

**PENALAAAN PID PADA KAPAL PENDETEKSI SEDIMEN
MENGUNAKAN SIMULINK**

LAPORAN TUGAS AKHIR

LAPORAN INI DISUSUN UNTUK MEMENUHI SALAH SATU SYARAT
MEMPEROLEH GELAR SARJANA STRATA SATU (S1) PADA
PROGRAM STUDI TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM SULTAN AGUNG



DISUSUN OLEH:

**MUHAMMAD KEVIN HARDIANSYAH
NIM 30601900033**

**PROGRAM STUDI TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM SULTAN AGUNG
SEMARANG**

2025

FINAL PROJECT

PID TUNING ON SEDIMENT DETECTION BOAT USING SIMULINK

Proposed to complete the requirement to obtain a bachelor's degree (S1) at

Departement of Electrical Engineering

Faculty of Industrial Technology

Universitas Islam Sultan Agung



Composed by:

MUHAMMAD KEVIN HARDIANSYAH

NIM 30601900033

ELECTRICAL ENGINEERING STUDY PROGRAM

FACULTY OF INDUSTRIAL TECHNOLOGY

UNIVERSITAS ISLAM SULTAN AGUNG

SEMARANG

2025

LEMBAR PENGESAHAN PEMBIMBING

Laporan Tugas Akhir dengan judul "PENALAAAN PID PADA KAPAL PENDETEKSI SEDIMEN MENGGUNAKAN SIMULINK" ini disusun oleh:

Nama : Muhammad Kevin Hardiansyah

NIM : 30601900033

Program Studi : Teknik Elektro

Telah disahkan oleh dosen pembimbing pada:

Hari : Kamis

Tanggal : 5 Juni 2025

Pembimbing

LEMBAR PENGESAHAN PEMBIMBING

Prof. Dr. Ir. Sri Arttini Dwi Prasetyowati, M.Si

NIDN. 0620026501

UNISSULA

جامعة سلطان أبو جعفر الإسلامية

Mengetahui,

Ketua Program Studi Teknik Elektro



TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
UNISSULA

100625

U.N.I.S. Sultan Hapsari, S.T., M.T.

NIDN. 0607018501

LEMBAR PENGESAHAN PENGUJI
LEMBAR PENGESAHAN PENGUJI

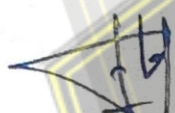
Laporan Tugas Akhir dengan judul "**PENALAAAN PID PADA KAPAL
PENDETEKSI SEDIMEN MENGGUNAKAN SIMULINK**" ini telah
dipertahankan di depan Dosen Penguji Tugas Akhir pada:

Hari : Kamis
Tanggal : 5 Juni 2025

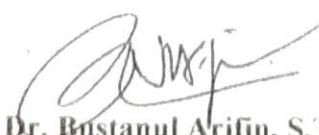
TIM PENGUJI

Anggota Tim Penguji I

Anggota Tim Penguji II


Jennv Putri Hapsari, S.T., M.T.
NIDN. 0607018501


Prof. Dr. Ir. Sri Arttini Dwi Prasetyowati, M.Si.
NIDN. 0620026501


Dr. Bustanul Arifin, S.T., M.T.
NIDN. 0614117701

SURAT PERNYATAAN KEASLIAN TUGAS AKHIR

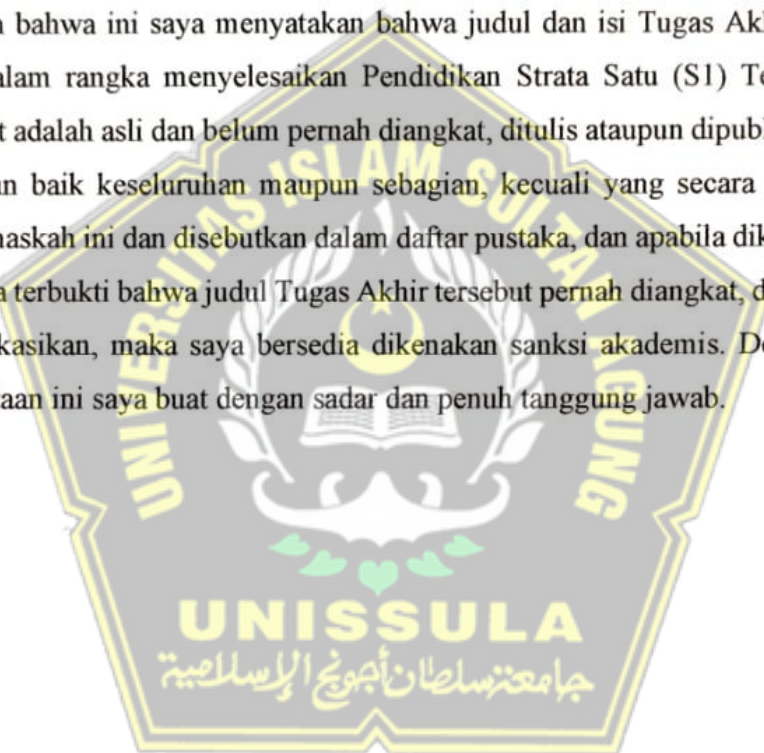
Yang bertanda tangan dibawah ini:

Nama : Muhammad Kevin Hardiansyah

NIM : 30601900033

Judul Tugas Akhir : **PENALAAAN PID PADA KAPAL PENDETEKSI
SEDIMEN MENGGUNAKAN SIMULINK**

Dengan bahwa ini saya menyatakan bahwa judul dan isi Tugas Akhir yang saya buat dalam rangka menyelesaikan Pendidikan Strata Satu (S1) Teknik Elektro tersebut adalah asli dan belum pernah diangkat, ditulis ataupun dipublikasikan oleh siapapun baik keseluruhan maupun sebagian, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka, dan apabila dikemudian hari ternyata terbukti bahwa judul Tugas Akhir tersebut pernah diangkat, ditulis ataupun dipublikasikan, maka saya bersedia dikenakan sanksi akademis. Demikian surat pernyataan ini saya buat dengan sadar dan penuh tanggung jawab.



Semarang, 5 Juni 2025

Yang menyatakan



Muhammad Kevin Hardiansyah

NIM. 30601900033

PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH

Saya yang bertanda tangan dibawah ini:

Nama : Muhammad Kevin Hardiansyah

NIM : 30601900033

Program Studi : Teknik Elektro

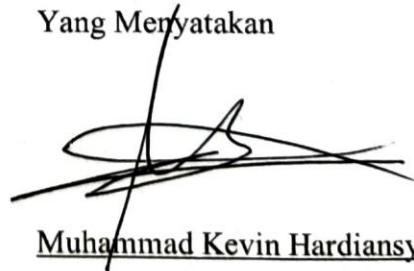
Fakultas : Teknologi Industri

Dengan ini menyatakan Karya Ilmiah berupa Tugas akhir dengan Judul :
**PENALAN PID PADA KAPAL PENDETEKSI SEDIMEN
MENGUNAKAN SIMULINK**

Menyetujui menjadi hak milik Universitas Islam Sultan Agung serta memberikan Hak bebas Royalti Non-Eksklusif untuk disimpan, dialihmediakan, dikelola dan pangkalan data dan dipublikasikan di internet dan media lain untuk kepentingan akademis selama tetap menyantumkan nama penulis sebagai pemilik hak cipta. Pernyataan ini saya buat dengan sungguh-sungguh. Apabila dikemudian hari terbukti ada pelanggaran Hak Cipta/Plagiarisme dalam karya ilmiah ini, maka segala bentuk tuntutan hukum yang timbul akan saya tanggung secara pribadi tanpa melibatkan Universitas Islam Sultan agung.

Semarang, 5 Juni 2025

Yang Menyatakan



Muhammad Kevin Hardiansyah

NIM. 30601900033

HALAMAN PERSEMBAHAN

Dengan penuh rasa syukur kepada Allah Subhanahu wa Ta'ala atas nikmat sehat, nikmat akal, nikmat cinta yang telah dilimpahkan kepada diri saya sehingga saya masih diberikan waktu dan kesempatan menjadi hamba yang lebih baik.

Sebagai bukti rasa kasih sayang dari saya kepada mereka yang selalu memberikan dukungan tanpa henti. Skripsi ini merupakan wujud nyata bahwa saya telah memenuhi kepercayaan yang diberikan kedua orang tua saya selama menempuh pendidikan di Universitas Islam Sultan Agung Semarang. Semoga dengan terselesaikannya skripsi ini, Ayah dan Ibu merasa bangga atas jerih payah dan pengorbanan yang telah diberikan kepada saya selama ini. Harapan saya, karya sederhana ini dapat menjadi salah satu persembahan terbaik untuk Ayah dan Ibu, sebagai tanda bakti dan cinta saya yang tulus.

Saya berjanji akan terus berusaha menjadi anak yang membanggakan dan tidak akan menyia-nyiakan kepercayaan, doa, dan dukungan yang telah Ayah dan Ibu berikan selama ini. Semoga keberhasilan ini menjadi awal dari langkah-langkah besar selanjutnya dan dapat menjadi penyejuk hati untuk Ayah dan Ibu tercinta.

Terima kasih untuk setiap tetes keringat, doa, serta kasih sayang yang tiada henti. Semoga Allah senantiasa membalas segala kebaikan Ayah dan Ibu dengan kebahagiaan dan keberkahan yang tak terhingga.

HALAMAN MOTTO

“No matter what you want out of life, your ability to adapt and move quickly and efficiently through the process of personal evolution will determine your success and your happiness.” — Principles by Ray Dalio

“Barang siapa yang tidak mampu menahan lelahnya belajar, maka ia harus mampu menahan perihnya kebodohan” — *Imam asy-Syafi'i*



KATA PENGANTAR

Dengan penuh rasa syukur ke hadirat Allah Yang Maha Kuasa, berkat limpahan rahmat serta petunjuk-Nya, penulis telah berhasil menuntaskan penulisan skripsi yang mengangkat judul “Penalaan PID Pada Kapal Pendeteksi Sedimen Menggunakan Simulink” dengan baik. Shalawat dan salam senantiasa mengalir kepada panutan umat, Rasulullah Muhammad SAW, sosok yang menjadi pedoman bagi mereka yang mendambakan kasih sayang dan petunjuk-Nya.

Dalam proses pengerjaan skripsi ini, penulis memahami sepenuhnya bahwa keberhasilan ini merupakan hasil dari dukungan dan kontribusi berbagai pihak. Dengan segala kerendahan hati, penulis menyampaikan ucapan terima kasih yang tak terhingga kepada:

1. Allah Subhanahu Wa Ta'ala yang senantiasa mencurahkan kemudahan, keberkahan, serta petunjuk-Nya sehingga karya ilmiah ini dapat terselesaikan dengan baik.
2. Teristimewa kepada kedua orang tua terkasih, Ayahanda Hariyanto dan Ibunda Budi Astuti, yang senantiasa mengalirkan doa-doa terbaik, mencurahkan kasih sayang tanpa batas, serta memberikan dukungan moral dan material yang tak terhingga sepanjang perjalanan studi.
3. Ibu Prof. Dr. Hj. Sri Artini Dwi Prasetyowati, M.Si yang dengan penuh dedikasi sebagai Dosen Wali dan Pembimbing I, memberikan arahan yang cermat serta inspirasi yang membangun.
4. Bapak Dr. Ir. Agus Adhi Nugroho, M.T., yang telah membimbing dengan penuh kesabaran, memberikan masukan yang berharga, serta dorongan semangat yang tak pernah putus.
5. Bapak Dr. Bustanul Arifin, S.T., M.T., atas kesediaannya meluangkan waktu dalam memberikan bimbingan yang konstruktif dan pencerahan akademis yang sangat berarti.
6. Ibu Jenny Putri Hapsari, S.T., M.T., selaku Kepala Program Studi Teknik Elektro UNISSULA yang telah memberikan dukungan administratif dan akademis yang sangat berharga..

7. Bapak Drajat Widiarto, S.T., Laboran Teknik Elektro yang telah memfasilitasi kebutuhan teknis dan memberikan bantuan praktis selama proses penelitian.
8. Saudari Hallasurra Alisain Chandrakirana, S.H., yang selalu memberikan bimbingan, saran, dan motivasi.
9. Saudara Muhammad Khoirun Faza, S.T., selaku mentor yang selalu memberikan bimbingan, saran, dan motivasi.
10. Saudara Muhammad Rifky Aulia, S.T., yang selalu memberikan bimbingan, saran, dan motivasi.
11. Saudari Azizah Nur Maghfiroh yang telah membantu penulis dalam pengumpulan data dan perancangan alat.
12. Ucapan terima kasih juga disampaikan kepada seluruh rekan sejawat dan berbagai pihak yang tidak dapat disebutkan satu per satu, yang telah memberikan bantuan baik secara langsung maupun tidak langsung.

Penulis menyadari sepenuhnya bahwa karya ilmiah ini masih memiliki berbagai keterbatasan dan kekurangan, mengingat keterbatasan pengetahuan serta berbagai kendala yang dihadapi selama proses pengerjaannya. Oleh karenanya, segala bentuk kritik dan saran yang konstruktif sangat diharapkan demi penyempurnaan karya ini di masa mendatang. Besar harapan penulis agar karya ini dapat memberikan manfaat serta menjadi landasan untuk pengembangan penelitian selanjutnya.

