



TESIS - TE142599

**RANCANG BANGUN SISTEM KONTROL
KECEPATAN MOTOR BLDC MENGGUNAKAN
ANFIS UNTUK APLIKASI SEPEDA MOTOR LISTRIK**

MOHAMAD RIDWAN
07111550010202

DOSEN PEMBIMBING
Heri Suryoatmojo, ST., MT., Ph.D
Dedet Candra Riawan, ST., M.Eng., Ph.D.

PROGRAM MAGISTER
BIDANG KEAHLIAN TEKNIK SISTEM TENAGA
DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI ELEKTRO
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2018



TESIS - TE142599

**RANCANG BANGUN SISTEM KONTROL
KECEPATAN MOTOR BLDC MENGGUNAKAN
ANFIS UNTUK APLIKASI SEPEDA MOTOR LISTRIK**

MOHAMAD RIDWAN

07111550010202

DOSEN PEMBIMBING

Heri Suryoatmojo, ST., MT., Ph.D

Dedet Candra Riawan, ST., M.Eng., Ph.D.

PROGRAM MAGISTER

BIDANG KEAHLIAN TEKNIK SISTEM TENAGA

DEPARTEMEN TEKNIK ELEKTRO

FAKULTAS TEKNOLOGI ELEKTRO

INSTITUT TEKNOLOGI SEPULUH NOPEMBER

SURABAYA

2018

LEMBAR PENGESAHAN

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Teknik (M.T)

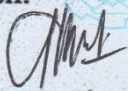
di
Institut Teknologi Sepuluh Nopember

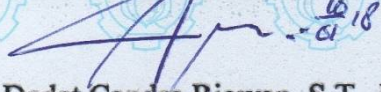
oleh:

Mohamad Ridwan
NRP. 07111550010202


Tanggal Ujian : 19 Desember 2017
Periode Wisuda : Maret 2018

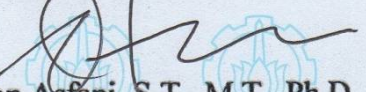
Disetujui oleh:


1. Heri Suryoatmojo, S.T., M.T., Ph.D. (Pembimbing I)
NIP: 198006032006041003


2. Dedet Candra Riawan, S.T., M.Eng., Ph.D. (Pembimbing II)
NIP: 197311192000031001


3. Dr. Ir. Soediby, M.MT. (Penguji)
NIP: 195512071980031004


4. Dr. Eng. Ardyono Priyadi, S.T., M.Eng. (Penguji)
NIP: 197309271998031004


5. Dimas Anton Asfani, S.T., M.T., Ph.D. (Penguji)
NIP: 198109052005011002

Dekan Fakultas Teknologi Elektro


Dr. Tri Arief Sardjono, S.T., M.T.
NIP: 197002121995121001

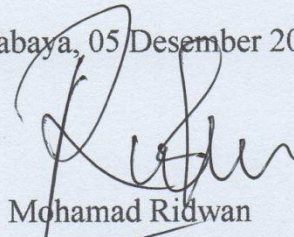
Halaman ini sengaja dikosongkan

PERNYATAAN KEASLIAN TESIS

Dengan ini saya menyatakan bahwa isi keseluruhan tesis saya dengan judul “**RANCANG BANGUN SISTEM KONTROL KECEPATAN MOTOR BLDC MENGGUNAKAN ANFIS UNTUK APLIKASI SEPEDA MOTOR LISTRIK**” adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 05 Desember 2017



Mohamad Ridwan

NRP. 07111550010202

Halaman ini sengaja dikosongkan

RANCANG BANGUN SISTEM KONTROL KECEPATAN MOTOR BLDC MENGGUNAKAN ANFIS UNTUK APLIKASI SEPEDA MOTOR LISTRIK

Nama mahasiswa : Mohamad Ridwan
NRP : 07111550010202
Pembimbing : 1. Heri Suryoatmojo, ST., MT., Ph.D
2. Dedet Candra Riawan, ST., M.Eng., Ph.D.

ABSTRAK

Sistem kontrol kecepatan motor BLDC menggunakan ANFIS telah didesain dan diimplementasikan. Algoritma ANFIS mampu mengontrol kecepatan motor BLDC sesuai dengan nilai referensi yang diinginkan. Rata-rata error *steady state* yang dicapai dengan menggunakan ANFIS adalah sebesar 0,1 % dengan *rise time* sebesar 2,7437 s untuk kecepatan referensi sebesar 4000 rpm. Proses pembelajaran ANFIS menggunakan metode *hybrid* PSO dan RLSE dengan supervisi dari Fuzzy-PID. PSO dan RLSE dapat mentraining data ANFIS multi output dengan sangat baik. Data training terbaik dicapai saat nilai $\lambda = 1$ dengan error RMSE sebesar 0,05364. Waktu eksekusi algoritma ANFIS pada mikrokontroler adalah sebesar 96 us.

Kata kunci: motor BLDC, sistem kontrol, *fuzzy logic*, PID, ANFIS, PSO, RLSE, kendaraan listrik.

Halaman ini sengaja dikosongkan

DESIGN AND IMPLEMENTATION OF BLDC MOTOR SPEED CONTROLLER USING ANFIS FOR ELECTRIC MOTORCYCLE APPLICATION

By : Mohamad Ridwan
Student Identity Number : 07111550010202
Supervisors : 1. Heri Suryoatmojo, ST., MT., Ph.D
2. Dedet Candra Riawan, ST., M.Eng., Ph.D.

ABSTRACT

BLDC motor speed control system using ANFIS has been designed and implemented. ANFIS algorithm is able to control the speed of the BLDC motor according to the desired reference value. The average of steady state error achieved using ANFIS is 0.1% and the rise time is 2.7437 s when the reference speed is 4000 rpm. ANFIS learning process uses hybrid PSO and RLSE methods supervised by Fuzzy-PID. PSO and RLSE can train the ANFIS multi-output data very well. The best training data is achieved when the value of $\lambda = 1$ with RMSE error of 0.05364. The execution time of ANFIS algorithm on microcontroller is 96 us.

Keywords: BLDC motor, control system, fuzzy logic, PID, ANFIS, PSO, RLSE, electric vehicle

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Bismillahirrohmanirrohim. Dengan menyebut nama Allah Yang Maha Pengasih lagi Maha Penyayang. *Alhamdulillah*, segala puji bagi Allah *Subhanahu Wata'ala*, yang telah menciptakan langit dan bumi, dan Maha Mengetahui segalanya, Sang Pemilik lautan ilmu seluruhnya. *Alhamdulillah*, atas bimbingan dan karunia-Nya, pada akhirnya tesis tentang kontrol kecepatan motor BLDC menggunakan ANFIS telah selesai dilakukan. Semoga dari hasil yang didapatkan pada penelitian ini dapat dimanfaatkan dan benar-benar bisa untuk diimplementasikan untuk kendaraan listrik buatan anak negeri, Indonesia. Sholawat dan salam semoga tetap tercurahkan kepada junjungan Nabi Muhammad *Shollallahu 'Alaihi Wasallam* yang telah menunjukkan jalan yang lurus dan terang benderang bagi seluruh alam.

Terima kasih dan syukur kami yang tak ternilai kepada bapak dan ibu kami, Bapak Misraki dan Ibu Mulyani, yang telah membesarkan dan membimbing kami, yang mana tesis ini dapat diselesaikan tak luput juga dari do'a yang selalu beliau panjatkan. Teruntuk anggota keluarga kami, Kakak Khusnul Khotimah, Anang Mu'allim, keponakan-keponakan kami, Adik M. Rizal Al-Farizi dan Ananda Ziyana Zahra (Jihan), semoga Allah *Subhanahu Wata'ala* senantiasa memberikan rahmat dan perlindungan-Nya sehingga apa yang dicita-citakan dapat terwujud.

Terima kasih juga kami sampaikan kepada dosen pembimbing kami, Bapak Heri Suryoatmojo, S.T., M.T., Ph.D. dan Bapak Dedet Candra Riawan, S.T., M.Eng., Ph.D. yang telah meluangkan waktu, tenaga dan pikirannya untuk membimbing dalam menyelesaikan tesis ini. Ucapan terima kasih juga kami sampaikan kepada seluruh dosen pengajar S2 Teknik Elektro, khususnya seluruh dosen bidang studi Teknik Sistem Tenaga yang telah membagikan ilmunya, baik ilmu teknik maupun ilmu kehidupan serta dorongan semangat dan motivasi yang selalu beliau-beliau sampaikan dalam perkuliahan.

Tak lupa pula ucapan terima kasih kami kepada teman-teman S2 khususnya angkatan 2015 Genap, Trina Wati, Annisaa Taradini, Firilia Filiana, Niken Adriaty Basyarach, Isa Hafidz, Muhammad Khalil, Wildan Arif Febrianto,

Alfin Syahrin, Arief Budi Ksatria, dan Alfredo Gusmao, teman seperjuangan yang tak mengenal lelah demi tercapainya cita-cita, semoga selalu mendapatkan rahmat dan karunia Allah *Subhanahu Wata'ala* serta selalu dalam perlindungan-Nya. Kepada seluruh teman-teman S2 lintas angkatan yang tak bisa kami sebutkan satu per satu, dengan tidak mengurangi rasa hormat, rasa terima kasih kami haturkan. Semoga cita-cita tak hanya sebatas angan, tapi dalam genggaman tangan.

Sahabat spesial yang sekaligus telah menjadi bagian dari keluarga kami, R.A.H.E.L, Andri Santoso, Heriyanto, Edwin Eka Apriliyanto, Luka Andian Putra, beserta keluarga semuanya, mudah-mudahan Allah *Subhanahu Wata'ala* selalu mengeratkan tali persaudaraan kita, terkabul cita-cita dan selalu dalam perlindungan dan rahmat-Nya.

Seluruh teman-teman yang berada di Pusat Unggulan IPTEK Sistem Kontrol Otomotif (PUI-SKO) ITS, terima kasih atas dukungan dan motivasinya. Kepada Lembaga Pengelola Dana Pendidikan (LPDP) yang telah mendukung biaya kuliah serta penelitian yang kami lakukan. Terima kasih juga kami sampaikan kepada seluruh pihak yang turut membantu dalam menyelesaikan penelitian tesis ini. Akhirnya, mudah-mudahan apa yang menjadi setetes dari lautan ilmu ini bisa membawa kemanfa'atan dan keberkahan bagi kita semua.

Semoga Allah *Subhanahu Wata'ala* senantiasa memberikan rahmat, petunjuk, dan perlindungan-Nya bagi kita semua. Aamiin.

Surabaya, 05 Desember 2017

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN.....	iii
PERNYATAAN KEASLIAN TESIS	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL	xix
DAFTAR SIMBOL DAN SINGKATAN.....	xxi
BAB 1 PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Sistematika Penulisan	4
BAB 2 KAJIAN PUSTAKA.....	5
2.1. Motor BLDC	5
2.1.1. Model Matematika Motor BLDC	7
2.2. Skema Kontrol Motor BLDC	10
2.3. Kontrol Kecepatan Motor BLDC	11
2.3.1. Kontroler PID	11
2.3.2. Sistem Kontrol Cerdas	13
2.3.2.1. <i>Fuzzy Logic Controller</i>	14
BAB 3 PARAMETER DAN PEMODELAN MOTOR BLDC.....	17
3.1. Identifikasi Parameter Elektrik.....	17
3.1.1. Pengukuran Resistansi dan Induktansi Motor	17
3.2. Identifikasi Parameter Mekanik	18
3.2.1. Konstanta BEMF	18
3.2.2. Konstanta Torsi.....	19
3.2.3. Konstanta Waktu Mekanis	20
3.2.4. Koefisien Gaya Gesek	21
3.2.5. Momen Inersia	22

3.2.6. Hasil Identifikasi Parameter Motor BLDC dan Pemodelan Sistem	23
3.2.7. Karakteristik Torsi terhadap Kecepatan	23
BAB 4 PERANCANGAN SISTEM KONTROL KECEPATAN MOTOR	
BLDC	25
4.1. <i>Direct Torque Control</i> (DTC)	25
4.2. Metode Estimasi Torsi Elektromagnetik	28
4.3. Kontrol Kecepatan	29
4.3.1. <i>Fuzzy-PID Controller</i>	30
4.3.2. ANFIS (<i>Adaptive Neuro Fuzzy Inference System</i>)	34
4.3.3. Perancangan ANFIS untuk Pengaturan Kecepatan Motor BLDC	36
4.3.4. Metode Pembelajaran Hybrid ANFIS Menggunakan Algoritma PSO dan RLSE.....	37
4.3.4.1. Metode Pembelajaran Menggunakan Algoritma PSO	37
4.3.4.2. <i>Recursive Least Square Estimation</i> (RLSE)	40
4.3.5. Simulasi Sistem	41
4.3.6. Perancangan <i>Hardware</i>	41
4.3.6.1. Sensor Arus	44
4.3.6.2. Sensor Tegangan BEMF	45
4.3.7. Perancangan <i>Firmware</i>	46
4.3.7.1. Proses Inisialisasi	47
4.3.7.2. Proses Rutin <i>Interrupt</i>	48
BAB 5 PENGUJIAN DAN ANALISIS	51
5.1. Simulasi Sistem <i>Open Loop</i> Motor BLDC Berbasis DTC	51
5.2. Simulasi Pengaturan Kecepatan Motor BLDC dengan Metode Fuzzy-PID	53
5.3. Pembelajaran ANFIS dengan Supervisi Fuzzy-PID.....	54
5.4. Simulasi Pengaturan Kecepatan Motor BLDC dengan Metode ANFIS	57

5.4.1. Simulasi dengan Kecepatan Referensi Tetap dan Tanpa Beban	58
5.4.2. Simulasi dengan Kecepatan Referensi Berubah dan Tanpa Beban	60
5.4.3. Simulasi dengan Kecepatan Referensi Tetap dan Beban Berubah	61
5.5. Pengujian Sensor Arus	64
5.6. Pengujian Algoritma ANFIS pada Mikrokontroler	64
5.7. Pengujian Pengaturan Kecepatan Motor BLDC dengan ANFIS	65
5.7.1. Pengujian dengan Kecepatan Referensi Tetap dan Tanpa Beban	65
BAB 6 KESIMPULAN DAN SARAN	69
DAFTAR PUSTAKA	71

Halaman ini sengaja dikosongkan

DAFTAR GAMBAR

Gambar 1.1. Kendaraan listrik di ITS.....	2
Gambar 2.1. Rangkaian ekuivalen motor BLDC	5
Gambar 2.2. Rangkaian driver 3 fasa motor BLDC	6
Gambar 2.3. Diagram bentuk gelombang BEMF motor BLDC dan arus pada masing-masing fasa berdasarkan posisi rotor	7
Gambar 2.4. Rangkaian ekuivalen motor DC	8
Gambar 2.5. Diagram sistem kontrol PID.....	12
Gambar 2.6. Diagram sistem kontrol cerdas	13
Gambar 2.7. Diagram standar dari sistem <i>fuzzy logic controller</i>	14
Gambar 2.8. Diagram tipe dari <i>fuzzy logic controller</i> berdasarkan sinyal inputnya	15
Gambar 2.9. Diagram model tipe <i>fuzzy logic controller</i>	15
Gambar 3.1. Sistem <i>drive train</i> sepeda motor listrik.....	18
Gambar 3.2. Respon kecepatan dengan komutasi <i>six-step</i>	21
Gambar 3.3. Respon kecepatan model sistem dengan sistem real	23
Gambar 3.4. Grafik torsi terhadap kecepatan.....	24
Gambar 4.1. Diagram blok kontrol kecepatan berbasis DTC	26
Gambar 4.2. Diagram vektor tegangan untuk dua skalar <i>on</i> pada bidang $\alpha\text{-}\beta$	28
Gambar 4.3. Struktur kontroler Fuzzy-PID.....	30
Gambar 4.4. Fungsi keanggotaan dari input Fuzzy-PID.....	31
Gambar 4.5. Fungsi keanggotaan dari output Fuzzy-PID.....	33
Gambar 4.6. Struktur ANFIS	34
Gambar 4.7. Struktur ANFIS dengan multi output	36
Gambar 4.8. Simulasi sistem kontrol kecepatan motor BLDC menggunakan ANFIS	41
Gambar 4.9. Simulasi model sistem kontrol BLDC berbasis DTC.....	41
Gambar 4.10. Desain layout PCB untuk modul mikrokontroler	42
Gambar 4.11. Desain layout PCB untuk modul driver	42
Gambar 4.12. Skematik diagram keseluruhan sistem pengaturan kecepatan motor BLDC	43
Gambar 4.13. Sensor arus Tamura L01ZXXXS05.....	44

Gambar 4.14. Sensor tegangan BEMF	45
Gambar 4.15. Simulasi pengaturan parameter <i>clock</i> pada STM32F407	47
Gambar 4.16. Simulasi pengaturan inialisasi pin I/O pada STM32F407	47
Gambar 4.17. <i>Flowchart</i> program rutin <i>interrupt timer 7</i>	49
Gambar 4.18. Perhitungan lebar pulsa <i>hall effect sensor</i> oleh <i>timer 2</i>	50
Gambar 5.1. Respon kecepatan motor BLDC sistem <i>open loop</i>	51
Gambar 5.2. Arus <i>start fasa-a</i> sistem <i>open loop</i>	52
Gambar 5.3. Arus <i>start fasa-a</i> dengan pembatas arus	53
Gambar 5.4. Respon kecepatan motor BLDC menggunakan kontrol Fuzzy-PID	53
Gambar 5.5. Fungsi keanggotaan ANFIS sebelum proses training	54
Gambar 5.6. Fungsi kenggotaan ANFIS setelah proses training.....	55
Gambar 5.7. Data training K_p K_i K_d	55
Gambar 5.8. Grafik konvergensi RMSE	56
Gambar 5.9. Respon kecepatan menggunakan ANFIS	57
Gambar 5.10. Respon kecepatan dengan referensi tetap dan tanpa beban	58
Gambar 5.11. Arus pada fasa- <i>a</i> (I_a) untuk kecepatan referensi tetap (1000 rpm) dan tanpa beban	58
Gambar 5.12. Torsi untuk kecepatan referensi tetap (1000 rpm) dan tanpa beban	59
Gambar 5.13. Perbandingan nilai torsi estimasi dengan torsi yang sebenarnya ..	59
Gambar 5.14. Respon kecepatan dengan kecepatan referensi berubah	60
Gambar 5.15. Arus I_a saat terjadi perubahan kecepatan	60
Gambar 5.16. Torsi elektromagnetik saat terjadi perubahan kecepatan	61
Gambar 5.17. Perubahan beban terhadap waktu	62
Gambar 5.18. Respon kecepatan terhadap perubahan beban	62
Gambar 5.19. Arus I_a pada kondisi beban berubah	63
Gambar 5.20. Torsi elektromagnetik pada kondisi beban berubah	63
Gambar 5.21. Respon kecepatan pada kecepatan referensi 500 rpm	65
Gambar 5.22. Arus I_a saat <i>steady state</i>	66
Gambar 5.23. Tegangan BEMF saat <i>steady state</i>	66

DAFTAR TABEL

Tabel 3.1. Tabel pengukuran R dan L motor BLDC	17
Tabel 3.2. Tabel pengukuran untuk konstanta BEMF	19
Tabel 3.3. Tabel pengukuran untuk konstanta gaya gesek	22
Tabel 3.4. Tabel parameter motor dan pemodelan sistem	23
Tabel 3.5. Parameter rating motor BLDC	24
Tabel 4.1. Penentuan sektor pada bidang α - β berdasarkan kombinasi sinyal dari <i>hall effect sensor</i>	27
Tabel 4.2. Pemilihan vektor tegangan berdasarkan sektor dan nilai dari sinyal kontrol	27
Tabel 4.3. Aturan inferensi fuzzy untuk Fuzzy-P	32
Tabel 4.4. Aturan inferensi fuzzy untuk Fuzzy-I	32
Tabel 4.5. Aturan inferensi fuzzy untuk Fuzzy-D	32
Tabel 4.6. Kalibrasi sensor arus L01Z400S05	44
Tabel 5.1. Hasil proses training ANFIS	56
Tabel 5.2. Performa ANFIS terhadap respon kecepatan berdasarkan error <i>steady state</i>	57
Tabel 5.3. Hasil pengujian sensor arus	64
Tabel 5.4. Perbandingan performa sistem kontrol Fuzzy-PID dan ANFIS	64

Halaman ini sengaja dikosongkan

DAFTAR SIMBOL DAN SINGKATAN

SIMBOL

V_s	tegangan sumber
V_a	tegangan terminal pada fasa a
V_b	tegangan terminal pada fasa b
V_c	tegangan terminal pada fasa c
R	hambatan pada stator motor DC
R_a	hambatan pada stator fasa a
R_b	hambatan pada stator fasa b
R_c	hambatan pada stator fasa c
R_{LL}	hambatan <i>line-to-line</i>
L	induktansi pada stator motor DC
L_a	induktansi pada stator fasa a
L_b	induktansi pada stator fasa b
L_c	induktansi pada stator fasa c
i	arus stator pada motor DC
i_a	arus pada fasa a
i_b	arus pada fasa b
i_c	arus pada fasa c
e	tegangan BEMF pada motor DC
e_a	tegangan BEMF pada fasa a
e_b	tegangan BEMF pada fasa b
e_c	tegangan BEMF pada fasa c
e_{LL}	tegangan BEMF <i>line-to-line</i>
v_{ab}	tegangan antara fasa a dan b
v_{bc}	tegangan antara fasa b dan c
v_{ca}	tegangan antara fasa c dan a
k_T	konstanta torsi
k_e	konstanta BEMF
τ_m	konstanta waktu mekanis

β	konstanta gesekan pada rotor
J	momen inersia rotor
ω_m	kecepatan rotor
p	jumlah kutub
θ_e	posisi rotor elektrik
θ_m	posisi rotor mekanis
P	daya
T_e	torsi <i>electromotive</i>
T_{em}	total torsi <i>electromotive</i>
T_L	torsi beban
K_P	konstanta proporsional
K_I	konstanta integral
K_D	konstanta diferensial
T_I	konstanta waktu integral
T_D	konstanta waktu diferensial
$e(t)$	<i>error</i> pada waktu ke- t
$e(k)$	<i>error</i> ke- k
c_T	sinyal kontrol
Ψ	fluks
u_{PID}	output dari PID
F_{K_p}	output dari Fuzzy-P
F_{K_i}	output dari Fuzzy-I
F_{K_d}	output dari Fuzzy-D
$\mu_{A_i}(x)$	fungsi keanggotaan dengan inpu x
O_i^1	output layer ke-1
O_i^2	output layer ke-2
O_i^3	output layer ke-3
O_i^4	output layer ke-4
O_i^5	output layer ke-5
x_i^n	posisi ke- n dari partikel ke- i

w_i^n	kecepatan ke- n dari partikel ke- i
c_1, c_2	koefisien akselerasi
y_d	nilai output yang diinginkan
y_{out}	output ANFIS
p_{best_i}	partikel terbaik dari lokasi tertentu
g_{best}	partikel yang paling baik secara global
\bar{x}_{pbest_i}	posisi terbaik pada lokasi tertentu
\bar{x}_{gbest}	posisi terbaik secara global
$\varphi^T(t)$	vektor regresi
$\hat{\theta}(t)$	vektor parameter estimasi
λ	<i>forgetting factor</i>

SINGKATAN

<i>ANFIS</i>	<i>Adaptive Neuro-Fuzzy Inference System</i>
<i>BEMF</i>	<i>Back Electro Motive Force</i>
<i>BLAC</i>	<i>Brushless Alternating Current</i>
<i>BLDC</i>	<i>Brushless Direct Current</i>
<i>DC</i>	<i>Direct Current</i>
<i>DTC</i>	<i>Direct Torque Control</i>
<i>EV</i>	<i>Electric Vehicles</i>
<i>FLC</i>	<i>Fuzzy Logic Controller</i>
<i>FNN</i>	<i>Fuzzy Neural Network</i>
<i>GA</i>	<i>Genetic Algorithm</i>
<i>HEV</i>	<i>Hybrid Electric Vehicles</i>
<i>ICE</i>	<i>Internal Combustion Engine</i>
<i>NN</i>	<i>Neural Network</i>
<i>PID</i>	<i>Proportional-Integral-Derivative</i>
<i>PMSM</i>	<i>Permanent Magnet Synchronous Motor</i>
<i>PWM</i>	<i>Pulse Width Modulation</i>
<i>WSC</i>	<i>World Solar Challenge</i>

Halaman ini sengaja dikosongkan

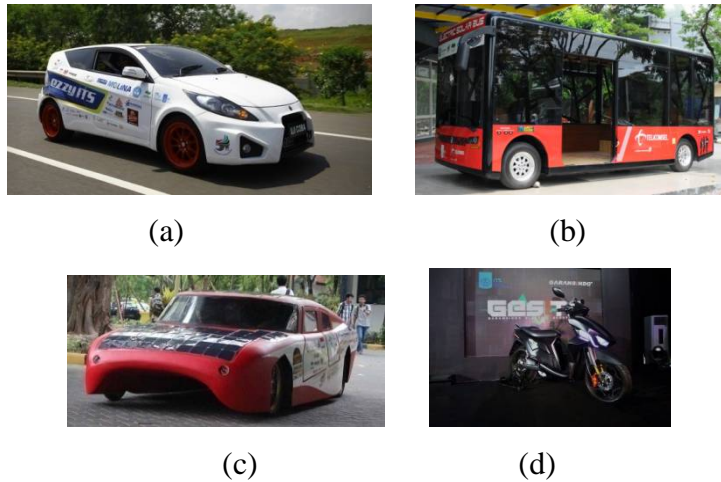
BAB I

PENDAHULUAN

1.1. Latar Belakang

Seiring dengan kesadaran masyarakat terkait dengan perlindungan lingkungan, maka penggunaan kendaraan listrik (*Electric Vehicles (EV)*) maupun kendaraan *hybrid (Hybrid Electric Vehicles (HEV))* mulai diperhatikan [1] sebagai dorongan untuk meminimalkan masalah polusi khususnya di daerah perkotaan [2]. Kendaraan listrik menawarkan solusi yang paling menjanjikan untuk mengurangi emisi kendaraan, yang merupakan masalah utama dari mesin pembakaran internal (*Internal Combustion Engine (ICE)*) pada kendaraan bermotor [3]. Saat ini di Indonesia tengah dikembangkan berbagai jenis kendaraan listrik, salah satunya di kampus Institut Teknologi Sepuluh Nopember (ITS). Sejak tahun 1985, ITS mulai mengembangkan mobil listrik yang diberi nama Widya Wahana I. Tahun 2012, ITS dan beberapa perguruan tinggi di Indonesia ditunjuk oleh pemerintah untuk mengembangkan mobil listrik nasional. Kendaraan listrik yang telah ITS wujudkan diantaranya adalah mobil Ezzy ITS, ITS *Electric Solar Bus*, dan mobil sapu angin surya (ITS *solar car*), yang sudah 2 (dua) kali mengikuti kejuaraan *World Solar Challenge (WSC)* di Australia (2013 dan 2015). Saat ini, ITS juga sedang mengembangkan sepeda motor listrik GESITS (*Garansindo Electric Scooter ITS*). Gambar 1.1. menunjukkan berbagai kendaraan listrik yang sedang dikembangkan di ITS.

Motor listrik merupakan sistem penggerak utama pada kendaraan listrik. Motor listrik yang digunakan pada kendaraan listrik pada umumnya menggunakan jenis *Permanent Magnet Brushless Direct Current (BLDC) motor*. Motor BLDC adalah motor yang sangat tepat digunakan untuk aplikasi kendaraan listrik karena mempunyai keandalan dan efisiensi yang tinggi, kinerja yang bagus dan torsi yang besar [4],[5]. Motor BLDC merupakan jenis motor yang mempunyai efisiensi paling tinggi dibandingkan dengan jenis motor yang lain [6]. Motor BLDC terdiri dari 2 (dua) komponen utama, yaitu magnet permanen sebagai rotor dan kumparan pada stator. Pada prinsipnya, motor BLDC sama halnya dengan motor DC biasa, namun perbedaan utamanya adalah motor BLDC dioperasikan



Gambar 1.1. Kendaraan listrik di ITS (a) Mobil Ezzy ITS, (b) ITS Electric Solar Bus, (c) ITS solar car, (d) Sepeda motor GESITS

tanpa sikat (*brush*), sehingga diperlukan komutasi secara elektrik untuk mengubah arus fase motor agar bisa menghasilkan torsi yang diinginkan [5]. Proses komutasi pada motor BLDC diperlukan proses pendeteksian posisi rotor agar dapat menentukan kumparan mana yang harus disuplai energi dengan tepat. Proses pendeteksian posisi rotor ini bisa menggunakan sensor (*sensor hall* atau *hall effect sensor*) maupun tanpa sensor (*sensorless*) [7]. Motor BLDC yang digunakan pada kendaraan listrik di ITS adalah motor BLDC dengan daya output 5 kW untuk sepeda motor listrik GESITS dan 30 kW untuk mobil listrik yang semuanya dilengkapi dengan sensor *hall* (sensor posisi).

Suatu sistem kontrol diperlukan agar motor BLDC bisa bekerja dengan baik dan menghasilkan torsi atau kecepatan yang diinginkan. Sistem kontrol PID (*Proportional, Integral, Derivative*) sangat banyak digunakan, dan lebih dari 90% kontroler yang digunakan baik untuk peralatan biasa maupun di dunia industri masih menggunakan kontroler PID [8]. Hal ini disebabkan karena kontroler PID memiliki struktur yang sederhana, mampu bekerja dengan baik, dan mudah diimplementasikan [9]. Meskipun telah banyak digunakan selama bertahun-tahun, untuk proses dengan parameter-parameter yang tidak pasti, kontroler PID sulit untuk didesain [10], karena pada prinsipnya kontroler PID hanya mampu menangani masalah-masalah pada sistem yang linear [11]. Hal ini mendorong

untuk melakukan strategi tuning PID dengan cara yang modern dan bisa diimplementasikan ke dalam sebuah *hardware* [12].

