, motor memiliki komutator fisik yang digerakkan dengan sikat untuk memindahkan rotor. Dalam motor BLDC, kekuatan arus listrik magnet permanen menyebabkan motor untuk bergerak, sehingga komutator fisik tidak diperlukan.

Motor BLDC sangat handal karena tidak memiliki sikat yang dapat aus dan harus diganti. Ketika dioperasikan dalam kondisi optimal, usia motor dapat lebih dari 10000 jam. Untuk aplikasi jangka panjang, hal ini dapat menjadi keuntungan yang besar. Setiap kali motor rusak atau perlu diganti, *plant* atau bagian dari *plant* harus dimatikan. Hal ini membutuhkan waktu dan uang, tergantung pada berapa lama waktu yang dibutuhkan untuk mengganti komponen yang aus dan rusak agar *plant* dapat berjalan seperti semula. Meskipun motor BLDC memakan biaya lebih dari motor DC dengan sikat, hal ini akan sepadan seiring dengan banyaknya waktu dan uang yang dihabiskan motor DC dengan sikat.

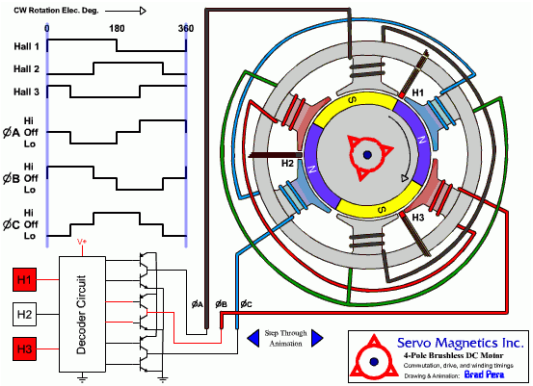
Motor yang digunakan adalah motor *Air Conditioner* (AC) Daikin *Inverter*. Motor Daikin *Inverter* ditunjukkan pada Gambar 2.1.

Gambar 2.1 Motor Daikin *Inverter*

2.1.1 Cara Kerja Motor BLDC

Prinsip kerja motor BLDC sama dengan motor DC konvensional tetapi berbeda pada penggunaan sikat. Sikat pada motor DC konvensional digunakan dalam proses komutasi sedangkan pada motor BLDC menggunakan sensor hall dalam proses komutasi.

Skema kerja motor BLDC ditunjukkan pada Gambar 2.2.



Gambar 2.2 Skema Kerja Motor BLDC [4]

Langkah pertama, sensor hall H1 dan H3 bernilai 1 karena mengalami perubahan medan magnet sehingga kontroler mengalirkan arus pada lilitan B dan C. Lilitan B menjadi kutub utara dan lilitan C menjadi kutub selatan. Kutub utara lilitan B memberikan tolakan pada

kutub utara magnet rotor sedangkan kutub selatan lilitan C menarik kutub utara magnet rotor.

Langkah kedua, hanya sensor H1 bernilai 1 sehingga kontroler akan menginstruksikan agar lilitan A dan B harus dialiri arus. Lilitan A menjadi kutub selatan dan lilitan B tetap menjadi kutub utara. Kutub selatan lilitan A akan menolak kutub selatan magnet rotor sedangkan kutub utara lilitan B menolak kutub utara magnet rotor.

Langkah ketiga, sensor H1 dan H2 bernilai 1 sehingga kontroler menginstruksikan agar lilitan A dan C dialiri arus. Lilitan A tetap menjadi kutub selatan dan lilitan C menjadi kutub utara. Kutub selatan lilitan A menolak kutub selatan dan menarik kutub utara magnet rotor sedangkan kutub utara lilitan C menarik kutub selatan magnet rotor.

Langkah keempat, hanya sensor H2 yang bernilai 1 sehingga kontroler menginstruksikan agar lilitan B dan C dialiri arus. Lilitan B menjadi kutub selatan dan lilitan C tetap menjadi kutub utara. Kutub selatan lilitan B menolak kutub selatan magnet rotor sedangkan kutub utara lilitan C menarik kutub selatan magnet rotor.

Langkah kelima, sensor H2 dan H3 bernilai 1 sehingga kontroler menginstruksikan agar lilitan A dan B dialiri arus. Lilitan A menjadi kutub utara dan lilitan B tetap menjadi kutub selatan. Kutub utara lilitan A menolak kutub utara dan menarik kutub selatan magnet rotor sedangkan kutub selatan lilitan B menolak kutub selatan magnet rotor.

Langkah keenam atau terakhir, hanya sensor H3 yang bernilai 1 sehingga kontroler menginstruksikan agar lilitan A dan C dialiri arus. Lilitan A tetap menjadi kutub utara dan lilitan C menjadi kutub selatan. Kutub utara lilitan A menarik kutub selatan magnet rotor sedangkan kutub selatan lilitan C menarik kutub utara magnet rotor.

Keenam proses tersebut mengalami pengulangan hingga membentuk suatu siklus yang menyebabkan motor terus berputar secara kontinyu selama sumber arus DC masih ada.

2.1.2 Kontruksi Motor BLDC

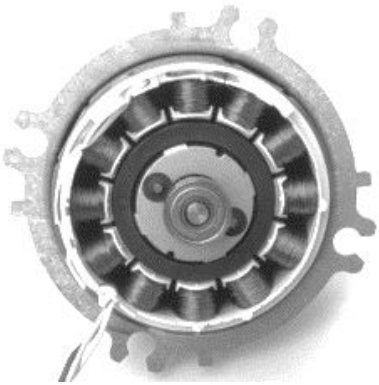
Inner rotor dan *outer rotor* merupakan dua desain motor BLDC. Setiap motor BLDC memiliki stator dan rotor. Bagian penting lain dari motor BLDC adalah gulungan rotor dan magnet rotor.

Gulungan rotor pada desain *outer rotor* terletak pada inti motor. Magnet rotor mengelilingi gulungan rotor. Magnet rotor bertindak sebagai insulator sehingga mengurangi laju disipasi panas dari motor.

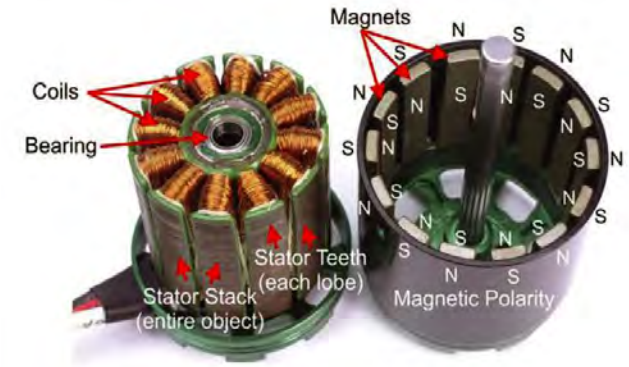
Desain *outer rotor* beroperasi pada *duty cycle* rendah. Keuntungan utama dari desain *outer rotor* motor BLDC adalah torsi *cogging* relatif rendah.

Gulungan rotor pada desain *inner rotor* mengelilingi rotor dan ditempelkan pada rumah motor. Keuntungan utama dari desain *inner rotor* adalah kemampuan untuk menghilangkan panas. Kemampuan motor untuk menghilangkan panas secara langsung berdampak pada kemampuan untuk menghasilkan torsi.

Bentuk dari desain *inner rotor* dan *outer rotor* motor BLDC ditunjukkan pada Gambar 2.3 dan 2.4.



Gambar 2.3 *Inner Rotor* Motor BLDC [4]



Gambar 2.4 *Outer Rotor* Motor BLDC [4]

Rotor merupakan bagian motor yang memiliki lilitan, apabila lilitan rotor diberi arus maka akan timbul gaya elektromagnetik. Oleh karena itu, apabila pada rotor diberi arus maka rotor akan menghasilkan gaya tarik-menarik dan tolak-menolak yang menyebabkan motor berputar.

Rotor dari motor BLDC terbuat dari baja terlaminasi tersusun untuk membawa gulungan. Gulungan pada rotor terdapat dua pola yaitu; pola *star* (Y) dan pola *delta* (Δ). Perbedaan utama antara kedua pola tersebut yaitu pada pola (Y) menghasilkan torsi besar pada rotasi per menit (rpm) rendah dan pada pola (Δ) menghasilkan torsi kecil pada rpm rendah. Hal ini dikarenakan pada pola (Δ), setengah dari tegangan yang ada pada gulungan tidak dikendalikan sehingga meningkatkan *loss* pada efisiensi dan torsi.

2.1.3 Motor BLDC Daikin

Motor BLDC yang digunakan merupakan motor BLDC dari AC Daikin *inverter*. Bentuk fisik dari motor BLDC ditunjukkan pada Gambar 2.1. Motor BLDC Daikin *inverter* memiliki 5 kabel *input-output*. Setiap kabel memiliki warna dan fungsi yang berbeda. Fungsi dari masing-masing kabel ditunjukkan pada Tabel 2.1.

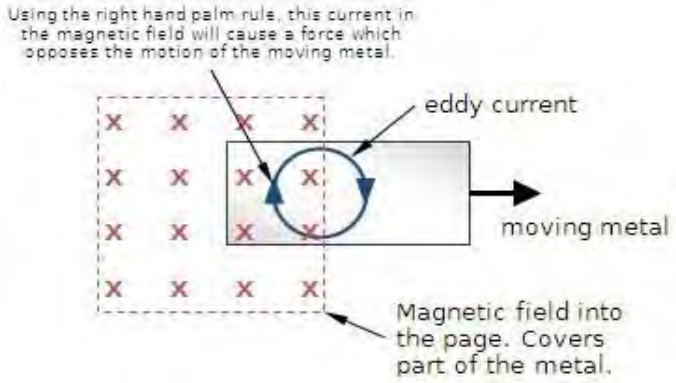
Tabel 2.1 Fungsi Kabel *Input-Output Driver* Motor BLDC [3][4]

Warna	Keterangan
Jingga	Kabel *input* sinyal kontrol. *Input* kabel jingga merupakan tegangan dengan rentang 0 - 5 Volt. Kecepatan putar motor akan berubah sesuai dengan tegangan yang diberikan pada kabel jingga.
Putih	Kabel *output* sinyal informasi kecepatan motor. Kabel putih meng-*generate* sinyal kotak dengan frekuensi sesuai dengan kecepatan putar motor. Hubungan antara frekuensi dan kecepatan motor adalah: $\omega = 15 \times f$ Di mana ω adalah kecepatan putar motor dalam rpm dan f adalah frekuensi sinyal dalam Hertz.
Merah	Kabel *power* yang men-*supply* daya ke motor dan *driver*.
Biru	Kabel *common* dari *driver* dan motor.
Coklat	*Motor adjustment pin*. *Low* untuk kecepatan rendah, *high* untuk kecepatan tinggi.

2.2 Rem Elektromagnetik [5]

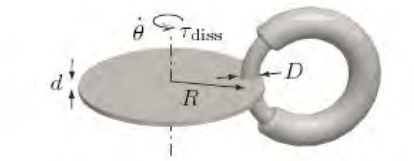
Sistem pengereman elektromagnetik menggunakan gaya elektromagnetik yang timbul dari magnet permanen tetap atau kumparan yang diberikan arus listrik untuk memperlambat suatu gerakan. Konstruksi dasar rem elektromagnetik berupa suatu piringan logam *non*-feromagnetik yang terpasang pada suatu poros yang berputar. Piringan logam diapit oleh kumparan yang dialiri arus listrik hingga menimbulkan medan magnet dengan kutub saling berlawanan. Logam piringan memotong medan magnet yang ditimbulkan oleh kumparan sehingga menimbulkan *eddy current* atau arus *eddy*.

Arus *eddy* merupakan arus listrik yang timbul ketika suatu piringan logam berada di sekitar medan magnet dimana garis-garis gaya magnet sedang berubah-ubah. Arus *eddy* mempunyai medan magnet yang berlawanan arah dengan arah gerak piringan logam sehingga laju piringan logam akan tertahan akibat dari arus *eddy*. Prinsip arus *eddy* pada logam yang bergerak ditunjukkan pada Gambar 2.5.



Gambar 2.5 Prinsip Arus *Eddy* pada Logam yang Bergerak

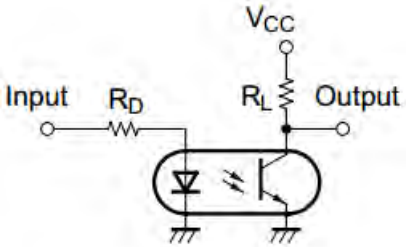
Rem elektromagnetik diletakkan didekat bagian yang bergerak dan bekerja pada suhu dingin serta memenuhi persyaratan energi pengereman kecepatan tinggi. Aplikasi rem elektromagnetik ditemukan pada sistem pengereman kereta api. Struktur rem elektromagnetik ditunjukkan Gambar 2.6.



Gambar 2.6 Struktur Rem Elektromagnetik

2.3 Rangkaian *Optocoupler* [6]

Rangkaian *optocoupler* digunakan untuk memisahkan *plant* dengan Arduino untuk menghindari kerusakan pada Arduino karena arus berlebih dari *power supply* apabila terjadi kecelakaan atau kesalahan dalam memasang *port*. Rangkaian isolator *optocoupler* menggunakan *optocoupler* PIC817 ditunjukkan pada Gambar 2.7.



Gambar 2.7 *Optocoupler*

2.4 Arduino [7]

2.4.1 Pengertian Arduino

Arduino merupakan mikrokontroler *open-source single-board* dirancang untuk memudahkan akuisisi data dalam berbagai bidang. Arduino memiliki prosesor Atmel AVR. Arduino sangat populer di seluruh dunia. Sebagian besar pemula belajar mengenal robotika dan elektronika melalui Arduino karena mudah dipelajari. Bahasa pemrograman yang digunakana Arduino adalah bahasa C. Arduino yang digunakan dalam tugas akhir adalah Arduino Uno.

