TUGAS AKHIR – TE 141599

PERANCANGAN KONTROL KECEPATAN MOTOR ARUS SEARAH TANPA SIKAT MENGGUNAKAN *SLIDING MODE* BERBASIS PID

Guntur Sadhiea Putra
NRP. 2211 100 075

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

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

***FINAL PROJECT* – TE 141599**

SPEED CONTROL DESIGN OF BRUSHLESS DIRECT CURRENT (BLDC) MOTOR USING SLIDING MODE BASED ON PID

Guntur Sadhiea Putra
NRP. 2211 100 075

Supervisor
Ir. Rusdhianto Effendie A.K., MT.
Ir. Ali Fatoni, MT.

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

PERANCANGAN KONTROL KECEPATAN MOTOR ARUS SEARAH TANPA SIKAT MENGGUNAKAN *SLIDING MODE* BERBASIS PID

TUGAS AKHIR

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

Menyetujui :

Dosen Pembimbing I	Dosen Pembimbing II
Ir. Rusdhianto Effendie A.K., MT.	**Ir. Ali Fatoni, MT.**
NIP. 19570424 198502 1 001	NIP. 19620603 198903 1 002

SURABAYA
JANAURI, 2016

PERANCANGAN KONTROL KECEPATAN MOTOR ARUS SEARAH TANPA SIKAT MENGGUNAKAN *SLIDING MODE* BERBASIS PID

Guntur Sadhiea Putra
2211 100 075

Dosen Pembimbing I : Ir. Rusdhianto Effendie A.K., MT.
Dosen Pembimbing II : Ir. Ali Fatoni, MT.

ABSTRAK

Brushless DC (BLDC) motor telah menjadi aplikasi industri yang paling cepat berkembang dan paling menjanjikan. BLDC motor ini berisi struktur sederhana, handal, dengan perawatan mudah, umur yang panjang, dan juga mengandung sifat mekanik dan kinerja yang baik. Oleh karena itu, sistem kontrol kecepatan motor sangat penting dalam penggunaan motor dalam suatu industri. *Sliding Mode Control* (SMC) merupakan salah satu metode yang sering digunakan untuk mengatasi ketidakpastian suatu sistem kontrol. Keunggulan dari SMC terdapat pada *robustness* terhadap berbagai variasi parameter dan gangguan dari luar. *Sliding Mode Control* sendiri *robust* terhadap ketidakpastian dari *plant* dan tidak sensitif terhadap gangguan dari luar. Metode ini biasa digunakan untuk mendapatkan performansi yang baik. Namun, adanya fenomena *chattering* dapat menimbulkan kerusakan pada aktuator ataupun pada *plant*. Sehingga digunakan metode PID untuk menghilangkan adanya fenomena *chattering* tersebut. Nilai parameter Kp, Ki, dan Kd sebesar 3, 1,25, dan 1. Setelah dilakukan simulasi, didapatkan bahwa respon *plant* dengan kontroler Sliding Mode-PID dapat mengikuti model referensi yang diinginkan dengan nilai *rise time* 5,053 untuk beban minimal, 5,05 untuk beban nominal, dan 5,049 untuk beban maksimal serta *chattering* pada *sliding surface* hilang.

Kata Kunci : *Brushless DC*, *Sliding Mode, PID*

SPEED CONTROL DESIGN OF BRUSHLESS DIRECT CURRENT (BLDC) MOTOR USING SLIDING MODE BASED ON PID

Guntur Sadhiea Putra
2211 100 075

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

ABSTRACT

Brushless DC (BLDC) motors have become industrial applications of the fastest growth and the most promising motor. BLDC motors contains a simple structure, reliable, with easy maintenance, long life, and also contains the mechanical properties and good performance. Therefore, the motor speed control system is very important in the use of the motor in an industry. Sliding Mode Control (SMC) is one method that is often used to cope with the uncertainty of a control system. The advantages of the SMC are the robustness against parameter variations and disturbances from outside. Sliding Mode Control itself robust to the uncertainty of the plant and is not sensitive to interference from outside. This method is used to get good performance. However, the chattering phenomenon can cause damage to the actuator or on the plant. So the PID method is used to eliminate the chattering of these phenomena. Parameter values of Kp, Ki, and Kd is 3, 1.25, *and* 1. *After the simulation, it was found that the plant's response to the Sliding Mode-PID controller can follow the desired reference model with a rise time value of* 5.053 *for the minimum load,* 5.05 *for nominal load, and* 5.049 *for the maximum load and chattering in the sliding surface is gone.*

Keywords **:** *Brushless DC*, *Sliding Mode, PID*

KATA PENGANTAR

Alhamdullilah, puji syukur penulis panjatkan kehadirat Allah SWT karena atas rahmat dan karunia-Nya penulis dapat menyelesaikan penulisan buku tugas akhir dengan judul **"PERANCANGAN KONTROL KECEPATAN MOTOR ARUS SEARAH TANPA SIKAT MENGGUNAKAN *SLIDING* MODE BERBASIS PID"**. Tugas akhir merupakan salah satu syarat yang harus dipenuhi untuk menyelesaikan program studi Strata-1 pada Jurusan Teknik Elektro Fakultas Teknologi Industri Institut Teknologi Sepuluh Nopember Surabaya.

Penulis menyadari bahwa dalam penulisan tugas akhir ini banyak mengalami kendala, namun berkat bantuan, bimbingan, dan kerja sama dari berbagai pihak sehingga kendala-kendala tersebut dapat di atasi. Untuk itu pada kesempatan ini penulis menyampaikan banyak terimakasih dan penghargaan *set*ingi-tingginya kepada :

1. Allah SWT. karena telah memberi hidayah serta inayah-Nya dalam menyelesaikan Tugas Akhir ini.
2. Kedua orang tua, Ayahanda Edy Yuwono dan Ibunda Sudarwati yang selalu memberikan dukungan, semangat, dan doa kepada penulis.
3. Bapak Rusdi dan Bapak Ali selaku Dosen Pembimbing atas segala bantuan, perhatian, dan arahan selama pengerjaan tugas akhir ini.
4. Bapak Rusdi selaku Koordinator Bidang Studi Sistem Pengaturan Jurusan Teknik Elektro ITS.
5. Bapak Ardyono selaku Ketua Jurusan Teknik Elektro ITS.
6. Rekan-rekan e51 khususnya bidang studi Sistem Pengaturan.
7. Teman-teman seperjuangan, dan teman satu kontrakan.

Penulis berharap tugas akhir ini dapat bermanfaat bagi yang membutuhkannya.

Surabaya, 16 Desember 2015

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN ..... **i**
ABSTRAK ..... **iii**
ABSTRACT ..... **v**
KATA PENGANTAR ..... **vii**
DAFTAR ISI ..... **ix**
DAFTAR GAMBAR ..... **xi**
DAFTAR TABEL ..... **xiii**
BAB 1 PENDAHULUAN **1**
1.1 Latar Belakang 1
1.2 Perumusan Masalah 1
1.3 Batasan Masalah 2
1.4 Tujuan Penelitian 2
1.5 Sistematika Penulisan 3
1.6 Relevansi 3
BAB 2 TINJAUAN PUSTAKA **5**
2.1 Motor *Brushless* DC 5
2.1.1 Cara Kerja Motor *Brushless* DC 6
2.1.2 Konstruksi Motor *Brushless* DC 8
2.1.3 Motor BLDC Daikin 9
2.2 Rem Elektromagnetik 10
2.3 Rangkaian *Optocoupler* 12
2.4 Arduino 12
2.4.1 Pengertian Arduino 12
2.4.2 *Input*/*Output* Arduino 13
2.4.3 *Pin-pin* Catu Daya 13
2.4.4 Soket Baterai 13
2.5 MATLAB 14
2.6 Identifikasi Sistem 15
2.7 Kontroler PID 16
2.8 *Sliding Mode Control* 20
BAB 3 PERANCANGAN SISTEM **25**
3.1 Gambaran Umum Sistem 25
3.2 Perancangan Perangkat Keras 26
3.2.1 Perancangan Mekanik 27
3.2.2 Perancangan Elektronik 28

3.3 Perancangan Perangkat Lunak 29
3.3.1 Perangkat Lunak Matlab 29
3.3.2 *Software* Arduino 30
3.4 Identifikasi dan Pemodelan Sistem 31
3.5 Perancangan Kontroler *Sliding Mode* berbasis PID 33
3.5.1 Perancangan Kontroler PID 33
3.5.2 Perancangan Kontroler *Sliding Mode* 33
BAB 4 PENGUJIAN DAN ANALISA 35
4.1 Pengujian *Sensor* Kecepatan Motor BLDC 35
4.2 Pengujian Open Loop Kecepatan Motor 36
4.3 Pengujian Simulasi Kontroler 36
4.3.1 Blok Diagram *Simulink* 37
4.3.2 Hasil dan Analisa Simulasi 39
4.3.3 Analisis *Sliding Surface* pada *Sliding Mode Control* 41
4.4 Implementasi Sistem 42
4.4.1 Realisasi *Plant* 42
4.4.2 Implementasi Kontroler *Sliding Mode – PID* pada *Plant* P-1 44
BAB 5 PENUTUP 47
5.1. Kesimpulan 47
5.2. Saran 47
DAFTAR PUSTAKA 49
LAMPIRAN 51

TABLE OF CONTENT

ABSTRAK ..... **i**
ABSTRACT ..... **iii**
PREFACE ..... **v**
TABLE OF CONTENT ..... **vii**
LIST OF FIGURES ..... **ix**
LIST OF TABLES ..... **xi**
CHAPTER 1 INTRODUCTION **1**
1.1 Background 1
1.2 Problem Formulation 1
1.3 Scope of Problem 2
1.4 Research Purpose 2
1.5 Writing Sistematics 3
1.6 Relevance 3
CHAPTER 2 LITERATURE **5**
2.1 Brushless DC Motor 5
2.1.1 Work Principle of Motor Brushless DC 6
2.1.2 Construction of Motor Brushless DC 8
2.1.3 BLDC Daikin Motor 9
2.2 Electromagnetic Brake 10
2.3 Optocoupler Circuit 12
2.4 Arduino 12
2.4.1 Definition of Arduino 12
2.4.2 Input/Output Arduino 13
2.4.3 Power Supply Pin 13
2.4.4 Battery Socket 13
2.5 MATLAB 14
2.6 System Identification 15
2.7 PID Controller 16
2.8 Sliding Mode Control 20
CHAPTER 3 SYSTEM DESIGN **25**
3.1 General Overview of the System 25
3.2 Hardware Design 26
3.2.1 Mechanical Design 27
3.2.2 Electronic Design 28
3.3 Software Design 29
3.3.1 Software Matlab 29

3.3.2 Software Arduino 30
3.4 System Identification and Modelling 31
3.5 PID Based Sliding Mode Control Design..... 33
3.5.1 PID Controller Design 33
3.5.2 Sliding Mode Controller Design 33
CHAPTER 4 EXAMINATION AND ANALYSIS 35
4.1 BLDC Motor Speed Sensor Testing..... 35
4.2 Motor Speed Open Loop Testing 36
4.3 Simulation Controller Testing..... 36
4.3.1 Simulink Block Diagram 37
4.3.2 Simulation Results and Analysis..... 39
4.3.3 Sliding Surface Analysis in Sliding Mode Control..... 41
4.4 System Implementation 42
4.4.1 Plant Realization..... 42
4.4.2 Sliding Mode – PID Controller Implementation in Plant P-1 44
CHAPTER 5 CONCLUSION..... 47
5.1. Conclusion 47
5.2. Suggestion..... 47
BIBLIOGRAPHY 49
ENCLOSURE 51

DAFTAR GAMBAR

Gambar 2.1 *Brushless* DC Motor 6
Gambar 2.2 Skema Kerja *Brushless* DC Motor 6
Gambar 2.3 *Inner Rotor Brushless* DC Motor 8
Gambar 2.4 *Outer Rotor Brushless* DC Motor 9
Gambar 2.5 Prinsip Arus *Eddy* Pada Logam yang Bergerak 11
Gambar 2.6 Rangkaian *Optocoupler* 12
Gambar 2.7 Arduino Uno 13
Gambar 2.8 Tampilan Sinyal PRBS 15
Gambar 2.9 Blok Diagram Kontroler PID 16
Gambar 2.10 Diagram Fasa Trajektori Status 20
Gambar 3.1 Blok Diagram Sistem Pengaturan Kecepatan Motor *Brushless DC* (BLDC). 25
Gambar 3.2 Konfigurasi Perangkat Keras Sistem Pengaturan Kecepatan Motor *Brushless DC* (BLDC). 26
Gambar 3.3 Konfigurasi Perancangan Rem Magnetik 28
Gambar 3.4 Diagram Blok *Simulink* Identifikasi 30
Gambar 3.5 Tampilan Arduino IDE 31
Gambar 3.6 Tampilan *System Identification Toolbox* 32
Gambar 4.1 *Input Output* Kecepatan Motor 36
Gambar 4.2 Simulasi Kontroler SMC 37
Gambar 4.3 *Subsystem* Sinyal Kontrol Ekivalen 38
Gambar 4.4 *Subsystem* Sinyal Kontrol Natural 38
Gambar 4.5 Respon *Plant* dengan Beban Minimal 39
Gambar 4.6 Respon *Plant* dengan Beban Nominal 39
Gambar 4.7 Respon *Plant* dengan Beban Maksimal 40
Gambar 4.8 *Sliding Surface* pada Beban Nominal 41
Gambar 4.9 *Sliding Surface* pada Beban Minimal 41
Gambar 4.10 *Sliding Surface* pada Beban Maksimal 42
Gambar 4.11 *Plant* Motor BLDC P-1 43
Gambar 4.13 *Step* Respon Sistem saat Beban Nominal 44
Gambar 4.12 *Step* Respon Sistem saat Beban Minimal 44
Gambar 4.15 *Step* Respon Sistem saat Beban Campuran 45
Gambar 4.14 *Step* Respon Sistem saat Beban Maksimal 45