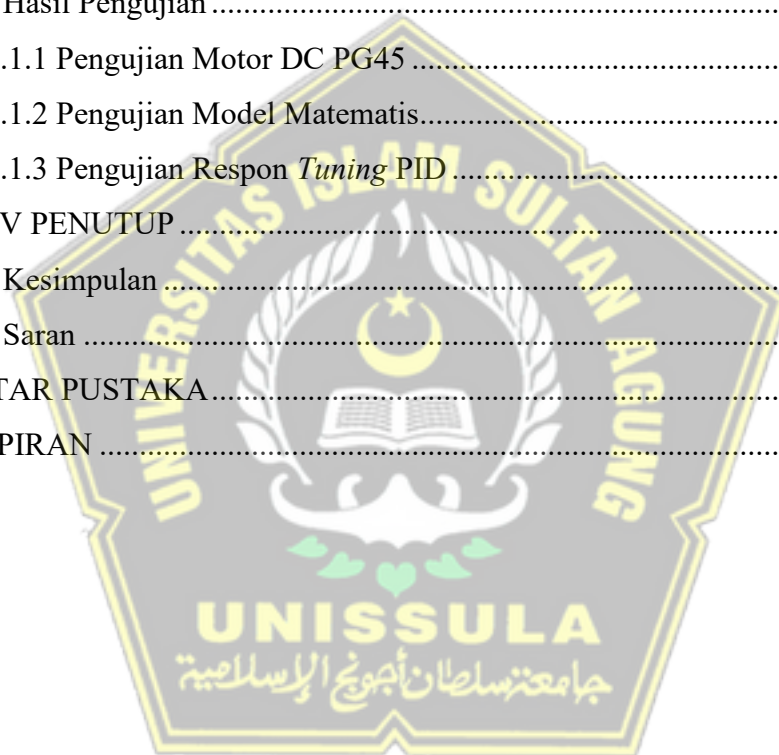
Semarang, 20 Februari 2025

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN PEMBIMBING	iii
LEMBAR PENGESAHAN PENGUJI.....	iv
SURAT PERNYATAAN KEASLIAN TUGAS AKHIR	v
PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH.....	vi
HALAMAN PERSEMBAHAN	vii
HALAMAN MOTTO.....	viii
KATA PENGANTAR.....	ix
DAFTAR ISI	xi
DAFTAR TABEL	xiii
DAFTAR GAMBAR.....	xiv
ABSTRAK.....	xvi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	2
1.3 Pembatasan Masalah.....	2
1.4 Tujuan Penelitian	2
1.5 Manfaat	2
1.6 Sistematika Penulisan	3
BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI.....	5
2.1 Tinjauan Pustaka	5
2.2 Landasan Teori.....	7
2.2.1 Motor DC	7
2.2.2 Baterai Lithium Polimer.....	9
2.2.3 Arduino Mega 2560 R3.....	10
2.2.4 Motor Driver BTS7960	12
2.2.5 <i>Pulse Width Modulation</i> (PWM).....	14
2.2.6 Kontroler	14
2.2.7 <i>Rotary Encoder</i>	17
2.2.8 Penalaan(<i>Tuning</i>) PID.....	18

2.2.9 Pemodelan Sistem Motor DC.....	19
2.2.10 MATLAB	20
BAB III METODE PENELITIAN	22
3.1 Perancangan Blok Diagram Sistem	24
3.2 Spesifikasi Desain.....	24
3.3 Pengujian Motor DC PG45	25
3.4 Membuat Model Matematis Sistem.....	29
3.5 Penalaan(<i>tuning</i>) PID.....	32
BAB IV DATA DAN ANALISA	38
4.1 Hasil Pengujian	38
4.1.1 Pengujian Motor DC PG45	38
4.1.2 Pengujian Model Matematis.....	41
4.1.3 Pengujian Respon <i>Tuning</i> PID	47
BAB V PENUTUP	50
5.1 Kesimpulan	50
5.2 Saran	50
DAFTAR PUSTAKA.....	51
LAMPIRAN	53



DAFTAR TABEL

Tabel 2.1 Spesifikasi motor DC PG45.....	9
Tabel 2.2 Spesifikasi Arduino Mega 2560 R3.....	10
Tabel 2.3 Spesifikasi Motor Driver BTS7960	13
Tabel 4.1 Hasil pengujian Motor DC PG45.....	38
Tabel 4.2 Hasil percobaan <i>tuning trial and error</i>	40



DAFTAR GAMBAR

Gambar 2.1 Rangkaian ekuivalen motor DC magnet permanen	8
Gambar 2.2 Motor DC PG45	9
Gambar 2.3 Baterai Li-Po LPB Power 5200 mAh 3s 12 Volt.....	9
Gambar 2.4 Arduino Mega 2560 R3.....	10
Gambar 2.5 Driver Motor BTS 7960	13
Gambar 2.6 Rangkaian <i>H-Bridge</i>	13
Gambar 2.7 <i>Duty Cycle</i> pada sinyal PWM	14
Gambar 2.8 Diagram Blok Kontroler Proporsional	15
Gambar 2.9 Diagram blok kontroler integral.....	16
Gambar 2.10 Blok Diagram Kontroler Derivatif.....	16
Gambar 2.11 Diagram blok kontroler PID (<i>Proporsional Integral Derivatif</i>)	17
Gambar 2.12 <i>Rotary Encoder</i>	18
Gambar 2.13 Rangkaian penampang motor DC	19
Gambar 3.1 <i>Flowchart</i> penelitian	22
Gambar 3.2 Diagram hubungan antar sub sistem	23
Gambar 3.3 Diagram tingkat kedua sub sistem motor DC	24
Gambar 3.4 Blok Diagram Sistem	24
Gambar 3.5 Membuat <i>workspace</i> dengan nama data_motor	29
Gambar 3.6 Memasukan data <i>input</i> dan <i>output</i> ke dalam <i>workspace</i>	29
Gambar 3.7 Menjalankan perintah <i>systemIdentification</i>	30
Gambar 3.8 Memilih <i>Time domain data</i>	30
Gambar 3.9 <i>Import</i> data dari <i>workspace</i>	31
Gambar 3.10 Memilih <i>Transfer function model</i>	31
Gambar 3.11 Tampilan saat model sudah jadi	32
Gambar 3.12 Melakukan <i>preprocess</i> dengan <i>Remove means</i>	32
Gambar 3.13 Melakukan estimate data yang telah dilakukan <i>preprocess</i>	33
Gambar 3.14 Diagram blok sistem pada simulink.....	34
Gambar 3.15 nilai K_p , K_i , dan K_d	35

Gambar 4.1 Grafik hubungan perubahan PWM terhadap kecepatan putar motor DC PG45	39
Gambar 4.2 Validasi data model dengan data awal	40
Gambar 4.3 Hasil model yang telah dibuat.....	40



ABSTRAK

Penelitian ini merupakan pengembangan dari penelitian sebelumnya tentang alat pengukur volume sedimen di sungai pada aspek pengujian dan metode tuning. Penelitian ini berfokus pada penentuan parameter kontrol PID pada kecepatan putar motor DC PG45 menggunakan simulink menggunakan metode *trial and error*.

Motor DC PG45 bekerja berdasarkan algoritma pemrograman arduino yang telah dirancang sehingga dapat bergerak dengan kecepatan yang telah ditentukan. Penelitian ini melakukan pemodelan sistem kapal pendeteksi sedimen pada Simulink dengan tingkat kemiripan 94.09%.

Hasil penelitian ini menunjukkan bahwa metode penalaan *trial and error* menghasilkan kontrol yang baik pada model sistem kapal pendeteksi sedimen yang telah dirangkai pada simulink dengan nilai $K_p=100$; $K_i=5$; $K_d=15$ didapatkan nilai *rise time* = 0.3183 detik dan *settling time* = 0.4104 detik serta *overshoot* = 1.000%.

Kata kunci: Motor DC, PID, penalaan, simulink, arduino.



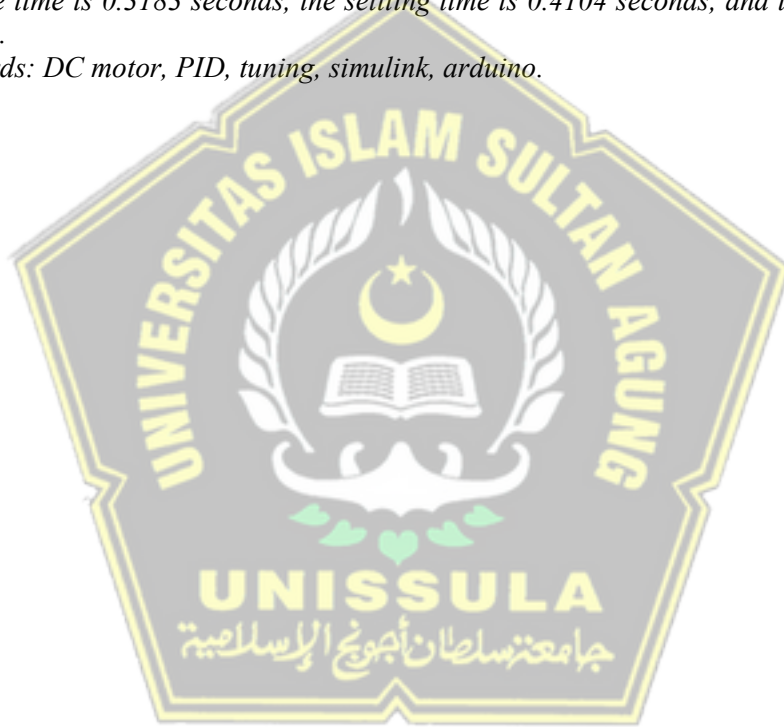
ABSTRACT

This research constitutes a progression of earlier studies conducted on the subject of measuring devices employed for the quantification of sediment volume within fluvial environments. The present study focuses on the testing and calibration methodologies employed in these earlier investigations. The present study is concerned with the identification of the PID control parameters that regulate the rotational speed of the PG45 DC motor. This investigation employs the trial-and-error method, utilising the software environment of Simulink to facilitate the process.

The PG45 DC motor functions in accordance with the Arduino programming algorithm, which has been engineered to facilitate movement at a predetermined velocity. The present study employed a modelling approach to develop a system for detecting sediment, utilising Simulink to achieve a similarity rate of 94.09%.

The findings of this study suggest that the trial-and-error tuning method is effective in achieving satisfactory control over the sediment detection ship system model, which has been assembled in Simulink with the following parameters: $K_p = 100$, $K_i = 5$, $K_d = 15$. The rise time is 0.3183 seconds, the settling time is 0.4104 seconds, and the overshoot is 1.000%.

Keywords: DC motor, PID, tuning, simulink, arduino.



BAB I

PENDAHULUAN

1.1 Latar Belakang

Intensitas kegiatan manusia yang semakin tinggi di sekitar daerah aliran sungai telah berdampak signifikan pada kondisi ekosistem perairan. Salah satu aktivitas yang berkontribusi pada kerusakan adalah praktik deforestasi di wilayah hulu sungai. Hal ini mengakibatkan terjadinya erosi tanah yang masif di sepanjang bantaran sungai. Konsekuensinya, terjadi peningkatan material tersuspensi (*suspended solid*) yang berujung pada sedimentasi dan pendangkalan aliran sungai[1].

Sedimen memainkan peran penting dalam ekosistem sungai. Mereka membantu dalam pembentukan habitat bagi berbagai organisme air, serta berkontribusi terhadap siklus nutrisi. Namun, kelebihan sedimen, yang sering kali diakibatkan oleh deforestasi atau konstruksi, dapat menyebabkan masalah seperti pendangkalan sungai, penurunan kualitas air, dan terganggunya kehidupan akuatik.

Dalam pengelolaan sungai, penting untuk memantau dan mengelola sedimen agar tidak menimbulkan dampak negatif terhadap lingkungan dan infrastruktur. Oleh karena itu, studi mengenai sedimen di sungai menjadi sangat penting untuk menjaga keseimbangan ekosistem dan menghindari bencana alam seperti banjir atau erosi yang berlebihan.

Di Indonesia, perairan yang tercemar sering kali ditemui di sekitar pemukiman, di mana limbah rumah tangga dialirkan ke sungai, membuat air tampak kotor dan keruh. Kotoran yang larut dalam air akhirnya mengendap di dasar perairan, menyebabkan sedimentasi. Oleh karena itu, diperlukan alat yang dapat mendeteksi sedimentasi di perairan dengan mengukur tinggi endapan sedimen di bawah air. Mengetahui adanya endapan sedimen di dasar air memungkinkan untuk mengambil tindakan pencegahan agar tidak menimbulkan dampak yang tidak diinginkan.

Dengan kemajuan teknologi, khususnya di bidang elektronika, memungkinkan pembuatan alat pengukur volume sedimentasi di dasar perairan. Pengukuran ini

dilakukan secara otomatis oleh prototipe perahu yang dilengkapi dengan sensor ultrasonik. Untuk mendukung kerja sensor diperlukan kecepatan perahu yang konstan supaya sensor dapat bekerja dengan akurat dan presisi.

Pada penelitian ini akan membuat kontrol kecepatan motor pada kapal pengukur volume sedimen di dasar sungai menggunakan pendekatan simulasi aplikasi Simulink. Setelah didapatkan kontrol kecepatan motor, kedepannya kontrol tersebut dapat diimplementasikan pada motor penggerak kapal pengukur volume sedimen.

1.2 Perumusan Masalah

Beralaskan latar belakang yang telah diuraikan di atas, dapat ditarik rumusan masalah:

1. Bagaimana cara membuat kontrol kecepatan motor?
2. Bagaimana supaya kecepatan motor konstan?

1.3 Pembatasan Masalah

Pembatasan masalah pada penelitian ini adalah sebagai berikut :

1. Metode penentuan parameter kontrol yang digunakan adalah *trial and error*.
2. Jenis motor yang digunakan dibatasi pada motor DC PG45.
3. Satuan kecepatan putaran yang digunakan adalah RPM (*revolution per minute*).
4. Penelitian ini tidak membahas aspek torsi pada motor yang digunakan.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah:

1. Mendapatkan model matematika motor PG45 dalam bentuk persamaan fungsi alih dengan melakukan identifikasi sistem.
2. Dapat menyimulasikan sistem pada simulink.
3. Mendapatkan nilai kestabilan setelah dilakukan penalaan pada kontrol kecepatan motor DC PG45 dengan menggunakan kontroler PID.

1.5 Manfaat

Adapun Manfaat dari penelitian ini yaitu:

1. Memperkaya literatur ilmiah terkait motor DC PG45 yang digunakan sebagai penggerak pada kapal pendeteksi sedimen.

2. Membuka jalan dan menemukan teknologi baru yang dapat digunakan oleh peneliti selanjutnya.
3. Membantu dalam pemantauan dan pengelolaan sumber daya perairan yang lebih baik.