Pada sistem yang mengandung parameter-parameter yang tidak pasti dan tidak linear, strategi kontrol yang didesain harus mampu mengatasi masalah-masalah yang ditimbulkan akibat parameter-parameter tersebut. *Fuzzy logic controller (FLC)* adalah salah satu metode kontrol yang telah terbukti mampu untuk menangani parameter-parameter yang tidak pasti dan tidak linear [12],[13]. Beberapa riset tentang perbandingan performa sistem kontrol pada motor BLDC telah dilakukan [12]-[16]. Berdasarkan riset yang telah dilakukan, *Fuzzy-PID controller* mempunyai respon dinamik yang lebih baik pada motor BLDC dibandingkan dengan kontrol PID biasa.

Seperti yang telah dilakukan pada penelitian sebelumnya, *parallel fuzzy PID controller* telah dibuat dan mampu memberikan respon dinamis yang baik pada sistem motor BLDC [12]. Meskipun menghasilkan respon dinamis yang baik, FLC masih mempunyai kekurangan, yaitu sistemnya yang masih belum adaptif, sehingga kesulitan akan muncul dalam pembentukan dan penyesuaian aturan-aturan fuzzy dari data masukan atau keluaran [17]. Dengan menambahkan kemampuan adaptif (*learning mechanism*), sistem kontrol akan mampu mempelajari aturan dari mekanisme pengendalian secara efisien. Dengan menggabungkan FLC dengan struktur *neural network (NN)* akan menghasilkan sistem yang adaptif yaitu mempunyai kemampuan pembelajaran dengan sistem yang kokoh.

Oleh sebab itu, berbeda dengan penelitian sebelumnya, pada penelitian ini dilakukan riset tentang desain dan implementasi sistem kontrol kecepatan motor BLDC menggunakan ANFIS untuk aplikasi sepeda motor listrik. Pada penelitian ini didapatkan desain ANFIS-multi output untuk kontrol kecepatan motor BLDC berbasis *Direct Torque Control (DTC)* dengan proses training ANFIS menggunakan metode *hybrid Particle Swarm Optimization (PSO)* dan *Recursive Least Square Estimation (RLSE)*. Sistem kontrol yang didesain telah disimulasikan dan diimplementasikan pada sebuah mikrokontroler 32-bit, STM32F407VGT.

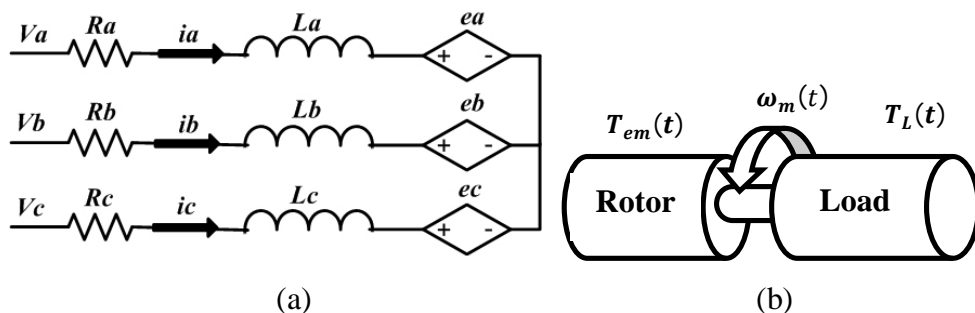
1.2. Sistematika Penulisan

Adapun susunan penulisan penelitian tesis ini adalah sebagai berikut. Bab 1 menjelaskan tentang pendahuluan dan latar belakang masalah. Bab 2 menjelaskan tentang teori dasar motor BLDC, skema kontrol motor BLDC, dan kontrol kecepatan motor BLDC. Identifikasi parameter motor BLDC dijelaskan pada Bab 3, sedangkan perancangan sistem kontrol kecepatan motor BLDC dijelaskan pada Bab 4. Bab 5 menjelaskan tentang hasil dan analisis. Bab 6 berisi tentang kesimpulan dan saran.

BAB 2 KAJIAN PUSTAKA

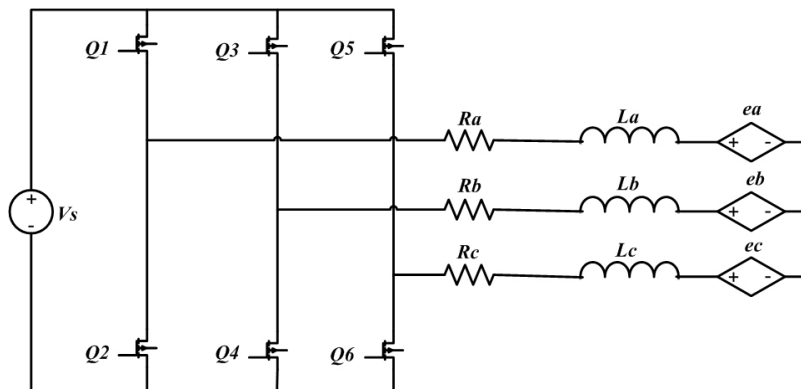
2.1. Motor BLDC

Karena mempunyai kinerja yang bagus dan torsi yang besar serta keandalan dan efisiensi yang tinggi [4],[5], motor BLDC sangat banyak digunakan dalam berbagai macam aplikasi, seperti pada kendaraan listrik. Berdasarkan dari sebuah studi, disebutkan bahwa motor BLDC merupakan jenis motor yang mempunyai efisiensi paling tinggi dibandingkan dengan jenis motor yang lain [6]. Pada prinsipnya, motor BLDC sama halnya dengan motor DC biasa, namun perbedaan utamanya adalah motor BLDC dioperasikan tanpa sikat, sehingga tidak ada gesekan yang ditimbulkan antara sikat dengan rotor. Oleh sebab itu motor BLDC mempunyai nilai efisiensi yang sangat tinggi. Selain itu, penggunaan sikat pada motor DC biasa akan menyebabkan gesekan dengan rotor, sehingga selain menyebabkan berkurangnya nilai efisiensi motor, lama kelamaan sikat akan mengalami aus, dan inilah yang menyebabkan biaya perawatan pada motor DC sangat mahal. Berbeda dengan motor BLDC yang biaya perawatannya murah karena tidak mempunyai sikat. Sehingga dikatakan dalam sebuah studi bahwa motor BLDC merupakan motor yang paling cocok digunakan untuk aplikasi kendaraan listrik [4],[5].



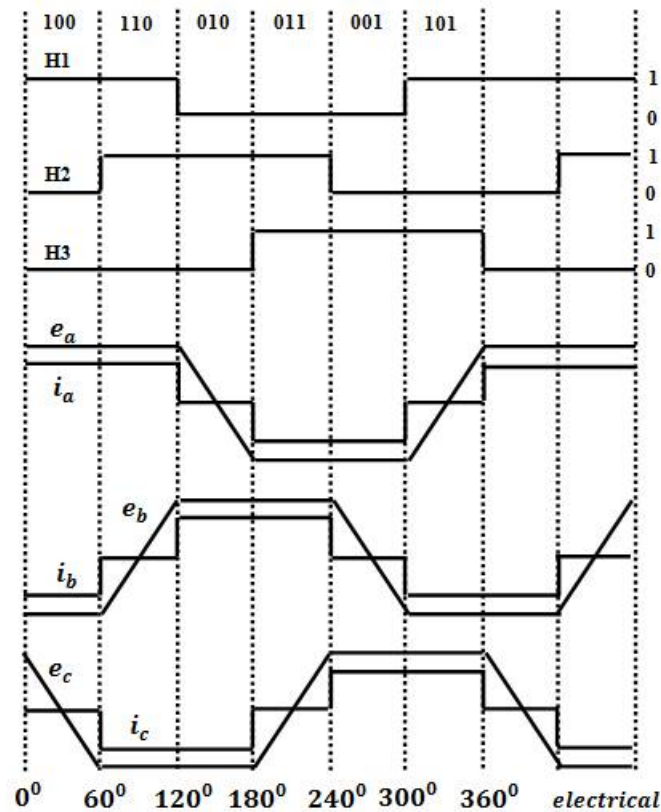
Gambar 2.1. Rangkaian ekuivalen motor BLDC (a) model elektrik tiga fasa pada sisi stator (b) model mekanik pada sisi rotor

Seperti yang ditunjukkan oleh gambar 2.1. motor BLDC terdiri dari 2 (dua) komponen utama, yaitu rotor yang berupa magnet permanen dan stator yang berupa kumparan yang terhubung pada sebuah kontroler. Kontroler ini dihubungkan ke suatu rangkaian driver agar motor BLDC dapat disuplai menggunakan level tegangan yang diperlukan. Rangkaian driver motor BLDC ditunjukkan oleh gambar 2.2. Kontroler tersebut berfungsi untuk menggantikan fungsi dari sebuah komutator, yaitu mensuplai energi kumparan pada stator sesuai dengan urutan step komutasinya. Untuk mengetahui urutan step komutasi, maka diperlukan proses pendeteksian posisi rotor oleh kontroler. Proses pendeteksian posisi rotor ini bisa menggunakan sensor *hall* maupun tanpa sensor (*sensorless*) [7]. Posisi sensor ini biasanya terpisah sejauh 60° atau 120° siklus elektris.



Gambar 2.2. Rangkaian driver 3 fasa motor BLDC

Motor BLDC merupakan jenis *Permanent Magnet Synchronous Motor* (PMSM) atau motor sinkron. Motor sinkron, secara umum, dapat diklasifikasikan menjadi 2 (dua) kategori berdasarkan bentuk gelombang *Back Electro-Motive Force* (BEMF). Satu berbentuk sinusoidal, yang dikenal dengan *Brushless Alternating Current (BLAC) motor* dan yang satu berbentuk trapezoidal, yaitu *Brushless Direct Current (BLDC) motor*. Selain menggunakan sensor *hall*, pendeteksian posisi rotor motor BLDC dapat menggunakan deteksi tegangan BEMF-nya [7]. Bentuk gelombang BEMF motor BLDC dapat digambarkan berupa fungsi sepotong-sepotong (*piece-wise function*) berbentuk gelombang



Gambar 2.3. Diagram bentuk gelombang BEMF motor BLDC dan arus pada masing-masing fasa berdasarkan posisi rotor [18]

trapezoidal dimana setiap nilai tegangan BEMF setiap fasanya ditentukan berdasarkan posisi rotor. Gambar 2.3. Menunjukkan bentuk gelombang BEMF motor BLDC berdasarkan posisi rotor [18].

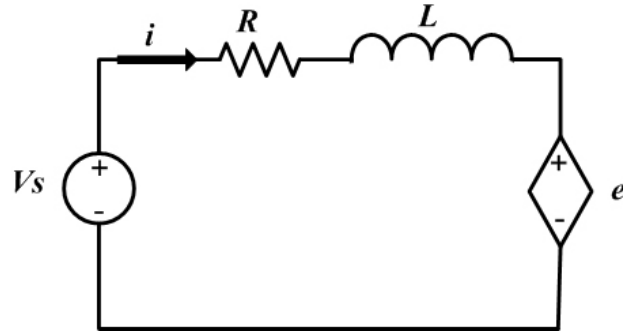
2.1.1. Model Matematika Motor BLDC

Rangkaian ekuivalen motor DC digambarkan pada gambar 2.4. R dan L masing-masing adalah nilai resistansi dan induktansi dari kumparan, sedangkan e adalah tegangan BEMF. Persamaan motor DC dapat dituliskan sebagai berikut:

$$V_s = Ri + L \frac{di}{dt} + e \quad (1)$$

$$T_e = \beta \omega_m + J \frac{d\omega_m}{dt} + T_L \quad (2)$$

V_s merupakan sumber tegangan DC dan i adalah arus yang mengalir pada kumparan (*armature current*). T_e dan T_L masing-masing adalah torsi listrik dan beban mekanis, sedangkan β , J , ω_m adalah konstanta gesekan, inersia motor, dan kecepatan angular rotor.



Gambar 2.4. Rangkaian ekuivalen motor DC

Pada prinsipnya motor BLDC sama dengan motor DC biasa, selain pada sikat, perbedaan motor BLDC dengan motor DC adalah pada motor BLDC terdapat 3-fasa belitan pada stator, sebagaimana yang ditunjukkan pada gambar 2.1. Sehingga persamaan motor BLDC dapat dituliskan sebagai berikut:

$$v_{ab} = R (i_a - i_b) + L \frac{d(i_a - i_b)}{dt} + (e_a - e_b) \quad (3)$$

$$v_{bc} = R (i_b - i_c) + L \frac{d(i_b - i_c)}{dt} + (e_b - e_c) \quad (4)$$

$$v_{ca} = R (i_c - i_a) + L \frac{d(i_c - i_a)}{dt} + (e_c - e_a) \quad (5)$$

$$T_{em} = J \frac{d\omega_m}{dt} + B \omega_m + T_L \quad (6)$$

v , i dan e masing-masing menunjukkan tegangan antar fasa, arus fasa, dan tegangan BEMF pada masing-masing fasa a , b , dan c . R dan L merupakan nilai resistansi dan induktansi pada belitan masing-masing fasa. T_{em} dan T_L menunjukkan torsi *electromotive* dan torsi beban pada motor. J adalah momen inersia, B adalah konstanta gesekan dan ω_m adalah kecepatan rotor.

T_{em} adalah total torsi *electromotive* yang merupakan penjumlahan dari torsi yang dihasilkan pada masing-masing fasa.

$$T_{em} = \sum_{k=a,b,c} T_{em,k} \quad (7)$$

Torsi yang dihasilkan pada masing-masing fasa dan tegangan BEMF dirumuskan pada persamaan (8) dan (9).

$$T_{em,k} = i_k \cdot k_{T,k}(\theta_e) \quad (8)$$

$$e_k = k_{e,k}(\theta_e) \cdot \omega_m \quad (9)$$

$k_{T,k}(\theta)$ dan $k_{e,k}(\theta)$ adalah konstanta torsi dan konstanta BEMF pada masing-masing fasa yang nilainya berubah berdasarkan fungsi dari posisi rotor. Dengan asumsi bahwa nilai $k_{T,k}(\theta)$ dan $k_{e,k}(\theta)$ adalah sama pada masing-masing fasa, maka persamaan tegangan BEMF dan torsi dapat dituliskan.

$$e_a = k_e f(\theta_e) \cdot \omega_m \quad (10)$$

$$e_b = k_e f\left(\theta_e - \frac{2\pi}{3}\right) \cdot \omega_m \quad (11)$$

$$e_c = k_e f\left(\theta_e + \frac{2\pi}{3}\right) \cdot \omega_m \quad (12)$$

$$T_{em} = k_T \left[i_a f(\theta_e) + i_b f\left(\theta_e - \frac{2\pi}{3}\right) + i_c f\left(\theta_e + \frac{2\pi}{3}\right) \right] \quad (13)$$

θ_e merupakan sudut elektrik rotor, yang besarnya adalah hasil kali antara posisi mekanik rotor dengan jumlah pasang kutub-kutubnya ($\theta_e = \frac{P}{2} \theta_m$). $f(\theta_e)$ adalah fungsi dari bentuk trapezoidal tegangan BEMF.

$$f(\theta_e) = \begin{cases} 1 & 0 < \theta_e < \frac{2\pi}{3} \\ -\frac{6}{\pi} \theta_e + 5 & \frac{2\pi}{3} < \theta_e < \pi \\ -1 & \pi < \theta_e < \frac{5\pi}{3} \\ \frac{6}{\pi} \theta_e - 11 & \frac{5\pi}{3} < \theta_e < 2\pi \end{cases} \quad (14)$$

Agar dapat direpresentasikan dalam bentuk *state space*, maka persamaan (3)-(5) perlu dimodifikasi. Dengan membuang satu persamaan dan mengeliminasi satu variabel dengan persamaan arus.

$$i_a + i_b + i_c = 0$$

maka persamaan tegangan antar fasa menjadi.

$$v_{ab} = R (i_a - i_b) + L \frac{d(i_a - i_b)}{dt} + (e_a - e_b) \quad (15)$$

$$v_{bc} = R (i_a + 2i_b) + L \frac{d(i_a + 2i_b)}{dt} + (e_b - e_c) \quad (16)$$

Dengan demikian, didapatkan model *state space* untuk motor BLDC sebagai berikut.

$$\begin{bmatrix} \dot{i}_a \\ \dot{i}_b \\ \dot{\omega}_m \\ \dot{\theta}_m \end{bmatrix} = \begin{bmatrix} -R/L & 0 & 0 & 0 \\ 0 & -R/L & 0 & 0 \\ 0 & 0 & -B/J & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ \omega_m \\ \theta_m \end{bmatrix} + \begin{bmatrix} 2/3L & 1/3L & 0 \\ -1/3L & 1/3L & 0 \\ 0 & 0 & 1/J \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_{ab} - e_{ab} \\ v_{bc} - e_{bc} \\ T_e - T_L \end{bmatrix}$$

$$\begin{bmatrix} i_a \\ i_b \\ i_c \\ \omega_m \\ \theta_m \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ \omega_m \\ \theta_m \end{bmatrix} \quad (17)$$

2.2. Skema Kontrol Motor BLDC

Agar dapat bekerja dengan baik sesuai dengan yang diinginkan, maka motor BLDC memerlukan sebuah kontroler. Variabel-variabel yang sering dikontrol pada motor BLDC adalah kecepatan, torsi dan posisi rotor. Skema kontrol yang sering digunakan diantaranya, *hysteresis band control*, *Pulse Width Modulation (PWM) control*, dan *variable DC-link control* [18]. *Hysteresis band control* adalah salah satu skema kontrol yang paling sederhana. Untuk mengontrol kecepatan, motor dimatikan ketika kecepatan mencapai nilai tertentu di atas kecepatan referensi dan dinyalakan kembali jika kecepatannya di bawah nilai referensi. Kelemahan dari skema ini adalah frekuensi pensaklaran yang

tinggi dan tidak bisa diatur jika lebar *hysteresis* yang digunakan kecil dan riak (*ripple*) yang tinggi jika lebar *hysteresis* yang digunakan besar [18]. Pada kontrol PWM digunakan frekuensi pensaklaran dengan nilai yang tetap dan siklus kerja (*duty cycle*) yang berubah-ubah sesuai dengan nilai kontrol error yang dikehendaki. Biasanya frekuensi pensaklaran yang digunakan berkisar antara 20-50 kHz. *Variable DC-link control* mempunyai kelebihan dibandingkan dengan 2 skema kontrol yang lain. Penggunaan power yang linear akan lebih murah dibandingkan dengan PWM, tetapi mempunyai kerugian yang tinggi pada tegangan rendah dan arus yang tinggi [18]. Akan tetapi dapat menjadi alternatif pilihan jika rugi-rugi pensaklaran pada control PWM menjadi sangat signifikan untuk aplikasi kecepatan motor yang tinggi.

2.3. Kontrol Kecepatan Motor BLDC

2.3.1. Kontroler PID

Kontroler PID sangat populer digunakan karena memiliki struktur yang sederhana, mampu bekerja dengan baik, dan mudah diimplementasikan, bahkan lebih dari 90% baik pada peralatan biasa maupun di industri masih menggunakan kontroler PID [8]. Sejak ditemukan sistem kontrol PID pada tahun 1910 (*Elmer Sperry's ship autopilot*) dan metode tuning Ziegler-Nichols pada tahun 1942, kepopuleran kontroler PID semakin lama semakin meningkat dengan cepat. Akan tetapi, metode tuning yang digunakan untuk mencari parameter-parameter kontroler PID sangat bervariasi. Hal ini karena memang tidak ada standar yang ditetapkan untuk struktur kontrol PID [8].

Persamaan umum sistem kontrol PID ditunjukkan oleh persamaan (18).

$$u(t) = K_p \left[e(t) + \frac{1}{T_I} \int_0^t e(t) + T_D \frac{de(t)}{dt} \right] \quad (18)$$

K_p adalah gain proporsional, T_I adalah konstanta waktu (*time constant*) integral dan T_D adalah konstanta waktu diferensial. Dalam prakteknya, kontroler PID tidak semuanya mengandung ketiga komponennya, proporsional, integral dan diferensial. Kontroler PID mempunyai banyak variasi bentuk, diantaranya

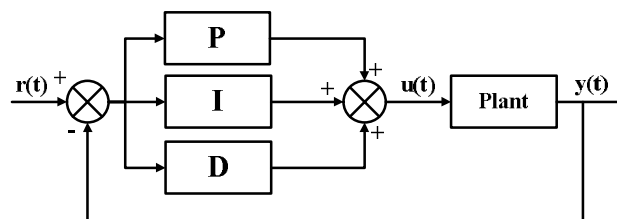
kontroler *proporsional* (P), *proporsional-integral* (PI), *proporsional-diferensial* (PD) dan *proporsional-integral-diferensial* (PID). Diantara bentuk kontroler PID, kontroler PI yang paling banyak digunakan dalam sistem kontrol. Secara umum, model blok diagram kontrol PID ditunjukkan oleh gambar 2.5.

Untuk meningkatkan reliabilitas sistem, pada sistem kontrol modern sering digunakan kontroler PID digital. Dalam hal ini, maka persamaan kontroler PID dalam bentuk kontinyu, sebagaimana yang ditunjukkan pada persamaan (18), tidak dapat langsung digunakan, tetapi harus didiskritkan. Persamaan kontroler PID dalam bentuk diskrit dapat dirumuskan.

$$\begin{aligned}
 u(k) &= K_p \left[e(k) + \frac{T}{T_I} \sum_{j=0}^k e(j) + \frac{T_D}{T} (e(k) - e(k-1)) \right] \\
 &= K_p e(k) + K_I \sum_{j=0}^k e(j) + K_D (e(k) - e(k-1)) \quad (19)
 \end{aligned}$$

K_I adalah koefisien integral, K_D adalah koefisien diferensial, dan T adalah periode sampling.

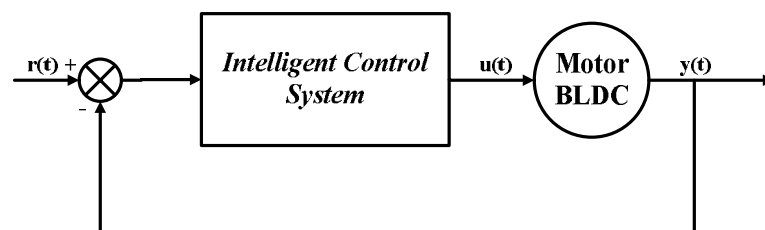
Meskipun mempunyai bentuk yang sederhana dan mudah diaplikasikan, kontroler PID hanya bisa menangani masalah-masalah pada sistem yang linear [11], sedangkan untuk permasalahan pada sistem non-linear dengan parameter-parameter yang tidak pasti, kontroler PID sulit untuk didesain [10]. Seiring dengan berkembangnya teknologi komputasi dan teori sistem kontrol cerdas (*intelligent control system*), terdapat berbagai bentuk tipe kontroler PID, seperti *trapezoid integration PID*, *variable-speed integration PID*, *fuzzy PID*, *neural-network PID*, dan sebagainya. Sehingga, dengan model yang baru ini, kontroler PID mampu menangani permasalahan sistem yang tidak linear dengan parameter-parameter yang tidak pasti.



Gambar 2.5. Diagram sistem kontrol PID

2.3.2. Sistem Kontrol Cerdas

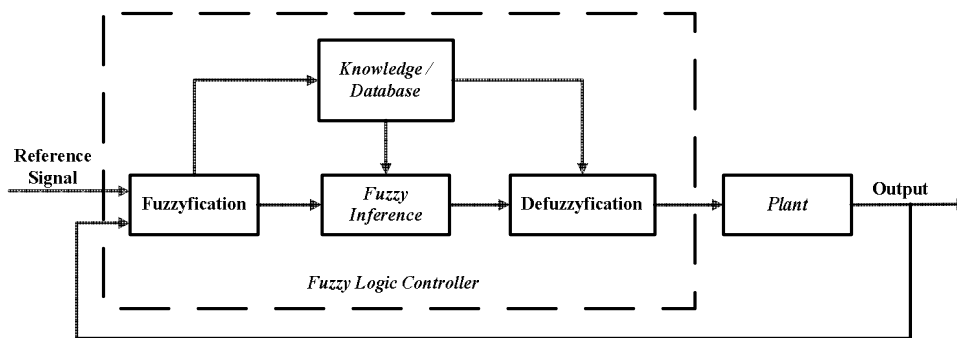
Sistem kontrol cerdas (*intelligent control system*) merupakan sistem kontrol dari penggabungan antara sistem kontrol otomatis dengan konsep kecerdasan buatan. Sistem kontrol cerdas yang sering digunakan diantaranya adalah *fuzzy logic controller (FLC)*, *neural networks (NN)*, *genetic algorithms (GA)*, dan sebagainya. Dengan menggunakan sistem kontrol cerdas, maka model matematis dari *plant* yang dikontrol tidak perlu diketahui secara tepat. FLC didasarkan pada *database* dari pengalaman dan membutuhkan sedikit perhitungan, namun kekurangannya adalah tidak dapat bersifat adaptif, atau tidak mempunyai kapasitas yang cukup jika aturan baru diterapkan. Di sisi lain, NN mempunyai kemampuan untuk menyelesaikan ketidakpastian dari dan gangguan pada sistem, tetapi membutuhkan proses komputasi yang lebih lama dan membutuhkan memori penyimpanan data yang lebih besar. Begitu juga dengan GA, meskipun mampu dalam mengoptimasi parameter-parameter dari kontroler, namun memerlukan proses komputasi yang lama dan penyimpanan data yang besar. Maka, dengan mengkombinasikan dari beberapa sistem kontrol cerdas, seperti *fuzzy neural network (FNN) controller*, akan mempunyai kemampuan pembelajaran dan penalaran yang baik. Lebih dari itu, dengan menggabungkan kontroler GA pada FNN dapat mengoptimasi aturan pada inferensi fuzzy dan struktur dari NN. Biasanya, untuk mendapatkan sistem kontrol yang optimal, sistem kontrol cerdas dikombinasikan dengan sistem kontrol linear dan metode kontrol modern yang lain [19]. Gambar 2.6. menunjukkan model blok diagram sistem kontrol cerdas.



Gambar 2.6. Diagram sistem kontrol cerdas

2.3.2.1. Fuzzy Logic Controller

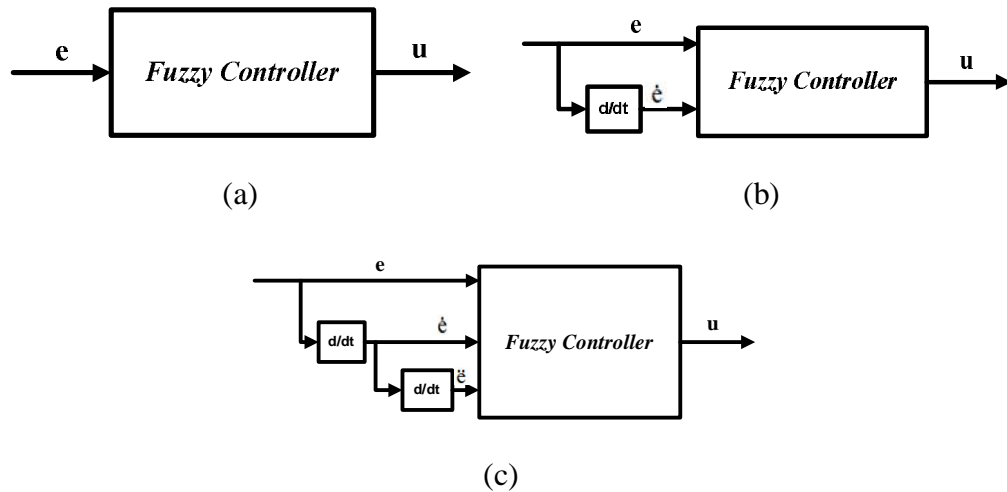
Fuzzy logic controller (FLC) merupakan salah satu dari sistem kontrol cerdas. FLC terdiri dari 4 (empat) komponen, yaitu fuzzifikasi, *database* pengalaman (termasuk *database* dan aturan fuzzy), sistem inferensi fuzzy dan defuzzifikasi. Dengan kemampuan intelijen yang dimiliki oleh *fuzzy logic*, maka model matematis *plant* yang akan dikontrol tidak perlu diketahui secara tepat. Diagram standar dari sistem FLC ditunjukkan oleh gambar 2.7.



Gambar 2.7. Diagram standar dari sistem *fuzzy logic controller* [19]

Karena kemampuannya yang bisa menangani masalah-masalah yang tidak linear serta membutuhkan sedikit perhitungan, FLC adalah pilihan yang tepat digunakan untuk sistem kontrol pada motor. Saat ini, metode kontrol *fuzzy logic* pada motor BLDC terbagi menjadi 3 (tiga) kategori, yaitu *standard fuzzy logic controller*, *fuzzy PID switch controller* dan *optimized fuzzy controller* [19].