DAFTAR TABEL

Tabel 2.1 Fungsi Kabel *Input-Output Driver* Motor BLDC 10
Tabel 2.2 Spesifikasi Arduino Uno 13
Tabel 2.3 Karakteristik Kontroler Proporsional, Integral, dan Derivatif 20
Tabel 3.1 Hasil Identifikasi *Plant* 32
Tabel 4.1 *Output* Pembacaan Kecepatan Motor. 35
Tabel 4.2 Indeks Performansi *Plant* saat Simulasi 40
Tabel 4.3 Indeks Performansi *Plant* saat Implementasi 46

BAB 1
PENDAHULUAN

1.1 Latar Belakang

Dalam beberapa tahun terakhir ini, penggunaan motor arus searah tanpa sikat atau biasa disebut dengan *Brushless* DC (BLDC) motor sedang berkembang secara pesat dalam dunia industri, otomotif, otomasi, serta robotika. Hal ini dikarenakan BLDC motor memiliki beberapa keunggulan diantaranya; tingkat efektifitas serta kesederhanaan dari konstruksi motor. Dalam BLDC motor, kontroler memiliki perang penting dalam mempengaruhi performansi dalam aplikasinya di dunia industri. Hal ini menyebabkan timbulnya ketertarikan manusia dalam kontroler cerdas dan adaptif. [1]

Proportional-integral-derivative (PID) banyak digunakan dalam aplikasi kontrol karena sifatnya yang *simple* dan efektif. Meskipun penggunakan kontrol PID mempunyai sejarah yang lama, tiga parameter *gain*, proporsional *gain* (Kp), integral *gain* (Ki), dan derivatif *gain* (Kd), biasanya bernilai tetap. Kelemahan kontroler PID adalah kapabilitasnya rendah dalam menghadapi ketidakpastian sistem, variasi parameter, dan gangguan dari luar. Sehingga *robustness* mendapatkan perhatian lebih. [2]

Sliding mode control (SMC) merupakan salah satu metode yang digunakan untuk mengatasi ketidakpastian sistem kontrol. [3] Keunggulan utama dari SMC adalah tingkat *robustness* dalam menghadapi variasi parameter dan gangguan luar. Berbagai aplikasi SMC telah dilakukan, seperti dalam manipulator robot, *aircrafts*, motor DC, dll. [4]

Sliding mode control robust terhadap ketidakpastian *plant* dan tidak sensitif terhadap gangguan luar. Biasanya digunakan untuk mendapatkan performansi sistem kontrol yang dinamik. Namun, adanya fenomena *chattering* karena terbatasnya kecepatan dari *switching* devices dapat mempengaruhi perilaku sistem secara signifikan. *Sliding mode control* membutuhkan *mode*l matematika sistem dengan ketidakpastian yang dibatasi. Fenomena *chattering* dapat dikurangi tanpa mengorbankan *robust* performansi dengan menggabungkan fitur PID dengan *sliding mode control*. [5]

1.2 Perumusan Masalah

BLDC motor (*Brushless* DC) merupakan motor arus searah tanpa sikat merupakan pengembangan dari motor arus searah namun tanpa sikat sehingga tidak perlu melakukan penggantian sikat. BLDC motor baru-baru ini banyak digunakan dalam dunia industri. Terdapat berbagai pengaturan motor arus searah tanpa sikat diantaranya, pengaturan posisi dan pengaturan kecepatan motor. Terdapat berbagai macam metode kontrol yang dapat digunakan dalam mengatur kecepatan maupun posisi motor.

BLDC motor merupakan suatu sistem *non-linear* dan memiliki kestabilan yang rendah sehingga rentan terhadap gangguan. Sehingga perlu dirancang suatu kontroler yang mampu mengatur kecepatan dari motor BLDC. Dalam tugas akhir ini membahas mengenai kontrol kecepatan pada BLDC motor dengan menggunakan metode *sliding mode control* (SMC) berbasis PID.

1.3 Batasan Masalah

Permasalahan pada tugas akhir ini dibatasi oleh beberapa hal antara lain:

a. Pembebanan yang dilakukan yaitu beban minimal (16 V), beban nominal (20 V), serta beban maksimal (20 V).
b. Perancangan dan implementasi kontroler *Sliding Mode-PID* untuk pengaturan kecepatan motor arus searah tanpa sikat.
c. Nilai parameter Kp, Ki, dan Kd didapatkan dari model referensi yang diinginkan.

1.4 Tujuan Penelitian

Tujuan dari pelaksanaan tugas akhir ini adalah:

a. Mengindentifikasi kinematika dan dinamika BLDC motor sehingga didapatkan *mode*l matematika sistem
b. Merancang kontroler *sliding mode*.
c. Mencari nilai parameter Kp, Ki, dan Kd kontroler PID berdasarkan model referensi yang diinginkan.
d. Menganalisa respon simulasi BLDC motor dengan mensimulasikannya sehingga diketahui tingkat keakuratan dari kontroler

Hasil yang diperoleh dari pelaksanaan tugas akhir ini diharapkan dapat memberikan manfaat dan kontribusi bagi dunia pendidikan,

industri dan masyarakat agar dapat dijadikan referensi bagi peneliti lainnya dan sebagai ilmu pengetahuan.

1.5 Sistematika Penulisan

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

BAB I: PENDAHULUAN

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

BAB 2: DASAR TEORI

Bab ini berisi tentang teori yang menunjang penelitian, berupa teori tentang BLDC dan komponennya, serta metode yang digunakan untuk pengaturan kecepatan motor BLDC.

BAB 3: PERANCANGAN SISTEM DAN KONTROLER

Bab ini berisi tentang perancangan perangkat keras, perangkat lunak, dan perancangan kontroler.

BAB 4: PENGUJIAN DAN ANALISA

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

BAB 5: KESIMPULAN DAN SARAN

Berisi kesimpulan dan saran yang dapat dijadikan pertimbangan pengembangan berdasar hasil pengerjaan tugas akhir ini.

1.6 Relevansi

Hasil dari tugas akhir ini diharapkan dapat dijadikan acuan dalam pembuatan sistem kontrol kecepatan BLDC motor. Dapat dijadikan referensi penelitian lebih lanjut mengenai kontrol kecepatan BLDC motor.

BAB 2
TINJAUAN PUSTAKA

2.1 Motor *Brushless* DC

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

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

Motor BLDC sangat handal karena tidak memiliki sikat yang bisa aus dan harus diganti. Ketika dioperasikan dalam kondisi optimal, usia motor dapat lebih dari 10.000 jam. Untuk aplikasi jangka panjang, hal ini bisa menjadi keuntungan yang besar. *Set*iap kali motor rusak atau perlu diganti, *plant* atau bagian dari *plant* harus dimatikan. Hal ini membutuhkan waktu dan uang, tergantung pada berapa lama waktu yang dibutuhkan untuk mengganti komponen yang aus atau bagian dan rusak agar *plant* dapat berjalan seperti semula. Meskipun motor BLDC memakan biaya lebih dari motor DC dengan sikat, hal ini akan sepadan seiring dengan banyaknya waktu dan uang yang dihabiskan motor DC dengan sikat jika sikat aus. [6]

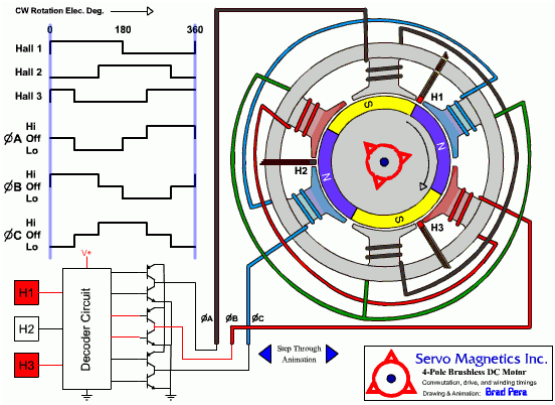
Pada tugas akhir kali ini motor yang kami gunakan adalah motor yang dipakai pada AC Daikin *Inverter*. Dan untuk lebih jelasnya bentuk dari motor ini dapat dilihat pada Gambar 2.1

Gambar 2.1 *Brushless* DC Motor [6]

2.1.1 Cara Kerja Motor *Brushless* DC

Prinsip kerja Motor BLDC sebenarnya sama dengan motor listrik DC konvensional. Perbedaan hanya terletak pada penggunaan *brush* (sikat). Pada motor DC konvensional, sikat dan komutator mekanik digunakan dalam proses komutasi. Sedangkan motor BLDC sudah menggunakan teknologi elektronik dalam proses komutasinya, yaitu *sensor* Hall dan kontroler. [7]

Dan untuk lebih jelasnya, skema kerja dari motor BLDC dapat dilihat pada Gambar 2.2



Gambar 2.2 Skema Kerja *Brushless* DC Motor [7]

Secara garis besar, proses kerja dari motor BLDC dapat dijelaskan dalam Gambar berikut. Motor yang dipakai adalah motor BLDC 3 fasa, berputar searah jarum jam dan *Hall sensor* menggunakan default kutub utara.

Pertama, *hall sensor* H1 dan H3 bernilai 1 karena mengalami perubahan medan magnet. Sehingga kontroler mengalirkan arus pada lilitan B dan C. Lilitan B menjadi kutub utara dan lilitan C menjadi kutub selatan. Kutub utara oleh lilitan B memberikan tolakan pada kutub utara magnet *rotor*, sedangkan kutub selatan lilitan C menarik kutub utara magnet *rotor*.

Langkah kedua, hanya *sensor* H1 yang bernilai "*high*", sehingga kontroler akan menginstruksikan agar lilitan A dan B harus dialiri arus. Lilitan A menghasilkan kutub selatan dan lilitan B tetap menghasilkan kutub utara. Kutub selatan lilitan A akan menolak kutub selatan pada magnet *rotor*. Sedangkan kutub utara lilitan B menolak kutub utara dari magnet *rotor*.

Langkah ketiga, *sensor* H1 dan H2 bernilai 1. Sehingga kontroler menginstruksikan agar lilitan A dan C dialiri arus. Lilitan A tetap menghasilkan kutub selatan dan lilitan C menghasilkan kutub utara. Kutub selatan lilitan A akan menolak kutub selatan dan menarik kutub utara pada magnet *rotor*. Sedangkan kutub utara lilitan C menarik kutub selatan dari magnet *rotor*.

Langkah keempat, hanya *sensor* H2 yang bernilai 1. Sehingga kontroler menginstruksikan agar lilitan B dan C dialiri arus. Lilitan B menghasilkan kutub selatan dan lilitan C tetap menghasilkan kutub utara. Kutub selatan lilitan B menolak kutub selatan pada magnet *rotor*. Sedangkan kutub utara lilitan C menarik kutub selatan dari magnet *rotor*.

Langkah kelima, *sensor* H2 dan H3 bernilai 1. Sehingga kontroler menginstruksikan agar lilitan A dan B dialiri arus. Lilitan A menghasilkan kutub utara dan lilitan B tetap menghasilkan kutub selatan. Kutub utara lilitan A akan menolak kutub utara dan menarik kutub selatan pada magnet *rotor*. Sedangkan kutub selatan lilitan B menolak kutub selatan dari magnet *rotor*.

Langkah keenam atau terakhir pada siklus komutasi, hanya *sensor* H3 yang bernilai 1. Sehingga kontroler menginstruksikan agar lilitan A dan C dialiri arus. Lilitan A tetap menghasilkan kutub utara dan lilitan C menghasilkan kutub selatan. Kutub utara lilitan A akan menarik kutub

selatan dan menolak kutub utara pada magnet *rotor*. Sedangkan kutub selatan lilitan C menarik kutub utara dari magnet *rotor*.

Keenam proses di atas mengalami pengulangan hingga membentuk suatu siklus. Hal inilah yang menyebabkan motor terus berputar secara kontinyu selama sumber arus DC masih ada. [7]

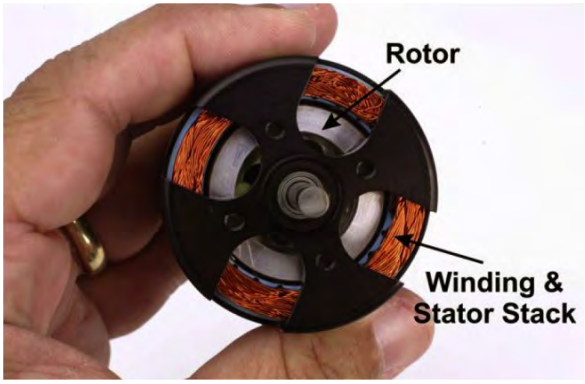
2.1.2 Konstruksi Motor *Brushless* DC

Terdapat dua jenis motor BLDC, dengan keuntungan dan kekurangan yang berbeda. *Set*iap BLDC motor memiliki dua bagian utama; *stator*, dan *rotor*. Bagian penting lain dari motor adalah gulungan *rotor* dan magnet *rotor*. Dua jenis motor BLDC tersebut yaitu *inner rotor* dan *outer rotor*

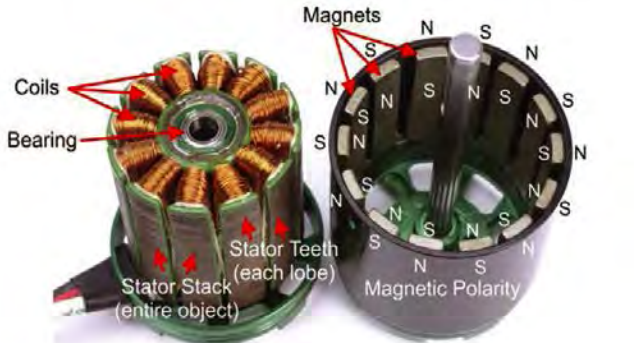
Dalam desain *outer rotor*, gulungan terletak pada inti motor. Magnet *rotor* mengelilingi gulungan *rotor*. Magnet *rotor* bertindak sebagai insulator, sehingga mengurangi laju disipasi panas dari motor. Karena lokasi gulungan *rotor*, desain *outer rotor* biasanya beroperasi pada *duty cycle* rendah. Keuntungan utama dari motor BLDC *outer rotor* adalah torsi *cogging* relatif rendah.

