TUGAS AKHIR (EE43430)

RANCANG BANGUN MOTOR *BRUSHER ROV HULL CLEANING* MENGGUNAKAN METODE PID

Moses Yudha Dua Lembang
NRP. 0422040045

Dosen Pembimbing
1. Dr. YUNING WIDIARTI, S.T., M.T.
2. ZINDHU MAULANA AHMAD PUTRA, S.ST. M.Tr.T.

PROGRAM STUDI D4 TEKNIK KELISTRIKAN KAPAL
JURUSAN TEKNIK KELISTRIKAN KAPAL
POLITEKNIK PERKAPALAN NEGERI SURABAYA
SURABAYA
2026

(Halaman ini sengaja dikosongkan)

Tugas Akhhir

RANCANG BANGUN MOTOR *BRUSHER ROV HULL CLEANING* MENGGUNAKAN METODE PID

Nama : Moses Yudha Dua Lembang
NRP : 0422040045

Dosen Pembimbing
1. Dr. YUNING WIDIARTI, S.T., M.T.
2. ZINDHU MAULANA AHMAD PUTRA, S.ST. M.Tr.T.

PROGRAM STUDI D4 TEKNIK KELISTRIKAN KAPAL
JURUSAN TEKNIK KELISTRIKAN KAPAL
POLITEKNIK PERKAPALAN NEGERI SURABAYA
SURABAYA
2026

(Halaman ini sengaja dikosongkan)

LEMBAR PENGESAHAN

TUGAS AKHIR

RANCANG BANGUN MOTOR *BRUSHER ROV HULL CLEANING* MENGGUNAKAN METODE PID

Disusun Oleh:
Moses Yudha Dua Lembang
0422040045

Diajukan Untuk Memenuhi Salah Satu Syarat Kelulusan
Program Studi D4 Teknik Kelistrikan Kapal
Jurusan Teknik Kelistrikan Kapal
POLITEKNIK PERKAPALAN NEGERI SURABAYA

Disetujui oleh Tim Penguji Tugas Akhir Tanggal Ujian : 03 Agustus 2026
Periode Wisuda : Oktober 2026

Menyetujui,

Dosen Penguji	NIDN/NUPTK	Tanda Tangan
1. Ii Munadhif, S.ST., M.T.	(0010079102)	(.....)
2. Dr. Yuning Widiarti, S.T., M.T.	(0016058004)	(.....)
3. Aulia Rahma Annisa, S.ST., M.T.	(0303069102)	(.....)
4. Sholahuddin Muhammad Irsyad, S.Tr.T.,M.Tr.T.	(0337772673130343)	(.....)

Dosen Pembimbing	NIDN	Tanda Tangan
1. Dr. Yuning Widiarti, S.T., M.T.	(0016058004)	(.....)
2. Zindhu Maulana Ahmad Putra, S.ST., M.Tr.T.	(0023099403)	(.....)

Menyetujui,
Ketua Jurusan

Isa Rachman, S.T., M.T.
NIP. 198008162008121001

Mengetahui,
Koordinator Program Studi

Dr. Yuning Widiarti, S.T., M.T.
NIP. 198005162006042001

Dipindai dengan CamScanner

PPNS	**PERNYATAAN BEBAS PLAGIAT**	No. : F.WD I. 021 Date : 3 Nopember 2015 Rev. : 01 Page : 1 dari 1

Yang bertandatangan dibawah ini :

Nama : Moses Yudha Dua Lembang

NRP. : 0422040045

Jurusan/Prodi : Teknik Kelistrikan Kapal/D4-Teknik Kelistrikan Kapal

Dengan ini menyatakan dengan sesungguhnya bahwa :

Tugas Akhir yang akan saya kerjakan dengan judul :

Rancang Bangun Motor *Brusher ROV Hull Cleaning* Menggunakan Metode PID

Adalah **benar karya saya sendiri** dan **bukan plagiat dari karya orang lain.**

Apabila dikemudian hari terbukti terdapat plagiat dalam karya ilmiah tersebut, maka saya bersedia menerima **sanksi** sesuai ketentuan peraturan yang berlaku.

Demikian surat pernyataan ini saya buat dengan penuh tanggung jawab.

Surabaya, 2 September 2026

Yang membuat pernyataan,

(Moses Yudha Dua Lembang)

NRP. 0422040045

Dipindai dengan CamScanner

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah SWT atas segala rahmat, karunia, pertolongan dan kasih sayang-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul **"Rancang Bangun Motor *Brusher ROV Hull Cleaning* Menggunakan Metode PID".** Tugas akhir ini disusun sebagai salah satu syarat untuk menyelesaikan pendidikan dan memperoleh gelar Sarjana Terapan Teknik pada Program Studi D4 Teknik Kelistrikan Kapal, Jurusan Teknik Kelistrikan Kapal, Politeknik Perkapalan Negeri
Surabaya.

Dengan penuh rasa hormat dan ketulusan penulis menyampaikan terima kasih kepada:

1. Tuhan Yesus Kristus yang senantiasa menemani penulis dalam setiap keadaan, baik dalam sukacita maupun dalam lembah kesesakan.
2. Bapak Antonius Herianto dan Ibu Nungki Widowati yang selalu menjadi sumber kekuatan penulis untuk tetap melangkah. Terimakasih telah menjadi orang tua penulis dengan segala doa, kasih sayang, pengorbanan, perjuangan, dukungan moral dan materi.
3. Yizreel, Eliezer, dan Yemima atas perhatian, inspirasi, dan doa yang selalu diberikan menjadikan selama penulis menempuh pendidikan hingga menyelesaikan Tugas Akhir.
4. Bapak Rachmad Tri Soelistijono, S.T., M.T. selaku Direktur Politeknik Perkapalan Negeri Surabaya.
5. Bapak Isa Rachman, S.T., M.T. selaku Ketua Jurusan Teknik Kelistrikan Kapal, Politeknik Perkapalan Negeri Surabaya.
6. Ibu Dr. Yuning Widiarti, S.T., M.T. selaku Koordinator Program Studi D4 Teknik Kelistrikan Kapal, Politeknik Perkapalan Negeri Surabaya sekaligus dosen pembimbing 1 yang telah meluangkan waktu, tenaga dan pikiran untuk memberikan bimbingan, arahan, masukan serta solusi selama proses perancangan, pengujian alat dan penyusuanan laporan Tugas Akhir.
7. Bapak Zindhu Maulana Ahmad Putra, S.ST. M.Tr.T. selaku Dosen Pembimbing 2 yang telah meluangkan waktu, tenaga dan pikiran untuk

memberikan bimbingan, arahan, masukan serta solusi selama proses perancangan, pengujian alat dan penyusuanan laporan Tugas Akhir.

8. Seluruh teman-teman Teknik Kelistrikan Kapal Angkatan 2022, Setrum Pride, Kemenlugri Kabinet Pijar Asa atas rasa kebersamaan selama masa perkuliahan sampai dengan Tugas Akhir selesai.
9. Rangga Irdyansyah Putra, Arya Eka Putra M, Farhan Pradana S, M. Prasetiya Ramadhan Putra, Panji Wahyu S, Petrus Chandra Imanuel dan Iusibius Versely Hermadi selaku sahabat penulis yang telah memberikan semangat dan rasa pantang menyerah.
10. Ricky Maulana Antrasena yang menginspirasi penulis dalam segala aspek pendidikan.
11. Seluruh pihak yang tidak dapat disebutkan satu per satu yang telah memberikan bantuan, dukungan dan doa penulis baik secara langsung maupun tidak langsung.
12. Terakhir, penulis menyampaikan apresiasi kepada diri sendiri yang telah memilih untuk terus berusaha dan bertahan ketika proses yang dijalani terasa sulit. Semoga seluruh proses ini menjadi pengingat bahwa setiap kesulitan dapat dilalui dengan usaha, doa, kesabaran dan keyakinan.

Penulis menyadari bahwa Tugas Akhir ini masih jauh dari kesempurnaan serta memiliki keterbatasan, baik dalam proses perancangan, pelaksanaan pengujian, pengolahan data maupun penyusunan laporan. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun sebagai bahan evaluasi dan pengembangan untuk penelitian selanjutnya. Akhir kata, penulis berharap Tugas Akhir dapat memberikan manfaat bagi para pembaca.

Surabaya, 10 September 2026

Penulis,

RANCANG BANGUN MOTOR *BRUSHER ROV HULL CLEANING* MENGGUNAKAN METODE PID

Moses Yudha Dua Lembang

ABSTRAK

Pertumbuhan *biofouling* pada lambung kapal meningkatkan hambatan hidrodinamis, konsumsi bahan bakar, dan emisi gas buang. Pembersihan manual menggunakan penyelam memiliki risiko keselamatan dan biaya operasional tinggi. Penelitian ini bertujuan merancang sistem kontrol sikat pada *Remotely Operated Vehicle* (ROV) *Hull Cleaning* berbasis YOLO dan sensor arus. Sistem menggunakan arsitektur hibrida PC stasiun permukaan dan ESP32. YOLO mendeteksi tingkat *biofouling* yang kemudian divalidasi menggunakan *counter voting* 5-frame untuk menjaga kestabilan *setpoint*. Data *gyroscope* Pixhawk melalui MAVLink digunakan sebagai *safety interlock* agar sikat aktif ketika ROV berada pada posisi stabil terhadap lambung. ESP32 mengendalikan kecepatan sikat menggunakan PI dengan Kp = 0,01 dan Ki = 0,005 berdasarkan umpan balik sensor Hall. *Planetary gearbox* rasio 5:1 meningkatkan torsi sikat hingga 4,25 kali. Sensor arus ACS758 menerapkan *Auto-Switch* untuk menurunkan *setpoint* dari *Heavy* 200 RPM, *Medium* 120 RPM, *Light* 80 RPM, hingga *Auto Stop* setelah beban motor menurun. Hasil pengujian menunjukkan waktu inferensi YOLO rata-rata 99 ms per frame. Pengendali PI mampu memulihkan putaran dalam 2,0 detik pada *Medium* dan 4,0 detik pada *Heavy*, dengan arus puncak 4,79 A dan 8,65 A. Validasi simulator menghasilkan MAE arus sebesar 0,575 A, 1,277 A, dan 3,221 A untuk *Light*, *Medium*, dan *Heavy*. Secara keseluruhan, sistem mampu melakukan pembersihan *biofouling* secara adaptif serta menjaga kestabilan kontak ROV dengan lambung kapal.

Kata Kunci: ROV, Inspeksi, Lambung Kapal, Penyikat, *Biofouling*, YOLO, PID

RANCANG BANGUN MOTOR *BRUSHER ROV HULL CLEANING* MENGGUNAKAN METODE PID

Moses Yudha Dua Lembang

ABSTRACT

Biofouling growth on ship hulls increases hydrodynamic resistance, fuel consumption, and exhaust emissions. Manual cleaning using divers involves high safety risks and operational costs. This study aims to design a brush control system for a Remotely Operated Vehicle (ROV) Hull Cleaning system based on YOLO and current sensing. The system uses a hybrid architecture consisting of a surface-station PC and an ESP32 microcontroller. YOLO detects the biofouling level, which is then validated using a 5-frame counter voting method to maintain setpoint stability. Pixhawk gyroscope data transmitted via MAVLink is used as a safety interlock to activate the brush only when the ROV is stably positioned against the hull. The ESP32 controls brush speed using a PI controller with Kp = 0.01 and Ki = 0.005 based on Hall sensor feedback. A 5:1 planetary gearbox increases brush torque by up to 4.25 times. The ACS758 current sensor applies an Auto-Switch mechanism to reduce the setpoint from Heavy 200 RPM, Medium 120 RPM, and Light 80 RPM to Auto Stop as the motor load decreases. Testing showed an average YOLO inference time of 99 ms per frame. The PI controller recovered brush speed within 2.0 seconds for Medium and 4.0 seconds for Heavy, with peak currents of 4.79 A and 8.65 A, respectively. Overall, the system demonstrated adaptive biofouling cleaning while maintaining stable ROV contact with the ship hull.

Keywords: *ROV, Inspection, Hull, Brush, Biofouling, YOLO, PID*

DAFTAR ISI

LEMBAR PENGESAHAN iii
KATA PENGANTAR vii
ABSTRAK… ix
DAFTAR ISI xiii
DAFTAR TABEL xvii
DAFTAR GAMBAR xix
DAFTAR NOTASI xxiii
BAB 1 PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Pertanyaan Penelitian 3
1.3 Batasan Penelitian 3
1.4 Tujuan Penelitian 4
1.5 Manfaat Penelitian 4
BAB 2 TINJAUAN PUSTAKA 7
2.1 Kajian Penelitian Terdahulu 7
2.2 Kajian Pustaka 12
2.2.1 Remotely Operated Vehicle (ROV) 12
2.2.2 Proposional-Integral-Derivative (PID) 13
2.2.3 You Only Look Once (YOLO) 15
2.2.4 PWM – Pulse Width Modulation 16
2.3 Kajian Pustaka Perangkat Keras dan Lunak 19
2.3.1 Kamera 19
2.3.2 Raspberry Pi 3B 20

2.3.3 ESP 32 22
2.3.4 Motor *Brushless* DC 23
2.3.5 ESC - *Electronic Speed Controller* 25
2.3.6 Brusher 27
2.3.7 Baterai 29
2.3.8 ACS758 30
2.3.9 Pixhawk 32
2.3.10 Google Colab 33
2.3.11 Visual Studio 35
2.3.12 QGround Control 35
2.3.13 ArduinoIDE 37
2.3.14 Matlab 38
BAB 3 METODE PENELITIAN 39
3.1 Konsep Penelitian 39
3.1.1 Diagram Blok Sistem 42
3.1.2 Flowchart Sistem 43
3.1.3 Perancangan Sistem Kontrol 45
3.1.4 Perancangan YOLO 47
3.1.5 Perancangan Skema Pembersihan 48
3.1.6 Perancangan Skema Pengujian 50
3.2 Tahapan Penelitian 52
3.2.1 Identifikasi Masalah dan Studi Literatur 53
3.2.2 Analisis Kebutuhan Sistem 53
3.2.3 Desain dan Perancangan Sistem 54
3.2.4 Pembuatan Alat 54
3.2.5 Pengujian Sistem 54

3.2.6 Analisis Data 55
3.2.7 Pembuatan Laporan Akhir 55
3.3 Rancangan Perangkat Keras/Lunak 55
3.3.1 Analisis Kebutuhan Sistem 55
3.3.2 Perancangan Desain Hardware 56
3.3.3 Penentuan Model Matematis 58
3.3.4 Perancangan You Only Look Once (YOLO) 60
3.3.5 Wiring Diagram Supply sistem 61
3.3.6 Rancangan Wiring PID Kontrol 62
3.3.7 Rancangan Wiring Clasification Object 63
3.3.8 Rancangan Wiring Keseluruhan Hardware 64
3.3.9 Perancangan Software 65
3.3.10 Perancangan Desain Mekanik 66
BAB 4 HASIL DAN PENELITIAN 71
4.1 Pengujian Sensor dan Aktuator 71
4.1.1 Pengujian Sensor Arus ACS-758 71
4.1.2 Pengujian Hall Sensor SS49E 73
4.1.3 Pengujian Pixhawk 74
4.1.4 Pengujian Baterai 76
4.1.5 Pengujian Pin ESP 32 76
4.1.6 Pengujian Motor Brush 77
4.1.7 Pengujian Electronic Speed Control 79
4.1.8 Pengujian Visualisasi Kamera 80
4.2 Perakitan *Hardware* 82
4.2.1 Instalasi Sistem Elektrik pada Model ROV 83
4.2.2 Instalasi *Gear Box* Motor *Brush* 83

4.2.4 Realisasi Layout Brushing Sistem 84
4.3 Pengujian Komunikasi Serial YOLO 85
4.4 Pengujian Kontrol motor menggunakan PID.....87
4.4.1 Penentuan Konstanta PID 88
4.4.2 Penentuan *Transfer Function* Motor.....90
4.4.3 Validasi Transfer Function.....92
4.4.4 Tuning *Gain Scheduling* PID Berdasarkan Tiga Profil Beban 94
4.5.4 Realisasi Metode Tanpa Beban.....100
4.5 Hasil dan Pembahasan 103
4.5.1 Hasil Implementasi *Manuver Holding* Vertikal dan Horizontal 103
4.5.2 Hasil Implementasi YOLO 110
4.5.3 Analisis Pengaruh YOLO Dalam Sistem.....114
4.5.3 Hasil Implementasi Metode PID.....118
4.5.4 Analisis *Response System* 124
4.5.5 Hasil dari Brushing System.....131
BAB 5 KESIMPULAN DAN SARAN 137
5.1 Kesimpulan.....137
5.2 Saran 138
DAFTAR PUSTAKA 141
LAMPIRAN.....143

BAB 1
PENDAHULUAN

1.1 Latar Belakang

Salah satu komponen penting dalam pemeliharaan kapal adalah pembersihan dan inspeksi lambung. Ini menjaga permukaan lambung bebas dari kotoran, kerak, dan makhluk laut, dan memastikan kondisi struktural kapal tetap ideal. Proses ini berdampak langsung pada masa pakai material, ketahanan terhadap korosi, efisiensi konsumsi bahan bakar karena hambatan air yang dikurangi, dan keselamatan penumpang dan kapal selama berlayar. Untuk mengurangi biaya perawatan dan risiko kecelakaan, pembersihan dan inspeksi rutin dapat membantu menemukan kerusakan potensial lebih awal (Wei dkk., 2024).

Pada permukaan lambung, pertumbuhan organisme laut dan kerak atau biofouling dapat meningkatkan kekasaran permukaan. Ini meningkatkan hambatan hidrodinamis dan meningkatkan konsumsi bahan bakar dan emisi gas buang. Pemberlakuan inspeksi dan pembersihan lambung kapal secara teratur sangat penting untuk mempertahankan kinerja kapal dan mengurangi kerusakan yang menimbulkan kerugian material maupun risiko keselamatan karena kotoran di lambung dapat mempercepat korosi dan menurunkan kinerja secara keseluruhan. Selain itu, inspeksi lambung kapal mencakup pemeriksaan kebersihan permukaan serta pengukuran ketebalan plat lambung untuk mengukur tingkat korosi dan keausan struktural. Untuk melakukan inspeksi ini, pembersihan lambung adalah tahap awal yang penting sebelum pengukuran ketebalan, agar hasil inspeksi dapat diperoleh secara akurat tanpa terpengaruh oleh kerak atau kotoran yang menempel (Feng dkk., 2023).

Di Indonesia, kewajiban pengedokan dan pemeriksaan kondisi lambung kapal diatur dalam ketentuan klasifikasi dan Peraturan Direktur Jenderal Perhubungan Laut No. HK.103/1/4/DJPL-14 tentang Pelaksanaan *Dock* (Pelimbungan) Kapal Berbendera Indonesia, yang menetapkan keharusan kapal untuk menjalani docking dan inspeksi struktur secara berkala (Direktorat Jenderal Perhubungan Laut, 2017). Regulasi tersebut juga memungkinkan penerapan skema

Under-Water Inspection In-Lieu of Dry-Docking (UWILD) sebagai alternatif pemeriksaan bawah air tanpa perlu membawa kapal ke dok kering. Namun, praktik inspeksi dan pembersihan lambung kapal di lapangan masih didominasi metode konvensional yang memiliki berbagai keterbatasan, seperti risiko keselamatan kerja yang tinggi, keterbatasan jangkauan area, ketergantungan pada kondisi cuaca, durasi pekerjaan yang panjang, serta biaya operasional yang besar. Selain itu, kualitas hasil inspeksi sering kali bergantung pada keterampilan operator, sehingga berpotensi menimbulkan ketidakkonsistenan dan kesalahan penilaian. Sementara itu, riset terkait pengembangan solusi yang dapat mempermudah dan memodernisasi proses inspeksi lambung kapal di Indonesia masih tergolong rendah. Oleh karena itu, diperlukan pendekatan inovatif seperti pemanfaatan *Remotely Operated Vehicle* (ROV) yang dilengkapi sensor *Ultrasonic Thickness* (UT) untuk memeriksa ketebalan pelat lambung secara efisien. Dalam penerapannya, proses pembersihan dari kotoran atau organisme laut menjadi tahap krusial, karena sensor UT hanya mampu membaca material berbahan dasar baja atau logam yang mengandung unsur Fe (Palanisamy dkk., 2024).

Dalam konteks pemeliharaan lambung kapal, ROV tidak hanya berperan sebagai alat inspeksi, tetapi juga dapat dirancang untuk melakukan pembersihan secara simultan. Integrasi sistem pembersih pada ROV memungkinkan kegiatan penghilangan kotoran yang menempel pada lambung dilakukan langsung sebelum proses inspeksi ketebalan pelat, sehingga hasil pengukuran menjadi lebih akurat dan efisien tanpa perlu melakukan *docking* kapal. Berdasarkan kebutuhan tersebut, penelitian ini berfokus pada perancangan ROV inovatif yang dilengkapi sistem pembersih bawaan dan sensor inspeksi untuk mendukung proses monitoring kondisi lambung kapal secara menyeluruh. Pendekatan ini diharapkan dapat menjadi solusi modern dalam mempermudah proses pemeliharaan struktur bawah air dengan waktu yang lebih singkat, biaya lebih rendah, dan tingkat keselamatan yang lebih tinggi dibandingkan metode konvensional.

Untuk mencapai tujuan tersebut, sistem pembersih pada ROV dirancang menggunakan mekanisme brush yang mampu menghilangkan kotoran yang menempel pada lambung tanpa merusak struktur permukaan lambung kapal. Implementasi permodelan sistem kontrol pada penelitian ini digunakan untuk

merancang dan menguji pengendali kecepatan motor brusher M200 agar putaran tetap stabil terhadap gangguan gaya sikat selama proses pembersihan lambung kapal. Informasi mengenai tingkat kotoran diperoleh melalui input visual dari kamera bawah air yang diolah menggunakan algoritma *You Only Look Once* (YOLO) guna mengklasifikasikan kondisi permukaan ke dalam beberapa kategori kebersihan. Hasil klasifikasi tersebut digunakan sebagai acuan penentuan setpoint kecepatan motor yang dikendalikan oleh algoritma Proportional–Integral–Derivative (PID), sehingga sistem dapat menjaga performa pembersihan secara efisien dan aman terhadap material lambung kapal.

Dengan pendekatan ini, diharapkan proses pembersihan dapat dilakukan secara adaptif sesuai tingkat kekotoran yang terdeteksi. Area dengan tingkat kotoran tinggi akan dibersihkan dengan kecepatan dan gaya sikat yang lebih besar, sedangkan area bersih diperlakukan lebih lembut untuk menjaga lapisan pelindung lambung. Pendekatan ini tidak hanya meningkatkan efektivitas pembersihan, tetapi juga mengurangi kebutuhan docking serta keterlibatan penyelam, sehingga menawarkan solusi yang lebih aman, efisien, dan hemat biaya bagi proses inspeksi dan pemeliharaan kapal.

1.2 Pertanyaan Penelitian

1. Bagaimana perencanaan desain dan sistem brushing untuk proses pembersihan kotoran pada lambung kapal?
2. Bagaimana pengaruh YOLO dalam penentuan kelas untuk sistem kontrol?
3. Bagaimana optimasi pengaturan kecepatan dan performa *response* motor sesuai dengan kondisi kelas *biofouling* pada struktur lambung kapal?

1.3 Batasan Penelitian

Penelitian ini memiliki beberapa batasan yang akan membatasi ruang lingkup pembahasan agar tidak terjadi penyempitan atau pelebaran pokok masalah. Adapun batasan-batasan yang dimaksud adalah sebagai berikut:

1. Area yang akan dibersihkan terbatas pada lambung kapal dengan tekstur standar yang mencakup kotoran dan kerak laut ringan yang umum ditemukan pada kapal. Jenis kotoran lain, seperti tumpahan minyak, biofouling yuang mengerak atau korosi berat, tidak akan dibahas dalam penelitian ini.

2. Fokus desain hanya pada desain sistem mekanisme *brushing*, efek samping brushing terhadap ROV, desain ROV tidak akan dibahas dalam penelitian ini.
3. Penggolongan kotoran hanya dilakukan pada lebar dan besar kotoran, jenis *biofouling* dan ketebalan kotoran bukan yang menjadi fokus utama pada penelitian kali ini.

Dengan batasan-batasan ini, diharapkan pembahasan penelitian dapat terfokus pada analisis penerapan PID pada ROV untuk pembersihan lambung kapal secara efisien.

1.4 Tujuan Penelitian

1. Meningkatkan efektifitas pembersihan lambung kapal melalui desain sistem dan mekanis.
2. Meningkatkan kecepatan deteksi kotoran untuk mempermudah pembersihan yang akan dilakukan.
3. Mengoptimalkan parameter PID agar kecepatan motor mengikuti setpoint berdasarkan kelas biofouling dengan overshoot dan error tunak minimum.

1.5 Manfaat Penelitian

Hasil dari penelitian ini diharapkan dapat memberikan kontribusi yang signifikan baik bagi pengembangan teknologi pembersihan lambung kapal, maupun bagi berbagai pihak yang terlibat dalam industri perkapalan dan teknologi robotika. Manfaat penelitian ini dapat dijelaskan sebagai berikut:

1. **Manfaat bagi Penulis (Peneliti):**
    a. Memberikan wawasan lebih dalam tentang penerapan teknologi kontrol PID dalam sistem pembersihan otomatis, khususnya dalam bidang ROV (*Remotely Operated Vehicle*) dan motor *brush*.
    b. Memperkaya pengetahuan teknis dan keterampilan dalam pengembangan sistem kontrol untuk perangkat robotika yang beroperasi di lingkungan ekstrem seperti kapal.
    c. Menjadi kontribusi dalam bidang teknologi otomasi dan pembersihan industri yang dapat digunakan dalam riset lanjutan atau pengembangan produk serupa di masa depan.

2. **Manfaat bagi Industri Perkapalan dan ABK:**
    a. Memungkinkan proses pembersihan lambung kapal dilakukan dengan lebih efisien dan cepat, tanpa memerlukan docking, yang dapat menghemat waktu dan biaya operasional kapal.
    b. Meningkatkan mobilitas ABK dalam menjalankan tugas mereka, terutama di area yang sulit dijangkau, sehingga meningkatkan efektivitas pekerjaan dan mengurangi risiko kecelakaan kerja.
    c. Menyediakan alternatif teknologi yang lebih aman dan otomatis untuk perawatan kapal, mengurangi ketergantungan pada tenaga manusia yang memerlukan keterampilan khusus dan kehadiran di lokasi.
3. **Manfaat bagi Stakeholder Industri Teknologi:**
    a. Memberikan kontribusi pada pengembangan kurikulum pendidikan yang berhubungan dengan robotika, sistem kontrol, dan aplikasi ROV di industri perkapalan, sebagai bahan pembelajaran bagi mahasiswa di bidang teknik elektro, robotika, dan perkapalan.
    b. Menjadi referensi penting dalam penelitian lanjutan mengenai penggunaan teknologi cerdas dalam aplikasi industri, khususnya dalam pembersihan otomatis dan pengembangan ROV dengan sistem kontrol lebih stabil dan efisien.
    c. Meningkatkan kolaborasi antara industri perkapalan dan institusi pendidikan dalam menciptakan solusi teknologi yang dapat diimplementasikan langsung di lapangan.
4. **Manfaat bagi Konsumen dan Operator Kapal:**
    a. Meningkatkan daya saing kapal dengan menyediakan fasilitas pembersihan lambung yang lebih efisien, yang dapat memperpanjang usia kapal dan mengurangi biaya pemeliharaan jangka panjang.
    b. Memberikan keuntungan operasional bagi operator kapal dengan mempercepat proses pembersihan, yang memungkinkan kapal untuk kembali beroperasi lebih cepat dan mengurangi waktu tidak produktif.

5. **Manfaat bagi Pengembangan Teknologi Robotika dan Sistem Kontrol:** Menyediakan model yang dapat digunakan untuk pengembangan ROV dan sistem kontrol lainnya di berbagai sektor industri yang memerlukan pembersihan otomatis, seperti industri minyak dan gas, serta pembangkit listrik.

BAB 2
TINJAUAN PUSTAKA

2.1 Kajian Penelitian Terdahulu

Sub-bab ini membahas penelitian sebelumnya yang memiliki kaitan erat dengan topik yang sedang diteliti, sehingga dapat menjadi pijakan dan bahan perbandingan dalam pengembangan Tugas Akhir.

1. *Automated Hull Cleaning and Monitoring Robot* (Sanjana et al, 2021).

   Artikel "Automated Hull Cleaning and Monitoring Robot" oleh Sanjana K. S. dkk. membahas perancangan robot bawah air yang dapat melakukan pembersihan lambung kapal dan pemantauan kondisi permukaan secara simultan. Robot yang dikembangkan menggunakan modul brush sebagai aktuator utama pembersih biofouling dan dilengkapi kamera berbasis ESP32-CAM untuk memberikan tayangan langsung (live monitoring) kepada operator selama proses inspeksi dan pembersihan berlangsung. Penulis menjelaskan bahwa biofouling berupa slime, alga, dan organisme keras seperti barnacle menyebabkan peningkatan tahanan kapal, konsumsi bahan bakar, dan emisi, sehingga lambung kapal perlu dibersihkan secara berkala agar kinerja hidrodinamis tetap optimal dan konsumsi bahan bakar dapat ditekan hingga lebih dari 5%.

   Dari sisi desain, robot pada jurnal tersebut terdiri atas silinder utama yang menampung rangkaian kendali dan catu daya, beberapa sirip yang digerakkan servo untuk manuver, sebuah unit brush yang digerakkan motor BLDC A2212 berkekuatan torsi tinggi, serta modul kamera untuk inspeksi visual. Sistem kendali gerak dan pembersihan dilakukan secara nirkabel melalui koneksi Wi-Fi ke ESP32-CAM, sehingga operator dapat menggerakkan robot, memantau area yang tertutup biofouling, dan mengaktifkan brush untuk membersihkan permukaan lambung tanpa perlu penyelam turun ke air. Kesimpulan utama artikel tersebut adalah bahwa penggunaan robot pembersih lambung dengan brush dan sistem monitoring mampu meningkatkan keselamatan, mengurangi ketergantungan pada tenaga manusia, serta

berpotensi menghemat biaya operasional melalui pengurangan drag dan penghematan bahan bakar.

Penggunaan brushless DC motor (BLDC) sebagai penggerak utama unit brushing sangat relevan dengan kebutuhan pembersihan lambung kapal karena motor jenis ini mampu menghasilkan torsi besar, efisiensi tinggi, dan respon dinamis yang baik pada kecepatan rendah hingga menengah, yang dibutuhkan untuk menyikat kerak dan biofouling secara efektif di bawah air. Pada artikel *Automated Hull Cleaning and Monitoring Robot*, BLDC tipe A2212 dipilih untuk memutar sikat pembersih karena karakteristik torsi dan efisiensinya, sehingga sikat dapat bekerja stabil saat menghilangkan fouling seperti slime, alga, dan barnacle pada permukaan lambung. Desain ini menunjukkan bahwa BLDC sangat cocok ditempatkan sebagai aktuator utama dalam sistem brushing, dan pada penelitianmu peran tersebut diambil alih oleh motor brushless M200 yang memiliki kapasitas torsi lebih besar untuk aplikasi ROV bawah air.

Dari sisi kontrol dan visualisasi, jurnal tersebut menempatkan BLDC dalam sistem yang dikendalikan secara nirkabel melalui ESP32-CAM, di mana operator dapat melihat kondisi permukaan lambung secara langsung lalu mengaktifkan dan mengatur gerakan robot serta brush berdasarkan tayangan video tersebut. Visualisasi yang diberikan hanya berupa live monitoring sehingga keputusan kapan dan di mana brush diaktifkan sepenuhnya bergantung pada pengamatan manusia. Pada risetmu, konsep ini dikembangkan lebih lanjut dengan menambahkan pengendali PID pada BLDC untuk menjaga kestabilan putaran motor sesuai setpoint, serta menggunakan visualisasi berbasis YOLO yang tidak hanya menampilkan citra, tetapi juga secara otomatis mendeteksi dan mengklasifikasikan tingkat biofouling menjadi beberapa kelas (kecil, sedang, besar) yang kemudian diterjemahkan menjadi nilai setpoint rpm. Dengan demikian, BLDC berfungsi sebagai aktuator utama brushing yang kecepatannya dikendalikan secara adaptif berdasarkan hasil visualisasi cerdas, sehingga proses pembersihan menjadi lebih terukur, konsisten, dan tidak sepenuhnya bergantung pada subjektivitas operator(Sanjana et al., 2021).

2. *Motion Control of a Ship Hull Cleaning Robot* (Yamoto et al, 2022).

Artikel “Motion Control of a Ship Hull Cleaning Robot” oleh Yamamoto dkk. membahas generasi ke-4 robot pembersih lambung kapal yang dirancang sebagai ROV dengan unit brush rotari di tengah dan enam buah thruster untuk mempertahankan posisi serta sikap terhadap permukaan hull. Robot berukuran sekitar 0,8 m × 0,3 m × 0,4 m dengan berat ±30 kg, dilengkapi dua silinder tahan tekanan yang memuat rangkaian catu daya di bagian depan dan rangkaian sensor serta kontrol di bagian belakang, serta mampu beroperasi hingga kedalaman 30 m. Arsitektur sistemnya memanfaatkan komunikasi Power Line Communication (PLC) antara robot dan komputer kendali di darat dengan bandwidth yang cukup untuk mengirim dua video kamera 30 fps, sementara di dalam robot digunakan TRAX2 sebagai Attitude and Heading Reference System (AHRS), sensor Hall pada roda pasif untuk mendeteksi kontak dengan lambung, serta sensor suhu, tekanan, dan kebocoran untuk pemantauan kondisi internal. Dua mikrokontroler (PIC32MX dan dsPIC33F) memisahkan fungsi komunikasi tingkat tinggi dan akuisisi sensor dari fungsi kontrol aktuator, sehingga struktur kontrol terbagi menjadi lapisan high-level dan low-level.