1.6 Sistematika Penulisan

Dalam penulisan Tugas Akhir ini menggunakan sistematika penulisan yang terbagi menjadi 5 bab, yaitu:

BAB I PENDAHULUAN

Bab ini menguraikan aspek latar belakang, perumusan masalah, pembatasan masalah, tujuan dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI

Bagian ini menyajikan kajian literatur dan fondasi teoretis yang menjadi basis penelitian, termasuk elaborasi tentang komponen-komponen utama yang diimplementasikan seperti mikrokontroler Arduino Mega 2560, sistem *power supply* berbasis baterai, *driver* motor BTS 7960, unit penggerak motor DC PG45, serta komponen pendukung lainnya.

BAB III METODE PENELITIAN

Bab ini mendeskripsikan metodologi penelitian dan tahapan pemodelan sistem, meliputi aspek perancangan *hardware* dan *software* yang terintegrasi dalam implementasi tugas akhir.

BAB IV DATA DAN ANALISA

Seksi ini memuat pembahasan komprehensif mengenai hasil pengujian sistem beserta analisis terhadap performa dan karakteristik alat yang telah dikembangkan.

BAB V PENUTUP

Bab final ini berisi sintesis dari keseluruhan penelitian dalam bentuk kesimpulan, dilengkapi dengan rekomendasi untuk pengembangan dan penelitian lanjutan di masa mendatang.

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

Dalam bab ini, teori dasar dijelaskan secara rinci untuk memudahkan pemahaman tentang fungsi rangkaian dan prinsip dasar konseps sistem. Kerangka teori ini berfungsi sebagai acuan dan pedoman dalam menangani masalah yang muncul selama implementasi penelitian ini.

2.1 Tinjauan Pustaka

“Rancang Bangun Kapal Pengukur Volume Sedimen” adalah tugas akhir dibuat pada tahun 2023 oleh Muhammad Khoirun Faza. Membahas tentang perbandingan nilai asli dan nilai hasil pengukuran alat berupa kapal pengukur volume sedimen. Nilai hasil pengukuran didasarkan pada penggunaan sensor inframerah dan *rotary encoder* sebagai acuannya yang kemudian dihubungkan ke Arduino Mega 2560. Nilai tersebut adalah panjang, lebar, tinggi dan volume dengan standar centimeter. Pada penelitian ini menggunakan algoritma PID sebagai pemrograman yang penggeraknya adalah motor PG45 dan PG28[2].

“Monitoring River Sediment by Optimizing Arduino Capabilities Controlled by the PID Algorithm” adalah jurnal penelitian yang diselesaikan pada tahun 2022 oleh Sri Artini Dwi Prasetyowati, Bustanul Arifin, Akhmad Syakhroni dan Muhammad Khoirun Faza. Pada penelitiannya membuat alat untuk memantau sedimen di sungai dan mengukur volumenya. Hasil dari penelitian ini yaitu perangkat dapat mendeteksi sedimen, mengukur ketinggian sedimen, menelusuri sedimen hingga ukur panjangnya, serta mengukur lebar sedimen. Dengan memanfaatkan arduino dan algoritma PID, perangkat dapat bergerak, mendeteksi bahkan mengukur sedimen secara akurat secara otomatis dan hasil pengukuran menunjukkan bahwa perangkat dapat bekerja dengan baik[3].

Tugas akhir berjudul “Prototype Alat Pengukur Volume Sedimen di Dasar Sungai Berbasis Arduino Mega 2560” dibuat oleh Junido Ardali pada tahun 2019. Fokus utama penelitian ini adalah prototipe alat pengukur volume sedimen yang ditempatkan di dasar sungai, yang menggunakan Arduino Mega 2560 sebagai pengontrol. Prototipe ini dirancang dalam bentuk perahu yang dilengkapi dengan

sensor inframerah yang mampu mengukur volume sedimen dengan menghitung panjang, lebar, dan tinggi sedimen. Desain ini dilengkapi dengan dua motor DC, masing-masing dilengkapi dengan *rotary encoder* untuk memudahkan pengendalian gerakan sensor secara presisi dan menghitung ukuran sedimen. Perhitungan ini dilakukan dengan mewakili rotasi motor DC melalui pengukuran oleh *rotary encoder*[4].

Jurnal “Optimasi Kecepatan Motor DC Menggunakan PID Dengan Tuning Ant Colony Optimization (ACO) Controller”, yang ditulis oleh Mochammad Nur Masrukhan dan timnya, adalah jurnal ilmiah yang mengeksplorasi perpaduan antara penyesuaian PID dan Optimasi Koloni Semut (ACO). Dalam studi ini, model pengendalian kecepatan motor DC dirancang menggunakan beberapa metode pengendalian, yaitu penyesuaian otomatis PID menggunakan Matlab, dan PID dengan penyesuaian menggunakan Optimasi Koloni Semut (ACO). Hasil kinerja model pengendalian kecepatan motor DC menggunakan PID-ACO menunjukkan kesalahan *steady-state*, waktu penyelesaian, dan *overshoot* yang lebih baik dibandingkan dengan pengaturan otomatis PID Matlab dan PID-ZN (Ziegler Nichols PID). Temuan studi ini menunjukkan bahwa pengendali PID-ACO adalah yang paling efektif, dengan waktu penyelesaian 0,55 detik dan *overshoot* 1,017, mewakili pengendali tercepat dan paling akurat dalam studi ini[5].

“Pengendalian Simulator Orde Dua Dengan Metode Auto Tuning PID Berbasis Kontroler TK4S-B4CR” adalah jurnal yang ditulis oleh Luthfi Aditya Suyitno dan tim membahas tentang metode *trial and error*, Ziegler-Nichols I, dan *auto tuning* yang digunakan untuk mendapatkan parameter PID. Metode Ziegler-Nichols I dan *auto tuning* memiliki pendekatan yang berbeda dalam penalaan sistem. Metode pertama bergantung pada kemampuan operator dalam menganalisis respons *plant*, sedangkan metode kedua mengandalkan kapabilitas kontroler untuk memahami karakteristik *plant*. Penggunaan model simulator orde dua sebagai sistem yang dikendalikan mampu merepresentasikan kondisi riil *plant*, dengan respons yang adaptif terhadap karakteristik kontroler. Hasil pengujian menunjukkan bahwa kontroler TK4S-B4CR dengan metode *auto tuning* PID mampu mengatasi

gangguan sebesar 30%, menghasilkan *error steady state* 0% dan membatasi *overshoot* pada level 6.8%[6].

“*System Identification and Control Strategy on Electric Power Steering DC Motor*” adalah jurnal penelitian yang dilakukan oleh Bustanul Arifin, Agus Adhi Nugroho, Eka Nuryanto Budisusila, Muhammad Khosyi’in pada tahun 2024. Jurnal ini membahas metode sub-ruang identifikasi sistem yang digunakan untuk mendapatkan pemodelan matematis motor DC EPS. Berdasarkan penelitian tersebut, kontrol Haugen menunjukkan hasil yang paling baik dari kedua kontrol lainnya, yaitu dengan nilai *rise time* sebesar 11,361 ms, *overshoot* sebesar 6,898%, dan *steady state* pada waktu 1,3 s. Hal ini menunjukkan bahwa kontrol PI dengan metode *tuning* Haugen dapat mengendalikan motor dengan baik[7].

Berbeda dengan penelitian-penelitian sebelumnya, pada penelitian ini penulis membuat kontrol motor DC PG45 dengan pendekatan simulasi menggunakan aplikasi Simulink. Kontribusi utama dari penelitian ini terletak pada pendekatan baru untuk strategi *tuning* untuk kontrol motor DC PG45, pemodelan berdasarkan eksperimen didasarkan pada hasil pengukuran data *input* dan *output* melalui identifikasi sistem.

2.2 Landasan Teori

2.2.1 Motor DC

Motor DC digunakan dalam berbagai aplikasi industri karena kemudahannya kecepatan dapat dikontrol[8]. Dalam sistem pengendalian posisi dan kecepatan, motor DC menjadi aktuator yang paling umum digunakan sebagai komponen pengendali akhir. Operasional motor DC mengikuti prinsip hukum Lorentz, di mana ketika arus mengalir melalui kumparan *armature* motor DC dan kumparan medan diberikan eksitasi, akan menghasilkan Gaya Lorentz pada setiap bagian kumparan *armature*.

Motor Planetary Gear (PG) 45 merupakan tipe motor DC brushless produksi Oriental Motor corp. Motor ini sudah dilengkapi gearbox dengan perbandingan yang bervariasi. Motor Planetary Gear (PG) 45 memiliki kecepatan maksimum 468,7 rpm. Dengan pembulatan maka kecepatan maksimum pada poros keluaran adalah 500 rpm, dengan torsi mencapai 15 kg/cm. Motor ini sering digunakan pada

bagian penggerak dari sebuah robot. Untuk bekerja motor ini membutuhkan suplai tegangan sebesar 24 V. Untuk mengatur kecepatan dan arah putaran motor digunakan rangkaian elektronik penggerak motor (*motor driver*). Bentuk rangkaian elektronik penggerak motor yang digunakan adalah *H-Bridge*. Driver motor akan mengolah sinyal modulasi lebar pulsa (*Pulse Wide Modulation*, PWM) dalam mengatur kecepatan putar motor, driver motor akan mengatur tegangan yang disuplai ke motor. Besarnya nilai tegangan yang diberikan ke motor akan mempengaruhi kecepatan putar mesin.

Rangkaian ekuivalen motor DC magnet permanen dapat dilihat dalam Gambar 2.1.



Gambar 2.1 Rangkaian ekuivalen motor DC magnet permanens[9]

Salah satu jenis motor DC magnet permanen adalah seperti terlihat dalam Gambar 2.2.



Gambar 2.2 Motor DC PG45

Tabel 2.1 Spesifikasi motor DC PG45[2]

No	Spesifikasi	Keterangan
1	Tipe Motor	<i>Brushed</i> motor
2	Rasio	1:19.2
3	Frekuensi putar	9000 RPM
4	<i>Output RPM Gearbox</i>	468.75
5	As Motor	10mm
6	Arus	2.5A-5A
7	<i>Eff Torque</i>	$\pm 15 \text{ kgfcm}$
8	<i>Force Torque</i>	$\pm 25 \text{ kgfcm}$
9	<i>Torque Stall</i>	$\pm 30 \text{ kgcm}$
10	<i>Power rated</i>	100W

2.2.2 Baterai Lithium Polimer



Gambar 2.3 Baterai Li-Po LPB POWER 5200 mAh 3s 12 Volt[9]

Teknologi baterai Lithium Polymer (Li-Po) merupakan pengembangan dari baterai Lithium Ion (Li-Ion), dengan inovasi utama pada penggunaan elektrolit polimer kering berbentuk lapisan film tipis, menggantikan elektrolit cair. Sistem ini mengadopsi struktur berlapis antara anoda dan katoda untuk mengakomodasi pertukaran ion, sekaligus memberikan keunggulan dalam hal fleksibilitas bentuk dan ukuran. Namun, implementasi elektrolit polimer kering ini menghadapi kendala berupa rendahnya efisiensi pertukaran ion, yang mengakibatkan keterbatasan pada laju *charging* dan *discharging*[10].

2.2.3 Arduino Mega 2560 R3



Gambar 2.4 Arduino Mega 2560 R3[2]

Arduino dikembangkan sebagai mikrokontroler *single-board* bersifat open *source* yang berasal dari evolusi *wiring* platform, dengan tujuan menyederhanakan penerapan teknologi elektronika di berbagai bidang. Sistem ini mengintegrasikan prosesor Atmel AVR pada sisi *hardware* dan didukung bahasa pemrograman khusus untuk pengembangannya. Khusus untuk model Arduino Mega 2560 R3, sistem dapat dioperasikan menggunakan daya dari koneksi USB atau melalui sumber eksternal yang dapat berupa adaptor daya atau sistem baterai[2].

Sistem *board* ini dirancang untuk beroperasi dengan catu daya eksternal pada rentang 6 hingga 20 volt. Ketika tegangan sumber berada di bawah 7 volt, *output* pada pin 5 volt akan mengalami penurunan di bawah nilai nominalnya, mengakibatkan ketidakstabilan operasional *board* arduino. Di sisi lain, pemberian tegangan *input* yang melampaui 12 volt dapat menyebabkan pemanasan berlebih pada regulator tegangan, yang berujung pada kerusakan *board* Arduino[4]. Oleh karena itu direkomendasikan tegangan input sebesar 7 hingga 12 volt.

Tabel 2.2 Spesifikasi Arduino Mega 2560 R3

No	Spesifikasi	Keterangan
1.	<i>Microcontroller</i>	ATmega2560
2.	<i>Operating Voltage</i>	5V
3.	<i>Input Voltage (recommended)</i>	7-12V
4.	<i>Input Voltage (limit)</i>	6-20V
5.	Digital I/O Pins	54 of which 15 provide PWM output

6.	Analog Input Pins	16
7.	DC Current per I/O Pin	20 Ma
8.	DC Current for 3.3V Pin	50 mA
9.	Flash Memory	256 KB of which 8 KB used by bootloader
10.	SRAM	8 KB
11.	EEPROM	4 KB
12.	Clock Speed	16 MHz
13.	LED_BUILTIN	13
14.	Length	101.52 mm
15.	Width	53.3 mm
16.	Weight	37 g

Beberapa pin Arduino Mega 2560 R3 yang memiliki fungsi khusus :

1. Serial, terdiri dari empat port individu, dengan masing-masing port memiliki dua pin. berikut adalah daftar lengkap port serial dan pin masing-masing: nomor seri 0 terkait dengan pin 0 (RX) dan pin 1 (TX). Dalam konfigurasi serial, pin nomor 19 ditugaskan untuk penerimaan (RX) dan pin nomor 18 untuk transmisi (TX). Dalam konfigurasi serial 2, pin 17 ditugaskan untuk RX dan pin 16 untuk TX. Dalam konfigurasi serial ketiga, pin 15 dan 14 ditugaskan untuk RX dan TX masing-masing. Pin 'RX' ditugaskan untuk penerimaan data, sedangkan pin 'TX' digunakan untuk transmisi data serial TTL. Penting untuk dicatat bahwa pin 0 dan 1 digunakan oleh chip USB-to-TTL ATmega16U2.
2. *External Interrupts*, yaitu pin 2 (untuk *interrupt* 0), pin 3 (*interrupt* 1), pin 18 (*interrupts*), pin 19 (*interrupt* 4), pin 20 (*interrupt* 3), dan pin 21 (*interrupt* 2). Pada Arduino Mega 2560 tersedia 6 buah pin *interrupt* yang digunakan untuk fungsi `attachInterrupt()`.
3. Pin 20 (SDA) dan pin 21 (SCL) berfungsi untuk mendukung komunikasi TWI, yang dapat dijalankan melalui penggunaan *Wire Library*.