Sedangkan dalam desain *inner rotor*, gulungan *rotor* mengelilingi *rotor* dan ditempelkan pada rumah motor. Keuntungan utama dari konstruksi *inner rotor* adalah kemampuannya untuk menghilangkan panas. Kemampuan motor untuk menghilangkan panas secara langsung berdampak pada kemampuannya untuk menghasilkan torsi. [8]

Bentuk dari *inner rotor* dan *outer rotor* BLDC motor dapat dilihat pada Gambar 2.3 dan Gambar 2.4



Gambar 2.3 *Inner Rotor Brushless* DC Motor [8]



Gambar 2.4 *Outer Rotor Brushless* DC Motor [8]

Rotor merupakan bagian motor yang memiliki lilitan, sehingga apabila pada lilitan *rotor* diberi arus timbul gaya elektromagnetik. Dan *rotor* merupakan bagian motor yang diberi magnet permanen. Oleh karena itu, apabila pada *rotor* diberi arus maka pada *rotor* akan timbul gaya tarik-menarik dan tolak-menolak yang akan menyebabkan motor akan berputar. [9]

Rotor dari BLDC motor terbuat dari baja terlaminasi tersusun untuk membawa gulungan. Gulungan pada *rotor* terdapat dua pola yaitu; pola *star* (Y) atau pola *delta* (Δ). Perbedaan utama antara kedua pola tersebut yaitu pada pola (Y) menghasilkan torsi yang besar pada RPM rendah dan pada pola (Δ) menghasilkan torsi yang kecil pada RPM rendah. Hal ini dikarenakan pada pola (Δ), *set*engah dari tegangan yang ada pada gulungan tidak dikendalikan, sehingga meningkatkan *loss*, pada efisiensi dan torsi. [10]

2.1.3 Motor BLDC Daikin

Dalam perancangan tugas akhir ini, motor BLDC yang digunakan merupakan motor BLDC dari *air conditioner inverter* buatan Daikin. Bentuk fisik dari motor BLDC ini dapat dilihat pada Gambar 2.1. Motor ini memiliki *driver* bawaan di dalam *air conditioner* yang memiliki 5 kabel *input-output*. Tiap kabel memiliki warna dan fungsi yang berbeda. Fungsi dari masing-masing kabel dapat dilihat pada Tabel 2.1.

Tabel 2.1 Fungsi Kabel *Input-Output Driver* Motor BLDC

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

2.2 Rem Elektromagnetik [11]

Rem adalah suatu alat yang digunakan untuk melakukan aksi deselerasi yang akan menurunkan kecepatan dalam selang waktu yang ditentukan. Tipe rem yang umum digunakan adalah rem yang menggunakan gaya gesek untuk memberikan gaya lawan terhadap gaya gerak. Namun pada sistem pengereman elektromagnetik menggunakan gaya elektromagnetik untuk memperlambat suatu gerakan, yang umumnya adalah gerakan poros.

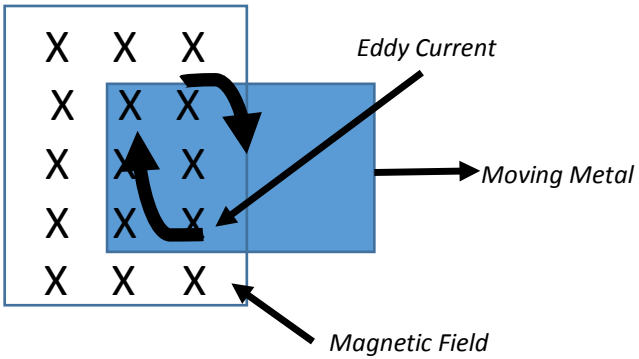
Sebuah piringan dengan bahan logam *non*-feromagnetik terpasang dengan poros yang berputar. Piringan tersebut diapit oleh sisi *rotor* berupa sistem lilitan elektromagnetik yang dapat membangkitkan medan magnet dari aliran listrik. Arus listrik menimbulkan medan magnet pada lilitan dan logam piringan yang memotong medan magnet tersebut akan menimbulkan arus *eddy* pada piringan itu sendiri.

Arus *eddy* ini akan menimbulkan medan magnet yang arahnya berlawanan dengan medan magnet sebelumnya, sehingga menghambat gerakan putar dari poros tersebut. Rem elektromagnetik akan optimal untuk memberikan penurunan kecepatan, bukan untuk menghentikan

gerak suatu objek. Sehingga rem ini sering diaplikasikan untuk sistem pengereman pada *roller coaster*, kereta api dan juga digunakan pada alat dinamometer untuk pengukuran torsi suatu mesin.

Arus *eddy* yang melingkar menyebabkan medan magnet induksi melawan arah medan magnet mula-mula. Hal ini menyebabkan gaya pengereman yang melawan arah kecepatan konduktor yang bergerak memotong medan magnet dari kedua solenoid.

Gaya pengereman yang dihasilkan oleh arus melingkar *eddy* ditunjukkan oleh Gambar 2.5, di mana ada medan magnet yang arahnya menjauhi pengamat. Kemudian sebuah konduktor memotong medan magnet tersebut dengan kecepatan (besar dan arah) tertentu. Berdasarkan hukum Faraday, apabila terjadi perubahan medan magnet, maka akan timbul ggl pada konduktor.

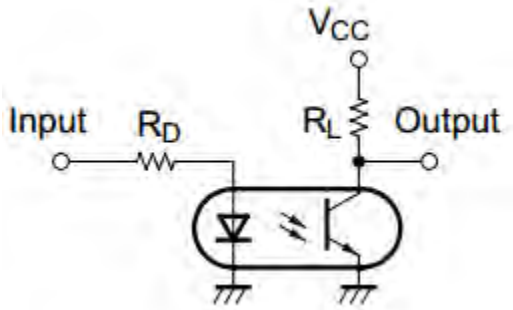


Gambar 2.5 Prinsip Arus *Eddy* Pada Logam yang Bergerak

Pada konduktor, bidang yang mengalami perubahan fluks magnet hanya pada kedua sisinya, yang pertama adalah saat keluar dari medan magnet (fluks magnet yang lewat pada konduktor berkurang) dan yang kedua adalah saat memasuki medan magnet (fluks magnet yang melewati konduktor bertambah). Sedangkan bagian tengah konduktor tidak mengalami perubahan fluks magnet sehingga tidak timbul lagi. Dengan artian, gaya lawan hanya dihasilkan apabila permukaan tersebut memiliki kecepatan. Semakin tinggi kecepatan maka gaya lawan yang dihasilkan juga semakin besar. Namun semakin rendah kecepatan, maka gaya lawan akan semakin kecil.

2.3 Rangkaian *Optocoupler* [12]

Rangkaian *optocoupler* digunakan untuk memisahkan *plant* dengan perangkat Arduino untuk menghindari kerusakan pada perangkat Arduino karena arus berlebih dari *power supply* jika terjadi kecelakaan atau kesalahan dalam memasang *port*. Rangkaian isolator *optocoupler* menggunakan *optocoupler* PIC817 dengan rangkaian seperti pada Gambar 2.6.



Gambar 2.6 Rangkaian *Optocoupler* [12]

2.4 Arduino

2.4.1 Pengertian Arduino [13]

Arduino adalah pengendali mikro *single-board* yang bersifat *open-source*, diturunkan dari *wiring platform*, dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang. *Hardware*nya memiliki prosesor Atmel AVR dan *software*nya memiliki bahasa pemrograman sendiri. Saat ini Arduino sangat populer di seluruh dunia. Banyak pemula yang belajar mengenal robotika dan elektronika lewat Arduino karena mudah dipelajari. Bahasa yang dipakai dalam Arduino bukan *assembler* yang relatif sulit, tetapi bahasa C yang disederhanakan dengan bantuan pustaka-pustaka (*libraries*) Arduino. Arduino juga menyederhanakan proses bekerja dengan mikrokontroler. Salah satu produk Arduino yang digunakan dalam Perancangan Tugas Akhir ini merupakan Arduino Uno.

Arduino Uno adalah *board* mikrokontroler berbasis Atmega328. Alat ini mempunyai 14 *pin input*/*output* digital dan 6 diantaranya 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. Tabel 2.2 menjelaskan mengenai spesifikasi Arduino Uno:

Tabel 2.2 Spesifikasi Arduino Uno

Mikrokontroler	Atmega328
Tegangan operasi	5V
Tegangan *input* (direkomendasikan)	7-12V
Tegangan *input* (batasan)	6-20V
Pin input/output digital	14 (6 diantaranya *output* PWM)
Pin input analog	6
Arus DC tiap *pin* I/O	40mA
Arus DC untuk *pin* 3,3 V	50mA

2.4.2 *Input/Output* Arduino [13]

Input/output digital atau digital *pin* adalah *pin-pin* untuk menghubungkan Arduino dengan komponen atau rangkaian digital. Komponen lain yang menghasilkan *output digital* atau menerima *input digital* bisa disambungkan ke *pin-pin* ini.

2.4.3 *Pin-pin* Catu Daya [13]

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

2.4.4 Soket Baterai [13]

Soket baterai digunakan untuk menyuplai Arduino dengan tegangan dari baterai 9V pada saat Arduino sedang tidak disambungkan ke komputer. Bentuk Arduino Uno sendiri dapat dilihat pada Gambar 2.7

Gambar 2.7 Arduino Uno [13]

2.5 MATLAB [14]

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, per*mode*lan, dan analisis.

Simulink merupakan bagian dari MATLAB untuk memodelkan, mensimulasikan, dan menganalisa sistem dinamik. *Simulink* dapat membentuk *mode*l dari awal atau memodifikasi *mode*l yang sudah ada sesuai dengan apa yang diinginkan. Selain itu *simulink* juga mendukung sistem *linier* dan *non-linier*, pemodelan waktu kontinyu atau diskrit, atau gabungan. *Simulink* ini dapat digunakan sebagai media untuk menyelesaikan masalah dalam industri nyata 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 kunpulan 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 berfugsi 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 dengan *mode*m, *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

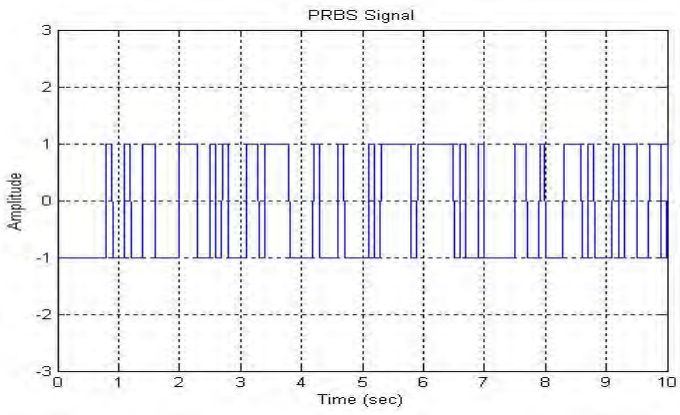
blok *Simulink* yang digunakan untuk mengubah karakter ASCII menjadi data *bytes* yang kemudian dapat dikonversi sesuai kebutuhan.

2.6 Identifikasi Sistem [15]

Identifikasi merupakan suatu metode untuk mendapatkan parameter-parameter dari suatu sistem berdasarkan data hasil pengukuran masukan atau keluaran dari suatu *plant*. Parameter–parameter yang diperoleh digunakan untuk mendapatkan model dari suatu sistem atau *plant.*

Identifikasi dapat dilakukan pada sistem *close loop* maupun *open loop*. Identifikasi *close loop* biasa digunakan pada sistem yang kurang stabil. Sedangkan identifikasi open *loop* lebih sederhana, namun sulit diterapkan untuk sistem–sistem yang memiliki ketidakpastian respon.

Metode identifikasi statis dilakukan dengan menggunakan pendekatan grafis, di mana sinyal uji diberikan pada sistem untuk mengetahui respon *open loop* sistem. Dari respon sistem, dapat diketahui karakteristik-karakteristik dari sistem. Sedangkan identifikasi dinamis, sinyal uji yang digunakan berupa sinyal acak atau sinyal semi acak yang biasa disebut dengan *Pseudo Random Binary Sequence* (PRBS). Pendekatan berbasis waktu dan berbasis frekuensi merupakan identifikasi statis, yaitu identifikasi *plant* dengan memberikan sinyal masukan ke *plant* dengan nilai tertentu dan tetap. Gambar 2.8 merupakan tampilan dari sinyal PRBS.

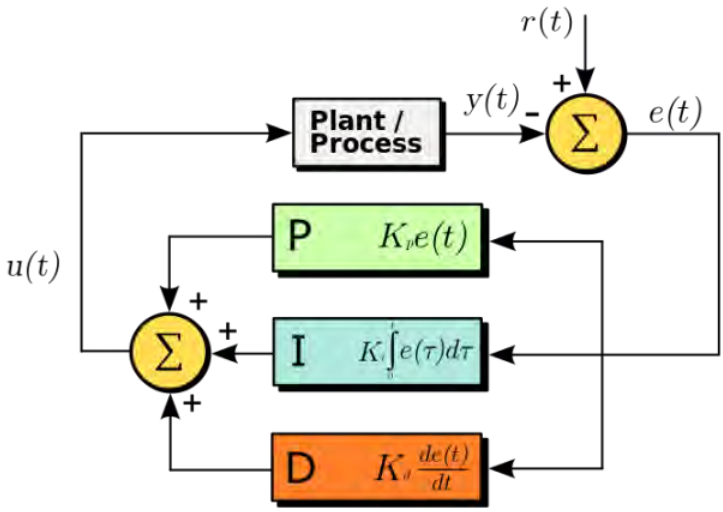


Gambar 2.8 Tampilan Sinyal PRBS [15]

Terdapat beberapa perbedaan mendasar antara identifikasi statis dan dinamis. Diantaranya 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.
Sinyal tersebut dinamakan sinyal *Pseudo-Random Binary Square* (PRBS). Sinyal PRBS seperti pada Gambar 2.8 mirip dengan bilangan acak secara nyata, tapi juga dapat disebut semu atau *pseudo* karena bersifat deterministik.