Diagram sistem FLC yang secara umum digunakan ditunjukkan pada gambar 2.7. Pada sinyal input mula-mula dilakukan proses fuzzifikasi. Kemudian dilakukan proses pada sistem inferensi fuzzy berdasarkan pengalaman database dan aturan-aturan fuzzy, setelah itu sinyal kontrol dihasilkan melalui proses defuzzifikasi. FLC mempunyai 3 (tiga) tipe berdasarkan jumlah sinyal inputnya, yaitu satu dimensi (*one-dimensional*), dua dimensi (*two-dimensional*), dan tiga dimensi (*three-dimensional*) [19]. Model diagram dari ketiga tipe ini ditunjukkan oleh gambar 2.8. Semakin banyak sinyal input pada FLC, performa sistem kontrol akan semakin baik, namun hal ini akan membutuhkan aturan-aturan fuzzy dan algoritma kontrol yang lebih kompleks. *Two-dimensional* adalah tipe kontrol

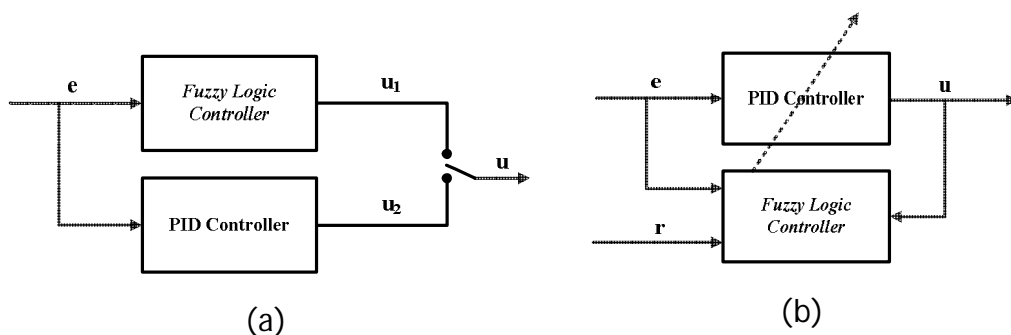


Gambar 2.8. Diagram tipe dari *fuzzy logic controller* berdasarkan sinyal inputnya (a) *one-dimensional* (b) *two-dimensional* (c) *three-dimensional* [19]

fuzzy yang sering digunakan, dimana sinyal input berupa sinyal error (e) dan perubahan error (\dot{e}).

Fuzzy PID switch controller merupakan sistem *hybrid* antara FLC dengan kontroler PID konvensional. Jika output fuzzy bernilai nol, maka kontroler akan beralih menjadi kontroler PID. Kontroler ini terbukti mampu mengurangi *overshoot* dan memperbaiki *settling time* dari sistem.

Optimized fuzzy controller juga merupakan sistem *hybrid* antara FLC dengan kontrol PID konvensional. Pada jenis kontroler ini, *fuzzy logic* digunakan untuk mengatur parameter-parameter dari kontroler PID. Selanjutnya proses kontrol dilakukan sebagaimana proses kontrol PID biasa. Gambar 2.9. menunjukkan blok diagram model tipe FLC.



Gambar 2.9. Diagram model tipe *fuzzy logic controller* (a) *Fuzzy PID switch controller* (b) *Optimized fuzzy controller* [19]

Halaman ini sengaja dikosongkan

BAB 3 PARAMETER DAN PEMODELAN MOTOR BLDC

Parameter motor BLDC didapatkan melalui proses identifikasi dan pengukuran. Selanjutnya, parameter-parameter yang telah didapatkan digunakan untuk memodelkan motor BLDC untuk disimulasikan. Hasil simulasi ini digunakan untuk mengetahui karakteristik sistem secara keseluruhan, yang selanjutnya akan didesain sebuah kontrol kecepatan motor BLDC berbasis ANFIS. Hasil desain kontrol kecepatan diimplementasikan pada sebuah sistem elektronik berbasis mikrokontroler 32 bit. Proses identifikasi ini terdiri dari 2 (dua) tahap, yaitu (1) identifikasi parameter elektrik, meliputi resistansi dan induktansi; dan (2) identifikasi parameter mekanik, meliputi jumlah pasang kutub, konstanta BEMF, konstanta torsi, koefisien gaya gesek, dan momen inersia motor.

3.1. Identifikasi Parameter Elektrik

3.1.1. Pengukuran Resistansi dan Induktansi Motor

Pengukuran resistansi motor dapat dilakukan dengan 2 (dua) cara, yaitu (1) menggunakan alat ukur LCR meter; (2) menggunakan perhitungan arus dan tegangan, berdasarkan persamaan hukum ohm.

$$V = I R \text{ atau } R = \frac{V}{I} \quad (3.1)$$

R adalah nilai resistansi (Ω), sedangkan V dan I berturut-turut adalah tegangan (Volt) dan arus (Ampere). Pada proses identifikasi ini, dilakukan dengan menggunakan cara yang ke-1, yaitu dengan menggunakan LCR meter.

Berdasarkan hasil pengukuran yang telah dilakukan, maka nilai R dan L dari motor BLDC ditunjukkan pada tabel 3.1.

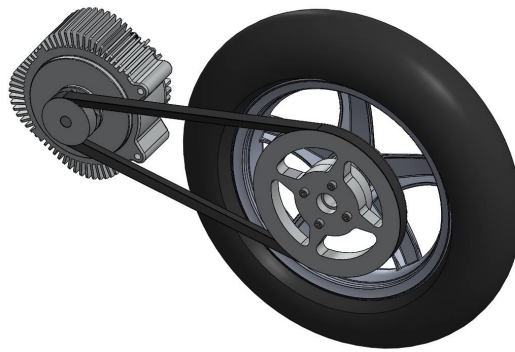
Tabel 3.1. Tabel pengukuran R dan L motor BLDC

$R (\Omega)$		$L (\mu H)$	
R_{ab}	0,08	L_{ab}	225,7
R_{bc}	0,09	L_{bc}	184
R_{ca}	0,09	L_{ca}	221,9
Rata-rata	0,0867	Rata-rata	210,533

Karena yang terhubung pada alat ukur adalah dua buah belitan, maka dengan mengasumsikan bahwa nilai resistansi dan induktansi pada masing-masing belitan adalah sama, maka nilai resistansi dan induktansi untuk setiap belitan adalah setengah dari nilai yang terukur. Sehingga didapatkan $R = 0,04335 \Omega$ dan $L = 105,2665 \mu H$.

3.2. Identifikasi Parameter Mekanik

Identifikasi parameter mekanis ini dilakukan setelah motor dihubungkan pada sebuah sistem *drive train* sepeda motor listrik, dimana motor terhubung dengan sebuah roda menggunakan *fan belt* dengan perbandingan *pulley* rotor dan roda adalah 1:4,18.



Gambar 3.1. Sistem *drive train* sepeda motor listrik

3.2.1. Konstanta BEMF

Konstanta BEMF, dinotasikan sebagai k_e adalah sebuah konstanta yang merupakan perbandingan antara kecepatan putar mekanik rotor terhadap tegangan BEMF yang terbangkit pada stator. Hal ini dapat dituliskan pada persamaan.

$$e = k_e \omega_m \quad (3.2)$$

k_e adalah konstanta BEMF dengan satuan *volt/rpm*. e adalah tegangan BEMF yang terbangkit pada stator akibat putaran rotor, sedangkan ω_m adalah kecepatan putar mekanik rotor (rad/s). nilai e dalam perhitungan ini adalah nilai tegangan puncak dari BEMF. Langkah-langkah identifikasi konstanta BEMF adalah sebagai berikut.

- ✓ Hubungkan motor BLDC pada sebuah kontroler motor dengan sumber tegangan 12 volt.
- ✓ Putar motor BLDC menggunakan algoritma komutasi *six-step*.
- ✓ Tunggu saat putaran motor mencapai kondisi *steady state* (dengan asumsi kondisi steady state tercapai saat $t \geq 3$ detik)
- ✓ Kondisikan inverter tidak mensuplai motor dengan cara memberi logik “0” pada setiap *gate* dari inverter, sehingga motor bergerak dengan sendiri karena efek inersia sampai berhenti.
- ✓ Ukur kecepatan dan tegangan BEMF
- ✓ Hitung konstanta BEMF berdasarkan persamaan 3.2.

Berdasarkan hasil pengukuran yang dilakukan, didapatkan nilai kecepatan dan tegangan BEMF pada tabel 3.2. berikut.

Tabel 3.2. Tabel pengukuran untuk konstanta BEMF

BEMF (line to line) (volt)	Speed (RPM)	k_e (V/krpm)
9,796	507,23	19,313
9,667	503,86	19,186
9,362	482,16	19,417
8,704	460,12	18,917
8,624	453,44	19,019
8,222	440,52	18,664
7,981	424,25	18,812
7,917	419,83	18,858
7,499	405,73	18,483
7,483	400,64	18,678
Rata-rata		18,935

Jadi dapat diketahui bahwa nilai konstanta BEMF dari motor BLDC yang digunakan adalah **18,935** volt/krpm.

3.2.2. Konstanta Torsi

Konstanta torsi, dinotasikan sebagai k_T , adalah sebuah konstanta yang merupakan perbandingan antara torsi terhadap arus pada stator. Hal ini dapat dituliskan pada persamaan.

$$T_e = k_T I \quad (3.3)$$

T_e adalah torsi elektromagnetik sedangkan I adalah arus yang mengalir pada stator. Konstanta ini dapat dicari dengan menurunkannya dari persamaan konstanta BEMF. Hubungan antara daya dan torsi dapat dituliskan dengan persamaan.

$$P = T_e \omega \quad (3.4)$$

P adalah daya pada motor yang merupakan perkalian antara e_{LL} dan I . Sehingga persamaan 3.4 dapat dituliskan menjadi.

$$e_{LL} I = T_e \omega \quad (3.5)$$

Dari persamaan ini, didapatkan bahwa:

$$\frac{e_{LL}}{\omega} = \frac{T_e}{I} \quad (3.6)$$

$\frac{e_{LL}}{\omega}$ merupakan sebuah konstanta BEMF k_e (volt/rads⁻¹) dan $\frac{T_e}{I}$ merupakan konstanta torsi k_T (N.m/A). karena konstanta BEMF yang didapatkan mempunyai satuan volt/krpm, maka untuk mendapatkan konstanta torsi, yaitu dengan cara mengubah nilai konstanta BEMF menjadi volt/rads⁻¹.

$$k_T = \frac{k_e \times 60}{2\pi} \quad (3.7)$$

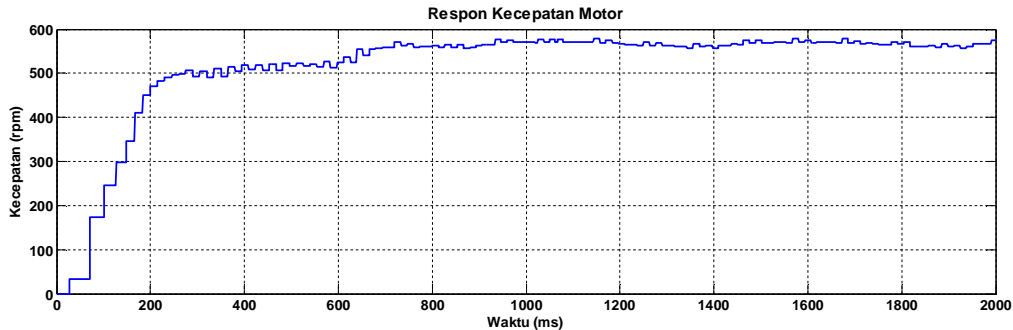
Berdasarkan hasil k_e yang telah didapatkan, maka nilai k_T adalah **0,180815** N.m/A

3.2.3. Konstanta Waktu Mekanis

Konstanta waktu mekanis, atau *mechanical time constant* (τ_m) didefinisikan sebagai waktu dimana motor mencapai kecepatan 63% dari kecepatan *steady state* saat diberi sinyal step. Langkah-langkah mendapatkan konstanta waktu mekanis adalah sebagai berikut:

- ✓ Hubungkan motor BLDC pada sebuah kontroler motor dengan sumber tegangan 12 volt.
- ✓ Putar motor BLDC menggunakan algoritma komutasi *six-step*.
- ✓ Rekam data kecepatan per waktu (dalam hal ini setiap 1 ms)
- ✓ Plot grafik kecepatan
- ✓ Waktu saat kecepatan mencapai 63% dari kecepatan *steady state* merupakan konstanta waktu mekanis.

Dari hasil percobaan yang dilakukan, didapatkan plot grafik respon kecepatan yang ditunjukkan oleh gambar 3.2.



Gambar 3.2. Respon kecepatan dengan komutasi *six-step*

Berdasarkan grafik kecepatan yang didapatkan, dapat diketahui nilai konstanta waktu mekanis τ_m adalah sekitar 150 mili detik atau 0,15 detik.

3.2.4. Koefisien Gaya Gesek

Koefisien gaya gesek dinotasikan sebagai B . Langkah-langkah untuk mencari nilai B adalah sebagai berikut:

- ✓ Hubungkan motor BLDC pada sebuah kontroler motor dengan sumber tegangan 12 volt.
- ✓ Putar motor BLDC menggunakan algoritma komutasi *six-step*.
- ✓ Tunggu saat putaran motor mencapai kondisi *steady state* (dengan asumsi kondisi *steady state* tercapai saat $t \geq 1$ detik)
- ✓ Hitung torsi elektromagnetik motor menggunakan persamaan:

$$T_e = k_T \times I \quad (3.8)$$

- ✓ Berdasarkan persamaan (6), pada saat kondisi *steady state*, maka nilai $\frac{d\omega}{dt} = 0$, sehingga jika $T_L = 0$, maka persamaan mekanik motor menjadi:

$$T_e = B \times \omega_m \quad (3.9)$$

- ✓ Hitung nilai B berdasarkan persamaan 3.9.

Dari percobaan yang telah dilakukan, maka didapatkan hasil yang ditunjukkan pada tabel 3.3.

Tabel 3.3. Tabel pengukuran untuk konstanta gaya gesek

Arus (ampere)	k_T (N.m/A)	Speed (rad/s)	B (N.m.s)
4,399	0,180815	59,418	0,013388
5,854	0,180815	59,110	0,017908
5,835	0,180815	59,168	0,017832
4,446	0,180815	58,769	0,013679
5,005	0,180815	59,899	0,015108
5,166	0,180815	59,057	0,015817
4,982	0,180815	59,562	0,015124
5,401	0,180815	58,791	0,016611
5,532	0,180815	58,959	0,016966
6,231	0,180815	58,835	0,019149
Rata-rata			0,016158

Jadi didapatkan nilai koefisien gaya gesek dari sistem adalah **0,016158** N.m.s.

3.2.5. Momen Inersia

Momen inersia dinotasikan dengan J merupakan sebuah nilai kelembaman motor untuk berotasi pada porosnya. Momen inersia berpengaruh pada saat motor dalam keadaan transien baik itu akselerasi maupun deselerasi. Karena konstanta waktu elektrik jauh lebih kecil dibandingkan dengan konstanta waktu mekanis, maka momen inersia dapat dihitung dengan mengabaikan efek dari induktor menggunakan persamaan.

$$J = \frac{\tau_m(R_{LL}B + k_e k_t)}{R_{LL}} \quad (3.10)$$

R_{LL} merupakan nilai resistansi *line-to-line* atau dua kali nilai R fasa. Sehingga dapat diketahui nilai momen inersia dari sistem adalah **0,059009** kg.m²

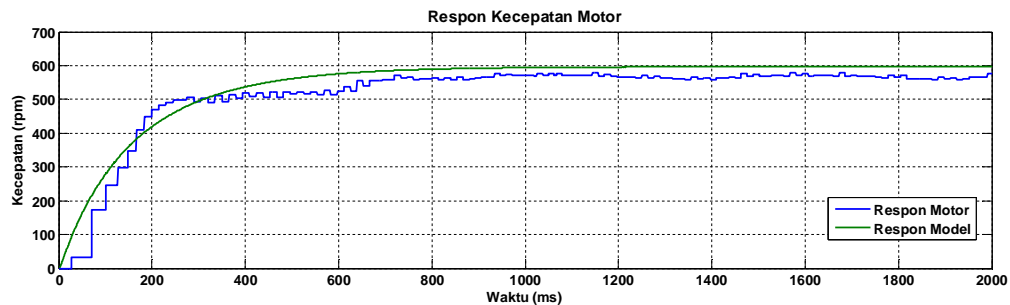
3.2.6. Hasil Identifikasi Parameter Motor BLDC dan Pemodelan Sistem

Berdasarkan data yang telah didapatkan, parameter motor BLDC dan sistem ditunjukkan pada tabel 3.4.

Tabel 3.4. Tabel parameter motor dan pemodelan sistem

Parameter	Nilai	Satuan
R	0,04335	Ω
L	105,2665	μH
k_e	18,935	volt/krpm
k_t	0,180815	N.m/A
τ_m	0,15	s
B	0,016158	N.m.s
J	0,059009	Kg.m^2
Jumlah pasang kutub	4	-

Dari parameter yang telah didapatkan, gambar 3.3 menunjukkan respon kecepatan dari pemodelan sistem dengan sistem yang sebenarnya.



Gambar 3.3. Respon kecepatan model sistem dengan sistem real

3.2.7. Karakteristik Torsi terhadap Kecepatan

Untuk mengetahui karakteristik torsi terhadap kecepatan dilakukan pendekatan berdasarkan parameter yang telah didapatkan. Persamaan arus yang mengalir pada belitan, dapat dituliskan sebagai.

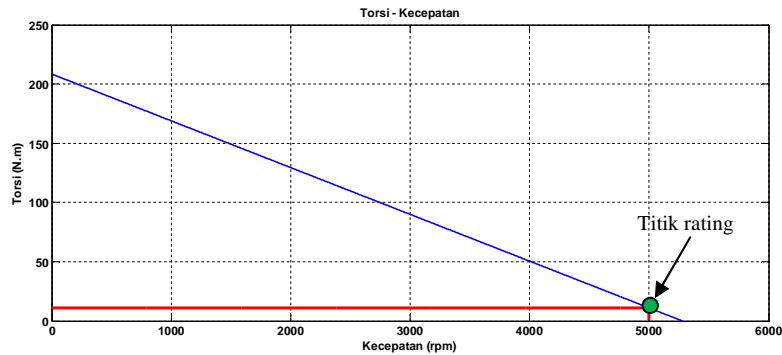
$$I = \frac{V - k_e n_r}{R_{LL}} \quad (3.11)$$

Jika persamaan 3.8 disubstitusikan dengan persamaan 3.11 akan didapatkan.

$$T_e = k_T \left[\frac{V - k_e n_r}{R_{LL}} \right] \quad (3.12)$$

Berdasarkan persamaan 3.12, dengan memasukkan parameter-parameter yang telah didapatkan, dengan nilai tegangan rating sebesar 100 volt, maka akan

didapatkan hubungan sebagai berikut. Jika T_e bernilai 0, maka n_r bernilai 5281 rpm. Jika n_r bernilai 0, maka T_e bernilai 208,55 N.m. Berdasarkan hubungan nilai dari T_e dan n_r , dapat diplot grafik sebagaimana ditunjukkan oleh gambar 3.4.



Gambar 3.4. Grafik torsi terhadap kecepatan

Rating dari motor BLDC didapatkan dan ditunjukkan pada tabel 3.5.

Tabel 3.5. Parameter rating motor BLDC

Parameter	Nilai	Satuan
V	100	volt
I	61,4	A
Daya	5	kW
Torsi	11,1	N.m
Kecepatan	5000	rpm

BAB 4

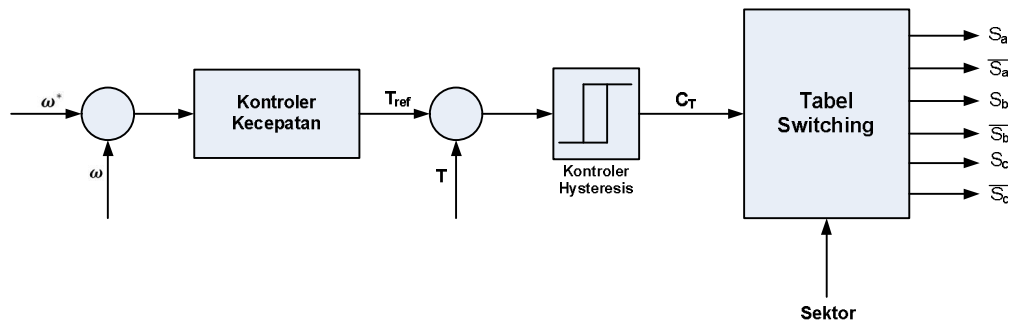
PERANCANGAN SISTEM KONTROL KECEPATAN MOTOR BLDC

Motor BLDC memerlukan sebuah *Variable Speed Drive* (VSD) agar bisa berputar sesuai dengan kecepatan yang berbeda-beda. Secara garis besar, metode kontrol motor BLDC dibagi menjadi 2 (dua) kategori, yaitu *scalar control* dan *vector control*. *Scalar control* merupakan sistem kontrol yang paling sederhana dan mudah diimplementasikan karena sistem ini hanya didasarkan pada hubungan antara variabel yang dikontrol saat kondisi *steady state*. Pada motor induksi sistem kontrol ini sering dikenal sebagai *V/f control*, yang mana dengan menjaga nilai *V/f* tetap konstan untuk mendapatkan kecepatan yang diinginkan. Sedangkan pada motor BLDC, sistem kontrol ini dikenal dengan *variable-DC link voltage control*. Dengan mengubah nilai tegangan input motor BLDC, maka akan didapatkan kecepatan motor BLDC yang diinginkan. Namun, *scalar control* menimbulkan *losses* yang tinggi terutama pada saat operasi pada tegangan rendah dan arus yang tinggi [18]. *Vector control* merupakan sistem kontrol dimana arus stator diidentifikasi dan divisualisasikan menjadi 2 (dua) komponen vektor ortogonal. Satu komponen menunjukkan suatu parameter fluks dan yang lainnya adalah torsi. Ada 2 (dua) macam bentuk *vector control*, *Field Oriented Control* (FOC) dan *Direct Torque Control* (DTC). Prinsip dari FOC adalah mengontrol vektor arus dan tegangan yang berakibat pada terbangkitnya fluks dan torsi elektromagnetik, sedangkan prinsip utama dari DTC adalah menentukan vektor tegangan berdasarkan error antara torsi dan fluks yang diinginkan dengan torsi dan fluks yang sebenarnya. Pada penelitian ini, sistem kontrol BLDC menggunakan DTC, karena DTC mempunyai respon yang cepat terhadap perubahan torsi yang dibutuhkan dan perhitungan matematis yang lebih sederhana dibandingkan dengan kontrol menggunakan FOC, sehingga akan lebih cocok jika diaplikasikan untuk kendaraan listrik dengan beban yang berubah secara dinamis.

4.1. Direct Torque Control (DTC)

Direct Torque Control (DTC) pertama kali diperkenalkan pada tahun 1980-an untuk kontrol torsi dan fluks pada motor induksi [20]. Tahun 1990-an,

teknik ini dikembangkan untuk diaplikasikan pada motor sinkron atau *Permanent Magnet Synchronous Motor* (PMSM). DTC menjadi sangat populer dan dikembangkan karena struktur kontrol yang sederhana dan mudah untuk diimplementasikan [21]. Kontrol kecepatan berbasis DTC ditunjukkan oleh gambar 4.1.



Gambar 4.1. Digram blok kontrol kecepatan berbasis DTC

Ada 3 (tiga) sinyal yang berpengaruh pada proses kontrol menggunakan DTC, yaitu:

1. Torsi elektromagnetik, T_{em}
2. Amplitude fluks bocor stator pada bidang α - β , $|\Psi_s|$
3. Sudut dari vektor resultan fluks bocor, δ

Torsi elektromagnetik digunakan sebagai umpan balik pada sistem, sehingga akan didapatkan nilai error torsi, yang kemudian diteruskan ke suatu kontroler hysteresis, dengan output berupa sinyal kontrol (c_T). Berdasarkan nilai sinyal kontrol tersebut dapat ditentukan pemilihan vektor tegangan yang diperlukan. Magnet permanen pada motor BLDC menghasilkan fluks hampir bernilai konstan, sehingga perhitungan vektor resultan dari fluks pada bidang α - β tidak diperlukan [22]. Sudut δ digunakan dalam penentuan sektor pada bidang α - β . Karena perhitungan vektor resultan dari fluks pada bidang α - β tidak lagi diperlukan, maka untuk menentukan sektor pada bidang α - β dilakukan berdasarkan kombinasi sinyal dari *hall effect sensors* sebagaimana ditunjukkan pada tabel 4.1.

Tabel 4.1. Penentuan sektor pada bidang α - β berdasarkan kombinasi sinyal dari *hall effect sensors*

(H_{abc})	1 1 0	0 1 0	0 1 1	0 0 1	1 0 1	1 0 0
Sektor	I	II	III	IV	V	VI

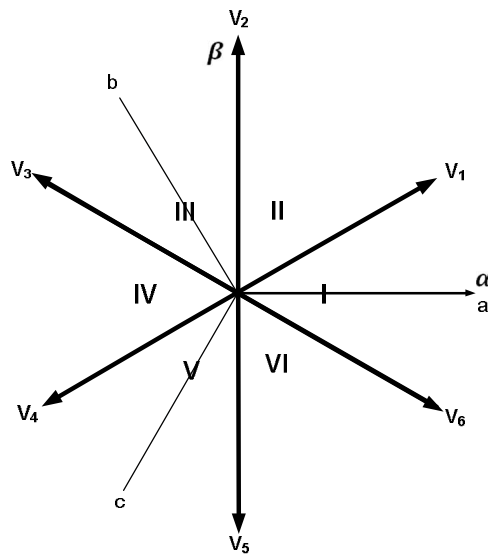
Strategi kontrol menggunakan DTC adalah untuk menentukan apakah torsi elektromagnetik perlu untuk dinaikkan atau diturunkan, kemudian memilih pensaklaran yang tepat untuk inverter. Pensaklaran ini, didasarkan pada nilai sektor dan nilai dari sinyal kontrol c_T . Tabel 4.2 menunjukkan pemilihan vektor tegangan berdasarkan sektor dan nilai c_T .

Tabel 4.2. Pemilihan vektor tegangan berdasarkan sektor dan nilai dari sinyal kontrol

c_T	Sektor					
	I	II	III	IV	V	VI
1	V2 (001001)	V3 (011000)	V4 (010010)	V5 (000110)	V6 (100100)	V1 (100001)
-1	V0 (000010)	V0 (001000)	V0 (100000)	V0 (000010)	V0 (001000)	V0 (100000)

Tanda (+) pada sinyal kontrol menunjukkan bahwa sistem perlu untuk meningkatkan torsi, sedangkan tanda (-) menunjukkan bahwa sistem perlu untuk menurunkan torsi. Ketika diperlukan untuk penurunan torsi, vektor-vektor yang ada pada tabel 4.2 digunakan dalam desain sitem kontrol kecepatan motor BLDC. Misal terjadi pada sektor I, ketika diperlukan penurunan torsi, *switch 6 off*, selanjutnya *switch 5 on*, sehingga arus yang tersimpan dalam belitan, mengalir melalui *switch 5* dengan maksimal, dibandingkan hanya melalui *freewheeling diode* pada *switch 5*.

Pemilihan vektor tegangan untuk menaikkan atau menurunkan torsi ini dapat dilihat pada diagram vektor tegangan pada gambar 4.2.



Gambar 4.2. Diagram vektor tegangan untuk dua saklar on pada bidang α - β

Pada saat posisi rotor berada pada sektor I, maka untuk menaikkan torsi diperlukan vektor tegangan V_2 , sedangkan untuk menurunkan torsi digunakan vektor tegangan V_0 . Untuk menaikkan torsi pada sektor II digunakan vektor tegangan V_3 , sedangkan untuk menurunkan torsi digunakan vektor tegangan V_0 . Pada sektor III, V_4 digunakan untuk menaikkan torsi, sedangkan V_0 digunakan untuk menurunkan torsi. Pada sektor IV, V_5 digunakan untuk menaikkan torsi, sedangkan V_0 digunakan untuk menurunkan torsi. Pada sektor V, V_6 digunakan untuk menaikkan torsi, sedangkan V_0 digunakan untuk menurunkan torsi dan pada sektor VI, V_1 digunakan untuk menaikkan torsi, sedangkan V_0 digunakan untuk menurunkan torsi. Untuk nilai vektor 0, berubah-ubah sesuai dengan posisi sektor, sebagaimana yang ditunjukkan pada tabel 4.2.