Arduino Uno adalah *board* mikrokontroler berbasis Atmega328. Arduino mempunyai 14 pin *input/output* digital dengan 6 pin dapat digunakan sebagai *output Pulse Width Modulator* (PWM), 6 *input analog*, resonator keramik 19MHz, koneksi *Universal Serial Bus* (USB), *electric*

socket, ICSP *header*, dan tombol *reset*. Kabel USB Arduino mudah dihubungkan ke komputer dan dapat diaktifkan dengan baterai atau adaptor *Alternating Current* (AC) ke DC. Spesifikasi Arduino Uno ditunjukkan pada Tabel 2.2.

Tabel 2.2 Spesifikasi Arduino

Mikrokontroler	Atmega328
Tegangan operasi	5V
Tegangan *input* (direkomendasikan)	7-12V
Tegangan *input* (batasan)	6-20V
Input/*output pin* digital	14 (6 pin dapat digunakan sebagai *output* PWM)
Input analog pin	6
Arus DC setiap *pin input/output*	40mA

2.4.2 *Input/Output Analog*

Digital input/output atau digital pin adalah pin-pin untuk menghubungkan Arduino dengan komponen atau rangkaian digital. Komponen yang menghasilkan *output* digital atau menerima *input* digital dapat disambungkan ke pin-pin digital.

2.4.3 Pin-pin Catu Daya

Pin-pin catu daya adalah pin yang memberikan tegangan untuk komponen atau rangkaian yang dihubungkan dengan Arduino. Pada pin-pin catu daya terdapat pin *Vin* dan *reset*.

2.4.4 Soket Baterai

Soket baterai digunakan untuk menyuplai daya Arduino dengan tegangan dari baterai 9V pada saat Arduino sedang tidak disambungkan ke komputer. Arduino Uno ditunjukkan pada Gambar 2.8.

Gambar 2.8 Arduino Uno

2.5 **MATLAB** [8]

MATLAB merupakan program dengan bahasa pemrograman tingkat tinggi yang didesain untuk mengembangkan algoritma, visualisasi data, dan komputasi numerik. MATLAB digunakan untuk menyelesaikan masalah komputasi lebih cepat dibandingkan dengan bahasa pemrograman C, C++, dan Fortran. MATLAB digunakan untuk aplikasi seperti *signal and image processing*, desain kontrol, pengujian, pengukuran, permodelan, dan analisis.

Simulink merupakan bagian dari MATLAB untuk memodelkan, mensimulasikan, dan menganalisis sistem dinamik. Simulink membentuk atau memodifikasi model sesuai dengan keinginan. Simulink mendukung sistem linier dan *non*-linier, pemodelan waktu kontinyu atau diskrit atau gabungan.

Instrument Control Toolbox merupakan modul Simulink yang digunakan untuk komunikasi perangkat keras. Modul merupakan kumpulan fungsi ***m-file*** untuk komputasi MATLAB. MATLAB *Toolbox* menyediakan kerangka kerja untuk komunikasi instrumen yang mendukung *General Purpose Interface* Bus (GPIB) *interface*, *Virtual Instrument Standard Architecture* (VISA) *standard*, Transmission Control Protocol/Internet Protocol (TCP/IP), dan *User Datagram Protocol* (UDP) *protocol*. MATLAB *Toolbox* memperluas fitur dasar *serial port* yang ada dalam MATLAB. MATLAB *Toolbox* berfugsi untuk komunikasi data antara *workspace* MATLAB dan peralatan pendukung MATLAB.

Komunikasi *serial* merupakan protokol dasar tingkat rendah untuk komunikasi antara dua peralatan atau lebih. *Serial port* mengirim dan menerima informasi dalam *bytes* dengan hubungan seri. *Bytes* dikirimkan menggunakan format biner atau karakter *American Standard Code for Information Interchange* (ASCII). ASCII *encode* dan *decode* yang pada *xPC Target Library for RS232* digunakan untuk memproses data ASCII secara *real time*. ASCII *encode* merupakan blok dalam Simulink yang digunakan untuk mengubah data *bytes* menjadi karakter ASCII sedangkan ASCII *decode* merupakan blok Simulink yang digunakan untuk mengubah karakter ASCII menjadi data *bytes* yang kemudian dikonversi sesuai kebutuhan.

2.6 **Identifikasi Sistem** [8]

Desain suatu kontroler dan melakukan simulasi dibutuhkan suatu model matematika dari *plant*. Identifikasi sistem digunakan untuk

mendapatkan model matematika dari *plant*. Model matematika yang baik dibutuhkan analisis dan prediksi. Model matematika memiliki bentuk yang bermacam-macam. Salah satu bentuk model matematika adalah *transfer function.*

Identifikasi sistem suatu *plant* dapat dilakukan menggunakan metode identifikasi statis maupun dinamis. Identifikasi statis dilakukan untuk mendapatkan *gain* dan *time sampling* dari suatu *plant* sedangkan identifikasi dinamis digunakan untuk mendapatkan model matematika dari suatu *plant* yang menggambarkan hubungan antara *input* dan *output* pada kondisi berbeban.

2.6.1 Prosedur Identifikasi Sistem

Identifikasi sistem adalah pendekatan eksperimental untuk menentukan model dinamis dari sebuah sistem. Langkah identifikasi sistem yang pertama adalah melakukan akuisisi data *input-output* kemudian menghitung estimasi dari struktur model dan mengestimasi parameter model lalu langkah terakhir adalah melakukan validasi pada model yang telah teridentifikasi.

Dalam akuisisi data *input-output*, sinyal *input* yang dipilih harus memiliki spektrum frekuensi yang kaya. Umumnya, sinyal *input* yang dipilih berupa sinyal *Pseudo Random Binary Sequence* (PRBS). Masalah yang sering timbul saat melakukan identifikasi sistem adalah menentukan derajat dari polinomial (numerator dan denumerator) dari *transfer function* yang merepresentasikan model *plant*.

2.6.1.1 Identifikasi Parameter

Identifikasi parameter dilakukan untuk mendapatkan pemodelan dari *plant*. Identifikasi parameter memiliki beberapa kelebihan dibandingkan identifikasi respon transien.

2.6.1.2 Validasi Model

Setelah *transfer function* didapat melalui proses identifikasi sistem, tahapan selanjutnya adalah tahapan validasi model. Validasi model diukur dengan cara mengukur nilai *error* setiap variabel. Tujuan dari validasi model adalah untuk mengukur agar nilai pada pemodelan mendekati model sesungguhnya.

Dalam validasi model, *Mean Square Error* (MSE) merupakan salah satu metode dan tolok ukur yang dapat digunakan. MSE mengukur akurasi pada nilai deret waktu secara statistik seperti regresi. MSE dapat merepresentasikan ukuran dari *error* rata-rata karena MSE

membandingkan hasil data pengukuran dan data pemodelan pada skala yang sama antara kedua data tersebut.

2.6.2 MSE

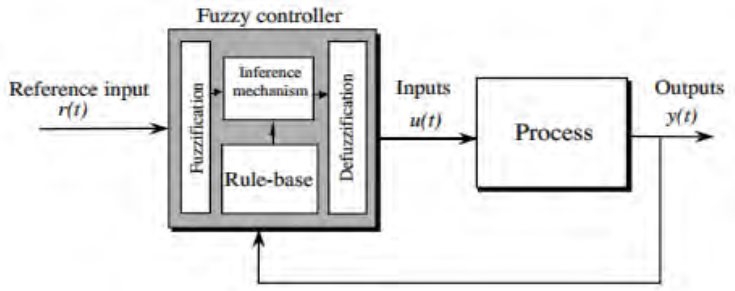
MSE dapat digunakan sebagai tolak ukur validasi model. Nilai MSE kecil menunjukkan nilai parameter sesuai keinginan. Bentuk umum pernyataan *fuzzy* ditunjukkan pada Persamaan 2.1.

$$MSE = \frac{\sum_{i=i}^{k}(y_i - \tilde{y}_i)^2}{k} \tag{2.1}$$

2.7 *Fuzzy Logic* [9]

Dasar dari *fuzzy logic* adalah himpunan *fuzzy*. Zadeh memperkenalkan konsep himpunan *fuzzy* sebagai perluasan dari himpunan konvensional. Suatu himpunan *fuzzy* adalah koleksi dari bilangan real yang memiliki keanggotaan parsial dalam himpunan [9].

Logika *fuzzy* sudah digunakan dalam dalam bidang kontrol. Secara umum, sistem *fuzzy* terdiri dari beberapa komponen, yaitu *fuzzification*, *fuzzy rule base*, *fuzzy inference engine*, dan *defuzzifier* seperti yang ditunjukkan pada Gambar 2.9.



Gambar 2.9 *Fuzzy Controller* [4]

2.7.1 *Fuzzy Rule Base*

Fuzzy rule base berisi pernyataan-pernyataan logika *fuzzy*, yang berbentuk pernyataan IF-THEN. Bentuk umum pernyataan *fuzzy* ditunjukkan pada Persamaan 2.2.

$$IF\ x_1\ is\ A_1^1\ and\ \dots and\ x_n\ is\ A_n^1\ THEN\ y\ is\ B^1 \quad (2.2)$$

A_1^l dan B^l adalah himpunan *fuzzy*, sedangkan $x = (x_1, x_2, \dots, x_n)^T$ dan y adalah *input* dan *output* dari variabel *fuzzy*.

2.7.2 *Fuzzy Inference System*

Fuzzy inference engine merupakan suatu modul pembuat keputusan dalam kontroler *fuzzy* yang berfungsi untuk menentukan suatu kesimpulan berdasarkan pada seberapa besar pengaruh setiap *rule* dalam *rule base* berdasarkan pada nilai *input* yang masuk.

2.7.3 *Fuzzification*

Fuzzifier digunakan untuk memetakan nilai variabel di dunia nyata kedalam himpunan *fuzzy*. Pemetaan nilai variabel dilakukan dengan menggunakan fungsi keanggotaan. Formulasi *singleton, gaussian,* dan *triangular fuzzyfier* ditunjukkan pada Persamaan 2.3, 2.4, dan 2.5.

- *Singleton fuzzyfier*

$$\mu A'(x) = \begin{cases} 1\ if\ x = x* \\ 0\ otherwise \end{cases} \quad (2.3)$$

- *Gaussian fuzzifier*

$$\mu A'(x) = e^{-\left(\frac{x_1 - x_1^*}{a_1}\right)} * \dots * e^{-\left(\frac{x_n - x_n^*}{a_n}\right)} \quad (2.4)$$

- *Triangular fuzzyfier*

$$\mu A'(x) = \begin{cases} \left(1 - e^{-\left(\frac{x_1 - x_1^*}{b_1}\right)} * \dots * e^{-\left(\frac{x_n - x_n^*}{b_n}\right)}\ if\ |x_n - x_n^*| \le b_i, i = 1,2,\dots,n\right) \\ 0\ otherwise \end{cases} \quad (2.5)$$

2.7.4 *Defuzzification*

Defuzzifier mengembalikan hasil perhitungan *fuzzy* menjadi variabel yang sesuai di dunia nyata. Sama dengan fuzzifier, defuzzifier juga menggunakan fungsi keanggotaan untuk memetakan nilai himpunan *fuzzy* menjadi variabel nyata. Beberapa metode *defuzzifier* adalah:

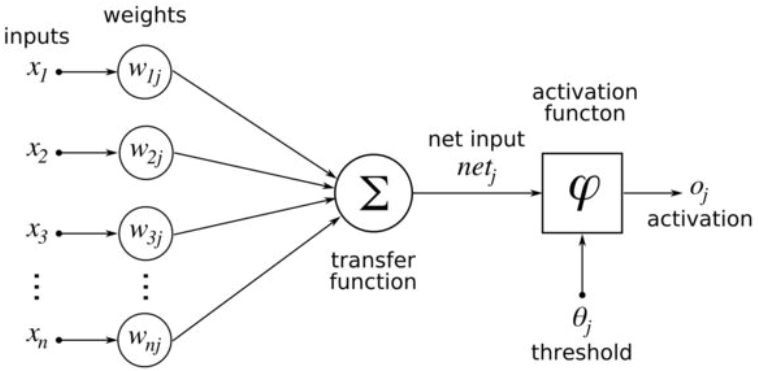
- *Center of gravity defuzzifier*. *Center of gravity* dinyatakan dengan y*, menunjukan pusat area yang diliputi oleh fungsi keanggotaan.
- *Center average defuzzifier*. *Center average* menunjukan *weight average* dari titik tengah (*center*) masing-masing fungsi keanggotaan.
- *Maximum defuzzifier*. *Maximum defuzzifier* memilih nilai tertinggi sebagai y*. Terdapat 3 pilihan, yaitu *smallest of maxima*, *largest of maxima* atau *mean of maxima*.

2.8 Jaringan Syaraf Tiruan (JST) [10][11]

JST adalah suatu program komputer yang dibuat berdasarkan cara kerja otak manusia. JST melakukan proses pelatihan yang meniru kerja otak manusia. Proses komputasi pada JST diilhami dari struktur dan cara kerja otak manusia. Pada jaringan syaraf otak manusia, informasi disalurkan dari satu *neuron* ke *neuron* lainnya. Sementara pada JST proses penyaluran informasi dari satu *neuron* ke *neuron* lainnya diimplementasikan pada program komputer. Proses pelatihan JST umumnya menggunakan metode pelatihan *backpropagation* yang sudah banyak diterapkan pada semua proses pelatihan yang sederhana sampai yang rumit.

Metode pelatihan *backpropagation* termasuk ke dalam metode pelatihan terawasi (*supervisory learning*). *Supervisory learning* adalah metode pelatihan yang memasukan target ke *output* dalam data untuk proses pelatihan. Metode pelatihan *backpropagation* telah banyak diaplikasikan dalam berbagai bidang, antara lain: bidang finansial, pengenalan pola tulis tangan, sistem kontrol, dan lain sebagainya. Metode *backpropagation* banyak diaplikasikan dalam berbagai proses karena metode ini didasarkan pada interkoneksi yang sederhana. Apabila *output* JST tidak sesuai dengan *output* yang diinginkan, maka metode *backpropagation* akan memperbaiki bobot (*weight*) yang ada pada lapisan tersembunyi (*hidden layer*) untuk mencapai *output* JST yang sesuai dengan target *output*.