2.7 Kontroler PID [16]

PID (*Proportional–Integral–Derivative control*ler) merupakan kontroler untuk menentukan presisi suatu sistem instrumentasi dengan karakteristik adanya umpan balik pada sistem tesebut. Pengontrol PID adalah pengontrol konvensional yang banyak dipakai dalam dunia industri. Blok diagram dari kontroler PID dapat dilihat pada Gambar 2.9 di bawah ini:



Gambar 2.9 Blok Diagram Kontroler PID [16]

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 proporsional
Ti : *time* integral
Td : *time* derivatif
e(t) : eror

Komponen kontrol PID ini terdiri dari tiga jenis yaitu proporsional, integratif dan derivatif. Ketiganya dapat dipakai bersamaan maupun sendiri-sendiri tergantung dari respon yang kita inginkan terhadap suatu *plant*.

2.7.1. Kontroler Proporsional

Kontroler proporsional jika G(s) = Kp, dengan Kp adalah konstanta proporsional. Kp berlaku sebagai *gain* (penguat) saja tanpa memberikan efek dinamik kepada kinerja kontroler. Penggunaan kontroler proporsional memiliki berbagai keterbatasan karena sifat kontrol yang tidak dinamik ini. Walaupun demikian dalam aplikasi-aplikasi dasar yang sederhana kontrol P ini mampu untuk memperbaiki respon transien khususnya *rise time* dan *settling time*. Pengontrol proporsional memiliki keluaran yang sebanding/proporsional dengan besarnya sinyal kesalahan (selisih antara besaran yang diinginkan dengan harga aktualnya).

Ciri-ciri pengontrol proporsional:

1. Jika nilai Kp kecil, pengontrol proporsional hanya mampu melakukan koreksi kesalahan yang kecil, sehingga akan menghasilkan respon sistem yang lambat (menambah *rise time*).
2. Jika nilai Kp dinaikkan, respon/tanggapan sistem akan semakin cepat mencapai keadaan mantapnya (mengurangi *rise time*).
3. Namun jika nilai Kp diperbesar sehingga mencapai harga yang berlebihan, akan mengakibatkan sistem bekerja tidak stabil atau respon sistem akan berosilasi.

4. Nilai Kp dapat di*set* sedemikian sehingga mengurangi *steady state* eror, tetapi tidak menghilangkannya.

2.7.2. Kontroler Integratif

Kontroler integral berfungsi untuk menghasilkan respon sistem yang memiliki kesalahan keadaan mantap nol (*Error Steady State* = 0). Jika sebuah pengontrol tidak memiliki unsur integrator, pengontrol proporsional tidak mampu menjamin keluaran sistem dengan kesalahan keadaan mantapnya nol.

Ketika e(T) mendekati konstan (bukan nol) maka u(t) akan menjadi sangat besar sehingga diharapkan dapat memperbaiki eror. Jika e(T) mendekati nol maka efek kontrol I ini semakin kecil. Kontroler integral dapat memperbaiki sekaligus menghilangkan respon *steady-state*, namun pemilihan Ki yang tidak tepat dapat menyebabkan respon transien yang tinggi sehingga dapat menyebabkan ketidakstabilan sistem. Pemilihan Ki yang sangat tinggi justru dapat menyebabkan *output* berosilasi karena menambah orde *system*

Keluaran pengontrol ini merupakan hasil penjumlahan yang terus menerus dari perubahan masukannya. Jika sinyal kesalahan tidak mengalami perubahan, maka keluaran akan menjaga keadaan seperti sebelum terjadinya perubahan masukan. Sinyal keluaran pengontrol integral merupakan luas bidang yang dibentuk oleh kurva kesalahan/ eror.

Ciri-ciri pengontrol integral:

1. Keluaran pengontrol integral membutuhkan selang waktu tertentu, sehingga pengontrol integral cenderung memperlambat respon.
2. Ketika sinyal kesalahan berharga nol, keluaran pengontrol bertahan pada nilai sebelumnya.
3. Jika sinyal kesalahan tidak berharga nol, keluaran menunjukkan kenaikan atau penurunan yang dipengaruhi oleh besarnya sinyal kesalahan dan nilai Ki.
4. Konstanta integral Ki yang berharga besar mempercepat hilangnya offset. Tetapi semakin besar nilai konstanta Ki mengakibatkan peningkatan osilasi dari sinyal keluaran pengontrol.

2.7.3. Kontroler Derivatif

Keluaran pengontrol diferensial memiliki sifat seperti halnya suatu operasi derivatif. Perubahan yang mendadak pada masukan pengontrol akan mengakibatkan perubahan yang sangat besar dan cepat. Ketika masukannya tidak mengalami perubahan, keluaran pengontrol juga tidak mengalami perubahan, sedangkan apabila sinyal masukan berubah mendadak dan menaik (berbentuk fungsi *step*), keluaran menghasilkan sinyal berbentuk impuls. Jika sinyal masukan berubah naik secara perlahan (fungsi *ramp*), keluarannya justru merupakan fungsi *step* yang besar magnitudonya sangat dipengaruhi oleh kecepatan naik dari fungsi *ramp* dan konstanta Kd.

Sinyal kontrol u yang dihasilkan oleh kontrol D dapat dinyatakan sebagai G(s)=s.Kd. Dari persamaan di atas, nampak bahwa sifat dari kontroler derivatif ini dalam konteks "kecepatan" atau *rate* dari eror. Dengan sifat ini ia dapat digunakan untuk memperbaiki respon transien dengan memprediksi eror yang akan terjadi. Kontroler derivatif hanya berubah saat ada perubahan eror sehingga saat eror statis kontrol ini tidak akan bereaksi, hal ini pula yang menyebabkan kontroler derivatif tidak dapat dipakai sendiri.

Ciri-ciri pengontrol derivatif:

1. Pengontrol tidak dapat menghasilkan keluaran jika tidak ada perubahan pada masukannya (berupa perubahan sinyal kesalahan)
2. Jika sinyal kesalahan berubah terhadap waktu, maka keluaran yang dihasilkan pengontrol tergantung pada nilai Kd dan laju perubahan sinyal kesalahan.
3. Pengontrol diferensial mempunyai suatu karakter untuk mendahului, sehingga pengontrol ini dapat menghasilkan koreksi yang signifikan sebelum kesalahan pembangkit menjadi sangat besar. Jadi pengontrol diferensial dapat mengantisipasi kesalahan pembangkit, memberikan aksi yang bersifat korektif dan cenderung meningkatkan stabilitas sistem.
4. Dengan meningkatkan nilai Kd, dapat meningkatkan stabilitas sistem dan mengurangi *overshoot*.

Berdasarkan karakteristik pengontrol ini, pengontrol diferensial umumnya dipakai untuk mempercepat respon awal suatu sistem, tetapi tidak memperkecil kesalahan pada keadaan tunaknya. Efek dari *set*iap

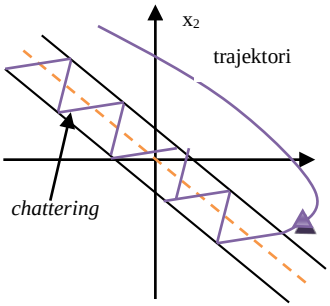
pengontrol Proporsional, Integral dan Derivatif pada sistem lup tertutup disimpulkan pada Tabel 2.3 berikut ini:

Tabel 2.3 Karakteristik Kontroler Proporsional, Integral, dan Derivatif

Respon *Close-Loop*	*Rise Time*	*Overshoot*	*Setting Time*	*Steady State Errror*
Proporsional	Turun	Naik	Perubahan Kecil	Turun
Integral	Turun	Naik	Naik	Hilang
Derivatif	Perubahan Kecil	Turun	Turun	Perubahan Kecil

2.8 *Sliding Mode Control*

Sliding mode control merupakan salah satu teknik kendali untuk sistem *linear* maupun *non-linear*. Pengendalian yang dilakukan dengan memaksa trajektori *state* suatu sistem untuk menuju ke dalam sebuah permukaan luncur tertentu. Setelah sampai ke permukaan luncur tersebut sinyal kendali berusaha mempertahankan agar trajektori status tetap berada disana. *Sliding mode control* termasuk kategori kendali umpan balik berkecepatan tinggi (*high speed switching feedback control*). Hal ini mengakibatkan trajektori status di sekitar permukaan luncur berosilasi dengan frekuensi tinggi (*chattering*). Tentu saja osilasi tersebut menjadi permasalahan stabilitas yang tidak dapat dihiraukan



Gambar 2.10 Diagram Fasa Trajektori Status [17]

begitu saja. Pada Gambar 2.10 dapat dijelaskan bahwa permasalahan dari *sliding mode control* dapat dibagi ke dalam dua kategori besar, yaitu masalah *hitting time* dan *chattering*. *Hitting time* adalah waktu yang dibutuhkan oleh trajektori status dari suatu sistem dari kondisi awalnya menuju permukaan luncur yang telah ditentukan. [17]

2.7.1. Representasi Sistem dalam *State-Space* [17]

Salah satu cara untuk merepresentasikan hubungan *input-output* suatu *plant* adalah dengan menggunakan fungsi penghantar seperti ditunjukan pada Persamaan (2.5):

$$G_{(s)} = \frac{Y_{(s)}}{X_{(s)}} \tag{2.2}$$

$$G(s) = \frac{b_n s_n + b_{n-1} s^{n-1} + ... + b_0}{a_n s^n + a_{n-1} s^{n-1} + ... + a_0} \tag{2.3}$$

Fungsi penghantar tersebut dapat juga ditulis dalam bentuk persamaan diferensial, dengan asumsi semua keadaan awal adalah nol:

$$y^{(n)} + a_{n-1} y^{(n-1)} + ... + a_0 y = b_n u^{(n)} + b_{n-1} u^{(n-1)} + ... + b_0 u \tag{2.4}$$

Dengan memilih:

$$\begin{aligned} x_1 &= y - \beta_0 u \\ x_2 &= \dot{y} - \beta_0 \dot{u} - \beta_1 u = \dot{x}_1 - \beta_1 u \\ x_3 &= \ddot{y} - \beta_0 \ddot{u} - \beta_1 \dot{u} - \beta_2 u = \dot{x}_2 - \beta_2 u \\ &\vdots \\ &\vdots \\ x_n &= y^{(n-1)} - \beta_0 u^{(n-1)} - ... - \beta_{n-1} u = \dot{x}_{n-1} - \beta_{n-1} u \end{aligned} \tag{2.5}$$

Di mana:

$$\begin{aligned} \beta_0 &= b_0 \\ \beta_1 &= b_1 - a_1 \beta \\ \beta_2 &= b_2 - a_1 \beta_1 - a_2 \beta_0 \\ &\vdots \\ &\vdots \end{aligned} \tag{2.6}$$

$$\beta_n = y^{(n-1)} - \beta_0 u^{(n-1)} - ... - \beta_{n-1} u = \dot{x}_{n-1} - \beta_{n-1} u$$

Maka:

$$\begin{aligned} x_1 &= x_2 + \beta_1 u \\ x_2 &= x_3 + \beta_2 u \\ &\vdots \\ &\vdots \\ x_{n-1} &= x_n + \beta_{n-1} u \\ x_n &= -a_n x_1 - a_{n-1} x_2 - ... - a_1 x_n + \beta_n \end{aligned} \tag{2.7}$$

Sehingga didapatkan matriks:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_{n-1} \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \cdots & -a_1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} + \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{n-1} \\ \beta_n \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + \beta_0 u \tag{2.8}$$

Atau dalam bentuk umum persamaan *state* :

$$\dot{x} = Ax + Bu \tag{2.9}$$

$$y = Cx + Du \tag{2.10}$$

Persamaan (2.10) dan Persamaan (2.11) merupakan representasi sistem dalam persamaan *state space.* Variabel yang dipilih sebagai variabel *state* tidak harus keluaran sistem seperti representasi fungsi penghantar, tetapi dapat juga digunakan variabel yang lain misalnya sinyal *error* atau kombinasi *linear* sinyal.

2.7.2. *Sliding Surface* [17]

Semua *state space* dari suatu sistem yang menyebabkan kondisi *Sliding Mode* dapat terjadi disebut *sliding surface* (permukaan luncur). Sebuah sistem yang mencapai *sliding surface* memiliki keuntungan dua kali lipat, pertama terjadi pereduksian orde sistem dan kedua tidak sensitif terhadap parameter yang mutlak pada saluran *input*.

Sliding surface dipilih dengan pertimbangan status trayektori sistem dapat menuju permukaan tersebut di manapun kondisi awalnya dan dalam waktu yang terbatas serta status trayektori sistem dapat dipertahankan di sekitar *sliding surface*. *Sliding surface* secara umum didesain dengan menggunakan kombinasi linier variabel *state*. *Sliding surface* didefinisikan sebagai σs(x):

$$s \cdot x(t) = 0 \tag{2.11}$$

Di mana x adalah vektor *state* sistem dan S adalah *matriks* permukaan luncur dengan nilai elemen ditentukan sendiri dan nilai n adalah orde sistem.