Fokus utama artikel ini adalah perancangan dan pemodelan sistem kontrol gerak dan sikap robot saat menempel dan bergerak di sepanjang lambung kapal. Penulis memodelkan dinamika gerak utama (khususnya gerak vertikal/heave dan sikap pitch) dengan mempertimbangkan gaya apung, berat, gaya thruster, dan gaya kontak pada roda/brusher, kemudian menerapkan pengendali berbasis umpan balik (PD/PID) untuk mengatur keluaran thruster agar robot dapat mempertahankan jarak dan sudut tertentu terhadap permukaan hull selama operasi pembersihan. Sensor AHRS digunakan untuk mengukur sudut dan laju sudut, sedangkan sensor tekanan memberikan informasi kedalaman; kedua besaran ini dibandingkan dengan nilai referensi (setpoint) dan error-nya diolah oleh pengendali (misalnya PID) untuk menghasilkan perintah gaya pada thruster yang menjaga sikap dan kedalaman robot stabil meskipun terdapat gangguan seperti gelombang atau variasi orientasi permukaan lambung. Pengujian menunjukkan bahwa dengan skema kontrol ini, robot mampu mengikuti perubahan sudut permukaan hull dan tetap menempel sambil

mengoperasikan brush, sehingga kualitas pembersihan meningkat karena sikat berada pada posisi dan tekanan yang relatif konstan(Yamamoto dkk., 2022).

Peneliti menyimpulkan bahwa studi literatur untuk pengembangan proyek serupa, jurnal ini memberikan contoh lengkap bagaimana pemodelan dinamik dan perancangan pengendali PID/PD diterapkan pada ROV hull cleaning, bukan hanya untuk menggerakkan robot tetapi untuk menjaga sikap ketika unit brush bekerja di dekat permukaan kapal. Hal ini dapat dijadikan rujukan kuat untuk menyusun bagian dasar teori dan penelitian terdahulu terkait sistem kontrol, sekaligus menunjukkan perbedaan dan pengembangan pada tugas akhirmu: jika Yamamoto dkk. memfokuskan PID pada kontrol sikap dan gerak ROV berbasis AHRS dan sensor tekanan, penelitianmu memfokuskan PID pada stabilitas kecepatan motor brushless brusher dengan umpan balik encoder dan sensor jarak ultrasonik, sementara aspek penentuan setpoint pembersihan diperkaya dengan deteksi dan klasifikasi biofouling otomatis menggunakan YOLO sebagai pengganti pengamatan visual konvensional.

3. *RSE-YOLOv8: An Algorithm for Underwater Biological Target Detection* (Peihang et sl., 2024)

   Artikel “RSE-YOLOv8: An Algorithm for Underwater Biological Target Detection” menunjukkan bahwa arsitektur YOLO dapat diadaptasi secara khusus untuk mengatasi permasalahan khas citra bawah air, seperti hilangnya informasi kanal akibat penyerapan dan hamburan cahaya. Penulis mengembangkan varian YOLOv8 dengan tiga modul utama—C2f_SAConv, RFESEConv, dan ESPPF—yang bersama-sama meningkatkan kemampuan ekstraksi fitur multi-skala, kepekaan terhadap distorsi kanal, serta efisiensi pooling, sehingga model lebih stabil dalam mendeteksi organisme bentik di lingkungan air yang keruh dan bercahaya rendah.

   Dari sisi performa, RSE-YOLOv8 diuji pada dua dataset publik (UnderwaterBio dan URPC2020) dan menunjukkan peningkatan mAP@0,5 serta mAP@0,5:0,95 sebesar sekitar 2,1% dibanding YOLOv8 standar, dengan jumlah parameter dan kebutuhan komputasi yang tetap rendah sehingga cocok untuk deployment pada perangkat bergerak dan sistem embedded. Ablation study yang dilakukan penulis memperlihatkan bahwa setiap modul

(C2f_SAConv untuk fitur multi-skala, RFESEConv untuk perhatian kanal/receptive field, dan ESPPF untuk pooling paralel ringan) memberikan kontribusi nyata terhadap peningkatan akurasi dan efisiensi, sekaligus menjaga kecepatan inference agar tetap sesuai untuk aplikasi real-time.(Song dkk., 2024)

Dalam penelitian ROV hull cleaning, jurnal ini memberikan justifikasi kuat bahwa penggunaan YOLO sebagai "otak visual" pada sistemmu adalah pendekatan yang realistis dan didukung hasil eksperimen, terutama ketika model perlu berjalan pada platform dengan sumber daya terbatas seperti komputer kecil yang terhubung ke ROV. Pendekatan desain RSE-YOLOv8 menunjukkan bahwa dengan beberapa modifikasi pada backbone dan modul pooling, deteksi objek bawah air dapat dibuat cukup akurat dan ringan untuk dipakai pada inspeksi otomatis; hal ini sejalan dengan idemu memanfaatkan YOLO untuk mendeteksi dan mengklasifikasikan patch biofouling sebagai dasar penentuan setpoint rpm brusher dan pengaturan strategi pembersihan pada ROV hull cleaning.

4. *Automating the assessment of biofouling in images using expert agreement as a gold standard* (Mannix et al., 2021).

Selain itu, ketertarikan pada kemajuan terbaru dalam inspeksi otomatis menggunakan deep learning untuk penilaian biofouling bawah air pada lambung kapal melalui ROV. Kemampuan jaringan saraf konvolusional (CNN) untuk mengklasifikasikan tingkat keparahan biofouling dengan akurasi setara ahli menginspirasi saya untuk mengadopsi Jaringan Saraf Tiruan (JST) tidak hanya untuk analisis data tetapi juga untuk meningkatkan adaptabilitas kontrol PID dalam sistem ROV. Strategi ganda ini menjanjikan mekanisme pembersihan lambung kapal yang responsif dan cerdas, mampu menyesuaikan diri dengan perubahan lingkungan (Mannix et al., 2021).

Ide-ide yang disintesis dari penelitian yang telah ada ini memberikan dasar ilmiah yang komprehensif untuk studi penulis. Penulis berupaya menggabungkan teori kontrol klasik dengan algoritma cerdas modern untuk mengoptimalkan kontrol PID pada motor BLDC dalam aplikasi(Mannix dkk., 2021).

2.2 Kajian Pustaka

Kajian pustaka berfungsi sebagai landasan teoretis yang memperkuat penelitian ini dengan merujuk pada berbagai sumber ilmiah. Penyusunan kajian pustaka ini juga bertujuan untuk mengidentifikasi kesenjangan penelitian yang menjadi dasar penelitian ini.

2.2.1 Remotely Operated Vehicle (ROV)

Kendaraan tanpa awak yang dioperasikan secara remote (ROV) dilengkapi dengan sistem kontrol jarak jauh yang dikendalikan dari permukaan dan digunakan untuk melakukan observasi, inspeksi, dan intervensi di lingkungan bawah laut (Kuruvilla dkk., 2023). ROV dilengkapi dengan berbagai sistem, termasuk kamera, sistem pencahayaan, dan sensor yang terhubung ke kabel umbilikal. Bentuk visual ROV ditunjukkan pada gambar 2.1.

Gambar 2. 1 *Remotely Operated Vechile*

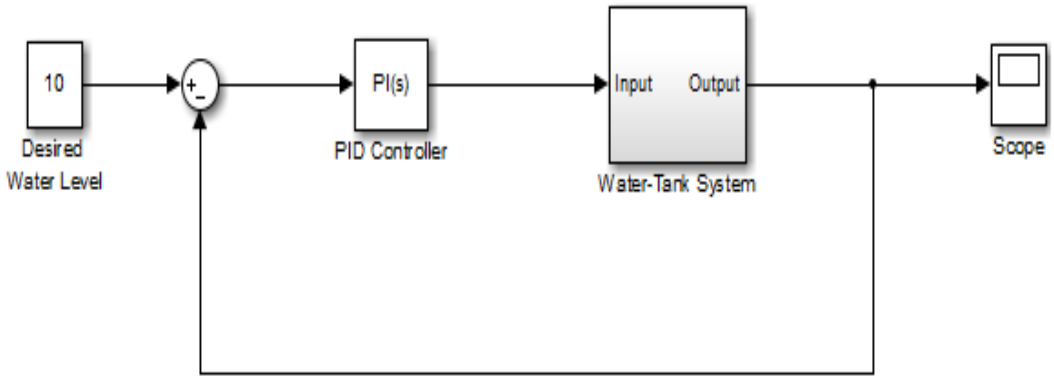
(Sumber: https://www.ecoptik.net/about-underwater-rov.html**)**

Dibutuhkan sistem tenaga, pencahayaan, dan sensor ROV yang andal karena lingkungan bawah laut yang sulit, seperti tekanan tinggi, korosivitas air laut, kekeruhan, dan kurangnya penerangan. Ini disebabkan oleh kebutuhan akan sistem catu daya dan pengisian baterai yang stabil dari permukaan. Oleh karena itu, ROV diperlukan untuk memaksimalkan proses eksplorasi bawah air dan membantu

pekerjaan manusia dalam misi bawah laut. Dalam penelitian tugas akhir ini, ROV Inpection Class digunakan. ROV ini dirancang untuk melakukan observasi dan pengumpulan data untuk keperluan inspeksi.

2.2.2 Proposional-Integral-Derivative (PID)

PID (Proportional-Integral-Derivative) Controller merupakan algoritma kontrol yang banyak digunakan dalam sistem pengendalian otomatis untuk mengatur variabel seperti kecepatan, suhu, dan posisi. Kontrol PID bekerja dengan meminimalkan **error** antara nilai yang diinginkan (*setpoint*) dan nilai aktual hasil pengukuran. Proses pengendalian dilakukan melalui tiga parameter utama, yaitu *Proportional (P)*, *Integral (I)*, dan *Derivative (D)*. Ketiga parameter tersebut memiliki karakteristik berbeda dalam merespons *error*, sehingga dapat disesuaikan untuk memperoleh respons sistem yang stabil, cepat, dan mampu mempertahankan kondisi operasi mendekati nilai *setpoint*.



Gambar 2. 2 PID *Block* Diagram

(sumber: https://www.mathworks.com/help/simulink/slref/pidcontroller.html)

2.2.2.1 *Proportional (P)*

Komponen P melakukan tindakan kontrol yang setara dengan error sesaat, yang merupakan perbedaan antara setpoint dan output sebenarnya. Sepertinya, lebih banyak "push" yang dibutuhkan oleh kontrol untuk memperbaiki sistem. Waktu naik pada tahap awal akan dipercepat dengan KP yang lebih tinggi. Namun, jika ukurannya terlalu besar, sistem dapat berosilasi, mengalami overshoot tinggi, atau bahkan menjadi tidak stabil secara marginal, seperti yang terlihat dalam metode Ziegler-Nichols untuk menemukan Ku dan Tu.

2.2.2.2 *Integral (I)*

Komponen integral sensitif terhadap kesalahan kecil yang bertahan lama karena mereka mengakumulasi kesalahan seiring waktu. Fungsi utamanya adalah untuk menghilangkan kesalahan steady-state yang masih ada ketika hanya menggunakan kontrol P. Misalnya, error kecepatan BLDC yang tidak pernah sampai ke setpoint atau error posisi/kecepatan ROV yang selalu meleset. Namun, Ki yang terlalu besar menyebabkan integrator "menumpuk" kesalahan, yang menyebabkan overshoot besar dan respons lamban karena sistem membutuhkan waktu untuk "menguras" akumulasi integral, juga dikenal sebagai windup integrator.

2.2.2.3 *Derivative (D)*

Komponen derivative bersifat prediktif dan berfungsi sebagai peredam karena bekerja berdasarkan laju perubahan error (turunan terhadap waktu). Term D memberikan aksi yang berlawanan untuk meredam gerakan jika error berubah terlalu cepat. Ini mengurangi overshoot dan osilasi dan meningkatkan settling time. Sebaliknya, Kd yang terlalu besar membuat sistem sensitif terhadap suara sensor karena D sensitif terhadap perubahan kecil. Akibatnya, sinyal kontrol menjadi "berderau" dan aktuator tidak bekerja dengan baik.

Secara matematis, berdasarkan “Comparative Analysis: Fractional PID vs. PID Controllers for Robotic Arm Using Genetic Algorithm Optimization“ oleh (Eltayeb dkk., 2024) kontrol PID dapat dijelaskan dengan persamaan berikut:

$$x(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt} \quad (2.1)$$

Dimana:

- u(t) adalah output kontrol yang diberikan ke sistem (misalnya, sinyal PWM untuk motor).
- e(t) adalah kesalahan (error), dengan r(t) adalah setpoint (nilai yang diinginkan) dan y(t) adalah nilai yang terukur.
- Kp adalah gain proporsional yang menentukan seberapa besar pengaruh komponen P terhadap kesalahan.

- Ki adalah gain integral yang menentukan seberapa besar pengaruh komponen I terhadap akumulasi kesalahan.
- Kd adalah gain derivatif yang menentukan seberapa besar pengaruh komponen D terhadap perubahan kesalahan.

Dalam aplikasi seperti ROV hull cleaner, kontrol PID digunakan untuk mengatur kecepatan motor yang menggerakkan sikat pembersih hull kapal. Kontrol PID memungkinkan pengaturan motor dengan sangat presisi, menjaga agar sikat bergerak sesuai dengan kecepatan yang telah ditentukan berdasarkan ukuran biofouling yang terdeteksi oleh kamera. Proses ini memastikan bahwa pembersihan dapat dilakukan secara efisien, tanpa merusak lapisan pelindung pada kapal.

Penelitian yang dilakukan oleh Al Fayyed dkk. memberikan landasan kuat bahwa kualitas kontrol kecepatan motor BLDC sangat bergantung pada proses penentuan parameter PID yang sistematis, bukan sekadar trial and error (Muhammad Nabel Al Fayyed dkk., 2025). Dengan menerapkan metode Ziegler Nichols, berhasil menurunkan rise time dan settling time beberapa orde lebih cepat sekaligus menghilangkan error steady state yang sebelumnya mencapai lebih dari 45%, sehingga respon kecepatan menjadi jauh lebih presisi dan stabil. Bagi peneliti yang mengembangkan sistem brusher pada ROV hull cleaning, hasil ini menunjukkan bahwa penerapan PID pada motor brushless M200 perlu diiringi dengan prosedur tuning terstruktur serupa, agar perubahan beban akibat tekanan sikat dan variasi tingkat biofouling dapat dikompensasi secara efektif dan kecepatan putar tetap sesuai setpoint yang ditentukan oleh keluaran YOLO(Muhammad Nabel Al Fayyed dkk., 2025).

2.2.3 You Only Look Once (YOLO)

You Only Look Once (YOLO) adalah metode deteksi objek berbasis pembelajaran mendalam yang digunakan untuk mengenali dan mengklasifikasikan objek dalam gambar secara *real-time*. Berbeda dengan metode lainnya yang melakukan deteksi objek dalam dua tahap (misalnya, pengenalan dan klasifikasi), YOLO menggabungkan kedua proses tersebut dalam satu tahap yang efisien. YOLO bekerja dengan membagi gambar menjadi grid dan memprediksi batas objek serta kelasnya dalam setiap grid secara simultan. Hal ini membuat YOLO sangat

cepat dan efisien, memungkinkan untuk deteksi objek dengan latensi rendah, bahkan dalam kondisi yang penuh tantangan.(Song dkk., 2024)

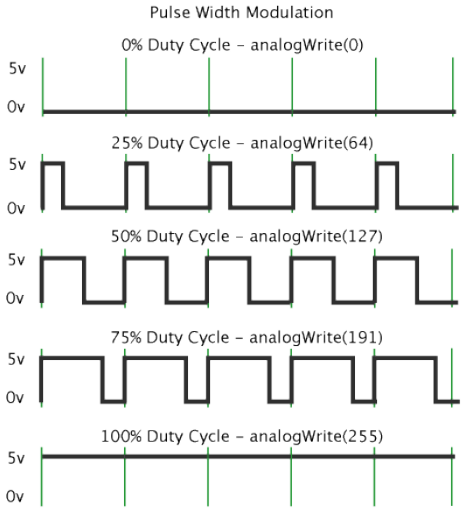
Dalam aplikasi pembersihan hull kapal otomatis, YOLO digunakan untuk mendeteksi dan mengklasifikasikan ukuran serta jenis biofouling berdasarkan data visual yang diambil dari kamera. Proses dimulai dengan pengambilan gambar biofouling yang ada di badan kapal, yang kemudian diproses oleh model YOLO untuk mendeteksi objek-objek *biofouling* dan mengklasifikasikannya sesuai kategori yang telah ditentukan. Model YOLO dilatih menggunakan dataset yang berisi gambar biofouling dengan label yang sesuai, memungkinkan jaringan untuk belajar mengenali pola-pola yang ada dalam gambar dan mendeteksi objek dengan akurasi yang tinggi.

Penggunaan YOLO dalam aplikasi ini memberikan keuntungan besar dalam hal kecepatan dan akurasi pengenalan objek di bawah air, terutama dengan kondisi pencahayaan rendah dan lingkungan yang sering terganggu oleh partikel. Sebuah penelitian oleh (Redmon dkk., 2021) menunjukkan bahwa YOLO, dengan kemampuannya untuk melakukan deteksi objek secara real-time, sangat efektif dalam mendeteksi dan mengklasifikasikan objek dalam gambar, termasuk objek yang sulit dikenali seperti biofouling di bawah air, dengan akurasi tinggi dibandingkan dengan metode deteksi objek lainnya

2.2.4 PWM – Pulse Width Modulation

PWM Controller (Pulse Width Modulation Controller) adalah perangkat elektronik yang digunakan untuk mengatur kecepatan motor, terutama pada motor listrik seperti motor BLDC (Brushless DC Motor). (Kuo dkk., 2022) dalam artikel yang berjudul “Position Tracking of an Underwater Robot Based on Floating-Downing PI Control” PWM bekerja dengan cara mengatur rasio waktu antara sinyal ON dan OFF yang dikirimkan ke motor. Rasio ini disebut *duty cycle*, yang mengontrol jumlah daya yang diberikan ke motor. Semakin tinggi *duty cycle*, semakin besar daya yang diterima oleh motor, yang menghasilkan kecepatan motor yang lebih tinggi. Sebaliknya, semakin rendah duty cycle, semakin sedikit daya yang diberikan, dan motor berputar lebih lambat. Dengan demikian, PWM *controller* memungkinkan pengaturan kecepatan motor yang sangat halus dan efisien.

Sistem PWM sangat efektif karena meminimalkan pemborosan energi dan mengurangi pemanasan berlebih pada motor. Ini sangat penting dalam aplikasi robotik dan kendaraan listrik, di mana efisiensi energi dan kontrol kecepatan yang tepat sangat dibutuhkan. Dalam aplikasi *ROV (Remotely Operated Vehicle)*, PWM controller digunakan untuk mengatur kecepatan motor BLDC yang menggerakkan sikat pembersih hull kapal, sehingga memungkinkan pengoperasian motor dengan presisi tinggi.



Gambar 2. 3 Representatif sinyal PWM

(sumber: https://www.arduino.cc/en/Tutorial/PWM)

Frekuensi PWM adalah jumlah siklus per detik (Hz) dari sinyal PWM. Frekuensi ini menentukan seberapa cepat sinyal PWM berulang, dan biasanya disesuaikan untuk aplikasi tertentu. Rumus untuk menghitung frekuensi PWM (f) adalah:

$$f = \frac{1}{T_{total}} \qquad (2.2)$$

Dimana:

- T total adalah periode dari satu siklus PWM (dalam detik).

Dengan PWM, daya yang diberikan ke motor atau beban lainnya dapat dihitung berdasarkan duty cycle dan tegangan yang diterapkan. Daya rata-rata P avg yang diterima oleh beban dari sinyal PWM dapat dihitung dengan rumus:

$$P_{avg} = D \times V_{in} \times I \qquad (2.3)$$

Dimana:

- P avg adalah daya rata-rata yang diberikan ke beban,
- D adalah *duty cycle*
- Vin adalah tegangan input yang diberikan ke beban (motor),
- I adalah arus yang mengalir ke beban (motor).

Untuk motor BLDC, yang menggunakan PWM untuk kontrol kecepatan, daya rata-rata yang diberikan ke motor bisa dihitung sebagai:

$$P_{avg} = D \times V_{supply} \times I_{motor} \quad (2.4)$$

Dimana:

- Vsupply adalah tegangan suplai yang diberikan ke motor (biasanya tegangan DC),
- ImotorI adalah arus yang mengalir ke motor.

Berdasarkan penelitian “Design and modeling of an experimental rov with six degrees of freedom” oleh (Kabanov dkk., 2021) pada sistem ROV, Pulse Width Modulation (PWM) berperan sebagai metode utama untuk mengatur kecepatan dan arah putar thruster maupun aktuator lain dengan cara mengubah lebar pulsa pada sinyal kendali tanpa mengorbankan efisiensi daya. Jurnal yang direview menjelaskan bahwa variasi duty cycle PWM memungkinkan pengaturan gaya dorong ROV secara halus, sehingga kendaraan dapat mempertahankan kedalaman, melakukan manuver maju–mundur, maupun mengikuti lintasan referensi meskipun dipengaruhi gangguan arus dan gelombang. Dengan memodulasi duty cycle, tegangan dan daya efektif yang diterima motor berubah secara proporsional, yang kemudian diterjemahkan menjadi perubahan kecepatan dan torsi thruster. Dari sudut pandang peneliti yang sedang menyusun dasar teori, temuan tersebut menegaskan bahwa PWM bukan hanya teknik switching, tetapi elemen kunci dalam loop kontrol ROV, karena menjadi jembatan antara keluaran algoritma kontrol seperti PID dengan respon mekanik thruster, sejalan dengan konsep penelitian ini yang menggunakan PWM ESC untuk mengatur kecepatan motor brushless M200 pada sistem hull cleaning secara adaptif berdasarkan hasil deteksi biofouling (Kabanov dkk., 2021).

2.3 Kajian Pustaka Perangkat Keras dan Lunak

2.3.1 Kamera

Kamera bawah air pada sistem *Remotely Operated Vehicle* (ROV) berfungsi sebagai sensor visual utama yang menangkap citra dan video lingkungan bawah air secara *real-time*(Akram dkk., 2022). Citra yang diambil kamera memungkinkan pengamatan langsung terhadap objek atau struktur yang diamati, seperti permukaan lambung kapal, penampang pipa, atau objek lain di lingkungan laut tanpa memerlukan penyelaman manusia. Kamera menyajikan bukti visual yang dapat dipantau oleh operator untuk menilai kondisi fisik struktur serta digunakan sebagai input data untuk berbagai proses pemrosesan citra dan *computer vision* dalam aplikasi pemantauan bawah air seperti inspeksi struktur maupun deteksi objek. Dalam konteks pemantauan bawah air, penggunaan kamera ROV sangat penting karena citra visual dapat memberikan informasi morfologi dan tekstur permukaan yang tidak dapat diukur hanya dengan sensor non-visual seperti sonar atau ultrasonik saja.

Gambar 2. 4 Kamera

Dalam analisis dan pengamatan otomatis, data dari kamera ROV dapat dianalisis menggunakan teknik pembelajaran mendalam seperti YOLOv8-Segmentation untuk mengidentifikasi dan memisahkan area korosi atau kerusakan pada permukaan kapal. Dengan menggunakan model segmentasi berbasis pembelajaran mendalam pada udara, dimungkinkan untuk mendeteksi area target secara otomatis dan kuantitatif dari data visual yang diambil oleh kamera, yang mempercepat proses diagnosis struktural dibandingkan dengan analisis manual.

Penelitian sebelumnya menunjukkan bahwa citra bawah air yang dikumpulkan oleh ROV dapat diproses menggunakan algoritma penglihatan komputer untuk mendukung tugas inspeksi seperti deteksi objek atau pola kerusakan, yang meningkatkan penggunaan kamera sebagai sumber data utama dalam sistem pemantauan cerdas bawah air.

2.3.2 Raspberry Pi 3B

Raspberry Pi adalah platform komputasi kecil dan hemat energi yang sangat populer untuk berbagai aplikasi, termasuk dalam bidang robotik, penginderaan, dan pemrosesan data. Raspberry Pi dilengkapi dengan prosesor berbasis ARM, memori RAM, serta port I/O yang memungkinkan penghubungan dengan berbagai sensor dan perangkat eksternal seperti kamera, motor, dan sensor jarak. Platform ini cocok digunakan dalam sistem yang membutuhkan pemrosesan data secara real-time, termasuk dalam aplikasi pembersihan hull kapal otomatis menggunakan kamera untuk deteksi biofouling dan pemrosesan data visual.

Gambar 2. 5 *Raspberry Pi* 3B

(sumber: https://www.arduinoindonesia.id/2024/03/spesifikasi-lengkap-raspberrypipico.html)

Pada tabel 2.1 merupakan spesifikasi dari Raspberry yang digunakan acuan peneliti dalam pembuatan sistem brushing bagaimana mikrokontrolernya.

Tabel 2. 1 *Spesifikasi Raspberry*

Spesifikasi	
Prosesor	Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.8GHz
Memori (RAM)	1GB, 2GB, 4GB or 8GB LPDDR4-3200 SDRAM (depending on model)
Jaringan	2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE
Port USB	2 USB 3.0 ports; 2 USB 2.0 ports.
Pin	Raspberry Pi standard 40 pin GPIO header (fully backwards compatible with previous boards)
Output Vidio	2 × micro-HDMI® ports (up to 4kp60 supported)
Output Audio	4-pole stereo audio and composite video port
Penyimpanan	Micro-SD card slot for loading operating system and data storage
Input Daya	5V DC via USB-C connector (minimum 3A*)
	5V DC via GPIO header (minimum 3A*)
Dimensi	85,6 mm x 56,5 mm x 17 mm

Artikel “A Raspberry Pi-based camera system and image processing procedure for low cost and long-term monitoring of forest canopy dynamics” menjelaskan pengembangan sistem monitoring kanopi hutan berbiaya rendah berbasis Raspberry Pi 3 Model B+ yang dikombinasikan dengan modul kamera fisheye 170°, catu daya UPS 12 V–5 V, serta enclosure IP65 dengan kubah akrilik untuk melindungi perangkat di lapangan. Unit Raspberry Pi diprogram dengan Python untuk mengambil citra kanopi menghadap ke atas setiap 30 menit, menyimpannya secara lokal, dan mengirimkannya ke Raspberry Pi kontrol melalui jaringan Wi-Fi lokal menggunakan rsync, sehingga tercapai pemantauan jangka panjang dengan resolusi temporal tinggi dan biaya sekitar £80 per unit. Penulis juga

menyusun rantai pemrosesan citra end-to-end berbasis R dan paket cimes-fisheye untuk mengekstraksi parameter kanopi secara semi-otomatis (Matthew et al, 2021).

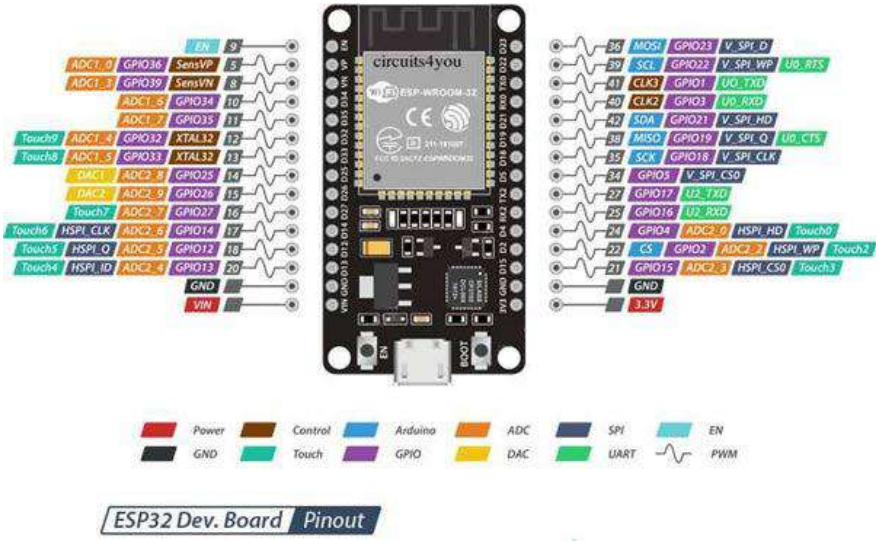
Dalam konteks penelitian ini ROV hull cleaning berbasis YOLO, Raspberry Pi layak diposisikan sebagai komputer utama untuk pemrosesan citra dan koordinasi jaringan. Dengan mengadaptasi pola arsitektur ini, peneliti dapat merancang sistem di mana Raspberry Pi menangani inference YOLO dan manajemen data, sedangkan lapisan bawah fokus pada kontrol real-time dan aktuasi.

2.3.3 ESP 32

ESP32 didefinisikan sebagai mikrokontroler berkemampuan komputasi dan komunikasi nirkabel yang berperan sebagai pusat kendali dalam sistem tertanam, khususnya pada aplikasi pemantauan dan proteksi berbasis sensor. Dalam jurnal yang berjudul "Perancangan Keputusan Cut OffBaterai ROV menggunakan Metode Decision Tree", ESP32 digunakan sebagai kontrol utama pada sistem cut off baterai ROV, di mana mikrokontroler ini mengolah data arus dan tegangan dari sensor INA219 untuk menentukan keputusan yang tepat untuk pemutusan atau kelanjutan suplai daya (Zhafirah dkk., 2024).

Tabel 2. 2 Spesifikasi ESP-32

Prosesor	Microprosesor Xtensa Dual-Core 32 Bit LX6
Frekuensi	Freq Clock up to 240 MHz
Memori(RAM)	SRAM 520 kB
Penyimpanan	Flash memori 4 MB
Internet	11b/g/n WiFi transceiver
Bluetooth	Bluetooth 4.2/BLE
Pin	48 pin GPIO
	15 pin channel ADC (*Analog to Digital Converter*)
	25 pin PWM (Pulse Width Modulation
	2 pin channel DAC (*Digital to Analog Converter*)



Gambar 2. 6 ESP 32 *Pin Out*

(sumber: https://www.ardutech.com/wp-content/uploads/2020/03/4.-ESP32-Pinout.jpg)

Dari sudut pandang peneliti yang sedang merancang brusher dengan kcepatan yang adaptif, ESP32 dipandang relevan karena mampu melakukan pemantauan parameter listrik secara real-time dan mengeksekusi algoritma pengambilan keputusan, untuk menjaga keamanan dan keandalan sistem brushing pada ROV.

2.3.4 Motor *Brushless* DC

Motor *Brushless* DC (BLDC) adalah motor listrik yang menggunakan komponen elektronik untuk menggantikan sikat (*brush*) dan komutator yang ada pada motor DC konvensional. Motor ini terdiri dari dua komponen utama: rotor permanen dan stator yang dilengkapi dengan kumparan. Keuntungan utama dari motor BLDC adalah efisiensi tinggi, daya tahan lama, serta minimnya perawatan karena tidak ada sikat yang aus seperti pada motor DC tradisional. Dalam motor BLDC, komutasi dilakukan secara elektronik menggunakan kontroler elektronik (*ESC - Electronic Speed Controller*), yang mengatur aliran arus ke stator dengan menggunakan sinyal PWM (*Pulse Width Modulation*) untuk mengatur kecepatan dan arah putaran rotor.

Motor BLDC banyak digunakan dalam aplikasi robotik, kendaraan listrik, dan sistem pembersihan *hull* kapal otomatis karena kemampuannya untuk memberikan torsi yang stabil dan kecepatan yang dapat dikontrol dengan presisi

tinggi. Salah satu aplikasi utamanya adalah pada *ROV (Remotely Operated Vehicle)*, di mana motor BLDC digunakan untuk menggerakkan thruster dan sistem pembersih *hull* dengan akurasi yang sangat baik. Selain itu, motor BLDC juga lebih efisien dalam konsumsi energi, menghasilkan lebih sedikit panas dan lebih sedikit suara dibandingkan motor *brushed* DC.

Gambar 2. 7 Motor *Brushless DC*

(sumber: https://bluerobotics.com/wp-content/uploads/2022/09/M200-RETAIL-1.jpg)

Pada tabel 2.3 merupakan spesifikasi dari BLDC M200 yang digunakan acuan peneliti dalam pembuatan Brusher bagaimana penggerak brush.

Tabel 2. 3 Spesifikasi BLDC M200

Spesifikasi	**Keterangan**
Operating Voltage	7-20 V
Max Current	32 A
Max Power	645 W
Motor Kv	470 RPM/V

Penelitian Al Fayyed dkk. menjelaskan bahwa motor BLDC menawarkan efisiensi energi lebih tinggi, rasio torsi terhadap berat yang besar, respon dinamis cepat, operasi senyap, minim percikan, dan kebutuhan perawatan yang rendah, sehingga banyak digunakan pada otomotif, peralatan rumah tangga, robotika, dan sistem otomatisasi yang menuntut keandalan dan presisi kecepatan (Al Fayyed et al, 2025).

Dalam konteks riset saya tentang ROV hull cleaning, karakteristik efisiensi tinggi, torsi besar pada ukuran kompak, serta kemampuan respon cepat terhadap perubahan beban membuat motor BLDC (misalnya M200) sangat sesuai sebagai aktuator brusher yang harus mempertahankan rpm stabil saat tekanan sikat dan tingkat biofouling berubah, sehingga temuan dalam jurnal ini menjadi landasan kuat untuk pemilihan jenis motor dan perancangan sistem kontrol kecepatan berbasis PID dan modulasi PWM ESC pada ROV yang dikendalikan oleh YOLO sebagai penentu kelas biofouling.

2.3.5 ESC - *Electronic Speed Controller*

ESC - *Electronic Speed Controller* adalah perangkat elektronik yang digunakan untuk mengontrol kecepatan, arah, dan posisi motor *Brushless* DC (BLDC). ESC bekerja dengan memberikan sinyal PWM (*Pulse Width Modulation*) ke motor, yang mengatur aliran arus ke kumparan stator motor, sehingga mengontrol torsi dan kecepatan rotasi rotor. Berbeda dengan motor DC yang memiliki brush, motor BLDC tidak memerlukan komutator mekanis, sehingga mengurangi kebutuhan perawatan dan meningkatkan efisiensi operasional. ESC ini sangat penting dalam aplikasi robotik dan sistem pembersihan bawah air otomatis, di mana kontrol yang presisi diperlukan untuk menjaga stabilitas pergerakan ROV (*Remotely Operated Vehicle*) atau sistem pembersih *hull*.