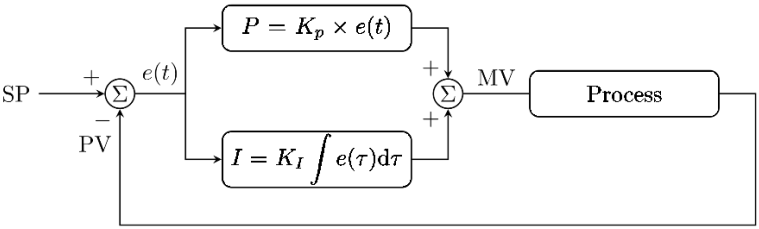
Pada dasarnya JST akan diberikan pola *input* sebagai pola pelatihan maka pola akan menuju ke unit-unit lapisan tersembunyi dan akan diteruskan ke lapisan *output*. Struktur JST ditunjukkan pada Gambar 2.10.



Gambar 2.10 Struktur JST [11]

2.9 Kontroler *Propotional Integral* (PI) [3]

Kontroler PI berguna menentukan presisi suatu sistem instrumentasi dengan karakteristik adanya umpan balik pada sistem tesebut. Kontroler PI adalah kontroler konvensional yang banyak dipakai dalam dunia industri. Blok diagram dan persamaan kontroler PI ditunjukkan pada Gambar 2.11 dan Persamaan 2.6.



Gambar 2.11 Blok Diagram Kontroler PI

$$mv(t) = \left(K_P e(t) + \frac{1}{T_i} \int_0^t e(t)dt \right) \tag{2.6}$$

mv(t) : *manipulated variable*
Kp : konstanta proporsional

Ti : *time* integral
e(t) : *error*

Komponen kontroler PI terdiri dari dua jenis yaitu proporsional dan integratif. Komponen proposional dan integral dapat dipakai bersamaan maupun sendiri-sendiri tergantung dari respon yang kita inginkan terhadap suatu *plant.*

2.9.1 Kontroler Proposional

Apabila pada kontroler proporsional nilai G(s) = Kp maka Kp berlaku sebagai *gain* (penguat) saja tanpa memberikan efek dinamik kepada kinerja kontroler. Penggunaan kontroler proporsional memiliki berbagai keterbatasan karena sifat kontrol yang tidak dinamik. Walaupun demikian dalam aplikasi-aplikasi dasar yang sederhana kontroler proposional mampu memperbaiki respon transien khususnya *rise time* dan *settling time*. Kontroler proporsional memiliki keluaran yang sebanding dengan besarnya sinyal kesalahan (selisih antara besaran yang diinginkan dengan *output*).

Ciri-ciri kontroler proporsional:

- Apabila nilai Kp kecil, kontroler proporsional hanya mampu melakukan koreksi kesalahan yang kecil, sehingga akan menghasilkan respon sistem yang lambat (menambah *rise time*)
- Apabila nilai Kp dinaikkan, respon sistem akan semakin cepat mencapai keadaan *steady state* (mengurangi *rise time*)
- Namun, apabila nilai Kp diperbesar sehingga mencapai harga yang berlebihan, akan mengakibatkan sistem bekerja tidak stabil atau respon sistem akan berosilasi
- Apabila nilai Kp dapat di-*set* sedemikian sehingga mengurangi *error steady state*, tetapi tidak menghilangkannya.

2.9.2 Kontroler Integral

Kontroler integral berfungsi untuk menghasilkan respon sistem yang memiliki *error steady state* nol. Apabila sebuah kontroler tidak memiliki unsur integrator, kontroler proporsional tidak mampu menjamin memiliki keluaran sistem dengan *error steady state* nol.

Ketika nilai *error steady state* mendekati nol maka efek kontrol integral semakin kecil. Kontroler integral dapat menghilangkan *error steady-state*, namun pemilihan Ki yang tidak tepat dapat menyebabkan

ketidakstabilan sistem. Pemilihan Ki yang sangat tinggi dapat menyebabkan *output* berosilasi karena menambah orde sistem.

Keluaran kontroler integral merupakan hasil penjumlahan yang terus menerus dari perubahan *input*. Ketika sinyal kesalahan tidak mengalami perubahan, maka kontroler integral menjaga keadaan *output* seperti sebelum terjadinya perubahan *input*. Sinyal keluaran kontroler integral merupakan luas bidang yang dibentuk oleh kurva *error*.

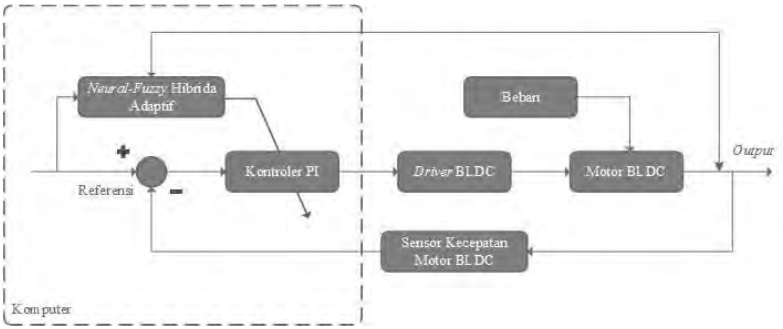
Ciri-ciri kontroler integral:

- Keluaran kontroler integral membutuhkan selang waktu tertentu, sehingga kontroler integral cenderung memperlambat respon
- Ketika sinyal kesalahan berharga nol maka keluaran kontroler bertahan pada nilai sebelumnya
- Apabila sinyal kesalahan tidak berharga nol, keluaran menunjukkan kenaikan atau penurunan yang dipengaruhi oleh besarnya sinyal kesalahan dan nilai Ki
- Ki yang berharga besar mempercepat hilangnya *offset* tetapi semakin besar nilai konstanta Ki mengakibatkan peningkatan osilasi dari sinyal keluaran kontroler.

BAB 3
PERANCANGAN SISTEM

3.1 Gambaran Umum Sistem

Penulis merancang sistem pengaturan kecepatan motor BLDC menggunakan kontroler PI berbasiskan *Neural-Fuzzy* Hibrida Adaptif. Hibrida merupakan kombinasi antara *fuzzy* dan *neural-network* sedangkan adaptif merupakan sifat sistem kendali dimana parameter-parameter sistem kendali dapat menyesuaikan terhadap perubahan kondisi beban. Kontroler *fuzzy* pada sistem pengaturan kecepatan motor BLDC berfungsi sebagai pengatur nilai Kp dan Ki pada kontroler PI, sehingga nilai Kp dan Ki menjadi fungsi keanggotaan pada *output* kontroler *fuzzy*. *Input* kontroler *fuzzy* berupa nilai tegangan beban. Nilai Kp dan Ki didapatkan dari proses *learning neural-network*. Blok diagram pengaturan kecepatan motor BLDC ditunjukkan pada Gambar 3.1.



Gambar 3.1 Blok Diagram Pengaturan Kecepatan Motor BLDC

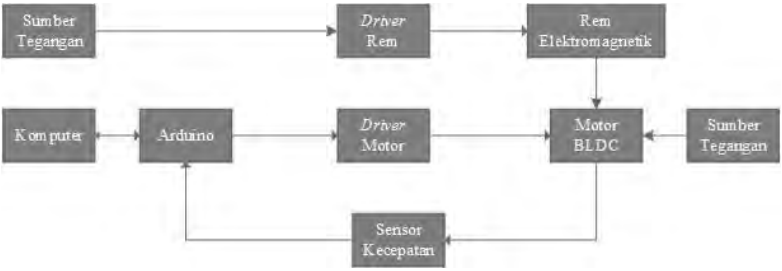
Plant menggunakan sistem kendali *negative feedback* yang terdiri dari komponen utama yaitu motor BLDC, *driver*, dan rem elektromagnetik. *Driver* berfungsi sebagai aktuator yang menghubungkan antara kontroler dengan *plant* sedangkan rem elektromagnetik berfungsi untuk memberikan efek pembebanan pada motor BLDC. Sinyal kontrol dari kontroler berupa sinyal PWM yang digunakan sebagai *input driver* untuk mengatur kecepatan motor BLDC. Pemorograman kontroler dilakukan di MATLAB R2013a sedangkan Arduino bertindak sebagai *interface* antara komputer dengan *driver* motor BLDC. Sensor yang

digunakan dalam sistem pengaturan kecepatan motor BLDC adalah sensor kecepatan yang digunakan sebagai *negative feedback* untuk mengukur kecepatan motor secara aktual.

3.2 Perancangan Perangkat Keras

Perancangan perangkat keras ada dua jenis, yaitu perancangan mekanik dan elektronik. Perancangan mekanik merupakan perancangan komponen utama pada *plant*, yaitu berupa motor BLDC dan rem elektromagnetik sedangkan perancangan elektronik merupakan perancangan pada kontroler, *driver* rem, dan rangkaian sensor kecepatan pada *plant*.

Arduino menerima data dari *sensor* kecepatan dan mengirimkan data pada komputer kemudian mengirimkan sinyal kontrol ke motor BLDC melaui Arduino Konfigurasi perangkat keras sistem pengaturan kecepatan BLDC ditunjukkan pada Gambar 3.2.



Gambar 3.2 Konfigurasi Perangkat Keras Sistem Pengaturan Kecepatan BLDC

3.2.1 Perancangan Mekanik

Plant sistem pengaturan kecepatan motor BLDC memiliki beberapa komponen utama, yaitu motor BLDC sebagai objek yang dikontrol dan rem elektromagnetik untuk memberikan efek pembebanan pada motor BLDC. Konstruksi motor BLDC dan rem elektromagnetik dijelaskan pada subbab 3.2.1.1 dan 3.2.1.2.

3.2.1.1 Motor BLDC

Penulis menggunakan motor BLDC Daikin D43F. Motor BLDC Daikin D43F merupakan motor BLDC yang memiliki 5 kabel untuk *supply* motor, *adjustment pin* motor, sinyal kontrol, sensor kecepatan, dan

ground. Spesifikasi dan bentuk fisik motor BLDC Daikin D43F ditunjukkan pada Tabel 3.1 dan Gambar 3.3.

Tabel 3.1 Spesifikasi Motor BLDC D43F [3]

Parameter		**Nilai**
Berat Motor		1200 gram
Tegangan Kerja		311 VDC
Kecepatan Motor	Beban Minimal	1950 rpm
	Beban Nominal	1760 rpm
	Beban Maksimal	1580 rpm

Gambar 3.3 Motor BLDC Daikin D43F [3]

3.2.1.2 Rem Elektromagnetik

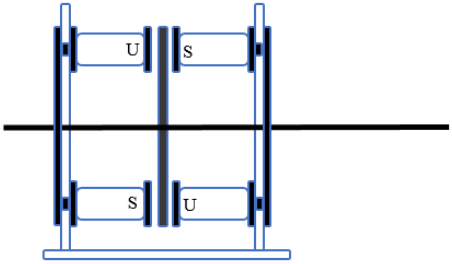
Rem elektromagnetik berfungsi sebagai pembebanan motor BLDC. Medan elektromagnetik dihasilkan oleh kumparan yang dialiri arus DC. Arus DC yang dialirkan kedalam rem elektromagnetik melalui adaptor yang memiliki *range ouput* 16 - 24 V.

Rem elektromagnetik pada simulator BLDC dipasang pada poros motor BLDC. Penulis menggunakan rem elektromagnetik yang tersusun atas cakram alumunium dan diapit empat pasang belitan. Belitan rem elektromganetik dialiri arus listrik sehingga timbul medan magnet dan

muncul fluks magnet yang memotong cakram alumunium. Ketika cakram berputar, muncul ggl induksi pada cakram alumunium karena alumunium merupakan konduktor, maka muncul arus listrik yang berputar pada cakram alumunium dengan arah berlawanan dengan perubahan fluks. Spesifikasi dan konstruksi rem elektromagnetik ditunjukkan pada Tabel 3.2 dan Gambar 3.4.

Tabel 3.2 Spesifikasi Rem Elektromagnetik [3]

Parameter	Nilai
Jumlah kumparan	8 (masing-masing 400 kumparan)
Resistansi kumparan (seri)	12,1 Ω
Diamater kumparan	3 cm
Diameter cakram	15 cm
Diamater kawat	0,6 mm
Tegangan kerja	0-30 V
Arus maksimal	2 A



Gambar 3.4 Konstruksi Fisik Rem Elektromagnetik [4]

3.2.1.3 Kopel Fleksibel

Konstruksi penahan *shaft* pada rem elektromagnetik harus lurus dengan *shaft* motor BLDC karena *shaft* motor BLDC dipasang seporos dengan rem elektromagnetik. Apabila kedua *shaft* tidak lurus, akan menyebabkan kerusakan pada *bearing* bahkan *shaft* dapat bengkok apabila kedua *shaft* diputar pada keadaan tidak lurus. Untuk menjaga kedua *shaft* tetap seporos sangat sulit karena saat motor berputar timbul getaran pada motor sehingga menyebabkan kedua *shaft* tidak lurus. Untuk

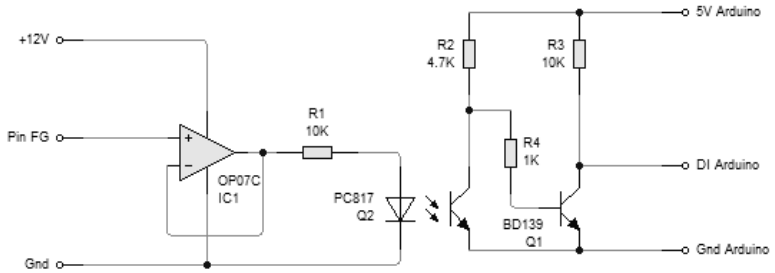
menghindari ketidakseporosan *shaft* perlu dipasang kopel yang menghubungkan *shaft* motor BLDC dengan *shaft* rem elektromagnetik.

3.2.2 Perancangan Elektronik

Perancangan elektronik motor BLDC meliputi rangkaian *driver* rem elektromagnetik, *driver* motor BLDC, dan rangkaian sensor kecepatan motor BLDC.