4. LED bawaan pada papan ini dikontrol melalui pin digital nomor 13, di mana LED tersebut memang sudah langsung terhubung ke pin 13.
5. PWM adalah pin yang menyediakan *output* PWM 8-bit dengan menggunakan fungsi `analogWrite()`. Pin PWM tersedia pada pin 2 sampai dengan 13 dan pin 44 sampai dengan 46,
6. SPI terletak pada pin 50 (MISO), 51 (MOSI), 52 (SCK), dan 53 (SS).
Pin-pin ini mendukung komunikasi SPI melalui penggunaan *library* SPI.

Platform Arduino Mega 2560 R3 dilengkapi dengan 16 pin input analog, masing-masing dengan kapabilitas resolusi 10 bit (menghasilkan 1024 nilai diskrit). Pengukuran default pin-pin ini menggunakan referensi ground hingga 5V, dengan opsi penggunaan pin AREF melalui implementasi fungsi `analogReference()`. Board ini juga memiliki beberapa pin tambahan dengan fungsi spesifik, yaitu:

1. AREF berfungsi menyediakan tegangan referensi untuk operasional input analog.
2. Reset berfungsi mengaktifkan fungsi reset mikrokontroler ketika diberi sinyal LOW, fungsinya setara dengan tombol reset yang tersedia pada *board*.

2.2.4 Motor Driver BTS7960

Dalam sistem ini, pengendalian motor DC diimplementasikan menggunakan *driver* motor BTS7960. Modul ini berfungsi mengontrol dua parameter utama motor: arah rotasi dan kecepatan putaran, yang responnya ditentukan oleh besaran tegangan *input* yang diberikan pada *driver*. Untuk perancangan sistem ini, dipilih *driver* motor BTS7960 yang visualisasinya dapat dilihat pada Gambar 2.5.

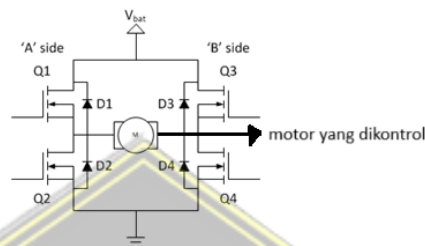


Gambar 2.5 *Driver* Motor BTS 7960[11]

Modul *driver* motor DC ini didesain dengan kapabilitas mengalirkan arus maksimal 43A dan dilengkapi dengan fitur PWM. Sistem ini beroperasi dengan rentang tegangan sumber DC 5.5V hingga 27VDC, sementara level tegangan *input*

berkisar antara 3.3V sampai 5VDC. Mengadopsi konfigurasi *full H-bridge* berbasis IC BTS7960, *driver* ini dilengkapi dengan sistem proteksi terhadap *overheating* dan *overcurrent*[12].

Sistem pengendali arah putaran motor diimplementasikan melalui rangkaian yang dikenal sebagai *H-Bridge*. Penamaan ini didasarkan pada topologi rangkaiannya yang memiliki kemiripan visual dengan huruf H, sebagaimana diilustrasikan dalam Gambar 2.6.



Gambar 2.6 Rangkaian *H-Bridge*[9]

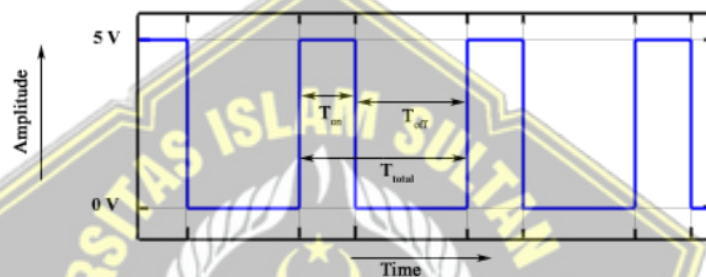
Sistem rangkaian menggunakan kombinasi dua MOSFET kanal P dan dua MOSFET kanal N. Mekanisme kerjanya diatur melalui pola switching keempat MOSFET tersebut. Dalam konfigurasi dimana segmen atas terhubung ke terminal positif dan segmen bawah ke terminal negatif catu daya, pengaktifan Q1 dan Q4 bersamaan dengan penonaktifan Q2 dan Q3 menciptakan jalur arus yang menghubungkan sisi kiri motor ke kutub positif dan sisi kanan ke kutub negatif, menghasilkan rotasi *clockwise*.

Tabel 2.3 Spesifikasi Motor Driver BTS7960

No	Spesifikasi	Keterangan
1.	<i>Input Voltage</i>	6 ~ 27VDC
2.	<i>Driver</i>	Dual BTS7960 H Bridge Configuration
3.	<i>Peak current</i>	43-Amp
4.	<i>PWM capability</i>	up to 25kHz
5.	<i>Control input level</i>	3.3 ~ 5V
6.	<i>Control Mode</i>	PWM or level
7.	<i>Working Duty Cycle</i>	0 ~ 100%
8.	<i>Board Size</i>	50mm x 50mm x 43mm
9.	<i>Weight</i>	~66g

2.2.5 Pulse Width Modulation (PWM)

Dalam sistem *pulse width modulation* (PWM), modulasi dicapai dengan menyesuaikan lebar pulsa sambil menjaga frekuensi dan amplitudo tetap konstan, yang mengakibatkan variasi pada tegangan rata-rata. PWM diterapkan dalam pengendalian kecepatan motor DC menggunakan prinsip siklus kerja, di mana siklus kerja dapat disesuaikan antara 0% dan 100%. Hal ini menentukan kecepatan motor. Siklus kerja adalah rasio antara durasi keadaan ON dan durasi keadaan OFF dalam satu siklus penuh. *Duty cycle* dapat diekspresikan sebagai rasio atau persentase. Sementara itu, frekuensi adalah jumlah siklus gelombang yang terjadi per detik.



Gambar 2.7 *Duty Cycle* pada sinyal PWM[2]

t_{total} merupakan waktu satu siklus atau penjumlahan antara t_{on} dengan t_{off} atau periode satu gelombang.

$$t_{total} = t_{on} + t_{off} \quad (2.6)$$

Siklus kerja atau *duty cycle* sebuah gelombang didefinisikan sebagai,

$$d = \frac{t_{on}}{(t_{on} + t_{off})} \times 100\% \quad (2.7)$$

dimana:

t_{on} = waktu pulsa “High”

t_{off} = waktu pulsa “Low”

d = *duty cycle* (lamanya pulsa *high* dalam satu perioda)

2.2.6 Kontroler

Dalam sistem kendali, kontroler merupakan subsistem dinamis yang diintegrasikan untuk memodifikasi persamaan matematis suatu *plant*. Istilah kontroler sering juga disebut sebagai kompensator atau pengendali[11]. Fungsi esensial dari kontroler adalah meminimalisasi *error*, yang didefinisikan sebagai

deviasi antara nilai *setpoint* dan *output* aktual *plant*. Dalam hal ini, *setpoint* berperan sebagai nilai referensi yang ditargetkan, sementara *output plant* merepresentasikan nilai riil yang terukur. Efektivitas sistem kontrol dinilai dari kemampuannya menekan nilai *error* sekecil mungkin.

Dalam operasinya, kontroler menjalankan tiga fungsi sekuensial: membandingkan nilai aktual plant dengan nilai referensi (*setpoint*), menghitung besaran *error* yang terjadi, dan menghasilkan sinyal kendali untuk mereduksi *error*[13].

2.2.6.1 Kontroler Proporsional

Kontroler proporsional merupakan jenis kontroler yang dicirikan oleh kemampuannya mengakselerasi respons keluaran. Relasi matematis antara *output* kontroler $u(t)$ dengan sinyal *error* $e(t)$ dapat diekspresikan melalui persamaan berikut:

$$u(t) = K_p e(t) \quad (2.8)$$

atau, dalam fungsi alih

$$\frac{U(s)}{E(s)} = K_p \quad (2.9)$$



Gambar 2.8 Diagram Blok Kontroler Proporsional[9]

Terlepas dari mekanisme spesifik dan sifat dari gaya penggerak, pengendali proporsional pada dasarnya dapat dianggap sebagai penguat dengan *gain* yang dapat disesuaikan[13].

2.2.6.2 Kontroler Integral

Dalam sistem kendali, kontroler integral berperan meminimalisasi *error* yang timbul dari penggunaan kontroler proporsional. Karakteristiknya ditandai dengan perubahan *output* kontroler $u(t)$ yang proporsional terhadap laju *error* $e(t)$.

Formulasi matematis kontroler integral (I) dapat diekspresikan melalui persamaan berikut[13]:

$$u(t) = K_i \int_0^t e(t) dt \quad (2.10)$$

$$\frac{U(s)}{E(s)} = \frac{K_i}{s} \quad (2.11)$$

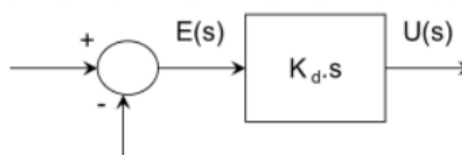
Ekspresi matematis di atas menggambarkan fungsi alih kontroler integral (I), dengan parameter K_i sebagai konstanta integral yang bersifat *adjustable*. Sistem ini memiliki karakteristik dimana nilai $m(t)$ tetap stabil saat $e(t)$ bernilai nol. Kontroler integral, yang juga dikenal dengan istilah kontrol *reset*, divisualisasikan dalam bentuk diagram blok pada Gambar 2.9.



Gambar 2.9 Diagram blok kontroler integral[9]

2.2.6.3 Kontrol Derivatif

Kontroler dengan parameter derivatif menghasilkan *output* yang proporsional terhadap laju perubahan *error*. Intensitas aksi kontrol berbanding lurus dengan kecepatan perubahan *error*. Karakteristik derivatif yang berbasis pada turunan *error* menghasilkan respons impulsif selama periode transien. Dalam aplikasi praktis, kontroler derivatif tidak dapat diimplementasikan secara independen karena ketidakmampuannya merespons *error* yang bersifat statis. Konfigurasi kontroler Derivatif diilustrasikan dalam diagram blok pada Gambar 2.10



Gambar 2.10 Blok diagram kontroler derivatif[9]

Persamaan kontroler Derivatif dalam Laplace adalah sebagai berikut:

$$U(s) \cdot E(s) = Kd \cdot s \quad (2.12)$$

Stabilitas sistem dapat ditingkatkan melalui implementasi parameter derivatif yang memiliki kepekaan tinggi terhadap perubahan *error*. Kemampuannya dalam memberikan efek peredaman pada sistem yang berosilasi membuka peluang untuk mengoptimalkan nilai K_p ke level yang lebih tinggi[13].

2.2.6.4 Pengendali PID (*Proporsional Integral Derivative*)

Penggunaan Pengendali *Proporsional-Integral-Derivative* (PID) dapat mempercepat respons keluaran sekaligus menekan terjadinya kesalahan secara bersamaan. Sistem ini umumnya diaplikasikan pada situasi di mana diperlukan pengoptimalan kinerja, baik terkait respons transien maupun pada kondisi tunak. Dalam implementasinya di lapangan, nilai *gain* derivatif sering kali disetel serendah mungkin guna menghindari meningkatnya sensitivitas terhadap gangguan *noise*[13].

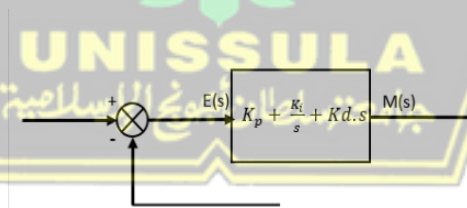
Persamaan kontroler *proporsional integral derivative* adalah

$$u(t) = K_p \cdot e(t) + K_i \cdot \int_0^t e(t) dt + K_d \cdot \frac{de(t)}{dt} \quad (2.13)$$

Adapun fungsi alihnya adalah

$$\frac{M(s)}{E(s)} = K_p + \frac{K_i}{s} + K_d \cdot s \quad (2.14)$$

Gambar 2.11 menunjukkan diagram blok kontroler *proporsional integral derivative*.



Gambar 2.11 Diagram blok kontroler PID (*Proporsional Integral Derivative*)[9]

2.2.7 Rotary Encoder

Sebagai perangkat elektromekanik, *rotary encoder* dirancang untuk memantau pergerakan serta posisi. Prinsip kerjanya melibatkan penggunaan sensor optik yang menghasilkan rangkaian pulsa, yang merepresentasikan data terkait gerakan, posisi, dan arah. Melalui teknologi ini, rotasi poros dapat dikonversikan menjadi kode digital sehingga memungkinkan pemrosesan lebih lanjut oleh sistem kendali[4].

Perangkat ini banyak diaplikasikan dalam berbagai sistem, termasuk robotika, kendali motor, dan aplikasi sejenis.

Spesifik untuk implementasi ini, digunakan *incremental rotary encoder* yang berperan sebagai sensor kecepatan putar motor. *Output* sensor berupa pulsa elektrik yang berfungsi sebagai *input* mikrokontroler. Visualisasi komponen *rotary encoder* pada motor DC PG45 disajikan dalam Gambar 2.12



Gambar 2.12 Rotary Encoder[9]

Dengan spesifikasi 200 P/R pada *incremental rotary encoder* yang digunakan, setiap putaran lengkap akan menghasilkan 200 pulsa. Kecepatan putaran motor dapat dikalkulasi menggunakan persamaan berikut, berdasarkan waktu sampling yang telah ditetapkan pada 50 ms:

$$rpm = \frac{n}{x} \cdot \frac{1000}{T} \cdot 60 \quad (2.15)$$

dengan:

n = jumlah pulsa yang terbaca dalam satu kali rotasi

x = resolusi pulsa *rotary encoder*

T = waktu penyuplikan (ms)

2.2.8 Penalaan(*Tuning*) PID

Terdapat beragam metode untuk melakukan *tuning* kontroler PID, mencakup: *loop tuning*, pendekatan analitis, teknik optimasi, metode *pole placement*, sistem *auto tuning*, serta pendekatan *trial and error*[14]. Pada skripsi ini digunakan cara *trial and error*. Åström & Hägglund[14] menguraikan prosedur *tuning trial and error* dalam empat tahapan:

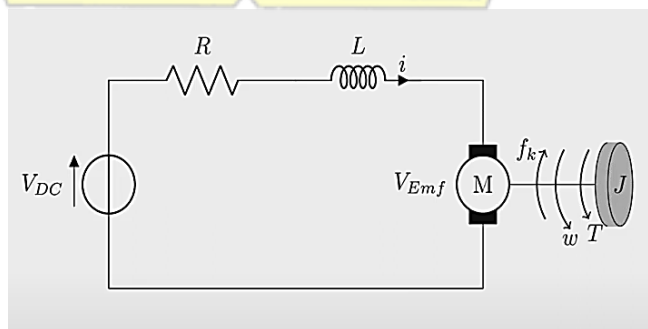
1. Inisiasi dengan menon-aktifkan komponen integral dan diferensial melalui penetapan $K_i = 0$ dan $K_d = 0$.