ESC pada motor BLDC bekerja dengan cara menerima sinyal dari mikrokontroler atau sistem kontrol seperti PID *controller*, yang kemudian diterjemahkan menjadi sinyal PWM untuk mengatur kecepatan dan arah motor. Sistem ini memberikan kontrol yang lebih halus dan responsif dibandingkan dengan motor DC biasa, memungkinkan motor untuk beroperasi pada kecepatan tinggi dengan efisiensi yang optimal. Selain itu, ESC juga dilengkapi dengan fitur proteksi,

seperti perlindungan terhadap suhu berlebih dan tegangan berlebih, yang meningkatkan daya tahan sistem secara keseluruhan.

Gambar 2. 8 *Electronic Speed Control*

(sumber: https://id.aliexpress.com/item/4000187883267.html)

Pada tabel 2.4 merupakan spesifikasi dari ESC yang digunakan acuan peneliti dalam pembuatan sistem brushing bagaimana kontrol kecepatan.

Tabel 2. 4 Spesifikasi ESC

Spesifikasi	Keterangan
Continous Current	70 A
Busrt Current	420 A
LiPo Cells	2-6 S
Output	5.5 V/5 A
Materials	*Metal Waterproof*
Size	57 x 49 x 18 mm
Weight	101 g

Penelitian Parag dkk. menunjukkan bahwa Electronic Speed Controller (ESC) berperan sebagai penghubung utama antara sumber daya DC, sinyal kendali

PWM dari mikrokontroler/servo tester, dan motor BLDC pada sistem ROV. Pada rancangan underwater drone empat thruster, ESC menerima sinyal PWM dari servo tester untuk mengatur kecepatan dan arah putar motor, sehingga gaya dorong dapat diatur halus dan stabil saat manuver di berbagai arah. Pengujian struktur dasar ROV dengan empat motor BLDC dan ESC menunjukkan bahwa konfigurasi ini mampu memberikan daya dorong yang cukup sekaligus mempertahankan kestabilan gerak selama pengujian di lingkungan bawah air, dengan kontrol kecepatan yang mulus dari ESC menjadi faktor penting dalam menjaga stabilitas tersebut (Parag et al, 2023).

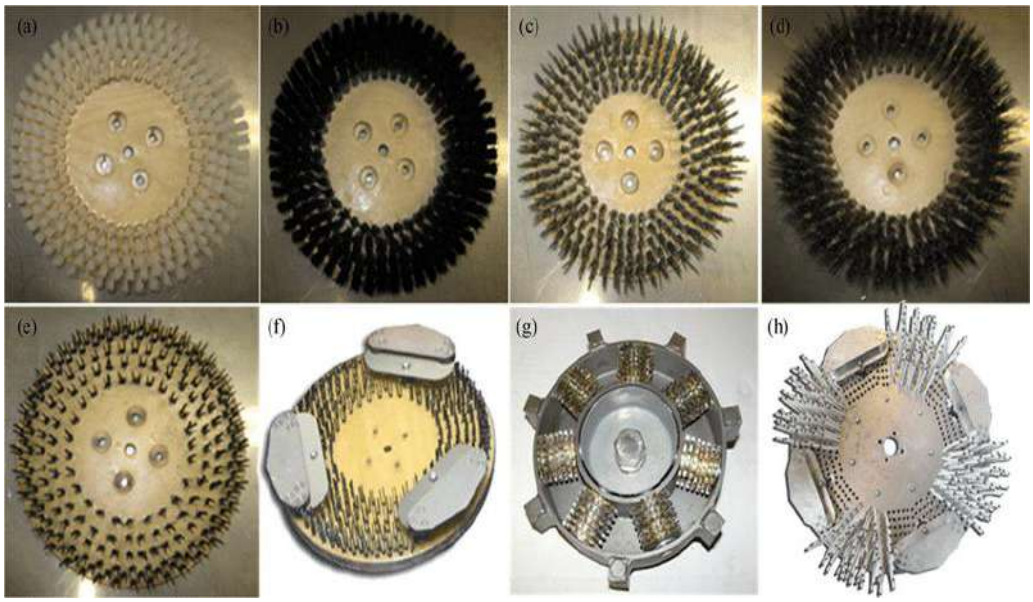
Dalam konteks riset saya mengenai ROV hull cleaning berbasis motor brushless M200, ESC berfungsi krusial sebagai aktuator yang menerjemahkan keluaran kendali PID (dalam bentuk duty cycle PWM) menjadi tegangan tiga fasa ke motor, sehingga torsi dan rpm brusher dapat disesuaikan terhadap perubahan beban sikat dan kelas biofouling yang dihasilkan YOLO, dengan tetap mempertahankan kehalusan respon dan kestabilan gerak sistem seperti yang ditekankan dalam hasil pengujian ROV underwater drone pada jurnal ini.

2.3.6 Brusher

Metode pembersihan berbasis bulu memanfaatkan gaya mekanik untuk membersihkan hull kapal melalui kontak langsung bulu sikat dengan lapisan biofouling dan, kadang-kadang, cat antifouling di bawahnya. Sistem ini biasanya menggunakan brush rotary yang ditekan ke permukaan lambung dengan gaya normal. Pada saat yang sama, gerakan cart yang diputar dan ditranslasi menghasilkan gaya geser yang mengikis biofouling dari substrat.

Pengikis non-abrasif dan pengikis abrasif adalah dua kategori utama pengikis dalam laporan ini. Pengikis non-abrasif menggunakan sikat lembut seperti nylon atau polypropylene untuk menghilangkan makrofouling dan slime dengan adhesi lemah sampai dengan sedang, serta lapisan cat yang sudah terkelupas, tanpa mengikis cat yang masih utuh secara signifikan, perbedaan fisik kedua jenis dapat dilihat pada gambar 2.9. Metode pembersihan proaktif yang melibatkan pembersihan ringan yang dilakukan secara berkala pada tingkat fouling rendah untuk menghindari fouling berat dianggap efektif untuk grooming proaktif. Berdasarkan literatur, pengikis abrasif (*steel brushes*) dapat digunakan untuk

membersihkan barnacle, heavy grass, dan organisme berkerang kecil, sehingga sesuai dengan kondisi biofouling campuran (*slime*, rumput laut, dan *barnacle*) pada lambung kapal yang menjadi objek pengujian (Song & Cui, 2021). Ini dapat mengangkat lapisan cat dan bahkan menyebabkan area lapisan yang sudah rusak menjadi laminasi. Kedalaman pengikisan sangat dipengaruhi oleh kekakuan bulu, tekanan sikat, dan kecepatan gerak cart.



Gambar 2. 9 *Brushes Manufactured*

(sumber: https://www.researchgate.net/figure/Brushes-manufactured-by-Armada-Systems-Inc-a-nylon-brush-b-polypropylene-brush-c_fig4_347152491)

Dalam kegiatan pembersihan lambung kapal dari biofouling, peneliti memilih stainless steel row brush sebagai tipe sikat utama karena karakteristiknya yang bersifat "*medium duty*". Pemilihan *steel brush* ini sangat relevan dengan medan pengujian, mengingat pelat lambung yang dibersihkan berbahan dasar metal (baja/aluminium), sehingga membutuhkan sikat berbahan logam yang mampu mentransfer gaya gesek secara efektif namun tetap terkontrol terhadap permukaan pelapis antifouling.

Dengan demikian, stainless steel row brush dipandang tidak terlalu agresif seperti flat wire steel brush tetapi juga tidak terlalu lembut seperti nylon/polypropylene brush. Dengan cara ini, kombinasi jenis brush, tekanan, dan

kecepatan putar dapat disesuaikan dengan kontrol PID dan PWM ESC untuk menyeimbangkan pembersihan dan perlindungan cat.

2.3.7 Baterai

White paper Jauch menjelaskan bahwa prinsip kerja dan konstruksi Li-Po pada dasarnya sama dengan Li-ion, namun bentuk pouch memberikan fleksibilitas ukuran dan bentuk yang jauh lebih besar, berkemasan foil aluminium berlapis plastik yang tipis dan fleksibel, sehingga memungkinkan desain baterai yang sangat rata (bahkan <1 mm) dengan densitas energi hingga sekitar 200 Wh/kg. Dokumen tersebut menekankan beberapa keunggulan utama LiPo dibanding kemasan Li-ion silindris, yaitu fleksibilitas dimensi (bentuk prismatic, curved, ultra-thin), bobot yang lebih ringan karena tidak menggunakan housing logam kaku, serta laju self-discharge yang rendah sehingga cocok untuk perangkat mobile jangka panjang. Namun, pengoperasian LiPo harus selalu disertai rangkaian proteksi (PCM) untuk mencegah overcharge, over-discharge, dan arus lebih, serta desain kompartemen baterai yang memperhitungkan kemungkinan pemuaian sel sampai sekitar 10% dan perlindungan mekanik terhadap kerusakan kemasan foil (Jauch Battery Solutions, 2018).

Gambar 2. 10 Baterai Li- Po 7200 mAh 50C

(sumber: Dokumen Pribadi)

Pada tabel 2.5 merupakan spesifikasi dari baterai yang digunakan acuan peneliti dalam pembuatan sistem brushing bagaimana *supply* daya.

Tabel 2. 5 Spesifikasi Baterai

Spesifikasi	**Keterangan**
Capacity	7200 mAh
Voltage	14.8 V/ 4 S
Continuos Discharge Current	50 C (110 A)
Burst Discharge Current	100 C (620 A)
Balance Ping	JST-XH
Connector	XT60
Type Battery	*Lithium Polymer*
Dimensions	160 x 49 x 29 mm
Weight	610 g

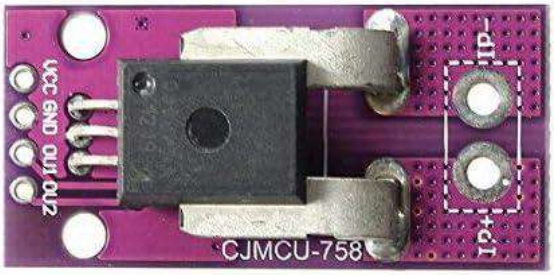
Dalam konteks riset yang sedang dikembangkan, peneliti memposisikan baterai LiPo sebagai sumber daya utama karena bentuk pouch yang tipis dan fleksibel memungkinkan integrasi yang lebih mudah ke dalam ruang terbatas pada bodi ROV tanpa menambah massa berlebih. Berdasarkan white paper Jauch, peneliti menyimpulkan bahwa tegangan sel sekitar 3,6–3,7 V per sel dan densitas energi hingga kurang lebih 200 Wh/kg memberikan kombinasi yang sesuai antara kerapatan energi dan bobot untuk menopang kebutuhan arus tinggi motor brushless dan sistem elektronika kendali. Selain itu, peneliti menjadikan rekomendasi bahwa setiap paket LiPo harus dilengkapi rangkaian proteksi overcharge, over-discharge, dan over-current sebagai acuan dalam merancang sistem manajemen baterai, sekaligus mempertimbangkan ruang ekspansi sekitar 10% di kompartemen baterai untuk mengakomodasi pemuaian sel selama siklus kerja.

2.3.8 ACS758

Berdasarkan dokumentasi Texas Instruments, ACS758 merupakan current shunt and power monitor yang mampu melakukan pengukuran tegangan shunt dan tegangan bus secara bersamaan dengan resolusi tinggi. Kemampuan tersebut memungkinkan perhitungan arus dan daya secara langsung sehingga sesuai digunakan pada aplikasi pemantauan konsumsi energi sistem motor listrik dan perangkat elektronika daya. (Guang Zhou, 2019). Selain itu penggunaan ACS758pada ROV hull cleaning juga didukung oleh peneliti lain yang tengah mengembangkan sistem ROV hull cleaning, ACS758 dipandang relevan sebagai komponen dasar teori karena mampu menyediakan umpan balik arus motor

brushless secara real-time, yang sangat berguna untuk pemantauan beban sikat dan implementasi strategi proteksi ketika terjadi kondisi overcurrent pada sistem brushing (Angga dkk., 2021).

.



Gambar 2. 11 Sensor ASC-758

(sumber: https://ielectrony.com/en/product/cjmcu-758-acs758-linear-hole-current-sensor-module-for-arduino)

Tabel 2. 6 Spesifikasi ACS758

Characteristic	Symbol	Rating	Units
Supply Voltage	Vcc	8	V
Reverse Supply Voltage	Vrcc	-0,1	V
Output Voltage	Viout	8	V
Reverse Output Voltage	Vriout	-0,1	V
Output Current Source	I (source)	3	mA
Output Current Sink	I (Sink)	10	mA
Overcurrent Transient Tolerance	Ip	100	A
Nominal Operating Ambient Temperature	T	─40 to 85	˚C
Maximum Junction temperature	Tj (Max)	165	˚C
Storage Temperature	Tstg	─65 to 170	˚C

Dalam konteks riset saya tentang ROV hull cleaning, peneliti dapat mengadaptasi konsep dual-loop PID tersebut dengan menempatkan satu loop PID untuk menjaga Torsi yang akan mempengaruhi kecepatan putar motor brusher saat berbeban, dan loop lainnya untuk mengoreksi beban berlanjut sehingga adaptasi torsi akan berjalan maksimal menggunakan IMU dan kamera, sehingga sistem tetap stabil dan akurat walaupun konfigurasi sensor dibatasi oleh kondisi bawah air dan keterbatasan hardware.

2.3.9 Pixhawk

Pixhawk merupakan *flight controller* atau unit pengendali utama yang banyak digunakan pada sistem kendaraan tanpa awak, termasuk *Remotely Operated Vehicle* (ROV), yang berfungsi sebagai pusat pengolahan data dan pengendalian sistem secara terintegrasi. Perangkat ini mengatur kerja berbagai sensor, aktuator, dan modul komunikasi sehingga dapat melakukan kendali, stabilisasi, dan pemantauan sistem dalam waktu nyata.

Gambar 2. 12 Pixhawk PX 4

(sumber: https://docs.px4.io/main/en/flight_controller/pixhawk4)

Prosesor tertanam berperforma tinggi, sensor inersia internal, dan berbagai antarmuka input-output seperti UART, I2C, CAN, dan PWM membuat Pixhawk lebih mudah diintegrasikan dengan sensor eksternal, motor, pengatur kecepatan elektronik (ESC), dan perangkat pendukung lainnya. Selain itu, Pixhawk menggunakan firmware *open-source* seperti ArduPilot atau PX4 yang mendukung sistem operasi waktu nyata dan protokol komunikasi MAVLink untuk pertukaran data telemetri, status sistem, serta perintah kendali antara *flight controller*, komputer pendamping, dan stasiun kendali darat. Pixhawk menjadi platform pengendali yang populer dalam penelitian dan pengembangan sistem ROV karena fleksibilitas, keandalan, dan dukungan komunitas yang luas.

Pada tabel 2.7 merupakan spesifikasi dari underwater ultrasonic yang digunakan acuan peneliti dalam pembuatan sistem brushing bagaimana menjadi input data jarak.

Tabel 2. 7 Spesifikasi *Underwater Ultrasonic*

Parameter	Spesifikasi Utama
Platform	FMUv5, menjalankan PX4 di atas NuttX RTOS
Main Processor	STM32F765, ARM Cortex-M7 32-bit, 216 MHz, 2 MB Flash, 512 KB RAM
Sensor IMU	ICM-20689 + BMI055/ICM20602 (Accel/Gyro ganda untuk redundansi)
Magnetometer	IST8310 (onboard dan pada modul GPS Neo-M8N)
Barometer	MS5611 untuk pengukuran tekanan/ketinggian/kedalaman relatif
PWM Output	8–16 kanal PWM (8 dari IO, 8 dari FMU) untuk motor dan lampu LED
Port Serial	5 port serial (GPS, TELEM1, TELEM2, TELEM4, Debug console)
Antarmuka I2C/SPI	3 port I2C dan 4 bus SPI untuk sensor tambahan
Input Power	POWER1/POWER2: 4.9–5.5 V; USB: 4.75–5.25 V

Produk komersial seperti underwater ultrasonic obstacle-avoidance sensor IP68 dirancang khusus untuk robot bawah air dengan jangkauan tipikal 5–600 cm, blind zone kecil (sekitar beberapa sentimeter), sudut bukaan sempit 5–16°, dan antarmuka komunikasi digital (UART/RS-485) sehingga dapat diintegrasikan langsung dengan mikrokontroler atau komputer ROV. Sensor ini mampu bekerja hingga kedalaman sekitar 10 m dan dimaksudkan menggantikan atau melengkapi kamera untuk tugas penghindaran halangan serta pengukuran kedalaman lokal. Literatur mengenai pengukuran kedalaman bawah laut dengan ultrasonik menegaskan bahwa akurasi sangat bergantung pada kalibrasi kecepatan suara dan geometri pemasangan transduser terhadap permukaan air atau objek.

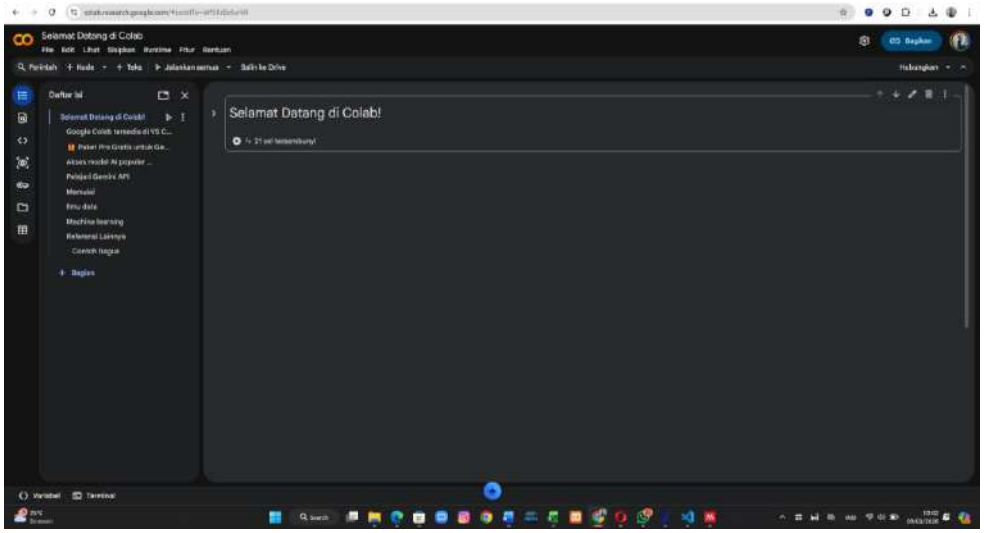
Dalam konteks riset ROV hull cleaning, informasi ini dapat dijadikan dasar teori bahwa sensor ultrasonik bawah air layak digunakan sebagai feedback jarak antara brusher dan permukaan lambung kapal maupun sebagai sensor penghindar halangan, di mana data jarak real-time kemudian diproses oleh pengendali (misalnya PID pada mikrokontroler/ESP32) untuk menjaga gap sikat tetap konstan dan mencegah tabrakan struktur ROV dengan hull.

2.3.10 Google Colab

Google Colab (Google Colaboratory) merupakan perangkat lunak berbasis komputasi awan yang menyediakan lingkungan pengembangan terintegrasi untuk menjalankan program Python secara langsung melalui peramban web. Platform ini

dikategorikan sebagai application software sekaligus software as a service (SaaS) karena seluruh proses komputasi, mulai dari eksekusi kode hingga pengelolaan sumber daya, dilakukan pada server milik Google tanpa memerlukan instalasi perangkat lunak tambahan di sisi pengguna. Google Colab banyak dimanfaatkan dalam bidang penelitian, pendidikan, dan pengembangan sistem cerdas karena kemudahan akses, fleksibilitas, serta dukungan terhadap berbagai pustaka ilmiah dan pembelajaran mesin.

Dari sisi fungsional, Google Colab menyediakan notebook interaktif berbasis Jupyter yang memungkinkan integrasi kode program, teks penjelasan, visualisasi data, dan hasil eksekusi dalam satu dokumen. Platform ini menyediakan sumber daya komputasi berupa CPU, GPU, dan TPU yang mendukung pemrosesan data, pelatihan model *deep learning*, serta pengolahan citra dan video. Selain itu, Google Colab terintegrasi dengan Google Drive sehingga memudahkan penyimpanan, pengelolaan, dan akses data maupun kode secara daring. Fitur kolaborasi juga memungkinkan beberapa pengguna mengakses dan mengembangkan notebook secara bersama-sama, sehingga mendukung proses penelitian yang membutuhkan pengujian, dokumentasi, dan pengembangan program secara fleksibel.



Gambar 2. 13 Tampilan *Software* Google Colab

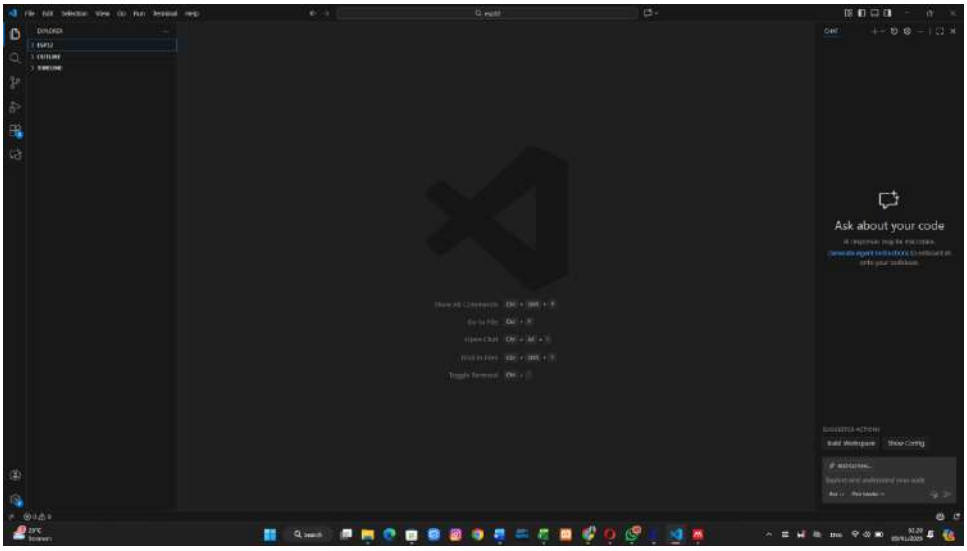
(sumber: Dokumen Pribadi)

Kemampuan menjalankan eksperimen tanpa ketergantungan pada spesifikasi perangkat keras lokal menjadikan platform ini relevan untuk

implementasi metode komputasi modern, termasuk pemrosesan citra, kecerdasan buatan, dan analisis data teknis.

2.3.11 Visual Studio

Visual Studio adalah alat pengembangan aplikasi yang berjalan di atas sistem .NET Framework, menggunakan bahasa pemrograman seperti BASIC, C, C++, dan C#. Dengan alat ini, para programmer dapat membuat aplikasi Windows Form, aplikasi web berbasis ASP.NET, serta aplikasi command-line.



Gambar 2. 14 Tampilan *software VSCode*

(sumber: Dokumen Pribadi)

Dalam penelitian ini, Visual Studio, seperti yang terlihat pada Gambar 2.14, digunakan untuk membuat aplikasi desktop yang digunakan untuk memonitoring pergerakan ROV. Perangkat lunak ini mengadopsi paradigma Pemrograman berorientasi objek, yang dapat dianggap sebagai evolusi dari versi Microsoft Visual Basic sebelumnya, yang diimplementasikan di atas .NET Framework.

2.3.12 QGround Control

QGroundControl adalah ground control station (GCS) lintas-platform yang menyediakan kemampuan penuh untuk konfigurasi, pemantauan, dan kendali wahana tanpa awak berbasis PX4 maupun ArduPilot melalui protokol MAVLink. Perangkat lunak ini dirancang agar cukup sederhana bagi pemula tetapi tetap mendukung fitur tingkat lanjut seperti perencanaan misi otonom, survei area, dan prosedur pendaratan aman bagi pengguna berpengalaman. Antarmukanya dibagi

menjadi beberapa tampilan utama, antara lain Fly View untuk kendali dan pemantauan penerbangan secara real-time, Plan View untuk membuat dan mengedit rencana misi berbasis waypoint di atas peta, Analyze Tools untuk melihat log dan data penerbangan, serta Setup View untuk konfigurasi detail parameter autopilot, sensor, frame, dan kalibrasi sistem.

Gambar 2. 15 Tampilan *Software QGround Control*

(sumber: Dokumen Pribadi)

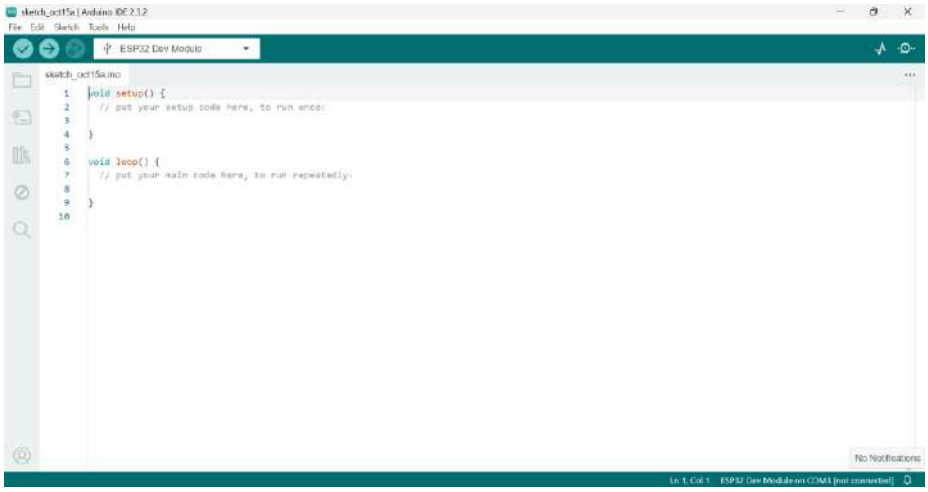
Sebagai GCS modern, QGroundControl mendukung koneksi ke kendaraan melalui USB, telemetry radio, maupun Wi-Fi/IP, dan secara otomatis mendeteksi serta melakukan handshake dengan flight controller yang kompatibel. Peta pada tampilan utama menampilkan posisi kendaraan, jejak lintasan, waypoint, serta instrumen penerbangan sehingga operator dapat memantau sikap, ketinggian, kecepatan, dan status baterai secara terpadu. Selain itu, QGroundControl juga menyediakan dukungan first person view (FPV) dengan video streaming dan overlay instrumen, kemampuan mengelola banyak kendaraan sekaligus, serta menjalankan fungsi analisis misi seperti estimasi waktu tempuh dan jarak survei.

Dokumentasi resmi versi 5.0 menekankan bahwa QGroundControl tidak hanya ditujukan bagi pengguna akhir, tetapi juga bagi pengembang yang ingin mengubah atau memperluas fungsionalitasnya. Antarmuka pengguna dibangun dengan Qt/QML sehingga dapat berjalan di Windows, macOS, Linux, Android, dan iOS, serta memudahkan pengembangan UI yang responsif di perangkat layar sentuh

maupun desktop. Bagi pengembang, panduan khusus disediakan untuk proses build dari kode sumber, aktivasi fitur opsional seperti video streaming via GStreamer, serta integrasi pesan MAVLink kustom. Dengan kombinasi fitur konfigurasi autopilot, perencanaan misi, pemantauan telemetri, dan dukungan pengembangan terbuka, QGroundControl dapat dijadikan dasar teori utama ketika menjelaskan subsistem ground control dan antarmuka operator pada tugas akhir yang melibatkan kendali wahana MAVLink, termasuk ROV atau drone berbasis PX4/ArduPilot.

2.3.13 ArduinoIDE

Arduino IDE adalah *software* yang digunakan untuk membuat suatu program kemudian mengompilasi serta mengunggah program ke dalam memori mikrokontroler. *Software* ini memiliki 3 hal utama. Pertama, editor program atau *sketch* berfungsi untuk membuat pemrograman dengan Bahasa C++. Kedua, *compiler*, mengubah hasil pemrograman yang ada pada *sketch* untuk diproses menjadi biner sehingga mikrokontroler dapat membaca hasil program yang ada pada *sketch*. Ketiga, *upload*, mengirim pemrograman yang telah dibuat dan diubah menjadi kode biner ke dalam mikrokontroler, sehingga memori pada mikrokontroler ber-isikan program yang telah dibuat pada Arduino IDE (Triyatna & Ardiansyah, dkk). ArduinoIDE ini juga memungkinkan pengembangan yang cepat untuk prototipe, lokakarya, atau proyek dadakan tanpa perlu konfigurasi yang rumit.

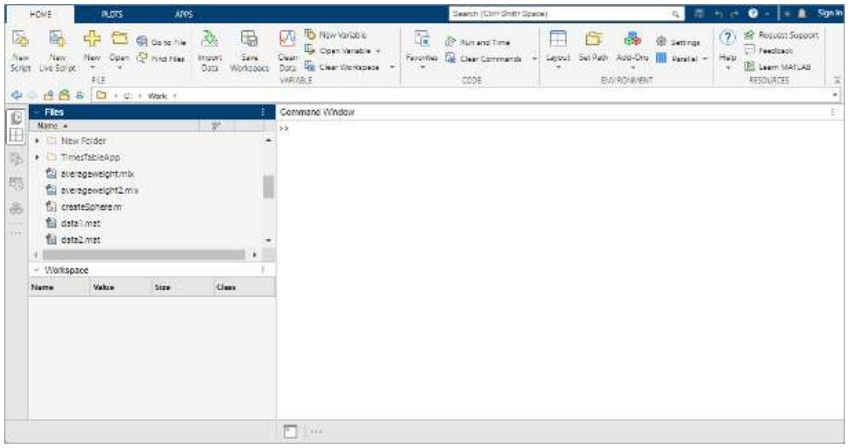


Gambar 2. 16 *Software ArduinoIDE*

(sumber: Dokumen Pribadi, 2025)

2.3.14 Matlab

Matlab adalah perangkat lunak untuk pemrograman, perhitungan, dan analisis yang banyak diterapkan dalam berbagai bidang matematika, baik di sektor pendidikan maupun dalam penelitian di universitas dan *industry*. Matlab adalah lingkungan pemprogaman dan Bahasa komputasi yang digunakan secara lusa dalam berbagai bidang pengetahuan, Teknik, matematika, dan industry. Nama "Matlab" sendiri merupakan singkatan dari "*Matrix Laboratory*", yang menggambarkan salah satu kekuatan utama Bahasa pemprogaman yang idrancang untuk melakukan berbagai tugas komputasi numerik dan analisi data.



Gambar 2. 17 Tampilan *Software Matlab*

(sumber: Dokumen Pribadi)

Matlab mendukung pembuatan grafik dan visualisasi data dengan mudah, baik dalam bentuk 2D maupun 3D. Selain itu, perangkat lunak ini memiliki fungsi matematika yang luas, memungkinkan pembuatan visualisasi data yang interaktif. Matlab juga mendukung pemrograman berbasis skrip, yang memungkinkan pengguna untuk mengembangkan program Matlab mereka sendiri. Matlab dapat digunakan untuk mengotomatiskan tugas, pengembangan aplikasi khusus, dan integrasi dengan perangkat keras serta perangkat lunak lainnya. Dengan komunitas yang besar dan aktif, Matlab menyediakan sumber daya online, tutorial, forum, serta dokumentasi untuk membantu pengguna memahami dan menggunakan perangkat lunak ini secara lebih efektif.

Dalam tugas akhir ini, Matlab digunakan untuk menghitung rumus yang didapatkan dari metode kontrol PID untuk mengatur kecepatan *brusher*.

BAB 3
METODE PENELITIAN

Metode penelitian harus mengacu pada tujuan penelitian yang telah ditetapkan. Peneliti harus menjelaskan metode yang akan digunakan untuk mencapai tujuan penelitian dan mengapa sebuah metode penelitian lebih tepat digunakan dibanding metode yang lain.Bagian ini berisi tentang alur penelitian dari Tugas Akhir. Pembahasan pada bab ini adalah mengidentifikasi masalah, studi literatur, analisis kebutuhan sistem, perancangan hardware dan software, implementasi dan pengujian prototipe serta analisis dan pembahasan.

3.1 Konsep Penelitian

Metode penelitian merupakan tahapan pendekatan yang digunakan untuk menyelesaikan permasalahan dalam sistem pembersihan lambung kapal berbasis Remotely Operated Vehicle (ROV). Penelitian ini berfokus pada pengembangan sistem pembersihan yang mampu menyesuaikan kecepatan motor brush berdasarkan tingkat biofouling yang terdeteksi pada permukaan lambung kapal. Sistem memanfaatkan integrasi kamera, metode deteksi dan klasifikasi YOLO (You Only Look Once), metode Voting, sensor Inertial Measurement Unit (IMU), serta sistem kendali PID untuk menghasilkan proses pembersihan yang lebih terarah dan terkendali. Integrasi beberapa subsistem tersebut dirancang agar proses pembersihan tidak hanya bergantung pada kemampuan operator dalam menentukan kondisi permukaan lambung, tetapi dapat memberikan keputusan berdasarkan data visual dan parameter gerak yang diperoleh secara langsung selama ROV beroperasi.

Proses sistem diawali dengan pengambilan citra permukaan lambung kapal menggunakan kamera yang terpasang pada ROV. Citra tersebut kemudian diproses menggunakan metode YOLO untuk mendeteksi keberadaan biofouling dan mengklasifikasikannya ke dalam tiga tingkat pembersihan, yaitu Light, Medium, dan Heavy. Klasifikasi tersebut digunakan untuk menentukan tingkat kebutuhan pembersihan berdasarkan kondisi biofouling yang terdapat pada permukaan lambung kapal. Dengan demikian, sistem tidak menerapkan satu kecepatan

pembersihan untuk seluruh kondisi permukaan, tetapi menyesuaikan proses pembersihan berdasarkan hasil deteksi visual. Proses deteksi dilakukan secara berulang pada citra yang diperoleh selama ROV bergerak menyusuri permukaan lambung, sehingga perubahan kondisi biofouling pada area yang diamati dapat diketahui secara lebih representatif.

