



TUGAS AKHIR - TE 141599

**PERANCANGAN KONTROLER *NEURAL NETWORK*-PID
UNTUK PENGATURAN KECEPATAN MOTOR DC
*BRUSHLESS***

Tri Wahyu Kurniawan
NRP 2211100073

Dosen Pembimbing
Ir. Rusdhianto Effendi A.K., MT.
Ir. Ali Fatoni, MT.

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

(halaman ini sengaja dikosongkan)



FINAL PROJECT - TE 141599

***DESIGN OF NEURAL NETWORK-PID CONTROLLER FOR
SPEED CONTROL OF BRUSHLESS DC MOTOR***

Tri Wahyu Kurniawan
NRP 2211100073

Advisors
Ir. Rusdhianto Effendi A.K., MT.
Ir. Ali Fatoni, MT.

DEPARTMENT OF ELECTRICAL ENGINEERING
Faculty of Industrial Technology
Insitut Teknologi Sepuluh Nopember
Surabaya 2016

(halaman ini sengaja dikosongkan)

**PERANCANGAN KONTROLER NEURAL NETWORK-PID
UNTUK PENGATURAN KECEPATAN MOTOR DC
BRUSHLESS**

TUGAS AKHIR

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

Menyetujui

Dosen Pembimbing I,

Dosen Pembimbing II,


Ir. Rusdhianto Effendi A.K., MT.

NIP. 195704241985021001


Ir. Ali Fatoni, MT.

NIP. 196206031989031002

**SURABAYA
JANUARI, 2016**

KATA PENGANTAR

Alhamdulillah, puji syukur penulis panjatkan kehadirat Allah SWT karena atas berkah, rahmat, ijin dan karunia-Nya penulis dapat menyelesaikan penulisan buku Tugas Akhir dengan judul **“Perancangan Kontroler *Neural Network*-PID Untuk Pengaturan Kecepatan Motor DC *Brushless*”**. Tugas Akhir ini disusun untuk memenuhi salah satu persyaratan menyelesaikan pendidikan sarjana pada Bidang Studi Teknik Sistem Pengaturan, Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember.

Penulis menyadari bahwa dalam pelaksanaan dan penyelesaian Tugas Akhir ini banyak mengalami kendala, namun berkat bantuan, bimbingan, dan kerjasama dari berbagai pihak semua kendala tersebut dapat diatasi. Oleh karena itu pada kesempatan ini penulis ingin menyampaikan banyak terimakasih, rasa hormat dan penghargaan setinggi-tingginya kepada :

1. Bapak Ir. Katjuk Astrowulan M.S. E. E. selaku orang tua yang telah memberikan dukungan, semangat, dan doa.
2. Bapak Rusdhi dan Bapak Ali selaku dosen pembimbing mata kuliah Tugas Akhir Teknik Elektro ITS.
3. Bapak Dr. Ardyono Priyadi, ST. M.Eng. selaku Ketua Jurusan Teknik Elektro ITS.
4. Rekan-rekan e51 angkatan 2011 Jurusan Teknik Elektro Institut Teknologi Sepuluh Nopember atas momen kekeluargaan dan kerja sama yang luar biasa selama berkuliah di Teknik Elektro ITS.

Penulis menyadari bahwa pada penyusunan laporan Tugas Akhir ini masih terdapat beberapa kekurangan dikarenakan keterbatasan kemampuan penulis, walaupun demikian penulis berharap Tugas Akhir ini dapat bermanfaat bagi khalayak dan pihak-pihak yang membutuhkan.

Surabaya, 12 Januari 2016

Penulis

(halaman ini sengaja dikosongkan)

PERANCANGAN KONTROLER *NEURAL NETWORK*-PID UNTUK PENGATURAN KECEPATAN MOTOR DC *BRUSHLESS*

Tri Wahyu Kurniawan
2211100073

Dosen Pembimbing I : Ir. Rusdhianto Effendi A.K., MT.
Dosen Pembimbing II : Ir. Ali Faton, MT.

ABSTRAK

Dalam perancangan tugas akhir ini, tujuan yang diinginkan adalah mengatur kecepatan Motor DC *Brushless* ketika kondisi tanpa beban maupun diberikan pembebanan dengan menggunakan metode kontroler *Neural Network*-PID. *Neural Network* digunakan untuk *learning* dengan menggunakan metode *back propagation* untuk mendapatkan bobot w_{kp} , w_{ki} , w_{kd} yang tepat agar nilai dari parameter K_p , K_i , K_d untuk kontroler PID sama dengan nilai K_p , K_i , dan K_d yang didapatkan dari parameter *plant*. Kontroler PID digunakan untuk mengatur kecepatan motor arus searah tanpa sikat. Berdasarkan hasil simulasi didapatkan nilai $K_p = 0,238$, $K_i = 1,148$, $K_d = 0,013$.

Kata kunci : *Neural Network* (NN), PID, Motor DC *Brushless*

(halaman ini sengaja dikosongkan)

**DESIGN OF NEURAL NETWORK-PID FOR SPEED CONTROL
OF BRUSHLESS DC MOTOR**

Tri Wahyu Kurniawan
2211100073

Supervisor I : Ir. Rusdhianto Effendi A.K., MT.
Supervisor II : Ir. Ali Fatoni, MT.

ABSTRACT

In designing this final project, the desired goal is to adjust the speed of Brushless DC motor no-load or with load using Neural Network-PID controller. Neural Network is used for learning with back propagation method to obtain a precise wk_p , wk_i , wk_d so the value of parameter K_p , K_i , K_d for the PID controller is equal with K_p , K_i , K_d obtained from plant parameters. PID controller is used to control the speed of BLDC motor. Based on simulation results, the value of K_p , K_i , and K_d respectively are $K_p = 0.238$, $K_i = 1.148$, and $K_d = 0.013$.

Keywords : Neural Network (NN), PID, Brushless DC Motor

(halaman ini sengaja dikosongkan)

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN	vii
ABSTRAK	ix
ABSTRACT	xi
KATA PENGANTAR	xiii
DAFTAR ISI	xv
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
BAB 1 PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Perumusan Masalah	1
1.3. Batasan Masalah	1
1.4. Tujuan Penelitian	2
1.5. Sistematika Penulisan	2
1.6. Relevansi.....	3
BAB 2 DASAR TEORI	5
2.1. Motor DC <i>Brushless</i>	5
2.1.1. Konstruksi Motor DC <i>Brushless</i> (BLDC).....	5
2.1.2. Prinsip Kerja Motor BLDC.....	5
2.2. Rem Elektromagnetik	6
2.3. Arduino Uno	8
2.4. Identifikasi Sistem	9
2.4.1. Sinyal <i>Pseudo Random Binary Sequence</i> (PRBS)	11
2.5. MATLAB.....	11
2.6. <i>Neural Network</i> (NN) / Jaringan Saraf Tiruan (JST)	12
2.7. <i>Back Propagation Neural Network</i> (BPNN).....	13
2.8. Kontroler PID (<i>Proportional-Integral-Derivative</i>).....	14
BAB 3 PERANCANGAN SISTEM	17
3.1. Gambaran Umum Sistem	17
3.2. Perancangan Perangkat Keras	17
3.2.1. Perancangan Mekanik	18
3.2.2. Perancangan Elektronik	20
3.3.1. Mikrokontroler Arduino.....	21
3.3. Perancangan Perangkat Lunak	22

3.3.1. <i>Software</i> Arduino.....	22
3.3.2. <i>Software</i> MATLAB	23
3.4. Identifikasi dan Pemodelan Sistem.....	24
3.5. Perancangan Kontroler <i>Neural Network</i> -PID	25
3.5.1. Perancangan Kontroler <i>Neural Network</i>	25
3.5.2. Perancangan <i>Learning</i> pada Kontroler <i>Neural Network</i>	25
BAB 4 PENGUJIAN DAN ANALISA	29
4.1. Gambaran Umum Pengujian Sistem.....	29
4.2. Pengujian Perangkat Keras	29
4.2.1. Pengujian <i>Open Loop</i> Kecepatan Motor.....	29
4.2.2. Pengujian Rem Elektromagnetik	30
4.3. Simulasi Sistem	30
4.3.1 Simulasi dengan Menggunakan Kontroler <i>Neural-Network</i>	30
4.3.2 Proses <i>Learning</i> Kontroler <i>Neural-Network</i>	30
4.3.3 Proses Simulasi Kontroler	31
4.3.4 Pengujian Simulasi Respon dengan Kontroler	31
4.4. Realisasi <i>Plant</i> Sistem Motor BLDC.....	31
4.4.1. Pengujian Respon Implementasi Kontroler <i>Neural-Network</i> ..	32
4.4.2. Respon pada Semua Kondisi Pembebanan	33
4.5. Hasil Simulasi.....	34
4.6. Implementasi	36
BAB 5 PENUTUP	38
5.1. Kesimpulan.....	38
5.2. Saran	38
DAFTAR PUSTAKA.....	39
LAMPIRAN	42
RIWAYAT HIDUP.....	44

TABLE OF CONTENT

ABSTRAK	ix
ABSTRACT	xi
PREFACE	xiii
TABLE OF CONTENT	xv
LIST OF FIGURES	xvii
LIST OF TABLES	xix
CHAPTER 1 INTRODUCTION	1
1.1. Background.....	1
1.2. Problem Formulation.....	1
1.3. Scope of Problem	1
1.4. Research Purpose	2
1.5. Writing Sistematics	2
1.6. Relevance.....	3
CHAPTER 2 LITERATURE	5
2.1. Brushless DC Motor.....	5
2.1.1. Construction of Motor Brushless DC (BLDC).....	5
2.1.2. Work Principle of Motor BLDC	5
2.2. Electromagnetic Brake	6
2.3. Arduino Uno	8
2.4. System Identification.....	9
2.4.1. Pseudo Random Binary Sequence Signal (PRBS)	11
2.5. MATLAB	11
2.6. Neural Network (NN) / Jaringan Saraf Tiruan (JST).....	12
2.7. Back Propagation Neural Network (BPNN)	13
2.8. PID Controller (Proportional-Integral-Derivative).....	14
CHAPTER 3 SYSTEM DESIGN	17
3.1. General Overview of the System.....	17
3.2. Hardware Design.....	17
3.2.1. Mechanical Design.....	18
3.2.2. Electronic Design.....	20
3.3.1. Arduino Microcontroller.....	21
3.3. Software Design.....	22
3.3.1. Software Arduino.....	22

3.3.2. Software MATLAB	23
3.4. System Identification and Modelling	24
3.5. Neural Network-PID Controller Design.....	25
3.5.1. Neural Network Controller Design.....	25
3.5.2. Learning in Neural Network Controller Design	26
CHAPTER 4 EXAMINATION AND ANALYSIS	29
4.1. General Overview System Testing	29
4.2. Hardware Testing	29
4.2.1. Motor Speed Open Loop Testing.....	29
4.2.2. Electromagnetic Brake Testing.....	30
4.3. System Simulation	30
4.3.1 Simulation with Neural-Network Controller	30
4.3.2 Learning Process in Neural-Network.....	30
4.3.3 Controller Simulation Process	31
4.3.4 Simultion Response Testing with Controller.....	31
4.4. BLDC Motor Plant Realization	31
4.4.1. Implementation Response of Neural-Network Controller Testing	33
4.4.2. Response in All Load Condition.....	33
4.5. Simulation Results	34
4.6. Implementation.....	36
CHAPTER 5 CONCLUSION	37
5.1. Conclusion	37
5.2. Suggestion.....	37
BIBLIOGRAPHY	39
ENCLOSURE.....	41
BIOGRAPHY.....	43