2.7.3. Sinyal Kontrol [17]

Sinyal kontrol merupakan sinyal yang mengendalikan keluaran untuk dapat mengikuti atau mendekati sinyal referensi yang ditentukan. Untuk mendapatkan sinyal kontrol yang mampu membawa status trayektori menuju permukaan luncur dan mempertahankan status trayektori agar tetap berada di sekitar permukaan luncur diperlukan dua macam sinyal kontrol.

Syarat kestabilan Lyapunov digunakan dalam perancangan sinyal kontrol ini. Syarat kestabilan Lyapunov ditunjukkan pada Persamaan (2.12).

$$\sigma(x) \times \dot{\sigma}(x) < 0 \tag{2.12}$$

Sinyal kontrol yang pertama adalah sinyal kontrol ekivalen yang berfungsi untuk membawa status trayektori menuju permukaan luncur (*ueq*) dan sinyal kontrol yang kedua adalah sinyal kontrol natural yang berfungsi untuk mempertahankan status trayektori agar tetap berada di sekitar permukaan luncur (u_N). Sehingga sinyal kontrol total merupakan penjumlahan dari dua sinyal kontrol tersebut.

$$u = u_{eq} + u_N \tag{2.13}$$

Dari Persamaan (2.9) dan Persamaan (2.13), maka persamaan *state* juga dapat dituliskan seperti pada Persamaan (2.14)

$$\begin{aligned} \dot{x} &= Ax + B(u_{eq} + u_n) \\ \dot{x} &= Ax + Bu_{eq} + Bu_n \end{aligned} \tag{2.14}$$

Sinyal kontrol ekivalen merupakan sinyal yang membawa status trayektori menuju permukaan luncur. Pada saat status trayektori tidak berada pada permukaan luncur (σ(x) tidak bernilai 0). Nilai σ(x) > 0 ketika status trayektori berada di atas permukaan luncur sedangkan nilai

$\sigma(x) < 0$ ketika status trayektori berada di bawah permukaan luncur. Sehingga dapat dituliskan menjadi:

$$\begin{aligned} &\dot{\sigma}(x) = 0 \\ &S\dot{x} = 0 \\ &S(A_x + Bu_{eq} + Bu_N) = 0 \end{aligned} \tag{2.15}$$

Pada perhitungan sinyal kontrol ekivalen, sinyal kontrol natural dianggap nol, sehingga:

$$\begin{aligned} &SA_x + SBu_{eq} = 0 \\ &SBu_{eq} = -SAx \\ &u_{eq} = -(SB)^{-1} SAx \end{aligned} \tag{2.16}$$

Perhitungan sinyal kontrol natural:

$$\begin{aligned} &\dot{\sigma}(x) = SAx + SBu_{eq} + SBu_N \\ &\dot{\sigma}(x) = SAx + SB(-SB)^{-1} SAx) + SBu_N \\ &\dot{\sigma}(x) = SAx - SB(SB)^{-1} SAx + SBu_N \\ &\dot{\sigma}(x) = SAx - SAx + SBu_N \\ &\dot{\sigma}(x) = SBu_N \end{aligned} \tag{2.17}$$

Untuk mempertahankan trayektori *state* tetap berada pada permukaan luncur, maka syarat kestabilan *Lyapunov* pada Persamaan (2.12) harus terpenuhi. Sehingga:

1. Pada saat $\sigma(x) < 0$ maka haruslah turunan $\sigma(x) > 0$
2. Pada saat $\sigma(x) > 0$ maka haruslah turunan $\sigma(x) < 0$

Maka dapat dipilih:

$$\dot{\sigma}(x) = -W \cdot sat(\sigma(x) \tag{2.18}$$

Di mana $W > 0$ dan dipilih berdasarkan posisi trayektori pada permukaan luncur agar didapat *chattering* yang tidak terlalu besar, dengan demikian dari Persamaan (2.20) didapat :

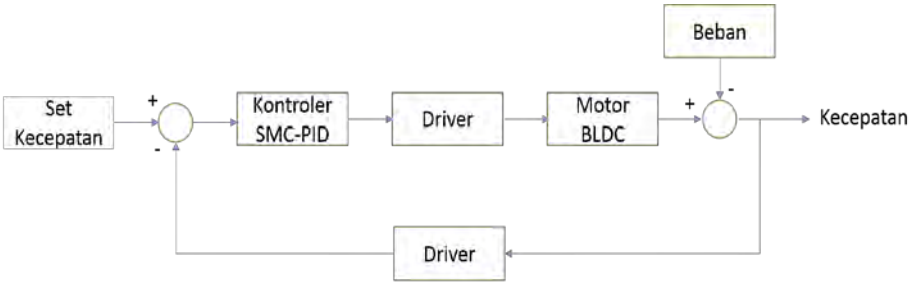
$$\begin{aligned} &SBu_N = -W \cdot sat(\sigma(x)) \\ &u_N = -(SB)^{-1} W \cdot sat(\sigma(x)) \end{aligned} \tag{2.19}$$

BAB 3
PERANCANGAN SISTEM

3.1. Gambaran Umum Sistem

Sistem yang digunakan dalam tugas akhir ini merupakan sistem yang dirancang sendiri. *Plant* yang dibuat merupakan *plant* motor BLDC yang terdiri dari beberapa komponen penyusun yang dirangkai membentuk suatu *plant* motor BLDC untuk melaksanakan suatu objektif tertentu sesuai dengan tujuan dilaksanakannya Tugas Akhir. *Plant* BLDC ini dibuat secara berkelompok yang terdiri dari penulis, M. Safrurriza, Tri Wahyu Kurniawan, Irwan Eko Prabowo, M. Iqbal Fauzi, serta Fairuzza Dinansyar. *Plant* motor BLDC sendiri terdiri dari komponen utama yaitu motor arus searah tanpa sikat (BLDC Motor) yang diambil dari AC *Inverter* merek Daikin, rem elektromagnetik, serta *sensor* arus. Rem elektromagnetik bekerja untuk memberikan efek pembebanan pada kinerja motor. Pembebanan motor merupakan beban yang diterima motor pada saat motor menggerakkan piringan alumunium. *Plant* motor BLDC yang kami buat ini kami beri nama P-1.

Blok diagram sistem yang digunakan pada Tugas Akhir ini ditunjukkan pada Gambar 3.1



Gambar 3.1 Blok Diagram Sistem Pengaturan Kecepatan Motor *Brushless DC* (BLDC).

Selain perangkat keras berupa motor BLDC dan rem elektromagnetik, ditambahkan pula beberapa komponen pendukung, seperti *driver* untuk motor BLDC dan *driver* untuk rem elektromagnetik. Mikrokontroler Arduino juga akan dipakai pada sistem ini sebagai

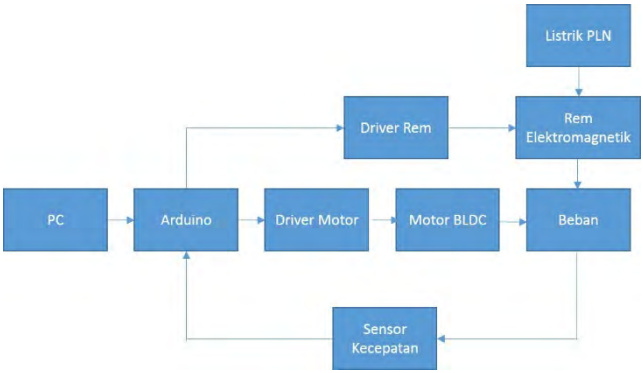
perantara antara *sensor* dan *driver* dengan komputer. Serta rangkaian pengatur PWM (*Pulse Width Modulation)* yang digunakan untuk mengatur besarnya sinyal kontrol yang diberikan pada rem untuk membantu besarnya pembebanan rem magnetik.

Metode yang digunakan dalam pengaturan kecepatan motor BLDC dalam Tugas Akhir ini menggunakan metode kontrol *Sliding Mode Controller* berbasis PID yang merupakan kombinasi antara kontroler PID dengan kontroler *Sliding Mode*. Penggunaan metode ini diharapkan mampu meningkatkan performa motor untuk menghasilkan respon performansi yang diinginkan.

3.2. Perancangan Perangkat Keras

Perancangan yang dilakukan dalam perancangan perangkat keras ada dua jenis, yaitu perancangan mekanik dan elektronik. Perancangan mekanik merupakan perancangan komponen utama pada *plant*, yaitu berupa motor BLDC serta rem elektromagnetik. Sedangkan perancangan elektronik merupakan perancangan untuk kontroler, *driver rem* dan rangkaian *sensor* yang akan digunakan pada *plant*.

Arduino menerima data dari *sensor* dan mengirimkannya kepada komputer. Selain itu, digunakan digunakan pula rangkaian *driver* untuk menggerakan motor BLDC dan memberi *input* arus pada rem elektromagnetik. Rangkaian *sensor* kecepatan berfungsi mengukur *input* arus yang masuk ke dalam rem elektromagnetik dan *sensor* kecepatan motor BLDC. Konfigurasi perangkat keras pada *plant* dapat dilihat pada Gambar 3.2.



Gambar 3.2 Konfigurasi Perangkat Keras Sistem Pengaturan Kecepatan Motor *Brushless DC* (BLDC).

3.2.1. Perancangan Mekanik

Pada *plant* sistem pengaturan kecepatan motor BLDC, terdapat beberapa komponen utama yang digunakan, diantaranya motor BLDC sebagai tenaga penggerak dan objek yang dikontrol. Selain itu, terdapat rem elektromagnetik yang digunakan untuk memberikan efek pembebanan pada motor BLDC. Rem elektromagnetik diberikan *input* arus DC yang berbentuk PWM. Penjelasan mengenai konstruksi motor BLDC dan rem elektromagnetik dapat dilihat pada sub-bab di bawah ini.

3.2.1.1. Motor Brushless DC

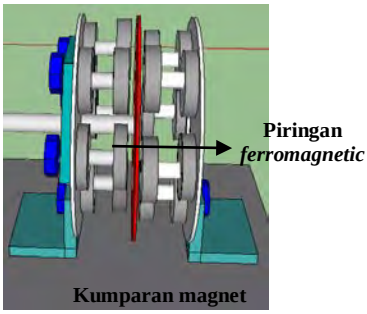
Motor *Brushless* DC (BLDC) dipilih karena efisiensi dan konstruksinya yang tidak menggunaakan *brush* atau sikat. Motor BLDC yang digunakan merupakan motor BLDC yang digunakan pada AC Daikin *Inverter*. Bentuk fisik motor BLDC yang digunakan pada *plant* ini dapat dilihat pada Gambar 2.1.

3.2.1.2. Rem Elektromagnetik

Rem elektromagnetik pada *plant* ini berguna sebagai pembebanan pada motor. Medan elektromagnetik dari rem ini dihasilkan oleh beberapa kumparan yang dihubungkan secara seri dan paralel dan diberikan masukan arus DC. Arus DC yang masuk ke dalam rem elektromagnetik berbentuk PWM yang *duty cycle*-nya diatur oleh *driver* dan Arduino. Adapun sumber tegangannya didapat dari jala-jala PLN. lalu *output* pada trafo disearahkan melalui rangkaian *rectifier*.

Rem elektromagnetik ini terbuat dari 8 kumparan, terdiri dari 4 kumparan di tiap sisinya, yang disusun secara seri. Diantara celah kedua sisi kumparan dipasang piringan aluminium. Piringan aluminium tersebut lalu dipasang ke dalam sebuah *shaft* yang selanjutnya dikopel dengan motor BLDC. Perbandingan *gear* yang digunakan untuk mengkopel motor BLDC dengan *shaft* tersebut.

Kumparan ini terbuat dari 8 buah baut dan 16 buah mur. Lalu pada baut dilapisi kertas sebagai isolator. Selanjutnya kawat tembaga dililit pada kertas tersebut hingga jumlah lilitan yang telah ditentukan sebanyak 400 lilitan. Konfigurasi dari Rem Elektromagnetik dapat dilihat pada Gambar 3.3



Gambar 3.3 Konfigurasi Perancangan Rem Magnetik

3.2.2. Perancangan Elektronik

Perancangan elektronik ini meliputi desain *layout* rangkaian PCB serta pengkabelan. Rangkaian elektronik pada *plant* ini meliputi rangkaian *driver* rem elektromagnetik serta rangkaian *sensor* kecepatan dan *sensor* arus. Serta rancangan pengkabelan pada mikrokontroler Arduino.

3.2.2.1. Rangkaian Driver Rem Elektromagnetik

Rem elektromagnetik yang digunakan pada *plant* ini dapat dikendalikan kekuatan medan magnetnya. Hal ini dapat dilakukan dengan mengubah besarnya arus yang masuk ke kumparan. Cara termudah untuk merubah arus pada kumparan adalah dengan mengatur tegangan masukan pada kumparan dikarenakan dengan berubahnya nilai tegangan masukan pada kumparan maka tentunya nilai arus masukan pada kumparan berubah menyesuaikan dengan tegangan. Hal ini berjalan sesuai dengan hukum Ohm yaitu V = I.R, karena R pada kumparan bernilai sama maka nilai tegangan sebanding dengan nilai arus. Hal inilah yang kami manfaatkan untuk dapat merubah besarnya medan magnet pada kumparan.

Ada beberapa cara untuk dapat mengatur tegangan. Misalnya saja rangkaian pembagi tegangan dan dengan membuat tegangan menjadi sebuah sinyal *Pulse Width Modulation* (PWM). Pada tugas akhir kali ini untuk mengatur tegangan masukan pada kumparan digunakan metode yang menggunakan sinyal PWM.

3.2.2.2. Pengkabelan Mikrokontroler Arduino

Arduino memiliki 14 *input output* digital dan 6 *input output analog*. Pada Arduino Uno juga terdapat *pin* yang menggunakan sinyal PWM sebagai *input* dan *output* yaitu *pin* 3, 5, 6, 9, 10, 11. Pada tugas akhir ini digunakan beberapa buah *pin input output* dengan rincian sebagai berikut:

a. *Pin* A0 : *input analog* dari potensio
b. *Pin* 7 : *input encoder*
c. *Pin* 9 : *output* PWM untuk motor BLDC
d. *Pin* 11 : *output* PWM untuk rem magnetik

Selain *pin-pin* di atas tentunya dibutuhkan juga *pin* 5 Volt dan *ground* karena setiap *sensor* dan motor membutuhkan *ground*.