4.2. Metode Estimasi Torsi Elektromagnetik

Salah satu yang menjadi tantangan untuk menghasilkan performa yang bagus pada kontrol menggunakan DTC adalah proses perhitungan torsi elektromagnetik. Untuk motor magnet permanen, perhitungan torsi elektromagnetik pada bidang α - β menggunakan persamaan 4.1.

$$T_{em} = \frac{3p}{2} [\Psi_{s\alpha} i_{s\alpha} + \Psi_{s\beta} i_{s\beta}] \quad (4.1)$$

p adalah jumlah kutub, sedangkan $\Psi_{s\alpha}$, $\Psi_{s\beta}$, $i_{s\alpha}$, dan $i_{s\beta}$ masing-masing adalah fluks bocor stator dan arus pada bidang α - β . Persamaan ini secara umum digunakan untuk motor dengan BEMF berbentuk sinusoidal (motor BLAC). Sedangkan untuk motor BLDC, untuk mengestimasi T_{em} digunakan persamaan 4.2 – 4.4.

$$T_{em} = \frac{3p}{2} \left[\frac{d\Psi_{r\alpha}}{d\theta_e} i_{s\alpha} + \frac{d\Psi_{r\beta}}{d\theta_e} i_{s\beta} \right] \quad (4.2)$$

$$\frac{d\Psi_{r\alpha}}{d\theta_e} = \frac{e_\alpha}{\omega_e} \quad (4.3)$$

$$\frac{d\Psi_{r\beta}}{d\theta_e} = \frac{e_\beta}{\omega_e} \quad (4.4)$$

θ_e adalah posisi sudut rotor elektrik, sedangkan ω_e adalah kecepatan angular elektrik motor BLDC. $\Psi_{r\alpha}$, $\Psi_{r\beta}$, e_α , dan e_β masing-masing adalah fluks bocor rotor dan tegangan BEMF pada bidang α - β . Dengan demikian, untuk estimasi torsi elektromagnetik motor BLDC pada bidang α - β dapat dituliskan sebagai.

$$T_{em} = \frac{3}{2} \left[\frac{e_\alpha i_{s\alpha} + e_\beta i_{s\beta}}{\omega_m} \right] \quad (4.5)$$

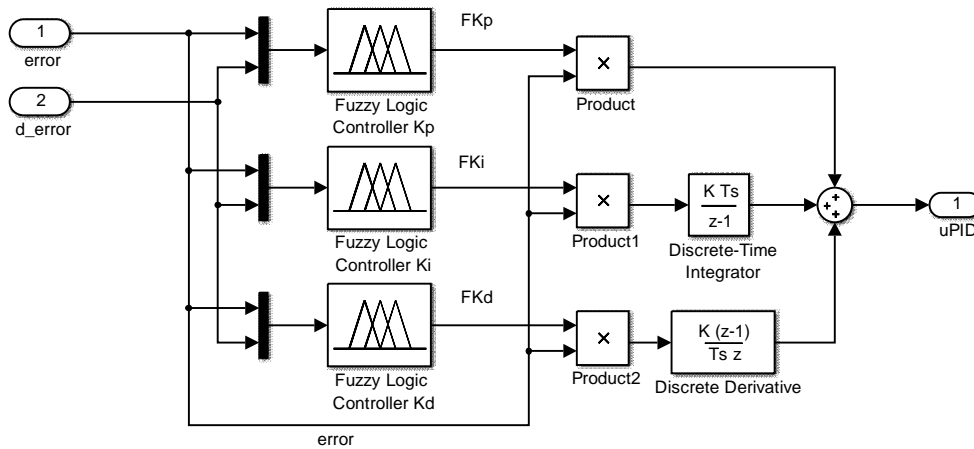
4.3. Kontrol Kecepatan

Kontroler kecepatan yang sering digunakan pada DTC adalah kontroler PID karena strukturnya yang sederhana dan mudah diimplementasikan. Meskipun demikian, kontroler PID mempunyai beberapa kekurangan, yaitu; bisa diaplikasikan hanya untuk sistem linear. Pada sistem yang mengandung parameter-parameter yang tidak pasti dan tidak linear, strategi kontrol yang didesain harus mampu mengatasi masalah-masalah yang ditimbulkan akibat parameter-parameter tersebut. Oleh sebab itu, dengan tidak merubah struktur yang sederhana dari kontroler PID, pada penelitian ini akan didesain sebuah kontroler kecepatan menggunakan ANFIS, yang mana output dari ANFIS ini digunakan untuk menentukan parameter PID (K_p , K_i dan K_d) secara adaptif. Fuzzy-PID *controller* digunakan sebagai supervisor untuk mendapatkan data yang akan dilatihkan pada ANFIS.

4.3.1. Fuzzy-PID Controller

Dengan semakin banyaknya peralatan di industri yang menggunakan kontrol PID, maka banyak teknik dikembangkan untuk menentukan parameter PID. Meskipun mempunyai bentuk yang sederhana, kontroler PID hanya mampu mengatasi masalah-masalah yang linear. Sistem dengan parameter-parameter yang tidak pasti perlu dikontrol dengan suatu kontroler yang mampu menangani masalah-masalah tersebut. Dengan tidak menghilangkan struktur PID, maka kontroler Fuzzy-PID didesain, yang selanjutnya digunakan sebagai supervisor untuk mendapatkan data pembelajaran yang akan ditrainingkan pada ANFIS.

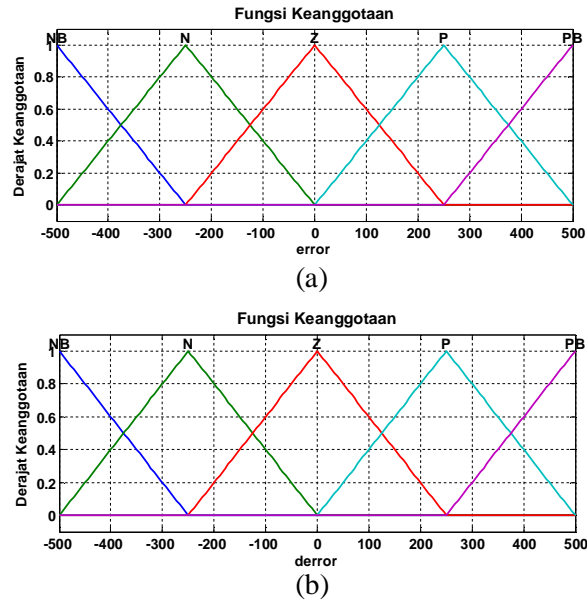
Fuzzy-PID yang didesain pada penelitian ini terdiri dari 3 (tiga) sub-kontroler fuzzy, yaitu Fuzzy-P, Fuzzy-I, dan Fuzzy-D. Input dari Fuzzy-PID berupa nilai *error* dan $\Delta error$, sedangkan output dari masing-masing Fuzzy-PID berupa nilai parameter K_p , K_i , dan K_d yang berubah secara online. Adapun struktur dari Fuzzy-PID pada penelitian ini ditunjukkan pada gambar 4.3.



Gambar 4.3. Struktur kontroler Fuzzy-PID

Setiap input dan output dari Fuzzy-PID terdiri dari 5 (lima) fuzzy set, dengan fungsi keanggotaan berupa fungsi segitiga. Fuzzy set untuk input adalah NB (*Negative Big*), N (*Negative*), Z (*Zero*), P (*Positive*), dan PB (*Positive Big*), sedangkan untuk output, Z (*Zero*), PS (*Positive Small*), P (*Positive*), PM (*Positive Medium*), dan PB (*Positive Big*). Kelima fuzzy set tersebut dapat dinotasikan sebagai F_i^j , $i = 1,2$ dan $j = 1,2, \dots, 5$, sedangkan nilai dari fungsi keanggotaan setiap input dinotasikan sebagai $\mu F_i^j(e, \Delta e)$, $j = 1,2, \dots, 5$. F_{K_p} , F_{K_i} dan F_{K_d} adalah

output dari masing-masing sub-kontroler. Adapun fungsi keanggotaan dari input kontroler Fuzzy-PID ditunjukkan oleh gambar 4.4.



Gambar 4.4. Fungsi keanggotaan dari input Fuzzy-PID (a) $error$
(b) $\Delta error$

Selanjutnya adalah menentukan aturan inferensi fuzzy, *IF-THEN rule*. Jumlah dari aturan fuzzy merupakan perkalian dari jumlah fuzzy set untuk setiap input, sehingga dengan 5 (lima) fuzzy set untuk masing-masing input, maka akan terdapat 25 aturan inferensi fuzzy. Secara umum aturan fuzzy dapat dituliskan sebagai.

$$R^k: \text{ IF } e \text{ is } F_1^j, \text{ AND } \Delta e \text{ is } F_2^l \text{ THEN } f \text{ is } C_k^l \quad (4.6)$$

for $j = 1, \dots, 5; l = 1, \dots, 3; k = 1, \dots, 25$

Pada penelitian ini, kedua puluh lima aturan inferensi fuzzy untuk Fuzzy-P, Fuzzy I, dan Fuzzy-D berturut-turut ditunjukkan pada tabel 4.3, 4.4. dan 4.5. Setiap aturan fuzzy ini dihubungkan dengan operator *min* untuk mendapatkan bobot kuatnya suatu aturan (*rule strength*). Hal ini dapat dituliskan sebagai:

$$\xi_{i,j} = \min(\mu F_i^e, \mu F_j^{\Delta e}) \quad (4.7)$$

i merupakan indeks yang berhubungan dengan variabel fuzzy untuk nilai $error$, sedangkan j adalah indeks yang berhubungan dengan variabel fuzzy untuk nilai $\Delta error$.

Tabel 4.3. Aturan inferensi fuzzy untuk Fuzzy-P

Δe e	NB	N	Z	P	PB
NB	Z	Z	PS	P	PM
N	Z	PS	P	PM	PM
Z	PS	P	PM	PM	PB
P	P	PM	PM	PB	PB
PB	PM	PM	PB	PB	PB

Tabel 4.4. Aturan inferensi fuzzy untuk Fuzzy-I

Δe e	NB	N	Z	P	PB
NB	Z	Z	PS	P	PM
N	Z	PS	P	PM	PB
Z	PB	PB	PM	PB	PB
P	PB	PM	P	PS	Z
PB	PM	P	PS	Z	Z

Tabel 4.5. Aturan inferensi fuzzy untuk Fuzzy-D

Δe e	NB	N	Z	P	PB
NB	PB	PB	PB	PM	PM
N	PB	PB	PM	PM	P
Z	PB	PM	PM	P	PS
P	PM	PM	P	PS	Z
PB	PM	P	PS	Z	Z

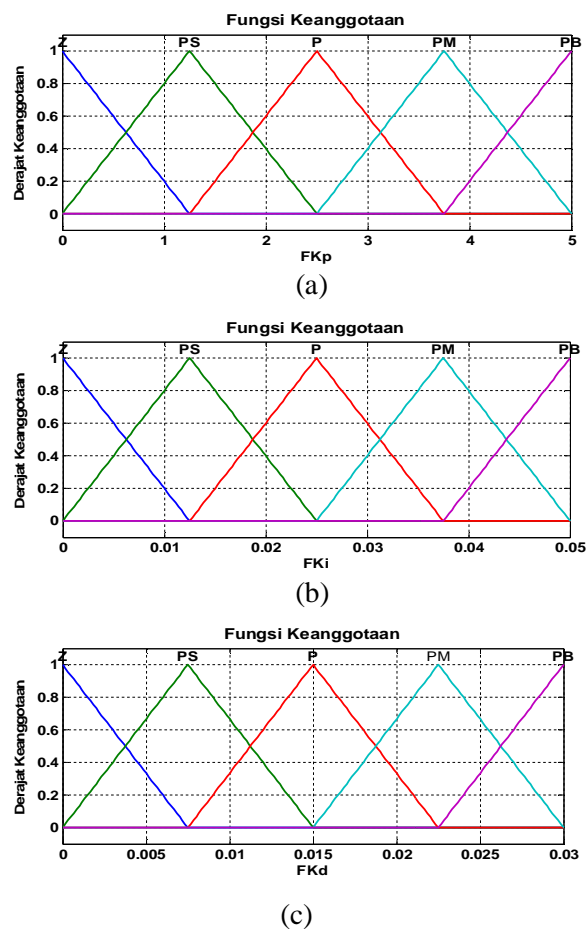
Selanjutnya adalah proses defuzzifikasi, yaitu proses untuk mendapatkan nilai output secara tegas. Pada proses defuzzifikasi digunakan metode *centroid*. Maka, output dari masing-masing sub-kontroler dapat dituliskan sebagai.

$$f(e, \Delta e) = \frac{\sum_{l=1}^{25} \mu_{O_l} \left(\min_{i=1,2} \mu_{F_i^l}(e, \Delta e) \right)}{\sum_{l=1}^{25} \left(\min_{i=1,2} \mu_{F_i^l}(e, \Delta e) \right)} \quad (4.8)$$

Nilai output dari masing-masing sub-kontroler akan menjadi parameter PID, K_p , K_i , dan K_d . Sehingga output dari kontroler PID dapat dituliskan menjadi.

$$u_{PID} = F_{K_p} e(k) + F_{K_i} \sum_{i=1}^k e(i) + F_{K_d} \frac{\Delta e(k)}{\Delta t} \quad (4.9)$$

Fungsi keanggotaan output Fuzzy-PID ditunjukkan oleh gambar 4.5.



Gambar 4.5. Fungsi keanggotaan dari output Fuzzy-PID (a) Fuzzy-P (b) Fuzzy-I (c) Fuzzy-D

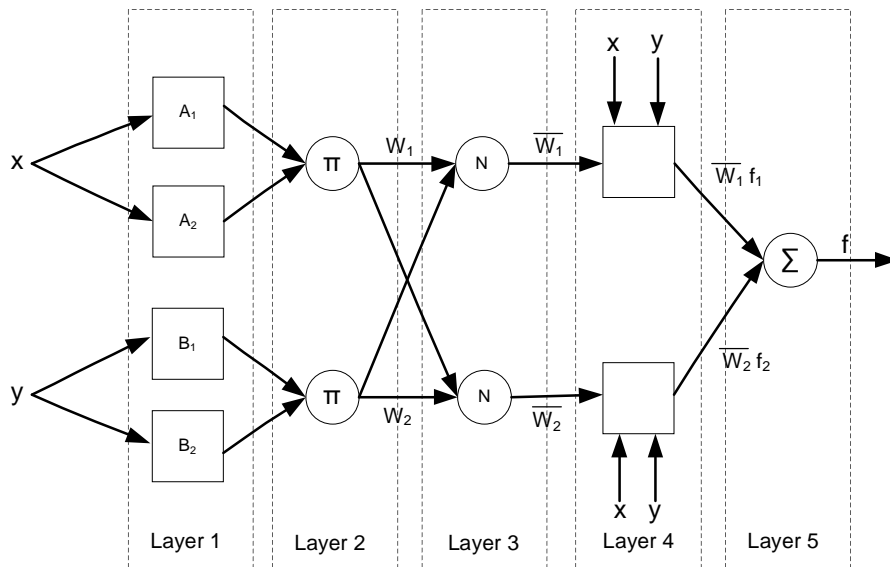
4.3.2. ANFIS (*Adaptive Neuro Fuzzy Inference System*)

ANFIS atau *Adaptive Neuro Fuzzy Inference System* merupakan sebuah jaringan yang mempunyai fungsi yang sama seperti model inferensi fuzzy Takagi-Sugeno yang bersifat adaptif dan bisa diterapkan pembelajaran pada jaringan tersebut menggunakan algoritma pembelajaran seperti pada sistem *neural network*. Aturan fuzzy yang digunakan pada model Takagi-Sugeno adalah dengan menggunakan metode aturan “*if-then*”. Misalkan ada 2 input x dan y , dan output f , maka dua aturan (*rules*) yang digunakan pada model inferensi fuzzy Takagi-Sugeno adalah sebagai berikut:

$$\text{Aturan 1} = \text{If } x \text{ is } A_1 \text{ and } y \text{ is } B_1 \text{ then } f_1 = p_1x + q_1x + r_1$$

$$\text{Aturan 2} = \text{If } x \text{ is } A_2 \text{ and } y \text{ is } B_2 \text{ then } f_2 = p_2x + q_2x + r_2$$

A_1, A_2 dan B_1, B_2 adalah fungsi keanggotaan dari masing-masing input x dan y (bagian premis), sedangkan p_1, q_1, r_1 dan p_2, q_2, r_2 adalah parameter linear (bagian konsekuen) dari model inferensi fuzzy Takagi-Sugeno. Gambar 4.6. menunjukkan mekanisme penalaran untuk model fuzzy Takagi-Sugeno dan arsitektur ANFIS.



Gambar 4.6. Struktur ANFIS

Masing-masing layer dari struktur ANFIS pada gambar dijelaskan sebagai berikut:

- Layer 1

Tiap-tiap titik simpul ke- i pada layer ini merupakan sebuah hasil dari suatu fungsi

$$O_i^1 = \mu_{A_i}(x) \quad (4.10)$$

x adalah input ke suatu titik simpul i . A_i adalah variabel linguistik yang berhubungan dengan titik simpul tersebut, sedangkan μ_{A_i} adalah fungsi keanggotaan dari A_i . Biasanya $\mu_{A_i}(x)$ dipilih sebagai fungsi Gauss atau bentuk lonceng (*bell shape*) yang masing-masing ditunjukkan oleh persamaan 4.11 dan 4.12.

$$\mu_{A_i}(x) = \exp\left\{-\frac{1}{2}\left(\frac{x - c_i}{a_i}\right)^2\right\} \quad (4.11)$$

$$\mu_{A_i}(x) = \frac{1}{1 + \left[\left(\frac{x - c_i}{a_i}\right)^2\right]^{b_i}} \quad (4.12)$$

yang mana x adalah input sedangkan a_i , b_i dan c_i adalah parameter premis.

- Layer 2

Masing-masing titik simpul pada layer ini merupakan titik simpul yang tetap (*fixed*), yang menghitung bobot w_i dari suatu aturan fuzzy (*rule*). Output dari titik simpul ini adalah hasil perkalian dari sinyal yang masuk pada layer ini. Hal ini ditunjukkan melalui persamaan....

$$O_i^2 = w_i = \mu_{A_i}(x) \times \mu_{B_i}(y) \quad (4.13)$$

- Layer 3

Masing-masing titik simpul pada layer ini merupakan titik simpul yang tetap (*fixed*). Setiap titik simpul ke- i menghitung perbandingan antara bobot aturan ke- i terhadap total dari seluruh pembobotan aturan fuzzy. Output dari titik simpul ini merupakan bentuk pembobotan yang dinormalisasi. Hal ini dapat ditunjukkan pada persamaan

$$O_i^3 = \bar{w}_i = \frac{w_i}{w_1 + w_2}, \quad i = 1,2 \quad (4.14)$$

- Layer 4

Masing-masing titik simpul pada layer ini merupakan titik simpul yang adaptif sesuai dengan suatu fungsi titik simpul yang diberikan oleh persamaan 4.15.

$$O_i^4 = \bar{w}_i f_i = \bar{w}_i (p_i x + q_i y + r_i), \quad i = 1, 2 \quad (4.15)$$

\bar{w}_i merupakan output dari layer 3, sedangkan $\{p_i, q_i, r_i\}$ adalah parameter konsekuen.

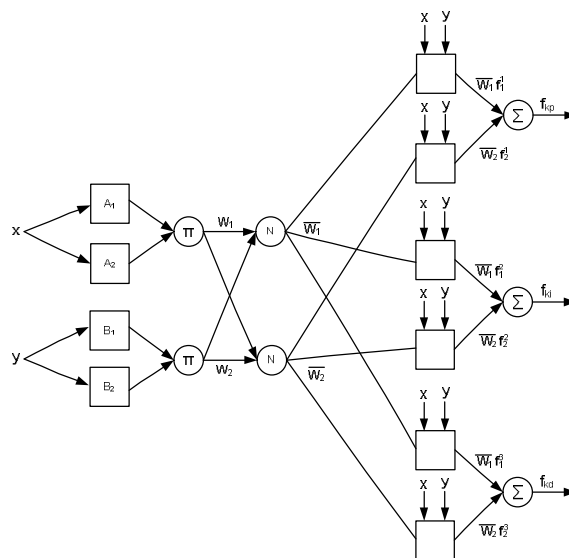
- Layer 5

Layer ini hanya terdiri sebuah titik simpul tetap yang merupakan titik simpul yang menghitung nilai keluaran fuzzy dengan menjumlahkan seluruh sinyal yang masuk pada layer ini.

$$O_i^5 = \sum_i \bar{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \quad (4.16)$$

4.3.3. Perancangan ANFIS untuk Pengaturan Kecepatan Motor BLDC

Struktur ANFIS yang digunakan dalam pengaturan kecepatan motor BLDC pada penelitian ini adalah menggunakan struktur inferensi fuzzy Takagi-Sugeno dengan multi output. Input dari ANFIS adalah berupa error dan perubahan error kecepatan, sedangkan output dari ANFIS adalah berupa parameter PID (K_p , K_i , dan K_d). Struktur ANFIS multi output ditunjukkan oleh gambar 4.7.



Gambar 4.7. Struktur ANFIS dengan multi output

Fungsi keanggotaan yang digunakan adalah fungsi Gauss, sebagaimana yang ditunjukkan oleh persamaan 4.11. Pada penelitian ini digunakan sebanyak 2 (dua) fungsi keanggotaan, karena mempertimbangkan kapasitas memori dan kecepatan perhitungan pada mikrokontroler. Parameter-parameter premis, \mathbf{a}_i , \mathbf{c}_i dilatih dengan menggunakan algoritma PSO (*Particle Swarm Optimization*) sedangkan parameter-parameter konsekuen, \mathbf{p}_i , \mathbf{q}_i , \mathbf{r}_i , diperbarui dengan menggunakan algoritma RLSE (*Recursive Least Square Estimator*). Adapun data pembelajaran yang digunakan adalah hasil kombinasi input-output pada Fuzzy-PID yang telah didesain sebelumnya. Data pembelajaran untuk ANFIS multi-output ditunjukkan pada lampiran 1.

4.3.4. Metode Pembelajaran Hybrid ANFIS Menggunakan Algoritma PSO dan RLSE

4.3.4.1. Metode Pembelajaran ANFIS Menggunakan Algoritma PSO

Particle Swarm Optimization (PSO) merupakan suatu teknik untuk mencari suatu area yang optimal melalui suatu interaksi dari individu dalam suatu kumpulan populasi tertentu. Teknik ini pertama kali diperkenalkan oleh J. Kennedy dan C. Eberhart pada tahun 1995 [23]. Algoritma PSO terinspirasi berdasarkan perilaku dari sekelompok binatang, seperti burung, ikan, dan sebagainya. Algoritma PSO telah banyak menunjukkan keberhasilannya dalam mengoptimasi berbagai macam fungsi [24],[25] dan menjadi suatu daya tarik bagi kalangan peneliti terutama dalam bidang *engineering*.

Pada PSO, sebuah individu disebut sebagai partikel. partikel-partikel beraksi dengan cara yang sama, yaitu memperbarui suatu populasi dengan menggunakan operator tertentu berdasarkan informasi dari suatu fungsi *fitness*, sehingga setiap partikel bisa berpindah ke tempat yang lebih baik dalam suatu area pencarian tertentu. Setiap partikel bergerak dengan kecepatan tertentu yang diatur secara dinamis berdasarkan pengalaman dari partikel yang lain.

Sebuah populasi dalam PSO mempunyai beberapa partikel, yang mana setiap partikel merepresentasikan suatu solusi yang optimal (*fitness*). Setiap partikel juga memuat informasi tentang kecepatan v_i , posisi x_i , posisi dimana *fitness* terbaik berada (p_{best}), serta indeks dari partikel yang terbaik dalam suatu

populasi (g_{best}). Adapun kecepatan dan posisi dari suatu partikel diperbarui berdasarkan persamaan 4.17 dan 4.18.

$$v_i^{n+1} = wv_i^n + c_1r_1^n(p_{best_i}^n - x_i^n) + c_2r_2^n(g_{best}^n - x_i^n) \quad (4.17)$$

$$x_i^{n+1} = x_i^n + v_i^{n+1} \quad (4.18)$$

c_1 dan c_2 merupakan koefisien akselerasi, yang masing-masing disebut dengan parameter kognitif dan parameter sosial dari suatu individu. r_1 dan r_2 adalah suatu bilangan random antara 0 dan 1 yang nilainya selalu berubah-ubah setiap iterasi. w disebut dengan bobot inersia (*inertia weight*) yang berperan dalam mengontrol pengaruh kecepatan partikel sebelumnya terhadap kecepatan saat ini. Nilai w dapat bernilai konstan atau berubah-ubah pada setiap iterasi, seperti pada metode *Linear Decreasing Inertia Weight* (LDM) [26].

$$w_k = w_{max} - \frac{w_{max} - w_{min}}{\max iter} \times k \quad (4.19)$$

w_{max} dan w_{min} nilai maksimum dan minimum dari bobot inersia. $\max iter$ adalah nilai maksimum iterasi, sedangkan k adalah iterasi saat ini.

Dalam metode pembelajaran ANFIS menggunakan PSO, posisi dari partikel merepresentasikan sebagai parameter-parameter premis, sedangkan parameter konsekuen diperbarui menggunakan algoritma RLSE. Sehingga parameter-parameter premis dan konsekuen selalu berubah-ubah setiap iterasi, dan dicari nilai posisi partikel dengan nilai *fitness* paling minimum. Adapun fungsi *fitness* yang digunakan dalam metode ini untuk setiap output adalah nilai minimum dari *root mean square error* (RMSE).

$$fitness = F(\bar{x}_i) = \min \left(\sqrt{\frac{1}{N} \sum_{n=1}^N (y_{d_n} - y_{out_n})^2} \right) \quad (4.20)$$

N merupakan jumlah data training, sedangkan y_d dan y_{out} masing-masing adalah nilai output yang diinginkan dan nilai output sebenarnya.

Algoritma pembelajaran ANFIS menggunakan PSO dilakukan menggunakan program Matlab. Adapun langkah-langkahnya adalah sebagai berikut:

1. Inisialisasi parameter-parameter PSO dan partikel, seperti jumlah partikel dan posisi partikel. Inisialisasi posisi partikel menggunakan fungsi *genfis1*. *genfis1* merupakan fungsi yang digunakan untuk membentuk struktur fuzzy, *Fuzzy Inference System* (FIS), dengan tipe Sugeno menggunakan metode *grid partition* pada data. Ini digunakan sebagai inisialisasi posisi partikel pada PSO
2. Modifikasi *rule base* dengan menambahkan fungsi keanggotaan pada sisi output.
3. Update parameter konsekuen menggunakan algoritma RLSE
4. Evaluasi performa dari setiap partikel berdasarkan posisi saat ini. Untuk sistem ANFIS multioutput, maka fungsi *fitness*-nya sebagai berikut.

$$fitness = F(\bar{x}_i) = \min \left(\frac{1}{K} \sum_{k=1}^K \sqrt{\frac{1}{N} \sum_{n=1}^N (y_{d_n} - y_{out_n})^2} \right) \quad (4.21)$$

K merupakan jumlah output dari ANFIS.

5. Bandingkan hasil evaluasi setiap partikel dengan individu yang mempunyai *fitness* terbaik. Selanjutnya, jika $F(\bar{x}_i) < p_{best_i}$, maka:

$$p_{best_i} = F(\bar{x}_i)$$

$$\bar{x}_{pbest_i} = \bar{x}_i$$
6. Bandingkan setiap individu terbaik dengan partikel yang paling baik (*global best*), jika $F(\bar{x}_i) < g_{best}$, maka:

$$g_{best} = p_{best_i}$$

$$\bar{x}_{gbest} = \bar{x}_{pbest_i}$$
7. Hitung vektor kecepatan untuk setiap partikel menggunakan persamaan 4.17.
8. Update posisi untuk setiap partikel menggunakan persamaan 4.18.
9. Ulangi langkah 4 – 8 sampai iterasi selesai.

4.3.4.2. Resursive Least Square Estimation (RLSE)

Algoritma RLSE digunakan untuk mengestimasi parameter-parameter konsekuen, \mathbf{p}_i , \mathbf{q}_i , \mathbf{r}_i . Persamaan prediksi pada sistem regresi linear dapat dituliskan sebagai.

$$\hat{y}(t) = \varphi^T(t) \theta \quad (4.22)$$

$\varphi^T(t)$ merupakan vektor regresi, sedangkan θ adalah vektor dari parameter-parameter yang tidak diketahui. Untuk data yang diberikan sampai waktu ke- t , maka proses estimasi menggunakan LSE dituliskan sebagai.

$$\hat{\theta}(t) = \left[\sum_{s=1}^t \varphi(s) \varphi^T(s) \right]^{-1} \sum_{s=1}^t \varphi(s) y(s) \quad (4.23)$$

Secara rekursif, persamaan 4.23 dapat dituliskan menjadi persamaan 4.24 dan 4.35.

$$\hat{\theta}(t) = \hat{\theta}(t-1) + P(t) \varphi(t) [y(t) - \varphi^T(t) \hat{\theta}(t-1)] \quad (4.24)$$

$$P(t) = P(t-1) - \frac{P(t-1) \varphi(t) \varphi^T(t) P(t-1)}{1 + \varphi^T(t) P(t-1) \varphi(t)} \quad (4.25)$$

Algoritma ini memerlukan memori yang kecil dan kompleksitas perhitungan yang tidak tinggi (karena tidak terdapat invers matriks). Untuk setiap perhitungan sampel yang baru, hanya matriks P yang diperbaharui.

Jika ada sebuah parameter *forgetting factor* diterapkan pada algoritma ini, maka persamaan 4.24 dan 4.25 menjadi persamaan 4.26 dan 4.27.

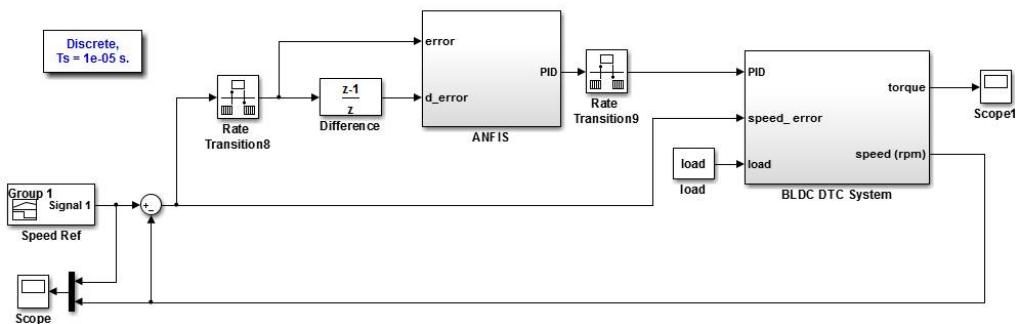
$$\hat{\theta}(t) = \hat{\theta}(t-1) + P(t) \varphi(t) [y(t) - \varphi^T(t) \hat{\theta}(t-1)] \quad (4.26)$$

$$P(t) = \frac{1}{\lambda} \left[P(t-1) - \frac{P(t-1) \varphi(t) \varphi^T(t) P(t-1)}{\lambda + \varphi^T(t) P(t-1) \varphi(t)} \right] \quad (4.27)$$

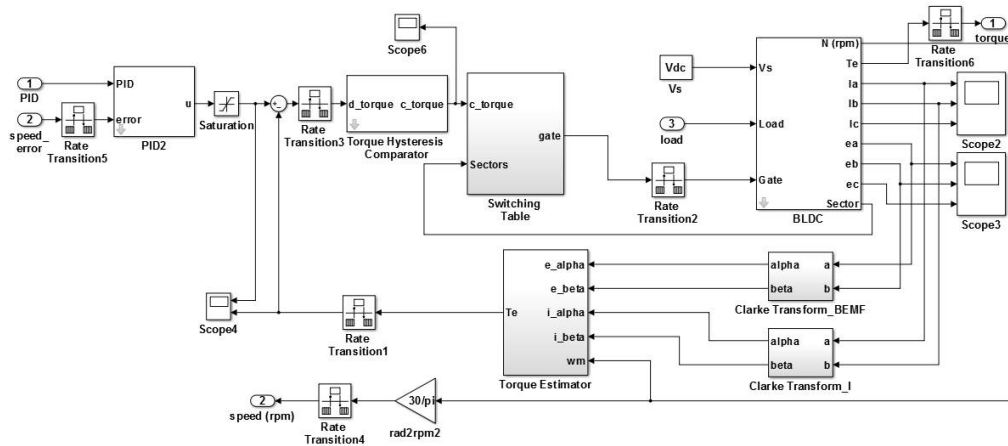
λ merupakan parameter *forgetting factor* yang berpengaruh pada proses *tracking* parameter yang berubah terhadap waktu. Nilai λ ini bernilai $0 < \lambda \leq 1$. Semakin kecil nilai λ , proses *tracking* parameter semakin cepat, tapi sensitifitas *noise* semakin tinggi. Biasanya nilai λ ini dipilih antara $0.94 - 0.999$ [27].