DAFTAR GAMBAR

Gambar 2.1 Pergerakan Eksitasi Fase Fluks pada Motor BLDC	6
Gambar 2.2 Tampilan Arus Eddy	7
Gambar 2.3 Tampilan Rem Elektromagnetik.....	8
Gambar 2.4 Tampilan Arduino Uno	9
Gambar 2.5 Sinyal Uji <i>Pseudo Random Binary Sequence</i>	11
Gambar 2.6 Struktur <i>Neural Network</i>	13
Gambar 2.7 Struktur Kontroler PID.....	15
Gambar 3.1 Motor DC <i>Brushless</i>	19
Gambar 3.2 Ilustrasi Penampang Rem Elektromagnetik	20
Gambar 3.3 Tampilan <i>Software</i> Arduino IDE.....	23
Gambar 3.4 Struktur <i>Neural Network</i>	25
Gambar 3.5 Blok <i>Simulink Neural Network-PID</i>	27
Gambar 3.6 Blok <i>Simulink Plant</i> Sistem Motor BLDC	27
Gambar 4.1 Respon Realisasi Motor BLDC	32
Gambar 4.2 Respon <i>Plant</i> Beban Minimal	35
Gambar 4.3 Respon <i>Plant</i> Beban Nominal	35
Gambar 4.4 Respon <i>Plant</i> Beban Maksimal	36
Gambar 4.5 Respon <i>Plant</i> Implementasi.....	36

(halaman ini sengaja dikosongkan)

DAFTAR TABEL

Tabel 2.1 Spesifikasi Arduino Uno	8
Tabel 2.2 Karakteristik Kontroler <i>Proportional, Integral</i> , dan <i>Derivative</i>	15
Tabel 3.1 Spesifikasi Rem Elektromagnetik	20
Tabel 3.2 Identifikasi Sistem.....	24
Tabel 3.3 Bobot Setiap Pembebanan.....	27

(halaman ini sengaja dikosongkan)

BAB 1

PENDAHULUAN

1.1. Latar Belakang

Motor listrik merupakan salah satu peralatan yang sering digunakan oleh masyarakat, baik dalam peralatan kebutuhan masyarakat maupun dalam dunia industri. *Brushless Direct Current* (BLDC) Motor ini merupakan inovasi baru dalam dunia industri motor listrik. Beberapa karakteristik yang menjadi pertimbangan pemilihan BLDC Motor, antara lain torsi yang besar, suaranya halus, memiliki daya yang besar, dan dapat dikontrol dengan mudah. [1]

Pada Tugas Akhir ini digunakan dua metode kontrol yang akan digunakan untuk mengatur kecepatan BLDC, yaitu metode kontrol *Neural Network* (NN), dan metode kontrol *Proportional* (P), *Integral* (I), *Derivative* (D). *PID Controller* (*Proportional, Integral and Derivative*) adalah suatu kontroler yang digunakan untuk menentukan sistem tersebut presisi atau tidak dengan menggunakan karakteristik *feedback* (umpan balik) pada sistem. *Neural Network* (NN) adalah suatu sistem yang memproses informasi yang memiliki karakteristik mirip dengan *biological neural network* (jaringan saraf biologis). Kedua metode kontrol tersebut dapat dipilih untuk melengkapi karakteristik pengaturan satu sama lain. Keunggulan *Neural Network* ialah dapat beradaptasi terhadap parameter PID dengan melakukan perubahan bobot kemudian perubahan bobot *Neural Network* tersebut akan digunakan untuk *tuning* PID agar mendapatkan K_p , K_i , T_i , T_d .

1.2. Perumusan Masalah

Adapun rumusan masalah yang akan dibahas dalam tugas akhir ini, yaitu:

1. Bagaimana merancang sistem yang menggunakan *plant* Motor DC *Brushless* untuk memperoleh kecepatan putar motor yang presisi.
2. Bagaimana merancang kontroler *Neural Network*-PID yang optimal pada sistem hasil rumusan butir 1 tersebut di atas.

1.3. Batasan Masalah

- a. Metode yang digunakan pada Tugas Akhir untuk pengaturan kecepatan Motor DC *Brushless* adalah metode kontrol berbasis *Neural Network*-PID menggunakan *interface* berupa

mikrokontroler Arduino Uno Rev 3, dan *software* MATLAB R2013a.

- b. *Range* kerja kecepatan dari Motor DC *Brushless* yang akan diteliti berada pada *range* kecepatan 1250-2250 RPM.

1.4. Tujuan Penelitian

Tujuan dari penelitian Tugas Akhir ini, adalah sebagai berikut:

1. Merancang sistem pengaturan kecepatan Motor DC *Brushless* untuk memperoleh kecepatan putar motor yang presisi.
2. Merancang kontroler *Neural Network*-PID yang optimal untuk memenuhi tujuan pada butir 1 tersebut di atas.

Hasil yang diperoleh dari pelaksanaan tugas akhir diharapkan dapat memberikan informasi atau pengetahuan mengenai perancangan Motor DC *Brushless* terhadap pengaturan kecepatan dengan menggunakan kontroler *Neural Network*-PID.

1.5. Sistematika Penulisan

Sistematika penulisan laporan tugas akhir ini terbagi atas enam bagian yang tersusun seperti berikut :

BAB 1: PENDAHULUAN

Bab ini berisikan tentang latar belakang, rumusan masalah, tujuan pelaksanaan, metodologi penulisan, relevansi, serta sistematika penulisan laporan tugas akhir.

BAB 2: DASAR TEORI

Bab ini berisikan penjelasan dasar mengenai hal-hal yang berhubungan dengan Tugas Akhir, yaitu Motor DC *Brushless*, *Neural Network* (NN), *Proportional Integral Derivatif* (PID).

BAB 3: PERANCANGAN SISTEM

Bab ini berisikan tentang penjelasan mengenai prancangan *hardware* dan *software* yang digunakan dalam pengerjaan Tugas Akhir.

BAB 4: PENGUJIAN DAN ANALISIS

Bab ini membahas mengenai implementasi dan hasil simulasi terhadap *plant* Motor DC *brushless* yang menggunakan kontroler *Neural Network*-PID.

BAB 5: PENUTUP

Bab ini berisikan kesimpulan dan saran dari pembahasan yang telah dijelaskan pada laporan Tugas Akhir ini.

1.6. Relevansi

Hasil yang diperoleh dari Tugas Akhir diharapkan dapat memberikan manfaat, antara lain:

1. Dapat dijadikan referensi dalam pembuatan sistem perancangan kontroler untuk pengaturan kecepatan motor jenis Motor DC *Brushless*.
2. Dapat dijadikan referensi perancangan kontroler yang berbasis *Neural Network-PID* untuk objek penelitian pengaturan kecepatan motor jenis lainnya atau penelitian yang berhubungan dengan *Neural Network-PID*.

(halaman ini sengaja dikosongkan)

BAB 2

DASAR TEORI

2.1. Motor DC *Brushless* [1]

Motor DC *Brushless* adalah jenis motor sinkron 3 fase yang medan magnetnya dihasilkan oleh stator dan rotor yang berputar pada frekuensi yang sama. Motor jenis ini mempunyai magnet permanen pada bagian rotor, sedangkan *elektromagnet*-nya berada pada bagian stator-nya. Setelah itu, dengan menggunakan sebuah rangkaian sederhana, maka kita dapat mengubah arus di *elektromagnet*-nya ketika bagian rotor-nya berputar. Sesuai dengan namanya, Motor DC *Brushless* tidak menggunakan sikat atau *brush* untuk pergantian medan magnet (komutator), melainkan menggunakan komutator elektronik. Motor BLDC mempunyai banyak keuntungan dibandingkan dengan motor DC dan Motor Induksi biasa.

2.1.1. Konstruksi Motor DC *Brushless* (BLDC)

Dua bagian inti yang dimiliki Motor DC *Brushless* (BLDC), yaitu stator (yang diam) dan rotor (yang berputar). Bagian inti yang penting lainnya adalah gulungan stator dan magnet rotor.

a. Rotor

Rotor adalah bagian dari motor yang berputar karena ada elektromagnetik dari stator. Rotor berasal dari dua magnet permanen, yaitu magnet utara (N) dan magnet selatan (S).

b. Stator

Stator adalah bagian dari motor yang tidak bergerak atau diam yang berfungsi sebagai medan putar pada motor untuk memberikan elektromagnetik pada rotor dan terdapat 12 belitan (elektromagnet).

2.1.2. Prinsip Kerja Motor BLDC

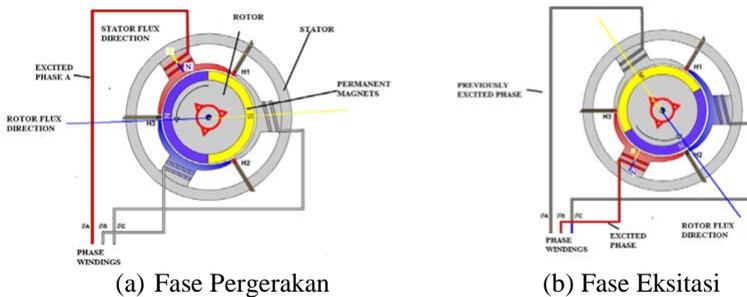
Hal yang paling dasar pada medan magnet adalah kutub yang sama akan saling tolak menolak, sedangkan apabila berlainan kutub akan tarik menarik. Jika ada dua buah magnet dan menandai satu sisi magnet tersebut dengan *north* (utara) dan yang lainnya *south* (selatan), maka bagian sisi *north* akan menarik *south*, sebaliknya sisi *north* magnet

pertama akan menolak sisi *north* yang kedua dan seterusnya apabila kedua sisi magnet mempunyai kutub yang sama.

Prinsip mengenai kutub magnet tersebut dapat diterapkan dalam prinsip kerja motor BLDC. Secara umum motor BLDC memiliki medan magnet permanen pada rotor dan magnet yang berasal dari gaya elektromagnet (magnet yang ditimbulkan dari pemberian *input* arus listrik) pada bagian kumparan stator.

Pada motor BLDC, kontroler berfungsi untuk mengatur arus masukan yang harus dialirkan ke kumparan stator untuk dapat menimbulkan medan elektromagnet yang sesuai untuk memutar rotor. Hal inilah yang menjadi pembeda dengan motor DC konvensional, dan menggantikan kerja komutasi mekanisnya.