2. Eksekusi penyesuaian nilai K_p sampai mencapai karakteristik respons yang ditargetkan, tanpa mempertimbangkan *offset* dari *setpoint*.
3. Peningkatan nilai K_d secara bertahap, bersamaan dengan kenaikan K_p , untuk meminimalisasi *overshoot*.
4. Penyesuaian nilai K_i untuk mengompensasi *offset* yang terjadi.

Metode *trial and error* menawarkan keunggulan berupa implementasi yang dapat dilakukan secara langsung dan *real-time*, memungkinkan observasi instan terhadap respons sistem setelah modifikasi parameter K_p , K_i , dan K_d . Namun, keterbatasan metode ini terletak pada sulitnya memastikan apakah konfigurasi parameter kontroler yang dihasilkan telah mencapai nilai optimal.

2.2.9 Pemodelan Sistem Motor DC

Langkah pertama dalam desain sistem kontrol adalah untuk mendapatkan model matematis dari plant atau objek kontrol[13]. Pada Gambar 2.13 adalah penampang motor DC yang di dalamnya terdiri dari beberapa bagian yaitu resistor, induktor, rotor, sumber tegangan. Pada motor DC dapat diberikan sumber yaitu tegangan dari baterai, aki, maupun sumber tegangan. Kemudian motor DC memiliki dua buah output yaitu posisi sudut (*angular motion*) dan kecepatan sudut (*angular velocity*). Motor DC ketika berputar memiliki kecepatan sudut (ω), gaya gesek (f_k), massa/momen inersia (J), dan Torsi (T). Untuk mencari model matematis dari motor DC dengan perhitungan matematis manual dapat dilakukan analisa pada bagian input yaitu elektrik dan pada bagian keluaran yaitu mekanikal motor DC. Pada bagian elektrik dapat dianalisis dengan hukum Kirchoff tegangan dan pada bagian mekanikal dapat dianalisis menggunakan hukum Newton tentang rotasi[8].



Gambar 2.13 Rangkaian penampang motor DC[15]

Untuk membuat pemodelan matematis motor DC menggunakan Matlab diperlukan data *input* dan *output* motor DC. Data *input* berupa tegangan dan *output* berupa posisi sudut atau kecepatan sudut[16]. Supaya model matematis yang didapat mendekati kemiripan yang tinggi dengan sistem riil yang diamati, diperlukan data *output* yang akurat. Dapat dilakukan pengukuran *output* dua cara berbeda. Untuk mengetahui *error relative*, menggunakan rumus[17]:

$$Error\ relatif(\%) = \left| \frac{Nilai\ pada\ alat\ ukur - Nilai\ pada\ serial\ monitor}{Nilai\ pada\ alat\ ukur} \right| \times 100 \quad (2.16)$$

2.2.10 MATLAB

Matlab merupakan platform komputasi numerik sekaligus bahasa pemrograman komputer generasi keempat yang sangat populer untuk menyelesaikan berbagai masalah terkait proses komputasi di berbagai bidang. Sebagai bahasa pemrograman tingkat tinggi, Matlab menggunakan matriks sebagai dasar utama dalam proses analisis dan perhitungan. Pengguna dimudahkan untuk melakukan manipulasi matriks, memvisualisasikan fungsi dan data, menerapkan algoritma, membuat antarmuka pengguna, serta mengintegrasikan Matlab dengan bahasa pemrograman lain.

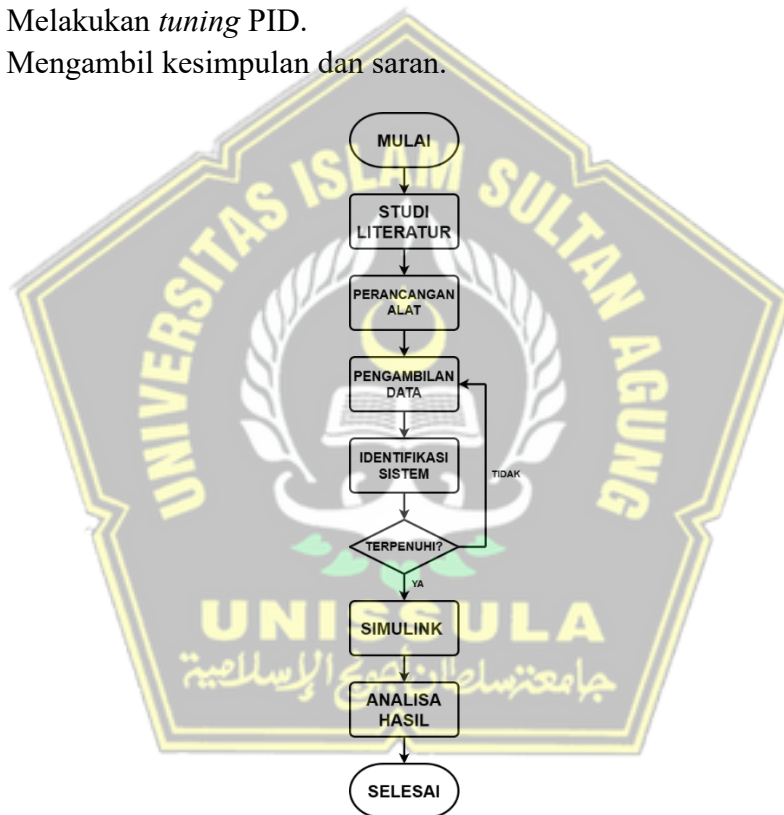
Sementara itu, Simulink adalah ekstensi grafis dari Matlab yang memungkinkan pengguna untuk memodelkan dan mensimulasikan sebuah sistem. Di dalam Simulink, sistem direpresentasikan dalam bentuk diagram blok, yang dapat berupa fungsi transfer, operasi penjumlahan (sum), maupun perangkat input dan output virtual seperti function generator serta scope yang terdapat pada library Simulink. Pada Simulink, data yang berasal dari berbagai blok dapat dikirim dan dihubungkan ke blok lain menggunakan garis penghubung.

BAB III

METODE PENELITIAN

Guna menjawab rumusan masalah dan mewujudkan tujuan penelitian, dibutuhkan serangkaian langkah metodis yang terencana. Berikut adalah uraian metode yang digunakan dalam penelitian ini.

1. Merancang blok diagram dari sistem.
2. Menentukan spesifikasi desain.
3. Pengujian Motor DC.
4. Membuat model matematis sistem.
5. Melakukan *tuning* PID.
6. Mengambil kesimpulan dan saran.



Gambar 3.1 Flowchart penelitian

Penelitian yang dilakukan oleh penulis adalah pengembangan dari skripsi yang berjudul “Prototipe Sistem Pengontrol Kapal Dan Pendeteksi Sedimen”. Kapal pendeteksi sedimen adalah alat yang memiliki fungsi mendeteksi dan mengukur volume sedimen di dasar sungai secara otomatis. Proses deteksi dilakukan dengan tujuan mencegah pendangkalan sungai dan memastikan aliran sungai lancar. Sedimen yang menumpuk di dasar sungai dapat menyebabkan pendangkalan,

mengurangi kapasitas sungai untuk menampung air hujan, sehingga meningkatkan risiko banjir serta dengan mengetahui volume sedimen, langkah-langkah pengangkatan atau pengerukan dapat dilakukan untuk mencegah penyumbatan aliran.

Kapal pendeteksi sedimen dibagi menjadi beberapa sub sistem:

1. Sub sistem Suplai Daya

Komponen subsistem suplai daya memiliki fungsi utama untuk mendistribusikan energi listrik ke berbagai elemen elektronis yang terpasang pada sistem.

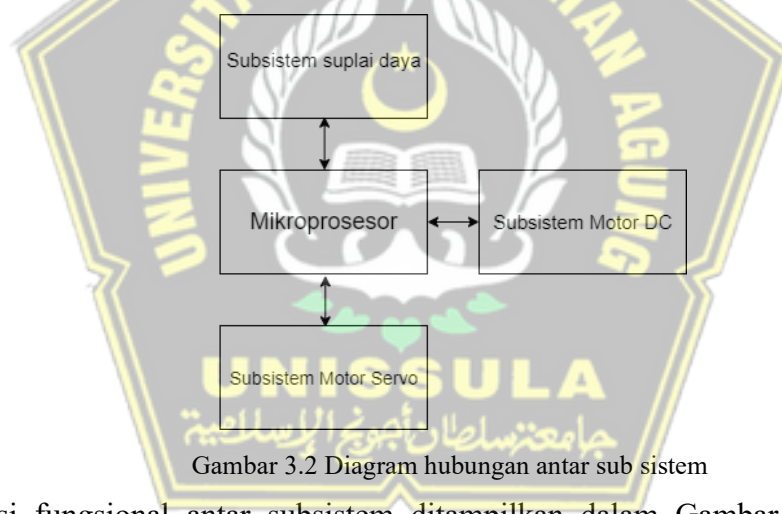
2. Sub sistem Motor Servo

Motor Servo pada sub sistem ini berperan dalam navigasi atau penentuan arah kapal, sehingga kapal dapat bergerak baik ke kanan maupun ke kiri..

3. Sub sistem motor DC

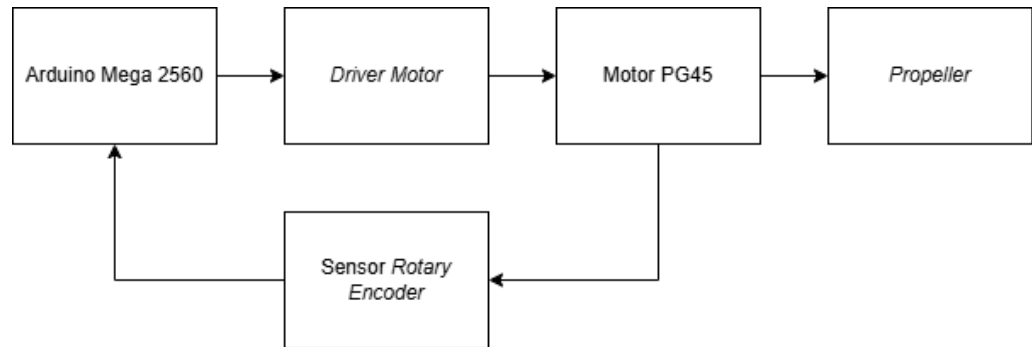
Sub sistem motor DC sebagai sistem penggerak kapal.

Interkoneksi antar subsistem divisualisasikan dalam Gambar 3.2.



Gambar 3.2 Diagram hubungan antar sub sistem

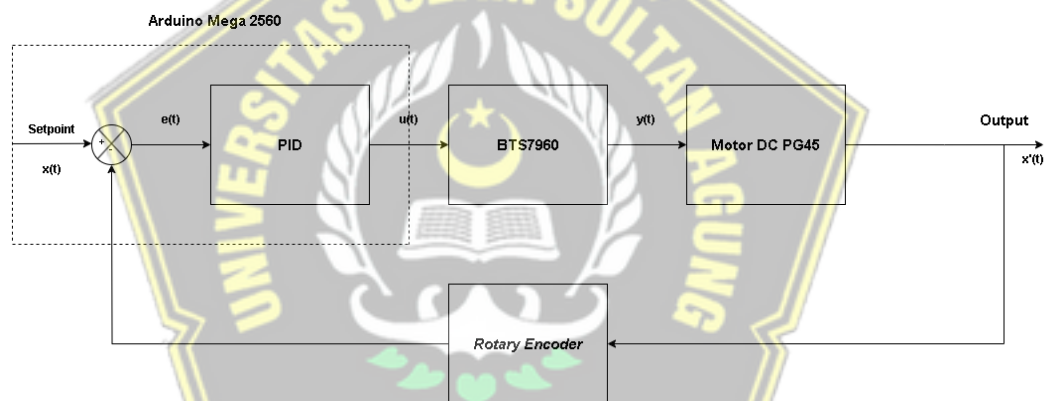
Relasi fungsional antar subsistem ditampilkan dalam Gambar 3.2. Lingkup penelitian ini difokuskan pada optimasi subsistem motor DC, terutama dalam hal pengaturan kecepatan untuk mempertahankan rotasi yang stabil ketika sensor sedimen mengidentifikasi objek. Struktur subsistem motor DC dijabarkan dalam diagram pada Gambar 3.3.



Gambar 3.3 Diagram tingkat kedua sub sistem motor DC

3.1 Perancangan Blok Diagram Sistem

Diagram blok sistem memberikan gambaran komprehensif tentang arsitektur sistem, yang didesain untuk mencapai performa sesuai dengan parameter yang ditentukan. Konfigurasi sistem kontrol kecepatan diilustrasikan dalam format diagram blok pada Gambar 3.4.



Gambar 3.4 Blok Diagram Sistem

Keterangan:

1. *Input* referensi $x(t)$, berupa target kecepatan putar motor dalam RPM.
2. *Error* sistem $e(t)$, yang mengindikasikan deviasi antara nilai *setpoint* dan nilai aktual.
3. Tegangan kontrol $u(t)$ dengan rentang 0-5 Volt
4. Tegangan *output* $y(t)$ dalam rentang 0-24 Volt.
5. Kecepatan aktual motor $x'(t)$ yang terukur dalam RPM.