3.2.2.1 Rangkaian Isolasi Sensor Kecepatan Motor BLDC

Rangkaian isolasi sensor kecepatan motor BLDC digunakan untuk mengukur kecepatan motor BLDC secara *realtime*. Pada pin *frequency generator* (FG), rangkaian *driver* menghasilkan sinyal. Frekuensi sinyal kotak merepresentasikan kecepatan motor BLDC. Sinyal kotak pada pin FG dimasukkan pin digital *input* arduino. Untuk mengisolasi rangkaian arduino dengan *plant* digunakan *optocoupler*. Rangkaian isolasi sensor kecepatan motor BLDC ditunjukkan pada Gambar 3.5.



Gambar 3.5 Rangkaian Isolasi Sensor Kecepatan Motor BLDC [3]

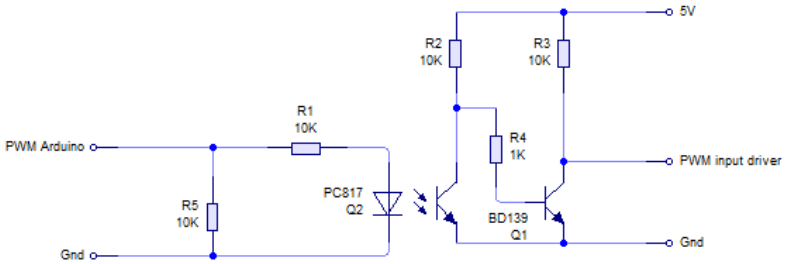
Untuk menghitung kecepatan motor dapat dilakukan dengan menghitung frekuensi gelombang kotak yang dihasilkan pin FG. Untuk mendapatkan kecepatan motor dapat digunakan Persamaan 3.1.

$$N = \frac{60FG}{P} \tag{3.1}$$

N	:	Kecepatan motor (rpm)
FG	:	*Frequency Generator* (Hz)
P	:	Jumlah pasang kutub (Volt)

3.2.2.2 Rangkaian Isolasi Sinyal Kontrol Motor BLDC

Driver isolasi sinyal kontrol motor BLDC berguna mengubah sinyal kontrol dari kontroler yang berupa tegangan analog menjadi urutan komutasi kumparan motor BLDC. Rangkaian isolasi Arduino dengan *driver* BLDC ditunjukkan pada Gambar 3.6.



Gambar 3.6 Rangkaian Isolasi Arduino dengan *Driver* BLDC [3]

3.2.2.3 Interface Arduino Uno R3

Arduino adalah pengendali mikro *single-board* yang bersifat *open-source*, diturunkan dari *Wiring platform*, dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang. Arduino memiliki prosesor Atmel AVR dan pemrograman pada arduino dilakukan menggunakan IDE (menggunakan bahasa C). Pada simulator BLDC, Arduino digunakan sebagai *interface* (penghubung) antara kontroler berbasis komputer dengan *plant*. Arduino digunakan untuk mengakuisisi data kecepatan motor BLDC, arus motor BLDC, dan arus rem elektromagnetik. Tegangan yang masuk ke arduino diubah menjadi data digital dengan resolusi 8 bit dan diolah oleh program kontroler pada komputer kemudian kontroler menghasilkan sinyal kontrol berbentuk PWM yang dikeluarkan melalui pin *output* digital PWM. Transfer data antara komputer dan arduino dilakukan melalui *port* komunikasi *serial*. Berikut ini pemilihan pin *input*/*output* pada Arduino:

- Pin 6 : *output* PWM untuk *driver* motor BLDC
- Pin 7 : *input digital* sensor kecepatan motor BLDC
- Pin +5 V

- Pin *Ground*

3.3 Perancangan Perangkat Lunak

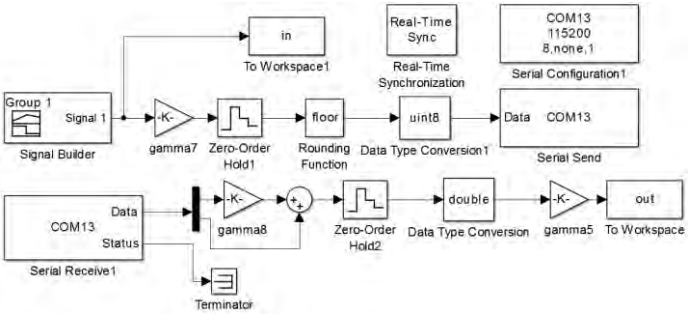
Dalam sistem pengaturan kecepatan motor BLDC, pernangkat lunak MATLAB dan Arduino digunakan untuk proses pengambilan data, perancangan kontroler, dan pengiriman data.

3.3.1 Perangkat Lunak MATLAB R2013a

Software yang digunakan pada perancangam sistem pengaturan kecepatan motor BLDC adalah *software* MATLAB R2013a. *Software Simulink* digunakan sebagai *Human Machine Interface* pada proses pengiriman dan penerimaan data melalui serial USB dari mikrokontroler Arduino.

Simulink MATLAB R2013a dapat mengolah data dari blok *serial receive* dan mengirimkan kembali melalui blok *serial send* melalui komunikasi serial Arduino. *Simulink* dapat digunakan untuk proses identifikasi sistem *open loop* maupun *closed loop* menggunakan blok *System Identification Toolbox*.

MATLAB R2013a dapat digunakan untuk keperluan proses perancangan kontroler PI berbasiskan *Neural-Fuzzy* Hibrida Adaptif. Perancangan kontroler PI berbasiskan *Neural-Fuzzy* Hibrida Adaptif dilakukan dengan cara membuat MATLAB *code* dan *Simulink*. Setelah itu, MATLAB R2013a digunakan untuk mensimulasikan hasil perancangan kontroler terhadap hasil pemodelan *plant* yang diperoleh. Terakhir, MATLAB R2013a digunakan sebagai antarmuka untuk melakukan implementasi kontroler yang telah dirancang. Blok Simulink identifikasi sistem ditunjukkan pada Gambar 3.7.



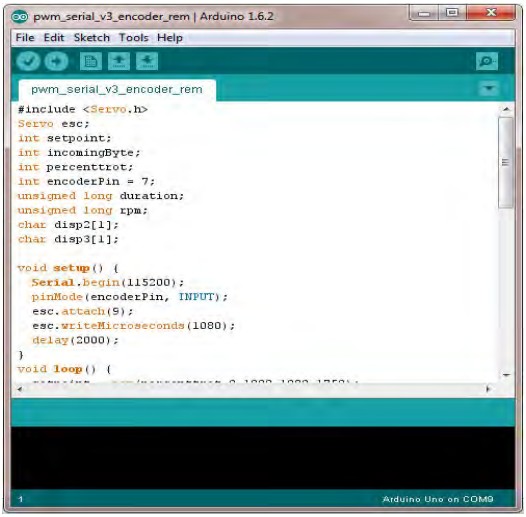
Gambar 3.7 Blok Simulink Identifikasi Sistem

Pada sisi penerima data, data kecepatan motor diterima oleh MATLAB R2013a dan dikeluarkan oleh blok serial *receive*. Data yang dikirimkan oleh Arduino merupakan data uint8 yang dikirimkan secara sepotong-sepotong sebesar 8 bit. Data pertama dikalikan dengan 256 dan ditambahkan dengan data kedua untuk mendapatkan kembali data uint16 kecepatan motor. Data kemudia di-*buffer* dengan blok *zero-order hold* dan dikonversi ke bentuk tipe data *double*.

3.3.2 *Software* Arduino

Selain Komponen-komponen dalam *plant* diakuisisi oleh mikrokontroler Arduino. *Software* yang sering digunakan untuk meng-*compile* bahasa pemrograman pada Arduino bernama Arduino IDE. Program akuisisi data yang dikehendaki ditulis pada Arduino IDE di komputer, lalu di-*compile* dan di-*upload* menuju mikrokontroler melalui serial USB.

Pada perancangan sistem pengaturan kecepatan motor BLDC, Arduino Uno digunakan untuk mengukur kecepatan melalui rangkaian pembaca kecepatan yang digunakan dan mengatur sinyal kontrol untuk menggerakan motor BLDC. Tampilan *software* Arduino IDE ditunjukkan pada Gambar 3.8.

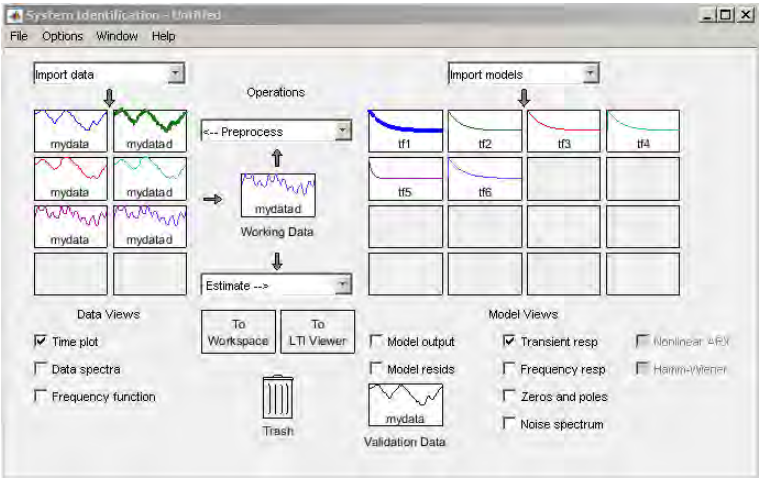


Gambar 3.8 Tampilan Arduino IDE

3.4 Identifikasi dan Pemodelan Sistem

Identifikasi pada pemodelan sistem pengaturan kecepatan motor BLDC digunakan identifikasi dinamis. Model matematika *plant* didapatkan melalui *toolbox matlab system identification* dengan cara meng-*input*-kan data *input-output*, melakukan *preprocessing*, dan memilih bentuk model yang diinginkan. Dalam proses identifikasi, data *input-output* yang ada dimasukkan ke dalam aplikasi dengan memilih "*time domain data"* saat menekan akan timbul tulisan "*import data"*. Setelah itu dilakukan *preprocessing*.

Data yang telah melalui tahap *preprocess* kemudian dipilih sebagai *working data* sekaligus *validation data* dengan cara *drag-and-drop* kotak yang berisi data tersebut ke kotak *working data* dan *validation data*. Kemudian dilakukan identifikasi model dengan memilih "*polynomial models*" *set*elah meng-klik *popout* bertuliskan "*estimate*". Tampilan dari *system identification toolbox* pada MATLAB ditunjukkan pada Gambar 3.9.



Gambar 3.9 Tampilan *System Identification Toobox* MATLAB

3.4.1 Metode Pembebanan *Plant*

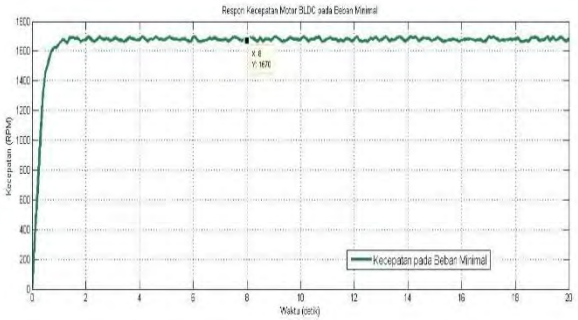
Pemodelan pada motor BLDC dilakukan pada tiga kondisi beban, yaitu beban minimal, nominal, dan maksimal.

3.4.2 Metode Identifikasi dan Pemodelan

Identifikasi pada motor BLDC menggunakan identifikasi statis. Hal ini dilakukan dengan cara memberikan *input* sinyal *step* ke *plant*. Pada tugas akhir ini sinyal *step* yang digunakan sebesar 0,9608.

3.4.2.1 Pembebanan Minimal

Respon kecepatan *open loop* motor BLDC pada beban minimal ditunjukkan pada Gambar 3.10.



Gambar 3.10 Respon Kecepatan *Open Loop* Motor BLDC pada Beban Minimal

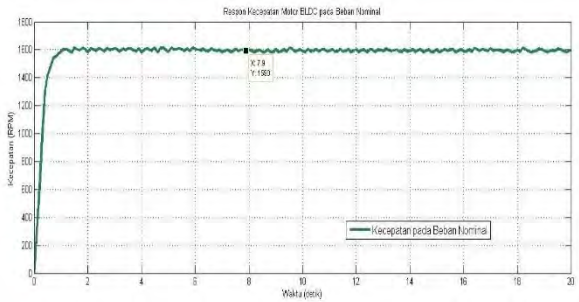
Pada kondisi beban minimal, rem elektromagnetik diberikan suplai tegangan 16 VDC. Model BLDC pada Kondisi Beban Minimal ditunjukkan pada Tabel 3.3.

Tabel 3.3 Model BLDC pada Kondisi Beban Minimal

Model	*Fit to Estimation*	MSE
$\frac{428{,}8s + 1279}{s^2 + 2{,}351s + 0{,}7318}$	78,9%	376,6

3.4.2.2 Pembebanan Nominal

Respon kecepatan *open loop* motor BLDC pada beban nominal ditunjukkan pada Gambar 3.11.



Gambar 3.11 Respon Kecepatan *Open Loop* Motor BLDC pada Beban Nominal

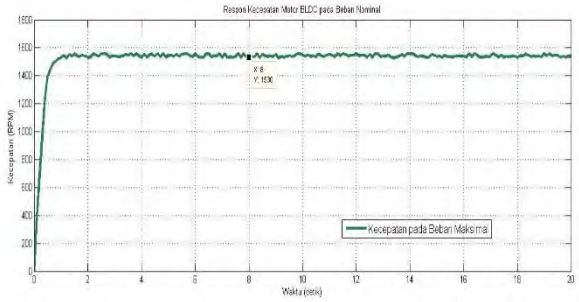
Pada kondisi beban nominal, rem elektromagnetik diberikan tegangan suplai 20 VDC. Model hasil identifikasi motor BLDC dengan pembebanan nominal ditunjukkan pada Tabel 3.4.

Tabel 3.4 Model BLDC pada Kondisi Beban Nominal

Model	Fit to Estimation	MSE
$\frac{536{,}2s + 1969}{s^2 + 3{,}461s + 1{,}185}$	78,12%	346,9

3.4.2.3 Pembebanan Maksimal

Respon kecepatan *open loop* motor BLDC pada beban maksimal ditunjukkan pada Gambar 3.12.