Magnet permanen pada motor BLDC dilengkapi dengan kumparan tiga fase. Kumparan-kumparan tersebut terletak di bagian stator. Magnet bergerak terletak di stator. Fase kumparan diaktifkan dengan menyesuaikan gerakan rotor. Rotasi berbasis medan magnet diilustrasikan pada Gambar 2.1 (a) dan Gambar 2.1 (b) menjelaskan pergerakan dan eksitasi fase. Pada Gambar 2.1 (a) fase A dieksitasi, fluks stator dihasilkan oleh eksitasi fase A, fluks rotor dihasilkan oleh magnet permanen.



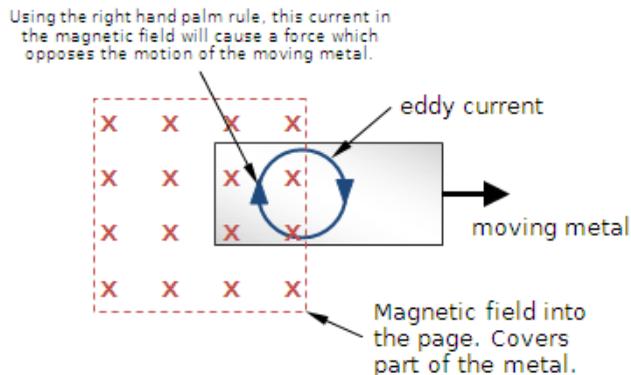
Gambar 2.1 Pergerakan Eksitasi Fase Fluks pada Motor BLDC [1]

2.2. Rem Elektromagnetik [2]

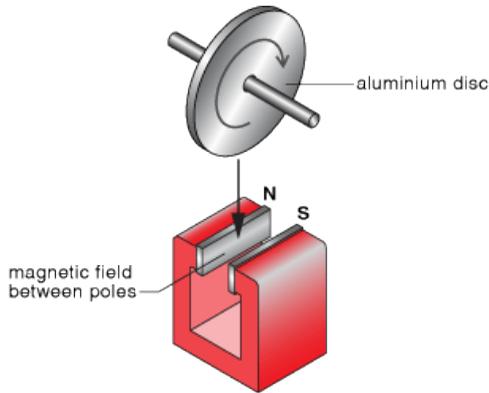
Rem elektromagnetik diperoleh dari suatu sistem pengereman yang menggunakan gaya elektromagnetik untuk memperlambat gerakan, yang umumnya adalah gerakan poros. Pada sebuah piringan dengan bahan logam *nonferromagnetic* terpasang sebuah poros berputar. Piringan

tersebut diapit oleh sisi stator berupa sistem lilitan elektromagnetik yang dapat membangkitkan medan magnet dari aliran listrik. Arus listrik menimbulkan medan magnet pada lilitan. Logam piringan yang memotong medan magnet tersebut menimbulkan arus Eddy pada piringan itu sendiri. Arus Eddy ini menimbulkan medan magnet yang arahnya berlawanan dengan medan magnet sebelumnya, sehingga menghambat gerakan putar dari poros tersebut.

Arus Eddy merupakan arus listrik yang timbul apabila suatu piringan logam non *ferromagnetik* berada di sekitar medan magnet yang garis gayanya sedang berubah-ubah. Medan magnet yang dihasilkan oleh arus Eddy ini berlawanan arah dengan arah gerakan piringan logam. Dengan demikian medan magnet yang ditimbulkan oleh arus Eddy akan menghambat laju piringan logam tersebut. Gambar 2.2 dan Gambar 2.3 merupakan tampilan arus Eddy dan tampilan rem elektromagnetik.



Gambar 2.2 Tampilan Arus Eddy [2]



Gambar 2.3 Tampilan Rem Elektromagnetik. [2]

2.3. Arduino Uno [3]

Arduino Uno adalah *board* mikrokontroler berbasis Atmega328. Alat ini mempunyai 14 pin *input/output* digital dan 6 di antaranya dapat digunakan sebagai *output* PWM, 6 *input* analog, resonator keramik 19MHz, koneksi USB, soket listrik, ICSP *header*, dan tombol *reset*. Dengan kabel USB, alat ini mudah dihubungkan ke komputer dan dapat diaktifkan dengan baterai atau adaptor AC ke DC. Spesifikasi Arduino Uno dapat dilihat pada Tabel 2.1.

Tabel 2.1 Spesifikasi Arduino Uno

Mikrokontroler Arduino	Atmega 328
Tegangan Operasi	5 V
Tegangan <i>Input</i> (direkomendasikan)	7 - 12 V
Tegangan <i>Input</i> (Batasan)	6 - 20 V
Pin <i>Input</i> / <i>Output</i> Digital	14 (6 di antaranya <i>output</i> PWM)
Pin <i>Input</i> Analog	6
Arus DC Tiap Pin I / O	40 mA
Arus DC Untuk Pin 3,3 V	50 mA

Empat belas pin *input* dan *output* Arduino Uno dapat digunakan dengan fungsi *pinMode()*, *digitalWrite()*, dan *digitalRead()*. Setiap pin

memiliki tegangan operasi 5V dan dapat menerima arus maksimum 40mA. Beberapa pin memiliki fungsi khusus, antara lain:

- a. *Serial*: 0 (RX) dan 1 (TX). Pin ini digunakan untuk menerima (RX) dan mengirim (TX) data *serial* TTL. Pin ini terhubung dengan pin pada Atmega8U2 USB ke cip USB ke TTL *serial*
- b. *External Interrupts*: 2 dan 3. Pin ini digunakan sebagai *trigger* gangguan pada nilai yang rendah, menaikkan dan menurunkan nilai, atau merubah nilai.
- c. PWM: 3, 5, 6, 9, 10, dan 11. Pin ini menyediakan *output* PWM 8bit dengan fungsi *analogWrite()*.
- d. SPI (*Serial Peripheral Interface*) : 10 (*Slave Select*), 11 (*Master Out Slave In*), 12 (*Master In Slave Out*), 13 (*Serial Clock*). Pin ini mendukung komunikasi SPI.
- e. *LED*: 13, *LED* ini terhubung dengan pin 13. Ketika pin memiliki nilai yang tinggi, maka *LED* akan *on* dan ketika pin memiliki nilai yang rendah, maka *LED* akan *off*

Ilustrasi rancangan penampang arduino uno yang digunakan pada plant motor BLDC dapat dilihat pada Gambar 2.4.



Gambar 2.4 Tampilan Arduino Uno [3]

2.4. Identifikasi Sistem [4]

Identifikasi sistem merupakan suatu langkah awal dalam menganalisa sistem dinamik. Menurunkan suatu model matematika yang baik dan sesuai merupakan salah satu bagian terpenting dalam proses menganalisa sebuah sistem secara keseluruhan. Model matematika yang baik dan sesuai cocok digunakan untuk analisa, prediksi, dan desain sistem, *regulator*, dan filter. Model matematika memiliki bentuk yang bermacam-macam. Salah satu bentuknya adalah *transfer function* yang

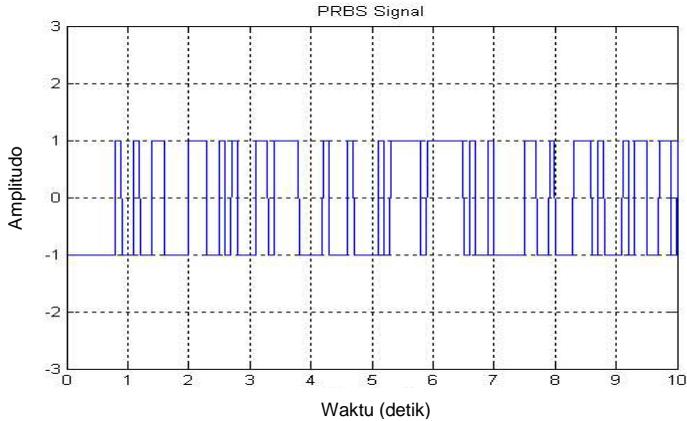
cocok untuk permasalahan analisa transien dan sebuah sistem LTI (*Linear Time Invariant*) dan sistem dengan *Single-Input Single-Output* (SISO). Di sisi lain, model matematika dengan bentuk *state space* sangat cocok untuk menganalisa suatu sistem dengan *Multiple-Input Multiple-Output* (MIMO).

Model matematika dari sebuah sistem dapat diperoleh melalui dua cara yaitu permodelan fisik dan identifikasi sistem. Permodelan fisik merupakan pendekatan analitik berdasarkan hukum fisika seperti hukum Newton dan hukum kesetimbangan. Permodelan ini menjelaskan dinamika dalam sistem. Yang kedua adalah identifikasi sistem. Hal ini dilakukan dengan pendekatan eksperimental. Model ini berdasarkan data dari eksperimen yang kemudian didapatkan nilai parameter sistem. Pada beberapa kasus yang sangat kompleks, sangat sulit untuk menentukan model berdasarkan pemahaman fisik.

Identifikasi sistem pada suatu sistem dapat dilakukan dengan menggunakan metode identifikasi statis maupun dinamis. Identifikasi statis dilakukan untuk mendapatkan *gain* dan *time sampling* dari suatu sistem. Identifikasi statis pada suatu sistem dilakukan dengan memberikan suatu *input setpoint* yang bernilai konstan terhadap waktu. Sistem yang digunakan untuk identifikasi statis adalah sistem dengan *loop* terbuka tanpa kontroler, sedangkan pada identifikasi dinamis, prosedur identifikasinya memiliki beberapa perbedaan mendasar, di antaranya adalah perbedaan antara sinyal uji dan metode pemodelan yang digunakan. Jika identifikasi statis menggunakan *input setpoint* yang konstan terhadap waktu, maka identifikasi dinamis menggunakan sinyal acak atau *random* sebagai sinyal ujinya. Pada Tugas Akhir kali ini digunakan metode identifikasi dinamis. Pemilihan metode ini disebabkan karena respon yang sangat cepat dari motor BLDC, sehingga sangat sulit untuk mengamati respon transien dari sistem jika menggunakan identifikasi statis.

Terdapat beberapa perbedaan mendasar antara identifikasi statis dan dinamis. Perbedaan tersebut di antaranya adalah perbedaan antara sinyal uji dan metode pemodelan yang digunakan. Jika identifikasi statis menggunakan *input setpoint* yang konstan terhadap waktu, maka identifikasi dinamis menggunakan sinyal acak atau *random* sebagai sinyal ujinya. Sinyal ini memiliki frekuensi yang berubah-ubah, sehingga memungkinkan karakteristik sistem dapat diketahui secara lebih teliti.

1.4.1. Sinyal *Pseudo Random Binary Sequence* (PRBS)



Gambar 2.5 Sinyal Uji *Pseudo-Random Binary Square* (PRBS) [4]

Sinyal tersebut dinamakan sinyal *Pseudo-Random Binary Square* (PRBS). Sinyal PRBS seperti pada Gambar 2.5 mirip dengan bilangan acak secara nyata, tapi juga dapat disebut semu atau *pseudo* karena bersifat deterministik. Sinyal PRBS dapat dihasilkan dari penggunaan *shift register*. Salah satu contoh sinyal uji PRBS dapat dilihat pada Gambar 2.5.