4.3.5. Simulasi Sistem

Sistem kontrol kecepatan motor BLDC menggunakan ANFIS disimulasikan menggunakan Simulink MATLAB, sebelum hasil dari perancangan diimplementasikan dalam sebuah *hardware*. Adapun desain simulasi keseluruhan sistem kontrol kecepatan motor BLDC berbasis ANFIS ini ditunjukkan pada gambar 4.8 dan 4.9.



Gambar 4.8. Simulasi sistem kontrol kecepatan motor BLDC menggunakan ANFIS

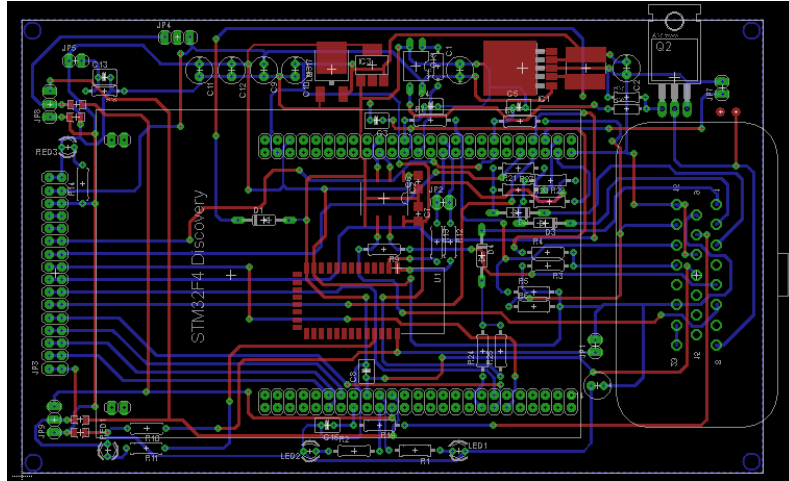


Gambar 4.9. Simulasi model sistem motor BLDC berbasis DTC

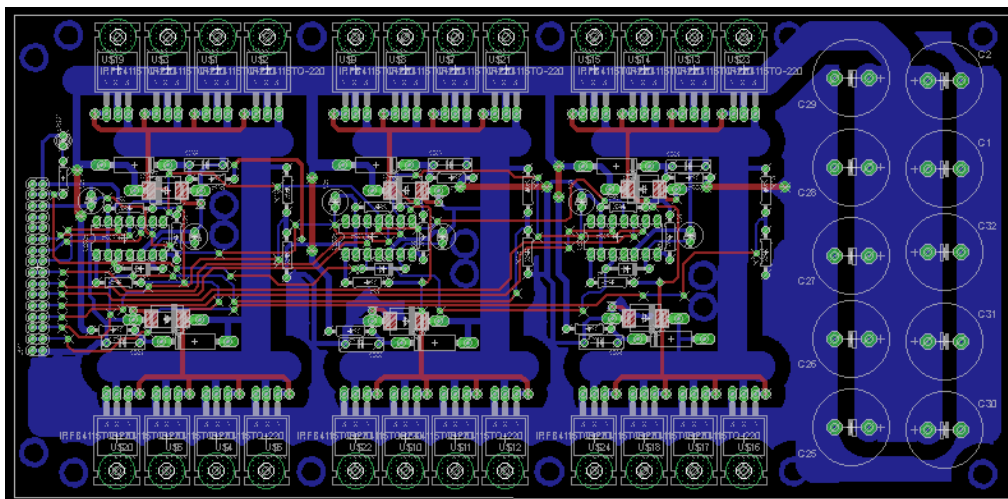
4.3.6. Perancangan Hardware

Hasil dari perancangan yang telah disimulasikan selanjutnya diimplementasikan dalam sebuah *hardware*. *Hardware* yang digunakan dirancang menggunakan software EAGLE (*Einfach Anzuwendender Grafischer Layout-Editor*). *Hardware* yang didesain terdapat 2 (dua) buah modul utama, modul

mikrokontroler dan modul driver, yang masing-masing desain ditunjukkan pada gambar 4.10 dan 4.11.



Gambar 4.10. Desain layout PCB untuk modul mikrokontroler



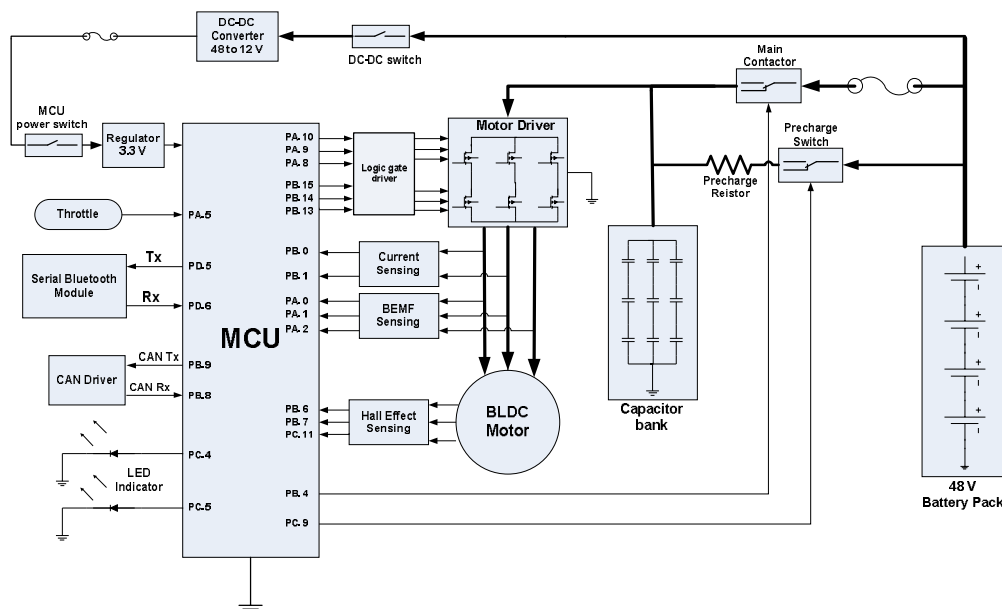
Gambar 4.11. Desain layout PCB untuk modul driver

Mikrokontroler yang digunakan dalam sistem kontrol ini adalah mikrokontroler berbasis 32 bit, yaitu STM32F407. Kelebihan dari mikrokontroler ini adalah frekuensi clock CPU-nya tinggi, yaitu bisa mencapai 168 MHz, sehingga kecepatan pemrosesan datanya dapat berlangsung dengan cepat, terutama pada saat melakukan perhitungan proses transformasi $\alpha\beta$. Mikrokontroler tipe ini juga dilengkapi *Analog to Digital Converter* (ADC) dengan resolusi 12 bit, sehingga mampu membaca data sensor arus dan tegangan dengan tingkat kepresisian yang tinggi, serta dilengkapi dengan kontroler DMA

(*Direct Memory Access*) yang mampu untuk mentransfer data dari peripheral ke memori maupun dari memori ke memori dengan kecepatan yang tinggi. Disamping itu, selain dilengkapi dengan *timer* biasa, mikrokontroler ini juga dilengkapi dengan *advanced-control timer* dengan fitur yang mampu membangkitkan 6-step PWM, sehingga sangat cocok diaplikasikan untuk kontrol inverter 3-fasa [28].

Pada sisi driver, digunakan MOSFET tipe IRFP4115GPBF. MOSFET tipe ini merupakan MOSFET untuk aplikasi power switching dengan kecepatan tinggi. Dengan ukurannya yang kecil, MOSFET ini mampu dialiri arus sampai 104 A, dan mampu disuplai dengan V_{dss} sebesar 150 V [29]. Untuk driver gate MOSFET digunakan IC gate driver IR2110. IC ini merupakan driver untuk MOSFET sisi atas dan bawah pada sistem H-bridge yang mempunyai kecepatan tinggi dan mampu diaplikasikan untuk MOSFET yang bertegangan tinggi (500 – 600 V) [30].

Adapun skematik diagram untuk keseluruhan sistem ditunjukkan oleh gambar 4.12.



Gambar 4.12. Skematik diagram keseluruhan sistem pengaturan kecepatan motor BLDC

4.3.6.1. Sensor Arus

Pada sistem kontrol kecepatan motor BLDC ini, digunakan sensor arus Tamura L01Z400S05 (gambar 4.13). Sensor ini merupakan salah satu sensor arus seri L01ZXXXS05 dengan disain yang terisolasi, sehingga proses pengukuran untuk arus yang tinggi tidak sampai mengganggu sistem mikrokontroler. Berdasarkan *datasheet*, tercantum bahwa sensor L01Z400S05 mampu mengukur arus hingga 400 A [31]. namun, kelemahan dari sensor ini adalah tidak dilengkapi dengan karakteristik hubungan antara tegangan output yang dihasilkan dengan arus yang terdeteksi. Sehingga diperlukan proses pengukuran secara manual untuk proses kalibrasi. Data hasil kalibrasi untuk sensor ini ditunjukkan pada tabel 4.6.



Gambar 4.13. Sensor arus Tamura L01ZXXXS05

Tabel 4.6. Kalibrasi sensor arus L01Z400S05

Tegangan output sensor (volt)	Arus (A)	ΔV	Faktor Kalibrasi (F_c)
1.3417	6.0	0.01344	446.3595
1.3455	7.8	0.01727	451.7202
1.3514	11.6	0.02319	500.2562
1.3623	17.3	0.03414	506.7575
1.3887	29.1	0.06045	481.3462
1.3970	34.5	0.06879	501.5112
1.4087	39.7	0.08045	493.4408

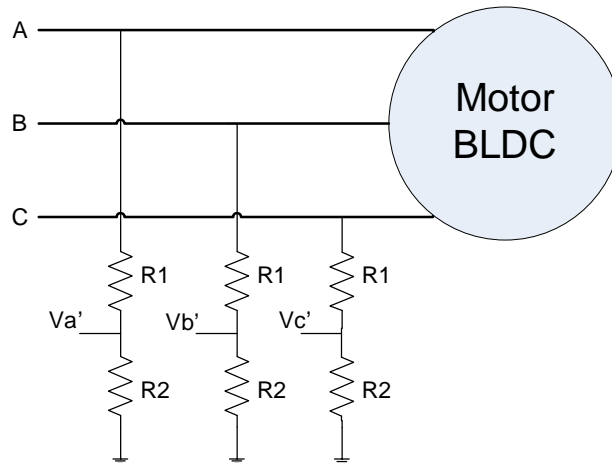
Sensor L01Z400S05 merupakan sensor arus DC *bidirectional*, maka dapat mendeteksi arus listrik dengan arah yang berlawanan. Sehingga, ketika tidak ada arus yang mengalir melalui sensor ini ($I = 0 A$), output sensor mengeluarkan nilai tegangan tertentu, yang kemudian disebut sebagai tegangan *offset*. Nilai

offset inilah yang nanti menjadi acuan dalam pengukuran arus yang mengalir pada sistem. ΔV merupakan selisih antara tegangan output sensor terhadap tegangan *offset*. Berdasarkan data pada tabel 4.4, didapatkan hasil bahwa terdapat 2 (dua) buah nilai faktor kalibrasi (F). Untuk $\Delta V < 0.02$, $F_c = 449.0399$ dan untuk $\Delta V \geq 0.02$, $F_c = 496.6624$. Nilai F_c ini didapatkan dengan merata-rata semua data sesuai dengan pengelompokan berdasarkan nilai ΔV . Dengan demikian, nilai arus yang terukur dapat dituliskan sebagai.

$$I = F_c * \Delta V \quad (4.28)$$

4.3.6.2. Sensor Tegangan BEMF

Sensor tegangan BEMF digunakan dua buah resistor pembagi tegangan, sebagaimana ditunjukkan pada gambar 4.14.



Gambar 4.14. Sensor tegangan BEMF

Jika V_x menunjukkan tegangan terminal pada masing-masing fasa motor BLDC dan V'_x adalah tegangan yang terukur pada sensor, maka tegangan terminal V_x dapat dituliskan sebagai.

$$V_x = \frac{R_1 + R_2}{R_2} \times V'_x \quad (4.29)$$

Persamaan elektrik untuk setiap belitan dituliskan sebagai berikut.

$$V_{xn} = R i_x + L \frac{di_x}{dt} + e_x \quad (4.30)$$

V_{xn} merupakan tegangan terminal terhadap titik netral belitan, atau bisa dituliskan $V_{xn} = V_x - V_n$. Jika *losses* pada setiap belitan diabaikan, maka dapat dituliskan persamaan tegangan untuk masing-masing fasa.

$$V_a - V_n = e_a \quad (4.31)$$

$$V_b - V_n = e_b \quad (4.32)$$

$$V_c - V_n = e_c \quad (4.33)$$

Jika diasumsikan sistem seimbang, maka $e_a + e_b + e_c = 0$, sehingga tegangan netral V_n dapat diketahui.

$$V_n = \frac{1}{3}(V_a + V_b + V_c) \quad (4.34)$$

Dengan demikian, tegangan BEMF dapat dihitung sebagai.

$$e_a = \frac{2}{3}V_a - \frac{1}{3}(V_b + V_c) \quad (4.35)$$

$$e_b = \frac{2}{3}V_b - \frac{1}{3}(V_a + V_c) \quad (4.36)$$

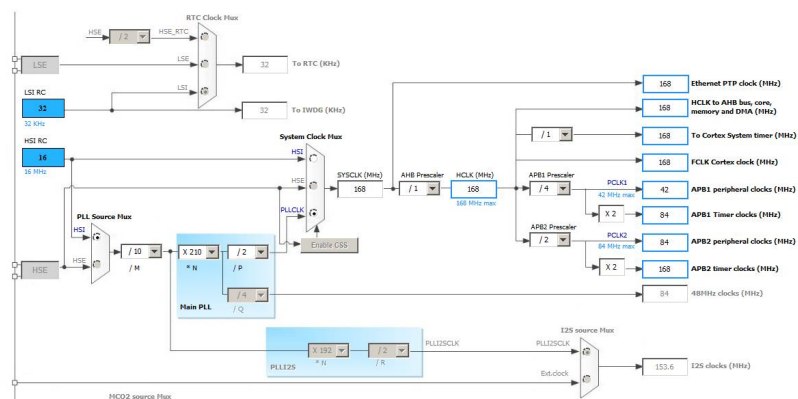
$$e_c = \frac{2}{3}V_c - \frac{1}{3}(V_a + V_b) \quad (4.37)$$

4.3.7. Perancangan *Firmware*

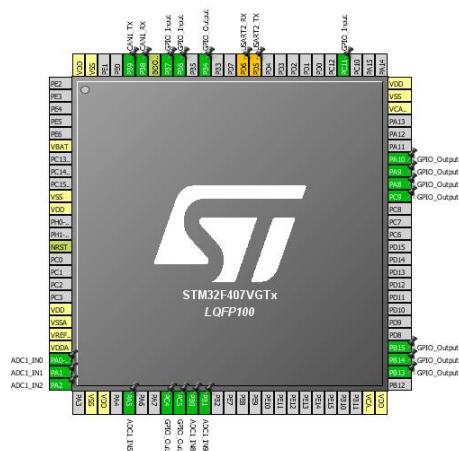
Firmware merupakan istilah yang mengacu pada sebuah perangkat lunak yang tersimpan pada sebuah ROM (*Read Only Memory*), biasanya berhubungan dalam sistem tertanam (*embedded system*). Hasil dari perancangan *firmware* ini nantinya akan ditanamkan pada sebuah mikrokontroler berbasis 32-bit (STM32F407). *Firmware* ini dirancang menggunakan bahasa pemrograman C dengan *compiler* ARM GCC dengan memanfaatkan *standard peripheral library* dari STM32F4xx. Secara garis besar proses perencanaan *firmware* ini terdiri dari 2 (dua) tahapan proses, yaitu proses inisialisasi dan proses perancangan untuk setiap rutin program *interrupt*.

4.3.7.1. Proses Inisialisasi

Proses inisialisasi merupakan proses yang pertama kali dieksekusi oleh sebuah mikrokontroler. Proses inisialisasi ini meliputi; inisialisasi clock untuk setiap periperal di dalam mikrokontroler, inisialisasi GPIO (*General Purpose Input/Output*), inisialisasi *timer*, dan inisialisai beberapa periperal seperti ADC (*Analog to Digital Converter*), komunikasi serial USART (*Universal Synchronous/Asynchronous Receiver/Transmitter*), dan komunikasi CAN (*Control Area Network*). Proses inisialisasi ini disimulasikan menggunakan STMCubeMX baik pengaturan *clock* maupun pengaturan I/O untuk *general purpose* maupun fungsi yang lain, sehingga tidak ada pin I/O yang terjadi *crash* atau terdefinisi sebagai 2 (dua) fungsi yang berbeda. Fungsi untuk setiap pin I/O yang digunakan dapat dilihat pada gambar 4.12. Adapun pengaturan *clock* dan inisialisasi periperal pada mikrokontroler ditunjukkan pada gambar 4.15 dan 4.16.



Gambar 4.15. Simulasi pengaturan parameter *clock* pada STM32F407



Gambar 4.16. Simulasi pengaturan inisialisai pin I/O pada STM32F407

Pada desain sistem kontrol kecepatan motor BLDC ini digunakan 4 (empat) buah *timer*, yaitu, *timer* 2 untuk perhitungan kecepatan, *timer* 3 untuk pengiriman data serial dengan update *interrupt* setiap 1 ms, *timer* 4 untuk perhitungan ANFIS dengan update *interrupt* setiap 1 ms, dan *timer* 7 digunakan untuk proses kontrol dan komutasi dengan update *interrupt* setiap 50 us.

Untuk pengukuran arus digunakan pin ADC1_IN8 dan ADC1_IN9, sedangkan untuk pengukuran tegangan BEMF digunakan pin ADC1_IN0, ADC1_IN1, dan ADC1_IN2 yang masing-masing terhubung pada DMA (*Direct Memory Access*) dengan resolusi pembacaan 8 bit, mode konversi secara *continous*, dan sampel pembacaan untuk setiap ADC adalah 144 *cycles*. Demikian juga untuk pengaturan ADC pada pembacaan sinyal dari *throttle*.

Komunikasi antara *board* mikrokontroler dengan komputer digunakan modul *Bluetooth* HC-05 dengan protokol komunikasi serial USART. Adapun pengaturan USART yang digunakan adalah dengan mode Tx-Rx, *baudrate* 115200, *stop bits* 1, *no parity* dan tanpa pengaturan *hardware flow control*. Sedangkan untuk komunikasi dengan perangkat lain seperti modul BMS (*Battery Management System*) dan IVC (*In-Vehicle Computer*) digunakan komunikasi serial dengan protokol komunikasi CAN yang merupakan standar komunikasi pada sistem otomotif, dengan kecepatan *baudrate* sebesar 500 kHz.

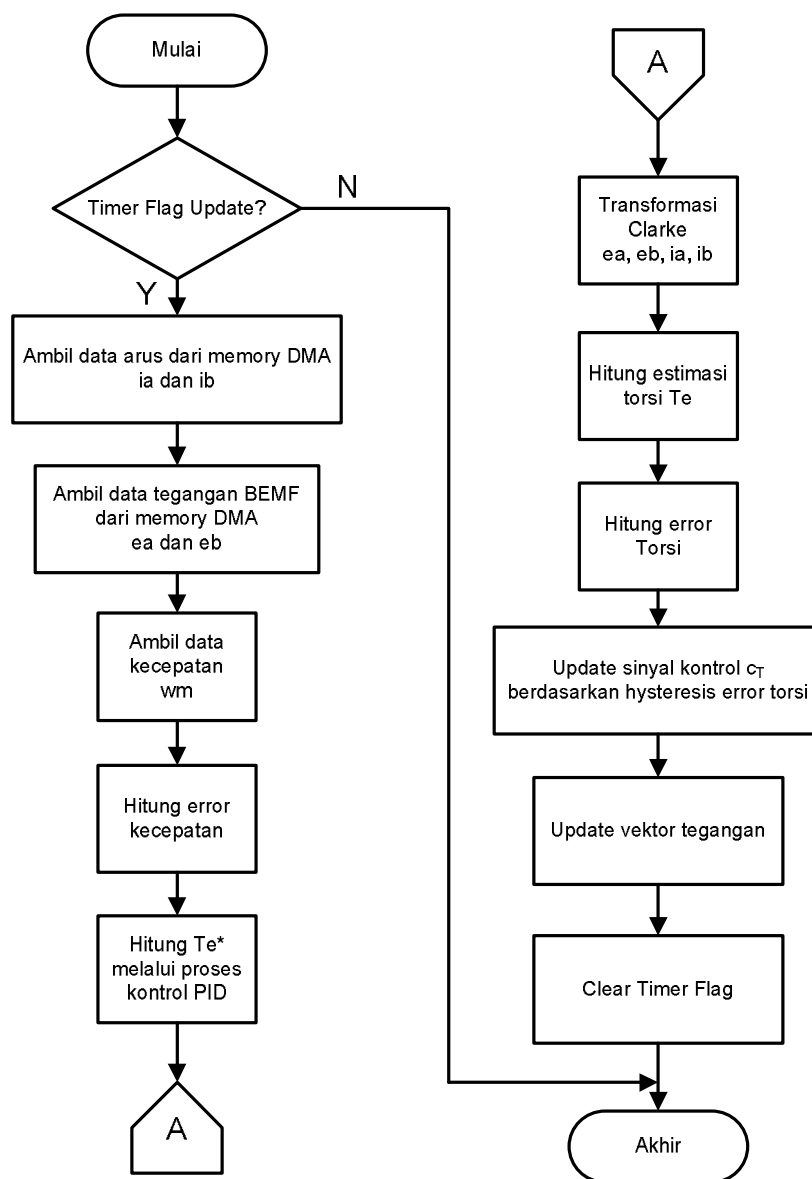
4.3.7.2. Proses Rutin *Interrupt*

Proses rutin *interrupt* merupakan proses yang sangat berhubungan erat dengan sistem yang *real time* karena program dalam rutin ini akan terjadi ketika sinyal interupsi diterima oleh mikrokontroler, baik itu berasal dari *timer* maupun sinyal yang berasal dari luar (*external interrupt signal*). Oleh sebab itu, pada aplikasi sistem kontrol kecepatan motor BLDC proses *interrupt* ini sangat diperlukan. Secara garis besar, proses rutin *interrupt* yang dijalankan untuk aplikasi ini adalah *timer interrupt* dan *external interrupt*. *Timer interrupt* terdiri dari 2 (dua) timer yang digunakan untuk update proses kontrol, yaitu *timer* 7 untuk proses kontrol PID, dan *timer* 4 untuk update parameter PID menggunakan ANFIS. *External interrupt* merupakan proses interupsi ketika sistem mendeteksi

perubahan sinyal pada *hall effect sensors*, yang digunakan untuk perhitungan kecepatan.

a. *Timer Interrupt*

Timer 7 digunakan sebagai *timer* untuk proses kontrol PID dan update proses komutasi berdasarkan nilai *hysteresis error* torsi pada motor BLDC. *Timer* ini akan membangkitkan sinyal interupsi setiap 50 us, karena sistem DTC memerlukan waktu proses kontrol yang cepat. Proses yang dilakukan saat sinyal interupsi dibangkitkan oleh *timer 7* ini dapat dilihat pada gambar 4.17.



Gambar 4.17. Flowchart program rutin *interrupt timer 7*

b. *External Interrupt*

External interrupt digunakan untuk mendeteksi adanya perubahan sinyal pada *hall effect sensor*, sehingga dapat digunakan untuk menghitung kecepatan motor BLDC. *External interrupt* bekerja bersama dengan *timer 2*, dimana *timer 2* digunakan untuk menghitung lebar pulsa ketika sinyal *hall effect sensor* mempunyai logik 0 atau 1. Ilustrasi pengukuran kecepatan dengan metode pengukuran lebar pulsa dari *hall effect sensor* ditunjukkan oleh gambar 4.18.



Gambar 4.18. Perhitungan lebar pulsa *hall effect sensor* oleh *timer 2*

Pin input *external interrupt* diatur untuk mendeteksi adanya pulsa *rising edge* dan *falling down*, sehingga untuk 1 pulsa *high* atau *low* merepresentasikan setengah dari periode *hall effect sensor*. Sehingga dapat dituliskan panjang periode untuk 1 siklus dari *hall effect sensor* sebagai berikut.

$$T = counter * \frac{1}{f_{timer}} \quad (4.38)$$

counter merupakan jumlah pulsa setiap sampling waktu, sedangkan f_{timer} merupakan frekuensi *clock* untuk *timer 2*, dimana frekuensi ini diatur sebesar 1 MHz. Dengan demikian, kecepatan mekanik motor BLDC (dalam rpm) dapat dirumuskan sebagai berikut.

$$w_m = \frac{2}{p} \left(\frac{f_{timer} \times 60}{2 \times counter} \right) \quad (4.39)$$

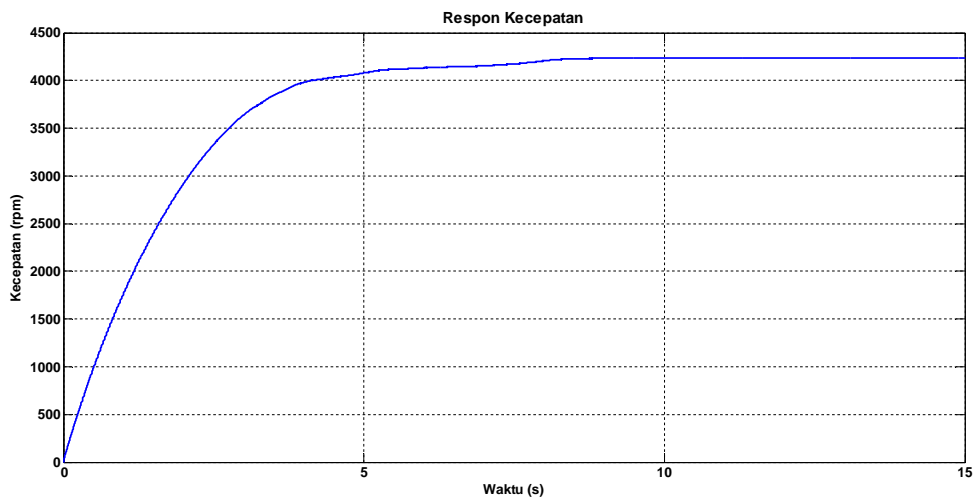
p adalah jumlah kutub pada motor BLDC, konstanta 60 merupakan representasi konversi dari detik ke menit (rpm) dan karena *counter* merepresentasikan sebagai setengah siklus perubahan *hall effect sensor*, maka untuk satu periode penuh dari *hall effect sensor* perlu dikalikan dengan 2.

BAB 5 PENGUJIAN DAN ANALISIS

Pengujian yang dilakukan pada penelitian ini meliputi, simulasi model dan pengujian implementasi pada *hardware*. Selanjutnya, hasil dari simulasi maupun implementasi dianalisis. Hasil yang didapatkan dari penelitian ini selanjutnya akan digunakan sebagai acuan dalam pengembangan sistem pengaturan pada kendaraan listrik, khususnya dalam pengimplementasian hasil desain sistem kontrol ke sebuah sistem tertanam (*embedded system*) yang menggunakan mikrokontroler maupun mikroprosesor.

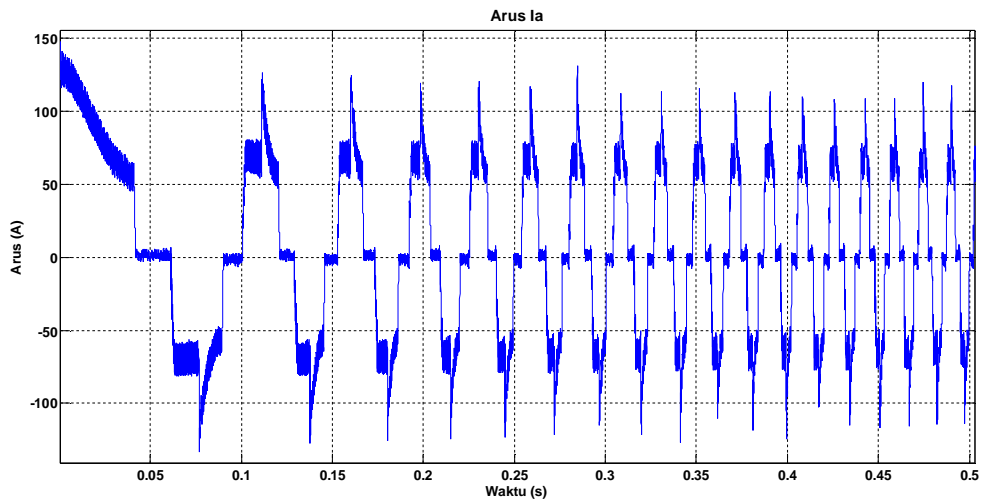
5.1. Simulasi Sistem *Open Loop* Motor BLDC Berbasis DTC

Sebelum melakukan simulasi pengaturan kecepatan motor BLDC, dilakukan uji sistem *open loop*. Hal ini dilakukan untuk mengetahui performa sistem sebelum diberikan sebuah sistem kontrol. Simulasi *open loop* dilakukan dengan cara diberi input tegangan sesuai dengan rating motor BLDC, selanjutnya diamati arus *start* motor.