Hasil deteksi YOLO selanjutnya tidak langsung digunakan sebagai keputusan akhir, tetapi dikumpulkan dalam sejumlah data deteksi dan diproses menggunakan metode Voting. Metode ini digunakan untuk menentukan kelas dominan dari beberapa hasil deteksi yang diperoleh selama ROV melakukan pengamatan pada suatu area. Penerapan Voting bertujuan untuk mengurangi pengaruh kesalahan klasifikasi pada satu atau beberapa frame akibat kondisi citra bawah air yang kurang optimal, seperti kekeruhan, pencahayaan yang tidak merata, gerakan ROV, maupun perubahan posisi kamera. Hasil Voting kemudian digunakan sebagai keputusan tingkat pembersihan, yaitu Light, Medium, atau Heavy. Dengan mekanisme tersebut, keputusan sistem tidak hanya didasarkan pada satu hasil deteksi, tetapi mempertimbangkan kecenderungan kelas dari sejumlah data yang masuk sehingga keputusan akhir diharapkan lebih stabil.

Setelah tingkat pembersihan ditentukan, sistem mengatur kecepatan putaran motor BLDC brush menggunakan metode PID (Proportional-Integral-Derivative). Pengendalian PID digunakan untuk mempertahankan kecepatan motor agar mendekati nilai setpoint yang telah ditentukan sesuai dengan tingkat biofouling. Pada tingkat Light, sistem dapat menggunakan kecepatan pembersihan yang lebih rendah, sedangkan pada tingkat Medium dan Heavy, sistem menggunakan kecepatan yang lebih tinggi sesuai dengan kebutuhan proses pembersihan. Pengaturan tersebut bertujuan untuk menjaga efektivitas proses brushing sekaligus mengurangi kemungkinan terjadinya slip pada motor brush maupun gaya penggoresan yang berlebihan terhadap permukaan lambung kapal. Penggunaan PID juga memungkinkan motor mempertahankan kecepatan yang lebih stabil ketika terjadi perubahan beban akibat perbedaan karakteristik biofouling atau perubahan gaya kontak antara brush dan permukaan lambung.

Selain klasifikasi biofouling dan pengendalian kecepatan motor, sistem memanfaatkan sensor gyro dan accelerometer sebagai bagian dari pemantauan

kondisi pergerakan ROV selama proses pembersihan. Nilai pembacaan sensor digunakan untuk mengidentifikasi perubahan gerakan atau kondisi tertentu pada ROV ketika melakukan kontak dan proses brushing terhadap permukaan lambung kapal. Apabila nilai pembacaan gyro/accelerometer mencapai lebih dari 900, kondisi tersebut digunakan sebagai parameter pemicu dalam logika pengendalian sistem untuk merespons perubahan gerakan atau gangguan selama proses pembersihan. Dengan pendekatan ini, sistem tidak bergantung pada sensor sonar atau ultrasonik untuk menentukan kedekatan ROV dengan lambung kapal, tetapi memanfaatkan informasi gerak dari sensor IMU sebagai salah satu parameter pengawasan kondisi operasi. Parameter tersebut menjadi bagian penting dalam menjaga kestabilan ROV karena proses brushing dapat menghasilkan gaya reaksi yang memengaruhi orientasi dan pergerakan kendaraan.

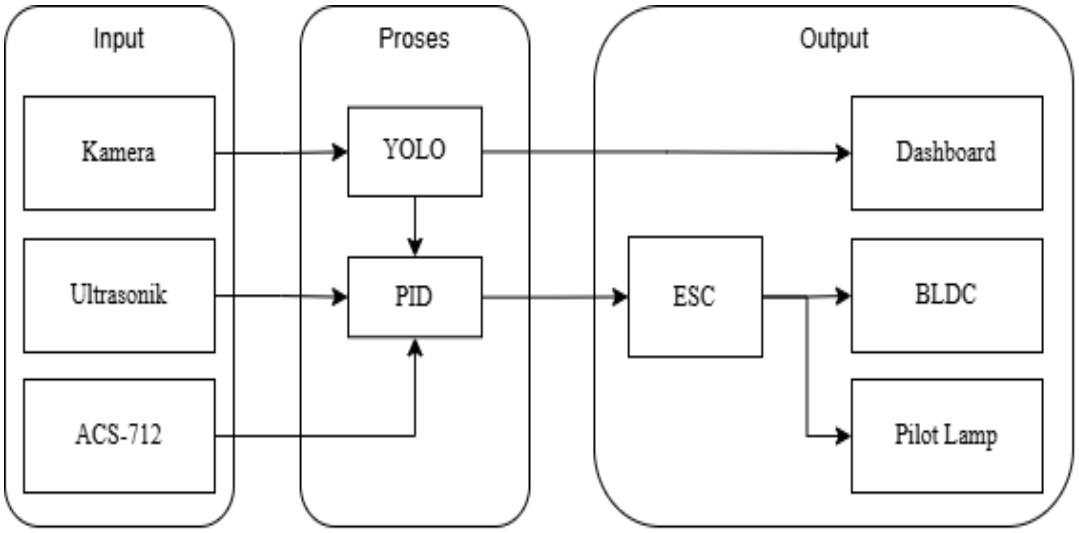
Secara keseluruhan, alur sistem dimulai dari pengambilan citra → deteksi dan klasifikasi YOLO → pengumpulan hasil deteksi → Voting → penentuan tingkat pembersihan → penentuan setpoint kecepatan → pengendalian PID motor BLDC → pemantauan kondisi ROV menggunakan gyro dan accelerometer. Integrasi tersebut memungkinkan sistem melakukan proses pembersihan berdasarkan kondisi biofouling yang terdeteksi sekaligus mempertahankan kestabilan operasi motor brush. Penggunaan PID memberikan pengendalian kecepatan yang lebih presisi sehingga dapat mengurangi risiko kegagalan proses pembersihan akibat perubahan beban pada motor ketika brush bersentuhan dengan permukaan lambung kapal. Sementara itu, data gyro dan accelerometer digunakan sebagai parameter tambahan untuk mengetahui perubahan kondisi gerak ROV selama sistem melakukan proses pembersihan.

Setelah sistem bekerja, proses operasi dan hasil pengendalian dapat dimonitor secara real-time melalui antarmuka sistem. Informasi yang dapat ditampilkan meliputi hasil klasifikasi biofouling, hasil Voting, tingkat pembersihan yang dipilih, setpoint dan RPM motor BLDC, nilai pembacaan gyro/accelerometer, serta status operasi sistem. Monitoring tersebut memungkinkan kondisi sistem diamati selama proses pembersihan sehingga respons terhadap perubahan kondisi

operasi dapat dilakukan secara langsung. Data yang diperoleh selama proses operasi juga dapat digunakan sebagai bahan evaluasi terhadap kinerja sistem, khususnya dalam melihat kesesuaian antara hasil klasifikasi biofouling, tingkat pembersihan yang dipilih, respons kecepatan motor, serta kondisi gerak ROV. Dengan demikian, sistem yang dikembangkan tidak hanya berfungsi untuk mendeteksi tingkat biofouling, tetapi juga mengintegrasikan hasil deteksi tersebut dengan sistem kendali motor untuk menghasilkan proses pembersihan lambung kapal yang adaptif, terarah, dan lebih terkendali.

3.1.1 Diagram Blok Sistem

Diagram ini memberikan pandangan yang terperinci tentang bagaimana komponen - komponen tersebut terhubung dan berinteraksi satu sama lain, memperlihatkan aliran informasi dan proses dalam sistem secara keseluruhan.



Gambar 3. 1 Diagram *Block system*

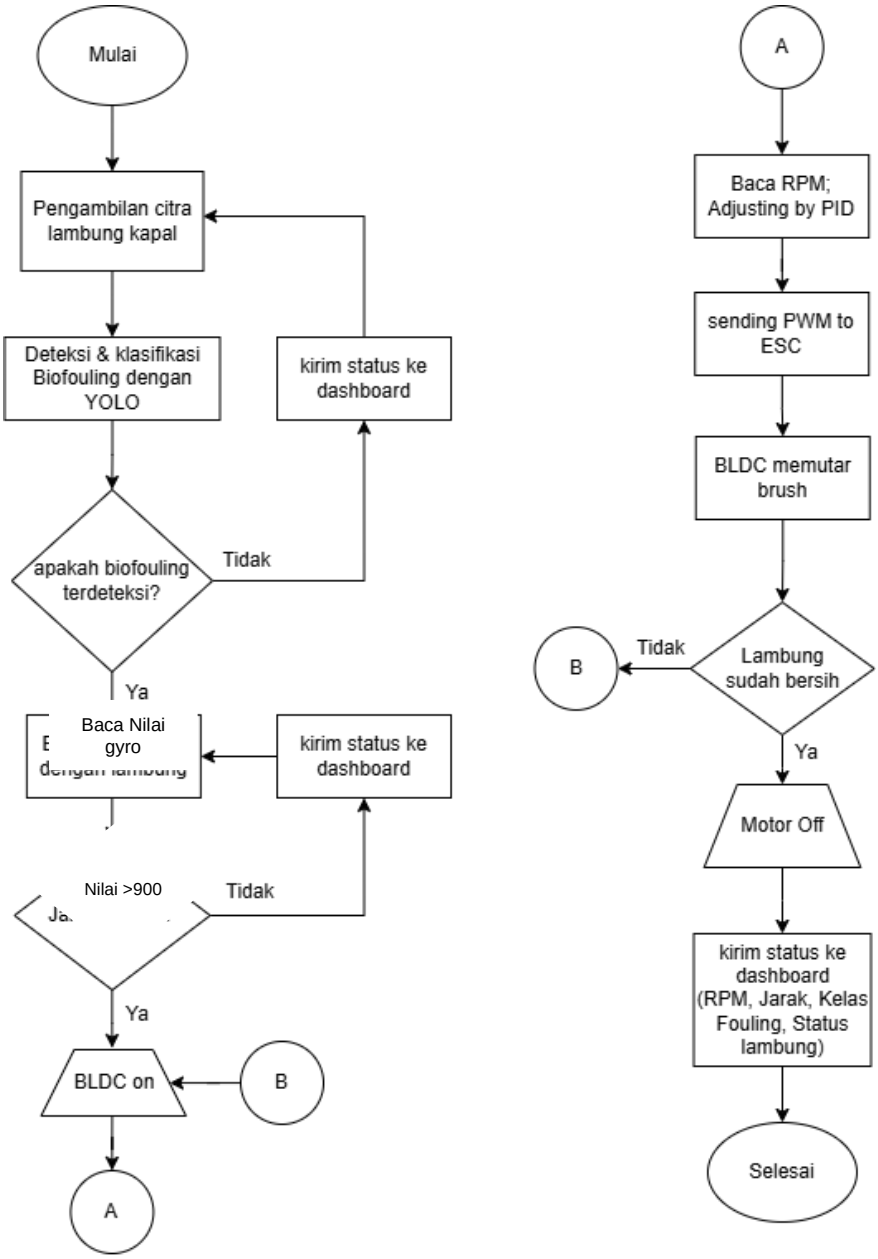
Blok diagram yang ditunjukkan memberikan gambaran alur kerja lengkap dari sistem kontrol pembersihan hull kapal berbasis PID. Sistem ini dimulai dengan pengambilan gambar permukaan hull menggunakan kamera multispektral, yang kemudian diproses menggunakan metode YOLO untuk mengklasifikasikan biofouling ke dalam tiga kategori, yaitu kecil, sedang, dan besar. Setelah itu, hasil klasifikasi tersebut akan menjadi input bagi sistem PID yang mengatur kecepatan perputaran motor BLDC. PID secara dinamis menyesuaikan RPM motor sesuai

dengan tingkat biofouling yang terdeteksi, memastikan pembersihan hull yang efisien dan optimal.

Selain itu, sensor ultrasonik digunakan untuk mengukur kedekatan motor dengan permukaan hull, memberikan umpan balik untuk menghindari kerusakan pada hull kapal. Semua data terkait pengoperasian sistem, termasuk kondisi biofouling dan status motor, dapat dipantau secara real-time melalui dashboard, memungkinkan pengguna untuk memonitor dan menyesuaikan sistem secara langsung. Dengan integrasi teknologi ini, sistem pembersihan hull kapal menjadi lebih cerdas, efisien, dan dapat beradaptasi dengan perubahan kondisi lingkungan secara otomatis, sehingga memastikan keberhasilan pencegahan biofouling pada kapal.

3.1.2 Flowchart Sistem

Implementasi proses merupakan tahapan penerapan solusi dan sistem yang telah dirancang secara konseptual ke dalam bentuk yang dapat dijalankan dan dioperasikan. Flowchart pada bagian ini menggambarkan alur kerja sistem kontrol motor BLDC untuk proses pembersihan hull kapal menggunakan metode PID, YOLO, dan sensor IMU. Proses diawali dengan pengambilan citra permukaan hull menggunakan kamera serta pembacaan data gerak ROV melalui sensor gyro dan accelerometer. Citra yang diperoleh kemudian diproses menggunakan YOLO untuk mendeteksi dan mengklasifikasikan tingkat biofouling menjadi tiga kelas, yaitu Light, Medium, dan Heavy. Hasil deteksi YOLO selanjutnya dikumpulkan selama 20 detik dan diproses menggunakan metode counter voting untuk menentukan kelas dominan yang menjadi dasar penetapan setpoint kecepatan brush. Setelah setpoint ditentukan, sistem membaca nilai IMU sebagai parameter validasi kondisi ROV. Apabila nilai pembacaan gyro dan accelerometer mencapai lebih dari 900, sistem menetapkan kondisi tersebut sebagai pemicu untuk memulai putaran motor brush. Selanjutnya, motor BLDC dikendalikan menggunakan metode PID dengan umpan balik sensor Hall untuk mempertahankan RPM aktual agar mendekati setpoint. Dalam implementasinya, sensor IMU tidak hanya berfungsi sebagai parameter pemantauan pergerakan ROV, tetapi juga sebagai mekanisme interlock untuk memastikan motor brush hanya mulai beroperasi ketika kondisi posisi dan gerakan ROV telah memenuhi kriteria yang ditentukan.



Gambar 3. 2 *Flowchart System*

Pada tahap pertama, sistem melakukan pengambilan citra lambung kapal sebagai input utama bagi algoritma deteksi. Citra yang diperoleh diproses pada blok deteksi dan klasifikasi biofouling dengan YOLO untuk mengidentifikasi apakah terdapat fouling pada area yang diamati. Pada langkah keputusan apakah biofouling terdeteksi?, jika tidak ada fouling maka sistem langsung mengirim status ke dashboard sebagai informasi bahwa area tersebut bersih, dan alur kembali ke proses pengambilan citra untuk pemantauan berkelanjutan. Jika YOLO mendeteksi

biofouling, sistem beralih ke blok baca jarak ROV dengan lambung menggunakan sensor ultrasonik untuk mengetahui posisi relatif brusher terhadap permukaan lambung.

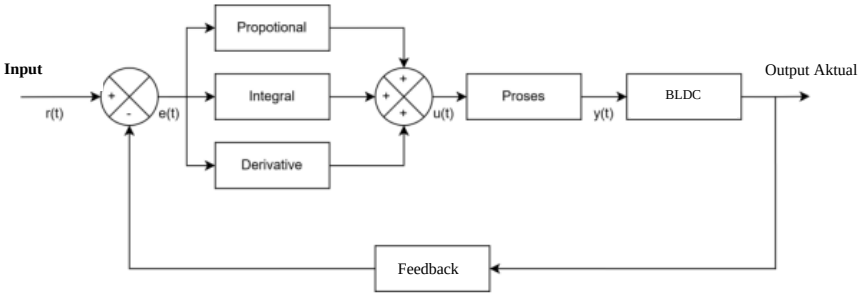
Nilai gyro dan accelerometer yang terbaca kemudian dievaluasi pada keputusan kondisi IMU > 900?. Jika nilai pembacaan masih kurang dari atau sama dengan 900, ROV belum memenuhi kondisi yang ditentukan sehingga sistem tetap melakukan pembacaan IMU hingga nilai melebihi ambang tersebut sehingga blok BLDC ON mengaktifkan motor brushless dan alur berpindah ke subbagian flowchart yang ditandai konektor A. Pada bagian kanan diagram, setelah konektor A, sistem melakukan pembacaan RPM dan penyesuaian menggunakan PID untuk mempertahankan kecepatan motor terhadap perubahan beban akibat kontak dan tekanan brush pada lambung. Keluaran PID berupa sinyal PWM yang dikirimkan ke ESC, kemudian ESC mengatur suplai daya ke motor pada blok BLDC sehingga brush berputar dan proses pembersihan biofouling dapat dilakukan. Dengan mekanisme ini, sensor IMU berfungsi sebagai parameter validasi kondisi operasi sekaligus interlock untuk mencegah brush berputar sebelum ROV memenuhi kondisi yang telah ditentukan.

Setelah brush berputar, flowchart masuk ke keputusan lambung sudah bersih? yang kembali didasarkan pada hasil pengamatan kamera dan YOLO; jika belum bersih, alur diarahkan ke konektor B yang mengembalikan proses ke bagian kiri untuk membaca ulang jarak dan melanjutkan pembersihan di area yang sama. Jika YOLO menyatakan lambung sudah bersih, blok Motor Off mematikan motor brush meskipun jarak masih di bawah 20 cm, kemudian sistem mengirim status ke dashboard yang memuat informasi RPM, jarak, kelas fouling, serta status kebersihan lambung sebagai umpan balik bagi operator. Setelah status dikirim, alur berakhir pada blok Selesai, atau dalam implementasi sebenarnya dapat kembali lagi ke tahap pengambilan citra untuk mencari area fouling berikutnya.

3.1.3 Perancangan Sistem Kontrol

Pada penelitian ini, implementasi sistem kontrol adalah untuk merancang dan menguji pengendali kecepatan motor brusher M200 agar putaran tetap stabil terhadap gangguan gaya sikat selama proses pembersihan lambung kapal melalui PID yang secara khusus dipakai untuk mengkompensasi perubahan beban (variasi

tingkat kotoran, tekanan sikat, perubahan kontak) sehingga rpm aktual mengikuti setpoint dari YOLO dengan error kecil dan settling time singkat. Sistem kontrol inspeksi ROV terdiri dari beberapa tahapan pemrosesan gambar yang saling terintegrasi.

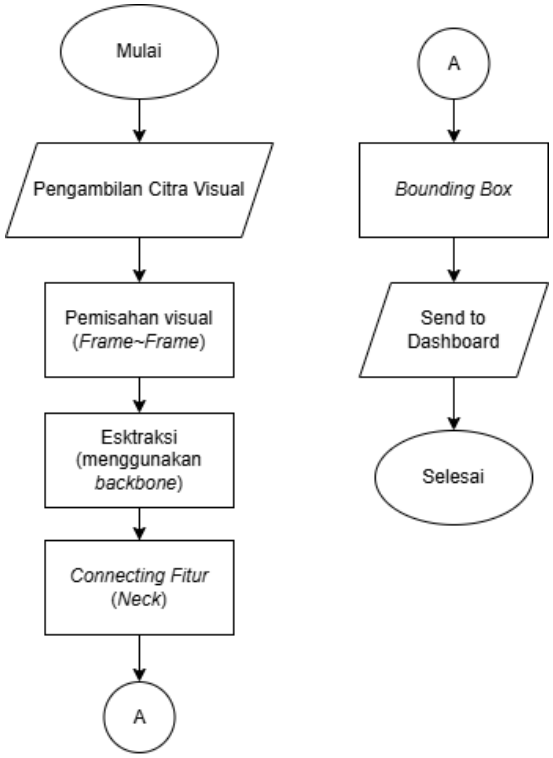


Gambar 3. 3 Diagram blok Kontrol *Classification Object*

Proses dimulai dengan menghitung nilai gambar yang diambil dari kamera bawah air yang mengawasi permukaan lambung kapal secara terus menerus. Setelah citra diakuisisi, format dan kualitas gambar diubah melalui tahap pra-proses dan penggabungan sinyal yang berfungsi. Ini dilakukan untuk memastikan bahwa masukan untuk modul pengolahan berbasis You Only Look Once (YOLO) sesuai. Pemrosesan YOLO adalah langkah berikutnya. Ini berfungsi untuk mengekstrak karakteristik seperti tekstur, pola kotoran, dan indikasi kerusakan struktur dari gambar dan mengonversinya menjadi informasi untuk deteksi atau klasifikasi objek. Setelah itu, YOLO dikirim ke Raspberry Pi sebagai perangkat pengendali utama. Perangkat ini menginterpretasikan hasil analisis gambar dan menerjemahkannya menjadi instruksi untuk sistem inspeksi ROV, seperti mengidentifikasi area yang perlu diamati atau melakukan tindakan untuk membersihkan area tertentu. Dashboard web menampilkan semua data hasil pengolahan, termasuk gambar terklasifikasi, status ROV, dan parameter operasional. Operator dapat melihat proses inspeksi secara real-time melalui antarmuka berbasis web dan membuat keputusan atau mengubah pengaturan sistem apabila diperlukan. Dengan demikian sistem kontrol digunakan sebagaimana pada proses pengalihan fungsi serta penyesuaian kecepatan yang tertuning sebelum proses implementasi dan pengembalian respon.

3.1.4 Perancangan YOLO

Subbab ini membahas perancangan algoritma YOLO yang digunakan sebagai inti sistem deteksi visual biofouling, meliputi alur pengolahan citra dari akuisisi kamera hingga keluaran berupa informasi bounding box dan klasifikasi objek yang terintegrasi dengan dashboard pemantauan. Penggolongan kotoran hanya dilakukan pada lebar dan besar kotoran, bukan pada jenis biofouling dan ketebalan kotoran. Hal ini disebabkan karena YOLO pada penelitian ini mengklasifikasikan patch biofouling berdasarkan luas area tampak pada citra, yang kemudian dipetakan ke tiga tingkat kekotoran: rendah, sedang, dan tinggi. Setiap tingkat kekotoran tersebut dihubungkan dengan setpoint kecepatan motor yang berbeda, sehingga brusher dapat bekerja lebih agresif untuk area yang kotor dan lebih lembut pada area yang relatif bersih untuk menjaga coating dan efisiensi energi oleh PID sebagaimana dibahas pada subbab 3.1.3. Dengan demikian, mekanisme pembersihan tidak membedakan jenis biofouling secara biologis, tetapi hanya tingkat kekotoran berdasarkan luas area citra, yang langsung menentukan seberapa cepat brush berputar di titik tersebut.



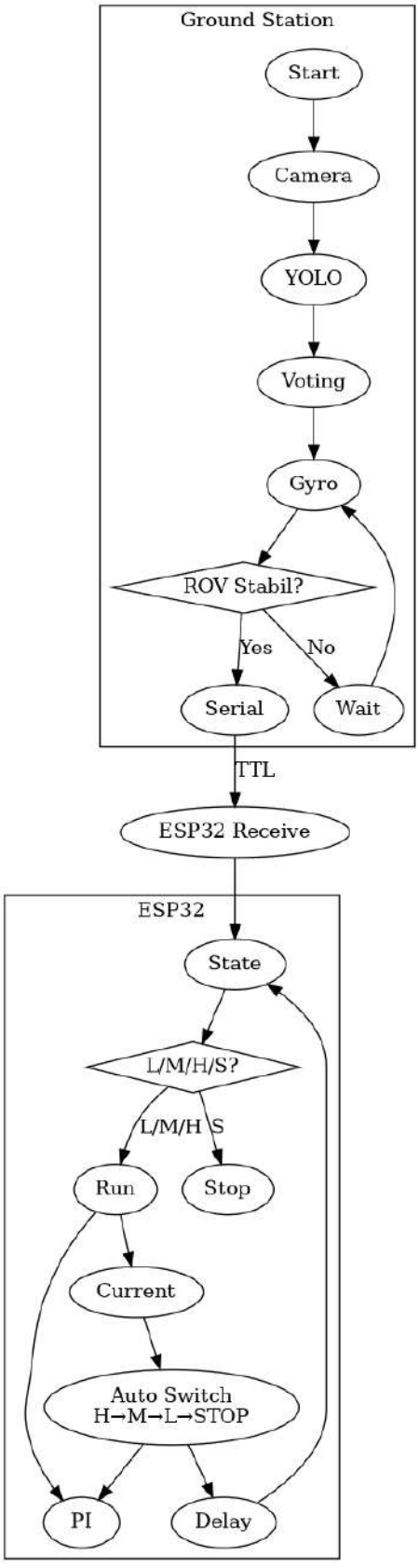
Gambar 3. 4 *Flowchart* YOLO

Proses dimulai dengan pengambilan gambar objek secara real-time menggunakan kamera. Selanjutnya, gambar diproses menjadi frame-per-frame untuk memungkinkan analisis terpisah untuk setiap frame. Untuk mendapatkan data visual penting dari gambar, backbone YOLOv8 digunakan untuk setiap frame berikutnya selama tahap ekstraksi fitur. Selanjutnya, fitur yang dibuat digabungkan pada bagian neck untuk mengintegrasikan informasi multi-skala. Hasil pengolahan fitur dilanjutkan ke tahap prediksi. Di sini, sistem mendeteksi objek dan lokasinya dalam kotak bounding, dan hasil prediksi ditampilkan pada frame yang telah diklasifikasikan. Setelah satu siklus pemrosesan selesai, proses diakhiri.

3.1.5 Perancangan Skema Pembersihan

Subbab ini membahas perancangan skema pembersihan yang digunakan sebagai inti sistem deteksi visual biofouling, meliputi proses akuisisi citra hingga keluaran berupa informasi setting RPM motor brush dan konfirmasi tingkat kebersihan. Flowchart proses pembersihan lambung kapal pada Gambar 3.5 menunjukkan alur kerja sistem yang dilakukan secara bertahap dan iteratif untuk memastikan proses pembersihan berlangsung efektif.

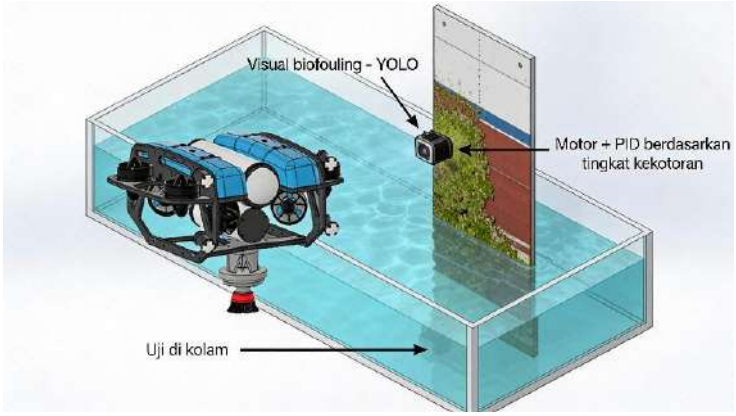
Proses diawali dengan Hull Scanning 1 untuk melakukan pemindaian awal permukaan lambung dan menentukan batas blok (cleaning block) yang akan dibersihkan. Selanjutnya dilakukan Hull Cleaning 1 menggunakan pengaturan RPM rendah sebagai tahap pembersihan awal. Penggunaan RPM rendah bertujuan untuk menghilangkan biofouling dengan tingkat adhesi ringan sekaligus mengurangi konsumsi energi dan potensi kerusakan lapisan antifouling. Setelah proses pembersihan dilakukan, sistem melakukan verifikasi terhadap batas blok. Apabila seluruh area belum tercapai, proses cleaning diulangi hingga batas blok terpenuhi. Setelah seluruh blok selesai dibersihkan, sistem melanjutkan ke Hull Scanning 2 untuk mengevaluasi hasil pembersihan menggunakan deteksi visual. Setelah itu, sistem kembali melakukan verifikasi hingga seluruh area dinyatakan bersih. Pendekatan bertingkat ini memungkinkan intensitas pembersihan disesuaikan dengan kondisi biofouling, sehingga proses menjadi lebih efisien, mengurangi penggunaan energi, serta meminimalkan risiko kerusakan lapisan antifouling pada lambung kapal.



Gambar 3. 5 *Flowchart* Skema Pembersihan

3.1.6 Perancangan Skema Pengujian

Skema pengujian dapat dilakukan di kolam uji dengan media pelat baja yang telah diberi biofouling alami atau buatan untuk mensimulasikan kondisi lambung kapal, sehingga algoritma YOLO tetap bekerja tanpa gangguan minyak dan kekeruhan ekstrem seperti di galangan atau laut terbuka.

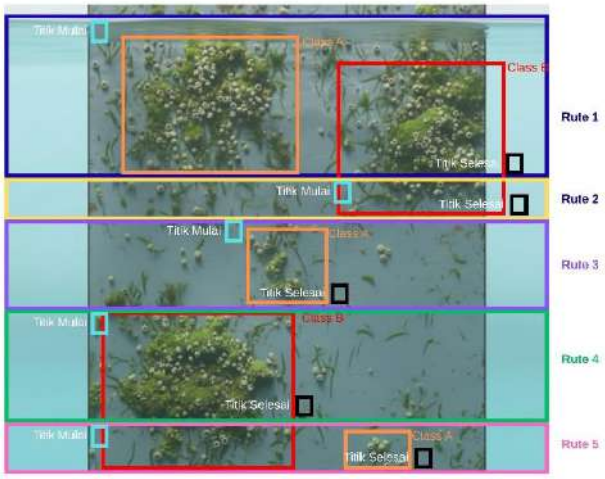


Gambar 3. 6 *Flowchart* Skema Pengujian

Rancangan dasarnya: robot/motor ditempatkan pada rig uji di depan pelat, kamera underwater menghadap pelat sebagai input YOLO dengan jarak ukur aktual sekitar 15 cm dari permukaan pelat, sensor ultrasonik mengukur jarak ke pelat agar jarak pandang dan sudut kamera konstan, sedangkan sensor arus ACS712 dihubungkan seri dengan motor sebagai umpan balik beban ke kendali PID.

Gambar 3. 7 Ilustrasi Kondisi Plat Uji

Gambar 3.7 merupakan ilustrasi kondisi plat yang akan diuji pada saat pengujian dengan tahapan pengujian: pertama, lakukan kalibrasi YOLO di kolam (lighting, white balance, dan threshold tingkat kekotoran) pada sampel pelat dengan tingkat fouling berbeda; kedua, jalankan mode otomatis dengan setpoint RPM 1 sebagai referensi untuk biofouling ringan dan amati respon PID (arus, RPM, dan output YOLO) hingga steady-state; ketiga, siapkan area pelat dengan fouling lebih tebal/keras, jalankan skenario di mana YOLO mengklasifikasikan tingkat kekotoran tinggi sehingga sistem berpindah ke setpoint RPM 2, dan catat dinamika transisi PID serta perubahan arus motor sebagai indikator kenaikan beban gesekan. Setiap skenario perlu direkam video dan data log (frame YOLO, nilai confidence, jarak ultrasonik, arus ACS712, PWM/RPM motor, serta waktu pembersihan per blok) untuk dianalisis terhadap dua indikator utama: akurasi keputusan pergantian RPM berbasis YOLO dan kestabilan kontrol PID (overshoot, rise time, dan error steady-state) pada masing-masing tingkat kekotoran dan konfigurasi blok sesuai struktur flowchart tahapan cleaning yang telah Anda rancang.



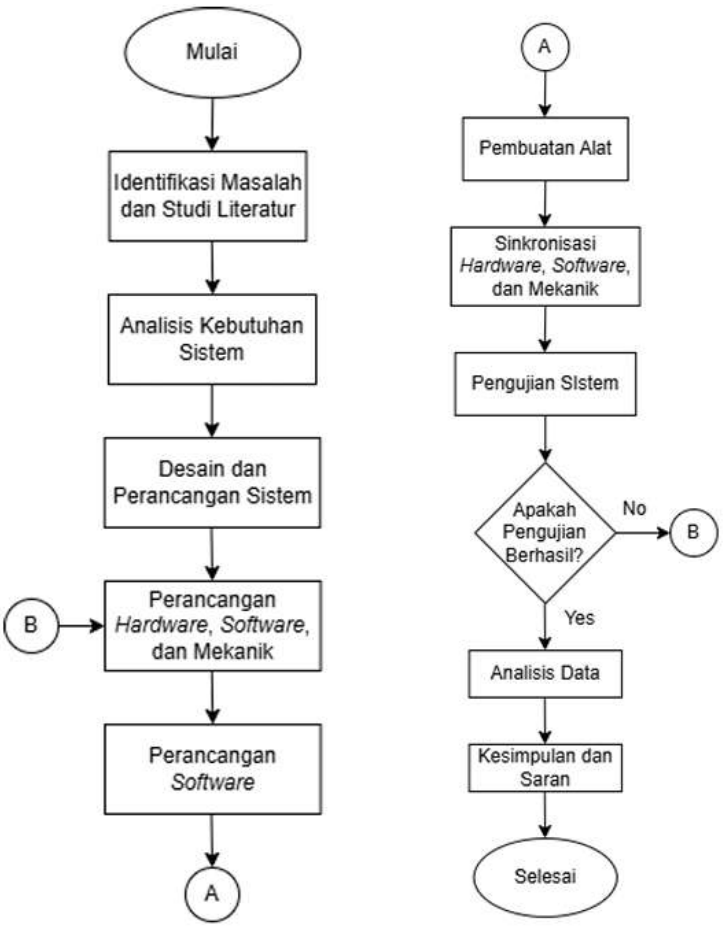
Gambar 3. 8 Ilustrasi Rute Uji Pembersihan

Gambar tersebut menunjukkan skema rute pembersihan permukaan pelat yang dibagi menjadi lima rute berbeda (Rute 1–Rute 5), di mana setiap rute direpresentasikan oleh kotak berwarna yang mengurung area biofouling dengan kepadatan dan sebaran yang bervariasi. Pada tiap rute terdapat penandaan “Titik

Mulai" dan "Titik Selesai" yang menentukan arah gerak ROV, serta kotak-kotak anotasi berlabel Class A dan Class B yang mengindikasikan kategori tingkat kekotoran atau jenis biofouling yang akan menjadi target prioritas pembersihan sesuai algoritma YOLO dan skema kontrol yang digunakan dalam penelitian ini.