2.5. MATLAB [5]

MATLAB merupakan paket program dengan bahasa pemrograman yang tinggi untuk mengembangkan algoritma, visualisasi data, dan komputasi numerik. Program MATLAB ini dapat digunakan untuk menyelesaikan masalah komputasi dengan lebih cepat dibandingkan dengan bahasa pemrograman tradisional, seperti C, C++, dan Fortran. MATLAB digunakan untuk banyak aplikasi seperti *signal and image processing*, desain kontrol, pengujian dan pengukuran, pemodelan, dan analisis.

Simulink merupakan bagian dari MATLAB untuk memodelkan, mensimulasikan, dan menganalisa sistem dinamik. *Simulink* dapat membentuk model dari awal atau memodifikasi model yang sudah ada sesuai dengan apa yang diinginkan. Selain itu *simulink* juga mendukung sistem linier dan nonlinier, pemodelan waktu kontinu atau diskrit, atau gabungan. *Simulink* ini dapat digunakan sebagai media untuk

menyelesaikan masalah dalam industri nyata, yang meliputi kedirgantaraan dan pertahanan, otomotif, komunikasi, elektronik dan pemrosesan sinyal.

Salah satu modul dalam *Simulink* yang dapat digunakan untuk komunikasi perangkat keras adalah *Instrument Control Toolbox*. Modul ini merupakan kumpulan fungsi *m-file* yang dibangun pada lingkungan komputasi teknis MATLAB. *Toolbox* ini menyediakan kerangka kerja untuk komunikasi instrumen yang mendukung GPIB *interface*, standar VISA, TCP/IP, dan protokol UDP. *Toolbox* ini memperluas fitur dasar *serial port* yang ada dalam MATLAB. Selain itu, *toolbox* ini berfungsi untuk komunikasi data antara *workspace* MATLAB dan peralatan lainnya. Data tersebut dapat berbentuk biner atau *text*.

Komunikasi *serial* merupakan protokol dasar tingkat rendah untuk komunikasi antara dua peralatan atau lebih. Pada umumnya satu komputer dilengkapi dengan modem, *printer*, mikrokontroler, atau peralatan lainnya. *Serial port* mengirim dan menerima informasi *bytes* dengan hubungan seri. *Bytes* tersebut dikirimkan menggunakan format biner atau karakter ASCII (*American Standard Code for Information Interchange*). Dalam komunikasi serial MATLAB, agar data ASCII dapat diproses *real time*, maka digunakan ASCII *encode* dan *decode* yang terdapat pada *xPC Target Library for RS232*. ASCII *encode* merupakan blok dalam *simulink* yang digunakan untuk mengubah data *bytes* menjadi karakter ASCII, sedangkan ASCII *decode* merupakan blok *Simulink* yang digunakan untuk mengubah karakter ASCII menjadi data *bytes* yang kemudian dapat dikonversi sesuai kebutuhan.

2.6. Neural Network (NN) / Jaringan Saraf Tiruan (JST) [6]

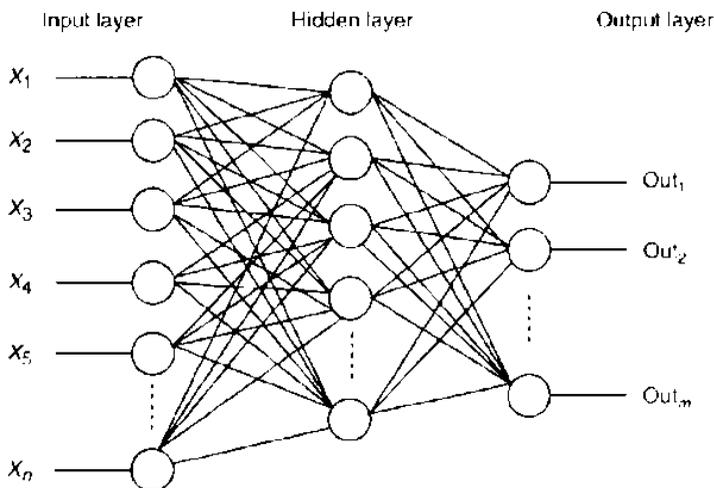
Neural Network (NN) adalah Suatu sistem pemroses informasi yang mempunyai karakteristik yang mirip dengan jaringan saraf biologis.

Neural Network ini merupakan generalisasi pemodelan matematis dalam proses kognitif berdasarkan asumsi :

- 1.) Pemrosesan informasi terjadi pada elemen sederhana yang dinamakan *neuron*.
- 2.) Sinyal antar *neuron* berhubungan melalui saluran penghubung.
- 3.) Setiap saluran penghubung mempunyai nilai bobot, dan melakukan operasi perkalian dengan sinyal yang di-transmisikan.

- 4.) Setiap *neuron* mempunyai fungsi aktivasi (biasanya tidak linier) pada masukan total (bagi *neuron* tersebut) untuk mendapatkan sinyal keluarannya.

Ilustrasi rancangan kontroler neural network yang digunakan pada *plant* motor BLDC dapat dilihat pada Gambar 2.6.



Gambar 2.6 Struktur *Neural Network* [6]

2.7. Back Propagation Neural Network (BPNN) [6]

BPNN merupakan algoritma pelatihan terbimbing yang mempunyai banyak lapisan. BPNN menggunakan *error output* untuk mengubah nilai bobot-bobotnya dalam arah mundur (*backward*). Untuk mendapatkan *error* ini, tahap perambatan maju (*forward propagation*) harus dikerjakan terlebih dahulu.

Syarat fungsi aktivasi dalam BPNN adalah bersifat kontinu, terdiferensial dengan mudah, dan merupakan fungsi yang tidak turun. Fungsi aktivasi yang dapat memenuhi ketiga syarat tersebut adalah logsig, tansig, dan purelin.

Metode pengenalan merupakan proses inialisasi data yang selanjutnya akan diolah oleh BPNN. Data yang akan dikenali disajikan dalam bentuk vektor. Masing-masing data mempunyai target yang disajikan juga dalam bentuk vektor. Target atau keluaran acuan

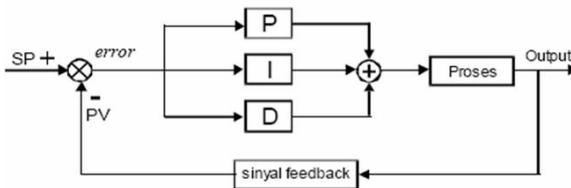
merupakan suatu peta karakter yang menunjukkan lokasi dari vektor masukan, sedangkan metode pelatihan merupakan proses latihan mengenali data dan menyimpan pengetahuan atau informasi yang didapat pada bobot-bobot (Heaton, 2003).

Dalam pelatihan BPNN Terdapat 3 fase, yaitu fase maju (*feed forward*), fase mundur (*back propagation*), dan fase modifikasi bobot. Dalam fase *feed forward*, pola masukan dihitung maju dimulai dari lapisan *input* hingga lapisan *output*. Dalam fase *back propagation*, tiap-tiap unit *output* menerima target pola yang berhubungan dengan pola *input* untuk dihitung nilai kesalahan. Kesalahan tersebut akan dipropagasikan mundur, sedangkan fase modifikasi bobot bertujuan untuk menurunkan kesalahan yang terjadi. Ketiga fase tersebut diulang secara terus menerus hingga kondisi penghentian dipenuhi.

2.8. Kontroler PID (*Proportional-Integral-Derivative*) [7]

Kontroler PID (*Proportional-Integral-Derivative Controller*) merupakan kontroler untuk menentukan presisi suatu sistem instrumentasi dengan karakteristik adanya umpan balik pada sistem. Kontroler PID adalah kontroler konvensional yang banyak dipakai dalam dunia industri. Kontroler PID akan memberikan aksi kontrol kepada *plant* berdasarkan besar kesalahan yang diperoleh. Kesalahan adalah perbedaan dari *setpoint* dengan *output* sistem pengaturan.

Ilustrasi rancangan kontroler *Proportional, Integral, Derivative* (PID) yang digunakan pada *plant* motor BLDC dapat dilihat pada Gambar 2.7.



Gambar 2.7 Struktur Kontroler PID [7]

Adapun persamaan Kontroler PID adalah:

$$mv(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right) \quad (2.1)$$

- mv(t) : *manipulated variable*
- Kp : *konstanta proportional*
- Ti : *time integral*
- Td : *time derivative*
- e(t) : *eror*

Komponen kontrol PID ini terdiri dari tiga jenis yaitu *proportional, integral dan derivative*. Ketiganya dapat dipakai bersamaan maupun sendiri-sendiri tergantung dari respon yang kita inginkan terhadap suatu *plant*. Efek dari setiap pengontrol *Proportional, Integral dan Derivative* pada sistem *loop* tertutup disimpulkan pada Tabel 2.2.

Tabel 2.2 Karakteristik Kontroler *Proportional, Integral, dan Derivative*

<i>Respon Close-Loop</i>	<i>Rise Time</i>	<i>Overshoot</i>	<i>Setting Time</i>	<i>Steady State Error</i>
<i>Proportional</i>	Turun	Naik	Perubahan Kecil	Turun
<i>Integral</i>	Turun	Naik	Naik	Hilang
<i>Derivative</i>	Perubahan Kecil	Turun	Turun	Perubahan Kecil

(halaman ini sengaja dikosongkan)

BAB 3

PERANCANGAN SISTEM

Bab ini menjelaskan segala hal terkait perancangan perangkat keras, dan perangkat lunak yang akan digunakan, serta perancangan kontroler menggunakan metode *Neural Network*-PID.

3.1. Gambaran Umum Sistem

Plant motor BLDC ini dibuat secara berkelompok yang terdiri dari Guntur Sadhiea Putra, Muhammad Safruriza, Fairuzza Dinansyar, Muhammad Iqbal Fauzi, Irwan Eko Prabowo. Komponen penggerak utama yang menjadi objek Tugas Akhir yang dirangkai adalah BLDC Motor. Selain itu, sistem pada *plant* motor BLDC ini juga menggunakan rem elektromagnetik, yang berfungsi sebagai beban kinerja Motor DC *Brushless*.

Sistem yang dibahas pada Tugas Akhir ini terdiri dari beberapa komponen yang dirangkai membentuk suatu *plant* motor BLDC terpadu untuk menjalankan objektif tertentu sesuai dengan tujuan dilaksanakan Tugas Akhir ini.

Untuk mendukung kinerja *plant* motor BLDC tersebut dibutuhkan komponen pendukung lain berupa sub sistem yang mendukung kerja dan fungsi masing-masing perangkat yang telah disebutkan sebelumnya. Perangkat pendukung tersebut antara lain *driver* motor BLDC yang digunakan untuk mengatur kecepatan, rangkaian isolasi dan juga rangkaian pengatur PWM (*Pulse Width Modulation*) pada rem yang digunakan untuk mengatur besarnya sinyal kontrol yang diberikan pada rem guna membantu besarnya pembebanan.

Pengaturan kecepatan motor BLDC yang sekaligus sebagai objek kontrol utama Tugas Akhir ini menggunakan metode kontrol *Neural Network* (NN) – *Proportional, Integral, Derivative* (PID). Penggunaan metode kontrol ini diharapkan dapat membantu meningkatkan performa motor untuk menghasilkan respon performansi yang diinginkan.