Gambar 5.1. Respon kecepatan motor BLDC sistem *open loop*

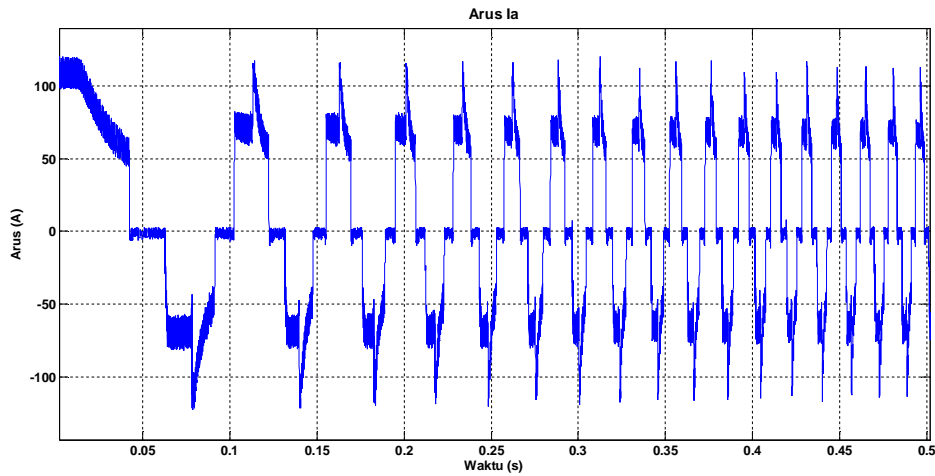
Dengan memberikan input torsi referensi sebesar 11,1 N.m, didapatkan respon kecepatan seperti pada gambar 5.1. Kecepatan *steady state* motor BLDC mencapai 4237 rpm dengan *rise time* sebesar 3,146 s. Arus *start* pada kondisi ini ditunjukkan oleh gambar 5.2.



Gambar 5.2. Arus *start* fasa-a sistem *open loop*

Dengan memberikan input torsi referensi sebesar 11,1 N.m, arus *start* motor BLDC mencapai 142,1 A. Jika menggunakan tipe MOSFET IRFP4115 untuk driver motor BLDC, maka diperlukan minimal 2 buah MOSFET paralel untuk setiap sisi (*high side-low side*) masing-masing fasa. Dengan mengetahui karakteristik sistem *open loop*, maka dapat menentukan perancangan *hardware* yang akan dibuat.

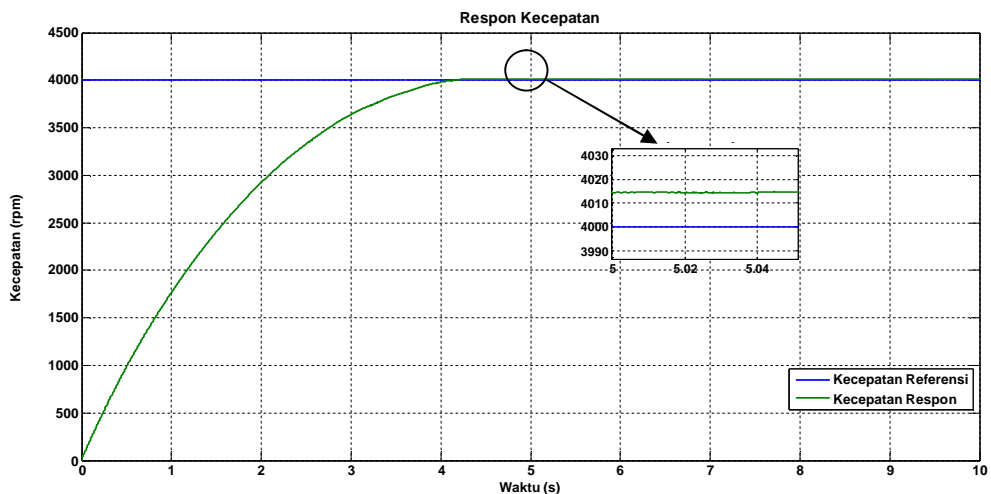
Dengan membatasi nilai torsi referensi, maka dapat membatasi arus pada stator. Akan tetapi, pada saat motor *start*, proses estimasi torsi elektromagnetik tidak berjalan dengan baik, karena saat motor *start*, tegangan BEMF belum terbentuk, sehingga proses estimasi menghasilkan nilai yang tidak akurat. Dengan demikian, untuk membatasi arus yang masuk pada stator, dilakukan 2 (dua) cara, yaitu dengan membatasi nilai torsi referensi sesuai dengan nilai rating pada motor, dan ditambahkan algoritma pembatas arus, jika arus melebihi nilai batas yang ditentukan, maka inverter memberikan vektor tegangan V_0 pada motor. Gambar 5.3 menunjukkan arus *start* setelah ditambahkan pembatas arus sebesar 100 A.



Gambar 5.3. Arus *start* fasa-*a* dengan pembatas arus

5.2. Simulasi Pengaturan Kecepatan Motor BLDC dengan Metode Fuzzy-PID

Fuzzy-PID yang telah didesain kemudian disimulasikan untuk meninjau performa dari sistem kontrol sebelum digunakan sebagai acuan pembelajaran pada ANFIS. Simulasi dilakukan dengan memberikan nilai referensi kecepatan tertentu, yaitu 4000 rpm dan tanpa pembebanan. Hasil respon kecepatan dari simulasi yang telah dilakukan ditunjukkan oleh gambar 5.4.



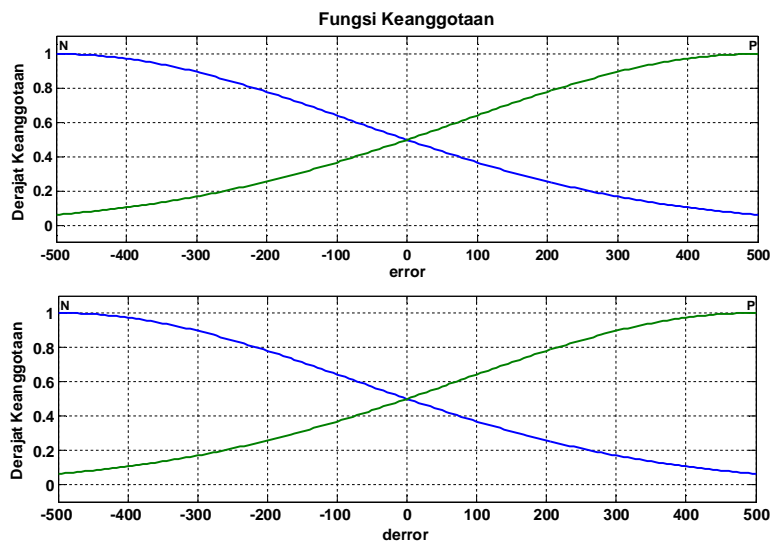
Gambar 5.4. Respon kecepatan motor BLDC menggunakan kontrol Fuzzy-PID

Berdasarkan hasil simulasi, dapat dilihat bahwa Fuzzy-PID yang telah didesain mampu mengontrol kecepatan motor BLDC dengan sangat baik. Sistem

kontrol mampu mencapai kecepatan yang diinginkan dengan error *steady state* sebesar 0,375 % dengan waktu *rise time* 2,768 s.

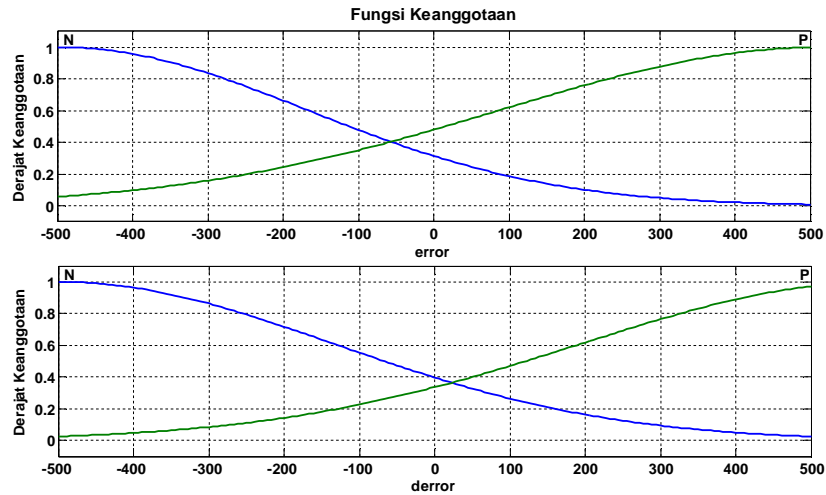
5.3. Pembelajaran ANFIS dengan Supervisi Fuzzy-PID

Fuzzy-PID yang telah didesain telah terbukti mampu mengatur kecepatan motor BLDC dengan baik. Selanjutnya, model ANFIS multi output yang telah didesain dilatih dengan kombinasi data input/output berdasarkan dengan nilai dari Fuzzy-PID. Input *error* maupun $\Delta error$ mempunyai range nilai dari -500 sampai 500, maka untuk data yang akan ditraining dipilih sejumlah 20 data *error* dan $\Delta error$ dari range nilai input dengan menggunakan metode jarak interval linear. Selanjutnya, nilai *error* dan $\Delta error$ dikombinasikan, sehingga akan terdapat 400 buah nilai pasangan input. Data ini ditunjukkan oleh table pada lampiran I. Berdasarkan pasangan nilai input ini, digunakan Fuzzy-PID untuk mendapatkan pasangan output K_p , K_i an K_d (f_{Kp} , f_{Ki} , f_{Kd}). Kombinasi data input/output ini menjadi data yang akan ditraining pada ANFIS. untuk test performa, hasil training ANFIS yang telah didapatkan langsung diujikan pada sistem kontrol kecepatan motor BLDC. Pada input ANFIS digunakan 2 (dua) buah fungsi keanggotaan, yaitu N (*Negative*) dan P (*Positive*) yang masing-masing merupakan fungsi Gauss. Gambar 5.5. menunjukkan fungsi keanggotaan sebelum ANFIS ditraining.

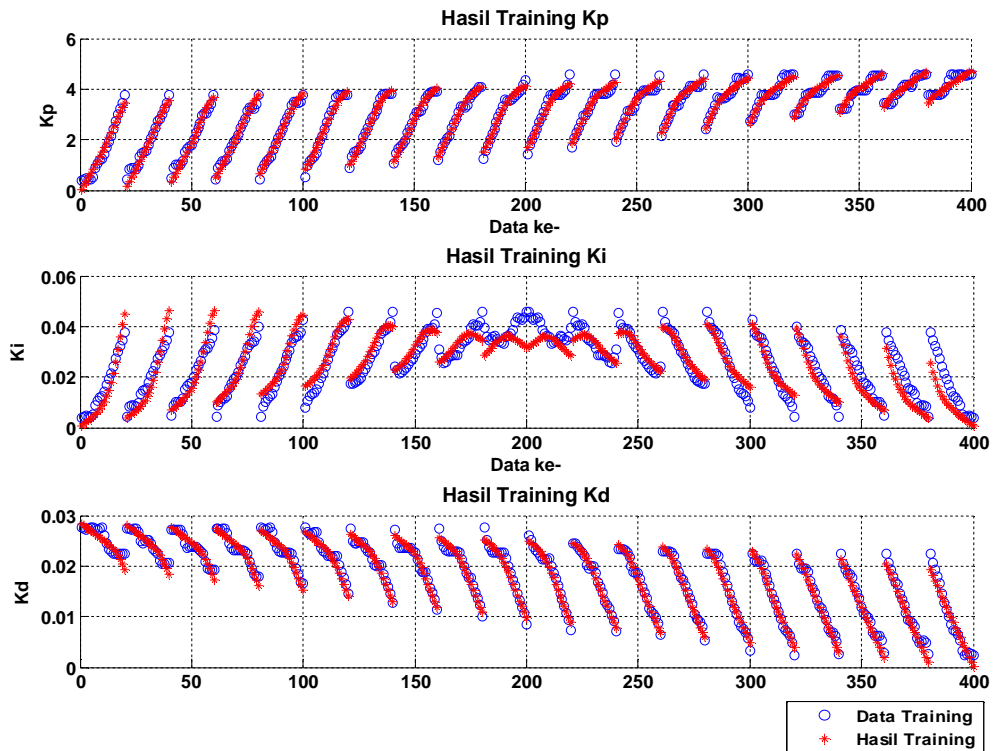


Gambar 5.5. Fungsi keanggotaan ANFIS sebelum proses training

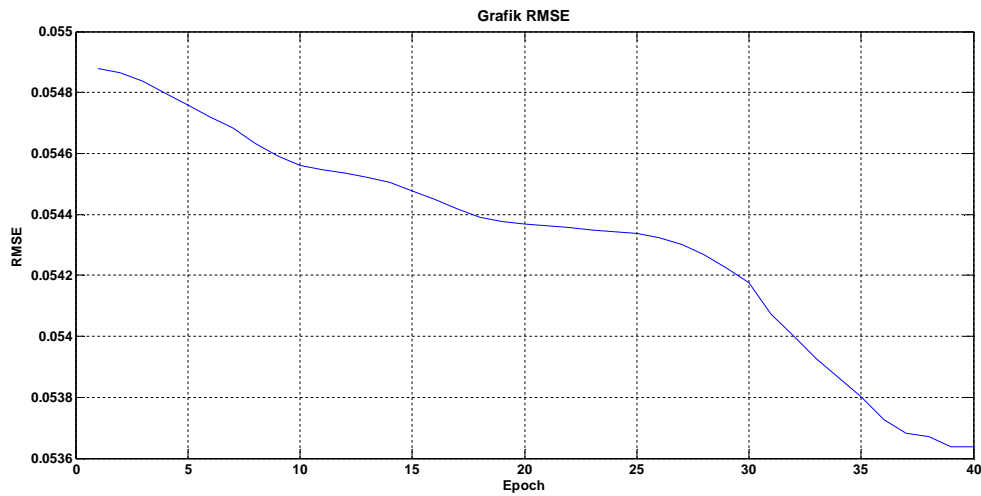
Percobaan pertama dilakukan dengan mengatur nilai $\lambda = 1$. Selanjutnya nilai λ diturunkan dari 0,99 sampai 0,94. Hasil training untuk $\lambda = 1$ ditunjukkan oleh gambar 5.6 – 5.8.



Gambar 5.6. Fungsi keanggotaan ANFIS setelah proses training



Gambar 5.7. Data training K_p K_i K_d



Gambar 5.8. Grafik konvergensi RMSE

Berdasarkan hasil training yang dilakukan, didapatkan bahwa untuk nilai $\lambda = 1$, nilai RMSE setelah proses training mencapai konvergen adalah 0,05364. Hasil proses training untuk nilai λ dari 0,99 sampai 0,94 dapat dilihat pada tabel 5.1.

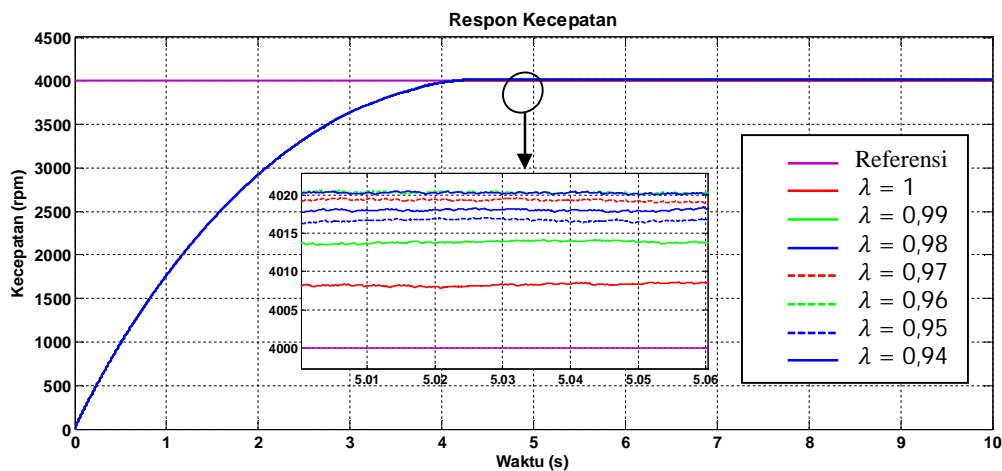
Tabel 5.1. Hasil proses training ANFIS

λ	RMSE
1	0,05364
0,99	0,05592
0,98	0,05539
0,97	0,09228
0,96	0,17806
0,95	0,26434
0,94	0,33473

Setelah dilakukan proses training, dapat dikatakan bahwa metode *hybrid* PSO dan RLSE mampu melakukan proses training pada ANFIS multi output dengan baik. Berdasarkan data pada tabel 5.2. dapat diketahui bahwa semakin kecil nilai λ , nilai RMSE semakin besar. Data hasil proses training berdasarkan nilai error RMSE tidak dapat langsung dipilih untuk diterapkan pada sistem. Dari setiap hasil proses training akan diujikan pada sistem kontrol kecepatan motor BLDC, sehingga didapatkan respon kecepatan dengan error *steady state* yang paling minimum.

5.4. Simulasi Pengaturan Kecepatan Motor BLDC dengan Metode ANFIS

Dari hasil training ANFIS yang telah didapatkan, selanjutnya dilakukan pengujian pada simulasi sistem kontrol kecepatan. Dilakukan perbandingan respon kecepatan untuk masing-masing hasil dengan *forgetting factor* (λ) yang berbeda. Hasil yang terbaik akan menjadi parameter untuk diimplementasikan pada *hardware*. Hasil respon kecepatan dengan kontrol ANFIS ditunjukkan oleh gambar 5.9.



Gambar 5.9. Respon kecepatan menggunakan ANFIS

Dari pengujian dari model ANFIS yang telah didapatkan, didapatkan hasil performa respon kecepatan motor BLDC seperti pada tabel 5.2.

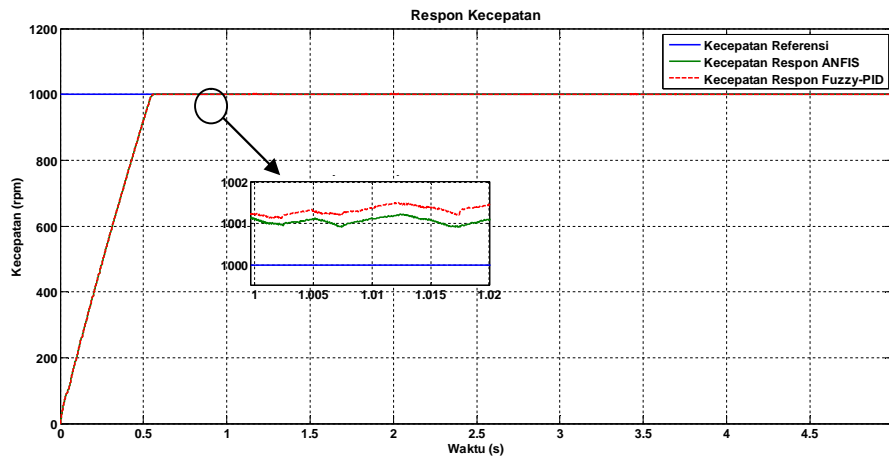
Tabel 5.2. Performa ANFIS terhadap respon kecepatan berdasarkan error *steady state*

λ	Error <i>steady state</i> (%)
1	0,2001
0,99	0,3297
0,98	0,4339
0,97	0,4652
0,96	0,4873
0,95	0,3960
0,94	0,4874

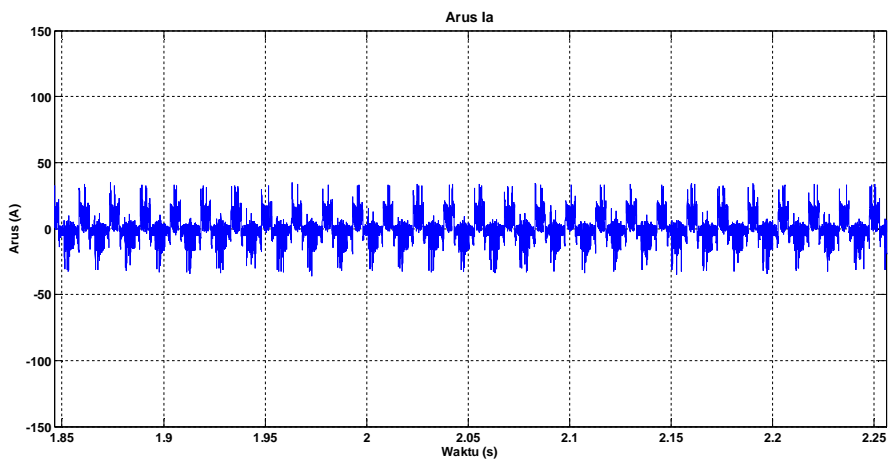
Berdasarkan hasil pengujian tersebut model ANFIS yang paling baik dalam pengujian respon kecepatan adalah saat $\lambda = 1$, dengan error *steady state* adalah 0,2 % dengan *rise time* 2,7437 s.

5.4.1. Simulasi dengan Kecepatan Referensi Tetap dan Tanpa Beban

Pada kondisi kecepatan referensi tetap dan tanpa beban, akan diamati arus dan karakteristik torsi pada sistem motor BLDC yang akan diuji. Kecepatan diatur sebesar 1000 rpm. Hasil simulasi untuk kecepatan tetap dan tanpa beban ditunjukkan oleh gambar 5.10 – 5.12.

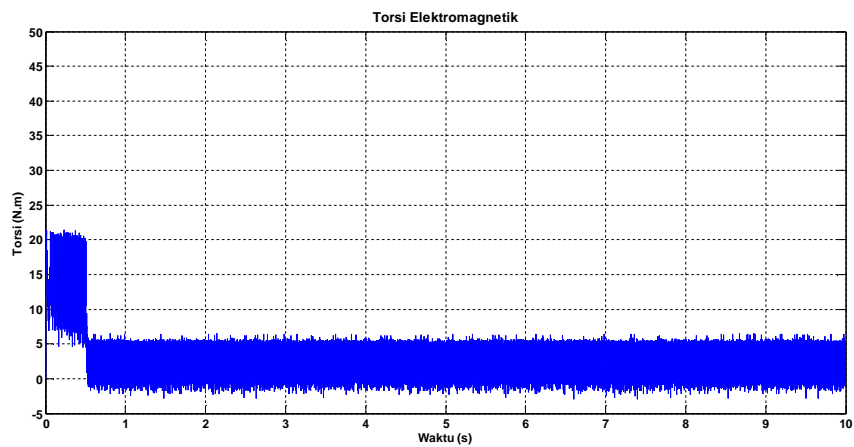


Gambar 5.10. Respon kecepatan dengan referensi tetap dan tanpa beban

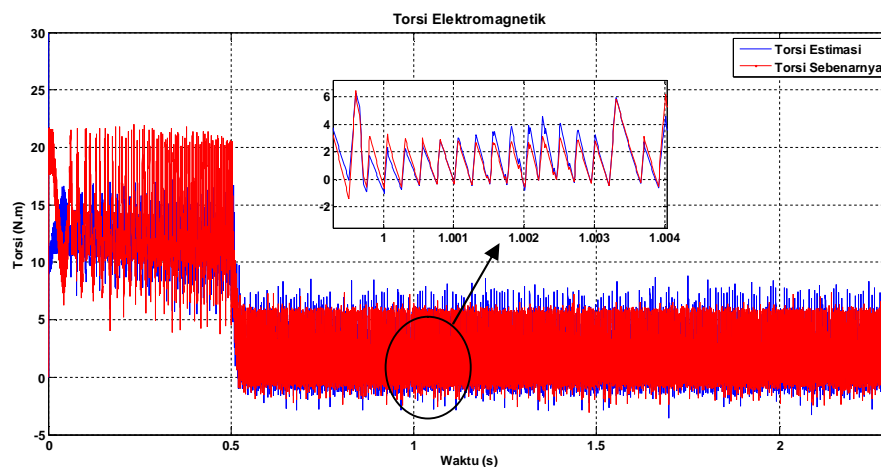


Gambar 5.11. Arus pada fasa-a (I_a) untuk kecepatan referensi tetap (1000 rpm) dan tanpa beban

Error kecepatan pada saat *steady state* untuk kontroler ANFIS sebesar 0,10 %, sedangkan 0,15% untuk Fuzzy-PID, sehingga dapat dikatakan bahwa sistem kontrol kecepatan menggunakan ANFIS mempunyai respon yang lebih baik dari pada Fuzzy-PID. Dengan kecepatan referensi tetap sebesar 1000 rpm, arus *start* dibatasi sebesar 100 A dan arus puncak *steady state* mencapai 33,45 A



Gambar 5.12. Torsi untuk kecepatan referensi tetap (1000 rpm) dan tanpa beban dengan nilai rms sebesar 10,5 A. Kondisi *steady state* dicapai pada saat 0,5 s dengan waktu *rise time* sebesar 0,415 s. Sistem kontrol kecepatan berbasis DTC mempunyai *ripple* torsi yang besar. Torsi pada saat motor *start* dibatasi sebesar 11,1 N.m. Gambar 5.13 menunjukkan perbandingan torsi estimasi dengan torsi yang sebenarnya.

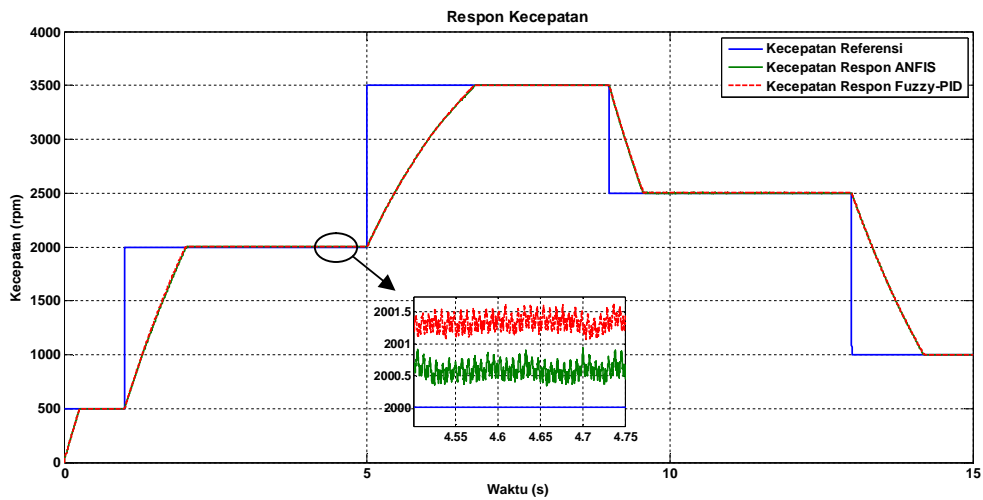


Gambar 5.13. Perbandingan nilai torsi estimasi dengan torsi yang sebenarnya

Berdasarkan perbandingan yang dilakukan, torsi dapat diestimasi dengan baik, tetapi pada saat *start*, estimasi torsi sangat buruk. Hal ini disebabkan karena saat *start*, tegangan BEMF belum bisa dideteksi dengan baik, sehingga menyebabkan proses estimasi tidak bekerja dengan baik.

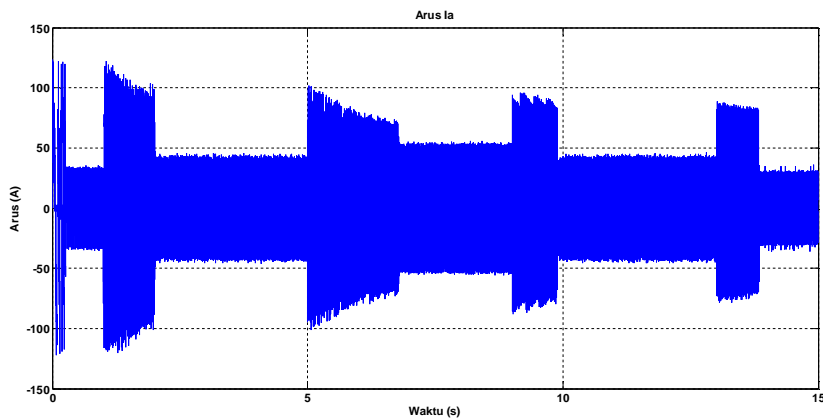
5.4.2. Simulasi dengan Kecepatan Referensi Berubah dan Tanpa Beban

Simulasi dengan kecepatan referensi berubah, dilakukan untuk mengetahui respon sistem kontrol yang telah didesain dapat mengikuti sesuai dengan nilai referensi yang diberikan. Kecepatan referensi diatur dari nilai 0 sampai 3500 rpm. Gambar 5.14 menunjukkan hasil respon kecepatan dengan kecepatan referensi berubah.