3.2 Spesifikasi Desain

Sistem dikembangkan dengan spesifikasi sebagai berikut:

1. Mekanisme kontrol berbasis algoritma PID.

2. Unit kontrol utama menggunakan Arduino Mega2560.
3. Proses *coding* diimplementasikan melalui Arduino IDE
4. Sistem diharapkan mampu mempertahankan *output* pada level *setpoint* atau nilai yang mendekatinya.

3.3 Pengujian Motor DC PG45

Uji coba terhadap motor DC bertujuan memperoleh data terkait beberapa aspek, yakni tegangan input yang diberikan ke motor DC (V), kecepatan rotasi aktual motor (RPM), serta kecepatan rotasi yang terpantau melalui serial monitor (RPM). Pengambilan data kecepatan rotasi nyata dan yang terbaca pada serial monitor dimaksudkan sebagai upaya validasi akurasi sistem, mengingat adanya perbedaan antara hasil pembacaan sistem dan nilai yang diperoleh dari alat ukur baku. Sementara itu, pengukuran terhadap tegangan *input* berguna untuk menganalisis kaitan antara variasi tegangan masukan dengan respons kecepatan rotasi motor. Proses pengujian dilakukan dengan mengaplikasikan nilai PWM pada kisaran 0 hingga 255.

Peralatan yang digunakan yaitu:

1. Driver motor BTS7960.
2. Baterai Lipo 12 volt 3 cell.
3. Motor DC PG45.
4. Voltmeter digital.
5. Takometer.
6. *Software* Arduino IDE.
7. Kabel USB *type A to type B*.
8. Program arduino.
9. Laptop.

Setelah peralatan disiapkan, maka selanjutnya melakukan langkah-langkah pengambilan data yang terdiri dari:

1. Menghubungkan driver motor BTS7960 ke sumber tegangan (baterai lipo 12 volt 3 cell).
2. Menghubungkan pin digital arduino nomor 11 ke pin R_PWM driver motor BTS7960.

3. Menghubungkan pin digital arduino nomor 10 ke pin L_PWM driver motor BTS7960.
4. Menghubunngkan laptop ke arduino dengan k.abel USB *type A to type B*
5. Mengatur nilai PWM pada program arduino dengan nilai yang sudah ditetapkan.
6. Menjalankan program arduino.
7. Mengamati pergerakan motor.
8. Mengukur tegangan driver motor BTS760 pada kedua *output* yang masuk ke motor PG45.
9. Mengukur kecepatan putar as baling-baling kapal dengan takometer.
10. Mencatat hasil pengukuran yang tertampil pada alat ukur.
11. Mencatat nilai RPM yang muncul pada *serial monitor*.
12. Mengulangi langkah 5 sampai 11 dengan nilai PWM yang berbeda-beda dengan nilai PWM berkelipatan 10.

Program arduino yang digunakan adalah sebagai berikut

a. Program untuk menggerakkan Motor DC PG45

```
#define motor_blk1 11 // pin motor belakang 1
#define motor_blk2 10 // pin motor belakang 2

void setup() {
  pinMode(motor_blk1, OUTPUT);
  pinMode(motor_blk2, OUTPUT);
}

void motor2(int speed) {
  if (speed > 0) {
    analogWrite(motor_blk1, speed);
    digitalWrite(motor_blk2, LOW);
  } else if (speed < 0) {
    analogWrite(motor_blk1, 255 + speed);
    digitalWrite(motor_blk2, HIGH);
  }
}
```

```

    } else {
        digitalWrite(motor_blk1, LOW);
        digitalWrite(motor_blk2, LOW);
    }
}

void loop() {
    motor2(255);
}

```

b. Program untuk mengetahui RPM

```

const int motor1 = 11; //motor driver
const int motor2 = 10; //motor driver

const int pEchA = 2; //rotary encoder
const int pEchB = 3; //rotary encoder

volatile long countPulses = 0;
const int timeSampling = 50; //ms
unsigned long lastTime = 0;

void setup() {
    // put your setup code here, to run once:
    Serial.begin(9600);

    pinMode(pEchA, INPUT);
    pinMode(pEchB, INPUT);

    pinMode(motor1, OUTPUT);
    pinMode(motor2, OUTPUT);
}

```

```

    attachInterrupt(digitalPinToInterrupt(pEchA),
ISR_Encoder_A, RISING);
}

void loop() {
    unsigned long currentTime = millis();
    float deltaTime = currentTime - lastTime; //ms
    if (deltaTime >= timeSampling){
        float          freqSignal          =
(float)countPulses/(deltaTime/1.0e3);      //Hz;
(deltaTime/1.0e3) // convert ms to s
        countPulses = 0;
        float rpm = freqSignal * 60.0 / 3840; // 3840 =
p*N; p=step-up gear (19.2); N=pulses per revolution
(200);
        Input = rpm;
        Serial.println(rpm);
        lastTime = currentTime;
        motor(229.5);
    }
}

void ISR_Encoder_A(){
    if(digitalRead(pEchB)){
        //clockwise
        countPulses++;
    }else{
        //anti clockwise
        countPulses--;
    }
}

```



```

    }

    void motor(int speed) {
        if (speed > 0) {
            analogWrite(motor1, speed);
            digitalWrite(motor2, LOW);
        } else if (speed < 0) {
            analogWrite(motor1, 255 + speed);
            digitalWrite(motor2, HIGH);
        } else {
            digitalWrite(motor1, LOW);
            digitalWrite(motor2, LOW);
        }
    }
}

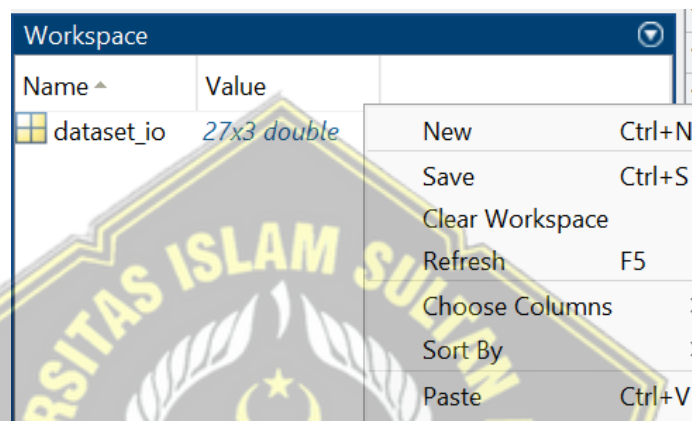
```

3.4 Membuat Model Matematis Sistem

Model matematis Motor DC yang digunakan belum terdefinisi. Mengingat model tersebut digunakan dalam proses penentuan parameter kontroler PID untuk sistem kendali kecepatan, maka perlu dilakukan identifikasi model matematis motor DC. Model matematis dapat dicari menggunakan *System Identification Tool* pada aplikasi Matlab. *System Identification Tool* pada Matlab adalah alat untuk membuat model matematika, dan didasarkan pada pengumpulan data *input* dan *output* dari sistem. Langkah-langkah dalam menggunakan *System Identification Tool* pada Matlab adalah sebagai berikut:

1. Membuka *software* Matlab.
2. Membuat *workspace* untuk menyimpan data *input* berupa tegangan dan *output* berupa kecepatan putar motor.
3. Memasukan data *input* dan *output* ke dalam *workspace*.
4. Menjalankan perintah *systemIdentification* pada *command window*.
5. Tekan *Import data*, lalu pilih *Time domain data*. Akan muncul sebuah *Dialog box*.

6. Pada isian Input masukan `data_motor(:,1)` (data motor merupakan nama dari file *workspace* yang telah dibuat sebelumnya dan `(:,1)` untuk memilih semua data di kolom nomor 1 pada *workspace*).
7. Pada isian Output masukan `data_motor(:,2)`. Lalu tekan *Import*.
8. Tekan *estimate* lalu pilih *Transfer function model*.
9. Melakukan langkah percobaan yang sama dengan preprosesing data *remove means*.
10. Melakukan validasi data.

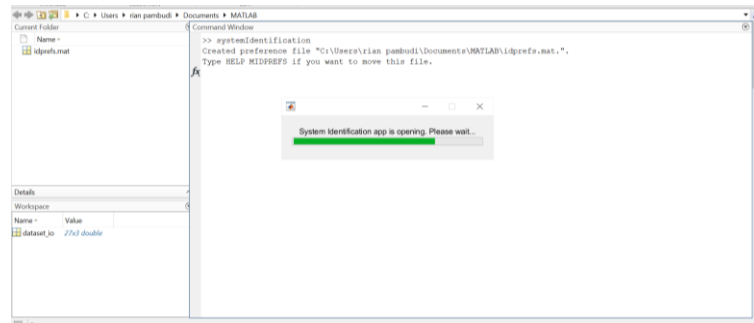
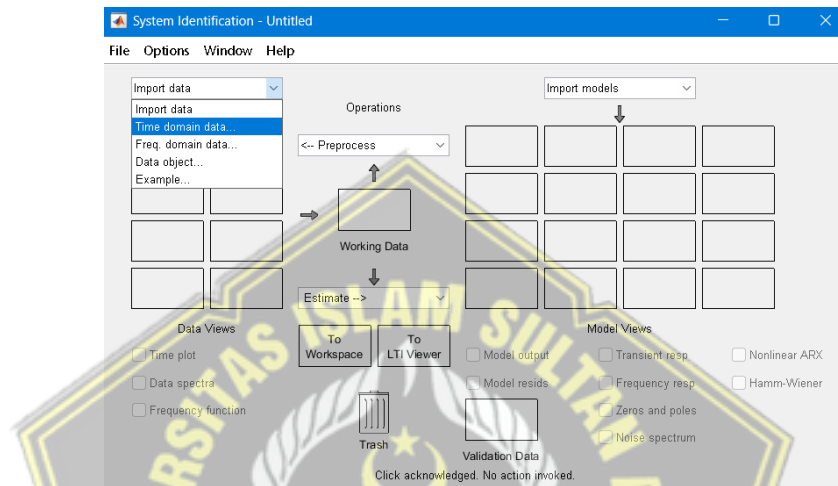
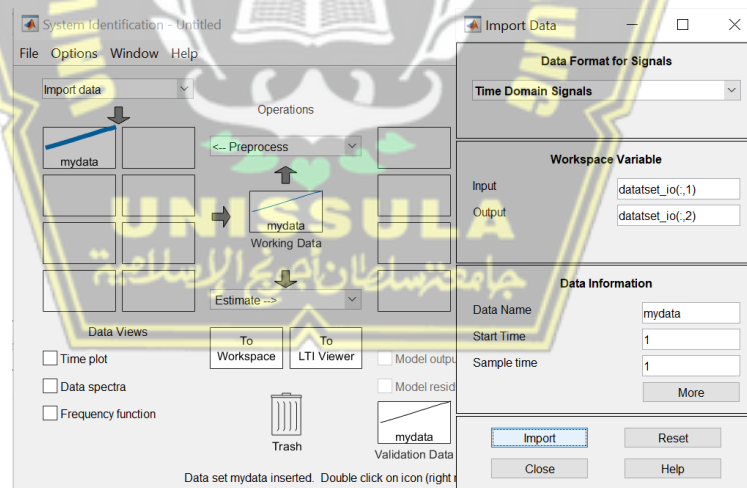


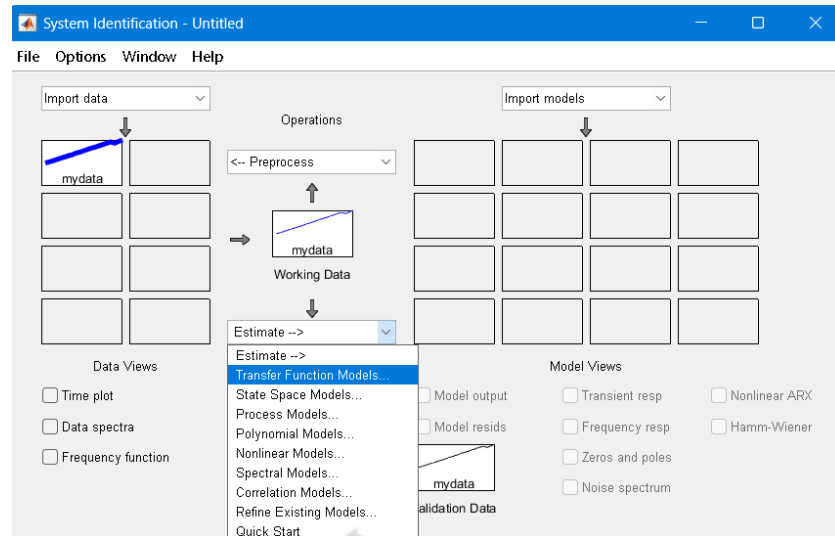
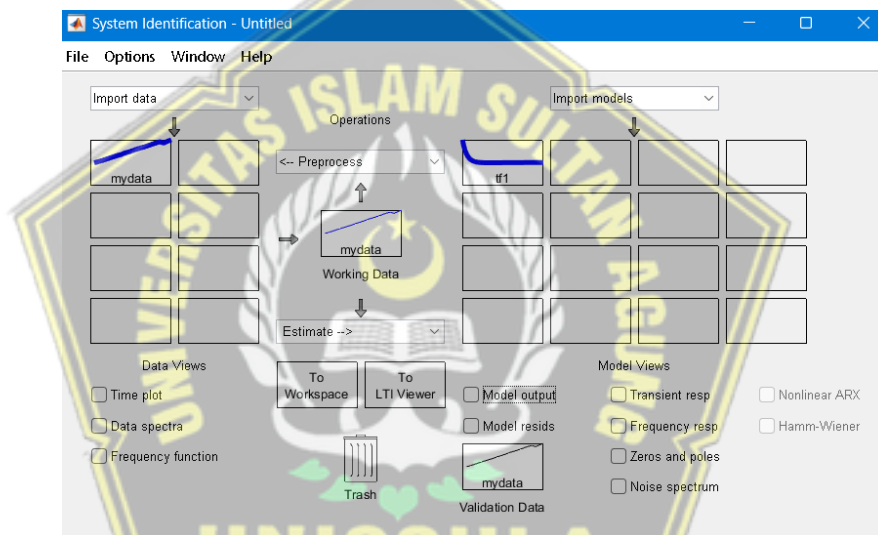
Gambar 3.5 Membuat *workspace* dengan nama `dataset_io`

The screenshot shows the MATLAB Variables window with the 'dataset_io' variable expanded into a table. The table has 27 rows and 13 columns. The first three columns contain numerical data, while the remaining columns are empty.