Gambar 3.12 Respon Kecepatan *Open Loop* Motor BLDC pada Beban Maksimal

Pada kondisi pembebanan maksimal, rem elektromagnetik diberikan tegangan suplai 24 VDC. Model yang didapatkan dari hasil identifikasi dalam keadaan pembebanan maksimal ditunjukkan pada Tabel 3.5.

Tabel 3.5 Model BLDC pada Kondisi Beban Maksimal

Model	Fit to Estimation	MSE
$\frac{460{,}3s + 1396}{s^2 + 2{,}47s + 0{,}8689}$	79,55%	271

3.4.3 Pengujian dan Validasi Model

Pengujian dan validasi model dilakukan untuk mengetahui kesesuaian model yang kita rancang dibandingkan dengan *plant* yang dimodelkan. Parameter yang digunakan untuk mengetahui keseuaian model adalah *Mean Square Error (MSE)*. Semakin kecil nilai MSE, maka model hasil pendekatan semakin sesuai dengan *plant* yang dimodelkan. Hasil Validasi Model BLDC ditampilkan pada Tabel 3.6.

Tabel 3.6 Hasil Validasi Model BLDC

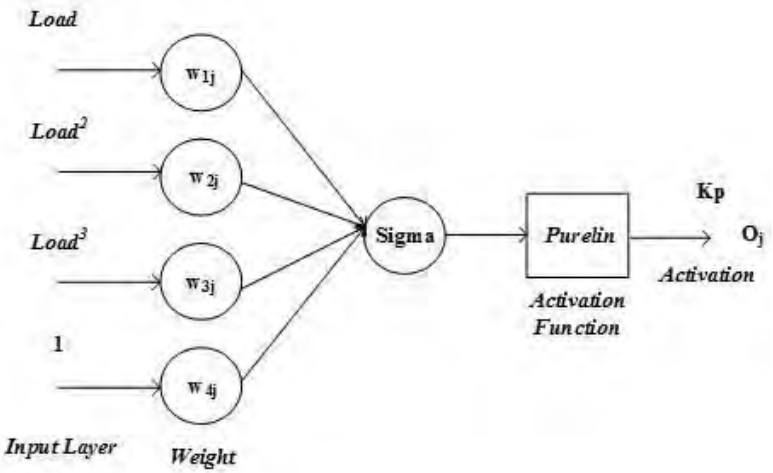
No	Model	*Fit to Estimation*	MSE
1	$\frac{428{,}8s + 1279}{s^2 + 2{,}351s + 0{,}7318}$	78,9%	376,6
2	$\frac{536{,}2s + 1969}{s^2 + 3{,}461s + 1{,}185}$	78,12%	346,9
3	$\frac{460{,}3s + 1396}{s^2 + 2{,}47s + 0{,}8689}$	79,55%	271

3.5 Perancangan Kontroler PI berbasis *Neural-Fuzzy* Hibrida Adaptif

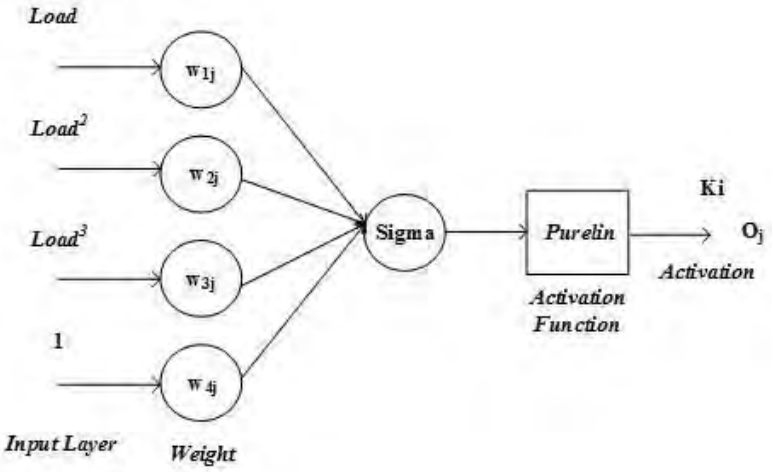
Setelah fungsi alih dari *plant* didapatkan kemudian melakukan perancangan kontroler untuk masing-masing pembebanan. Kontroler digunakan untuk mengembalikan respon ke nilai *set point* meskipun motor BLDC diberi beban. Tahapan desain kontroler meliputi perancangan kontroler PI berbasiskan *Neural-Fuzzy* Hibrida Adaptif.

3.5.1 Perancangan *Neural-Network*

Neural-Network digunakan sebagai *tuning* parameter PI untuk mengatur kecepatan motor BLDC.



Gambar 3.13 Struktur *Neural-Network* untuk Kp



Gambar 3.14 Struktur *Neural-Network* untuk Ki

Algoritma pembelajaran *neural-network* yang dipakai adalah *feed forward*. *Input layer neural-network* berupa nilai tegangan beban. Persamaan *input layer neural-network* ditunjukkan pada Persamaan 3.1.

$$O_j^1 = x(j)(j = 1,2, \dots, M) \tag{3.1}$$

M meruapakan jumlah variabel *input* pada *input layer*, pada *neural network* digunakan 4 variabel *input*. Variabel *input* dari *input layer* kemudian menuju *output layer*.

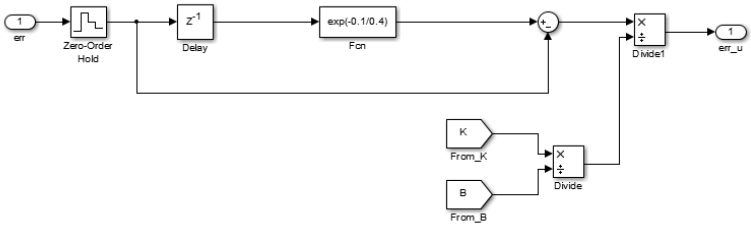
$$net^1 = \sum_{j=0}^{M} w_j^1 O_j^1 \tag{3.2}$$

$$O^2 = purelin(net^1) \tag{3.3}$$

Untuk merivisi bobot *neural-network* dipakai algoritma *gradient steepest descent* sehingga dipenuhi kriteria *error* kuadarat tiap saat minimum. Formulasi kriteria *error* kuadrat tiap saat minimum ditunjukkan pada Persamaan 3.4.

$$J = \frac{1}{2} e^2 \tag{3.4}$$

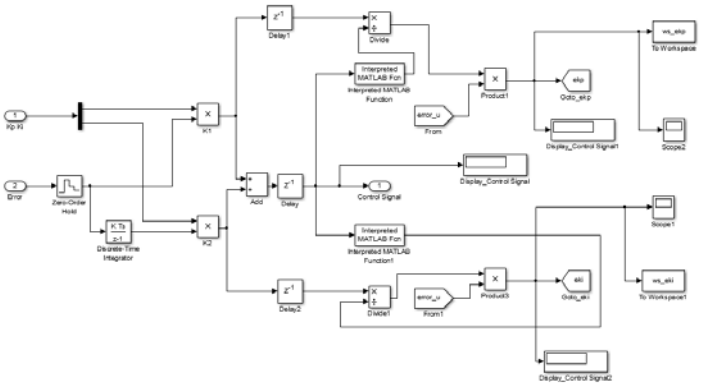
Error berupa *error* Kp dan *error* Ki. Untuk mendapatkan *error* Kp dan *error* Ki, nilai *error* sinyal kontrol harus diketahui terlebih dahulu. Blok Simulink dan persamaan *error* sinyal kontrol ditunjukkan pada Gambar 3.15 dan Persamaan 3.5.



Gambar 3.15 Blok Simulink *Error* Sinyal Kontrol

$$em(k) = e^{-\frac{Ts}{\tau}} * e(k-1)$$
$$e(u) = (em(k) - e(k)) * \frac{B}{K} \quad (3.5)$$

Error model, $em(k)$, merupakan nilai *error* yang harus dituju oleh nilai *error* sesungguhnya. $e(u)$ merupakan *error* sinyal kontrol *plant* pengaturan kecepatan motor BLDC. Kemudian, didapatakan nilai *error* Kp dan Ki yang ditunjukkan pada Persamaan 3.6 dan 3.7. Blok Simulink *error* Kp dan Ki ditunjukkan pada Gambar 3.16.



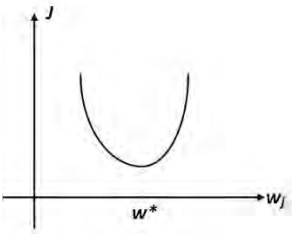
Gambar 3.16 Blok Simulink *Error* Kp dan Ki

$$ekp = \frac{e.Kp}{u} e(u) \quad (3.6)$$
$$eki = \frac{e.Ki}{u} e(u) \quad (3.7)$$

Kriteria *error* kuadrat akan minimum apabila memenuhi formulasi yang ditunjukkan pada Persamaan 3.8.

$$\frac{\partial J}{\partial w_j} = 0 \quad (3.8)$$

Karena kriteria *error* kuadarat tiap minimum, *J*, merupakan fungsi kuadrat dari w_j maka dapat digambarkan pada Gambar 3.17.



Gambar 3.17 Fungsi *J* terhadap w_j

Kemudian didapatkan persamaan $\frac{\partial J}{\partial w_j}$ yang ditunjukkan pada Persamaan 3.9.

$$\frac{\partial J}{\partial w_j} = \frac{\partial J}{\partial e}\frac{\partial e}{\partial O_j^2}\frac{\partial O_j^2}{\partial net^1}\frac{\partial net^1}{\partial w_j}$$
$$\frac{\partial J}{\partial w_j} = -ef'(O_j^2)x(j) \quad (3.9)$$

Kemudian didapatkan persamaan *gradient steepest descent* yang ditunjukkan pada Persamaan 3.10.

$$w_j(k+1) = w_j(k) + \Delta w * e * f'(O_j^2)x(j) \quad (3.10)$$

3.5.2 Perancangan Kontroler *Fuzzy* PI

Tahap perancangan kontroler *fuzzy* PI memiliki berberapa tahap yaitu penentuan fungsi keanggotaan dan bentuk fungsi keanggotaan kemudian membuat *rule base* dan menentukan fungsi keanggotaan pada *output*.

Kontroler *fuzzy* pada sistem pengaturan kecepatan motor BLDC berfungsi sebagai pengatur nilai Kp dan Ki pada kontroler PI, sehingga nilai Kp dan Ki menjadi fungsi keanggotaan pada *output* kontroler *fuzzy*. Pada *input fuzzy* memiliki 1 *input* yaitu besar beban pengereman yang diberikan.

3.5.2.1 Rule Base

Kontroler *fuzzy* PI digunakan untuk mengeluarkan nilai Kp dan Ki sesuai dengan kondisi pembebanan maka dalam *rule base* memiliki 3 pernyataan di setiap nilai Kp dan Ki. Misalkan dipilih representasi fungsi keanggotaan *input* sebagai berikut:

Beban Minimal = BK
Beban Nominal = BS
Beban Maksimal = BB

Maka *rule base* untuk kontroler *fuzzy* akan terlihat seperti berikut.

If X = BK *Then* K_P = 2,497369583
If X = BS *Then* K_P = 7,911902938
If X = BB *Then* K_p = 14,50763256

Pernyataan diatas merupakan *rule base* untuk pencarian nilai K_p. Apabila semakin besar pembebanan maka semakin besar nilai K_p yang digunakan. Kemudian menentukan *rule base* untuk Ki.

If X = BK *Then* K_i = 1,892985981
If X = BS *Then* K_i = 2,20533539
If X = BB *Then* K_i = 7,721486311

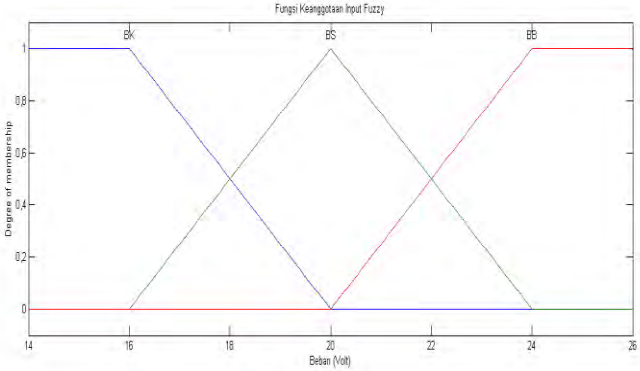
Pernyataan diatas merupakan *rule base* untuk pencarian nilai Ki. Apabila semakin besar pembebanan maka semakin besar nilai Ki yang digunakan.

3.5.2.2 Fungsi Keanggotaan

Fungsi keanggotaan untuk *input* pada kontroler *fuzzy* PI digunakan untuk mengubah nilai Kp dan Ki pada setiap pembebanan, maka *input* fungsi keanggotaan berupa nilai tegangan pembebanan. Karena sistem pengaturan kecepatan motor BLDC memiliki pembebanan minimal, nominal, dan maksimal maka pada fungsi keanggotaan memiliki 3 anggota. Berikut ini fungsi keanggotaan dari *input kontroler fuzzy*.

Pada Gambar 3.18 dapat dilihat bahwa jenis fungsi keanggotaan yang dipakai adalah jenis *triangular*. Fungsi keanggotaan *input* kontroler *fuzzy* terdiri dari 3 *fuzzy set* yaitu BK, BS, dan BB.

Pada fungsi keanggotaan *output* terdapat 3 anggota pada setiap Kp dan Ki. Bentuk dari fungsi keanggotaan hanya berupa konstanta atau *singleton* dari nilai Kp dan Ki. Proses defuzzifikasi menggunakan metode *center average* yaitu nilai Kp dan Ki yang telah didapatkan dikalikan dengan nilai bobot pada tiap *rule*-nya kemudian hasil kali antara *weight* dengan nilai Kp dan Ki dijumlahkan dan dibagi dengan jumlah total bobot.