3.3. Perancangan Perangkat Lunak

Dalam sistem pengaturan motor BLDC, beberapa perangkat lunak harus digunakan untuk proses pengambilan data, perancangan kontroler, dan pengiriman data. Perangkat lunak yang digunakan yaitu MATLAB dan *software* Arduino.

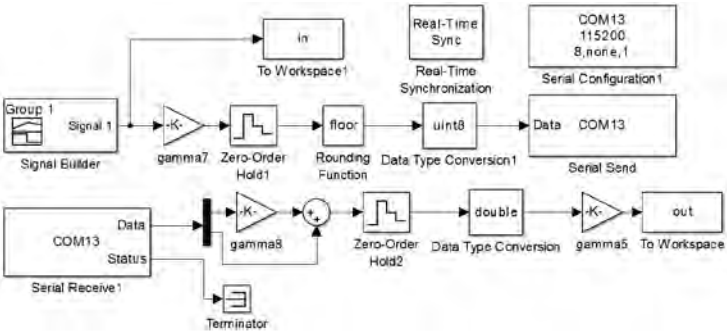
3.3.1. Perangkat Lunak Matlab

Software yang digunakan pada Tugas Akhir kali ini adalah *software* MATLAB. Versi MATLAB yang digunakan pada pelaksanaan Tugas Akhir kali ini mempunyai versi 2013a. *Software Simulink* digunakan sebagai *Human Machine Interface* pada proses pengiriman dan penerimaan data melalui serial USB dari mikrokontroler Arduino

Dengan menggunakan komunikasi serial pada Arduino, *Simulink* MATLAB dapat mengolah data dari blok *serial receive* dan mengirimkannya kembali melalui blok *serial send. Simulink* juga digunakan untuk proses identifikasi sistem *open loop* maupun *closed loop* menggunakan blok *System Identification Toolbox*. Selain digunakan untuk keperluan identikasi *open loop* untuk mendapatakan model matematika *plant* motor BLDC.

MATLAB juga digunakan untuk keperluan proses perancangan kontroler *Sliding Mode* berbasis PID yang digunakan. Perancangan kontroler ini dilakukan dengan membuat *Simulink* dari kontroler *Sliding Mode-PID*. Setelah itu MATLAB digunakan untuk mensimulasikan hasil perancangan kontroler terhadap hasil pemodelan *plant* yang diperoleh. Terakhir, MATLAB digunakan sebagai antarmuka untuk

melakukan implementasi kontroler yang telah dirancang sebelumnya. Diagram blok *simulink* untuk identifikasi dapat dilihat pada Gambar 3.4.



Gambar 3.4 Diagram Blok *Simulink* Identifikasi

Blok *signal builder* dari sinyal PRBS memiliki batas bawah senilai 0,7 dan batas atas 0,9 sehingga keluaran sinyal kontrol maksimum yang dapat dikeluarkan oleh kontroler bernilai 1. Ouput dari *signal builder* dikalikan dengan 255 untuk penskalaan pada *output* PWM Arduino. Data tersebut lalu masuk ke blok *zero-order hold* yang berfungsi sebagai *buffer*. Data tersebut kemudian dibulatkan menggunakan blok *rounding function* dan tipe data diubah menjadi data *byte* menggunakan blok *data type conversion.* Setelah itu, data dikirimkan ke arduino menggunakan blok *serial send*.

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

3.3.2. Software Arduino

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

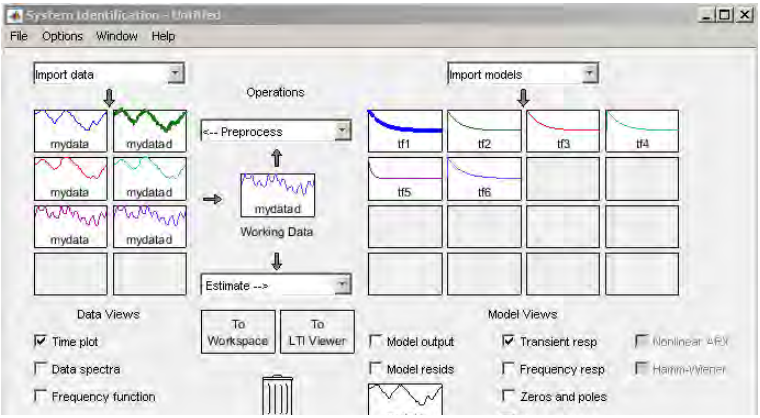
Pada Tugas Akhir ini, Arduino Uno digunakan untuk membaca kecepatan melalui rangkaian pembaca kecepatan yang digunakan, mengatur penggunaan besarnya efek pembebanan yang dihasilkan rem elektromagnetik melalui ragkaian *driver* rem elektromagnetik, serta mengatur sinyal kontrol untuk menggerakan motor BLDC yang direpresentasikan sebagai *throttle* yang sekaligus sebagai representasi dari besarnya sinyal PWM yang diberikan untuk menggerakkan motor. Tampilan *software* ini dapat dilihat di Gambar 3.5

Gambar 3.5 Tampilan Arduino IDE

3.4. Identifikasi dan Pemodelan Sistem

Pada pemodelan sistem di tugas akhir ini menggunakan *System Identification Toolbox* pada MATLAB. Model matematika *plant* didapatkan melalui *toolbox* ini dengan cara meng*input*kan data *input-output*, melakukan *preprocessing*, dan memilih bentuk model yang diinginkan.. Dalam proses identifikasi ini, data *input-output* yang ada dimasukkan ke dalam aplikasi dengan memilih "*time domain data"* saat menekan akan timbul tulisan *"import data"*. Setelah itu dilakukan *preprocessing*.

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*" *set*elah meng-klik *popout* bertuliskan "*estimate*". Tampilan dari *system identification toolbox* pada Matlab dapat dilihat pada Gambar 3.6.
Hasil identifikasi dapat dilihat pada Tabel 3.1.



Gambar 3.6 Tampilan *System Identification Toolbox*

Tabel 3.1 Hasil Identifikasi *Plant*

Beban	Tegangan	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 Sliding Mode berbasis PID

Setelah fungsi alih dari sistem telah didapatkan, langkah selanjutnya adalah merancang kontroler untuk pengaturan kecepatan motor. Kontroler digunakan untuk membuat *plant* mengikuti model referensi yang diinginkan sekalipun motor BLDC diberi beban. Tahapan desain kontroler ini meliputi perancangan kontroler *Sliding Mode* serta perancangan kontroler PID.

3.5.1. Perancangan Kontroler PID

Sebelum merancang kontroler *Sliding Mode*, diperlukan model referensi yang diinginkan. Model yang diinginkan memiliki *rise time* sebesar 6 detik, dengan *time* constant (Tc) 2 detik dan *pole* λ_2 = 1/Tc = 0,5. Nilai *pole* dominan ditentukan λ_1 sebesar 5x *pole* λ_2 yaitu sebesar 2,5. Sehingga fungsi *transfer* model menjadi:

$$\frac{1.25}{s^2 + 3s + 1.25}$$

Nilai dari parameter Kp, Ki, dan Kd didapatkan dari persamaan di bawah ini:

$$\begin{aligned} K_p &= \lambda_1 + \lambda_2 = 2{,}5 + 0{,}5 = 3 \\ K_i &= \lambda_1 \lambda_2 = (2{,}5)(0{,}5) = 1{,}25 \\ K_p &= 1 \end{aligned} \tag{3.1}$$

3.5.2. Perancangan Kontroler Sliding Mode

Perancangan kontroler didesain dan diimplementasikan untuk mempercepat *risetime* dan mempertahankan kecepatan motor BLDC pada saat diberi beban tertentu. Pada hasil identifikasi dengan metode ARX, didapat fungsi alih dari *plant*:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{K}{s^2 + As + B} \tag{3.2}$$

Di mana :

K = 122,2386

A = 28,9879

B = 134,5795

Fungsi alih tersebut dapat direpresentasikan dalam bentuk persamaan diferensial (dengan asumsi nilai awal adalah nol):

$$\ddot{y} + A\dot{y} + By = Ku \tag{3.3}$$

Sehingga nilai y sebesar:

$$\ddot{y} = -A\dot{y} - By + Ku \tag{3.4}$$

Didefinisikan suatu permukaan luncur s sebagai berikut:

$$s = \dot{e} + (\lambda_1 + \lambda_2)e + (\lambda_1\lambda_2)\int e \quad (3.5)$$

Dengan nilai turunannya sebagai berikut:

$$\dot{s} = \ddot{e} + (\lambda_1 + \lambda_2)\dot{e} + (\lambda_1\lambda_2)e = 0 \quad (3.6)$$

Persamaan untuk sinyal eror adalah:

$$e = y_R - y \quad (3.7)$$

Subtitusi Persamaan (3.7) ke Persamaan (3.6) didapatkan:

$$(\ddot{y_R} + \ddot{y}) + (\lambda_1 + \lambda_2)\dot{e} + (\lambda_1\lambda_2)e = 0 \quad (3.8)$$

Karena input berupa *step*, maka nilai $\ddot{y_R} = \dot{y_R} = 0$, sehingga Persamaan (3.8) menjadi:

$$\ddot{y} + (\lambda_1 + \lambda_2)\dot{e} + (\lambda_1\lambda_2)e = 0 \quad (3.9)$$

Subtitusi Persamaan (3.4) ke Persamaan (3.9) didapatkan:

$$\left(A\dot{y} + By - Ku_{eq}\right) + (\lambda_1 + \lambda_2)\dot{e} + (\lambda_1\lambda_2)e = 0 \quad (3.10)$$

Dari Persamaan (3.10) didapatkan persamaan u_{eq}:

$$u_{eq} = \frac{1}{K}\{A\dot{y} + By + (\lambda_1 + \lambda_2)\dot{e} + (\lambda_1\lambda_2)e\} \quad (3.11)$$

Setelah ditemukan sinyal kontrol ekivalen, akan dicari sinyal kontrol natural:

$$u_N = ksat(s) = ksat(\dot{e} + (\lambda_1 + \lambda_2)e + (\lambda_1\lambda_2)\int e)$$
$$u_N = ksat\left\{(\lambda_1 + \lambda_2)\left(\frac{1}{(\lambda_1+\lambda_2)}\dot{e} + e + \frac{(\lambda_1\lambda_2)}{(\lambda_1+\lambda_2)}\int e\right)\right\} \quad (3.12)$$

Persamaan (3.11) dan (3.12) merupakan persamaan diferensial untuk sinyal kontrol ekivalen dan natural yang nantinya akan didesain pada diagram blok program kontroler dengan menggunakan MATLAB. Sinyal kontrol total merupakan hasil penjumlahan dari sinyal kontrol ekivalen dan sinyal kontrol natural.

BAB 4
PENGUJIAN DAN ANALISA

Pada tahapan ini dilakukan beberapa jenis pengujian yaitu pengujian *sensor*, pengujian *open loop* dari motor BLDC, lalu pengujian kontroler yang disimulasikan pada hasil identifikasi model, lalu yang terakhir merupakan pengujian implementasi kontroler pada *plant* motor BLDC.

4.1. Pengujian *Sensor* Kecepatan Motor BLDC

Pengujian *sensor* kecepatan motor dilakukan dengan cara menghitung frekuensi dari sinyal kotak *driver* motor menggunakan fungsi *pulsein* Arduino. Kecepatan motor dihitung dengan hubungan $\omega = 15 \times f$. *Output* dari *sensor* dibandingkan dengan pengukuran menggunakan *laser tachometer*. Hasil pembacaan *sensor* kecepatan memiliki kesalahan *output* maksimal sebesar 1,5%. Tabel 4.1 merupakan hasil pembacaan kecepatan motor.

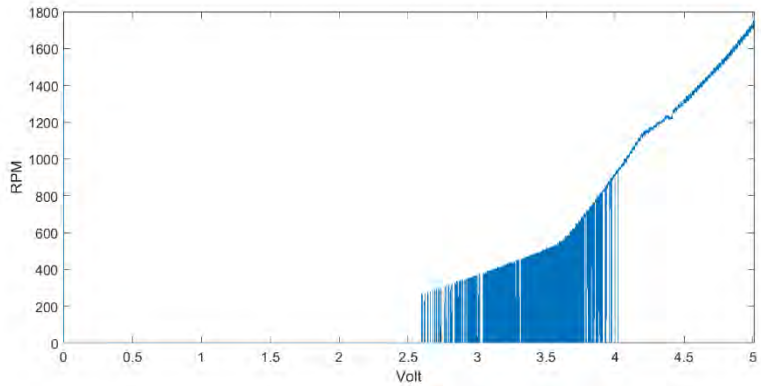
Tabel 4.1 *Output* Pembacaan Kecepatan Motor.

Hasil Pembacaan		Selisih	% error
Arduino	*Tachogenerator*		
972,36	984,6	12,24	1,258793
1022,1	1035,41	13,31	1,302221
1201,8	1218,04	16,24	1,351306
1363,4	1382,67	19,27	1,413378
1429,5	1448,82	19,32	1,351522
1703,6	1726,00	22,40	1,314863
2253	2290,45	37,45	1,670000
2397	2441,21	44,21	1,840000

Dari hasil perbandingan pembacaan Arduino dan *tachogenerator* pada Tabel 4.1, selisih antara keduanya tidak terlalu besar. Eror pembacaan hanya berkisar 1,2 – 1,8%. Sehingga dapat ditarik kesimpulan bahwa metode pengukuran kecepatan motor menggunakan frekuensi pulsa dari *output driver* motor sudah benar. Semakin besar kecepatan motor maka persentase eror dari pembacaan juga semakin besar.