3.2. Perancangan Perangkat Keras

Perancangan perangkat keras yang digunakan pada Tugas Akhir ini terbagi menjadi dua jenis perancangan, yaitu perancangan mekanik dan perancangan elektronik.

3.2.1. Perancangan Mekanik

Sistem *plant* motor BLDC terdiri dari beberapa komponen yang bekerja sama. Komponen pendukung tersebut antara lain motor BLDC sebagai penggerak utama yang sekaligus menjadi objek kontrol utama dalam pelaksanaan Tugas Akhir ini. Selanjutnya terdapat komponen yang berfungsi untuk memberikan efek pembebanan pada motor, yaitu rem elektromagnetik. Rem elektromagnetik tersebut dirancang untuk memberikan efek pembebanan pada piringan yang dihubungkan dengan poros yang digerakkan langsung oleh motor BLDC. Untuk rancangan selengkapannya akan dijelaskan pada sub bab berikutnya.

3.2.1.1. Motor DC *Brushless*

Motor BLDC dipilih karena memiliki beberapa keunggulan bila dibandingkan dengan motor listrik yang lain. Beberapa karakteristik yang menjadi pertimbangan memilih motor BLDC antara lain dapat menggerakkan dengan daya yang besar, suaranya halus, ukuran kompak, torsi besar, efisiensi tinggi, memiliki umur pakai yang panjang (keandalan yang tinggi), mudah dikontrol, dan perawatan yang rendah.

Umumnya kecepatan motor dan torsi motor BLDC ini bergantung pada kekuatan medan magnet yang dihasilkan oleh kumparan motor yang bergantung terhadap suplai arus. Oleh sebab itu, kecepatan motor BLDC diatur dengan mengatur tegangan dan arus yang ada pada tegangan. Berdasarkan kondisi tersebut, maka metode pengaturan yang digunakan dalam kasus ini membutuhkan metode pengaturan yang memiliki respon cepat.

Pada Tugas Akhir ini motor BLDC yang digunakan merupakan motor BLDC yang biasa digunakan pada AC *Inverter* Daikin dengan tipe Daikin D43F. Motor BLDC jenis ini merepresentasikan karakteristik motor BLDC yang sebenarnya, hanya saja dalam konstruksi dan penampang mengacu pada skala yang lebih kecil dari motor BLDC yang digunakan pada kendaraan listrik. Bentuk fisik *plant* motor BLDC dapat dilihat pada Gambar 3.1.



Gambar 3.1 Motor DC *Brushless*

3.2.1.2. Rem Elektromagnetik

Untuk menghasilkan efek pembebanan pada *plant* motor BLDC digunakan bantuan alat berupa rem elektromagnetik. Rem elektromagnetik yang digunakan pada *plant* BLDC ini dibuat sedemikian rupa agar dapat diatur kekuatan medan magnetnya yang berasal dari efek arus Eddy dengan mengatur besar tegangan input yang masuk pada kumparan.

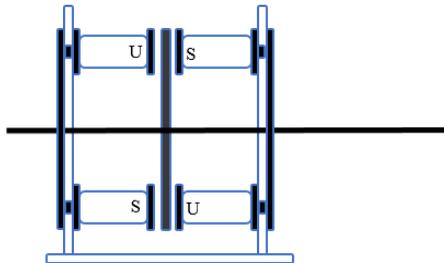
Efek pengereman dihasilkan oleh efek torsi yang berlawanan arah hasil dari induksi medan magnet yang terbentuk akibat aliran arus listrik yang mengalir pada kumparan rem. Sumber tegangan yang digunakan rem elektromagnetik didapat dari tegangan jala-jala PLN yang kemudian diturunkan menggunakan trafo *step down* dan disearahkan menggunakan rangkaian penyearah gelombang penuh. Pengaturan arus DC *input* berupa PWM (*Pulse Width Modulation*) yang dialirkan ke rem elektromagnetik dikontrol menggunakan rangkaian pengatur *duty cycle* yang diatur menggunakan sebuah *driver* dengan menggunakan Arduino uno.

Rem elektromagnetik pada *plant* motor BLDC ini tersusun dari delapan buah kumparan rem yang terhubung secara seri dan terbuat dari inti besi yang dililit dengan kawat tembaga. Delapan kumparan ini terpisah menjadi dua bagian dan diberikan celah untuk ditempatkan sebuah piringan beban yang berbahan *nonferromagnetic* dan terhubung dengan poros yang digerakkan langsung oleh motor BLDC. Konstruksi rem terdiri dari 4 kumparan pada tiap sisinya dengan masukan tegangan DC yang diberikan pada tiap sisi sebesar 16 V, 20 V, dan 24 V. Spesifikasi rem elektromagnetik dapat dilihat pada Tabel 3.1.

Tabel 3.1 Spesifikasi Rem Elektromagnetik

Jumlah Kumputaran	8 (masing –masing 400 kumputaran)
Resistansi Kumputaran (Seri)	12 Ω
Diameter Kumputaran	3 cm
Diameter Kawat	0,6 mm
Tegangan Kerja	0-30 V
Arus Maksimal	3 A

Ilustrasi penampang rem elektromagnetik yang digunakan pada *plant* motor BLDC dapat dilihat pada Gambar 3.2.



Gambar 3.2 Ilustrasi Penampang Rem Elektromagnetik

3.2.2. Perancangan Elektronik

Perancangan elektronik pada *plant* motor BLDC meliputi rangkaian elektronik menggunakan PCB serta pengkabelan pada *plant* motor BLDC. Rangkaian elektronik pada *plant* motor BLDC terdiri dari beberapa rangkaian, yaitu rangkaian *driver* motor BLDC, dan rangkaian *driver* rem elektromagnetik.

3.2.2.1. Rangkaian Driver Motor BLDC

Rangkaian *driver* motor BLDC merupakan hal paling utama untuk dapat menggerakkan motor. Tanpa *driver*, motor BLDC tidak dapat bekerja sesuai dengan keinginan. Pada *plant* motor BLDC P-1 ini *driver* yang digunakan untuk menggerakkan motor BLDC ialah sebuah *driver* elektronik yang sudah terintegrasi menjadi modul dan banyak digunakan untuk AC *Inverter*. Dalam Tugas Akhir ini yang digunakan adalah motor BLDC AC *Inverter* Daikin.

3.2.2.2. Rangkaian Driver Rem Elektromagnetik

Pada Tugas Akhir ini akan diberlakukan tiga kondisi pembebanan yaitu kondisi beban minimal, nominal dan maksimal. Efek pembebanan tersebut didapat dari efek yang dihasilkan oleh rem elektromagnetik. Untuk mengatur besar kecilnya intensitas kekuatan magnet yang akan digunakan, maka dibutuhkan suatu rangkaian elektronik yang dapat mengaturnya. Dalam hal ini pengaturan yang dilakukan ialah dengan mengatur arus masukan yang digunakan untuk menyuplai kumparan elektromagnetiknya.

Berdasarkan hukum Ohm, tegangan akan berbanding lurus dengan arus. Pada kasus ini pengaturan arus masukan dilakukan dengan mengatur tegangan yang masuk dengan menggunakan teknik PWM (*Pulse Width Modulation*).

Rem elektromagnetik sangat berperan penting dalam keseluruhan sistem ini. Rem digunakan untuk memberikan pembebanan pada motor BLDC. Untuk mengatur nilai pembebanan yang akan diberikan kepada motor BLDC, dibutuhkan suatu rangkaian yang digunakan untuk mengatur besaran arus DC yang masuk ke dalam kumparan. Oleh karena itu, dibuatlah suatu *driver* yang digunakan untuk mengatur arus yang masuk ke dalam kumparan. Arus ini berupa arus DC yang telah disearahkan oleh rangkaian penyearah. Pengaturan arus yang masuk ke dalam kumparan tersebut dilakukan menggunakan teknik PWM atau *Pulse Width Modulation*. Dengan teknik PWM ini tegangan dan arus yang ingin digunakan akan dapat diatur sesuai keinginan.

3.3.1. Mikrokontroler Arduino

Untuk menghubungkan komunikasi antara instrumen berupa sensor maupun aktuator serta rangkaian pada sistem dengan program yang diatur melalui komputer dibutuhkan suatu *interface* atau penghubung. Pada Tugas Akhir ini digunakan sebuah mikrokontroler sebagai penghubung.

Arduino Uno Rev3 ini memiliki total 14 pin *input/output* yang terdiri dari 6 pin sebagai *output* PWM, 6 pin sebagai analog *input*, *crystal* osilator 16 MHz, koneksi USB, *jack power*, kepala ICSP, dan tombol *reset*.

3.3. Perancangan Perangkat Lunak

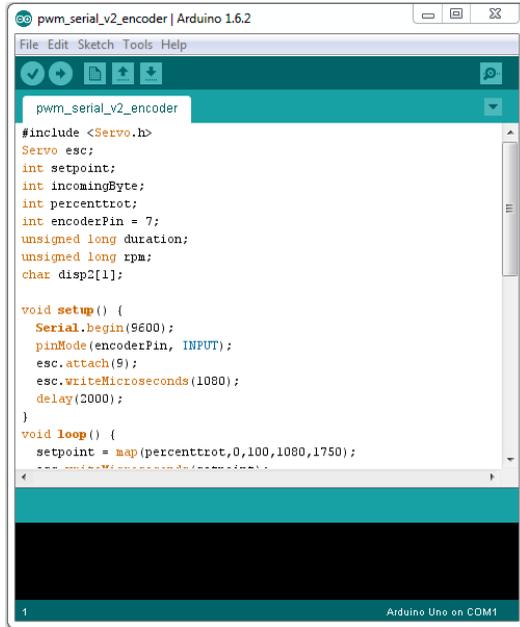
Perangkat lunak diperlukan dalam perancangan sistem sebagai *interface* antara *plant* motor BLDC dan komputer. Perangkat lunak yang digunakan dalam pengerjaan Tugas Akhir ini antara lain Arduino dan

MATLAB. Arduino digunakan sebagai alat akuisisi data, sedangkan MATLAB digunakan untuk membaca nilai dari sensor untuk keperluan identifikasi sistem dan mengirimkan sinyal kontrol pada Arduino. Selain itu, MATLAB digunakan pula sebagai *software* untuk desain, simulasi dan implementasi kontroler.

3.3.1. *Software* Arduino

Proses yang dilakukan Arduino Uno Rev3 dalam melakukan akuisisi dan pengaturan antara instrumen pada sistem dengan komputer memerlukan program yang harus di *compile* terlebih dahulu ke dalam mikrokontroler Arduino Uno Rev 3. Untuk melakukan proses pemrograman dan *compile* pada arduino menggunakan *software* Arduino IDE. Langkah untuk memprogram Arduino Uno Rev3 sendiri dapat dilakukan dengan menghubungkan Arduino Uno Rev3 tersebut ke komputer dengan menggunakan kabel USB. Pada tampilan *software* Arduino IDE harus disesuaikan terlebih dahulu *port* mana yang akan digunakan serta pemilihan jenis Arduino yang digunakan.