Gambar 3.18 Fungsi Keanggotaan *Input*

BAB 4
PENGUJIAN DAN ANALISA

4.1 Pengujian Sistem

Pengujian sistem dilakukan agar komponen sistem yang telah dirancang dapat beroperasi sesuai desain. Pada tahap pengujian sistem dilakukan beberapa jenis pengujian, yaitu pengujian sensor, pengujian *open loop* dari motor BLDC, pengujian kontroler yang disimulasikan pada hasil identifikasi model, kemudian yang terakhir merupakan pengujian implementasi kontroler pada motor BLDC.

Pengujian pertama merupakan pengujian sensor kecepatan motor BLDC yang dilakukan dengan membandingkan hasil keluaran sensor kecepatan pada motor BLDC yang dibaca melalui Arduino dengan hasil pengukuran dari *tachometer*. Motor BLDC yang digunakan pada tugas akhir ini merupakan tipe motor *sensorless* sehingga tidak memerlukan sensor seperti sensor kecepatan dalam pengoperasiannya karena terdapat *feedback* dari motor yang memberikan informasi nilai kecepatan motor yang dibaca lewat Arduino. Dalam tahap pengujian ini dilakukan pengecekan hasil pembacaan sensor kecepatan melalui kabel dari *driver* didalam motor BLDC yang mengeluarkan sinyal kotak dengan *duty cycle* dengan kecepatan motor. Pengujian kedua merupakan pengujian sistem *open loop* pada motor BLDC yang dilakukan dengan memberi *input* berupa sinyal *step* pada motor BLDC dan dilihat hasil respon kecepatan yang terbaca dan dilihat hubungan keduanya. Pengujian ketiga merupakan simulasi sistem, hal ini dilakukan dengan cara memasang kontroler yang telah dibuat pada hasil identifikasi model motor BLDC dan yang terakhir merupakan implementasi kontroler pada motor BLDC.

4.2 Pengujian Sensor Kecepatan Motor BLDC

Pada pengujian sensor kecepatan motor BLDC dilakukan pengecekan hasil pembacaan sensor kecepatan. Hasil keluaran sensor dibandingkan dengan pengukuran menggunakan *laser tachometer*. Hasil pembacaan sensor kecepatan memiliki kesalahan pengukuran paling besar 1,93%. Hasil pengujian sensor kecepatan motor BLDC ditunjukkan pada Tabel 4.1.

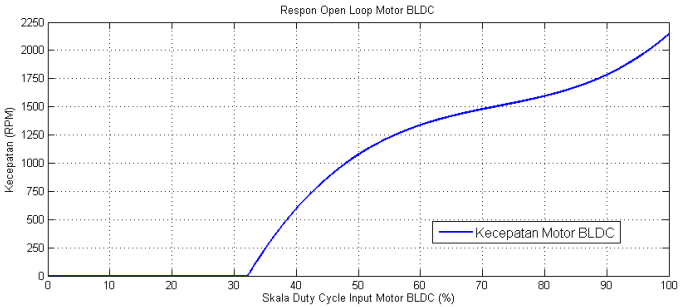
Tabel 4.1 Hasil Pengujian Sensor Kecepatan Motor BLDC

Tachometer **(RPM)**	**Sensor (RPM)**	***Error*** **(%)**		***Tachometer*** **(RPM)**	**Sensor (RPM)**	***Error*** **(%)**
972,36	984,6	1,26		1365,1	1382,67	1,29
972,12	984,6	1,28		1429,5	1448,82	1,35
971,82	984,6	1,32		1430,1	1448,82	1,31
1022,1	1035,41	1,30		1432,2	1448,82	1,16
1020,6	1035,41	1,45		1472,5	1491,69	1,30
1021,8	1035,41	1,33		1471,2	1491,69	1,39
1059,1	1070,83	1,11		1473,1	1491,69	1,26
1057,2	1070,83	1,29		1703,6	1726	1,31
1058,5	1070,83	1,16		1707,4	1726	1,09
1201,8	1218,04	1,35		1705,2	1726	1,22
1202,2	1218,04	1,32		2253	2290,45	1,66
1203,4	1218,04	1,22		2255	2290,45	1,57
1292,8	1310,42	1,36		2256	2290,45	1,53
1293,4	1310,42	1,32		2397	2441,21	1,84
1294,1	1310,42	1,26		2395	2441,21	1,93
1363,4	1382,67	1,41		2399	2441,21	1,76
1364,6	1382,67	1,32				

4.3 Pengujian *Open Loop* Motor BLDC

Pengujian *open loop* dilakukan untuk melihat hubungan antara *input* yang berupa *duty cycle* dan *output* yang berupa kecepatan dari motor BLDC. *Driver* BLDC diberikan masukan berupa sinyal *step* dan kemudian diukur kecepatannya. Hasil pengujian *open loop* kecepatan motor ditunjukkan dalam Gambar 4.1.

Pada kondisi awal, terjadi lonjakan kecepatan hingga sekitar 2216 RPM. Hal ini karena mekanisme *starting driver* motor. Saat motor mulai berjalan dari keadaan berhenti, *driver* secara otomatis memberikan masukan tegangan PWM 5 V sampai ggl balik di setiap fasa terdeteksi. Setelah itu proses komutasi baru berjalan normal.



Gambar 4.1 Hubungan *Input Output* Kecepatan Motor

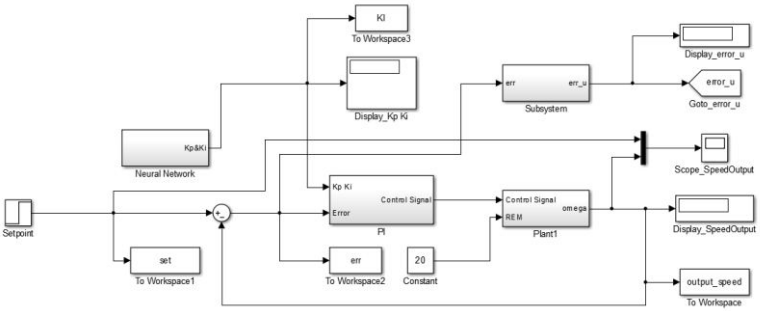
Pada kecepatan 1000-1500 rpm nilai kecepatan *output* bernilai linier terhadap besar tegangan *input*, oleh karena itu dalam tugas akhir ini dipilih rentang kerja motor BLDC 1300 rpm.

4.4 Simulasi Sistem

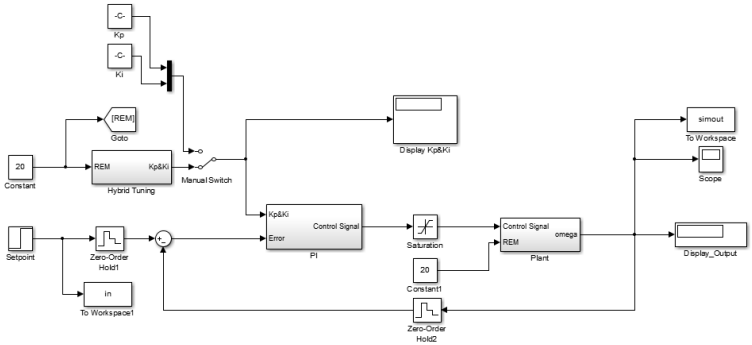
Simulasi merupakan salah satu hal yang harus dilakukan sebelum mengimplementasikan kontroler pada *plant*. Dengan hasil simulasi yang sesuai akan mempermudah dalam penerapan kontroler agar sistem mencapai kriteria yang diinginkan oleh penulis.

4.4.1 Blok Simulink Simulasi Sistem

Blok Simulink proses *learning neural network* untuk mendapatkan Kp dan Ki dan *fuzzy* PI ditunjukkan pada Gambar 4.2 dan 4.3.

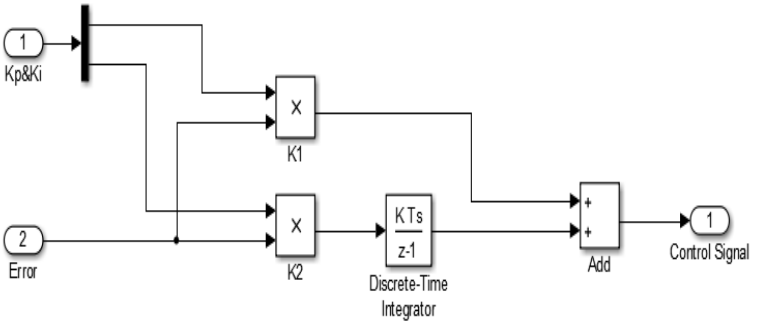


Gambar 4.2 Blok Simulink Proses *Learning Neural Network*

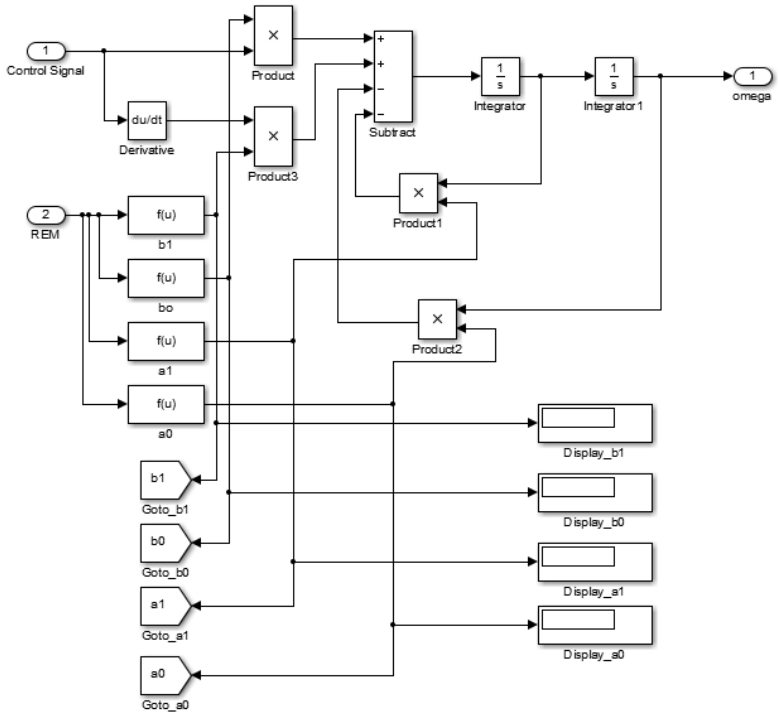


Gambar 4.3 Blok Simulink *Fuzzy* PI

Pada Gambar 4.2 dan 4.3 ditunjukkan untuk simulasi sistem pengaturan motor BLDC berupa sistem *close-loop* dengan *negative feedback*. Simulasi berupa sistem *close-loop* karena terdapat *summing point* yang membandingkan nilai *set point* dengan *feedback* kecepatan motor BLDC. Pada Gambar 4.2 merupakan proses *learning neural network* untuk mendapatakan Kp dan Ki kemudian pada Gambar 4.3 merupakan simulasi kontroler *fuzzy* PI. Kontroler *fuzzy* PI dirancang untuk mengendalikan *plant* ketika *online*, sedangkan *neural network* dijalankan ketika *plant offline*. Model *plant* yang digunakan dalam bentuk *transfer function* berganti bergantung nilai beban yang diberikan. Blok kontroler PI dan Simulink *plant* ditunjukkan pada Gambar 4.4 dan 4.5.



Gambar 4.4 Blok Simulink Kontroler PI



Gambar 4.5 Blok Simulink *Plant*

4.5 Simulasi Sistem Menggunakan Kontroler PI berbasiskan *Neural Fuzzy* Hibrida Adaptif

Kontroler PI berbasiskan *Neural Fuzzy* Hibrida Adaptif merupakan kontroler *fuzzy* PI dimana *input fuzzy* berupa tegangan beban dengan fungsi keangotaan *output* berupa nilai Kp dan Ki. Pemilihan nilai Kp dan Ki sesuai dengan tegangan beban, yaitu tegangan beban minimal, nominal, dan maksimal. Nilai Kp dan Ki didapatkan dari *neural network*.

4.5.1 Simulasi Sistem dengan Kontroler PI *Neural-Network*

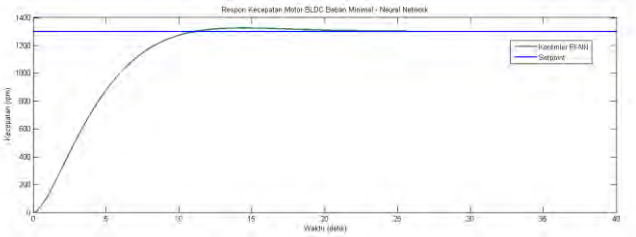
Simulasi sistem dengan kontroler PI *neural-network* dilakukan pada kondisi beban minimal, nominal, dan maksimal. Kontroler PI yang telah dirancang kemudian dijalankan pada model hasil identifikasi motor BLDC. Pengujian ini dilakukan dengan memberikan *input step* pada

kondisi beban minimal, nominal, dan maksimal. Setelah itu, respon dianalisis untuk mencari nilai *settling time*, *overshoot*, dan *time constant*.

Settling time adalah nilai di mana respon *output* telah memasuki zona *steady state* respon, nilai *overshoot* yang dilakukan adalah menghitung nilai nilai maksimum respon lalu dikurangi nilai *steady state* respon lalu dibagi dengan nilai *steady state* respon, dan *time constant* adalah waktu yang dibutuhkan sistem untuk mencapai nilai 63,2% dari *steady state*.

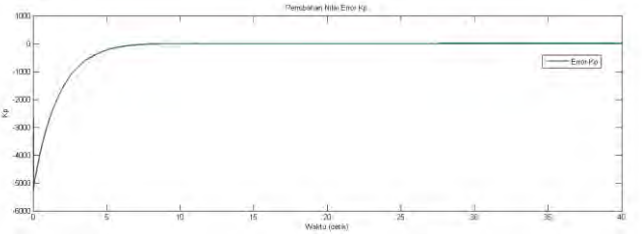
4.5.1.1 Simulasi Sistem dengan Kontroler PI Neural-Network pada Beban Minimal

Pada beban minimal, PI *neural network* menghasilkan respon yang ditunjukkan pada Gambar 4.6.