3.2 Tahapan Penelitian

Penentuan alur penelitian bertujuan untuk mempermudah dalam memahami alur tindakan dan tahapan penelitian tahapan penelitian penyusunan. Diagram alir penelitian adalah representasi visual dari langkah langkah yang diambil dalam sebuah penelitian dari awal hingga akhir. Diagram alir penelitian membantu memvisualisasikan langkah- langkah ini dalam urutan yang logis, sehingga memudahkan pemahaman tentang bagaimana penelitian tersebut dilakukan dan bagaimana setiap tahapan berhubungan satu sama lain.



Gambar 3. 9 *Flowchart* pengerjaan penelitian

Flowchart pada Gambar 3.9 merupakan flowchart pengerjaan Tugas Akhir ini. Pengerjaan dimulai dengan mengidentifikasi masalah di lingkungan industri. Setelah masalah diidentifikasi, studi literatur dilakukan untuk memahami masalah dan menemukan solusi untuk masalah tersebut. Kemudian Analisis data yang diperlukan sebelum merancang sistem. Setelah perancangan sistem yang berisi metode yang digunakan dalam penelitian dibuat, perancangan hardware dan software sistem dibuat sesuai data yang telah disiapkan untuk menentukan spesifikasi hardware hingga model software yang akan digunakan. Langkah selanjutnya adalah integrasi antara hardware dan software dengan melakukan percobaan cara kerja sistem secara berurutan. Menentukan parameter-parameter yang digunakan seperti sensor-sensor yang digunakan. Kemudian membuat konsepan dan desain sistem alat yang akan dibuat. Kemudian pengadaan alat dan bahan yang akan dibuat, setelah itu melakukan pembuatan alat. Lalu alat dilakukan pengujian dan pengambilan data hasil kerja alat, dan setelah itu diberikan kesimpulan dari penelitian.

3.2.1 Identifikasi Masalah dan Studi Literatur

Proses identifikasi masalah dilakukan secara bersama dengan tinjauan literatur yang relevan, sehingga peneliti dapat memperoleh pemahaman yang lebih mendalalam tentang permasalahan yang ada sekaligus menggali berbagai potensi solusi yang dapat diterapkan. Berdasarkan hasil tinjauan literatur tersebut, peneliti telah menetapkan metode PID dan YOLO sebagai pendekatan yang digunakan untuk mendukung pengambilan keputusan dalam sistem proteksi kelistrikan pada ROV. Metode ini dipilih karena kemampuanya dalam memetakan berbagai kemungkinan solusi secara terstruktur dan logis, sehingga dapat memberikan hasil yang lebih efektif dan sesuai dengan kebutuhan sistem.

3.2.2 Analisis Kebutuhan Sistem

Proses analisis kebutuhan sistem dilakukan secara individu dengan tinjauan literatur yang sesuai, sehingga peneliti dapat memperoleh pemahaman yang lebih mendalalam tentang kebutuhan pada sistem dengan sebaik baiknya. Berdasarkan hasil analisis, peneliti telah menetapkan macam perangkat hardware dan software sistem permodelan kontrol pada ROV RoTi.

3.2.3 Desain dan Perancangan Sistem

Setelah proses analisis kebutuhan sistem selesai, langkah berikutnya adalah tahap desain dan perencanaan sistem. Tahap ini bertujuan untuk merancang integrasi komponen-komponen yang telah diidentifikasi sebelumnya, sehingga membentuk satu kesatuan yang fungsional. Desain dan Perencanaan ini berfungsi sebagai paduan utama dalam proses pembuatan dan implementasi sistem. Kegiatan pada tahap ini mencakup penyusunan desain sistem, baik dari sisi mekanik maupun kontrol, yang disesuaikan dengan kebutuhan dan spesifikasi yang telah ditentukan. Selain itu, tahap ini juga memastikan bahwa semua elemen sistem dirancang untuk bekerja secara optimal, sehingga mampu mencapai kinerja optimal sesuai tujuan yang diharapkan.

3.2.4 Pembuatan Alat

Tujuan pembuatan sistem adalah mengimplementasikan dan merealisasikan hasil perencanaan yang telah disusun pada tahap sebelumnya ke dalam bentuk nyata. Pada penelitian ini, proses pembuatan sistem mencakup pengembangan fungsi ROV untuk melakukan inspeksi sesuai dengan kebutuhan dan spesifikasi yang telah ditentukan. Tahap ini bertujuan memastikan setiap rancangan dapat direalisasikan menjadi sistem yang mampu bekerja secara optimal dan sesuai dengan tujuan penelitian. Selain itu, pembuatan sistem diarahkan untuk menghasilkan perangkat yang handal, stabil, dan memiliki ketahanan yang baik selama proses pengujian secara langsung, sehingga mampu beroperasi dalam berbagai kondisi yang sesuai dengan lingkungan pengujian.

3.2.5 Pengujian Sistem

Tahap pengujian sistem merupakan bagian penting dalam proses pengembangan yang bertujuan memastikan sistem berfungsi sesuai dengan spesifikasi yang telah dirancang. Pengujian dilakukan setelah tahap pembuatan sistem selesai dengan berfokus pada verifikasi kinerja dan keandalan sistem. Pada Tugas Akhir ini, pengujian dilakukan dengan mengaplikasikan sistem *brush* yang telah dikembangkan pada *Remotely Operated Vehicle* (ROV). Pengujian secara langsung dilakukan untuk mengevaluasi kinerja sistem dalam kondisi operasional serta memastikan fungsi pembersihan berjalan secara optimal.

3.2.6 Analisis Data

Setelah seluruh langkah, mulai dari analisis kebutuhan hingga pengujian sistem, diselesaikan dengan sukses dan sistem dipastikan berfungsi dengan baik, tahap selanjutnya adalah pengambilan data. Tujuan pengambilan data adalah untuk mendapatkan informasi yang relevan dari hasil pengujian sistem. Data yang diperoleh tidak hanya digunakan untuk memvalidasi kinerja sistem, tetapi juga dianalisis secara menyeluruh untuk mendapatkan pemahaman yang lebih komprehensif tentang kinerja dan efektivitas sistem yang dirancang. Analisis data ini merupakan langkah penting untuk memastikan bahwa sistem benar-benar mampu menjawab rumusan masalah yang telah ditentukan pada awal penelitian, sekaligus memberikan landasan yang kuat untuk pengambilan kesimpulan dan rekomendasi tentang solusi yang lebih baik untuk masalah tersebut.

3.2.7 Pembuatan Laporan Akhir

Pembuatan laporan akhir menjadi tahap yang sangat penting dalam sebuah penelitian. Tahap ini melibatkan proses pengumpulan, pengolahan, dan penyajian data secara terorganisir, dengan tujuan menyampaikan informasi yang terstruktur, jelas, dan mudah dimengerti oleh pembaca. Laporan ini tidak hanya berfungsi sebagai dkumentasi hasil penelitian, tetapi juga memberikan penjelasan mendalam tentang metode, analisis, serta temuan yang diperoleh, sehingga dapat dijadikan acuan yang berguna bagi berbagai pihak yang membutuhkannya.

3.3 Rancangan Perangkat Keras/Lunak

Tahap perancangan ini bertujuan untuk pembuatan konsep sebagai langkah awal perancangan. Pada tahap ini dilakukan sistem sistem mulai dari hardware dan software dari alat yang akan dibuat. Kemudian dilakukan perancangan desain baik secara mekanik maupun eletrik untuk alat yang akan dibuat.

3.3.1 Analisis Kebutuhan Sistem

Analisis kebutuhan merupakan langkah awal untuk mengidentifikasi dan menentukan kebutuhan sistem yang perlu digunakan dengan mempertimbangkan serta memperhitungkan berbagai elemen yang diperlukan. Adapun kebutuhan yang diperlukan dalam merancang sistem diantaranya**:**

Tabel 3. 1 Kebutuhan *Hardware*

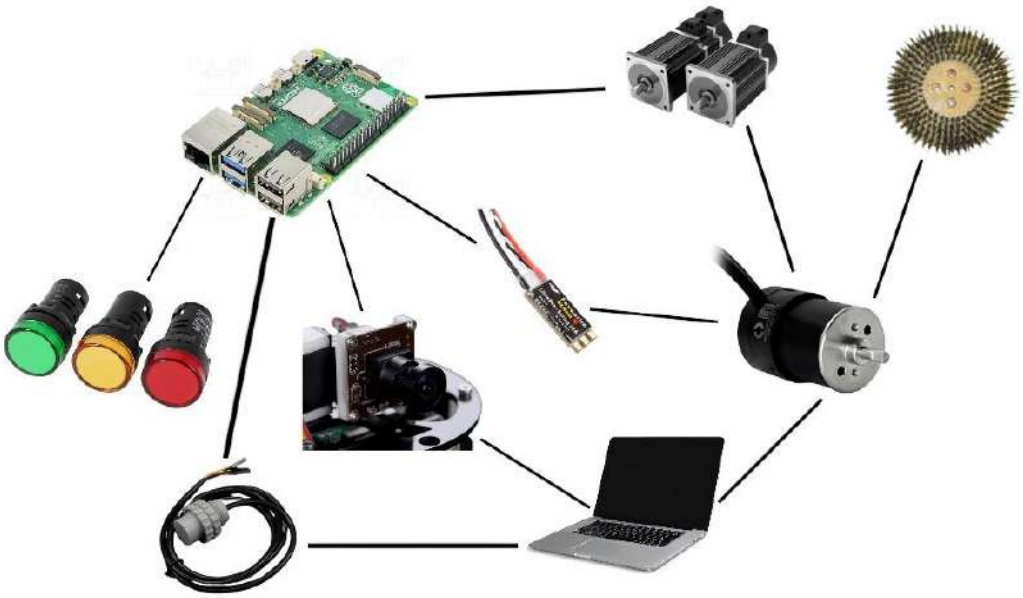
No	*Hardware*	Jumlah Unit
1	ROV (*Remotely Operated Vehicle*)	1 Buah
2	Sensor *Underwater Ultrasonic*	1 Buah
3	Laptop	1 Buah
4	*Encoder*	1 Buah
5	BLDC M200	1 Buah
6	*Camera*	1 Buah
7	*Brush*	3 Buah

Tabel 3. 2 Kebutuhan *Software*

No.	*Software*
1	phpMYAdmin
2	Matlab
3	Visual Studio
4	Qground Control
5.	Google Colab

3.3.2 Perancangan Desain Hardware

Perancangan hardware (perangkat keras) adalah proses merancang, mengembangkan dan mengimplementasikan komponen fisik dari sistem komputer atau perangkat elektronik. Proses ini mencakup berbagai langkah dan keputusan yang bertujuan unutk memastikan bahwa perangkan keras yang dirancang memenuhi spesifikasi dan kebutuhan yang telah ditentukan**.**



Gambar 3. 10 Perancangan *Hardware*

Gambar ini menampilkan berbagai komponen utama dalam perancangan sistem kendali motor BLDC berbasis Raspberry Pi. Berikut penjelasan singkat mengenai komponen-komponen tersebut:

1. Kamera

   Kamera digunakan untuk menangkap citra lambung kapal yang kemudian diolah oleh Raspberry Pi dengan algoritma YOLO untuk mendeteksi serta mengklasifikasikan biofouling menjadi beberapa kelas.

2. Raspberry Pi

   Raspberry Pi berperan sebagai otak sistem yang mengolah input darikamera, sensor ultrasonik, dan encoder, menjalankan YOLO dan algoritma PID kecepatan, lalu menghasilkan sinyal kendali berupa setpoint RPM dan PWM menuju ESC.

3. Sensor Ultrasonik Bawah Air

   Sensor ultrasonik mengukur jarak antara brusher dan permukaan lambung kapal, mengirimkan data jarak ke Raspberry Pi untuk memutuskan apakah motor diizinkan berputar (jarak < 20 cm) serta menjaga gap kerja yang aman selama proses pembersihan.

4. Encoder

   Encoder dipasang pada poros motor BLDC untuk mengukur kecepatan putar aktual (RPM) dan mengirimkan umpan balik ke Raspberry Pi, sehingga pengendali PID dapat mengoreksi duty cycle PWM guna menjaga stabilitas kecepatan saat beban brushing berubah.

5. ESC + Motor BLDC (M200 + Brusher)

   ESC menerima sinyal PWM dari Raspberry Pi dan mengatur suplai tiga fasa ke motor BLDC M200, yang memutar brusher untuk mengikis biofouling; kecepatan motor disesuaikan terhadap kelas fouling yang dihasilkan YOLO dan distabilkan oleh PID.

6. Laptop / Dashboard

   Laptop berfungsi sebagai dashboard pemantauan dan konfigurasi, menerima data dari Raspberry Pi berupa RPM motor, jarak ultrasonik, kelas biofouling, dan status kebersihan lambung sehingga operator dapat memonitor dan mengendalikan operasi hull cleaning secara real time.

7. Lampu Indikator

   Lampu indikator dikendalikan oleh Raspberry Pi untuk menampilkan status sistem secara cepat kepada operator, misalnya kondisi standby, proses pembersihan sedang berlangsung, atau terjadi gangguan/error pada sensor atau aktuator

Seluruh komponen ini bekerja secara terintegrasi untuk membentuk sistem kendali motor BLDC yang efisien. Data dari kamera, sensor ultrasonik, dan encoder akan diproses oleh Raspberry Pi, yang kemudian mengontrol motor BLDC melalui driver L298N. Selain itu, data dan status sistem ditampilkan secara real-time pada LCD atau dashboard untuk pemantauan pengguna.

3.3.3 Penentuan Model Matematis

Perancangan Permodelan Sistem Kontrol dimulai dengan penentuan model matematis yang akan dikenakan, dan dilanjutkan dengan proses merancang, mengembangkan dan mengimplementasikan permodelan sistem kontrol, dimana penelitian ini menggunakan pendekatan PID. Proses ini mencakup berbagai langkah dan keputusan yang bertujuan unutk memastikan bahwa permodelan sistem kontrol menggunakan PID yang dirancang memenuhi spesifikasi dan kebutuhan sistem Brush kecepatan adaptif.

Tabel 3. 3 Perbedaan Orde 1 dan Orde 2

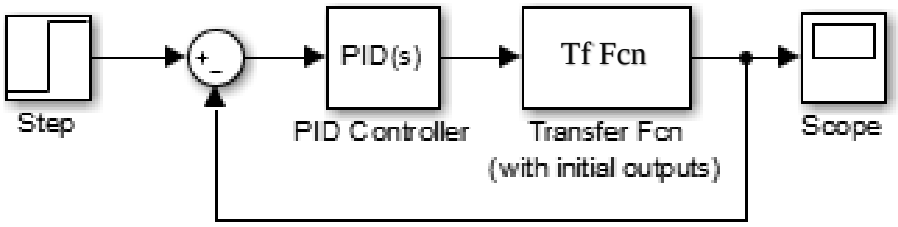
Aspek	Orde 1	Orde 2 (Underdamped, ζ<1)
Fungsi Alih	$\frac{K}{Ts+1}$ (3.1)	$\frac{\omega_n^2}{e^2+2\zeta\omega_n s+\omega_n^2}$ (3.2)
Respon Step	Monotonik eksponensial, 63.2% di t=τ	Osilasi teredam, overshoot
Overshoot	0% (tidak osilasi)	$e^{-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}} Hingga\ 100\% +$
Settling Time	ts≈4τ (2% kriteria)	ts≈4/(ζωn)*ts*≈4/(*ζωn*)
Rise Time	tr≈2.2τ*tr*≈2.2*τ* (10-90%)	tr=π−β)/(ωd)*tr*=*π*−*β*)/(*ωd*) dengan β=tan[fo] −1(1−ζ2/ζ)*β*=tan−1(1−ζ2/ζ)
Parameter Tuning	Hanya τ (kecepatan)	ζ (damping), ωn (kecepatan)

Dalam permodelan persamaan matematis dibagi menjadi dua, yakni orde 1 dan orde 2. Tabel 3.1 merupakan perbedaan dari Orde 1 dan Orde 2, pada penelitian kali ini akan diuji menggunakan kedua Orde dan hasilnya akan dibandingkan. Pada penelitian kali ini pengujian akan menggunakan permodelan sistem kontrol yang dibagi menjadi 2, Permodelan dengan simulasi kecepatan digunakan untuk model awal dan Permodelan close-loop dengan basis torsi/arus akan digunakan untuk implementasi yang lebih akurat.

1. Permodelan sistem menggunakan simulasi kecepatan dengan orde 1

$$G(s) = \frac{\Omega(s)}{U(s)} = \frac{K}{Ts + 1}$$

Didapatkan Transfer function Orde 1 menggunakan asumsi voltase input 16V, dengan asumsi T=0,2 sebagai titik awal yang nantinya akan digunakan untuk koreksi data pengukuran.



Gambar 3. 11 Blok Diagram Permodelan Sistem menggunakan Orde 1

Gambar 3.6 menjelaskan susunan diagram blok permodelan sistem kontrol untuk mengendalikan BLDC dengan menggunakan simulasi kecepatan.

2. Permodelan sistem menggunakan simulasi kecepatan dengan orde 2

Dalam menentukan permodelan matematis menggunakan orde dua dibutuhkan beberapa variabel tambahan seperti Dumping Ratio (ζ) dan Natural frequency (ωn).

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}.$$

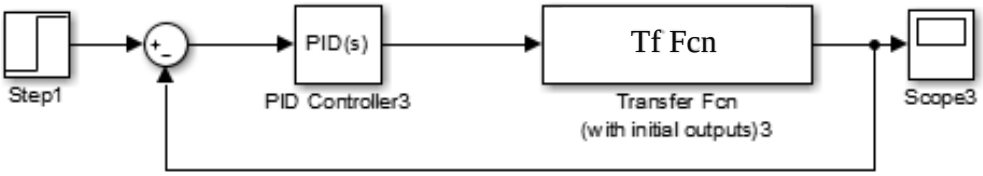
Untuk menyusun denominator orde 2 yang fisik, masih diperlukan momen inersia rotor + brusher J dan koefisien friksi viskos B, yang tidak tercantum dalam datasheet BLDC yang digunakan dalam penelitian ini (M200 Bluerobotic).

Tanpa$J$dan$B$, transfer function orde 2 hanya bisa dibuat sebagai model aproksimasi (disesuaikan dengan data eksperimen, bukan murni dari datasheet). Untuk mendapatkan Jdan$B$ pada penelitian ini digunakan model teoretis mengambil *grain* Kecepatan-Tegangan (dihitung dari Kv) dan Dumping Ratio (ζ) dan Natural frequency (ωn) dipilih dari target Settling Time atau Overshoot. Pada permodelan teoretis matematis kali ini penulis menargetkan Settling Time maksimum 1 ms dan Overshoot 2%, menggunakan pendekatan dari

$$M_p = e^{-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}} \tag{3.3}$$

$$t_p = \frac{\pi}{\omega_d} = \frac{\pi}{\omega_n\sqrt{1-\zeta^2}}. \tag{3.4}$$

$PO = 0{,}02 \Rightarrow \zeta \approx 0{,}78$, dari hasil Dumping Ratio (ζ) dan Natural frequency (ωn) didapatkan persamaan matematis untuk Orde dua adalah sebagai berikut:



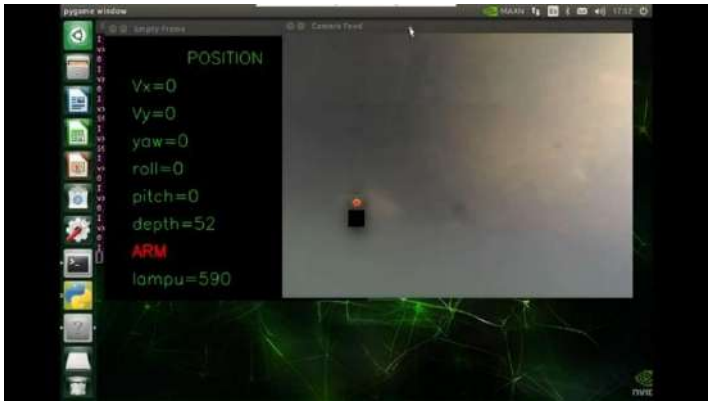
Gambar 3. 12 Blok Diagram Permodelan Sistem menggunakan Orde 2

Gambar 3.8 menjelaskan susunan diagram blok permodelan sistem kontrol untuk mengendalikan BLDC dengan simulasi kecepatan menggunakan pendekatan matematis orde 2. Permodelan matematis yang nantinya akan didukung nilai PID dan menggunakan Input RPM dibagi 3 yakni 80, 120 dan 200.

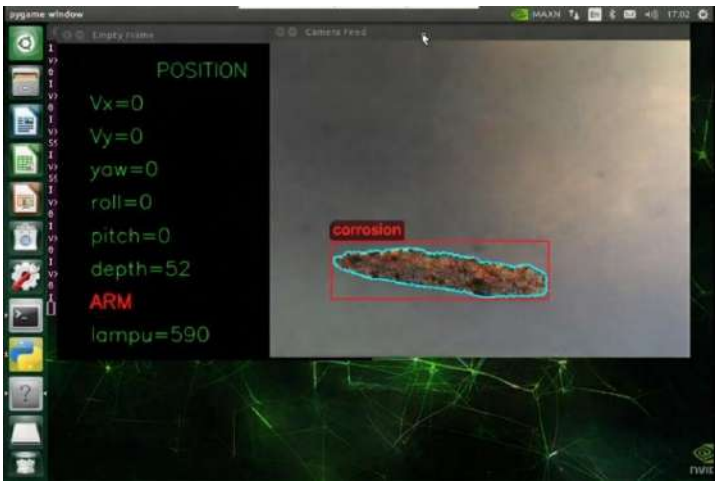
3.3.4 Perancangan You Only Look Once (YOLO)

Metode YOLOv8-Segmentation dimaksudkan untuk secara otomatis mendeteksi dan mensegmentasi area korosi pada citra lambung kapal. Ini dipilih karena metode ini dapat melakukan segmentasi objek hingga tingkat piksel secara

real-time, yang memungkinkan representasi visual yang tepat dari bentuk dan luasan area korosi.



(a)



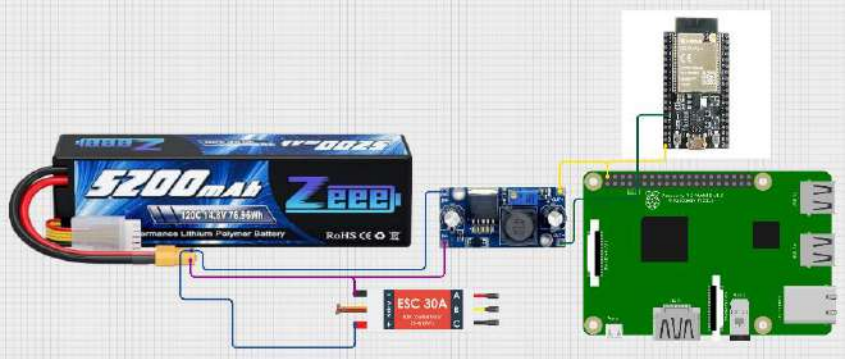
(b)

Gambar 3. 13 *Image* Hasil *Processing by YOLO Result*, (a) sebelum YOLO aktif, (b) sesudah YOLO aktif

Hubungan antara ukuran objek pada gambar dan ukuran objek sebenarnya dapat digambarkan sebagai berikut, dengan asumsi kamera ditempatkan pada jarak tetap 25 cm dari permukaan lambung kapal.

3.3.5 Wiring Diagram Supply sistem

Pada bagian ini menjelaskan *supply system Brushing* yang ada pada ROV, yang tertera pada gambar.

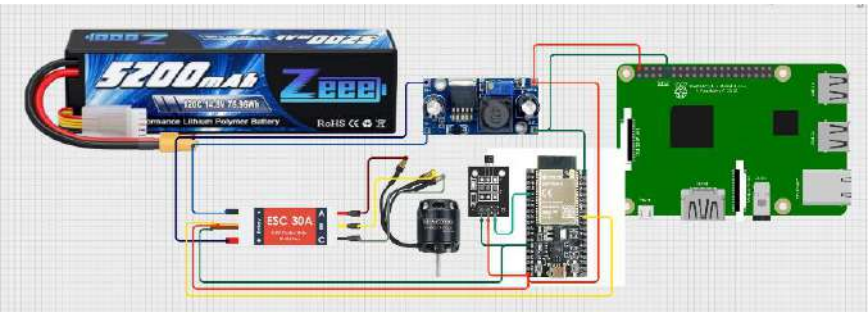


Gambar 3. 14 *Wiring* Diagram Sistem *Brush*

Sistem suplai daya pada penelitian ini menggunakan satu baterai Li-Po 4S 5200 mAh sebagai sumber energi utama yang didistribusikan ke beberapa jalur sesuai kebutuhan sistem. Distribusi daya dibagi menjadi tiga bagian utama, yaitu jalur daya motor, jalur regulasi tegangan, dan jalur logika, dengan seluruh rangkaian menggunakan referensi ground yang disatukan. Tegangan baterai disalurkan secara langsung menuju ESC 30 A untuk menyuplai motor brushless M200 melalui kabel daya yang memiliki kemampuan membawa arus tinggi. Jalur lainnya terhubung ke modul buck converter untuk menurunkan tegangan baterai menjadi tegangan yang sesuai dan aman bagi Raspberry Pi sebagai komputer utama dalam pemrosesan YOLO dan pengoperasian dashboard. Selanjutnya, Raspberry Pi digunakan sebagai sumber suplai bagi ESP32 pada sisi logika. ESP32 berfungsi sebagai pengendali real-time untuk algoritma PID sekaligus menghasilkan sinyal PWM yang dikirimkan ke ESC. Pemisahan jalur daya motor dan jalur logika memungkinkan sistem kontrol bekerja lebih stabil dengan mengurangi potensi gangguan elektrik dari beban motor, sekaligus mempertahankan rancangan yang ringkas dan efisien.

3.3.6 Rancangan Wiring PID Kontrol

Pada rancangan *wiring* ini merupakan bagaimana sistem PID ter-*install*, dimulai dengan perangkat keras sebagai *input* terhadap proses yang dilakukan oleh *Microcontroller Unit* (MCU).

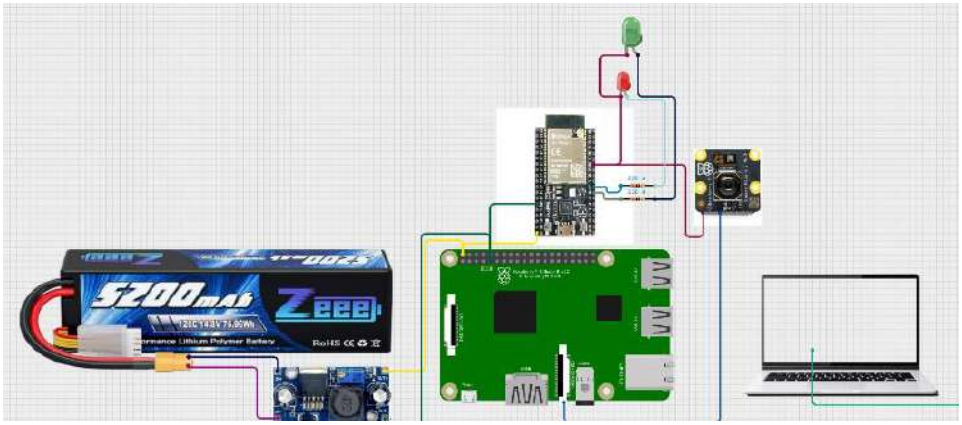


Gambar 3. 15 *Wiring* Diagram PID control

Rangkaian wiring kontrol PID dirancang untuk membentuk sistem kendali kecepatan tertutup pada motor M200 dengan ESP32 sebagai pengendali utama dan Raspberry Pi sebagai pemberi referensi setpoint. Encoder yang terpasang pada poros motor menghasilkan sinyal pulsa sebagai umpan balik kecepatan. Pulsa tersebut dibaca oleh ESP32 dan dikonversi menjadi nilai RPM aktual, kemudian dibandingkan dengan nilai setpoint RPM yang diterima dari Raspberry Pi melalui komunikasi serial. Selisih antara nilai setpoint dan RPM aktual digunakan sebagai error masukan pada algoritma PID. Berdasarkan nilai error tersebut, ESP32 menghitung koreksi kontrol dan menghasilkan sinyal PWM yang dikirimkan ke pin sinyal ESC 30 A untuk mengatur daya motor. Proses pembacaan RPM dan koreksi PWM berlangsung secara berulang sehingga kecepatan motor dapat dipertahankan mendekati setpoint meskipun terjadi perubahan beban selama proses brushing. Dengan konfigurasi ini, jalur daya tinggi tetap berada pada baterai, ESC, dan motor, sedangkan pengukuran, perhitungan PID, dan komunikasi berlangsung pada jalur logika bertegangan rendah.

3.3.7 Rancangan Wiring Clasification Object

Pada rancangan wiring ini merupakan bagaimana sistem klasifikasi objek menggunakan YOLO ter-install, dimulai dengan perangkat keras (kamera) sebagai input terhadap proses yang dilakukan oleh Microcontroller Unit (MCU).

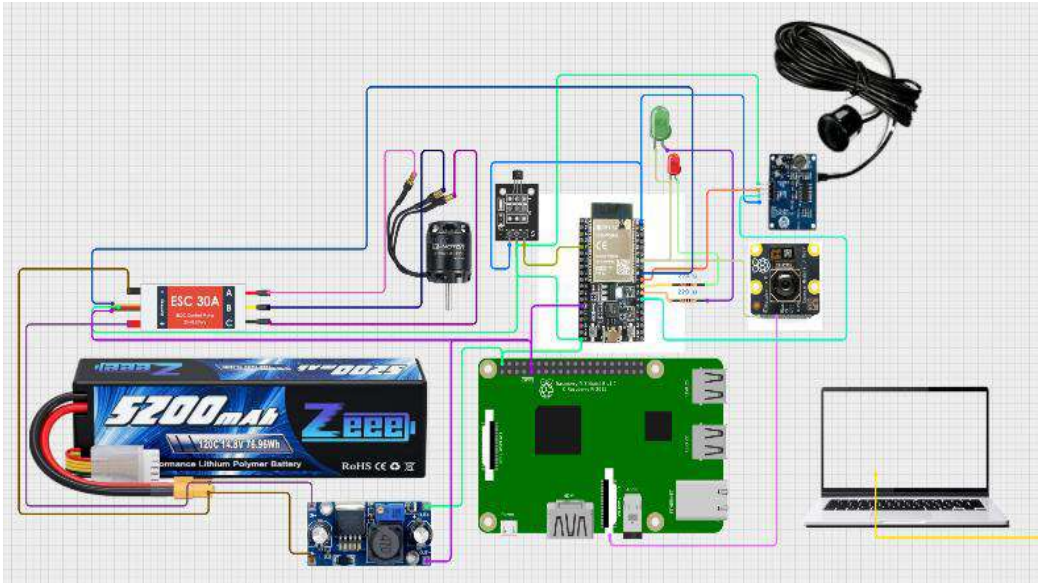


Gambar 3. 16 *Wiring* Diagram klasifikasi objek

Rangkaian wiring klasifikasi objek mengintegrasikan sistem visi, komputasi, komunikasi, dan indikator status dalam satu sistem ROV. Baterai Li-Po melalui modul buck converter digunakan sebagai sumber tegangan yang telah diturunkan dan distabilkan untuk menyuplai Raspberry Pi dan ESP32, dengan referensi ground yang sama pada rangkaian kontrol. Kamera bawah air terhubung ke Raspberry Pi sebagai sumber input citra permukaan lambung kapal. Citra yang diperoleh kemudian diproses menggunakan model YOLO untuk mendeteksi keberadaan biofouling dan mengklasifikasikannya ke dalam tiga tingkat, yaitu Light, Medium, dan Heavy. Hasil klasifikasi selanjutnya dikirimkan dari Raspberry Pi ke ESP32 melalui komunikasi serial dalam bentuk kode kelas atau nilai setpoint RPM. ESP32 kemudian menggunakan informasi tersebut untuk menentukan kondisi operasi motor brush dan mengaktifkan LED indikator sebagai penanda status sistem, seperti kondisi siap, proses pembersihan, atau kegagalan deteksi. Selain itu, Raspberry Pi terhubung dengan laptop operator melalui jaringan Wi-Fi atau router untuk menampilkan citra hasil deteksi, kelas biofouling, dan parameter sistem pada dashboard secara real-time selama proses inspeksi dan pembersihan berlangsung.