4.2. Pengujian Open Loop Kecepatan Motor

Selanjutnya dilakukan pengujian pada respon kecepatan motor BLDC yang dipergunakan. Pengujian ini dilakukan untuk mengetahui grafik hubungan antara *input-output plant*. Pengujian dilakukan pada saat beban nominal dengan memberikan sinyal tangga pada *input plant* dan diukur kecepatannya. Hubungan antara *input-output* kecepatan motor dapat dilihat pada Gambar 4.1.



Gambar 4.1 *Input Output* Kecepatan Motor.

Dari grafik di atas dapat dilihat bahwa hubungan *input* dan *output* pada motor BLDC linier ketika kecepatan di atas 1000 rpm. Pada kecepatan di bawah 1000 rpm, kecepatan motor tidak dapat terukur dengan baik. Hal ini menunjukkan adanya kekurangan dari *sensor* kecepatan motor sehingga rentang kerja dari *plant* harus lebih dari 1000 rpm. Sehingga pada tugas akhir ini, dipilih rentang kerja motor antara 1000-1535 rpm.

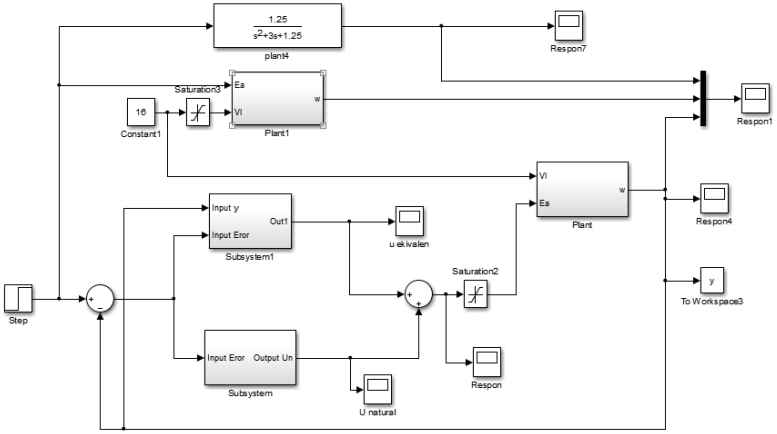
4.3. Pengujian Simulasi Kontroler

Sistem yang telah dirancang pada Bab III disimulasikan terlebih dahulu sebelum diimplementasikan, simulasi menggunakan *software* Matlab R2013a untuk mengetahui respon sinyal kontrol dengan menggunakan beban maupun tanpa beban.

4.3.1 Blok Diagram *Simulink*

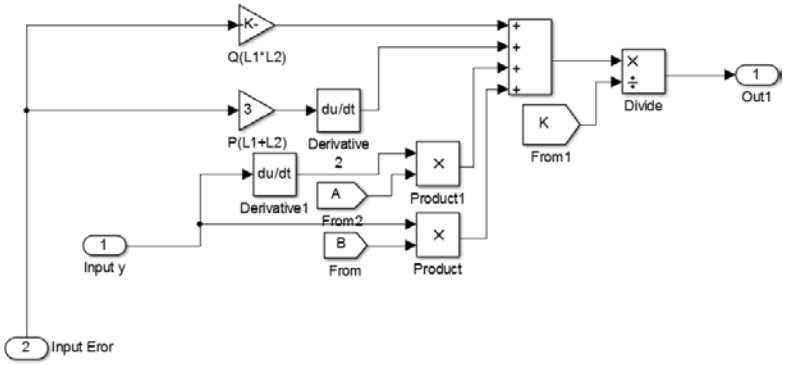
Pada blok diagram simulasi kontroler, keluaran dari kontroler merupakan penjumlahan antara sinyal kontrol ekivalen dan sinyal kontrol natural. *Input* pada sinyal kontrol ekivalen berupa eror dan *output* dari kontroler, sedangkan *input* pada sinyal kontrol natural hanya eror saja.

Keluaran dari *plant* berupa *output* y yang menjadi masukan dari sinyal kontrol ekivalen. Model referensi yang diharapkan mendapatkan masukan berupa sinyal *step*. Keluaran dari *plant* diharapkan dapat mengikuti keluaran dari model referensi yang sudah di desain.
Simulasi kontroler ditampilkan pada Gambar 4.2



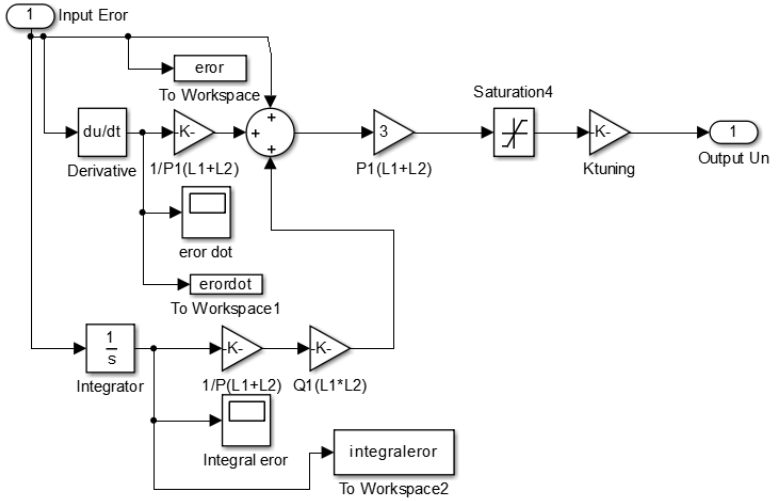
Gambar 4.2 Simulasi Kontroler SMC

Gambar 4.3 merupakan *subsystem* sinyal kontrol ekivalen di mana nilai dari parameter didapatkan dari parameter *plant* yang ditulis pada bab 3. Adapun parameter yang digunakan dalam merancang sinyal kontrol ekivalen yaitu nilai K, A, B, Q ($\lambda_1\lambda_2$), dan P ($\lambda_1 + \lambda_2$).



Gambar 4.3 *Subsystem* Sinyal Kontrol Ekivalen

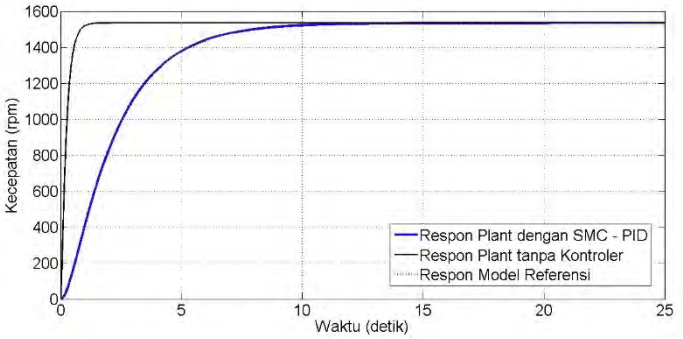
Gambar 4.4 merupakan *subsystem* sinyal kontrol natural di mana nilai dari parameter didapatkan dari model referensi yang diharapkan yaitu Q $(\lambda_1\lambda_2)$, dan P $(\lambda_1 + \lambda_2)$. Serta nilai Ktuning sebesar 0,000000095.



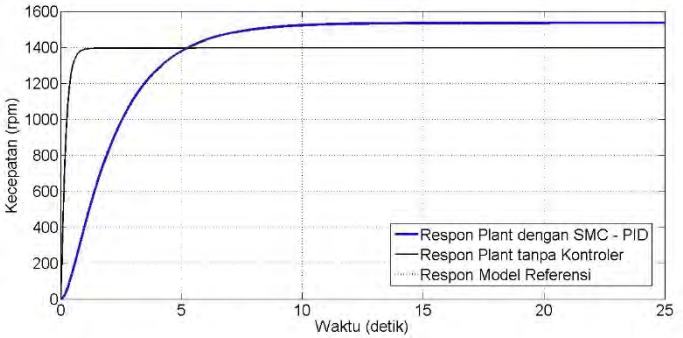
Gambar 4.4 *Subsystem* Sinyal Kontrol Natural

4.3.2 Hasil dan Analisa Simulasi

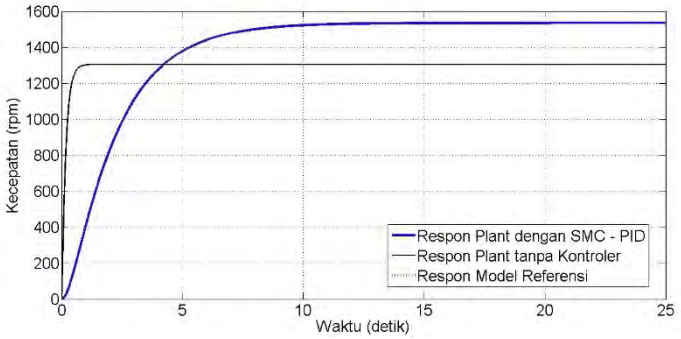
Pengujian dilakukan dengan memberikan sinyal *input* y_R berupa sinyal *step*. Sebelum dilakukan pengujian, dimasukkan nilai parameter yang didapatkan dari model matematis *plant* serta model referensi. Simulasi yang dilakukan dengan membandingkan respon model referensi dengan respon *plant* ketika diberikan beban minimal, beban nominal, serta beban maksimal.



Gambar 4.5 Respon *Plant* dengan Beban Minimal



Gambar 4.6 Respon *Plant* dengan Beban Nominal



Gambar 4.7 Respon *Plant* dengan Beban Maksimal

Gambar 4.5, 4.6, dan 4.7 menunjukkan perbandingan antara respon *plant* dengan kontroler, respon *plant* tanpa kontroler, serta respon model referensi ketika diberikan beban minimal, nominal, serta maksimal. Terlihat bahwa respon pemodelan *plant* dengan menggunakan *sliding mode control-PID* berimpit dengan respon dari model yang diinginkan.

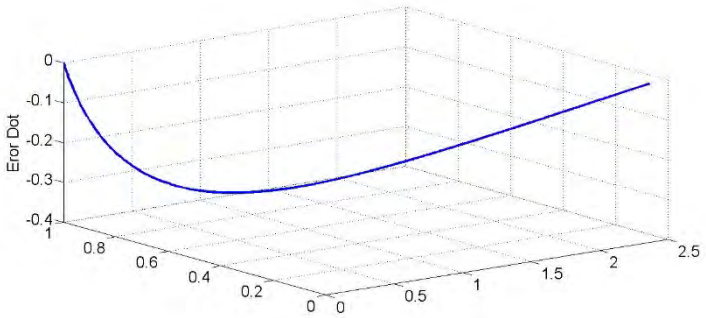
Dari ketiga gambar tersebut terlihat bahwa pemodelan *plant* membutuhkan waktu *steady* state yang lebih lama dari respon *plant* yang sebenarnya. Hal ini dikarenakan pemodelan *plant* mengikuti model yang diinginkan. Waktu untuk mencapai *steady state* pada respon asli *plant* sebesar 1,2 detik sedangkan saat diberi kontroler SMC untuk mengikuti *steady* state pada model yang diinginkan sebesar 6,4 detik. Kecepatan dari motor dalam bentuk persentase sehingga perlu dikalikan dengan kecepatan maksimal sebesar 1535 rpm. Indeks performansi dari respon *step* dapat dilihat pada Tabel 4.2.

Tabel 4.2 Indeks Performansi *Plant* saat Simulasi

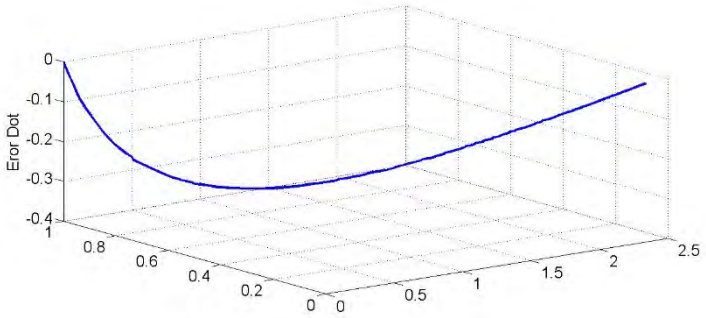
Indeks Performansi	SMC - PID		
	Beban Minimal	Beban Nominal	Beban Maksimal
T_R (detik)	5,053	5,05	5,049
T_s (detik)	6,431	6,443	6,439
RMSE	0	0	0

4.3.3 Analisis *Sliding Surface* pada *Sliding Mode Control*

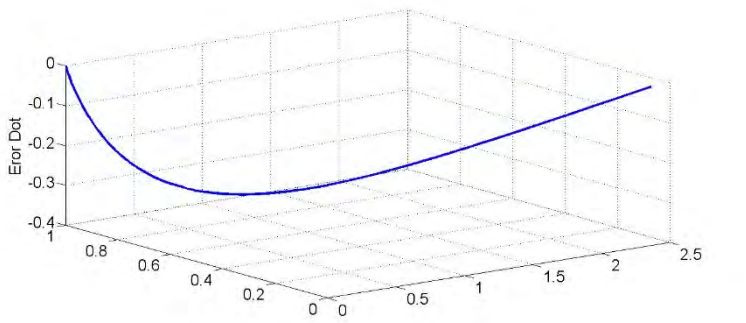
Pada prinsipnya, SMC menggunakan hukum kendali pensaklaran berkecepatan tinggi *(high-speed switching)* untuk membawa trayektori status dari sistem linier dan *non*-linier ke dalam sebuah permukaan luncur *(sliding surface)* yang selanjutnya akan dipelihara agar tetap bertahan meluncur pada *sliding surface*. Proses pemeliharaan trajektori akan mengakibatkan terjadinya osilasi pada *sliding surface* yang sering disebut dengan *chattering*. *Chattering* ini berdampak pada stabilitas sistem kontrol yang didapat berdasarkan perancangan kontroler. Maka dari itu, analisis ini diperlukan dalam berbagai kondisi, di mana saat mendapat beban minimal, beban nominal dan beban maksimal. Gambar 4.8, 4.9, dan 4.10 merupakan tampilan *sliding surface* dari nilai eror *dot.*



Gambar 4.8 *Sliding Surface* pada Beban Nominal



Gambar 4.9 *Sliding Surface* pada Beban Minimal



Gambar 4.10 *Sliding Surface* pada Beban Maksimal

4.4. Implementasi Sistem

Pada tahapan ini kontroler yang telah dibuat dicoba dijalankan pada *plant real* motor BLDC P-1. Pengujian ini dilakukan dengan memberikan respon *step* pada saat *plant* diberi beban minimal, nominal, dan maksimal.