Secara garis besar, pada Tugas Akhir kali ini Arduino Uno Rev3 digunakan untuk membaca kecepatan melalui sensor kecepatan motor yang digunakan, mengatur penggunaan besarnya efek pembebanan yang dihasilkan rem elektromagnetik melalui rangkaian *driver* rem elektromagnetik, serta mengatur sinyal kontrol untuk menggerakkan motor BLDC yang direpresentasikan sebagai *throttle* yang sekaligus sebagai representasi dari besarnya sinyal PWM yang diberikan untuk menggerakkan motor. Tampilan program Arduino IDE dapat dilihat pada Gambar 3.3.



Gambar 3.3 Tampilan Software Arduino IDE [3]

3.3.2. Software MATLAB

Perangkat lunak lain yang digunakan dalam Tugas akhir ini ialah *software* MATLAB. Pada Tugas Akhir ini *software* MATLAB yang digunakan ialah MATLAB 2013a yang merupakan versi MATLAB keluaran tahun 2013.

Proses yang dilakukan dengan *software* MATLAB ini ialah proses komunikasi antar *device* yakni komputer dengan instrumen pada *plant* BLDC pengiriman berupa sensor, aktuator dan lainnya. Komunikasi yang dimaksud di sini adalah proses pengiriman dan penerimaan data dengan memanfaatkan komunikasi serial yang ada pada Arduino. Selanjutnya Simulink pada MATLAB dapat mengolah data dari blok *serial receive* dan mengirimkannya kembali melalui blok *serial send*.

3.4. Identifikasi dan Pemodelan Sistem

Untuk melakukan pengaturan pada sistem dibutuhkan karakteristik model dari sistem yang telah dibangun. Model yang dimaksud dapat

berupa model matematika sebagai representasi karakteristik dari perilaku sistem. Metode identifikasi pemodelan sistem yang digunakan pada Tugas Akhir ini dilakukan secara dinamis dengan cara memberikan masukan berupa sinyal acak atau biasa disebut sinyal PRBS (*Pseudo Random Binary Sequence*). Sinyal tersebut akan dimasukkan sebagai sinyal referensi untuk menggerakkan motor BLDC untuk kemudian dilakukan pengidentifikasian dari hasil *output* yang dihasilkan motor BLDC dengan menggunakan metode identifikasi ARX (*Auto Regressive with eXogenous*) yang sudah ada pada *software* MATLAB.

Dalam melakukan identifikasi menggunakan *System Identification Toolbox*, data *input-output* yang telah didapatkan dimasukkan ke dalam aplikasi dengan memilih “*time domain data*” saat meng-klik *popout* bertuliskan “*import data*”. Setelah itu, dilakukan *preprocessing*. Proses yang di pilih pada tahap ini adalah pembuangan nilai rata-rata atau dengan kata lain mengurangi seluruh nilai *input* dan *output* dengan rata-ratanya. Hal ini dapat dilakukan dengan memilih “*remove means*” setelah meng-klik *popout* bertuliskan “*preprocess*”. Data yang telah melalui tahap *preprocess* kemudian dipilih sebagai *working data* sekaligus *validation data* dengan cara *drag-and-drop* kotak yang berisi data tersebut ke kotak *working data* dan *validation data*. Kemudian dilakukan identifikasi model dengan memilih “*transfer function*” setelah meng-klik *popout* bertuliskan “*estimate*”. Fungsi transfer hasil identifikasi dibatasi hanya berupa fungsi transfer orde dua tanpa *zero*. Hasil identifikasi dapat dilihat pada Tabel 3.2.

Tabel 3.2 Identifikasi Sistem

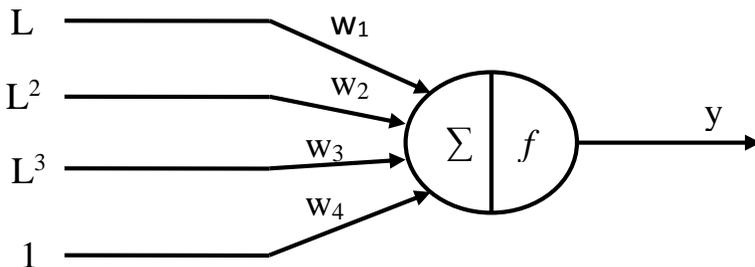
Beban	Tegangan <i>input</i> beban	Fungsi Transfer
Minimal	16 V	$\frac{98,578}{s^2 + 25,3125s + 98,578}$
Nominal	20 V	$\frac{122,2386}{s^2 + 28,9879s + 134,5795}$
Maksimal	24 V	$\frac{232,7394}{s^2 + 49,9617s + 273,8755}$

3.5. Perancangan Kontroler *Neural Network*-PID

Setelah fungsi alih dari sistem didapatkan, langkah selanjutnya adalah merancang kontroler untuk pengaturan kecepatan motor. Kontroler digunakan untuk mengembalikan respon ke nilai *setpoint* yang diinginkan sekalipun motor BLDC diberi beban. Tahapan desain

kontroler ini meliputi perancangan *learning* pada kontroler NN-PID dan perancangan kontroler NN.

Pada Tugas Akhir ini pengaturan kecepatan motor BLDC akan menggunakan metode kontrol berbasis NN-PID. Kontroler NN-PID yang digunakan menggunakan invers model sebagai target *output* sinyal kontrol yang ingin dicapai. Struktur *Neural Network* dapat dilihat pada Gambar 3.4, dan diagram Blok *Simulink Neural Network-PID* dapat dilihat pada Gambar 3.5.



Gambar 3.4 Struktur *Neural Network*

3.5.1. Perancangan Kontroler *Neural Network*

Setelah melalui proses *learning*, *neural network* memiliki nilai bobot yang optimal. Selanjutnya, kontroler bekerja secara *Feed-Forward Controller* yang dibuat menggunakan 4 node pada *layer input* dan 1 node pada *layer output*. Pada struktur *Neural Network* yang dirancang tidak menggunakan *hidden layer* (*Single Layered Neural Network*). Struktur ini sendiri dibuat dengan harapan agar kontroler *Neural Network* yang dibuat dapat bekerja menyerupai kerja kontroler PID Diskrit.

3.5.2. Perancangan *Learning* pada Kontroler *Neural Network*

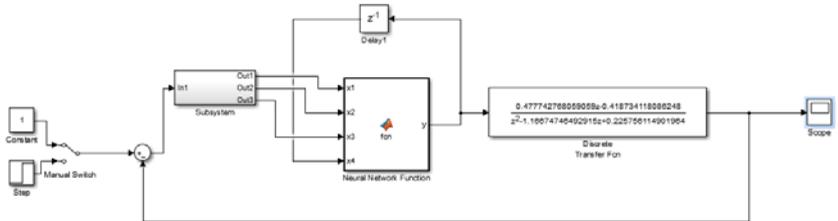
Sebelum merancang kontroler *Neural Network* diperlukan penyesuaian nilai bobot untuk kontroler NN. Dengan merancang *learning* ini maka akan didapatkan nilai bobot yang sesuai untuk output yang diharapkan dari kontroler *Neural Network*.

Pada tahap ini dibuat rancangan *learning* pada kontroler *Neural Network-PID*. Tipe *learning* yang digunakan termasuk ke dalam tipe *reinforced learning*. Artinya, algoritma pembelajaran dari *mapping input-output* secara terus-menerus berinteraksi dengan lingkungan, di mana hal

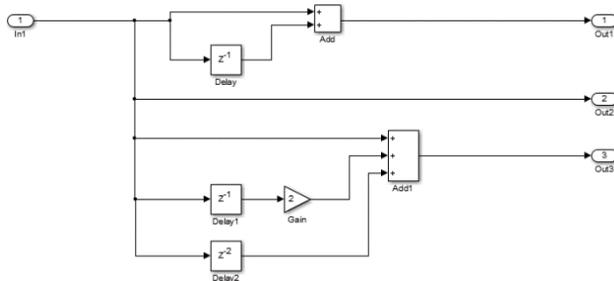
ini dapat memberikan pengaruh dari setiap langkah yang diambil akan memberikan *feedback* kepada model. Dengan *supervised learning* ini, proses *learning* pada *plant* dapat disesuaikan dengan kebutuhan *user*.

Pada proses perancangan *learning Neural Network* dengan memberikan *setpoint* berupa nilai K_p , K_i , dan K_d untuk setiap pembebanan serta dalam proses *learning* memberikan nilai K_p , K_i , dan K_d *random* sehingga dengan metode *learning* ini ditemukan nilai bobot yang tepat untuk masing-masing K_p , K_i , dan K_d pada setiap pembebanan. Script yang digunakan dapat dilihat pada Lampiran 1. Pada tahap ini dibuat rancangan *learning* pada kontroler *Neural Network* dengan menggunakan PID. Tipe *learning* yang digunakan termasuk ke dalam tipe *reinforced learning*. Artinya, Algoritma pembelajaran dari *mapping input-output* secara terus menerus berinteraksi dengan lingkungan, di mana hal ini dapat memberikan pengaruh dari setiap langkah yang diambil, memberikan *feedback* kepada model. Dengan *reinforced learning* ini, proses *learning* pada *plant* dapat disesuaikan dengan kebutuhan *user*. Selain itu dengan menggunakan algoritma, nilai optimal dapat diperoleh dengan waktu yang lebih singkat.

Struktur *Neural Network* yang digunakan dapat dilihat pada Gambar 3.4, dan diagram simulink pada Gambar 3.5 dan Gambar 3.6.



Gambar 3.5 Blok Simulink Neural Network - PID



Gambar 3.6 Blok Simulink Plant Sistem Motor BLDC

Nilai bobot dari masing-masing K_p , K_i , K_d untuk setiap pembebanan dapat dilihat pada Tabel 3.3.

Tabel 3.3 Bobot Setiap Pembebanan

Bobot	Beban			
	L	L ²	L ³	1
wk _p	0,170640084 646581	- 0,09798412285325 68	0,0162135 827314936	0,039518635 3936268
wk _i	0,069126343 288383	- 0,00621583534562 7	-	0,437089489 006012
wk _d	0,000458318 1377225474	- 0,00048000874976 11182	-	0,005093814 537365

(halaman ini sengaja dikosongkan)

BAB 4

PENGUJIAN DAN ANALISA

4.1. Gambaran Umum Pengujian Sistem

Pengujian sistem dimaksudkan untuk mengetahui performansi sistem dari *plant* motor BLDC terhadap adanya beban tambahan dengan menggunakan rem elektromagnetik. Pengujian sistem dengan dua cara, yaitu simulasi dan implementasi.

Simulasi dilakukan dengan bantuan program simulink yang ada pada *software* MATLAB. Simulasi dilakukan pada semua kondisi beban yakni kondisi tanpa beban, kondisi beban nominal, serta kondisi beban maksimal untuk melihat bagaimana pengaruh kontroler *Neural Network* yang telah dirancang.

Setelah dilakukan pengujian berupa simulasi kemudian dilakukan pengujian dengan melakukan implementasi kontroler yang telah dirancang terhadap *plant* motor BLDC yang telah dibuat untuk melihat pengaruh kontroler secara langsung untuk kemudian dilakukan analisa dan perbandingan antara hasil simulasi dan implementasi yang dijalankan.