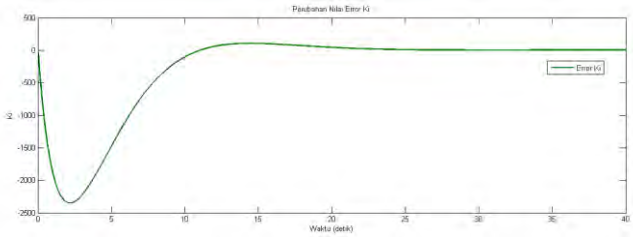


Gambar 4.6 Respon Kecepatan Motor BLDC PI-NN Beban Minimal

Setpoint kecepatan motor BLDC yang diberikan sebesar 1300 RPM dan beban yang digunakan adalah beban minimal dari 0-40 detik. Karakteristik respon hasil simulasi memiliki *settling time* 10 detik dan memiliki *overshoot* sebesar 1,9%, dan *time constant* 4,6 detik. Perubahan *error* Kp dan Ki pada beban minimal ditunjukkan pada Gambar 4.7 dan 4.8.



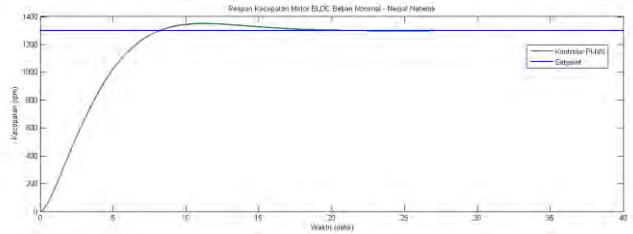
Gambar 4.7 Perubahan *Error* Kp pada Beban Minimal



Gambar 4.8 Perubahan *Error* Ki pada Beban Minimal

4.5.1.2 Simulasi Sistem dengan Kontroler PI Neural-Network pada Beban Nominal

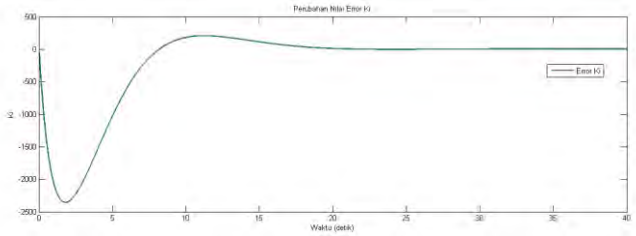
Pada beban nominal, PI *neural network* menghasilkan respon yang ditunjukkan pada Gambar 4.9.



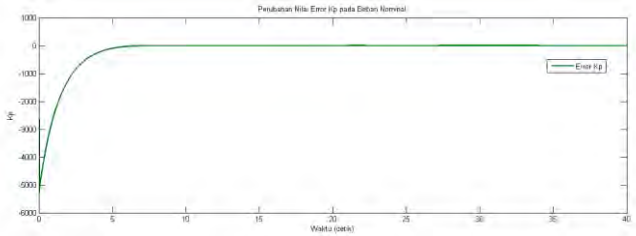
Gambar 4.9 Respon Kecepatan Motor BLDC PI-NN Beban Nominal

Setpoint kecepatan motor BLDC yang diberikan sebesar 1300 RPM dan beban yang digunakan adalah beban nominal dari 0-40 detik. Karakteristik respon hasil simulasi memiliki *settling time* 8,2 detik dan memiliki *overshoot* sebesar 3,46%, dan *time constant* 3,8 detik. Perubahan *error* Kp dan Ki pada beban nominal ditunjukkan pada Gambar 4.10 dan 4.11.

Gambar 4.10 dan 4.11 menunjukkan bahwa perubahan *error* Kp maupun Ki menuju nilai nol, sehingga nilai Kp dan Ki sistem sudah sesuai untuk menghasilkan respon sesuai *setpoint*.



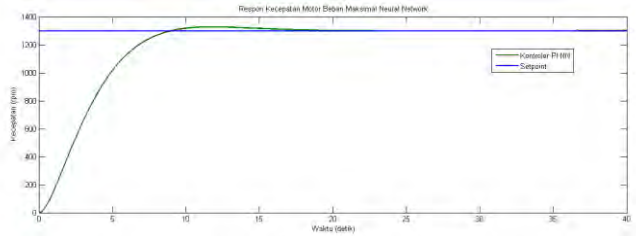
Gambar 4.10 Perubahan *Error* Kp pada Beban Nominal



Gambar 4.11 Perubahan *Error* Ki pada Beban Nominal

4.5.1.3 Simulasi Sistem dengan Kontroler PI Neural-Network pada Beban Maksimal

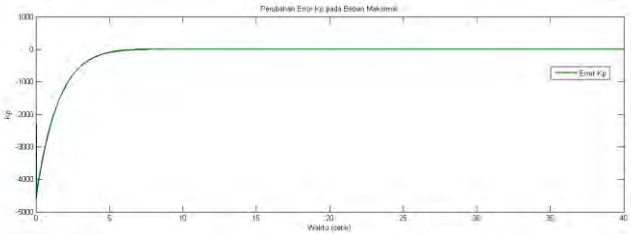
Pada beban maksimal, PI *neural network* menghasilkan respon yang ditunjukkan pada Gambar 4.12.



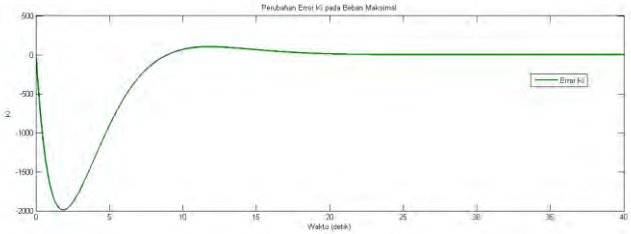
Gambar 4.12 Respon Kecepatan Motor BLDC PI-NN Beban Maksimal

Setpoint kecepatan motor BLDC yang diberikan sebesar 1300 RPM dan beban yang digunakan adalah beban maksimal dari 0-40 detik. Karakteristik respon hasil simulasi memiliki *settling time* 9 detik dan memiliki *overshoot* sebesar 1,5%, dan *time constant* 3,8 detik. Perubahan

error Kp dan Ki pada beban maksimal ditunjukkan pada Gambar 4.13 dan 4.14.



Gambar 4.13 Perubahan *Error* Kp pada Beban Maksimal



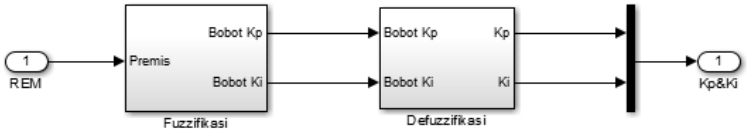
Gambar 4.14 Perubahan *Error* Ki pada Beban Maksimal

4.5.2 Simulasi Sistem dengan Kontroler *Fuzzy* PI

Pada Gambar 4.3 ditunjukkan simulasi sistem pengaturan kecepatan motor BLDC ketika *plant online*. Pada simulai pada Gambar 4.3 terdapat dua kontroler yaitu kontroler *fuzzy* dan kontroler PI. Kontroler *fuzzy* menentukan Kp dan Ki pada kontroler PI berdasarkan pada *input* tegangan beban rem.

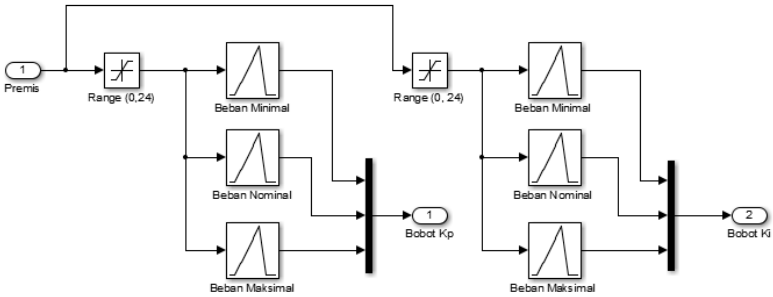
Pada Gambar 4.4 merupakan blok Simulink untuk kontroler PI. Kontroler PI pada Gambar 4.4 memiliki Kp dan Ki yang dipisahkan melalui Mux. Kp dan Ki didapatkam dari proses *learning neural-network*.

Blok Simulink kontroler *fuzzy* terdiri dari dua tahapan yaitu fuzzifikasi yang mengubah nilai pembebanan yang masuk menjadi nilai bobot untuk setiap Kp dan Ki dan blok defuzzifikasi mendapatkan *input* berupa nilai bobot untuk setiap Kp dan Ki dan mengeluarkan *output* berupa Kp dan Ki untuk kontroler PI. Blok Simulink kontroler *fuzzy* ditunjukkan pada Gambar 4.15.



Gambar 4.15 Blok Simulink Kontroler *Fuzzy*

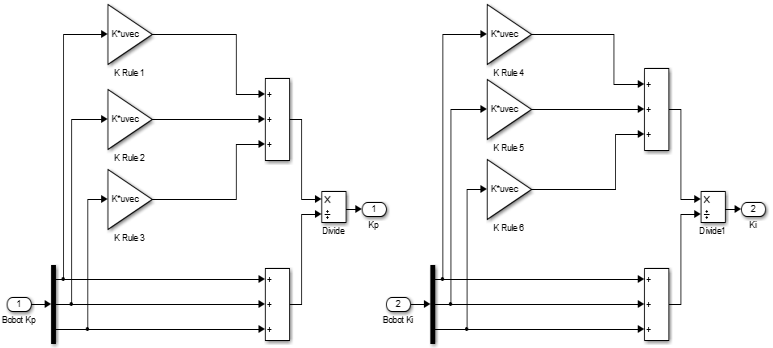
Fuzzy memliki 4 elemen yaitu *fuzzy rule base*, *fuzzy inference mechanism*, fuzzifikasi, defuzzifikasi. Pada blok Simulink Gambar 4.15 merupakan blok fuzzifikasi dan blok defuzzifikasi. Blok Simulink fuzzifikasi ditunjukkan pada Gambar 4.16.



Gambar 4.16 Blok Simulink Fuzzifikasi

Fuzzifikasi memiliki 3 fungsi keanggotaan yaitu beban minimal, beban nominal, dan beban maksimal yang diwakili dengan fungsi keanggotaan yang berbentuk *triangular*, yang memiliki nilai tengah 16V, 20V, dan 24V. *Output* blok Simulink fuzzifikasi menghasilkan tiga bobot untuk setiap *input* di setiap nilai Kp dan Ki sehingga jumlah keseluruhan bobot adalah 6. Nilai bobot diolah pada blok Simulink defuzzifikasi. Blok Simulink defuzzifikasi ditunjukkan pada Gambar 4.17.

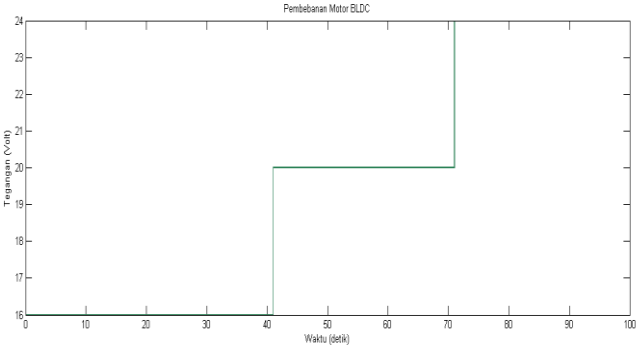
Defuzzifikasi menggunakan metode *center average* yang memiliki 6 nilai bobot kemudian bobot dikalikan dengan setiap fungsi keanggotaan *output* yang telah didapatkan dari proses *learning neural-network*. Nilai bobot yang telah dikalikan dengan fungsi keanggotaan *output* kemudian dibagi dengan jumlah nilai bobot pada setiap fungsi keanggotaan *output* dan menghasilkan Kp dan Ki.



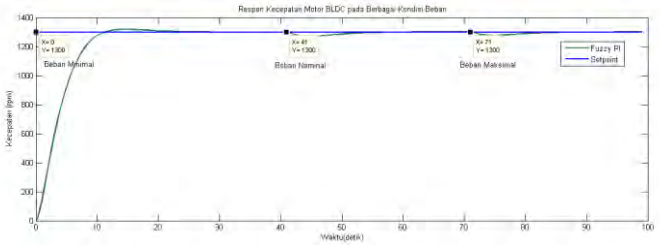
Gambar 4.17 Blok Simulink Defuzzifikasi

4.5.2.1 Simulasi Sistem dengan Kontroler Fuzzy PI pada Semua Kondisi Beban

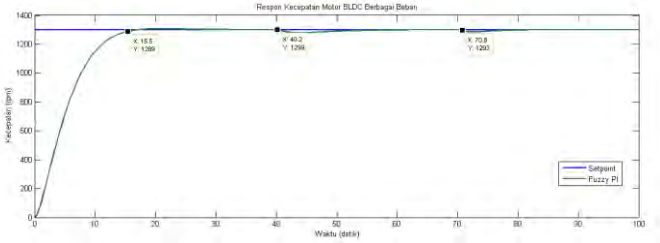
Pembebanan dan hasil respon simulasi dengan kontroler *fuzzy* PI dapat dilihat pada Gambar 4.18, 4.19 dan 4.20. *Setpoint* yang diberikan sebesar 1300 RPM dan beban yang digunakan adalah beban minimal pada 0-40 detik, beban nominal pada 41-70 detik, kemudian beban maksimal pada 71-99 detik. Karakteristik respon hasil simulasi memiliki *settling time* 11 detik, *overshoot* sebesar 1,5%, dan *time constant* sebesar 5 detik.



Gambar 4.18 Pembebanan pada Motor BLDC



Gambar 4.19 Respon Kecepatan Motor BLDC pada berbagai Beban



Gambar 4.20 Respon Kecepatan Motor BLDC pada berbagai Beban 2

4.6 Implementasi Sistem

Pada tahapan implementasi sistem kontroler yang telah dibuat dicoba dijalankan pada model hasil identifikasi sistem. Pengujian dilakukan dengan memberikan respon ***step*** di semua pembebanan mulai dari kondisi beban minimal, nominal, dan maksimal.