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	0	0										
2	0.8000	15.7000	15.6300										
3	1.6000	33	33.1300										
4	2.5000	51.8000	51.8800										
5	3.3000	70.4000	70.3100										
6	4.2000	89.1000	89.0100										
7	5	108	108.4400										
8	5.8000	127.1000	127.8800										
9	6.7000	146.1000	147.1900										
10	7.5000	166.3000	166.8800										
11	8.3000	186.2000	186.8800										
12	9.2000	206.6000	207.1900										
13	10	225.7000	226.2500										
14	11.7000	246.3000	246.8800										
15	12.5000	267.6000	267.8100										
16	13.4000	288.7000	288.7500										
17	14.2000	303.1000	301.8800										
18	15	324.6000	327.3800										
19	15.8000	345.9000	345.6300										
20	16.7000	367	367.5000										
21	17.5000	388.4000	386.8800										

Gambar 3.6 Memasukkan data *input* dan *output* ke dalam *workspace*

Gambar 3.7 Menjalankan perintah *systemIdentification*Gambar 3.8 Memilih *Time domain data*Gambar 3.9 *Import data dari workspace*

Gambar 3.10 Memilih *Transfer function model*

Gambar 3.11 tampilan saat model sudah jadi

3.5 Penalaan(tuning) PID

PID (*Proportional-Integral-Derivative*) perlu di-tuning untuk memastikan bahwa sistem kontrol berfungsi dengan baik dan responsif terhadap perubahan[13]. Tuning PID bertujuan untuk mengoptimalkan tiga komponen utama dari kontroler PID, yaitu:

1. Proportional (P):

Mengatur seberapa besar respons sistem terhadap kesalahan saat ini. Jika nilai P terlalu tinggi, sistem bisa menjadi tidak stabil, sedangkan jika terlalu rendah, responsnya akan lambat.

2. Integral (I):

Mengatasi kesalahan yang terakumulasi dari waktu ke waktu. Jika nilai I terlalu tinggi, sistem bisa mengalami *overshoot* dan osilasi, sedangkan jika terlalu rendah, kesalahan *steady-state* mungkin tidak teratasi.

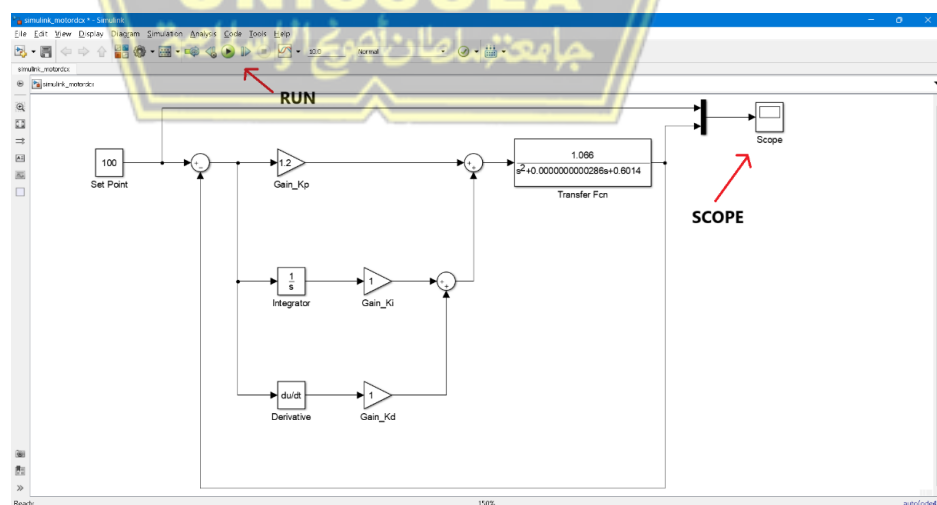
3. Derivative (D):

Mengantisipasi perubahan kesalahan di masa depan berdasarkan laju perubahan kesalahan saat ini. Nilai D yang terlalu tinggi dapat menyebabkan sistem menjadi sensitif terhadap *noise*, sedangkan nilai yang terlalu rendah dapat mengurangi responsifitas.

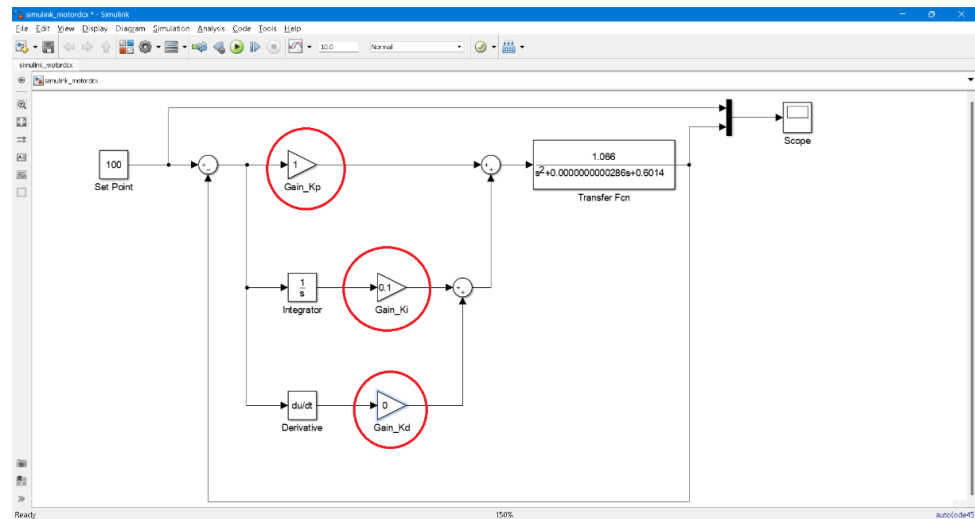
Pada penelitian ini penulis melakukan penalaan PID dengan metode *trial and error*. Metode *trial and error* dilakukan dengan cara mencoba-coba masukan Kp, Ki, dan Kd sehingga mendapatkan respon yang diharapkan.

Berikut adalah langkah-langkah tuning PID dengan metode *trial and error*:

1. Membuat diagram blok sistem di simulink Matlab
2. Memasukkan nilai *transfer function* yang telah didapatkan ke dalam blok *transfer function*.
3. Masukan nilai Kp, Ki, dan Kd, lalu tekan *Run*.
4. Menekan ikon *scope* untuk melihat respon sistem.
5. Mengamati hasil respon sistem.
6. Mengubah-ubah nilai Kp, Ki, dan Kd sampai menghasilkan hasil yang diharapkan.



Gambar 3.14 diagram blok sistem pada simulink



Gambar 3.15 nilai Kp, Ki, dan Kd



BAB IV

DATA DAN ANALISA

4.1 Hasil Pengujian

4.1.1 Pengujian Motor DC PG45

Pengujian motor DC PG45 dilakukan untuk pengumpulan data *output* yang akan digunakan untuk identifikasi sistem. Pengujian dilakukan dengan dua cara sehingga dapat diketahui *error* relatifnya. Error relatif didapatkan dengan persamaan 2.16. Berikut adalah contoh perhitungan *error relative* dengan kecepatan putar pada alat ukur takometer 246.3 RPM dan kecepatan putar pada serial monitor 246.88 RPM.

$$Error\ relative(\%) = \left| \frac{246.3 - 246.88}{246.3} \right| \times 100 = 0.236$$

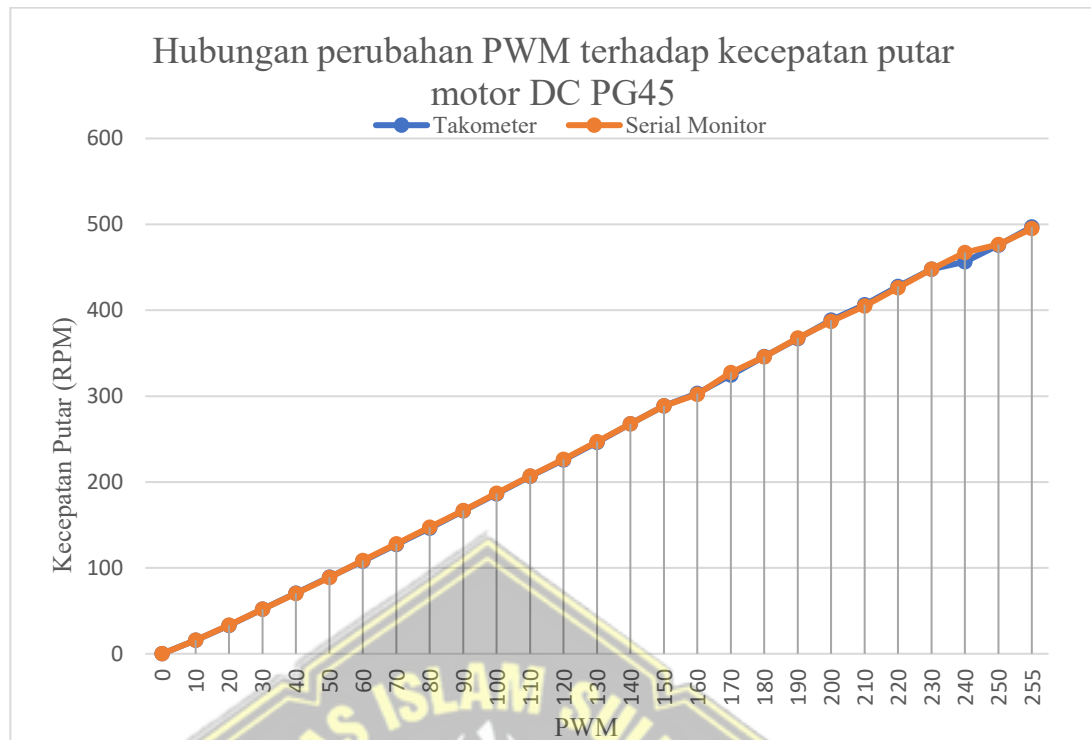
Berdasarkan serangkaian prosedur pengujian yang telah dilaksanakan, data kecepatan motor DC terangkum dalam Tabel 4.1.

Tabel 4.1 Hasil pengujian motor DC PG45

PWM	Tegangan masukan (V)	Kecepatan putar dari takometer (RPM)	Kecepatan putar dari serial monitor (RPM)	<i>Error</i> relatif (%)
0	0	0	0	0
10	0.8	15.7	15.63	0.446
20	1.6	33.3	33.13	0.394
30	2.5	51.8	51.88	0.154
40	3.3	70.4	70.31	0.128
50	4.2	89.1	89.01	0.101
60	5	108	108.44	0.407
70	5.8	127.1	127.88	0.613
80	6.7	146.1	147.19	0.748
90	7.5	166.3	166.88	0.349
100	8.3	186.2	186.88	0.364
110	9.2	206.6	207.19	0.285

120	10	225.7	226.25	0.244
130	11.7	246.3	246.88	0.236
140	12.5	267.6	267.81	0.078
150	13.4	288.7	288.75	0.017
160	14.2	303.1	301.88	0.403
170	15	324.6	327.38	0.857
180	15.8	345.9	345.63	0.078
190	16.7	367	367.50	0.136
200	17.5	388.4	386.88	0.392
210	18.3	406.6	404.76	0.452
220	19.2	427.6	426	0.374
230	20	447.7	447.88	0.040
240	20.8	456.2	467.19	2.409
250	21.7	475.9	476.56	0.138
255	22.4	497	495	0.402
Rata-rata nilai <i>error</i> (%)				0.38

Evaluasi karakteristik motor DC PG45 dilaksanakan melalui analisis respons kecepatan terhadap variasi tegangan *input*. Proses pengujian ini bertujuan mengidentifikasi hubungan antara perubahan tegangan masukan dengan dinamika kecepatan rotasi motor DC.



Gambar 4.1 Grafik hubungan perubahan PWM terhadap kecepatan putar motor DC PG45

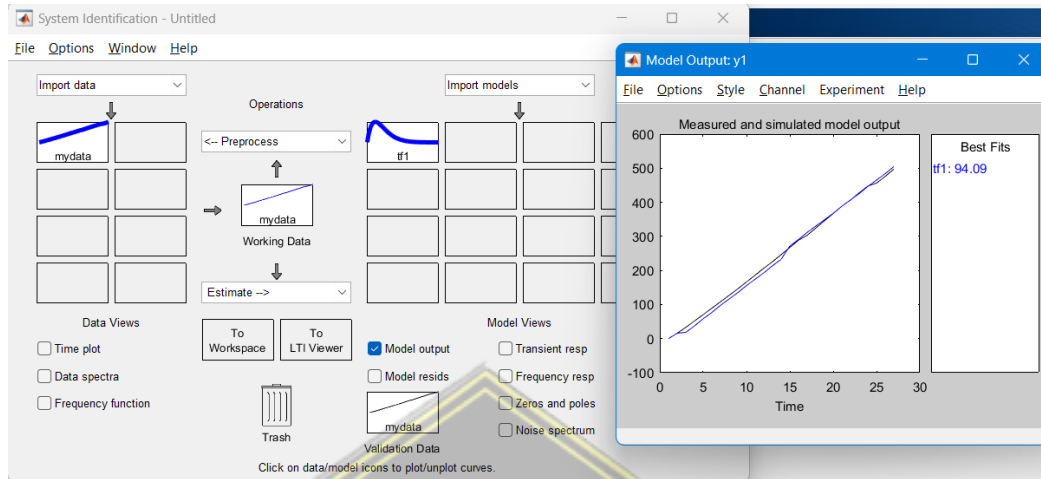
Berdasarkan data pengujian tabel 4.1, terdapat hubungan linear antara nilai PWM dan kecepatan putar motor (RPM). Saat PWM meningkat, RPM juga meningkat, tabel 4.1 perubahan PWM terhadap kecepatan putar motor menunjukkan bahwa kontrol PWM efektif dalam mengatur kecepatan motor.