4.4.1 Realisasi *Plant*

Berikut ini merupakan *plant* dari sistem pengaturan motor BLDC yang telah di realisasikan. *Plant* terbagi menjadi 2 sistem, yaitu sistem yang bekerja menjalankan motor, lalu sistem yang menjalankan rem. Dari kedua sistem ini untuk bisa terhubung dengan kontroler harus memiliki *interface. Interface* yang kami gunakan adalah Arduino Uno.

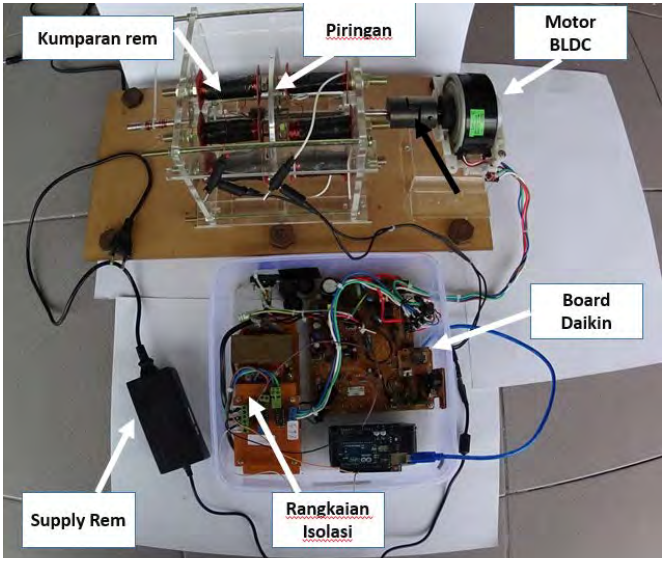
Board bawaan Daikin yang ada di dalam *air conditioner* merupakan *board* yang digunakan untuk mengatur kecepatan motor BLDC. Selain itu juga memiliki fungsi sebagai *power supply driver* motor yang berada di dalam motor. Pada perancangan *plant* motor BLDC P-1, *board* ini hanya digunakan sebagai *power supply driver* motor, sedangkan pengaturan kecepatannya melalui kontroler yang sudah dibuat di Matlab.

Arduino Uno digunakan sebagai *interface* yang menghubungkan antara *plant* dengan laptop. Arduino menghitung kecepatan motor degan cara mengukur periode pulsa yang dikirim oleh *driver* motor lalu mengirim nilai kecepatan ke laptop. Arduino juga berfungsi untuk menerima masukan data dari laptop untuk mengubah nilai PWM dari

pin output yang akan digunakan untuk sinyal kontrol yang mengatur kecepatan motor. Daya untuk rem dapat diubah-ubah dengan mengubah nilai *power supply* yang digunakan untuk men*supply* rem elektromagnetik.

Gambar 4.11 merupakan tampilan fisik dari *plant* motor BLDC yang diambil dari *plant* yang telah direalisasikan. Pada *plant* ini sebagian besar bahan dasar yang dipakai untuk dudukan terbuat dari akrilik. Pada *plant* ini terdapat dua buah PCB yang diantaranya rangkaian *board* Daikin dan rangkaian isolasi. *Set*elah itu terdapat 8 buah lilitan kawat yang berfungsi sebagai rem magnetik. Selain itu terdapat 1 buah piringan yang satu terbuat dari alumunium yang berfungsi sebagai beban pada poros.

Kontruksi rem magnetik adalah baut yang memiliki diameter 1 cm. lalu baut tersebut dililit sebanyak 400 lilitan agar didpatkan medan magnet yang cukup kuat untuk membebani motor BLDC. Pada Gambar 4.11 tersebut juga terlihat ada piringan diantara dua baut yang terbuat dari alumunium setebal 5 mm. diameter dari pringan ini adalah 15 cm. piringan terpasang pada sebuah poros yang terpasang dengan kopel yang terhubung dengan motor. Poros ini juga ikut menyebabkan piringan berputar.

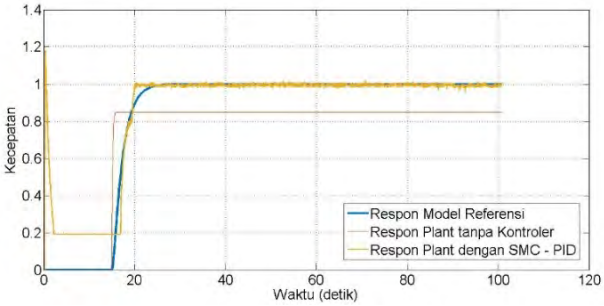


Gambar 4.11 *Plant* Motor BLDC P-1

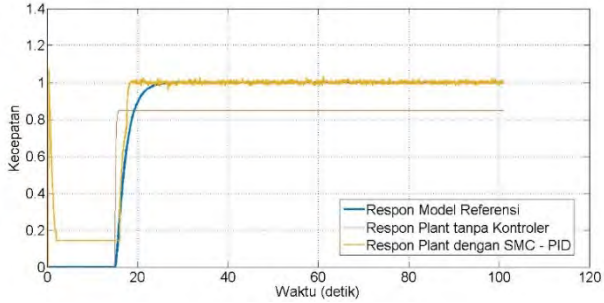
4.4.2 Implementasi Kontroler *Sliding Mode – PID* pada *Plant* P-1

Dalam tahap implementasi ini pertama-tama menentukan parameter yang digunakan pada kontroler *Sliding Mode. Set*elah menentukan parameter, selanjutnya menjalankan sistem dengan menggunakan kontroler *Sliding Mode* dan telah mendapatkan nilai *gain* proporsional, integral dan derivatif. Implementasi yang dilakukan dengan memberikan beban minimal, nominal, dan maksimal pada saat motor dijalankan. Terdapat beberapa perubahan yang dilakukan pada parameter K, A, dan B di mana nilai K, A, dan B diubah menjadi 3, 2,5, dan 1 dikarenakan adanya perbedaan pemodelan *plant* dengan *plant* real.

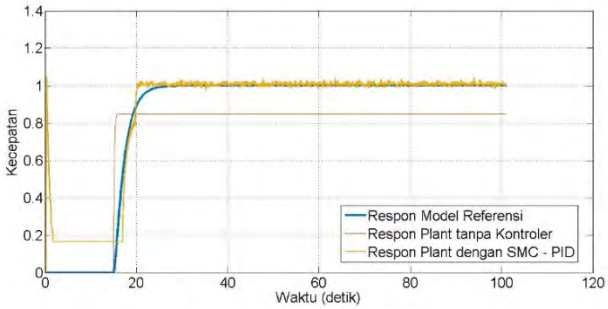
Gambar 4.12, 4.13, 4.14, merupakan respon *output plant* dengan *Sliding Mode – PID* pada kondisi beban minimal, nominal, dan maksimal. Dari gambar dapat ditarik kesimpulan bahwa pada saat diberi beban minimal, nominal, maupun maksimal, motor dapat mengikuti respon model referensi yang diinginkan.



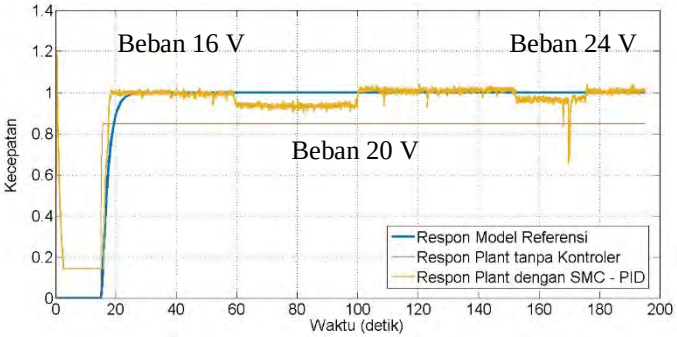
Gambar 4.12 *Step* Respon Sistem saat Beban Nominal



Gambar 4.13 *Step* Respon Sistem saat Beban Minimal



Gambar 4.14 *Step* Respon Sistem saat Beban Maksimal



Gambar 4.15 *Step* Respon Sistem saat Beban Campuran

Gambar 4.15 merupakan hasil respon sistem ketika motor diberi beban minimal, lalu *set*elah *steady* beban diubah menjadi beban nominal, dan *set*elah *steady* diubah menjadi beban maksimal.

Dari Gambar 4.15 dapat dilihat bahwa saat diberi 16 V kondisi sudah *steady* dengan waktu 2,8 detik. Lalu beban diubah menjadi 20 V di mana terjadi penurunan kecepatan, lalu kecepatan akan menjadi *steady*. *Set*elah kondisi *steady*, maka beban diubah lagi menjadi 24 V sehingga terjadi penurunan kecepatan, hingga kecepatan akan menjadi *steady* lagi. Tabel 4.3 merupakan indeks performansi *plant* pada saat implementasi dilakukan.

Tabel 4.3 Indeks Performansi *Plant* saat Implementasi

Indeks Performansi	SMC - PID		
	Beban Minimal	Beban Nominal	Beban Maksimal
T_R (detik)	4,7	2,75	4,9
T_s (detik)	4,8	2,87	5,01
RMSE	0	0	0

BAB 5
PENUTUP

Pada bab ini akan dibahas mengenai hasil akhir dari pengerjaan Tugas Akhir meliputi kesimpulan dan saran sebagai referensi tambahan agar penelitian yang akan dilakukan setelahnya dapat lebih baik lagi.

5.1. Kesimpulan

Dari analisa yang telah dilakukan terhadap hasil simulasi maka dapat disimpulkan bahwa pengaturan kecepatan menggunakan *sliding mode control-PID* menghasilkan respon sistem yang dapat mengikuti respon model referensi yang diharapkan. Pada implementasi kontroler, respon plant dapat mengikuti model referensi namun masih ada eror yang terjadi saat peningkatan kecepatan motor.

5.2. Saran

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

LAMPIRAN

Program Arduino

```
int pin = 7;
int out = 6;
byte u = 255;
unsigned long duration;
unsigned long spd=0;
word kec;
byte w1;
byte w2;

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

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

void baca(){
  duration = pulseIn(pin, LOW, 30000);
     if (duration == 0 && i<3){
        spd = 0;}
      else{
        spd=(15*1000000/(2*duration));}
  kec=int(spd);
  w1=kec/256;
  w2=kec%256;
  Serial.write(w1);
  Serial.write(w2);}
```

DAFTAR PUSTAKA

[1] E. Cerruto, A. Consoli dkk, "A robust adaptive controller for PM motor drives in robotic application", *IEEE Trans. Ind. Applicat*, 1992.

[2] Permana, Putra Eka, "*PID (Proportional-Integral-Derivative) Controller*", <URL: https://putraekapermana.wordpress.com/2013/11/21/pid/>, November, 2013.

[3] K.D. Young, V.I. Utkin, dan Ü. Özgüner, "A control engineer's guide to sliding mode control". *IEEE Trans. Control Sys. Tech.*, 1999.

[4] H. S. Choi, Y. H. Park, Y. S. Cho, and M. Lee, "Global sliding-mode control improved design for a brushless DC motor", *IEEE Control Systems Magazine*, 1991.

[5] M.Fallahi, S.Azadi, "Robust Control of DC Motor Using Fuzzy Sliding Mode Control with PID Compensator", *International Multi Conference of Engineers and Computer Scientists*, 2009.

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

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

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

[9] Covey, Greg, "*Brushless Basics*", <URL: http://www.rcuniverse.com/magazine/article_display.cfm?article_i

d=1344/>, Mei, 2011.

[10] Madaan, Pushek, "*Brushless DC Motors – Part 1: Construction and Operating Principles*", <URL: http://www.edn.com/design/sensors/4406682/1/Brushless-DC-Motors---Part-I--Construction-and-Operating-Principles/>, Februari, 2013.

[11] Sholihah, Suci Endah, "Desain dan Implementasi Kontroler Sliding Mode Untuk Pengaturan Akselerasi Pada Simulator Hybrid Electric Vehicle", *Tugas Akhir*, Jurusan Teknik Elektro ITS Surabaya, 2012.

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

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

[14] Elrosa, Ilmiyah, "Traction Control pada Parallel Hybrid Electric Vehicle dengan Metode Generalized Predictive Control", *Tugas Akhir*, Jurusan Teknik Elektro ITS Surabaya, 2014.

[15] Ljung, Lenart dan Torkel, Glad, "*Modelling of Dynamic Systems*", Prentice – Hall International: New Jersey. 1994.

[16] Ogata, Katsuhiko, "*Modern Control Engineering*", Prentice-Hall Inc,1970

[17] Harapan, Okky Sugianto, "Perancangan dan Implementasi Kontroler Sliding Mode untuk Pengaturan Level pada Couple Tanks", *Tugas Akhir*, Jurusan Teknik Elektro ITS Surabaya, 2010.

RIWAYAT HIDUP

Guntur Sadhiea Putra lahir di Sidoarjo tanggal 27 Maret 1993, merupakan anak pertama dari Edy Yuwono dan Sudarwati. *Set*elah lulus dari SMA Negeri 1 Sidoarjo tahun 2011, melanjutkan studi di Jurusan Teknik Elektro Institut Teknologi Sepuluh Nopember (ITS) Surabaya pada tahun yang sama.