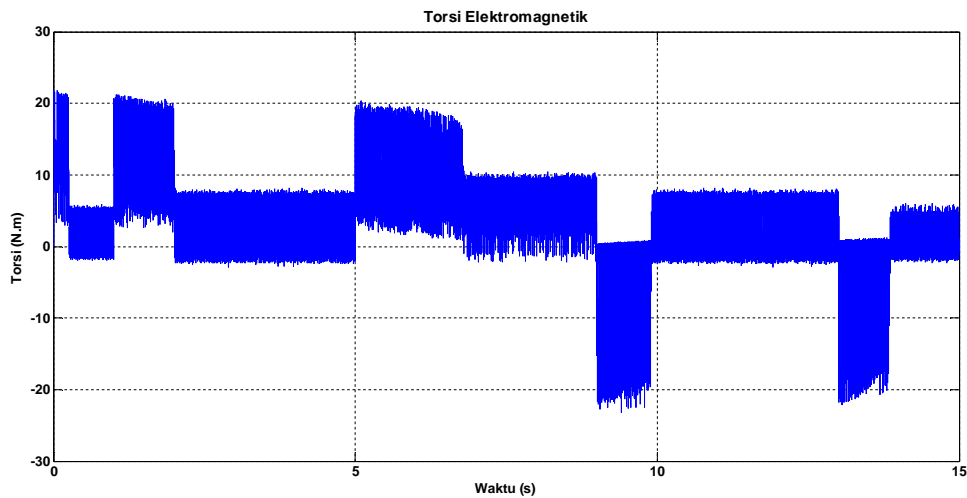


Gambar 5.14. Respon kecepatan dengan kecepatan referensi berubah

Berdasarkan simulasi yang telah dilakukan, untuk proses tracking kecepatan, kontroler ANFIS mampu mengikuti sesuai dengan pola kecepatan referensi yang diberikan dan memiliki rata-rata error *steady state* sebesar . Error steady state dari kontroler ANFIS memiliki nilai lebih kecil daripada Fuzzy-PID, dengan rata-rata sebesar .



Gambar 5.15. Arus I_a saat terjadi perubahan kecepatan

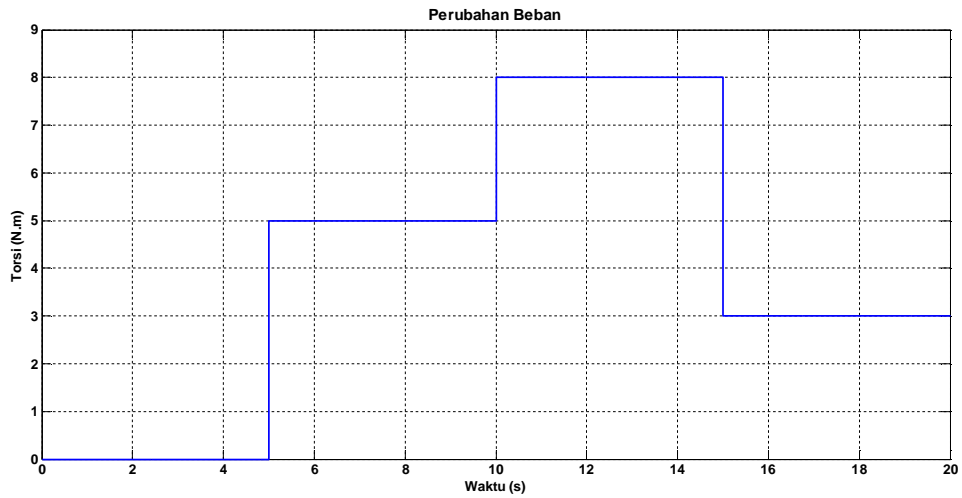


Gambar 5.16. Torsi elektromagnetik saat terjadi perubahan kecepatan

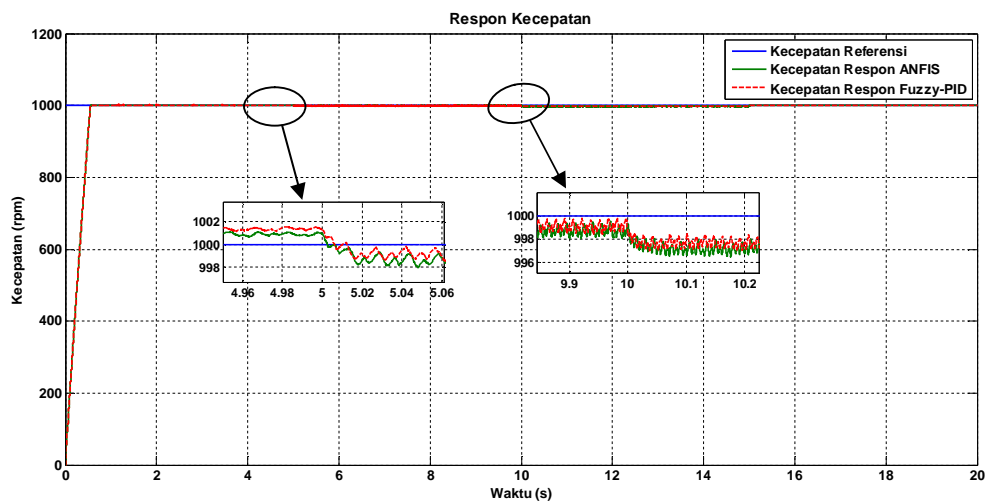
Kondisi arus dan torsi elektromagnetik saat terjadi perubahan kecepatan referensi, masing-masing ditunjukkan oleh gambar 5.15 dan 5.16. berdasarkan simulasi yang dilakukan, terjadi lonjakan arus pada stator, saat terjadi perubahan kecepatan, baik kenaikan kecepatan maupun penurunan kecepatan. Hal ini disebabkan karena pada saat terjadinya percepatan, diperlukan torsi yang lebih besar, sehingga sumber tegangan V_{dc} memperbesar suplai arus pada stator. Ketika terjadi perlambatan, terjadi pengereman secara regeneratif, dimana arus yang tersimpan pada belitan stator, dialirkan kembali ke sistem, hal ini ditandai dengan adanya torsi yang bernilai negatif sebagaimana yang ditunjukkan oleh gambar 5.16.

5.4.3. Simulasi dengan Kecepatan Referensi Tetap dan Beban Berubah

Simulasi dengan kecepatan referensi tetap dengan perubahan beban pada waktu tertentu, merepresentasikan perubahan beban yang dinamis pada kendaraan listrik. Perubahan beban yang terjadi pada simulasi ini ditunjukkan oleh gambar 5.17. Selanjutnya sistem kontrol diuji apakah mampu mengembalikan kondisi sistem sesuai dengan kecepatan referensi yang diberikan.



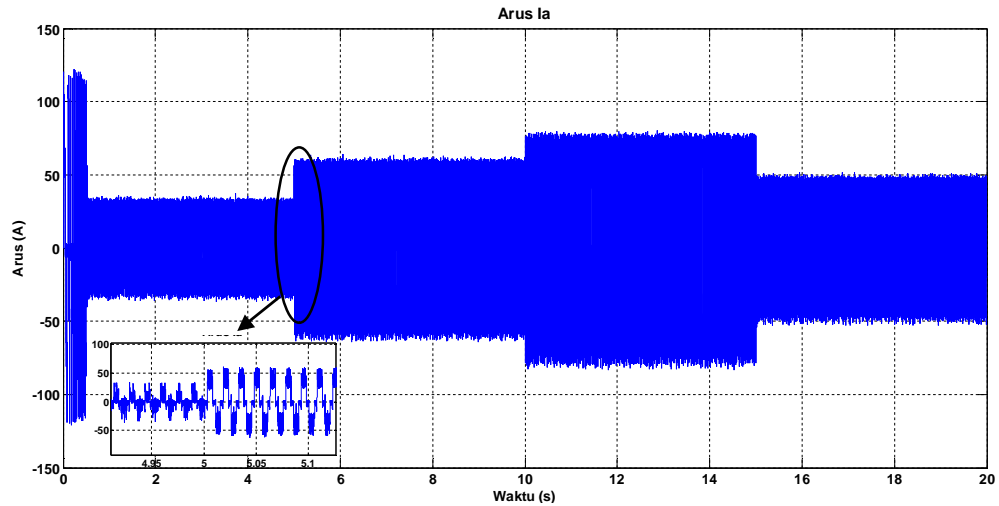
Gambar 5.17. Perubahan beban terhadap waktu



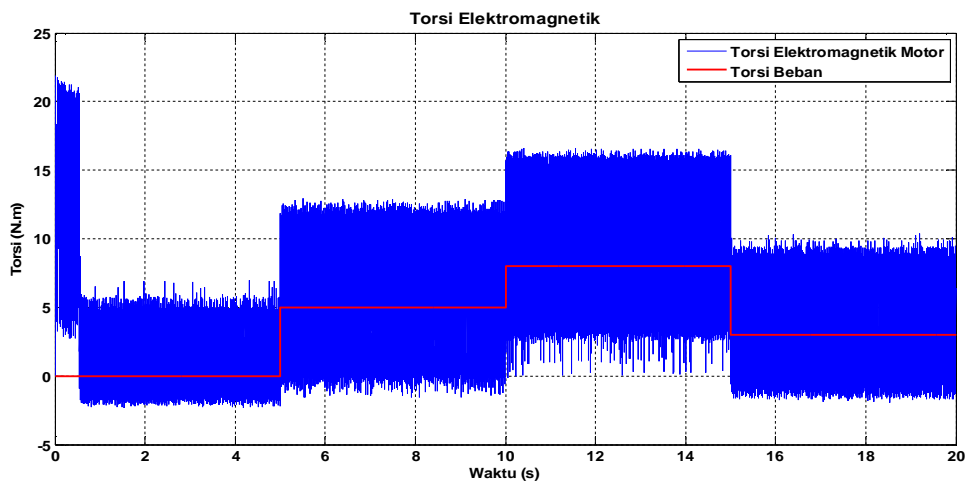
Gambar 5.18. Respon kecepatan terhadap perubahan beban

Gambar 5.18 menunjukkan respon kecepatan saat terjadi perubahan beban. Pada saat detik ke-5, sistem diberi beban sebesar 5 N.m, dan pada detik ke-8 beban dinaikkan menjadi 8 N.m. Dari kedua kasus tersebut terlihat bahwa pada saat beban dinaikkan, respon kecepatan turun, namun, penurunan kecepatan masih dalam toleransi nilai *error*, dengan nilai *error* sebesar 0,1 %. Penurunan kecepatan pada saat terjadi perubahan beban, tidak merubah nilai kecepatan secara signifikan, hal ini karena pada sistem kontrol yang dirancang, selain mempertimbangkan nilai kecepatan, juga, mempertimbangkan nilai torsi yang

diperlukan oleh sistem. Adapun arus dan torsi elektromagnetik saat pada kondisi beban berubah, masing-masing ditunjukkan oleh gambar 5.19 dan 5.20.



Gambar 5.19. Arus I_a pada kondisi beban berubah



Gambar 5.20. Torsi elektromagnetik pada kondisi beban berubah

Berdasarkan hasil simulasi yang ditunjukkan oleh gambar 5.19 dan 5.20, ketika terjadi perubahan beban, sistem dapat langsung merespon kebutuhan torsi dengan baik. Hal ini terlihat pada perubahan dari torsi elektromagnetik dan arus pada stator pada saat terjadi perubahan beban.

5.5. Pengujian Sensor Arus

Setelah dilakukan proses kalibrasi, hasil dari nilai kalibrasi selanjutnya diuji dan dibandingkan dengan alat ukur. Pengujian ini dilakukan untuk menguji hasil kalibrasi yang telah dilakukan. Pengujian dilakukan dengan mengukur arus pada sebuah sumber tegangan yang diberi beban resistif berubah-ubah. Hasil dari pengujian sensor arus ditunjukkan oleh tabel 5.3.

Tabel 5.3. Hasil pengujian sensor arus

Alat Ukur (A)	Sensor Arus (A)	Error (%)
5,6	5,327	4,875
17,2	16,18	5,930
27,9	27,601	1,072
33,6	32,717	2,628
39	38,164	2,144
Rata-rata		3,330

Berdasarkan hasil pengujian sensor arus yang dilakukan, prosentase error pengukuran arus rata-rata sebesar 3,33 %.

5.6. Pengujian Algoritma ANFIS pada Mikrokontroler

Pengujian ANFIS pada mikrokontroler dilakukan untuk mengetahui waktu eksekusi algoritma ANFIS yang telah dibuat., sehingga dapat menjadi acuan dalam pengembangan sistem yang berbasis *real time* untuk aplikasi pengaturan kecepatan motor BLDC untuk kendaraan listrik selanjutnya. Perbandingan sistem kontrol menggunakan Fuzzy-PID dan ANFIS dapat dilihat pada tabel 5.4.

Tabel 5.4. Perbandingan performa sistem kontrol Fuzzy-PID dan ANFIS

Pembanding	Fuzzy-PID	ANFIS
Risetime (4000 rpm)	2,768 s	2,7437 s
Error steady state (rata-rata)	0,2 %	0,1 %
Jumlah variabel	135	26
Waktu eksekusi	204 us	94 us

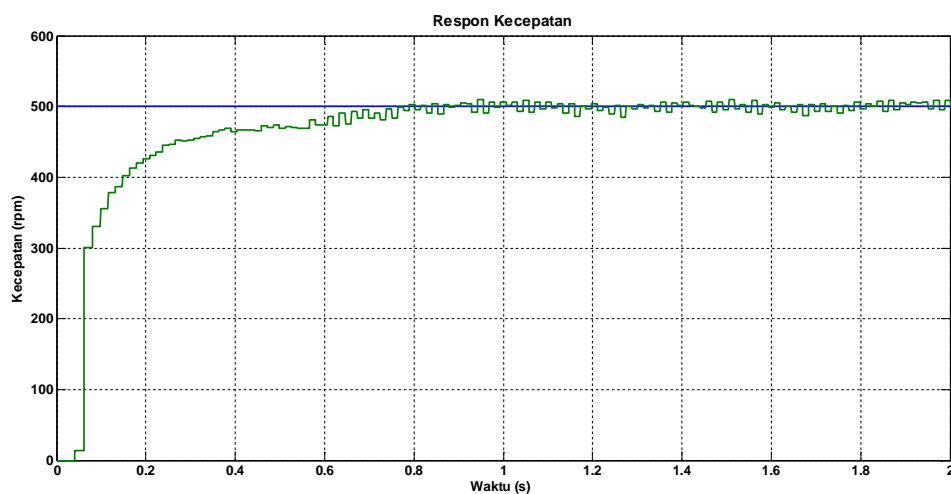
Berdasarkan hasil perbandingan pada tabel 5.3, ANFIS dan Fuzzy-PID mempunyai respon yang tidak jauh berbeda pada sistem. Akan tetapi, dalam implementasi pada *hardware*, ANFIS jauh lebih unggul dalam hal kecepatan eksekusi dan penggunaan *memory* pada mikrokontroler.

5.7. Pengujian Pengaturan Kecepatan Motor BLDC dengan ANFIS

Nilai-nilai parameter yang didapatkan dari hasil simulasi, selanjutnya diprogramkan ke mikrokontroler berbasis 32-bit (STM32F407) dan akan dilakukan pengujian sistem kontrol yang telah didesain pada sistem real. Sehingga didapatkan performa dari sistem kontrol pada sepeda motor listrik. Diantara pengujian yang dilakukan pada penelitian ini adalah dengan memberikan input berupa kecepatan referensi tertentu tanpa pembebanan, pengujian dengan kecepatan referensi tetap dengan pembebanan dan pengujian dengan *tracking* kecepatan referesi.

5.7.1. Pengujian dengan Kecepatan Referensi Tetap dan Tanpa Beban

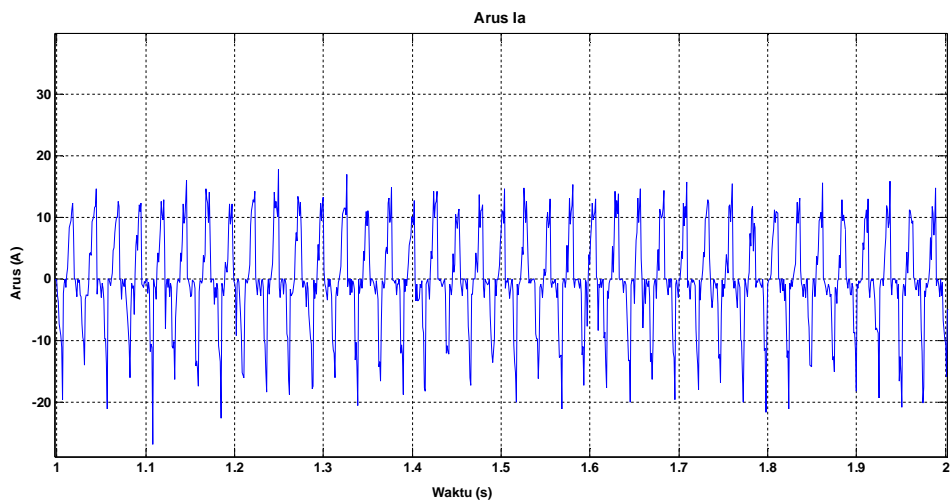
Setelah desain sistem kontrol diimplementasikan, selanjutnya diuji pada sistem real. Pengujian dengan kecepatan referensi tetap dilakukan dengan memberikan nilai referensi sebesar 500 rpm. Hasil respon kecepatan dari percobaan yang telah dilakukan ditunjukkan oleh gambar 5.21.



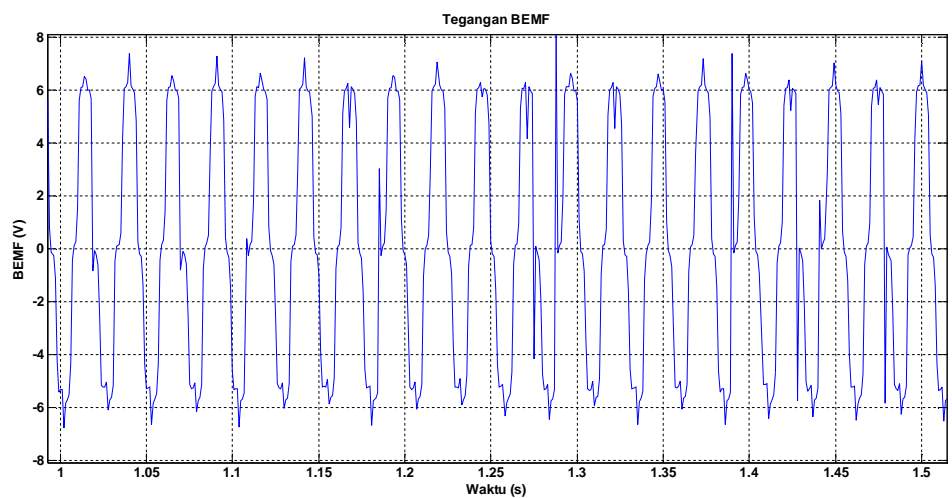
Gambar 5.21. Respon kecepatan pada kecepatan referensi 500 rpm

Berdasarkan percobaan yang telah dilakukan, sistem kontrol yang telah didesain mampu mengontrol kecepatan motor BLDC dengan baik. Namun, kekurangan pada sistem yang dirancang adalah pengukuran yang hanya mengandalkan identifikasi dari satu *hall effect sensor*, sehingga pengukuran kecepatan kurang maksimal, terutama pada kondisi kecepatan rendah. Perlu pengembangan lebih lanjut untuk mengintegrasikan ketiga *hall effect sensor* dalam hal pengukuran kecepatan.

Arus pada stator dan bentuk gelombang BEMF masing-masing ditunjukkan oleh gambar 5.22 dan 5.23.



Gambar 5.22. Arus I_a saat *steady state*



Gambar 5.23. Tegangan BEMF saat *steady state*

Berdasarkan hasil percobaan yang telah dilakukan, terlihat bahwa nilai puncak arus yang terukur sangat fluktuatif. Terlihat bahwa arus mencapai nilai 26,924 A negatif, dan 17,73 A positif dengan nilai RMS sebesar 7,59 A. Pada *hardware* yang didesain pada penelitian ini digunakan *low pass filter* pasif untuk mengurangi terjadinya fluktuasi pada pengukuran arus, sehingga untuk pengembangan selanjutnya, diperlukan desain filter yang lebih baik pada *hardware*. Sedangkan tegangan BEMF dapat dideteksi dengan baik.

Halaman ini sengaja dikosongkan

BAB 6 KESIMPULAN DAN SARAN

6.1. Kesimpulan

Metode *hybrid* PSO dan RLSE dapat menjadi salah satu metode untuk melatih sistem ANFIS multi output. Dari hasil yang didapatkan pada penelitian ini, dengan menggunakan Fuzzy-PID sebagai supervisor pembelajaran, metode ANFIS berbasis PSO-RLSE mampu mempelajari pola yang telah ditentukan dengan RMSE sebesar 0,05364 dengan nilai $\lambda = 1$. Nilai λ sangat berpengaruh terhadap proses pembelajaran pada ANFIS. Semakin kecil nilai λ akan dapat membuat proses pembelajaran menjadi tidak konvergen.

Sistem kontrol kecepatan motor BLDC menggunakan ANFIS telah dirancang dengan baik. Hasil menunjukkan bahwa sistem kontrol kecepatan dengan menggunakan ANFIS berbasis DTC (*Direct Torque Control*) mampu mempertahankan nilai kecepatan dengan mempertimbangkan torsi yang diperlukan oleh sistem. Rata-rata error *steady state* yang dicapai dengan menggunakan ANFIS adalah sebesar 0,1 % dengan *rise time* sebesar 2,7437 s untuk kecepatan referensi sebesar 4000 rpm. Pada saat terjadi peningkatan beban, performa kecepatan menjadi turun, namun, nilai penurunan ini tidak terlalu signifikan dan masih dapat ditoleransi dengan menggunakan sistem kontrol kecepatan yang telah dirancang. Sistem kontrol yang telah dirancang telah diimplementasikan dan bekerja dengan baik untuk kecepatan referensi sebesar 500 rpm. Akan tetapi, *hardware* yang telah dirancang masih memerlukan penyempurnaan dan pengkajian lebih lanjut.

Berdasarkan simulasi dan implementasi yang telah dilakukan, sistem kontrol menggunakan ANFIS jauh lebih baik dibandingkan dengan Fuzzy-PID. Dengan performa yang sama, ANFIS mempunyai keunggulan yaitu variabel yang lebih sedikit daripada Fuzzy-PID, sehingga proses eksekusi pada mikrokontroler menjadi lebih cepat dan membutuhkan memori penyimpanan yang lebih sedikit. Dengan demikian, ANFIS dapat diimplementasikan pada sistem *real time* yang mana proses eksekusi menjadi suatu prioritas yang diutamakan.

6.2. Saran

Pada penelitian ini kesulitan yang dihadapi adalah dalam perancangan *hardware*, maka diperlukan analisis lebih lanjut terhadap motor yang digunakan sehingga *hardware* dapat dirancang dengan lebih baik, terutama pada rangkaian proteksi antara sisi logik dan sisi tegangan tinggi akibat *spike* yang ditimbulkan oleh beban induktif.

Saran untuk pengembangan sistem kontrol, khususnya pada sistem tertanam (*embedded system*), yaitu perlu dilakukan optimasi pada *compiler* (-O3 -time) untuk menghasilkan proses perhitungan yang cepat pada mikrokontroler. Hal ini telah diimplementasikan pada penelitian yang telah dilakukan, sehingga menghasilkan proses eksekusi ANFIS dengan sangat cepat.

DAFTAR PUSTAKA

- [1] Hong-xing Wu, Shu-kang Cheng, and Shu-mei Cui, “A Controller of Brushless DC Motor for Electric Vehicle”, IEEE Trans. On Magnetics, vol. 41, no. 1, Jan. 2005
- [2] B. K. Bose, “Energy, Environment, And Advances in Power Electronics”, IEEE Trans. Power Electron., vol. 15, no. 4, pp. 688–701, Jul. 2000
- [3] B. K. Lee, M. Ehsani, “Advanced BLDC Motor Drive for Low Cost and High Performance Propulsion System in Electric and Hybrid Vehicles”, IEEE International Electric Machines and Drives Conference (IEMDC), Jun 2001.
- [4] Bapayya Naidu Kommula, Venkata Reddy Kota, “Performance Evaluation of Hybrid Fuzzy PI Speed Controller for Brushless DC Motor for Electric Vehicle Application”, IEEE Conference on Power, Control, Communication and Computational Technologies for Sustainable Growth (PCCCTSG), Dec. 2015
- [5] Madhusudan Singh, Archana Garg, “Performance Evaluation of BLDC Motor with Conventional PI and Fuzzy Speed Controller”, IEEE 5th India International Conference on Power Electronics (IICPE), Dec. 2012
- [6] Merve Yildirim, Mehmet Polat, Hasan Kürüm, “A Survey On Comparison Of Electric Motor Types And Drives Used For Electric Vehicles”, IEEE 16th International Power Electronics and Motion Control Conference and Exposition, Turkey, Sept. 2014
- [7] R. Somanatham, P. V. N. Prasad, A. D. Rajkumar, “Modeling and Simulation of Sensorless Control of PMBLDC Motor Using Zero-Crossing Back E.M.F Detection”, IEEE International Symposium on Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM), May 2006
- [8] Kiam Heong Ang, Gregory Chong, and Yun Li “Patents, Software, and Hardware for PID Control: An Overview and Analysis of The Current Art”, IEEE Control Systems Magazine, Feb 2006
- [9] Kiam Heong Ang, Gregory Chong, and Yun Li “PID Control System Analysis, Design, and Technology”, IEEE Transactions on Control Systems Technology, vol. 13, no. 4, Jul 2005

- [10] Stanisław Skoczowski, Stefan Domek, Krzysztof Pietruszewicz, and Bogdan Broel-Plater, “A Method for Improving the Robustness of PID Control”, IEEE Transactions On Industrial Electronics, vol. 52, no. 6, Dec 2005
- [11] B.M. Mohan, Arpita Sinha, “Analytical structure and stability analysis of a fuzzy PID controller”, Elsevier, Applied Soft Computing, vol. 8, Jan 2008
- [12] Ahmed Rubaai, Marcel J. Castro-Sitiriche, and Abdul R. Ofoli, “Design and Implementation of Parallel Fuzzy PID Controller for High-Performance Brushless Motor Drives: An Integrated Environment for Rapid Control Prototyping”, IEEE Transactions on Industry Applications, vol. 44, no. 4, Jul/Aug 2008
- [13] R. Arulmozhiyal, R.Kandiban, “Design of Fuzzy PID Controller for Brushless DC Motor”, IEEE International Conference on Computer Communication and Informatics (ICCCI), Jan 2012
- [14] Dr. R. Arulmozhiyal, Salem, “Design and Implementation of Fuzzy PID Controller for BLDC Motor using FPGA”, IEEE International Conference on Power Electronics, Drives and Energy Systems, Dec 2012
- [15] Wang Yuanxi, Yu Yali, Zhang Guosheng, Sheng Xiaoliang, “Fuzzy Auto-adjust PID Controller Design of Brushless DC Motor”, Elsevier, International Conference on Medical Physics and Biomedical Engineering (ICMPBE), vol. 33, 2012
- [16] Guoqiang Cheng, “Brushless DC Motor Speed Control System Based on Fuzzy PID Controller”, Springer, Second International Conference on Network Computing and Information Security (NCIS), Dec 2012
- [17] Ahmed Rubaai, Paul Young, “Hardware/Software Implementation of Fuzzy-Neural-Network Self-Learning Control Methods for Brushless DC Motor Drives”, IEEE Transactions on Industry Applications, vol. 52, 2016
- [18] Stefan Badurson. “BLDC Motor Modelling and Control – A Matlab/Simulink Implementation”. Master thesis work, International master program in Electrical Power Engineering, Chalmer Tekniska Hogskola, 2005.
- [19] Chang-liang Xia, “Permanent Magnet Brushless DC Motor Drives and Controls”, Science Press, 2012.

- [20] I. Takahashi, T. Noguchi. “A New Quick-Response and High Efficiency Control Strategy of An Induction Machine,” IEEE Transactions on Industry Application, Vol 22, 1986.
- [21] Sariati Binti Dalib, “The Simulation of The Direct Torque Control of Permanent Magnet Synchronous Motor”, Project Report, University of Technology Malaysia, May, 2007.
- [22] Mourad Masmoudi, Bassem El Badsy, Ahmed Masmoudi, “Direct Torque Control of Brushless DC Motor Drives With Improved Reliability”, IEEE Transactions on Industry Applications, Vol. 50, No. 6, November/December 2014.
- [23] J. Kennedy, “The particle swarm: Social Adaptation of Knowledge,” in Proc. IEEE Int. Conf. Evol. Comput., pp. 303–308, Apr. 1997.
- [24] H.E.A. Ibrahim, F.N. Hassan, Anas O. Shomer, “Optimal PID control of a brushless DC motor using PSO and BF techniques”, Ain Shams Engineering Journal 5, 391–398, 2014.
- [25] Mehdi Nasri, Hossein Nezamabadi-pour, and Malihe Maghfoori, “A PSO-Based Optimum Design of PID Controller for A Linear Brushless DC Motor”, International Science Index, Control and Information Engineering, Vol. 1 No. 2, 2007.
- [26] J. C. Bansal, et al, “Inertia Weight Strategies in Particle Swarm Optimization”, IEEE, Nature and Biologically Inspired Computing (NaBIC), 2011.
- [27] Bengt Carlsson, “Recursive Identification”, Uppsala Universitet, Institutionen för Informations Teknologi, Sept, 2011.
- [28] STMicroelectronic, “Datasheet – Production Data STM32F405xx STM32F407xx”, 2016.
- [28] International Rectifier, “Datasheet IRFB4115PbF HEXFET Power MOSFET”, 2014.
- [30] International Rectifier, “Datasheet IR2110(-1-2)(S)PbF/IR2113(-1-2)(S)PbF”, Data Sheet No. PD60147 rev.U, 2005.
- [31] Tamura, “Datasheet Hall Effect Current Sensor L01Z***S05 Series”, Rev D, 2012.