3.3.8 Rancangan Wiring Keseluruhan Hardware

Pada rancangan *wiring* ini merupakan bagaimana keseluruhan sistem ter-integrasi, dimulai dengan perangkat keras sebagai *input* terhadap proses yang dilakukan oleh *Microcontroller Unit* (MCU) dan akan ditampilkan oleh perangkat keras untuk penampil data sebagai *output.*



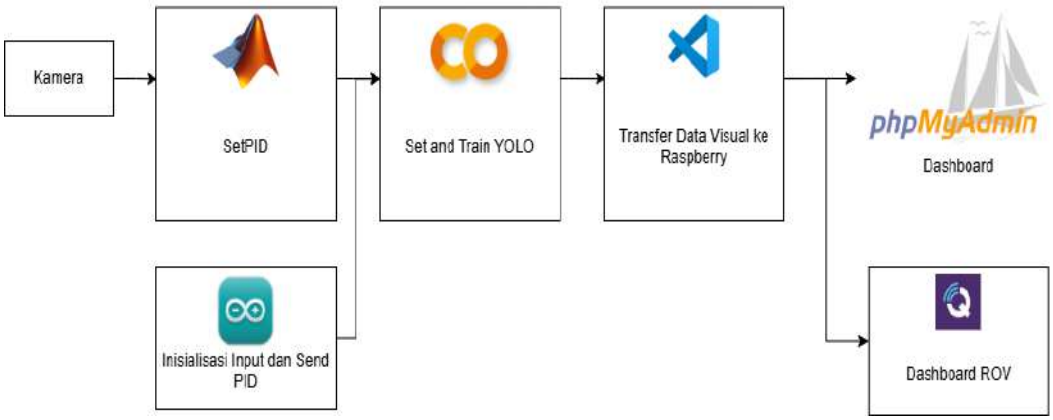
Gambar 3. 17 *Wiring* diagram keseluruhan sistem *brush*

Rangkaian wiring keseluruhan sistem ini mengintegrasikan tiga subsistem utama—daya, kontrol PID motor, dan klasifikasi objek berbasis YOLO—dalam satu arsitektur ROV brusher. Baterai Li-Po 4S menjadi sumber energi tunggal yang melalui buck converter disalurkan sebagai tegangan ter-regulasi untuk Raspberry Pi, ESP32, sensor, dan kamera, sementara cabang tegangan tinggi langsung memasok ESC 30 A dan motor brushless M200 sehingga jalur daya besar dan jalur logika tetap terpisah. Encoder pada motor memberikan umpan balik ke ESP32 untuk menghitung kecepatan aktual, yang kemudian dibandingkan dengan setpoint rpm hasil pemetaan kelas biofouling dari Raspberry Pi; error kecepatan ini diproses oleh algoritma PID di ESP32 untuk mengatur sinyal PWM ke ESC dan menjaga stabilitas putaran brush saat kontak dengan lambung kapal. Di sisi lain, kamera bawah air mengirim citra ke Raspberry Pi untuk diproses oleh model YOLO, yang mengklasifikasikan tingkat kekotoran dan mengirimkan kelas tersebut beserta parameter pendukung ke ESP32 serta ke dashboard pada laptop operator melalui jaringan Wi-Fi/router.

3.3.9 Perancangan Software

Perancangan *software* (perangkat lunak) adalah proses merancang, mengembangkan dan mengimplementasikan perangkat lunak dari sistem. Proses ini mencakup berbagai langkah dan keputusan yang bertujuan unutk memastikan

bahwa perangkan lunak yang dirancang memenuhi spesifikasi dan kebutuhan sistem Brush kecepatan adaptif.



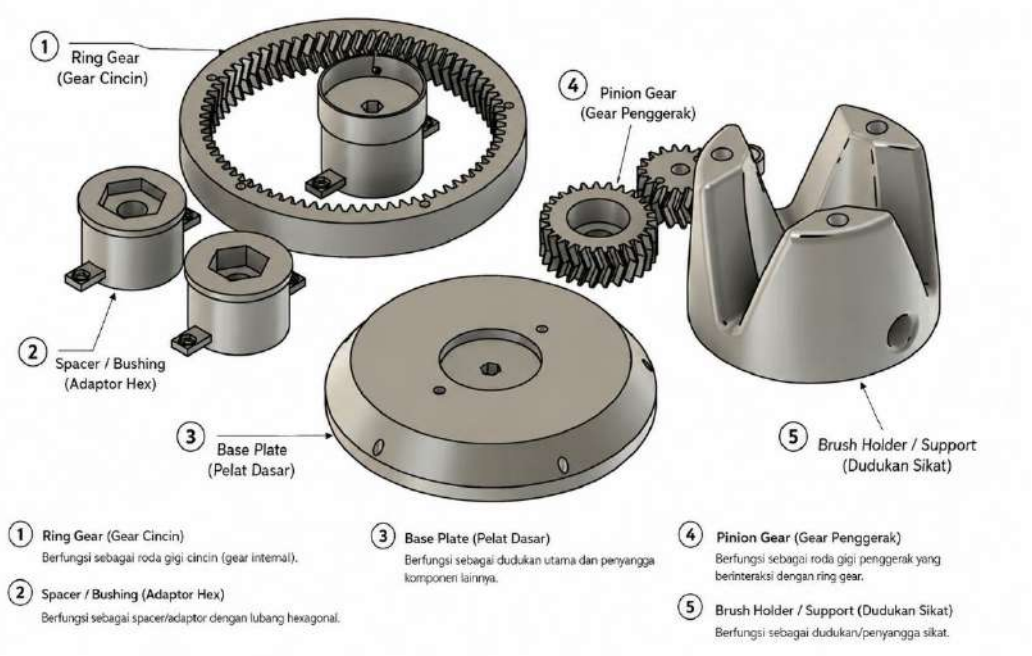
Gambar 3. 18 Rancangan *Software*

Alur perangkat lunak pada sistem ini dimulai dari akuisisi citra lambung kapal oleh kamera bawah air yang kemudian digunakan sebagai data dasar untuk dua proses utama, yaitu penentuan parameter PID dan pelatihan model YOLO.

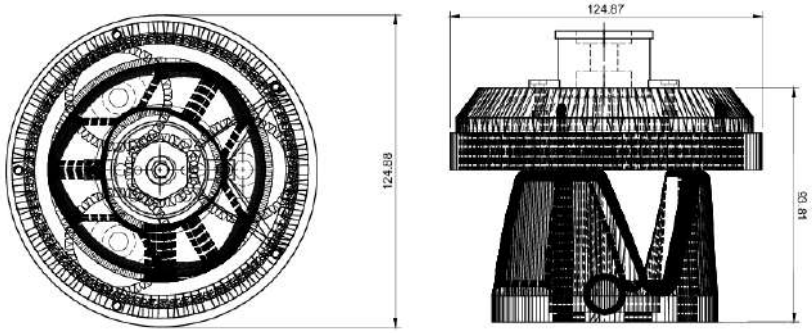
Pada tahap awal, data citra dan hasil pengujian motor dimanfaatkan di Matlab untuk melakukan pemodelan sistem dan penalaan parameter pengendali PID, sementara di Google Colab dilakukan proses anotasi, set, dan training model YOLO menggunakan dataset biofouling yang sama. Setelah model YOLO terlatih dan parameter PID ditetapkan, keduanya diintegrasikan dalam program utama yang dikembangkan di Visual Studio Code dan dijalankan pada Raspberry Pi, sehingga Raspberry Pi dapat menerima streaming citra dari kamera, menjalankan inferensi YOLO secara real-time, mengklasifikasikan tingkat kekotoran, dan mengirimkan kelas serta setpoint rpm ke ESP32 untuk dikendalikan oleh algoritma PID. Secara paralel, Raspberry Pi juga mencatat data visual dan parameter operasi ke basis data MySQL yang diakses melalui phpMyAdmin untuk ditampilkan pada dashboard web, sedangkan data status ROV seperti kelas fouling, kecepatan motor, dan jarak sensor dikirim ke Dashboard ROV (misalnya QGroundControl) sehingga operator dapat memantau dan mengevaluasi performa pembersihan secara menyeluruh.

3.3.10 Perancangan Desain Mekanik

Berikut merupakan perancangan bentuk dari ROV’s Brusher dalam penelitian tugas akhir ini.



Gambar 3. 19 Proyeksi Mekanik *Brusher*



(a) (b)

Gambar 3. 20 (a) Geometri mekanik *Brusher* tanpa *Brush* tampak atas, (b) Geometri mekanik *Brusher* tanpa *Brush* tampak samping

Gambar 3.20 menunjukkan dimensi utama modul brusher yang berbentuk silinder dengan pandangan atas dan samping. Modul memiliki diameter luar 124,88 mm. Pada pandangan samping tampak tinggi keseluruhan, mulai dari bagian paling bawah hingga ke bagian paling atas dudukan, yaitu sekitar 95,81 mm. Gearbox yang dibuat berupa Planetory Gear dengan spesifikasi diameter poros output M200 6,421mm, jumlah gigi sun : 18, jumlah gigi planet : 27 dan jumlah gigi ring : 72. Artinya dengan kombinasi ini kamu dapat reduksi 5:1. Kecepatan berkurang jadi 1/5 dan torsi idealnya naik 5× (minus losses) dengan perhitungan sebagai berikut:

$$(R = 1 + \frac{Z_{\text{sun}}}{Z_{\text{ring}}} = 1 + \frac{18}{72} = 5)$$

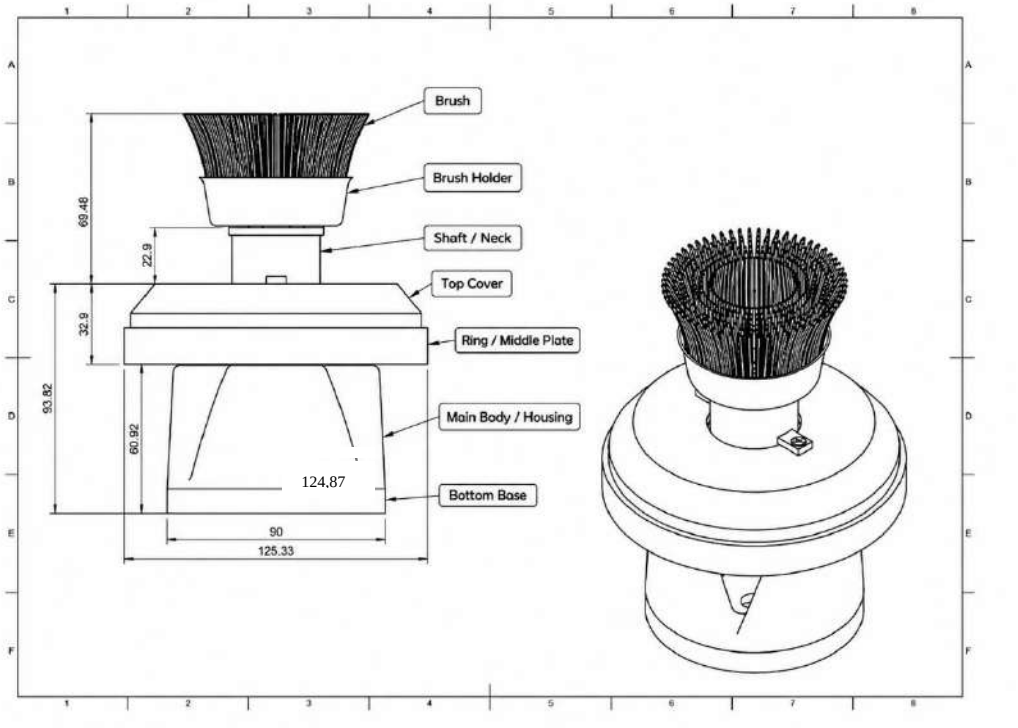
Peningkatan torsi teoritis di brush

$(T_{\text{motor}} = 0{,}5\ \text{Nm})$

$(T_{\text{out,ideal}} = R \times T_{\text{motor}} = 5 \times 0{,}5 = 2{,}5\ \text{Nm})$

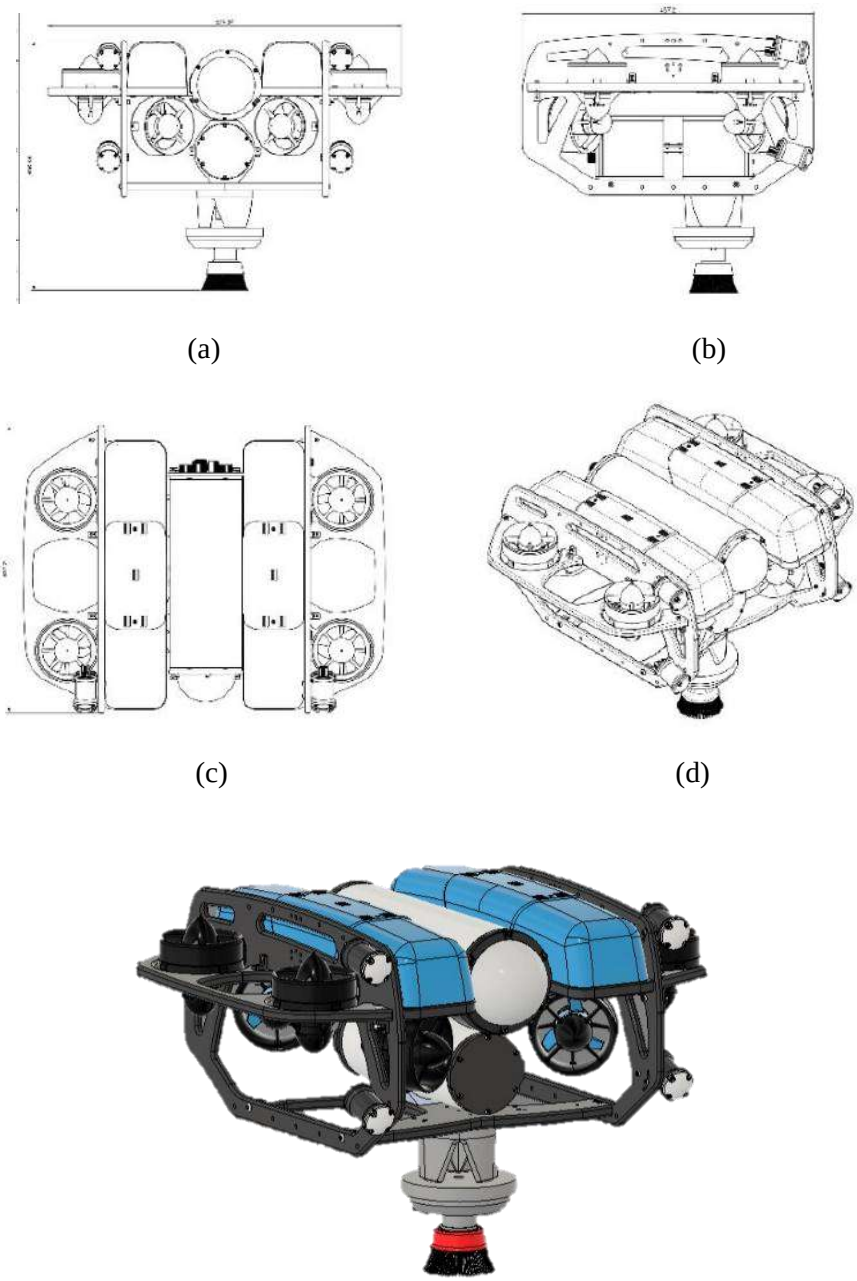
Dengan pengaruh dari gear 3D print + bearing + seal akan membuat efisiensi dari kondisi nyata 70-85%, maka dapat diasumsikan nilai efisiensi $\eta = 0{,}8$.

$(T_{\text{out,real,max}} \approx 0{,}8 \times 5 \times 0{,}5 = 2{,}0\ \text{Nm (mendekati stall)})$



Gambar 3. 21 Desain Mekanik dengan *Brush*

Gambar 3.21 menampilkan dimensi modul brusher lengkap dengan sikatnya dari pandangan samping ortografis dan pandangan isometrik. Pada pandangan samping, diameter luar dasar modul tercatat 124,87 mm. Lebar bagian tengah modul yang lebih sempit ditunjukkan sebesar 90 mm, yang menggambarkan lebar badan/bracket di bawah housing brush. Tinggi total modul dari dasar sampai ujung tertinggi sikat adalah 159,82 mm, sedangkan tinggi housing utama tanpa sikat 99,92 mm. Di atas housing terdapat bagian leher dan dudukan brush dengan tinggi 22,9 mm, dan tinggi kumpulan filamen sikat sendiri sekitar 60,92 mm dari dudukannya.



Gambar 3. 22 Desain ROV *brusher* Ortografis, Isometri, dan 3D, (a) *Front view*, (b) *Side View*, (c) *Top View*, (d) *Isometric View* dan (e) *3D Visualization*

Gambar tersebut menampilkan model 3D ROV jenis inspection class yang telah dimodifikasi dengan modul brusher di bagian bawah. Struktur utama ROV berupa rangka berbentuk kubus terbuka berwarna gelap yang menopang empat thruster horizontal di sisi kiri-kanan dan dua thruster vertikal di bagian dalam frame,

sementara di bagian atas terdapat bodi streamline berwarna biru–putih yang menjadi rumah elektronik dan pelampung. Di sisi samping ROV tampak beberapa dome atau housing silinder yang berfungsi sebagai tempat sensor dan komponen kedap air, selaras dengan konfigurasi tipikal ROV komersial. Pada bagian tengah bawah rangka terpasang unit brusher berbentuk silinder dengan rumah gear dan dudukan berbentuk kerucut terbalik yang berujung pada sikat bundar berwarna merah-hitam sebagai aktuator utama pembersih biofouling. Secara visual, gambar ini memperlihatkan integrasi antara platform ROV standar dengan modul brusher yang ditempatkan di pusat massa bawah untuk memungkinkan operasi pembersihan lambung kapal secara stabil dan terarah.

BAB 4
HASIL DAN PENELITIAN

Pada bab ini berisi tentang proses ppengujian alat pada penelitian yang bertujuan dapat mengetahui tingkat keakuratan dari komponen yang digunakan dan mengetahui besar persentase error yang dihasilkan.

4.1 Pengujian Sensor dan Aktuator

Pada proses pengujian sensor dan aktuator pada penelitian Tugas Akhir ini dilakukan menggunakan perangkat bantu untuk mengetahui kondisi serta respon perangkat berdasarkan data yang tertera pada datasheet perangkat. Melalui pengujian tersebut, persentase error kinerja sensor maupun aktuator dapat dihitung dan dianalisis apakah perangkat berfungsi dengan baik atau tidak.

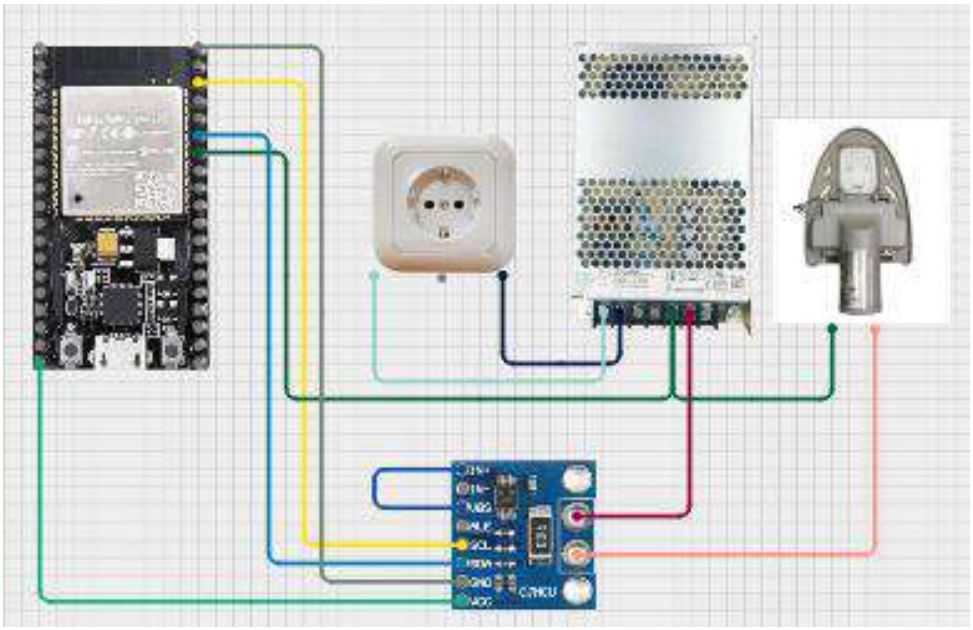
4.1.1 Pengujian Sensor Arus ACS-758

pengujian sensor arus dalam tugas akhir ini dilakukan melalui dua rangkaian berbeda, yaitu menggunakan sensor ACS758 dan pengukuran menggunakan multimeter digital.

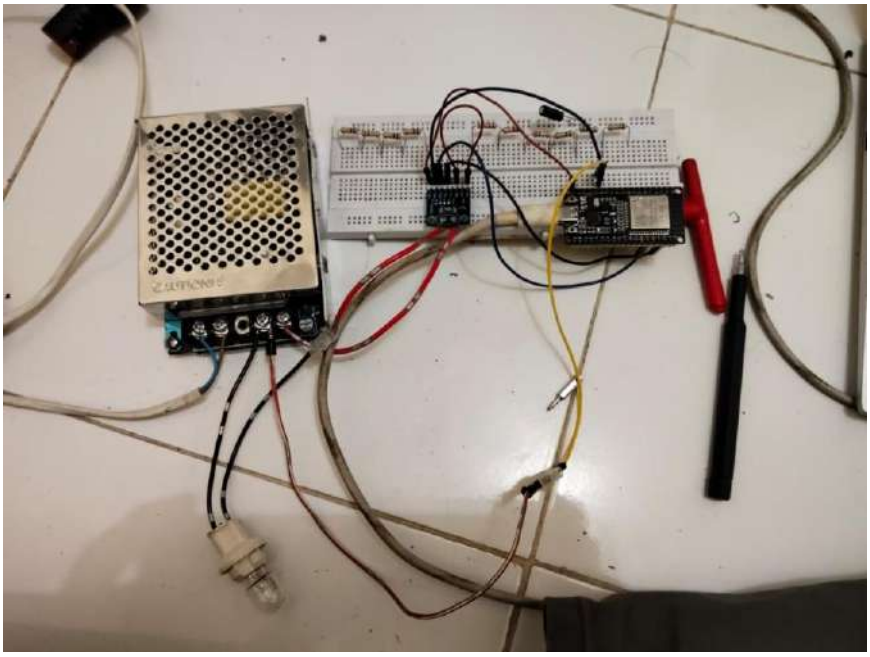
Tabel 4. 1 Hasil Pengujian ACS758

Waktu Percobaan (s)	ACS758 (mA)	Multimeter (mA)	*Error* (%)
5	667.90	700	4.585714286
10	668.00	700	4.571428571
15	668.00	700	4.571428571
20	667.90	700	4.585714286
25	668.00	700	4.571428571
30	668.00	700	4.571428571
35	668.10	700	4.557142857
40	668.20	700	4.542857143
45	668.30	700	4.542857143
50	668.20	700	4.542857143
55	668.00	700	4.571428571
60	667.90	700	4.571428571
Rata-rata Error			4.56547619

Pada pengujian ini, lampu 12V digunakan sebagai beban. Rangkaian pengujian menggunakan sensor arus ACS758 ditunjukkan pada gambar 4.5 dan gambar 4.6 adalah dokumentasi pengujian sensor secara *real*. Implementasi dari gambar wiring ke percobaan secara real untuk menghindari kesalahan dan gangguan saat pengujian.



Gambar 4. 1 Rancangan rangkaian pengetesan ACS758



Gambar 4. 2 Realisasi rangkaian pengetesan ACS758

Berdasarkan pengujian yang dilakukan selama 60 detik dengan interval pengambilan data setiap 5 detik, sensor ACS758 menunjukkan performa pembacaan arus yang sangat stabil. Data hasil pengukuran dibandingkan secara langsung dengan multimeter standar sebagai instrumen kalibrasi untuk menentukan tingkat akurasi sensor. Selisih antara pembacaan sensor dan instrumen ukur menghasilkan nilai persentase *error* yang konsisten di angka 4,54% hingga 4,58%. Perhitungan rata-rata persentase *error* dari keseluruhan titik uji adalah sebesar 4,56%. Secara keseluruhan, sensor menunjukkan reliabilitas yang kuat untuk digunakan dalam sistem ini karena mampu mempertahankan konsistensi pembacaan dalam jangka waktu lama.

4.1.2 Pengujian Hall Sensor SS49E

Pengujian sensor Hall effect dilakukan untuk memvalidasi akurasi pembacaan umpan balik (feedback) kecepatan putaran motor sikat (RPM terbaca) terhadap nilai referensi aktual alat ukur standar (RPM Aktual). Pengujian ini mengevaluasi linearitas sensor pada spektrum sinyal Pulse Width Modulation (PWM) dari 1100 µs hingga 1900 µs, dengan titik netral (motor diam) berada tepat pada parameter PWM 1500 µs.

Tabel 4. 2 Hasil Pengujian Sensor Hall sesudah terdapat Gear Box

PWM	**RPM terbaca**	**RPM Aktual**	**Error %**
1100	2005.00	2008.10	0.154374782
1200	1690.00	1693.70	0.218456633
1300	1198.00	1202.00	0.332778702
1400	518.00	518.30	0.057881536
1500	0.00	0.00	0
1600	521.00	516.20	0.929872143
1700	1200.00	1200.80	0.066622252
1800	1680.00	1682.20	0.13078112
1900	2002.00	2005.80	0.189450593
Rata-rata Error%			0.231135307

Berdasarkan tabulasi data pengujian pada Tabel 4.2, hasil pembacaan sensor menunjukkan tingkat presisi yang sangat tinggi pada kedua arah putaran motor (maju dan mundur). Deviasi kesalahan tertinggi hanya berada di angka 0,9298%

yang terjadi pada pengondisian awal PWM 1600, di mana karakteristik transien motor baru mulai mengatasi gaya inersia mekanis diam.Secara keseluruhan, pemetaan komparatif dari sembilan variasi pulsa kendali ini menghasilkan nilai rata-rata galat (error) total yang sangat kecil, yaitu sebesar 0,2311%. Rendahnya nilai akumulasi error (jauh di bawah threshold toleransi 1%) ini secara ilmiah membuktikan bahwa sensor Hall effect yang diintegrasikan pada aktuator memiliki linearitas dan reliabilitas yang optimal. Dengan demikian, data keluaran dari sensor ini dinyatakan valid untuk digunakan secara real-time oleh algoritma kendali PID sebagai parameter umpan balik loop tertutup (closed-loop feedback) dalam mempertahankan kestabilan kecepatan putar sikat ROV saat beroperasi di bawah air.

4.1.3 Pengujian Pixhawk

Pengujian sensor IMU dilakukan untuk memastikan Pixhawk mampu membaca data percepatan dan perubahan sudut gerak pada sistem ROV. Sensor IMU terdiri dari accelerometer dan gyroscope yang bekerja pada sumbu X, Y, dan Z. Data sensor dikirim melalui protokol MAVLink dan diamati menggunakan perangkat komputer. Hasil pengujian menunjukkan bahwa perubahan posisi Pixhawk menghasilkan perubahan nilai pada data sensor, sehingga IMU dapat digunakan untuk mendukung pemantauan orientasi dan kestabilan gerak ROV.

Tabel 4. 3 Hasil Pengujian sensor IMU sumbu X

No	Sudut Pengujian (°)	Pembacaan Sensor (°)	Nilai Error (%)
Sumbu X atau Pitch			
1	0	0	0,00
2	5	4,88	2,40
3	10	9,97	0,30
4	15	15,15	1,00
5	20	19,77	1,15
6	25	25,11	0,44
7	30	29,89	0,37
8	35	35,03	0,09
9	40	40,20	0,50
10	45	45,08	0,18
Rata-rata Error (%)			0,71

Sumbu Y atau Roll			
1	0	0,01	0,01
2	5	5,09	1,80
3	10	9,91	0,90
4	15	15,19	1,27
5	20	19,86	0,70
6	25	24,93	0,28
7	30	30,02	0,07
8	35	35,19	0,54
9	40	39,79	0,53
10	45	45,21	0,47
Rata-rata Error (%)			0,73

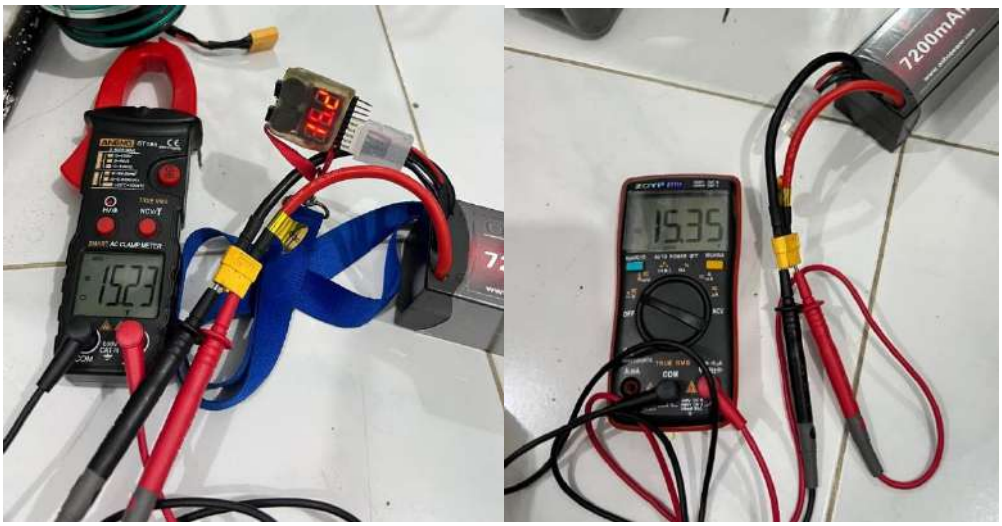
Gambar 4. 3 Pengujian Pixhawk

Selain dipengaruhi oleh jarak operasional, struktur pengumpulan *dataset* pengujian dalam lingkungan terbatas juga dipengaruhi oleh geometri pancaran gelombang sensor yang memiliki sudut kerucut (*beamwidth*) selebar 25 derajat pada penurunan daya -3 dB. Sudut pancaran yang cukup lebar ini berpotensi memicu fenomena *multipath reflection*, di mana sebagian gelombang akustik menabrak dinding sasis bawah ROV atau lantai area pengujian sebelum mengenai target utama lambung kapal, sehingga menyumbang fluktuasi galat acak pada beberapa titik ukur. Kendati demikian, dengan nilai akumulasi rata-rata *error* keseluruhan yang relatif kecil yaitu sebesar 2,6115%, data validasi ini secara ilmiah

membuktikan bahwa sensor *ping sonar* tersebut memiliki tingkat presisi dan reliabilitas yang sangat layak untuk diintegrasikan sebagai subsistem navigasi jarak pada wahana ROV.

4.1.4 Pengujian Baterai

Pengujian catu daya dilakukan untuk memverifikasi tegangan keluaran baterai Lithium Polymer (LiPo) 4S 7200mAh sebelum didistribusikan ke seluruh sistem elektronik ROV. Berdasarkan hasil pengukuran menggunakan baterai *checker*, multimeter digital (ZOYI ZT111) dan *clamp* meter (ANENG ST184).



Gambar 4. 4 Validasi output baterai

Diperoleh nilai tegangan masing-masing sebesar 15,35 V dan 15,23 V, yang menunjukkan bahwa kondisi baterai berada pada level tegangan yang stabil dan memadai untuk menyuplai komponen beban seperti motor *thruster* dan sistem kontrol.

4.1.5 Pengujian Pin ESP 32

Pengujian ini dilakukan untuk memastikan setiap pin *General Purpose Input Output* (GPIO) dan pin daya pada ESP32 DevKit V1 berfungsi dengan normal sebelum diintegrasikan ke dalam sistem. Proses pengujian dilakukan dengan memberikan logika *High* pada seluruh pin output melalui pemrograman, kemudian diukur menggunakan multimeter digital ANENG 622A untuk memverifikasi level tegangan yang keluar. Berdasarkan hasil pengukuran yang tercantum pada Tabel 4.2, seluruh pin daya (5V, 3V3, dan GND) serta pin digital menunjukkan nilai tegangan yang stabil dan sesuai dengan spesifikasi teknis (±3.3V untuk GPIO).

Dengan demikian, modul ESP32 dinyatakan berada dalam kondisi baik dan layak untuk digunakan dalam tahap perakitan perangkat.