4.2. Pengujian Perangkat Keras

Pada Tugas Akhir ini, pengujian perangkat keras dilakukan untuk mengetahui apakah perangkat keras yang digunakan bekerja dan berfungsi pada sistem secara semestinya atau tidak. Pengujian perangkat keras terdiri dari dua buah pengujian, yaitu pengujian *open loop*, dan pengujian rem elektromagnetik

4.2.1. Pengujian *Open Loop* Kecepatan Motor

Pada dasarnya, *input* motor BLDC merupakan tegangan 3 fase yang dihasilkan oleh *driver*. Tegangan 3 fase tersebut kemudian diubah ke dalam bentuk PWM dengan nilai *duty cycle* yang berkisar antara 5% hingga 10%. Nilai dari PWM ini kemudian di *mapping* menjadi nilai *throttle* dari 0-1000 sebagai *input* bagi motor BLDC. *Output* dari motor BLDC ini merupakan kecepatan dalam bentuk RPM yang dibaca oleh rangkaian pembaca kecepatan.

4.2.2. Pengujian Rem Elektromagnetik

Pada Tugas Akhir ini, rem elektromagnetik berfungsi sebagai pemberi efek pembebanan yang dirancang sedemikian rupa sehingga besar kecilnya pembebanan dapat diatur sesuai dengan kondisi pembebanan yang diinginkan. Pengujian ini dilakukan dengan tujuan untuk mengetahui secara langsung bagaimana pengaruh sinyal PWM terhadap besarnya arus DC yang diberikan pada rem elektromagnetik. Pada pengujian ini, pengukuran dilakukan dengan menggunakan amperemeter. Hubungan besarnya PWM dengan arus DC yang dihasilkan tergantung pada kondisi pembebanan.

4.3. Simulasi Sistem

Simulasi sistem dilakukan dengan menggunakan program Simulink yang terdapat pada program MATLAB 2013. Merancang kontroler *Neural Network* (NN) pada diagram Simulink MATLAB.

4.3.1 Simulasi dengan Menggunakan Kontroler *Neural-Network*

Proses kontrol yang ada pada kontroler ini terbagi menjadi dua tahap, yakni proses *learning* dan proses simulasi. Pada proses *learning*, struktur *Neural Network* akan bekerja untuk meminimalisir *error* yang ada dengan cara melakukan perbaikan bobot yang ada pada struktur tersebut. Operasi yang dilakukan oleh kontroler *Neural Network* bekerja dalam dua tahapan yakni *forward propagation* dan *Reinforced Learning*. *Forward propagation* merupakan proses maju ke depan melalui struktur jaringan dengan hasil akhir sebuah sinyal kontrol. Di akhir operasi, sinyal kontrol tersebut akan dibandingkan dengan target *output* sinyal kontrol yang telah ditentukan sebelumnya. Ketika target *output* sinyal kontrol belum sesuai dengan *output* yang diinginkan, maka sistem akan melakukan perbaikan bobot untuk meminimalisir *error* yang ada.

4.3.2 Proses *Learning* Kontroler *Neural-Network*

Sebelum melaksanakan simulasi terlebih dahulu dilakukan proses *learning* oleh program dari *Neural Network* tersebut. Pada proses *learning* ini jumlah *epoch* yang diinginkan disesuaikan dengan banyaknya proses perulangan yang diinginkan. Jumlah *epoch* ini berpengaruh terhadap perubahan bobot yang terjadi. Semakin kecil *error* yang diinginkan dan banyaknya *epoch* atau perulangan yang dilakukan, maka akan membuat hasil semakin baik. Namun apabila jumlah *epoch*

yang dilakukan semakin ditingkatkan, maka proses *learning* yang berlangsung akan semakin lama.

Setelah proses *learning* selesai, kemudian dilanjutkan dengan proses penggunaan nilai bobot yang telah didapat ke dalam simulasi *plant* BLDC untuk mendapatkan hasil respon *output* dari *plant* BLDC.

4.3.3 Proses Simulasi Kontroler

Dari hasil *learning* yang dilakukan sebelumnya didapat nilai bobot yang berasal dari nilai bobot *error* yang terkecil. Nilai bobot tersebut digunakan untuk menjalankan kontroler *Neural Network*.

4.3.4 Pengujian Simulasi Respon dengan Kontroler

Pada tahapan ini kontroler yang telah dibuat akan dicoba dijalankan pada model hasil identifikasi sistem. Pengujian ini dilakukan dengan memberikan respon *step* pada semua kondisi pembebanan, mulai dari kondisi tanpa beban (rem 0%) sampai dengan kondisi beban maksimal. Setelah itu respon dianalisa untuk dicari nilai *rise time*, *settling time*, dan *error steady state*. Pengujian dilakukan dengan memberikan sinyal *step* pada semua fungsi alih pada setiap pembebanan.

Setelah dilakukan simulasi, maka dilakukan analisa terhadap respon hasil dari kontroler. Selanjutnya adalah menghitung *settling time*, *rise time* dan *error steady state*.

Perhitungan nilai *settling time* dilakukan dengan cara menghitung nilai respon *output* yang berada pada $\pm 5\%$ atau $\pm 2\%$ atau $\pm 0,5\%$ dari nilai *steady state* respon, sedangkan pada penghitungan *rise time* yang perlu dilakukan adalah menghitung waktu yang dibutuhkan oleh respon dari mulai 10% dari nilai *steady state* hingga ke 90% dari nilai *steady state*.

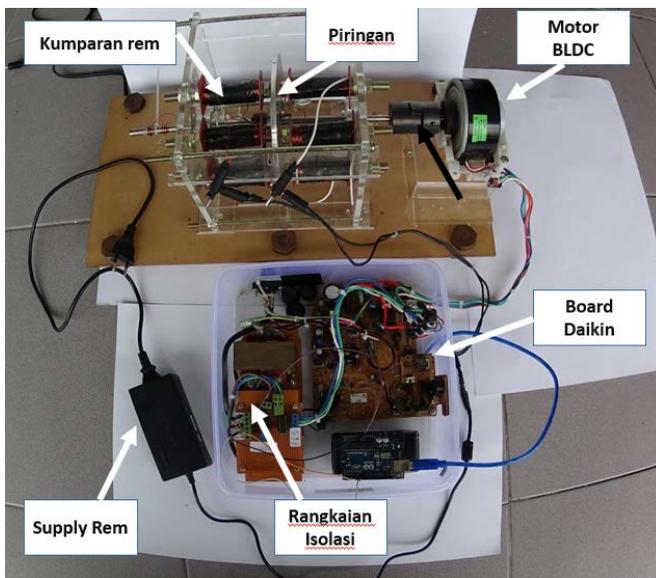
4.4. Realisasi *Plant* Sistem Motor BLDC

Secara garis besar, *plant* sistem motor BLDC ini terdiri dari beberapa komponen inti, di antaranya motor BLDC, rem elektromagnetik, serta rangkaian elektronik pendukung seperti *driver* rem elektromagnetik, rangkaian sinyal *conditioning*, dan Arduino Uno sebagai *interface* serta sebagai penghubung antara *plant* motor BLDC dengan komputer. Dimensi fisik dari *plant* motor BLDC ini sekitar 60 x 30 x 20 cm. Hasil akhir realisasi *plant* motor BLDC dapat dilihat pada Gambar 4.1.

Pada *plant* BLDC ini, motor BLDC yang digunakan adalah motor BLDC yang biasa digunakan pada perangkat *Air Conditioner*

(AC/Pendingin Ruangan). Motor BLDC tersebut merupakan hasil pabrikan dari produsen pendingin ruangan, yaitu Daikin D43F. Motor jenis ini memiliki karakteristik dengan tipe *inrunner*. Memiliki tegangan nominal 307 VDC, dengan kecepatan tanpa beban maksimal pada kisaran 1282 rpm.

Dilihat dari struktur fisik motor yang cukup lebar, biasanya motor jenis ini memiliki kecepatan putar yang cukup rendah, namun memiliki torsi yang besar. Pada Gambar 4.1 dapat dilihat tampilan realisasi *plant* motor BLDC yang dibuat. Beberapa komponen yang digunakan, yaitu motor BLDC, piringan aluminium, kumparan rem, *supply* rem, rangkaian isolasi, *board* daikin.



Gambar 4.1 Bentuk Realisasi Motor BLDC

4.4.1. Pengujian Respon Implementasi Kontroler *Neural Network*

Sebelum melakukan tahap implementasi, terlebih dahulu dilakukan penentuan parameter yang akan digunakan pada kontroler *Neural Network*.

Setelah parameter ditentukan, selanjutnya sistem dijalankan dengan menggunakan kontroler *Neural Network* yang sebelumnya telah melalui proses *learning* dan kemudian di analisa hasil respon yang didapat pada semua kondisi pembebanan.

Hasil respon yang tidak stabil tersebut dapat disebabkan oleh beberapa faktor. Faktor pertama adalah adanya masalah pada pemodelan. Seperti yang telah dijelaskan pada sub bab sebelumnya terkait pemodelan motor BLDC, disana terdapat nilai RMSE yang cukup besar pada fungsi alih hasil pemodelan masing-masing orde pada tiap kondisi pembebanan. Hal ini juga dapat mempengaruhi kondisi kontroler *Neural Network* yang sebelumnya melalui tahap *learning* dengan menggunakan fungsi alih hasil pemodelan tersebut, sehingga ketika proses implementasi dilakukan terdapat ketidaksesuaian nilai bobot yang dapat menyebabkan terjadinya *error steady state*.

Faktor selanjutnya yang mungkin ialah adanya gangguan teknis pada pengiriman dan penerimaan data (*data loss*) yang terjadi antara komputer dengan *plant* BLDC. *Data loss* ini mengganggu sinyal kontrol yang dikirimkan oleh komputer ke *plant* motor BLDC P-1 karena data yang dikirimkan akan hilang untuk beberapa saat. Sehingga PWM yang dihasilkan oleh *driver* ke motor BLDC tidak sesuai dengan apa yang diinginkan. Faktor yang dapat menyebabkan terjadinya *data loss* ini dapat berasal dari gangguan teknis berupa gangguan proses yang dihasilkan *traffic data* karena ada program lain yang juga sedang bekerja sehingga mempengaruhi performa komputer dalam memproses data.

4.4.2. Respon pada Semua Kondisi Pembebanan

Pada sub bab ini akan dipaparkan karakteristik dari respon hasil implementasi kontroler *Neural Network*-PID pada semua kondisi.

4.4.2.1. Kondisi Beban Minimal

Pada kondisi tanpa beban, pengereman diatur sebesar 0% dari kemampuan maksimal rem atau setara dengan 0 V tegangan masuk ke rem elektromagnetik. Pada tahap ini, diperhatikan bagaimana respon yang diperoleh. Parameter yang digunakan oleh kontroler *Neural Network* seperti yang telah disebut di tabel sebelumnya.