Perbandingan antara kecepatan putar yang diukur dengan takometer dan yang diukur melalui serial monitor menunjukkan bahwa kedua metode pengukuran memberikan hasil yang sangat mirip. *Error* relatif rata-rata sebesar 0.38% menunjukkan bahwa sistem pengukuran berbasis arduino cukup akurat. *Error* relatif bervariasi dari 0.017% hingga 2.409%. Nilai *error* yang lebih tinggi pada beberapa titik misalnya pada PWM 240 mungkin disebabkan oleh faktor eksternal atau ketidakstabilan dalam pengukuran. Sebagian besar *error* relatif berada di bawah 1%, menunjukkan konsistensi yang baik dalam pengukuran.

Sistem pengukuran menunjukkan konsistensi yang baik di seluruh rentang pengujian, dengan sebagian besar *error* relatif yang rendah. Ini menunjukkan bahwa sistem berbasis arduino dapat diandalkan untuk aplikasi yang memerlukan pengukuran RPM yang akurat.

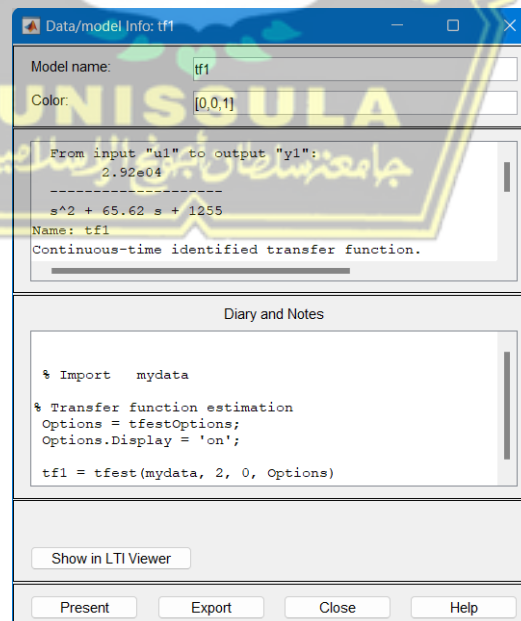
4.1.2 Pengujian Model Matematis

Setelah melakukan prosedur pengujian, didapatkan model *transfer function* seperti yang ditunjukkan dalam Gambar 4.1.



Gambar 4.2 Validasi data model dengan data awal

Pada tahap ini bertujuan untuk mengetahui *transfer function* dari sistem dari motor DC PG45. Garis biru merupakan *transfer function* pertama, garis hijau merupakan *transfer function* kedua, dan garis hitam merupakan data awal. Dari jendela *model output* yang memiliki nilai *best fits* tertinggi adalah model pertama yaitu sebesar 94.09%. Maka dari itu akan digunakan model *transfer function* dari tf1.



Gambar 4.3 hasil model yang telah dibuat

4.1.3 Pengujian Respon *Tuning* PID

Setelah melakukan langkah-langkah percobaan, didapatkan hasil percobaan sebagai berikut:

Tabel 4.2 Hasil percobaan tuning *trial and error*

Kp	Ki	Kd	Rise time (s)	Settling time (s)	Overshoot (%)
1	0	0	1.1107	NaN	53.007
1	0	1	NaN	NaN	NaN
10	0	1	0.3605	NaN	50.6155
10	1	1	0.3551	9.6262	54.6773
10	5	1	0.3357	NaN	70.0437
10	5	5	0.6288	5.0123	13.7091
10	5	10	1.1647	6.9236	18.7689
20	5	10	0.7829	7.2569	6.7491
50	5	10	0.3133	0.4872	1.3698
70	5	10	0.3033	0.6179	1.000
70	5	15	0.4550	0.6531	1.000
100	5	15	0.3183	0.4104	1.000

Tabel 4.2 menunjukkan hasil dari beberapa kombinasi parameter kontrol PID yaitu Kp (*proportional gain*), Ki (*integral gain*), dan Kd (*derivative gain*) dan respon sistem berupa *rise time* atau waktu sistem mulai merespon hingga mencapai *setpoint*, *settling time* atau waktu yang dibutuhkan sistem untuk stabil, dan *overshoot* atau seberapa banyak sistem melampaui nilai *setpoint* (dalam persentase).

Berdasarkan data tabel 4.2 didapatkan bahwa nilai *rise time* berada dalam rentang 0.3033 detik hingga 1.1647 detik. Rata-rata waktu naik yang tercatat adalah sekitar 0.538 detik, yang menunjukkan bahwa sebagian besar konfigurasi menghasilkan respon awal sistem yang relatif cepat. Waktu yang dibutuhkan sistem untuk mencapai keadaan stabil berkisar antara 0.4104 detik hingga 9.6262 detik. Rata-rata berada di sekitar 4.34 detik, namun data ini cukup bervariasi karena adanya beberapa konfigurasi yang menghasilkan waktu stabil yang jauh lebih lama. Sedangkan nilai *overshoot* menunjukkan variasi yang cukup besar, yaitu mulai

dari 1.000% hingga 70.0437%. Nilai rata-ratanya adalah sekitar 23.36%, yang berarti bahwa sebagian besar konfigurasi masih mengalami sedikit lonjakan di atas nilai *setpoint*, meskipun beberapa kombinasi parameter berhasil menekannya hingga hampir nol.

Berdasarkan hasil tuning kontrol PID yang dilakukan secara *trial and error* menggunakan Simulink, dapat dilihat adanya pengaruh yang cukup signifikan dari masing-masing parameter (K_p , K_i , dan K_d) terhadap performa sistem kendali, terutama terhadap *rise time*, *settling time*, dan *overshoot*. Nilai K_p memiliki dampak yang paling langsung terhadap kecepatan respon awal sistem. Ketika nilai K_p ditingkatkan dari 1 hingga 100, waktu naik (*rise time*) cenderung semakin pendek, menandakan bahwa sistem menjadi lebih responsif. Namun, peningkatan K_p secara berlebihan tanpa penyesuaian pada parameter lainnya dapat menyebabkan *overshoot* yang besar. Hal ini terlihat pada beberapa konfigurasi di mana nilai *overshoot* masih tinggi meskipun *rise time* cepat. Kemudian penambahan nilai K_i bertujuan untuk menghilangkan *error* jangka panjang (*steady-state error*). Dalam pengujian ini, penambahan K_i umumnya tidak memengaruhi *rise time* secara drastis, tetapi berperan dalam mengurangi nilai *overshoot* jika disesuaikan dengan tepat. Misalnya, konfigurasi dengan $K_i = 5$ menunjukkan performa *overshoot* yang lebih baik dibandingkan saat hanya K_p dan K_d yang disesuaikan, selama nilainya tidak terlalu besar. Penggunaan K_i yang berlebihan dapat memperpanjang waktu stabil sistem, sehingga perlu dikombinasikan dengan hati-hati. Kemudian nilai K_d berfungsi sebagai peredam (*damping*) terhadap perubahan sinyal yang cepat. Dalam eksperimen ini, peningkatan nilai K_d terbukti efektif dalam menurunkan *overshoot* dan meningkatkan stabilitas sistem. Hal ini terlihat dari beberapa konfigurasi di mana *overshoot* dapat ditekan hingga 1% atau bahkan kurang, dengan *rise time* dan *settling time* yang sangat singkat.

Berdasarkan hasil pengujian berbagai kombinasi parameter kontrol PID, diperoleh konfigurasi yang menunjukkan performa sistem paling optimal. Kriteria optimal dalam hal ini mencakup waktu naik (*rise time*) yang singkat, waktu stabil (*settling time*) yang cepat, serta nilai *overshoot* yang sangat kecil atau mendekati

nol. Parameter-parameter tersebut merupakan indikator penting dalam menilai kualitas respons sistem kendali.

Hasil pengujian menunjukkan bahwa konfigurasi parameter dengan nilai *proportional gain* (K_p) yang tinggi, yaitu antara 70 hingga 100, *integral gain* (K_i) pada rentang 5 hingga 15, dan *derivative gain* (K_d) sebesar 10 hingga 15, memberikan hasil yang paling baik. Kombinasi tersebut mampu menghasilkan sistem yang responsif, cepat mencapai kestabilan, dan memiliki tingkat keakuratan yang tinggi. Sebagai contoh, pada konfigurasi $K_p = 70$, $K_i = 5$, dan $K_d = 10$, sistem menunjukkan waktu naik sebesar 0,3033 detik, waktu stabil sebesar 0,6179 detik, dan nilai *overshoot* hanya sebesar 1%. Hasil serupa juga ditunjukkan pada konfigurasi $K_p = 100$, $K_i = 5$, dan $K_d = 15$, dengan waktu naik sebesar 0,3183 detik, waktu stabil sebesar 0,4104 detik, dan *overshoot* yang tetap berada pada angka 1%.



BAB V

PENUTUP

5.1 Kesimpulan

Dari tugas akhir ini, dapat diambil beberapa kesimpulan sebagai berikut:

1. Perubahan PWM terhadap kecepatan putar motor menunjukkan bahwa kontrol PWM efektif dalam mengatur kecepatan motor. Dengan kontrol PWM tersebut mampu menghasilkan fungsi transfer $\frac{2.92e04}{s^2+62.65s+1551}$ yang memiliki kemiripan dengan sistem sebesar 94.09%.
2. Menggunakan *tuning* PID *trial and error* didapatkan nilai parameter $K_p=100$; $K_i=5$; $K_d=15$ mampu menghasilkan kontrol yang stabil pada model matematis kapal pendeteksi sedimen milik FTI Unissula.

5.2 Saran

Adapun saran dari penulis yaitu:

1. Melakukan pengambilan data kecepatan sudut pada motor DC PG45 secara berulang sampai mendapatkan hasil yang linear.
2. Memilih komponen yang memiliki data spesifikasi yang jelas.
3. Semakin banyak dan bervariasinya jenis metode penalaan. Maka perlu diujikan menggunakan jenis metode kontrol yang lainnya untuk diterapkan pada sistem kontrol kecepatan motor PG45.
4. Teknologi sistem kontrol kecepatan motor DC PG45 perlu dikembangkan seiring perkembangan teknologi *wireless* atau tanpa kabel.

DAFTAR PUSTAKA

- [1] C. Asdak, *Hidrologi dan Pengelolaan Daerah Aliran Sungai*. Ugm Press, 2023.
- [2] M. K. Faza, S. A. D. Prasetyowati, and B. Arifin, "Rancang Bangun Kapal Pengukur Volume Sedimen dengan Algoritma PID," *Aviation Electronics, Information Technology, Telecommunications, Electricals, and Controls (AVITEC)*, vol. 5, no. 2, pp. 111–125, 2023.
- [3] S. A. D. Prasetyowati, B. Arifin, A. Syakhroni, and M. K. Faza, "Monitoring River Sediment by Optimizing Arduino Capabilities Controlled by the PID Algorithm," *WSEAS Transactions on Systems*, vol. 21, pp. 233–240, 2022.
- [4] J. Ardalli, "Prototipe Alat Pengukur Volume Sedimen Di Dasar Sungai Berbasis Arduino Mega 2560," Universitas Islam Sultan Agung, Semarang, 2019.
- [5] M. N. Masrukhan, M. P. Mulyo, D. Ajiatmo, and M. Ali, "Optimasi Kecepatan Motor DC Menggunakan Pid Dengan Tuning Ant Colony Optimization (ACO) Controller," *SENTIA-2016, Polinema, Malang*, pp. B49–B52, 2016.
- [6] L. A. Suyitno, D. Aming, and S. W. Jadmiko, "Pengendalian Simulator Orde Dua Dengan Metode Auto Tuning Pid Berbasis Kontroler TK4S-B4CR," in *Prosiding Industrial Research Workshop and National Seminar*, 2021, pp. 81–84.
- [7] B. Arifin, A. A. Nugroho, E. N. Budisusila, and M. Khosyi'in, "System Identification and Control Strategy on Electric Power Steering DC Motor," *Journal of Robotics and Control (JRC)*, vol. 5, no. 3, pp. 655–666, 2024.
- [8] D. M. Ionel, O. P. Malik, V. Rallabandi, and N. Taran, "Electric Machines: Motors and Drives," in *Standard Handbook for Electrical Engineers*, 17th ed., S. Suryanto and H. W. Beaty, Eds., New York: McGraw-Hill Education, 2017, pp. 919–960.

- [9] MK FAZA, “PROTOTIPE SISTEM PENGONTROL KAPAL DAN PENDETEKSI SEDIMEN,” Thesis, Universitas Islam Sultan Agung, Semarang, 2023.
- [10] M. T. Afif and I. A. P. Pratiwi, “Analisis Perbandingan Baterai Lithium-Ion, Lithium-Polymer, Lead Acid Dan Nickel-Metal Hydride Pada Penggunaan Mobil Listrik - Review,” *Jurnal Rekayasa Mesin Universitas Brawijaya*, vol. 6, no. 2, 2015, doi: 10.21776/ub.jrm.2015.006.02.1.
- [11] S. A. D. Prasetyowati, B. Arifin, A. Syakhroni, and M. K. Faza, “Monitoring River Sediment by Optimizing Arduino Capabilities Controlled by the PID Algorithm,” *WSEAS Transactions on Systems*, vol. 21, pp. 233–240, 2022.
- [12] Y. D. Widiarto, M. E. I. Najoan, and M. D. Putro, “Sistem penggerak robot beroda vacuum cleaner berbasis mini computer Raspberry pi,” *jurnal teknik elektro dan komputer*, vol. 7, no. 1, pp. 25–32, 2018.
- [13] K. Ogata, *Modern Control Engineering*. New York: Prentice Hal, 2020.
- [14] K. J. Åström and T. Hägglund, *PID Controllers: Theory, Design, and Tuning*. Research Triangle Park, North Carolina: ISA - The Instrumentation, Systems and Automation Society, 1995.
- [15] F. F. Fahmizal and R. Susanto, “Identifikasi sistem motor dc dan kendali linear quadratic regulator berbasis arduino-simulink matlab,” *LQR*, vol. 12, p. 13, 2018.
- [16] www.mathworks.com, “Transfer Function Models.” Accessed: Feb. 01, 2025. [Online]. Available: https://www.mathworks.com/help/ident/transfer-function-models.html?s_tid=CRUX_lftnav
- [17] M. A. ULUM and S. I. Haryudo, “Perancangan Sistem Monitoring Kecepatan Putar Motor DC Berbasis Internet Of Things Menggunakan Aplikasi BLYNK,” *Jurnal Teknik Elektro*, vol. 9, no. 1, 2020.