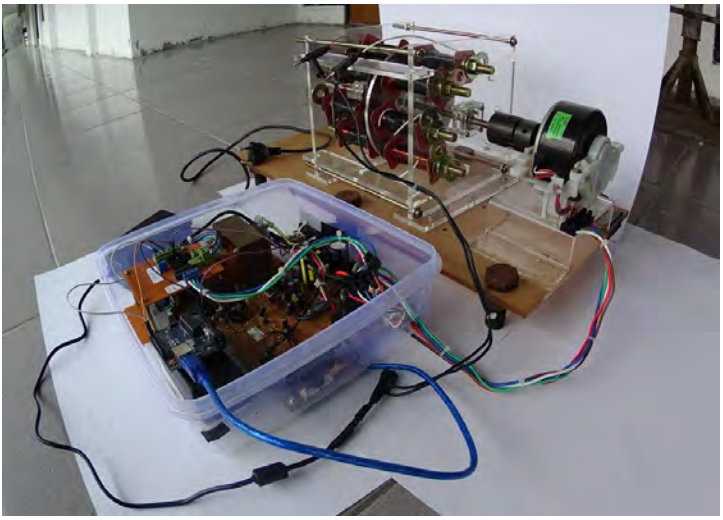
4.6.1 Realisasi Plant

Plant terbagi menjadi 3 sistem yaitu sistem yang bekerja menjalankan motor, sistem yang menjalankan rem, dan satu lagi sistem yang membaca sensor. Dari 3 sistem ini untuk bisa terhubung dengan kontroler harus memiliki *interface*. *Interface* yang digunakan adalah Arduino Uno.

4.6.1.1 Realisasi Fisik Plant

Pada Gambar 4.21 ditunjukkan realisasi fisik *plant.* Pada *plant* terdiri dari motor BLDC, rem elektromagnetik, rangkaian pembaca tegangan ***supply*** untuk rem magnetik, rangkaian pembaca kecepatan motor BLDC yang terdiri dari rangkaian isolator dan rangkaian *voltage*

follower, adaptor DC sebagai *supply* tegangan untuk rem elektromagnetik kemudian terdapat kontroler motor yang berfungsi untuk memberikan *supply* daya pada motor BLDC dan didalam motor juga terdapat *driver*, Arduino juga digunakan sebagai *interface*.



Gambar 4.21 *Plant* Motor BLDC [3]

4.6.1.2 Program Arduino

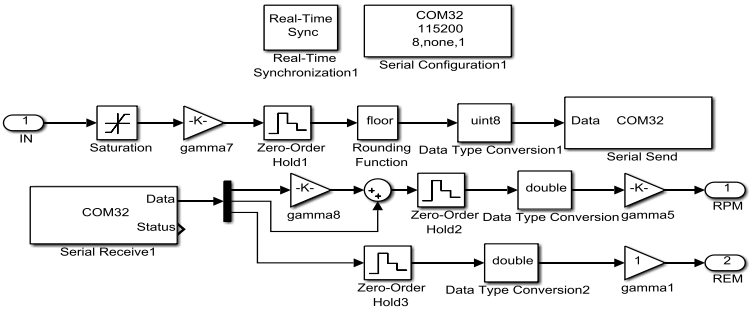
Program Arduino sebagai pengirim data dan penerima data dari komputer. Program Arduino memiliki 2 sistem utama yaitu inisialisasi dan program utama.

Pada program inisialisasi terdapat beberapa hal yang dilakukan yaitu menyebutkan jenis-jenis *library* yang dipakai, variabel-variabel yang dipakai, nilai *baudrate*, dan menyebutkan pin-pin yang dipakai pada Arduino. Terdapat beberapa jenis variabel yang dipakai diantaranya *integer*, *character*, dan *unsigned long* kemudian *baudrate* yang dipakai bernilai 115200, pin yang digunakan pada *plant* adalah pin 7 sebagai pembaca sensor kecepatan dan pin 6 yang mengeluarkan *output* PWM.

Pada program utama terbagi menjadi 2 buah sistem, pertama merupakan program yang membaca angka yang dikirim dari MATLAB, kemudian yang kedua merupakan program untuk membaca nilai kecepatan motor dalam satuan rpm.

4.6.2 Blok Implementasi Sistem

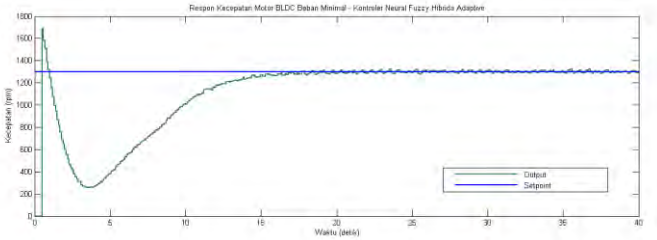
Blok implementasi sistem pada simulasi dan implementasi memiliki struktur yang sama, perbedaannya hanya dibagian blok pada *plant* saja. Jika pada simulasi sistem blok Simulink pada *plant* diisi dengan fungsi alih yang telah didapatkan dari identifikasi. Pada implementasi sistem, blok dari *plant* berisi blok komunikasi *serial* ditunjukkan pada Gambar 4.22.



Gambar 4.22 Blok Simulink Komunikasi Serial *Plant* Motor BLDC

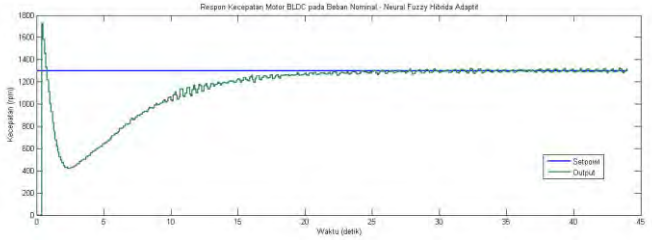
4.6.3 Pengujian Respon Motor BLDC dengan Kontroler PI berbasiskan *Neural Fuzzy* Hibrida Adaptif

Pada implementasi sistem diberikan *setpoint* berupa sinyal *step* yang bernilai 1300 rpm. Pengujian dilakukan pada setiap beban dengan *setpoint* yang sama. Setelah itu respon motor BLDC dianalisis dengan mencari nilai *settling time*, *overshoot*, dan *time constant*. Respon kecepatan motor BLDC ketika beban minimal, nominal, maksimal, dan berbagai beban ditunjukkan pada Gambar 4.23, 4.24, 4.25, dan 4.26.



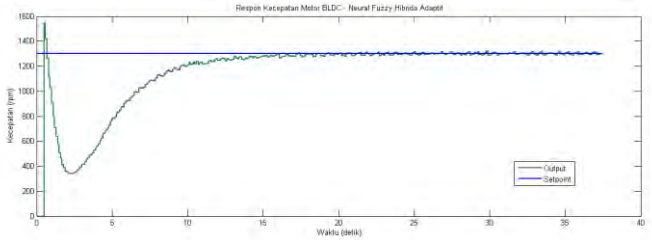
Gambar 4.23 Implementasi Respon Kecepatan Motor BLDC Beban Minimal.

Setpoint kecepatan motor BLDC yang diberikan sebesar 1300 RPM dan beban yang digunakan adalah beban minimal dari 0-40 detik. Karakteristik respon hasil simulasi memiliki *settling time* 15 detik dan memiliki *overshoot* sebesar 1,5%, dan *time constant* 8 detik



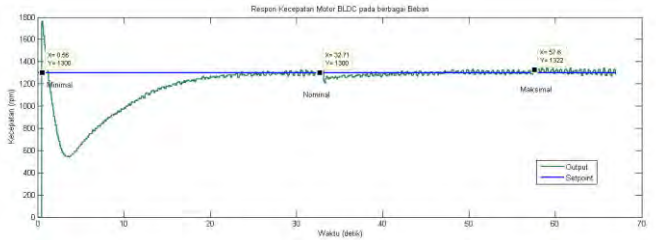
Gambar 4.24 Implementasi Respon Kecepatan Motor BLDC Beban Nominal

Setpoint kecepatan motor BLDC yang diberikan sebesar 1300 RPM dan beban yang digunakan adalah beban nominal dari 0-40 detik. Karakteristik respon hasil simulasi memiliki *settling time* 17,7 detik dan memiliki *overshoot* sebesar 1,3%, dan *time constant* 9 detik.



Gambar 4.25 Implementasi Respon Kecepatan Motor BLDC Beban Maksimal

Setpoint kecepatan motor BLDC yang diberikan sebesar 1300 RPM dan beban yang digunakan adalah beban maksimal dari 0-40 detik. Karakteristik respon hasil simulasi memiliki *settling time* 13 detik dan memiliki *overshoot* sebesar 1,1%, dan *time constant* 5 detik.



Gambar 4.26 Implementasi Respon Kecepatan Motor BLDC pada berbagai Beban

Setpoint kecepatan motor BLDC yang diberikan sebesar 1300 RPM pada berbagai beban. Karakteristik respon hasil simulasi memiliki *settling time* 20 detik dan memiliki *overshoot* sebesar 1,1%, dan *time constant* 7,7 detik.

LAMPIRAN

Program Arduino

```
#include <Wire.h>
int pin = 7;
int out = 6;
byte u = 255;
unsigned long duration;
unsigned long spd=0;
word kec;
byte w1;
byte w2;

void setup()
{
  Serial.begin(115200);
  pinMode(pin,INPUT);
  pinMode(out,OUTPUT);
  analogWrite(out,u);
}

void loop()
{
  //sinyal kontrol
  if (Serial.available()) {
    u=255-Serial.read();
    analogWrite(out,u);
    baca();
    }
}

void baca()
{
//baca kecepatan motor
  duration = pulseIn(pin, LOW, 30000);
     if (duration == 0)
      {
        spd = 0;
```

```
      }
      else{
        spd=(15*1000000/(2*duration));
        }
  kec=int(spd);

  //kirim ke matlab
    //kecepatan
    w1=kec/256;
    w2=kec%256;
    Serial.write(w1);
    Serial.write(w2);
    Serial.write(beban);
}
```

BAB 5
PENUTUP

5.1 Kesimpulan

Dari analisis yang telah dilakukan terhadap hasil simulasi maka dapat disimpulkan bahwa pengaturan kecepatan menggunakan kontroler PI berbasiskan *Neural Fuzzy* Hibrida Adaptif menghasilkan respon pada berbagai beban dengan karakteristik respon hasil simulasi memiliki *settling time* 11 detik, *overshoot* sebesar 1,5%, dan *time constant* sebesar 5 detik sedangkan terhadap hasil implementasi memiliki karakteristik respon *settling time* 20 detik, *overshoot* sebesar 1,1%, dan *time constant* 7,7 detik.

5.2 Saran

Untuk penelitian selanjutnya disarankan agar membuat *sensor* arus untuk membaca nilai arus pada rem elektromagnetik dikarenakan pada perancangan Tugas Akhir ini dianggap bahwa resistansi dari rem elektromagnetik tetap sehingga nilai arus akan sebanding dengan nilai tegangan.

(halaman ini sengaja dikosongkan)

DAFTAR PUSTAKA

[1] ....., "*Brushless DC Motor Engineering*", <URL: http://www.nmbtc.com/brushless-dc motors/engineering/brushless_dc_motors_engineering/>, 2007.

[2] Husaini, Achmad Nur, "*Prinsip Kerja Motor Brushless DC (BLDC Motor)*", <URL: http://www.insinyoer.com/prinsip-kerja-motor-brushless-dc-bldc-motor/3/>, September, 2015.

[3] M. Safrurriza, "Desain Sistem Pengaturan Kecepatan Motor Arus Searah Tanpa Sikat Menggunakan PID Mutiobjektif Berdasarkan Algoritma Genetika", *Tugas Akhir*, Jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember, Surabaya, 2016.

[4] Fairuzza D., "Pengaturan Kecepatan Motor Brushless DC Menggunakan Kontroler Fuzzy Berbasis Linear Quadratic Regulator", *Tugas Akhir*, Jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember, Surabaya, 2016.

[5] ....., "*The Relative Motion Between A Conductor And Magnetic Field Is Used To Generate An Electrical Voltage*", <URL:http://www.boredofstudies.org/wiki/The_relative_motion_between_a_conductor_and_magnetic_field_is_used_to_generate_an_electrical_Voltage>, April, 2015.

[6] Kho, Dickson, "*Pengertian Optocoupler dan Prinsip Kerjanya*", <URL: http://teknikelektronika.com/pengertian-*optocoupler*-fungsi-prinsip-kerja-*optocoupler*/>, April, 2014.

[7] ....., "*Arduino*", <URL: http://arduino.cc/>, Mei, 2015.

[8] Guntur P., "Perancangan Kontrol Kecepatan Motor Arus Searah Tanpa Sikat Menggunakan Sliding Mode Berbasis PID", *Tugas Akhir*, Jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember, Surabaya, 2016.

[9] Passino, Kevin M., Stephen Yurkovich, "*Fuzzy Control*", California: Addison Wesley Longman, Inc, 1998.

[10] Mansouri, Mahdi, *"*Hybrid Adaptive Neuro Fuzzy Tuned PI*", 4th Power Electronics, Drive Systems & Technologies Conference*, Februari, 2013.

[11] Beny, Setiyadi Hidayat, "Perancangan Dan Implementasi Pengaturan Kecepatan Motor Arus Searah Tanpa Sikat Menggunakan Metode Kontrol Neuro-Fuzzy", *Tugas Akhir*, Jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember, Surabaya, 2015.

RIWAYAT HIDUP

Agung Setyadi Wicaksono, lahir di Sukoharjo, 19 Mei. Riwayat pendidikannya, menamatkan pendidikan dasar di SD Negeri 2 Sukoharjo, pendidikan menengah di SMP Negeri 1 Sukoharjo, dan pendidikan tinggi di SMA Negeri 1 Sukoharjo. Saat ini telah menempuh kuliah di Jurusan Teknik Elektro Institut Teknologi Sepuluh Nopember dengan mengambil bidang studi Teknik Sistem Pengaturan. Selama kuliah penulis aktif dalam beberapa kegiatan akademis maupun non akademis. Penulis aktif dalam UKM Tenis Lapangan ITS dan kegiatan perlombaan dan keilmiahan seperti seminar dan berbagai macam pelatihan bertema keilmiahan. Penulis dapat dihubungi melalui email: **agung.setyadi.w12@mhs.ee.its.ac.id**