4.4.2.2. Kondisi Beban Nominal

Selanjutnya tahap implementasi dilanjutkan untuk kondisi pembebanan nominal. Sebelum dilakukan implementasi, terlebih dahulu dilakukan proses *learning* dengan menggunakan fungsi alih hasil pemodelan pada kondisi beban nominal orde 1. Pemilihan orde ini berdasarkan pada nilai RMSE yang dihasilkan tiap orde. Fungsi alih orde 1 diperoleh pada tiap kondisi pembebanan sebagai fungsi alih dengan nilai RMSE yang paling kecil atau dengan kata lain paling mendekati hasil yang sebenarnya jika dibandingkan dengan fungsi alih orde lain.

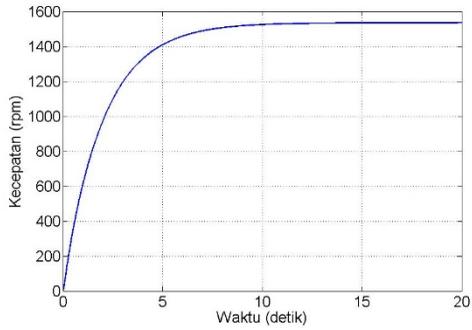
4.4.2.3. Kondisi Beban Maksimal

Pada tahap implementasi kondisi ini, beban pengereman diatur pada kondisi beban maksimal. Selanjutnya dilakukan proses *learning* dengan menggunakan fungsi alih hasil pemodelan pada kondisi beban maksimal orde 1. Setelah proses *learning* pada kondisi ini selesai dilanjutkan dengan implementasi dengan memberikan kecepatan referensi sama seperti pada kondisi pembebanan sebelumnya. Kemudian didapatkan hasil respon dan dilakukan analisa terhadap karakteristik yang ada pada kondisi beban maksimal.

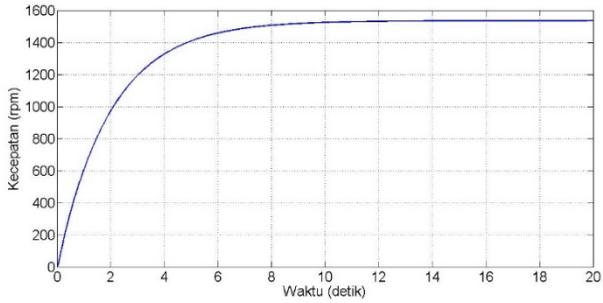
4.5. Hasil Simulasi

Simulasi dilakukan dengan mengubah-ubah nilai beban yang masuk pada *plant*. Dari simulasi yang dilakukan, maka didapatkan bentuk respon seperti pada Gambar 4.2, Gambar 4.3, dan Gambar 4.4.

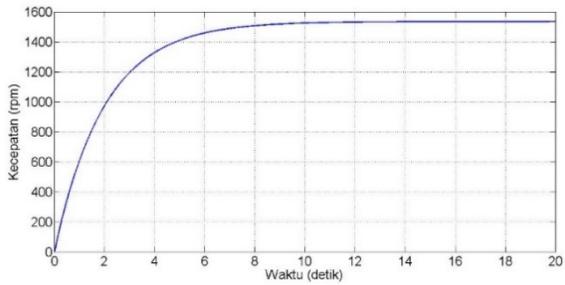
Pada Gambar 4.2 dapat dilihat bahwa respon *plant* ketika diberikan beban minimal dapat mengikuti respon *step* yang ditentukan. *Rise time* untuk beban minimal sebesar 6,1 detik.



Gambar 4.2 Respon *Plant* Beban Minimal



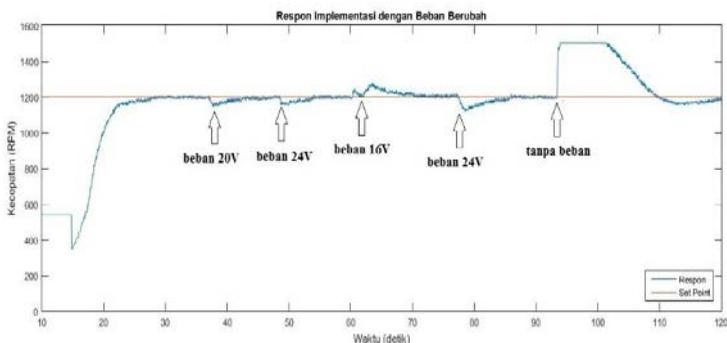
Gambar 4.4 Respon *Plant* Beban Maksimal



Gambar 4.3 Respon *Plant* Beban Nominal

Dapat dilihat dari ketiga gambar, Gambar 4.2, Gambar 4.3, dan Gambar 4.4 di atas bahwa pada saat diberikan beban minimal, nominal, dan maksimal pada *plant*, maka respon yang dihasilkan dapat *rise time* setelah 6,1 detik. *Steady state time* yang diperlukan sebesar 6,3 detik.

4.6. Implementasi



Gambar 4.5 Respon *Plant* Implementasi

Pada kondisi awal, motor diberikan beban nominal (20VDC). Pada saat $t=40$ detik beban diganti menjadi beban maksimal (24VDC). Pada Gambar 4.5 dapat dilihat bahwa ketika diberikan beban nominal maka kecepatan akan *steady* pada waktu 24 detik, lalu beban diubah menjadi beban maksimal ketika 40 detik dan membuat kecepatan turun, lalu setelah beberapa saat akan *steady* kembali. Setelah *steady* maka beban diubah menjadi beban minimal, sehingga kecepatan naik setelah beberapa saat, maka kecepatan akan *steady* kembali.

BAB 5 PENUTUP

5.1. Kesimpulan

Dari hasil respon *plant* pada pembebanan beban minimal, nominal, dan maksimal, maka dapat disimpulkan bahwa motor dapat mempertahankan kecepatan.

Dari hasil *tuning* kontroler PID menggunakan *Neural Network* didapatkan parameter paling optimal sebagai berikut, $K_p = 0,238$, $K_i = 1,148$, $K_d = 0,013$. *Rise time* untuk beban minimal sebesar 6,1 detik, beban nominal sebesar 6,11 detik, dan beban maksimal sebesar 6,15 detik. *Steady state time* untuk ketiga beban sebesar 6,3 detik.

Ketika dilakukan perubahan pembebanan pada saat implementasi, kecepatan pada *plant* dapat berada pada *steady state*.

5.2. Saran

Apabila ada yang ingin menggunakan *plant* motor BLDC ini untuk dilakukan penelitian lebih lanjut atau menggunakan kontroler *Neural Network*, ada beberapa hal yang dapat dijadikan pertimbangan, yaitu:

Faktor pendukung yang dapat mempengaruhi kecepatan harus diperhatikan. Faktor-faktor tersebut secara langsung maupun tidak langsung dapat mempengaruhi kecepatan putaran motor BLDC

(halaman ini sengaja dikosongkan)

DAFTAR PUSTAKA

- [1] Xia, Chang Liang., *Permanent Magnet Brushless DC Motor Drives and Controls*, John Wiley & Sons Singapore Pte. Ltd., Singapore, 2012
- [2] R. Shemanske, "Electronic Motor Braking," *IEEE Trans. On Industry*, vol. IA, no. 19, pp. 824-831, 1983.
- [3], "Arduino", <URL: <http://arduino.cc/>>, Desember, 2015.
- [4] Ljung, Lenart dan Torkel, Glad, "*Modelling of Dynamic Systems*", Prentice – Hall International: New Jersey. 1994.
- [5] Elrosa, Ilmiyah, "Traction Control pada Parallel Hybrid Electric Vehicle dengan Metode Generalized Predictive Control", *Tugas Akhir*, Jurusan Teknik Elektro ITS Surabaya, 2014.
- [6] Gerard Dreyfus, *Neural Networks, Methodology and Applications*. Springer, 2005.
- [7] Permana, Putra Eka, "*PID (Proportional-Integral-Derivative) Controller*", <URL: <https://putraekapermana.wordpress.com/2013/11/21/pid/>>, November, 2013.

(halaman ini sengaja dikosongkan)

LAMPIRAN

1. Program Bobot Kp

```
function kpn = kpjst(xx)
global wkp lmd alp
L=xx(1);
Kp=xx(2);
tt=xx(3);
if tt==0
    wkp=rand(4,1);
% wkp=evalin('base','wkp');
    alp=0.00001;
    lmd=1;
end
for eppoh=1:10
    zkp=wkp(1)*L+wkp(2)*L^2+wkp(3)*L^3+wkp(4)*1;
    kpn=lmd*zkp;
    ekp=Kp-kpn;
    wkp(1)=wkp(1)+alp*ekp*lmd*L;
    wkp(2)=wkp(2)+alp*ekp*lmd*L^2;
    wkp(3)=wkp(3)+alp*ekp*lmd*L^3;
    wkp(4)=wkp(4)+alp*ekp*lmd*1;
    assignin('base','wkp',wkp);
end
end
```

2. Program Bobot Ki

```
function kin = kijst(xx)
global wki lmd alp
L=xx(1);
Ki=xx(2);
tt=xx(3);
if tt==0
    wki=rand(3,1);
% wki=evalin('base','wki');
    alp=0.000000001;
    lmd=1;
end
```

```

for eppoh=1:10
    zki=wki(1)*L+wki(2)*L^2+wki(3)*1;
    kin=lmd*zki;
    eki=Ki-kin;
    wki(1)=wki(1)+alp*eki*lmd*L;
    wki(2)=wki(2)+alp*eki*lmd*L^2;
    wki(3)=wki(3)+alp*eki*lmd*1;
    assignin('base','wki',wki);
end
end

```

3. Program Bobot Kd

```

function kdn = kdjst(xx)
global wkd lmd alp
L=xx(1);
Kd=xx(2);
tt=xx(3);
if tt==0
wkd=rand(3,1);
% wkd=evalin('base','wkd');
alp=0.001;
lmd=1;
end
for eppoh=1:10
    zkd=wkd(1)*L+wkd(2)*L^2+wkd(3)*1;
    kdn=lmd*zkd;
    ekd=Kd-kdn;
    wkd(1)=wkd(1)+alp*ekd*lmd*L;
    wkd(2)=wkd(2)+alp*ekd*lmd*L^2;
    wkd(3)=wkd(3)+alp*ekd*lmd*1;
    assignin('base','wkd',wkd);
end
end

```

RIWAYAT HIDUP



Tri Wahyu Kurniawan lahir di Surabaya pada tanggal 22 Mei 1993. Menyelesaikan pendidikannya di SDN Ketabang 1 No. 288 Surabaya (1999-2005), SMP Muhammadiyah 5 Surabaya (2005-2008), dan SMA Muhammadiyah 2 Surabaya (2008-2011). Pada tahun 2011, melanjutkan pendidikan di Institut Teknologi Sepuluh Nopember (ITS), Jurusan Teknik Elektro Surabaya. Di Jurusan Teknik Elektro ini, kemudian memilih Teknik Sistem Pengaturan sebagai bidang studi yang ditekuni. Pada bulan Januari 2016, mengikuti seminar dan ujian Tugas Akhir sebagai salah satu persyaratan untuk memperoleh gelar Sarjana Teknik

(halaman ini sengaja dikosongkan)