Tabel 4. 4 Hasil Pengujian Pin ESP 32

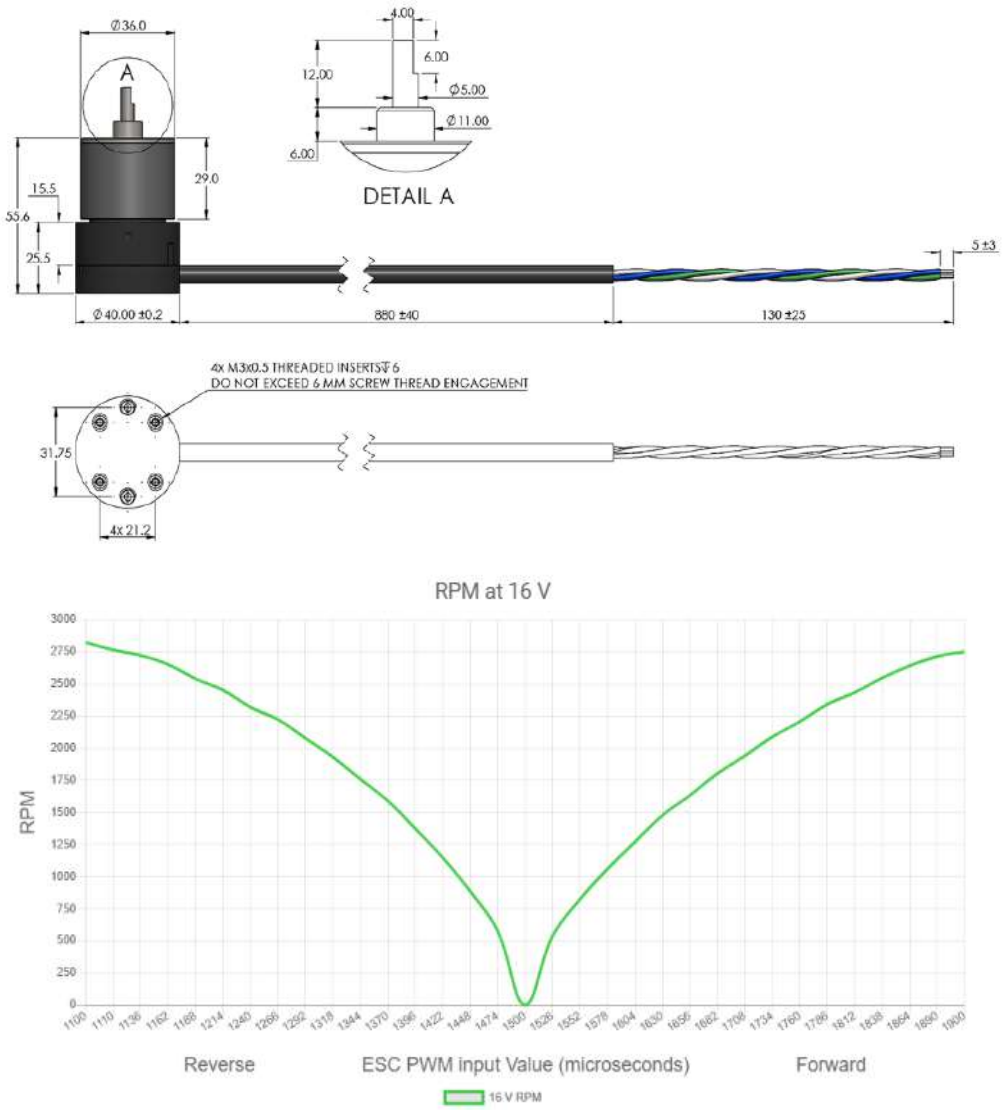
PIN	Parameter	Hasil (v)	Status	PIN	Parameter	Hasil (v)	Status
GND			√	GPIO 17	Logika HIGH	3.290	√
5V	Tegangan DC	5.067	√	GPIO 18	Logika HIGH	5.067	√
3V3	Tegangan DC	3.307	√	GPIO 19	Logika HIGH	3.307	√
GPIO 2	Logika HIGH	3.302	√	GPIO 21	Logika HIGH	3.297	√
GPIO 4	Logika HIGH	3.302	√	GPIO 22	Logika HIGH	3.304	√
GPIO 5	Logika HIGH	3.302	√	GPIO 23	Logika HIGH	3.293	√
GPIO 12	Logika HIGH	3.301	√	GPIO 25	Logika HIGH	3.279	√
GPIO 13	Logika HIGH	3.292	√	GPIO 26	Logika HIGH	3.304	√
GPIO 14	Logika HIGH	3.293	√	GPIO 27	Logika HIGH	3.286	√
GPIO 15	Logika HIGH	3.281	√	GPIO 32	Logika HIGH	3.293	√
GPIO16	Logika HIGH	3.291	√	GPIO 33	Logika HIGH	3.290	√

Gambar 4. 5 Pengujian Pin ESP 32

4.1.6 Pengujian Motor Brush

Pada pengujian Motor Brush M200, proses dilakukan dengan cara memberikan variasi sinyal input PWM secara bertahap, dengan suplai daya berupa tegangan sebesar 15,2V DC yang bersumber dari catu daya eksternal. Tujuan dari pengujian ini adalah untuk mengevaluasi performa motor dalam merespons perubahan sinyal kendali yang diberikan. Informasi teknis mengenai spesifikasi motor yang diuji dapat dilihat pada gambar 4.5 spesifikasi motor *brush Bluerobotic* M200. Pengukuran ini bertujuan untuk mengetahui seberapa baik motor motor *brush Bluerobotic* mampu menghasilkan putaran sesuai dengan variasi nilai PWM

yang diberikan, serta untuk memastikan bahwa motor dapat bekerja secara optimal dan responsif dalam rentang kendali yang dirancang.

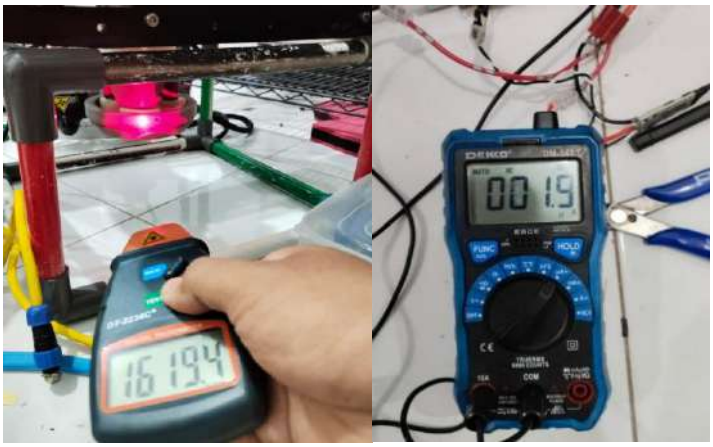


Gambar 4. 6 Spesifikasi Motor *Brush Buerobotic* M200

(https://bluerobotics.com/store/thrusters/t100-t200-thrusters/m200-motor/)

Diakses pada 05 Mei 2026

Pengujian ini dilakukan pada kondisi motor di darat dengan kondisi *gear box*. Proses pengujian dapat dilihat pada gambar 4.7 pengujian motor *brush* serta pada tabel 4.5 yang menyajikan hasil pengujian motor *brush* yang diimplementasikan pada sistem propulsi ROV. Dalam sistem ini, ESP 32 berperan dalam mengirimkan sinyal ke ESC, sementara ESC memberikan PWM sesuai dengan sinyalir dari ESC.



Gambar 4. 7 Pengujian Motor dengan *Gear Box*

Tabel 4. 5 Hasil Pengujian Motor *Brush* M200 dengan *Gear Box*

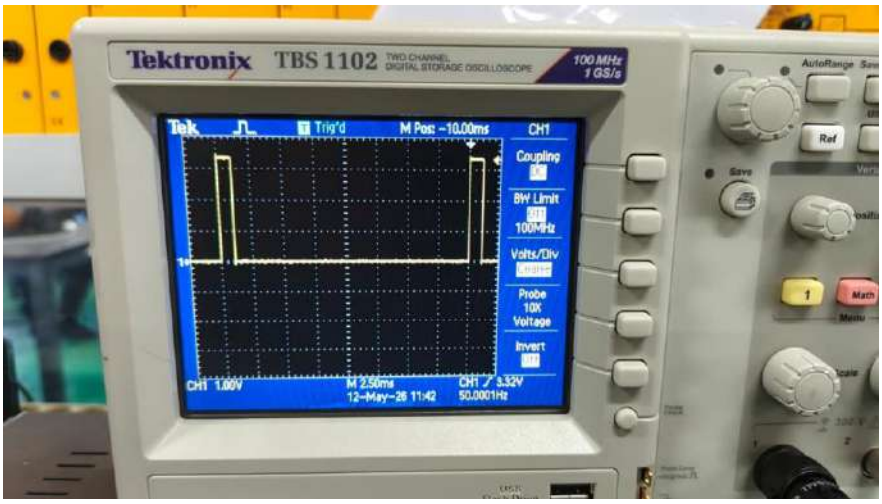
PWM	Arus (A)	RPM
1100	3,10	436,00
1200	2,70	902,50
1300	1,90	1346,60
1400	0,90	1619,40
1500	0,00	0,00
1600	0,50	414,00
1700	1,60	915,30
1800	2,50	1340,20
1900	2,90	1620,00

Berdasarkan hasil pengujian, tabel 4.3 merupakan hasil pengujian ke Motor *Brush Bluerobotic* M200 dengan variasi input PWM menggunakan ESP 32 yang digunakan pada sistem propulsi ROV. Hasil pengujian pada Tabel 4.5 didapatkan nilai RPM maksimal 1620 µs dengan menggunakan tegangan input 15,2 V.

4.1.7 Pengujian Electronic Speed Control

Pengujian sinyal masukan (*input*) pada *Electronic Speed Controller* (ESC) dilakukan untuk memastikan bahwa *Pulse Width Modulation* (PWM) yang dihasilkan oleh pengontrol utama telah sesuai dengan spesifikasi rentang kerja ESC. Berdasarkan hasil pengamatan menggunakan osiloskop Tektronix TBS 1102, sinyal PWM memiliki frekuensi sebesar 50,00 Hz dengan periode pulsa yang diukur pada skala waktu 2,50 ms per divisi. Karakteristik sinyal ini menjadi

parameter penentu dalam mengatur kecepatan putaran motor secara linear. Kestabilan sinyal 50 Hz ini sangat penting agar tidak terjadi *jitter* pada motor penggerak saat berada di dalam air..

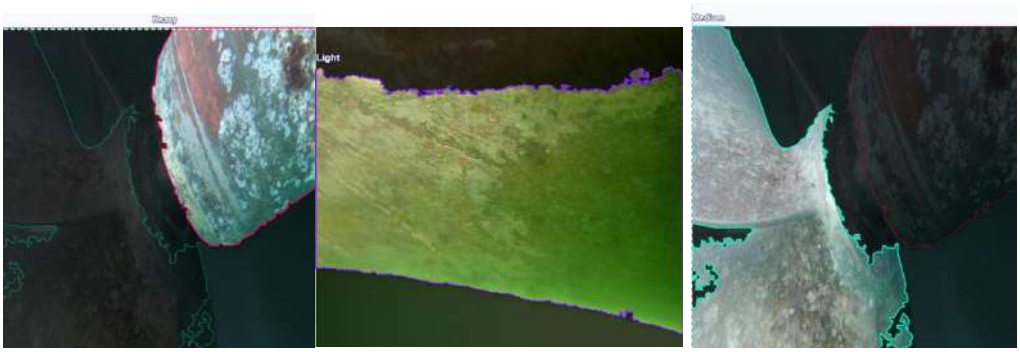


Gambar 4. 8 Pengujian PWM *Electronic Speed Control*

Berdasarkan hasil pembacaan osiloskop, sinyal PWM tersebut beroperasi pada frekuensi standar ESC 50,0001 Hz dengan amplitudo 3,3 V hingga 5 V pada skala 1,00 V/Div, serta memiliki periode siklus 20 ms dan lebar pulsa aktif antara 1,25 ms hingga 1,5 ms pada skala 2,50 ms/Div yang mengindikasikan posisi netral.

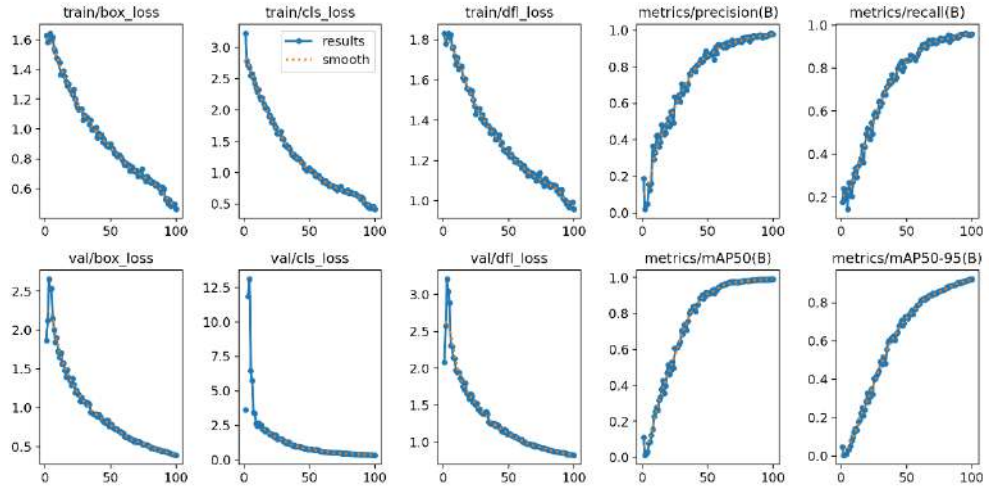
4.1.8 Pengujian Visualisasi Kamera

Pada pengujian visualisasi kamera menggunakan metode YOLO, dilakukan pengujian dengan memberikan variasi dataset yang telah diberi label sesuai dengan tiga kelas biofouling yang digunakan, yaitu Light, Medium, dan Heavy. Pengujian bertujuan untuk mengevaluasi kemampuan sistem dalam merespons perubahan kondisi visual yang ditangkap oleh kamera serta menentukan tingkat biofouling secara tepat. Dataset yang digunakan memiliki variasi kondisi, jenis, ukuran, pencahayaan, dan sudut pengambilan gambar sebagaimana ditunjukkan pada Gambar 4.9. Variasi tersebut digunakan untuk menguji ketahanan model terhadap perubahan kondisi citra selama proses operasi ROV. Hasil pengujian kemudian digunakan untuk mengetahui kemampuan kamera dalam menghasilkan citra yang dapat diproses serta kemampuan YOLO dalam mendeteksi dan mengklasifikasikan objek biofouling secara konsisten.



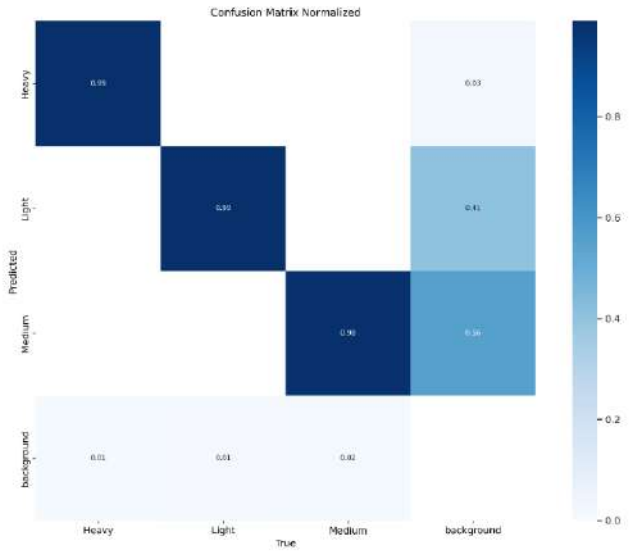
Gambar 4. 9 Dataset dan Label jenis *biofouling*

Pengujian ini dilakukan pada kamera ROV dalam kondisi didarat dengan asumsi gambar sebagai objek yang dideteksi. Proses. Proses *Training* dapat dilihat pada gambar 4.9 pengujian kamera yang diimplementasikan pada sistem visual ROV.



Gambar 4. 10 Grafik Train Result YOLO

Grafik menunjukkan perkembangan performa model YOLO selama 100 epoch pelatihan. Nilai training loss dan validation loss pada box loss, classification loss, serta DFL loss mengalami penurunan secara konsisten, menandakan proses pembelajaran berjalan dengan baik. Sementara itu, nilai precision dan recall meningkat hingga mendekati 0,97. Nilai mAP@50 mencapai hampir 0,99, sedangkan mAP@50-95 mendekati 0,91. Hasil tersebut menunjukkan model telah mencapai konvergensi dengan kemampuan deteksi dan klasifikasi objek yang baik serta konsisten.



Gambar 4. 11 Hasil *Training* YOLO V8

Berdasarkan Gambar 4.10 yang merupakan hasil pelatihan dataset yang digunakan pada sistem propulsi ROV. Hasil pengujian pada Gambar 4.10 didapatkan nilai prediksi yang cukup baik dengan angka diatas 0,8. Selain itu dari hasil pelatihan dihasilkan nilai presisi 0,87-1,00; akurasi nilai *Recall* dengan angka 0,99-0,00 dan Loss diangka 2,5.

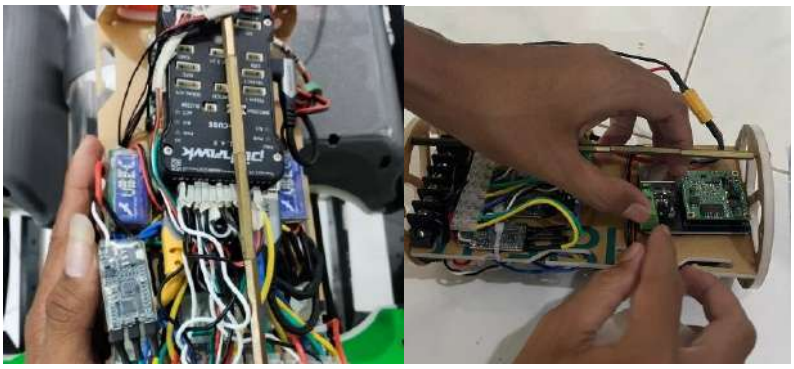
Gambar 4. 12 *Test Cam* ROV dengan dataset

4.2 Perakitan *Hardware*

Perangkat keras sistem merupakan komponen fisik yang berperan untuk menjalankan perintah dan mengolah data secara elektronik pada ROV. Perangkat keras yang digunakan untuk membangun sistem pada prototipe ROV meliputi berbagai hal seperti sensor, aktuator, raspeberry pi, pixhawk, modul ethernet, dan sistem suplai daya.

4.2.1 Instalasi Sistem Elektrik pada Model ROV

Instalasi komponen elektrikal dan sistem kendali ROV diintegrasikan sepenuhnya di dalam *top enclosure* ROV guna menjamin isolasi dari lingkungan ekosistem air sekaligus mempermudah manajemen kabel (*wiring management*)..

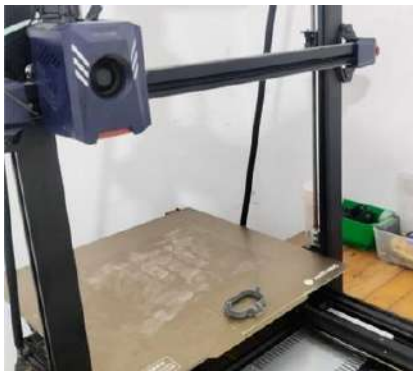


Gambar 4. 13 Perakitan sistem elektrik kontrol ROV dan kontrol motor *brush*

Pada kompartemen ini, modul kontrol utama berbasis Pixhawk ditempatkan bersama beberapa komponen pendukung yang berperan dalam pengoperasian sistem pembersihan. Komponen tersebut meliputi *Electronic Speed Controller* (ESC) yang digunakan untuk mengatur kecepatan motor sikat pembersih, modul sensor INA219/INA226 untuk melakukan pemantauan nilai arus dan tegangan secara *real-time*, serta blok terminal sebagai pusat distribusi daya ke setiap rangkaian. Seluruh komponen disusun secara modular pada akrilik pembawa (*electronic tray*) dengan mempertimbangkan keterbatasan ruang di dalam kompartemen ROV. Penataan ini bertujuan menjaga kerapian jalur kabel, mengurangi kemungkinan interferensi elektromagnetik antara jalur daya dan komunikasi, serta mempermudah proses inspeksi dan *maintenance*. Selain itu, konfigurasi tersebut memungkinkan setiap komponen dapat diakses dan diganti dengan lebih mudah apabila terjadi gangguan selama pengoperasian sistem.

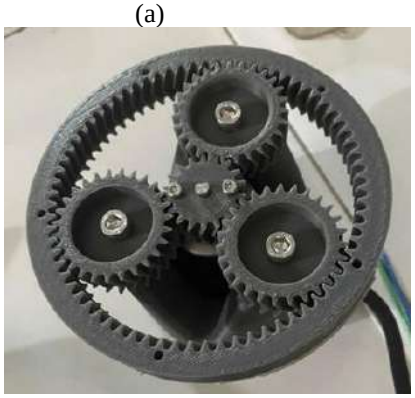
4.2.2 Instalasi *Gear Box* Motor *Brush*

Proses manufaktur dan instalasi *planetary gearbox* pada sistem motor sikat (*brush motor*) dilakukan menggunakan metode *additive manufacturing* melalui teknologi *3D printing*. Komponen dicetak menggunakan material dengan ketahanan mekanis yang memadai untuk menahan beban dan torsi selama proses pembersihan. Desain gearbox disesuaikan dengan dimensi motor dan kebutuhan mekanis sistem ROV.

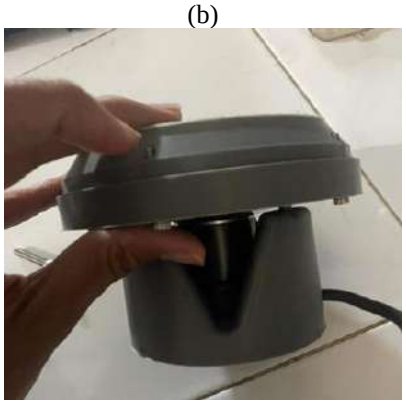


(a)

(b)



(c)



(d)

Gambar 4. 14 (a) proses printing, (b) Pemasangan Casing Motor, (c) Pemasangan Planetory, (d) Pemasangan Tutup *Gear*

Perakitan sistem transmisi ini menggunakan konfigurasi roda gigi planet (*planetary gear*) yang terdiri dari *sun gear* pada poros motor inti, tiga buah *planet gear*, dan sebuah *ring gear* eksternal sebagai pembatas dinding luar. Integrasi komponen-komponen ini dirancang secara presisi untuk mereduksi kecepatan putaran motor sekaligus meningkatkan torsi keluaran (*output torque*) yang diperlukan guna mengatasi beban gesek sikat saat melakukan pembersihan pada permukaan lambung kapal.

4.2.4 Realisasi Layout Brushing Sistem

Realisasi sistem sikat (*brush system*) pada *Remotely Operated Vehicle* (ROV) dilaksanakan dengan mengacu sepenuhnya pada perancangan desain mekanik yang telah dipaparkan pada sub-bab 3.3.10. Realisasi tata letak sistem penyikatan (*brushing system layout*) pada wahana *Remotely Operated Vehicle* (ROV) diimplementasikan secara terpusat dengan memanfaatkan ruang pada bagian bawah rangka utama (*underframe*). Penempatan komponen *planetary gearbox* terintegrasi dan aktuator sikat ini sengaja dirancang serta dipasang dengan

posisi vertikal tegak lurus terhadap sumbu horizontal wahana. Konfigurasi vertikal ini sangat krusial guna memastikan bahwa seluruh permukaan sikat pembersih dapat menapak secara tegak lurus dengan tekanan kontak (*contact pressure*) yang merata dan konstan pada permukaan lambung kapal saat proses pembersihan berlangsung.

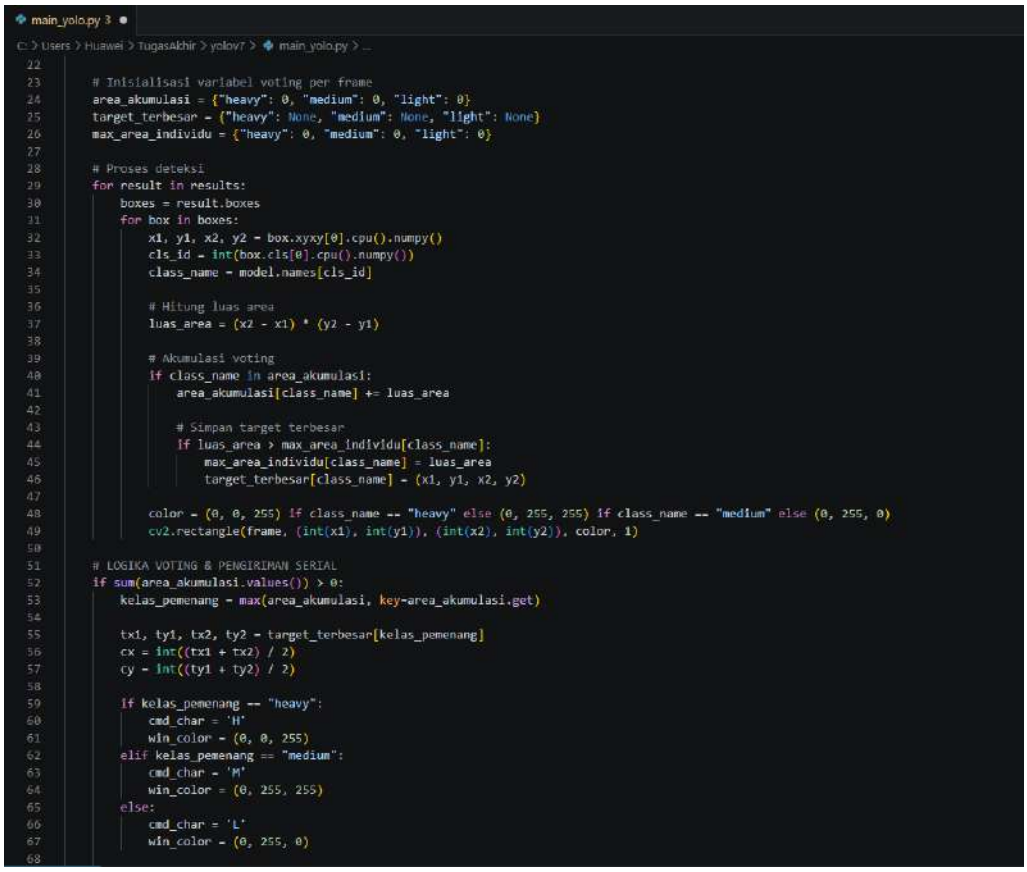
(a) (b)

Gambar 4. 15 (a) *Placement Brusher*, (b) ROV tampak depan dengan *brusher*

Berdasarkan hasil perakitan fisik dan inspeksi dimensional pasca-manufaktur, konfigurasi penempatan ini menunjukkan kesesuaian yang presisi terhadap batas ruang bebas (*clearance*) yang telah direncanakan pada tahap pemodelan 3D. Jarak aman yang tercipta memastikan bahwa operasional mekanik dari putaran sikat maupun pergerakan dinamis wahana sama sekali tidak mengganggu area pandang (*field of view*) dari kamera utama yang terletak di kompartemen depan. Selain itu, tata letak yang tersentralisasi di bawah pusat geometri ini juga dirancang agar tidak memotong atau mendistorsi lintasan aliran air (*wash/fluid slipstream*) yang dihasilkan oleh *thruster* vertikal wahana. Hal ini meminimalkan timbulnya efek turbulensi lokal di bawah lambung ROV yang dapat mengacaukan performa kontrol stabilitas wahana saat mengeksekusi manuver merapat.

4.3 Pengujian Komunikasi Serial YOLO

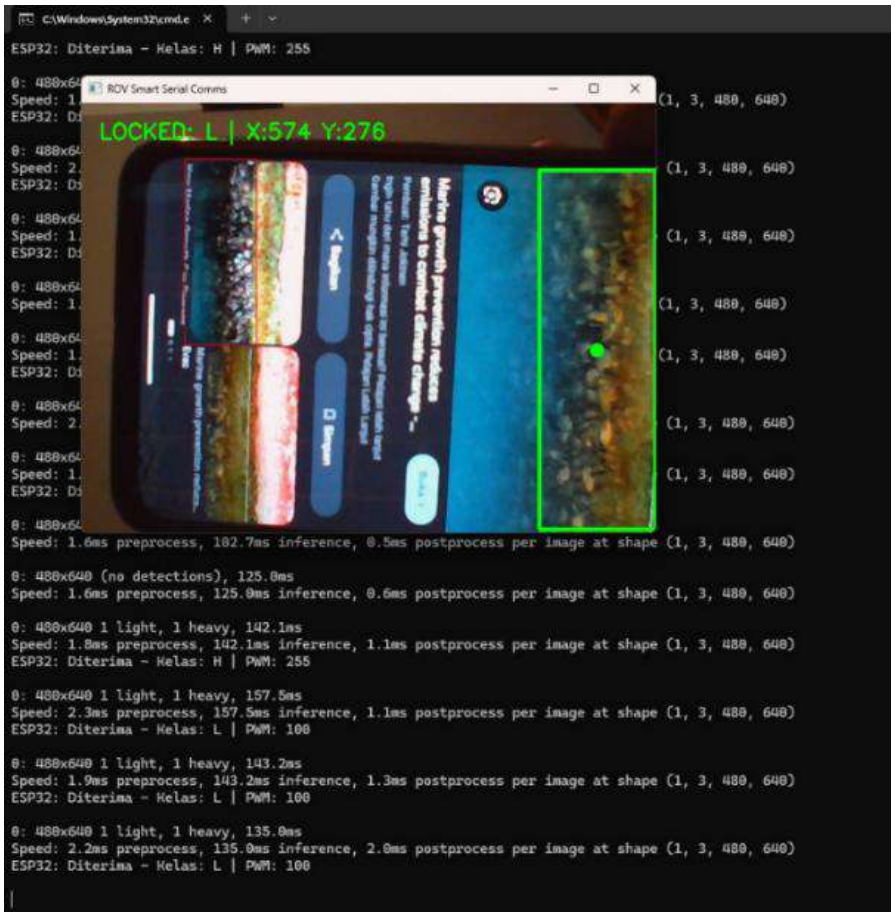
Pengujian sistem komunikasi serial dilakukan untuk memvalidasi keandalan pengiriman data hasil deteksi objek berbasis YOLOv7 dari komputer pemroses (*ground station*) menuju mikrokontroler ESP32 pada wahana.



```
main_yolo.py 3 ●
C: > Users > Huawei > TugasAkhir > yolov7 > main_yolo.py > ...
22
23      # Inisialisasi variabel voting per frame
24      area_akumulasi = {"heavy": 0, "medium": 0, "light": 0}
25      target_terbesar = {"heavy": None, "medium": None, "light": None}
26      max_area_individu = {"heavy": 0, "medium": 0, "light": 0}
27
28      # Proses deteksi
29      for result in results:
30          boxes = result.boxes
31          for box in boxes:
32              x1, y1, x2, y2 = box.xyxy[0].cpu().numpy()
33              cls_id = int(box.cls[0].cpu().numpy())
34              class_name = model.names[cls_id]
35
36              # Hitung luas area
37              luas_area = (x2 - x1) * (y2 - y1)
38
39              # Akumulasi voting
40              if class_name in area_akumulasi:
41                  area_akumulasi[class_name] += luas_area
42
43                  # Simpan target terbesar
44                  if luas_area > max_area_individu[class_name]:
45                      max_area_individu[class_name] = luas_area
46                      target_terbesar[class_name] = (x1, y1, x2, y2)
47
48              color = (0, 0, 255) if class_name == "heavy" else (0, 255, 255) if class_name == "medium" else (0, 255, 0)
49              cv2.rectangle(frame, (int(x1), int(y1)), (int(x2), int(y2)), color, 1)
50
51      # LOGIKA VOTING & PENGIRIMAN SERIAL
52      if sum(area_akumulasi.values()) > 0:
53          kelas_pemenang = max(area_akumulasi, key=area_akumulasi.get)
54
55          tx1, ty1, tx2, ty2 = target_terbesar[kelas_pemenang]
56          cx = int((tx1 + tx2) / 2)
57          cy = int((ty1 + ty2) / 2)
58
59          if kelas_pemenang == "heavy":
60              cmd_char = 'H'
61              win_color = (0, 0, 255)
62          elif kelas_pemenang == "medium":
63              cmd_char = 'M'
64              win_color = (0, 255, 255)
65          else:
66              cmd_char = 'L'
67              win_color = (0, 255, 0)
68
```

Gambar 4. 16 *Runing* YOLO dan sistem *voting* YOLO

Pengujian ini menerapkan metode *voting* berdasarkan akumulasi luas area dari kelas *Heavy*, *Medium*, hingga *Light* untuk menentukan satu kelas dominan pada setiap *frame*. Hasil deteksi YOLO kemudian diekstraksi dengan mengambil koordinat target terbesar dari kelas pemenang. Data tersebut dikonversi menjadi paket berformat CSV menggunakan pemisah koma dan *terminator newline* (\n). Setiap *frame* menghasilkan satu paket data yang dikirimkan secara berurutan agar ESP32 dapat menerima referensi posisi target secara *real-time*. Data yang diterima selanjutnya digunakan sebagai parameter masukan pada sistem kontrol PID untuk mengatur pergerakan *thruster* ROV berdasarkan posisi target. Proses transmisi dilakukan melalui komunikasi serial RS232/TTL dengan *baud rate* 115200 bps untuk menjaga kecepatan dan sinkronisasi pengiriman data. Pengujian juga dilakukan untuk memastikan paket data diterima secara konsisten, tidak terjadi kehilangan data, serta tidak terbentuk *delay* akumulatif yang dapat mengurangi respons sistem. Dengan demikian, komunikasi antara sistem visi dan kontrol dapat berlangsung stabil selama operasi ROV.



Gambar 4. 17 *Serial Monitor* pengiriman *serial* ke ESP 32

Gambar menunjukkan proses komunikasi serial antara sistem visi berbasis Raspberry Pi dan ESP32 selama proses deteksi biofouling. Pada Serial Monitor terlihat data hasil klasifikasi YOLO yang dikirimkan secara berulang, seperti "ESP32: Diterima = Kelas L" dan "ESP32: Diterima = Kelas M", yang menunjukkan bahwa ESP32 berhasil menerima kode kelas hasil deteksi. Selain informasi kelas, Serial Monitor juga menampilkan nilai kecepatan pemrosesan (preprocess, inference, dan postprocess) serta koordinat dan ukuran citra. Pada tampilan kamera, objek yang terdeteksi ditandai dengan bounding box hijau dan titik posisi target. Hasil ini menunjukkan bahwa komunikasi serial mampu meneruskan hasil klasifikasi YOLO ke ESP32 secara real-time untuk digunakan sebagai dasar penentuan parameter kendali sistem pembersihan ROV.

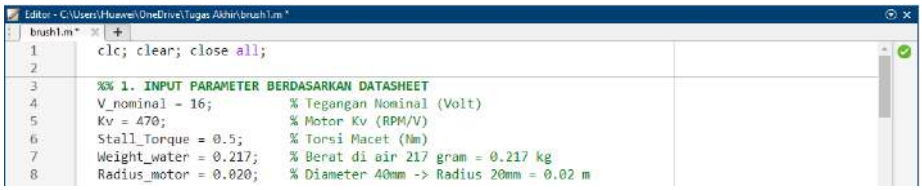
4.4 Pengujian Kontrol motor menggunakan PID

Pada sub bab ini pengujian difokuskan terhadap performa kontrol PID terhadap motor. Pengujian dilakukan dalam beberapa tahap, dimulai dengan

penentuan konstanta PID yang sesuai, dilanjutkan dengan validasi transfer function dan validasi sistem.