Halaman ini sengaja dikosongkan

LAMPIRAN I
DATA TRAINING ANFIS MULTI-OUTPUT

Data ke-	<i>error</i>	<i>Δerror</i>	FKp	FKi	FKd
1	-500	-500	0,4	0,004	0,0276
2	-500	-447,368	0,417619	0,004176	0,027494
3	-500	-394,737	0,455329	0,004553	0,027268
4	-500	-342,105	0,444508	0,004445	0,027333
5	-500	-289,474	0,410887	0,004109	0,027535
6	-500	-236,842	0,534562	0,005346	0,027588
7	-500	-184,211	0,880292	0,008803	0,027447
8	-500	-131,579	1,07091	0,010709	0,027198
9	-500	-78,9474	1,183257	0,011833	0,027393
10	-500	-26,3158	1,240407	0,012404	0,027566
11	-500	26,31579	1,423897	0,014239	0,026133
12	-500	78,94737	1,685626	0,016856	0,024372
13	-500	131,5789	1,901335	0,019013	0,02337
14	-500	184,2105	2,123051	0,021231	0,022786
15	-500	236,8421	2,407827	0,024078	0,022519
16	-500	289,4737	2,747528	0,027475	0,0225
17	-500	342,1053	2,991544	0,029915	0,0225
18	-500	394,7368	3,204368	0,032044	0,0225
19	-500	447,3684	3,435356	0,034354	0,0225
20	-500	500	3,75	0,0375	0,0225
21	-447,368	-500	0,417619	0,004176	0,027494
22	-447,368	-447,368	0,813816	0,008138	0,027494
23	-447,368	-394,737	0,883245	0,008832	0,027268
24	-447,368	-342,105	0,8616	0,008616	0,027333
25	-447,368	-289,474	0,813816	0,008138	0,027494
26	-447,368	-236,842	0,972009	0,00972	0,026673
27	-447,368	-184,211	1,346605	0,013466	0,025037
28	-447,368	-131,579	1,444644	0,014446	0,024552
29	-447,368	-78,9474	1,50967	0,015097	0,024942
30	-447,368	-26,3158	1,554402	0,015544	0,025117
31	-447,368	26,31579	1,791736	0,017917	0,025117
32	-447,368	78,94737	2,062991	0,02063	0,024372
33	-447,368	131,5789	2,239009	0,02239	0,02337
34	-447,368	184,2105	2,440847	0,024408	0,022786
35	-447,368	236,8421	2,721666	0,027217	0,02252
36	-447,368	289,4737	2,814644	0,02843	0,020989
37	-447,368	342,1053	2,991544	0,030323	0,020452
38	-447,368	394,7368	3,204368	0,032404	0,020369
39	-447,368	447,3684	3,435356	0,034686	0,020612
40	-447,368	500	3,75	0,037818	0,020612

Data ke-	<i>error</i>	<i>Δerror</i>	FKp	FKi	FKd
41	-394,737	-500	0,455329	0,004553	0,027268
42	-394,737	-447,368	0,883245	0,008832	0,027268
43	-394,737	-394,737	1,032136	0,010321	0,027268
44	-394,737	-342,105	1,032136	0,010321	0,027268
45	-394,737	-289,474	1,032136	0,010321	0,027268
46	-394,737	-236,842	1,152842	0,011528	0,02635
47	-394,737	-184,211	1,490358	0,014904	0,024381
48	-394,737	-131,579	1,653931	0,016539	0,02367
49	-394,737	-78,9474	1,722899	0,017229	0,023807
50	-394,737	-26,3158	1,784577	0,017846	0,023807
51	-394,737	26,31579	1,991648	0,019916	0,023807
52	-394,737	78,94737	2,278018	0,02278	0,023807
53	-394,737	131,5789	2,458466	0,024585	0,02337
54	-394,737	184,2105	2,651691	0,026517	0,022822
55	-394,737	236,8421	2,94883	0,029488	0,022524
56	-394,737	289,4737	3,045632	0,030693	0,020784
57	-394,737	342,1053	3,045632	0,031495	0,019447
58	-394,737	394,7368	3,204368	0,033231	0,019226
59	-394,737	447,3684	3,394823	0,035192	0,019226
60	-394,737	500	3,75	0,038653	0,019226
61	-342,105	-500	0,444508	0,004445	0,027333
62	-342,105	-447,368	0,8616	0,008616	0,027333
63	-342,105	-394,737	1,032136	0,010321	0,027268
64	-342,105	-342,105	1,160735	0,011607	0,027333
65	-342,105	-289,474	1,160735	0,011607	0,027333
66	-342,105	-236,842	1,262988	0,01263	0,026448
67	-342,105	-184,211	1,554394	0,015544	0,024511
68	-342,105	-131,579	1,751288	0,017513	0,023575
69	-342,105	-78,9474	1,925815	0,019258	0,023036
70	-342,105	-26,3158	1,99596	0,01996	0,023036
71	-342,105	26,31579	2,18933	0,021893	0,023036
72	-342,105	78,94737	2,450843	0,024508	0,023036
73	-342,105	131,5789	2,628462	0,026285	0,023091
74	-342,105	184,2105	2,824184	0,028242	0,022807
75	-342,105	236,8421	3,152113	0,031521	0,022522
76	-342,105	289,4737	3,258456	0,032797	0,020854
77	-342,105	342,1053	3,258456	0,033509	0,019551
78	-342,105	394,7368	3,241248	0,034019	0,018274
79	-342,105	447,3684	3,408601	0,035996	0,017949
80	-342,105	500	3,75	0,040118	0,017949

Data ke-	<i>error</i>	<i>Δerror</i>	FKp	FKi	FKd
81	-289,474	-500	0,410887	0,004109	0,027535
82	-289,474	-447,368	0,813816	0,008138	0,027494
83	-289,474	-394,737	1,032136	0,010321	0,027268
84	-289,474	-342,105	1,160735	0,011607	0,027333
85	-289,474	-289,474	1,231018	0,01231	0,027535
86	-289,474	-236,842	1,324699	0,013247	0,026726
87	-289,474	-184,211	1,606515	0,016065	0,024718
88	-289,474	-131,579	1,826383	0,018264	0,023575
89	-289,474	-78,9474	2,038749	0,020387	0,0229
90	-289,474	-26,3158	2,23721	0,022372	0,022614
91	-289,474	26,31579	2,427141	0,024271	0,022614
92	-289,474	78,94737	2,681439	0,026814	0,022623
93	-289,474	131,5789	2,878935	0,028789	0,022643
94	-289,474	184,2105	3,063544	0,030635	0,022619
95	-289,474	236,8421	3,376367	0,033764	0,02252
96	-289,474	289,4737	3,502472	0,035222	0,021015
97	-289,474	342,1053	3,475587	0,035709	0,019551
98	-289,474	394,7368	3,463918	0,036536	0,018274
99	-289,474	447,3684	3,498109	0,037809	0,016888
100	-289,474	500	3,75	0,042636	0,016485
101	-236,842	-500	0,534562	0,007684	0,027588
102	-236,842	-447,368	0,972009	0,010614	0,026673
103	-236,842	-394,737	1,152842	0,012284	0,02635
104	-236,842	-342,105	1,262988	0,013313	0,026448
105	-236,842	-289,474	1,324699	0,013897	0,026726
106	-236,842	-236,842	1,485235	0,014935	0,026793
107	-236,842	-184,211	1,741452	0,017482	0,024718
108	-236,842	-131,579	1,949314	0,019554	0,023575
109	-236,842	-78,9474	2,158706	0,021645	0,0229
110	-236,842	-26,3158	2,418466	0,024243	0,022558
111	-236,842	26,31579	2,673897	0,026794	0,021943
112	-236,842	78,94737	2,935626	0,0294	0,021891
113	-236,842	131,5789	3,151335	0,031552	0,021799
114	-236,842	184,2105	3,373051	0,033766	0,02191
115	-236,842	236,8421	3,657827	0,036611	0,021947
116	-236,842	289,4737	3,753312	0,03769	0,020258
117	-236,842	342,1053	3,753736	0,038393	0,018913
118	-236,842	394,7368	3,753925	0,039679	0,017693
119	-236,842	447,3684	3,75338	0,041862	0,01633
120	-236,842	500	3,75324	0,04598	0,014447

Data ke-	<i>error</i>	<i>Δerror</i>	FKp	FKi	FKd
121	-184,211	-500	0,880292	0,017484	0,027447
122	-184,211	-447,368	1,346605	0,017471	0,025037
123	-184,211	-394,737	1,490358	0,017767	0,024381
124	-184,211	-342,105	1,554394	0,01819	0,024511
125	-184,211	-289,474	1,606515	0,018709	0,024718
126	-184,211	-236,842	1,741452	0,019517	0,024718
127	-184,211	-184,211	2,093522	0,021688	0,024718
128	-184,211	-131,579	2,259508	0,023278	0,023575
129	-184,211	-78,9474	2,446054	0,02511	0,0229
130	-184,211	-26,3158	2,699945	0,027136	0,022561
131	-184,211	26,31579	2,876949	0,028906	0,021398
132	-184,211	78,94737	2,935626	0,029985	0,02018
133	-184,211	131,5789	3,151335	0,032059	0,019901
134	-184,211	184,2105	3,373051	0,034229	0,020238
135	-184,211	236,8421	3,651589	0,037007	0,020238
136	-184,211	289,4737	3,769877	0,037977	0,018381
137	-184,211	342,1053	3,801248	0,038393	0,016945
138	-184,211	394,7368	3,803725	0,039679	0,01591
139	-184,211	447,3684	3,797744	0,041729	0,014645
140	-184,211	500	3,797744	0,045745	0,012738
141	-131,579	-500	1,07091	0,024217	0,027198
142	-131,579	-447,368	1,444644	0,022758	0,024552
143	-131,579	-394,737	1,653931	0,02135	0,02367
144	-131,579	-342,105	1,751288	0,021743	0,023575
145	-131,579	-289,474	1,826383	0,022778	0,023575
146	-131,579	-236,842	1,949314	0,023461	0,023575
147	-131,579	-184,211	2,259508	0,02492	0,023575
148	-131,579	-131,579	2,461812	0,026174	0,023575
149	-131,579	-78,9474	2,639375	0,027173	0,022961
150	-131,579	-26,3158	2,911692	0,029235	0,022573
151	-131,579	26,31579	3,098665	0,031103	0,021204
152	-131,579	78,94737	3,098665	0,031756	0,019588
153	-131,579	131,5789	3,151335	0,032995	0,018908
154	-131,579	184,2105	3,316813	0,034745	0,018908
155	-131,579	236,8421	3,633174	0,037852	0,018908
156	-131,579	289,4737	3,77377	0,038951	0,017274
157	-131,579	342,1053	3,84856	0,038951	0,015771
158	-131,579	394,7368	3,895082	0,03945	0,014751
159	-131,579	447,3684	3,895082	0,040919	0,013434
160	-131,579	500	3,895082	0,045331	0,011408

Data ke-	<i>error</i>	<i>Δerror</i>	FKp	FKi	FKd
161	-78,9474	-500	1,183257	0,030692	0,027393
162	-78,9474	-447,368	1,50967	0,027741	0,024942
163	-78,9474	-394,737	1,722899	0,025649	0,023807
164	-78,9474	-342,105	1,925815	0,026115	0,023036
165	-78,9474	-289,474	2,038749	0,02752	0,0229
166	-78,9474	-236,842	2,158706	0,028121	0,0229
167	-78,9474	-184,211	2,446054	0,028774	0,0229
168	-78,9474	-131,579	2,639375	0,028867	0,022961
169	-78,9474	-78,9474	2,800662	0,028745	0,0229
170	-78,9474	-26,3158	3,10552	0,031165	0,022563
171	-78,9474	26,31579	3,314374	0,03325	0,021365
172	-78,9474	78,94737	3,314374	0,033838	0,019886
173	-78,9474	131,5789	3,264586	0,034355	0,018592
174	-78,9474	184,2105	3,363255	0,035518	0,017614
175	-78,9474	236,8421	3,648467	0,03926	0,017614
176	-78,9474	289,4737	3,770533	0,04062	0,016089
177	-78,9474	342,1053	3,839265	0,040435	0,014705
178	-78,9474	394,7368	3,967864	0,040221	0,013668
179	-78,9474	447,3684	4,062007	0,041571	0,012378
180	-78,9474	500	4,062007	0,045655	0,010114
181	-26,3158	-500	1,240407	0,039294	0,027566
182	-26,3158	-447,368	1,554402	0,035843	0,025117
183	-26,3158	-394,737	1,784577	0,034468	0,023807
184	-26,3158	-342,105	1,99596	0,034911	0,023036
185	-26,3158	-289,474	2,23721	0,036043	0,022614
186	-26,3158	-236,842	2,418466	0,036195	0,022558
187	-26,3158	-184,211	2,699945	0,035012	0,022561
188	-26,3158	-131,579	2,911692	0,034213	0,022573
189	-26,3158	-78,9474	3,10552	0,033659	0,022563
190	-26,3158	-26,3158	3,323004	0,033337	0,022558
191	-26,3158	26,31579	3,576103	0,03586	0,021457
192	-26,3158	78,94737	3,560809	0,03631	0,019886
193	-26,3158	131,5789	3,534017	0,037151	0,018592
194	-26,3158	184,2105	3,566332	0,038496	0,017262
195	-26,3158	236,8421	3,657089	0,041635	0,016043
196	-26,3158	289,4737	3,768982	0,04348	0,014563
197	-26,3158	342,1053	3,839265	0,042947	0,013136
198	-26,3158	394,7368	3,967864	0,042752	0,01195
199	-26,3158	447,3684	4,186184	0,043381	0,01075
200	-26,3158	500	4,355428	0,045943	0,008543

Data ke-	<i>error</i>	<i>Δerror</i>	FKp	FKi	FKd
201	26,31579	-500	1,423897	0,045943	0,026133
202	26,31579	-447,368	1,791736	0,043381	0,025117
203	26,31579	-394,737	1,991648	0,042752	0,023807
204	26,31579	-342,105	2,18933	0,042947	0,023036
205	26,31579	-289,474	2,427141	0,04348	0,022614
206	26,31579	-236,842	2,673897	0,041635	0,021943
207	26,31579	-184,211	2,876949	0,038496	0,021398
208	26,31579	-131,579	3,098665	0,037151	0,021204
209	26,31579	-78,9474	3,314374	0,03631	0,021365
210	26,31579	-26,3158	3,576103	0,03586	0,021457
211	26,31579	26,31579	3,759593	0,033337	0,019938
212	26,31579	78,94737	3,760527	0,033659	0,018633
213	26,31579	131,5789	3,762203	0,034213	0,01747
214	26,31579	184,2105	3,760188	0,035012	0,0162
215	26,31579	236,8421	3,759593	0,036195	0,014511
216	26,31579	289,4737	3,768982	0,036043	0,013423
217	26,31579	342,1053	3,839265	0,034911	0,011976
218	26,31579	394,7368	3,967864	0,034468	0,010707
219	26,31579	447,3684	4,186184	0,035843	0,009326
220	26,31579	500	4,594336	0,039294	0,007442
221	78,94737	-500	1,685626	0,045655	0,024372
222	78,94737	-447,368	2,062991	0,041571	0,024372
223	78,94737	-394,737	2,278018	0,040221	0,023807
224	78,94737	-342,105	2,450843	0,040435	0,023036
225	78,94737	-289,474	2,681439	0,04062	0,022623
226	78,94737	-236,842	2,935626	0,03926	0,021891
227	78,94737	-184,211	2,935626	0,035518	0,02018
228	78,94737	-131,579	3,098665	0,034355	0,019588
229	78,94737	-78,9474	3,314374	0,033838	0,019886
230	78,94737	-26,3158	3,560809	0,03325	0,019886
231	78,94737	26,31579	3,760527	0,031165	0,018633
232	78,94737	78,94737	3,816743	0,028745	0,016804
233	78,94737	131,5789	3,826752	0,028867	0,015836
234	78,94737	184,2105	3,816743	0,028774	0,014676
235	78,94737	236,8421	3,816743	0,028121	0,012952
236	78,94737	289,4737	3,816743	0,02752	0,012232
237	78,94737	342,1053	3,839265	0,026115	0,011555
238	78,94737	394,7368	3,967864	0,025649	0,010337
239	78,94737	447,3684	4,157081	0,027741	0,009058
240	78,94737	500	4,565462	0,030692	0,0071

Data ke-	<i>error</i>	<i>Δerror</i>	FKp	FKi	FKd
241	131,5789	-500	1,901335	0,045331	0,02337
242	131,5789	-447,368	2,239009	0,040919	0,02337
243	131,5789	-394,737	2,458466	0,03945	0,02337
244	131,5789	-342,105	2,628462	0,038951	0,023091
245	131,5789	-289,474	2,878935	0,038951	0,022643
246	131,5789	-236,842	3,151335	0,037852	0,021799
247	131,5789	-184,211	3,151335	0,034745	0,019901
248	131,5789	-131,579	3,151335	0,032995	0,018908
249	131,5789	-78,9474	3,264586	0,031756	0,018592
250	131,5789	-26,3158	3,534017	0,031103	0,018592
251	131,5789	26,31579	3,762203	0,029235	0,01747
252	131,5789	78,94737	3,826752	0,027173	0,015836
253	131,5789	131,5789	3,92909	0,026174	0,014771
254	131,5789	184,2105	3,92909	0,02492	0,013557
255	131,5789	236,8421	3,92909	0,023461	0,011696
256	131,5789	289,4737	3,92909	0,022778	0,010958
257	131,5789	342,1053	3,92909	0,021743	0,010508
258	131,5789	394,7368	3,944964	0,02135	0,009924
259	131,5789	447,3684	4,091928	0,022758	0,008668
260	131,5789	500	4,533051	0,024217	0,006425
261	184,2105	-500	2,123051	0,045745	0,022786
262	184,2105	-447,368	2,440847	0,041729	0,022786
263	184,2105	-394,737	2,651691	0,039679	0,022822
264	184,2105	-342,105	2,824184	0,038393	0,022807
265	184,2105	-289,474	3,063544	0,037977	0,022619
266	184,2105	-236,842	3,373051	0,037007	0,02191
267	184,2105	-184,211	3,373051	0,034229	0,020238
268	184,2105	-131,579	3,316813	0,032059	0,018908
269	184,2105	-78,9474	3,363255	0,029985	0,017614
270	184,2105	-26,3158	3,566332	0,028906	0,017262
271	184,2105	26,31579	3,760188	0,027136	0,0162
272	184,2105	78,94737	3,816743	0,02511	0,014676
273	184,2105	131,5789	3,92909	0,023278	0,013557
274	184,2105	184,2105	4,119708	0,021688	0,012561
275	184,2105	236,8421	4,119708	0,019517	0,010449
276	184,2105	289,4737	4,119708	0,018709	0,009639
277	184,2105	342,1053	4,085128	0,01819	0,009326
278	184,2105	394,7368	4,063444	0,017767	0,008942
279	184,2105	447,3684	4,172893	0,017471	0,00808
280	184,2105	500	4,574452	0,017484	0,005282

Data ke-	<i>error</i>	<i>Δerror</i>	FKp	FKi	FKd
281	236,8421	-500	2,407827	0,04598	0,022519
282	236,8421	-447,368	2,721666	0,041862	0,02252
283	236,8421	-394,737	2,94883	0,039679	0,022524
284	236,8421	-342,105	3,152113	0,038393	0,022522
285	236,8421	-289,474	3,376367	0,03769	0,02252
286	236,8421	-236,842	3,657827	0,036611	0,021947
287	236,8421	-184,211	3,651589	0,033766	0,020238
288	236,8421	-131,579	3,633174	0,031552	0,018908
289	236,8421	-78,9474	3,648467	0,0294	0,017614
290	236,8421	-26,3158	3,657089	0,026794	0,016043
291	236,8421	26,31579	3,759593	0,024243	0,014511
292	236,8421	78,94737	3,816743	0,021645	0,012952
293	236,8421	131,5789	3,92909	0,019554	0,011696
294	236,8421	184,2105	4,119708	0,017482	0,010449
295	236,8421	236,8421	4,465438	0,014935	0,008911
296	236,8421	289,4737	4,454256	0,013897	0,007948
297	236,8421	342,1053	4,408002	0,013313	0,007578
298	236,8421	394,7368	4,391585	0,012284	0,006917
299	236,8421	447,3684	4,44549	0,010614	0,005832
300	236,8421	500	4,598031	0,007684	0,003207
301	289,4737	-500	2,747528	0,042636	0,0225
302	289,4737	-447,368	2,814644	0,037809	0,020989
303	289,4737	-394,737	3,045632	0,036536	0,020784
304	289,4737	-342,105	3,258456	0,035709	0,020854
305	289,4737	-289,474	3,502472	0,035222	0,021015
306	289,4737	-236,842	3,753312	0,033764	0,020258
307	289,4737	-184,211	3,769877	0,030635	0,018381
308	289,4737	-131,579	3,77377	0,028789	0,017274
309	289,4737	-78,9474	3,770533	0,026814	0,016089
310	289,4737	-26,3158	3,768982	0,024271	0,014563
311	289,4737	26,31579	3,768982	0,022372	0,013423
312	289,4737	78,94737	3,816743	0,020387	0,012232
313	289,4737	131,5789	3,92909	0,018264	0,010958
314	289,4737	184,2105	4,119708	0,016065	0,009639
315	289,4737	236,8421	4,454256	0,013247	0,007948
316	289,4737	289,4737	4,589113	0,01231	0,007386
317	289,4737	342,1053	4,555492	0,011607	0,006964
318	289,4737	394,7368	4,544671	0,010321	0,006193
319	289,4737	447,3684	4,582381	0,008138	0,004883
320	289,4737	500	4,589113	0,004109	0,002465

Data ke-	<i>error</i>	<i>Δerror</i>	FKp	FKi	FKd
321	342,1053	-500	2,991544	0,040118	0,0225
322	342,1053	-447,368	2,991544	0,035996	0,020452
323	342,1053	-394,737	3,045632	0,034019	0,019447
324	342,1053	-342,105	3,258456	0,033509	0,019551
325	342,1053	-289,474	3,475587	0,032797	0,019551
326	342,1053	-236,842	3,753736	0,031521	0,018913
327	342,1053	-184,211	3,801248	0,028242	0,016945
328	342,1053	-131,579	3,84856	0,026285	0,015771
329	342,1053	-78,9474	3,839265	0,024508	0,014705
330	342,1053	-26,3158	3,839265	0,021893	0,013136
331	342,1053	26,31579	3,839265	0,01996	0,011976
332	342,1053	78,94737	3,839265	0,019258	0,011555
333	342,1053	131,5789	3,92909	0,017513	0,010508
334	342,1053	184,2105	4,085128	0,015544	0,009326
335	342,1053	236,8421	4,408002	0,01263	0,007578
336	342,1053	289,4737	4,555492	0,011607	0,006964
337	342,1053	342,1053	4,555492	0,011607	0,006964
338	342,1053	394,7368	4,544671	0,010321	0,006193
339	342,1053	447,3684	4,555492	0,008616	0,00517
340	342,1053	500	4,555492	0,004445	0,002667
341	394,7368	-500	3,204368	0,038653	0,0225
342	394,7368	-447,368	3,204368	0,035192	0,020369
343	394,7368	-394,737	3,204368	0,033231	0,019226
344	394,7368	-342,105	3,241248	0,031495	0,018274
345	394,7368	-289,474	3,463918	0,030693	0,018274
346	394,7368	-236,842	3,753925	0,029488	0,017693
347	394,7368	-184,211	3,803725	0,026517	0,01591
348	394,7368	-131,579	3,895082	0,024585	0,014751
349	394,7368	-78,9474	3,967864	0,02278	0,013668
350	394,7368	-26,3158	3,967864	0,019916	0,01195
351	394,7368	26,31579	3,967864	0,017846	0,010707
352	394,7368	78,94737	3,967864	0,017229	0,010337
353	394,7368	131,5789	3,944964	0,016539	0,009924
354	394,7368	184,2105	4,063444	0,014904	0,008942
355	394,7368	236,8421	4,391585	0,011528	0,006917
356	394,7368	289,4737	4,544671	0,010321	0,006193
357	394,7368	342,1053	4,544671	0,010321	0,006193
358	394,7368	394,7368	4,544671	0,010321	0,006193
359	394,7368	447,3684	4,544671	0,008832	0,005299
360	394,7368	500	4,544671	0,004553	0,002732

Data ke-	<i>error</i>	<i>Δerror</i>	FKp	FKi	FKd
361	447,3684	-500	3,435356	0,037818	0,0225
362	447,3684	-447,368	3,435356	0,034686	0,020612
363	447,3684	-394,737	3,394823	0,032404	0,019226
364	447,3684	-342,105	3,408601	0,030323	0,017949
365	447,3684	-289,474	3,498109	0,02843	0,016888
366	447,3684	-236,842	3,75338	0,027217	0,01633
367	447,3684	-184,211	3,797744	0,024408	0,014645
368	447,3684	-131,579	3,895082	0,02239	0,013434
369	447,3684	-78,9474	4,062007	0,02063	0,012378
370	447,3684	-26,3158	4,186184	0,017917	0,01075
371	447,3684	26,31579	4,186184	0,015544	0,009326
372	447,3684	78,94737	4,157081	0,015097	0,009058
373	447,3684	131,5789	4,091928	0,014446	0,008668
374	447,3684	184,2105	4,172893	0,013466	0,00808
375	447,3684	236,8421	4,44549	0,00972	0,005832
376	447,3684	289,4737	4,582381	0,008138	0,004883
377	447,3684	342,1053	4,555492	0,008616	0,00517
378	447,3684	394,7368	4,544671	0,008832	0,005299
379	447,3684	447,3684	4,582381	0,008138	0,004883
380	447,3684	500	4,582381	0,004176	0,002506
381	500	-500	3,75	0,0375	0,0225
382	500	-447,368	3,75	0,034354	0,020612
383	500	-394,737	3,75	0,032044	0,019226
384	500	-342,105	3,75	0,029915	0,017949
385	500	-289,474	3,75	0,027475	0,016485
386	500	-236,842	3,75324	0,024078	0,014447
387	500	-184,211	3,797744	0,021231	0,012738
388	500	-131,579	3,895082	0,019013	0,011408
389	500	-78,9474	4,062007	0,016856	0,010114
390	500	-26,3158	4,355428	0,014239	0,008543
391	500	26,31579	4,594336	0,012404	0,007442
392	500	78,94737	4,565462	0,011833	0,0071
393	500	131,5789	4,533051	0,010709	0,006425
394	500	184,2105	4,574452	0,008803	0,005282
395	500	236,8421	4,598031	0,005346	0,003207
396	500	289,4737	4,589113	0,004109	0,002465
397	500	342,1053	4,555492	0,004445	0,002667
398	500	394,7368	4,544671	0,004553	0,002732
399	500	447,3684	4,582381	0,004176	0,002506
400	500	500	4,6	0,004	0,0024

BIODATA PENULIS



Mohamad Ridwan, adalah penulis dalam penelitian tesis ini. Penulis dilahirkan di Pasuruan, 19 Februari 1992, seorang anak dari orang tua yang bernama Bapak Misraki dan Ibu Mulyani. Terlahir dari keluarga yang sederhana, di desa Nguling, kecamatan Nguling, kabupaten Pasuruan. Meskipun kehidupan keluarga yang sederhana, namun berbekal tekad yang kuat, pendidikan formal pun ditempuh oleh penulis. Pendidikan formal yang dimulai dari tahun 1996, di T.K R.A. El-Fatimah Nguling, SDN Nguling IV (1997 – 2003), SMPN 1 Nguling (2003 – 2006), dan SMAN 1 Grati (2006 – 2009). Penulis juga aktif dalam kegiatan ekstrakurikuler, Pramuka. Penulis juga pernah menjadi ketua Saka Bhayangkara Kepolisian Sektor Nguling (2008 – 2009). Demi menggapai cita-cita, akhirnya tahun 2009 penulis mulai menempuh pendidikan tinggi di jurusan Teknik Elektro, bidang studi Elektronika, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember. Tahun 2013 penulis berhasil menyelesaikan studi S1 nya. Penulis juga pernah menjadi salah satu anggota tim Solar Car ITS, yang mengikuti kejuaraan di Darwin-Adelaide, Australia, tahun 2013. Penulis pernah bekerja di salah satu perusahaan elektronik multinasional, Atmel Cooperation, sebagai *Application Engineer* (Jan – Aug 2014). Karena kecintaannya pada dunia pendidikan, akhirnya penulis mendapatkan beasiswa, menjadi salah satu *awardee* LPDP PK-46 dan melanjutkan jenjang studinya untuk mengambil Master di jurusan Teknik Elektro, bidang keahlian Teknik Sistem Tenaga, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember. Sebelumnya penulis juga pernah menjadi pengajar di salah satu lembaga kursus bahasa Inggris, di Kampung Inggris, Pare, Kediri, Mr. Bob English Club (2014 – 2015). Saat ini penulis aktif sebagai tim riset di Pusat Unggulan IPTEK Sistem Kontrol Otomotif (PUI-SKO) ITS, yang turut mengembangkan sistem elektronik pada kendaraan listrik. Selanjutnya, terkait riset maupun hal-hal lain yang perlu didiskusikan, penulis dapat dihubungi melalui email : chochoedogawa@gmail.com

ACKNOWLEDGEMENT

Penelitian ini didanai dan disponsori oleh :

Lembaga Pengelola Dana Pendidikan (LPDP)
Kementerian Keuangan Republik Indonesia