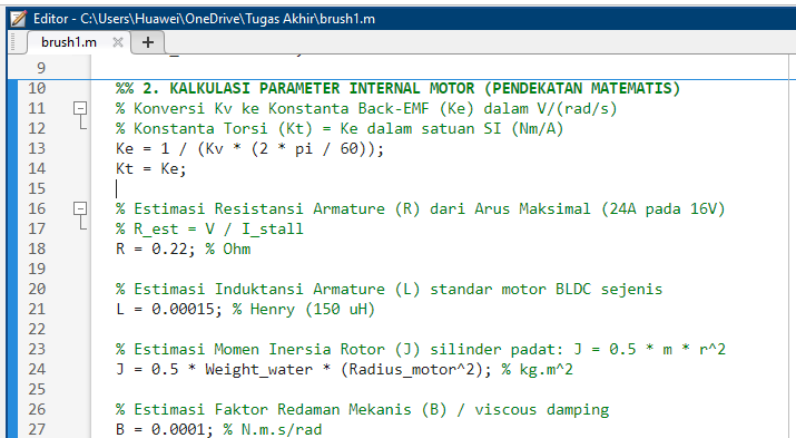
4.4.1 Penentuan Konstanta PID

Tahap awal penentuan konstanta kendali PID dilakukan dengan memasukkan parameter spesifikasi motor sikat ke dalam lingkungan simulasi MATLAB. Data teknis seperti voltase operasi, nilai KV, torsi maksimum, dan inersia massa beban dikompilasi dalam kode program untuk membentuk basis data fungsi transfer sistem aktuator. Melalui pendekatan berbasis model (*model-based design*) ini, penentuan nilai konstanta dan dapat dilakukan secara analitis untuk mempercepat proses *tuning* parameter kendali pada perangkat keras asli.



Gambar 4. 18 *Input* parameter spesifikasi motor *brush* ke dalam simulasi MATLAB

Setelah parameter eksternal berdasarkan *datasheet* ditentukan, langkah selanjutnya adalah melakukan kalkulasi parameter internal motor DC/BLDC yang akan digunakan untuk menyusun fungsi transfer sistem kendali. Parameter internal ini meliputi konstanta elektrikal, resistansi, induktansi, inersia rotor, dan faktor redaman mekanis.



Gambar 4. 19 *Input* parameter spesifikasi eksternal motor *brush* kedalam simulasi MATLAB

1. Konstanta Back-EMF (K_e) dan Konstanta Torsi (K_t)

   Konstanta Back-Electromotive Force (K_e) dihitung dengan mengonversi nilai Kv motor (RPM/V) ke dalam satuan standar SI $\left(\mathrm{V/(rad/s)}\right)$. Secara teoritis, pada sistem motor ideal, nilai konstanta torsi (K_t) dalam satuan $(\mathrm{Nm/A})$ adalah setara dengan nilai K_e. Persamaan konversi matematis yang diintegrasikan ke dalam skrip MATLAB adalah sebagai berikut:

$$K_e = \frac{1}{K_v \times \left(\frac{2\pi}{60}\right)}$$

$$K_t = K_e \tag{4.1}$$

2. Estimasi Resistansi (R) dan Induktansi Armature (L)

   Resistansi armature (R) diestimasi berdasarkan kondisi arus macet maksimum (*stall current*) saat motor diberi tegangan nominal, menggunakan pendekatan Hukum Ohm $R = \frac{V}{I_{stall}}$. Berdasarkan spesifikasi arus 24A pada tegangan16V, diperoleh nilai resistansi internal sebesar $0{,}22\ \Omega$. Sementara itu, nilai induktansi armature (L) ditentukan sebesar $0{,}00015H$ $(150\mu H)$, yang merupakan nilai standar karakteristik elektrikal kumparan untuk motor BLDC sejenis.

3. Estimasi Momen Inersia Rotor (J) dan Faktor Redaman Mekanis (B)

   Untuk memodelkan respons mekanis komponen berputar, nilai momen inersia rotor (J) didekati dengan rumus silinder padat. Perhitungan ini melibatkan variabel berat efektif komponen di air $(m = 0{,}217kg)$serta radius fisik motor $r = 0{,}02\ m)$ dengan persamaan:

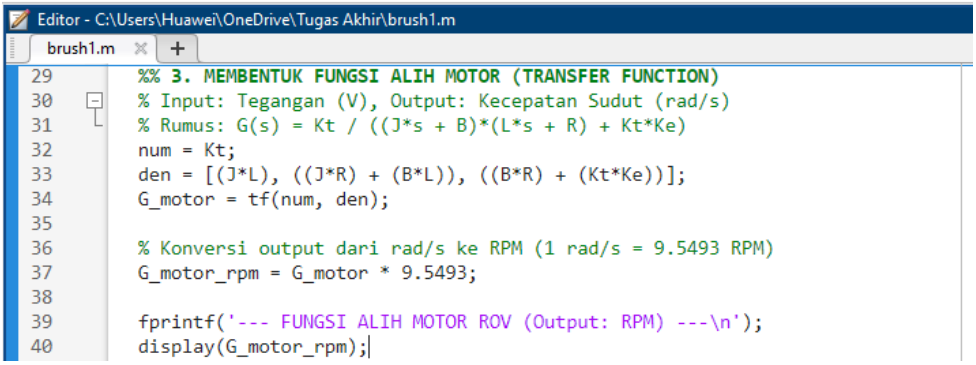
$$J = \frac{1}{2} \times m \times r^2 \tag{4.2}$$

$$J = 0{,}5 \times 0{,}217 \times (0{,}02)^2 = 4{,}34 \times 10^{-5\ \mathrm{kg}} \cdot \mathrm{m}^2$$

Selanjutnya, faktor redaman mekanis (viscous damping coefficient, B) ditetapkan melalui pendekatan empiris sebesar $0{,}0001\ \mathrm{N \cdot m \cdot s/rad}$. Nilai B ini merepresentasikan redaman gesek mekanis internal dan beban fluida minimal saat poros motor berputar.

4.4.2 Penentuan *Transfer Function* Motor

Pembentukan fungsi alih (*transfer function*) dilakukan untuk mentransformasikan model matematika sistem motor sikat dari domain waktu ke dalam domain s (Laplace) menggunakan perangkat lunak MATLAB.



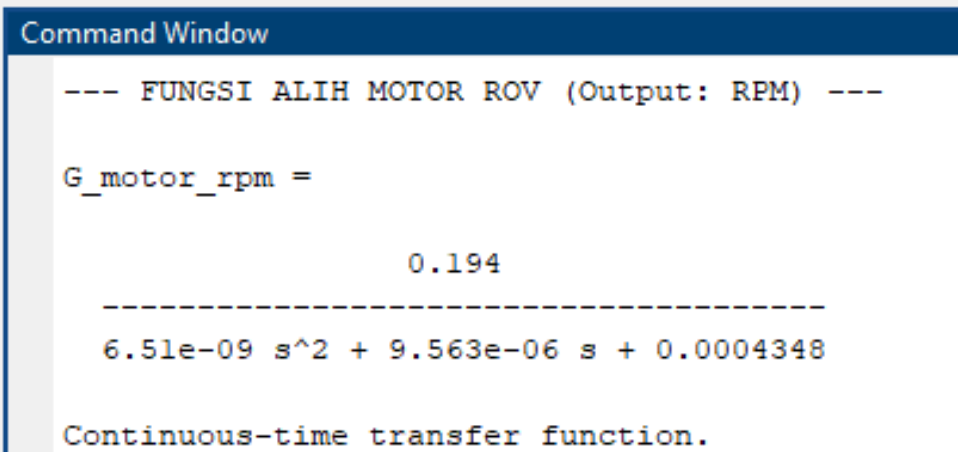
Gambar 4. 20 Membentuk *Transfer Function* di MATLAB

Berdasarkan datasheet dan pengolahan *transfer function* yang ditunjukkan pada Gambar 4.19, parameter masukan (*input*) sistem ditetapkan berupa tegangan listrik V dalam satuan Volt, sedangkan parameter keluaran (*output*) yang dimodelkan adalah kecepatan sudut poros (ω) dalam satuan radian per sekon $(\mathrm{rad/s})$. Penyusunan persamaan fungsi alih ini didasarkan pada substitusi nilai parameter internal mekanis dan elektrikal yang telah dikalkulasi sebelumnya ke dalam bentuk polinomial pembilang (*numerator*) dan penyebut (*denominator*) dengan rumus dasar:

$$G(s) = \frac{K_t}{(J \cdot L)s^2 + (J \cdot R + B \cdot L)s + (B \cdot R + K_t \cdot K_e)} \qquad (3.2)$$

Agar pemodelan ini merepresentasikan kondisi operasional riil di lapangan, nilai luaran fungsi alih yang semula bersatuan $\mathrm{rad/s}$ dikonversi secara matematis ke dalam satuan Revolutions Per Minute (RPM) dengan mengalikan sistem menggunakan nilai konstanta pengali sebesar 9,5493. Melalui pengeksekusian fungsi tf(num, den) pada kode program tersebut, dihasilkan sebuah fungsi alih linier ordo dua terkompensasi (G_{motor_rpm}) yang mendeskripsikan karakteristik respons transien kecepatan motor secara presisi. Representasi matematis akhir inilah yang bertindak sebagai basis model tanaman (*plant*) utama dalam menentukan serta

mensimulasikan parameter kendali optimal pada tahap perancangan konstanta PID selanjutnya.



Gambar 4. 21 *Transfer Function* dengan output RPM

Berdasarkan pengeksekusian skrip matematika kalkulasi parameter internal yang telah disubstitusikan ke dalam formula sistem, diperoleh nilai fungsi alih kontinu (*continuous-time transfer function*) nyata dari *plant* motor sikat ROV. Sebagaimana yang ditunjukkan pada hasil *output Command Window* MATLAB pada Gambar 4.X, representasi linier ordo dua dari model motor penggerak dengan parameter keluaran bersatuan RPM (G_{motor_rpm}) menghasilkan persamaan matematis terkompensasi sebagai berikut:

$$G_{motor_rpm}(s) = \frac{0{,}194}{6{,}51 \times 10^{-9}s^2 + 9{,}563 \times 10^{-6}s + 0{,}0004348}$$

Nilai konstanta pada pembilang (*numerator*) sebesar 0,194 merepresentasikan penguatan keseluruhan sistem yang sudah terkonversi ke domain kecepatan putar RPM. Sementara itu, nilai koefisien pada penyebut (*denominator*) yang terdiri dari nilai $6{,}51 \times 10^{-9}$ untuk orde s^2 dan $9{,}563 \times 10^{-6}$ untuk orde s merepresentasikan interaksi dinamis antara inersia rotor mekanis dan induktansi elektrikal kumparan motor di dalam air. Fungsi alih dengan nilai riil eksponensial ini menjadi cetak biru (*blueprint*) mutlak yang akan diintegrasikan ke dalam blok kendali *closed-loop* Simulink guna menguji kestabilan respons transien sistem sebelum konstanta PID diimplementasikan pada perangkat keras ROV.

4.4.3 Validasi Transfer Function

Subbab ini membahas proses validasi transfer function yang telah diperoleh melalui pemodelan sistem dengan membandingkan hasil simulasi terhadap implementasi aktual. Validasi dilakukan pada tiga kondisi operasi, yaitu kelas Light, Medium, dan Heavy, dengan parameter yang dibandingkan meliputi nilai PWM, kecepatan putar motor pada kondisi tanpa beban (RPM Darat), kecepatan putar saat sistem beroperasi di dalam air (RPM Laut), serta setpoint yang ditentukan oleh sistem kontrol. Pengujian ini bertujuan untuk memastikan bahwa model yang dikembangkan mampu merepresentasikan karakteristik dinamis motor setelah dipengaruhi oleh reduksi planetary gearbox dan beban hidrodinamika akibat massa jenis air.

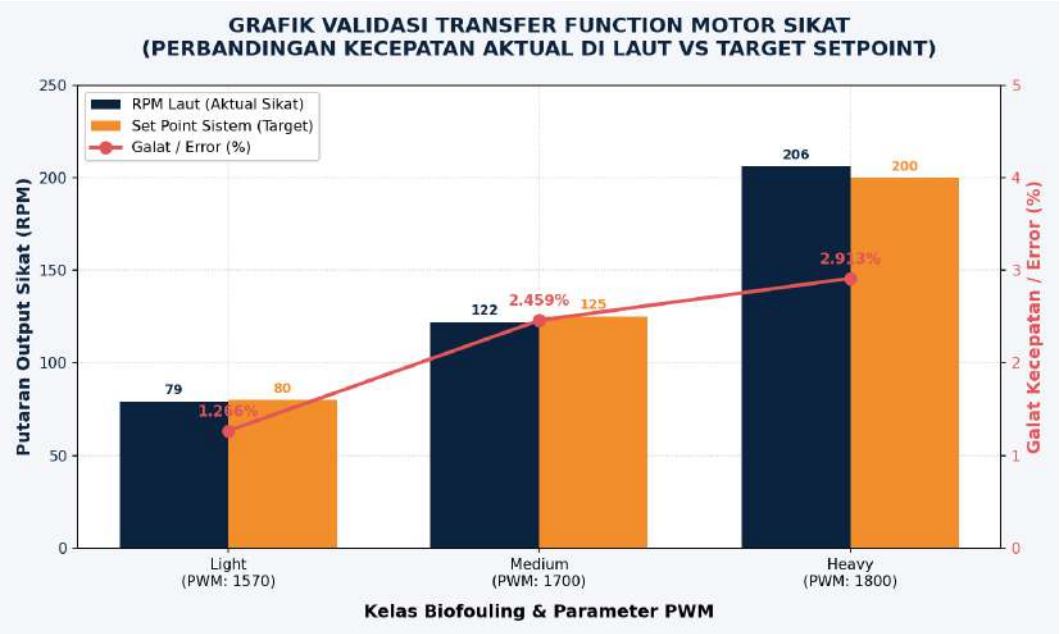
Tabel 4. 6 Hasil Validasi Transfer Function

Validasi Transfer Function					
Kelas	PWM	RPM Darat	RPM Laut	Set Point Sistem	Error %
Light	1570	1580	79	80	1.2658228
Medium	1700	2440	122	125	2.4590164
Heavy	1800	4120	206	200	2.9126214

NB	
RPM Darat	: Pure Kecepatan Motor
RPM Laut	: Reduksi Gear Box dan Massa Jenis Air

Berdasarkan hasil validasi pada Tabel 4.6 performa dari transfer function menunjukkan bahwa pengaturan *Pulse With Modulation* mampu menghasilkan kecepatan motor yang mendekati setpoint pada setiap kelas biofouling. Pada kelas *Light*, PWM sebesar 1570 µs menghasilkan kecepatan motor tanpa beban sebesar 1580 RPM, yang setelah melalui reduksi gearbox dan pengaruh media air menjadi 79 RPM. Nilai tersebut mendekati setpoint 80 RPM dengan galat sebesar 1,27%. Pada kelas *Medium*, PWM 1700 µs menghasilkan 2440 RPM tanpa beban dan 122 RPM dalam air, dengan galat 2,46% terhadap setpoint 125 RPM. Sementara itu, kelas *Heavy* menggunakan PWM 1800 µs dengan kecepatan 4120 RPM tanpa beban dan 206 RPM dalam air, menghasilkan galat 2,91% terhadap setpoint 200 RPM. Hasil tersebut menunjukkan pengaturan transfer function sudah sesuai

dengan yang semestinya, namun hasil diatas apabila dibandingkan dengan datasheet motor yang seharusnya terdapat sedikit galat. Hal itu diperhitungkan untuk variable bebas dari biofouling dan laut yang tidak bisa masuk kedalam model..



Gambar 4. 22 Grafik Hasil Validasi Transfer Function dan penentuan Set Point

Secara keseluruhan, seluruh hasil pengujian menunjukkan nilai galat di bawah 3%, sehingga dapat disimpulkan bahwa transfer function yang digunakan mampu merepresentasikan hubungan antara sinyal PWM dan kecepatan putar brush dengan tingkat akurasi yang baik. Nilai RPM Darat merupakan hasil pengukuran aktual (real measurement) pada motor BLDC menggunakan tachometer sebelum diintegrasikan dengan sistem pengendali PI, planetary gearbox, serta sebelum motor menerima beban hidrodinamika akibat media air. Data tersebut digunakan sebagai karakteristik dasar motor (*open-loop characteristic*) dalam proses identifikasi sistem dan penyusunan transfer function. Selanjutnya, nilai tersebut dikonversi menjadi RPM Laut melalui perhitungan yang mempertimbangkan pengaruh rasio reduksi planetary gearbox sebesar 5:1, efisiensi transmisi mekanis, serta peningkatan beban akibat massa jenis air. Perbedaan antara RPM Darat dan RPM Laut menunjukkan bahwa penurunan putaran tidak hanya dipengaruhi oleh rasio transmisi, tetapi juga oleh karakteristik hidrodinamika lingkungan operasi.

Oleh karena itu, model transfer function yang telah divalidasi pada kondisi operasi aktual dinilai lebih representatif sebagai dasar perancangan pengendali PI dibandingkan apabila hanya menggunakan karakteristik motor tanpa beban, karena telah mengakomodasi pengaruh sistem mekanis dan lingkungan kerja yang sesungguhnya.

4.4.4 Tuning *Gain Scheduling* PID Berdasarkan Tiga Profil Beban

Langkah berikutnya setelah memperoleh fungsi alih linier aktual adalah menentukan konstanta PID menggunakan pendekatan *Gain Scheduling*. Metode ini diterapkan untuk menyesuaikan parameter kendali terhadap perubahan karakteristik sistem. Pendekatan tersebut diperlukan karena beban mekanis motor dan tingkat kekotoran lambung (*biofouling*) bersifat dinamis, sehingga kebutuhan respons kendali dapat berbeda pada setiap kondisi operasi.

4.4.4.1 Tuning Gain PID menggunakan Auto Tune

Berdasarkan hasil *transfer function*, proses *tuning* otomatis dilakukan menggunakan fungsi bawaan *pidtune()* pada MATLAB dengan membagi wilayah kerja motor sikat menjadi tiga profil operasional, yaitu *Light*, *Medium*, dan *Heavy*. Setiap profil memiliki karakteristik penguatan kendali yang disesuaikan dengan kebutuhan kecepatan dan beban pembersihan. Pengondisian adaptif ini memungkinkan parameter PID berubah secara otomatis berdasarkan kelas biofouling yang dikirimkan sistem visi setelah proses *voting*, sehingga respons motor dapat disesuaikan dengan kondisi permukaan lambung.

Editor - C:\Users\Huawei\OneDrive\Tugas Akhir\brush1.m

brush1.m

```
42  %% 4. TUNING GAIN SCHEDULING PID UNTUK 3 PROFIL
43  % fungsi pidtune()
44  % PROFIL 1: LIGHT (Target 1000 RPM, Cenderung respons lembut/hemat energi)
45  % Crossover frequency (wc)
46  wc_light = 15;
47  [C_light, info_light] = pidtune(G_motor_rpm, 'PID', wc_light);
48
49  % PROFIL 2: MEDIUM (Target 1600 RPM, Respons kecepatan sedang)
50  wc_medium = 30;
51  [C_medium, info_medium] = pidtune(G_motor_rpm, 'PID', wc_medium);
52
53  % PROFIL 3: HEAVY (Target 2000 RPM, Respons kecepatan maksimum)
54  wc_heavy = 60;
55  [C_heavy, info_heavy] = pidtune(G_motor_rpm, 'PID', wc_heavy);
```

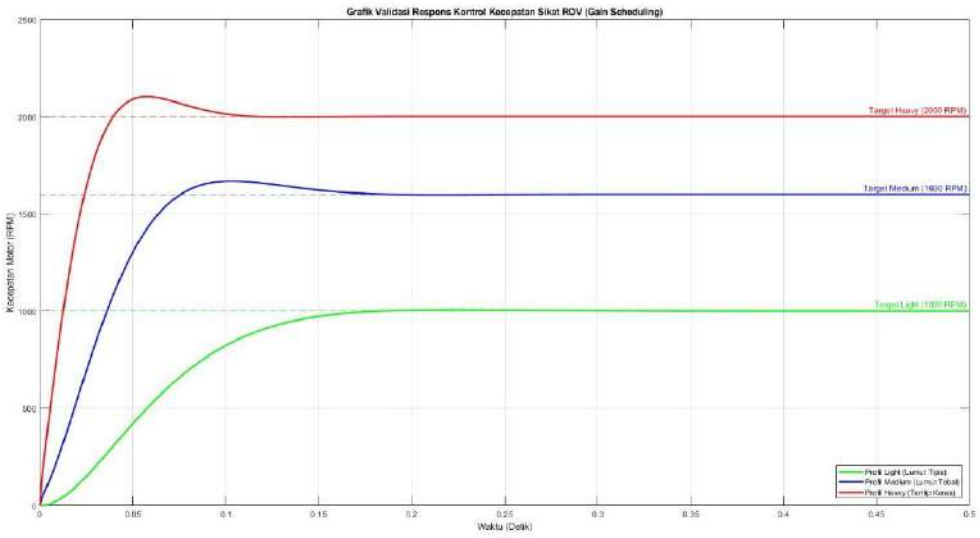
Gambar 4. 23 Penentuan Tuning Gain Scheduling PID untuk tiga profil

Untuk memperoleh karakteristik respons sistem yang berbeda pada setiap kondisi kerja, dilakukan manipulasi nilai frekuensi penyilangan (*crossover frequency*) atau ωc sebagai salah satu parameter acuan utama dalam proses kompensasi dan penalaan kendali. Pemilihan frekuensi penyilangan disesuaikan dengan tingkat biofouling yang terdeteksi serta kebutuhan kecepatan putar sikat. Semakin berat kondisi biofouling, semakin tinggi respons kendali yang dibutuhkan agar motor mampu mempertahankan putaran ketika menerima peningkatan beban mekanis.

Profil 1 (*Light*) ditujukan untuk kondisi biofouling ringan dengan kecepatan putar sikat sebesar 80 RPM. Frekuensi penyilangan digunakan pada tingkat rendah untuk menghasilkan respons transien yang lebih lembut (*smooth*). Pengaturan ini bertujuan meminimalkan *overshoot* dan osilasi kecepatan serta menjaga konsumsi energi baterai tetap efisien selama proses penyikatan awal.

Profil 2 (*Medium*) digunakan untuk kondisi biofouling sedang dengan kecepatan putar sikat sebesar 120 RPM. Frekuensi penyilangan dinaikkan dibandingkan profil *Light* sehingga sistem memiliki respons kecepatan yang lebih cepat dan moderat. Pengaturan tersebut diperlukan untuk mempertahankan kestabilan putaran ketika motor mulai menerima pembebanan mekanis yang lebih besar akibat kontak sikat dengan permukaan lambung kapal.

Profil 3 (*Heavy*) ditujukan untuk kondisi biofouling berat dan keras, khususnya teritip, dengan kecepatan putar maksimum sebesar 200 RPM. Frekuensi penyilangan ditetapkan paling tinggi agar sistem menghasilkan respons transien yang agresif dan waktu naik (*rise time*) yang lebih cepat. Karakteristik ini diperlukan agar motor mampu menghasilkan respons torsi yang cepat terhadap peningkatan beban, sehingga risiko penurunan RPM secara signifikan dan *motor stall* akibat resistansi mekanis dapat diminimalkan. Dengan demikian, ketiga profil kendali tersebut disusun berdasarkan tingkat kekotoran dan kebutuhan respons aktuator.



Gambar 4. 24 Hasil Tuning *Gain Scheduling* PID Berdasarkan Tiga Profil Beban

Berdasarkan hasil simulasi *closed-loop step response* yang disajikan pada Gambar 4.22, implementasi metode *Gain Scheduling* PID berhasil memvalidasi performa kendali kecepatan motor sikat sesuai dengan karakteristik target masing-masing profil operasional, menghasilkan nilai Kp : 0.01, Ki : 0.005 dan Kd : 0. Pengujian dilakukan secara simultan dalam domain waktu dari 0 hingga 0,5 detik untuk mengamati parameter respons transien, meliputi *rise time*, *overshoot*, dan *settling time*. Grafik tersebut membuktikan secara visual bahwa modifikasi frekuensi penyilangan (ω_c) pada tahap *tuning* sebelumnya menghasilkan karakteristik kendali yang adaptif dan stabil saat menuju ke nilai referensi (*setpoint*). Analisis detail terhadap kurva respons transien pada masing-masing profil pengondisian beban dijabarkan sebagai berikut:

- Profil *Light* (Kurva Hijau): Respons sistem menunjukkan karakteristik *overdamped* atau sangat halus dengan target $1000\,\mathrm{RPM}$ yang diperuntukkan bagi pembersihan lumut tipis. Sesuai dengan perancangan awal yang menggunakan frekuensi penyilangan rendah $\omega_c = 15$, kurva bergerak stabil tanpa menghasilkan lonjakan nilai (*zero overshoot*). Sistem mencapai kondisi tunak (*steady state*) sepenuhnya pada waktu (*settling time*) sekitar 0,2 detik. Ketiadaan *overshoot* ini sangat menguntungkan untuk menghemat konsumsi energi baterai pada fase awal operasi penyikatan.

- Profil *Medium* (Kurva Biru): Respons sistem menunjukkan karakteristik *underdamped* moderat dengan target kecepatan 1600 RPM untuk mengatasi beban lumut tebal. Kurva memiliki waktu bangkit (*rise time*) yang lebih cepat dibanding profil *light*, dengan memunculkan sedikit *overshoot* terkendali di kisaran 1670 RPM sebelum akhirnya melandai dan mencapai *steady state* stabil pada detik ke-0,18. Respons transien yang cukup cekatan ini menjamin sikat tidak kehilangan momentum putar secara drastis saat mendapati perubahan kekentalan lapisan biofouling.

Profil *Heavy* (Kurva Merah): Respons sistem menunjukkan karakteristik yang sangat agresif dengan target kecepatan maksimum 2000 RPM untuk membersihkan kotoran keras seperti teritip. Karena dikompensasi dengan frekuensi penyilangan tinggi $\omega_c = 60$, kurva ini menghasilkan *rise time* tercepat, di mana motor mampu melonjak mendekati target hanya dalam waktu kurang dari 0,04 detik. Konsekuensi dari agresivitas ini adalah timbulnya *overshoot* tertinggi mencapai kisaran 2100 RPM pada detik ke-0,06. Meskipun demikian, algoritma kendali PID mampu meredam osilasi secara instan, sehingga kurva langsung mengunci posisi stabil tepat pada nilai referensi 2000 RPM dalam waktu kurang dari 0,15 detik. Kecepatan respons instan ini menjadi parameter kritis untuk mencegah terjadinya kemacetan mekanis (*stalling*) akibat torsi lawan yang ekstrem dari teritip keras pada lambung kapal.

4.4.4.2 Tuning Gain PID menggunakan Ziegler Nichole

Berdasarkan proses identifikasi sistem, diperoleh fungsi alih motor brush sebagai $G(s) = \frac{0.194}{6.51\times10^{-9}s^2+9.563\times10^{-6}s+0.0004348}$. Untuk mempermudah analisis karakteristik dinamis, pembilang dan penyebut dinormalisasi terhadap konstanta 0,0004348 sehingga diperoleh bentuk standar sistem orde dua, yaitu $G(s) = \frac{446.182}{1.497\times10^{-5}s^2+0.02199s+1}$. Dari fungsi alih tersebut diperoleh nilai penguatan keadaan tunak (*steady-state gain*) sebesar 446,182, rasio redaman (ζ) sebesar 2,842, yang menunjukkan bahwa sistem termasuk kategori overdamped (ζ>1). Selain itu, sistem memiliki dua pole riil negatif, yaitu $s1 = -46{,}969\ rad/s$ dan $s2 = -1422{,}002\ rad/s$, yang menandakan sistem bersifat stabil tanpa osilasi. Nilai pole

tersebut menghasilkan konstanta waktu masing-masing sebesar 0,02129 s dan 0,000703 s, di mana pole pertama menjadi pole dominan yang menentukan respons transien sistem, sedangkan pole kedua memiliki dinamika yang jauh lebih cepat sehingga pengaruhnya relatif kecil terhadap respons keseluruhan. Oleh karena itu, fungsi alih dapat dinyatakan kembali dalam bentuk faktor konstanta waktu, yaitu

$$G(s) = \frac{446.182}{(0.02199s+1)(0.000703s+1)}.$$

Bentuk ini mempermudah analisis respons sistem serta menjadi dasar dalam proses perancangan dan penalaan pengendali PI agar kecepatan putar brush mampu mengikuti nilai setpoint secara stabil tanpa menghasilkan osilasi yang berlebihan.

Parameter awal pengendali diperoleh menggunakan metode reaksi proses (Process Reaction Curve) dengan pendekatan Ziegler–Nichols Metode I. Pendekatan ini dilakukan dengan membentuk garis singgung pada titik belok (inflection point) dari respons anak tangga (step response) sistem untuk menentukan karakteristik dinamis proses berupa apparent dead time (L) dan apparent time constant (T). Berdasarkan hasil identifikasi, diperoleh nilai apparent dead time sebesar 0,000553 detik dan apparent time constant sebesar 0,023921 detik. Kedua parameter tersebut selanjutnya digunakan sebagai dasar penerapan aturan penalaan (tuning rules) Ziegler–Nichols Metode I untuk menentukan nilai awal parameter pengendali PI, sehingga sistem memiliki karakteristik respons yang sesuai sebelum dilakukan proses penyempurnaan (fine tuning) berdasarkan hasil pengujian aktual.

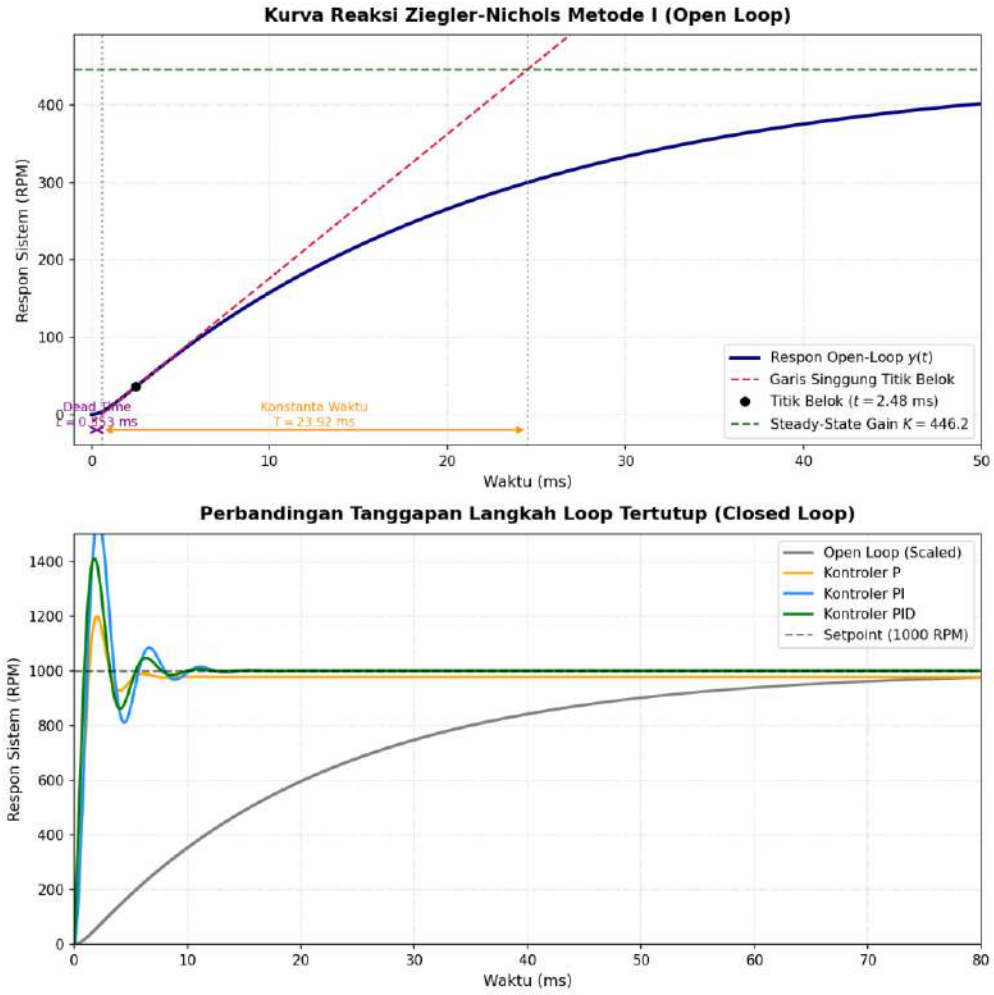
Tabel 4. 7 Hasil Konstanta PID Aturan ZN 1 berdasarkan Garis Singgung

Tipe Kontroler	Formula Z-N	Parameter Hasil Tuning
Tipe P	$Kp = (T/K \cdot L)$	Kp =0,09695
Tipe PI	Kp =0,9$(T/K \cdot L)$	Kp =0,08725
	$Ti = L$/0,3	Ki =47,3357
Tipe PID	Kp =1,2$(T/K \cdot L)$	Kp =0,11634
	Ti =2L	Ki =105,1904
	Td =0,5L	Kd =0,000032

Berdasarkan hasil identifikasi fungsi alih, diperoleh dua konstanta waktu dengan perbedaan orde magnitudo yang cukup signifikan, yaitu T1=0,021290 detik dan T2=0,000703 detik, sehingga memenuhi kondisi T1≫T2. Perbedaan nilai tersebut menunjukkan bahwa sistem memiliki dua karakteristik dinamika, yaitu dinamika lambat yang menjadi respons dominan dan dinamika cepat yang memberikan pengaruh relatif kecil terhadap respons keseluruhan. Untuk menyederhanakan representasi sistem, dilakukan pendekatan reduksi menjadi model *First Order Plus Dead Time* (FOPDT). Dalam pendekatan ini, konstanta waktu tercepat T2 diasumsikan sebagai *apparent dead time* (L), sedangkan konstanta waktu paling lambat T1 digunakan sebagai *apparent time constant* (T). Berdasarkan pendekatan tersebut diperoleh nilai L=0,000703 detik dan T=0,021290 detik. Kedua parameter tersebut selanjutnya digunakan sebagai dasar penerapan metode penalaan Ziegler–Nichols Tipe I untuk memperoleh nilai awal parameter pengendali PI. Penggunaan model FOPDT memungkinkan karakteristik dinamis motor *brush* direpresentasikan dalam bentuk yang lebih sederhana, tetapi tetap mempertahankan perilaku dominan sistem yang diperlukan dalam proses perancangan kendali. Hasil penalaan awal kemudian menjadi acuan dalam proses *tuning* lanjutan menggunakan pendekatan *gain scheduling*. Penyesuaian tersebut diperlukan karena karakteristik beban motor berubah sesuai tingkat biofouling yang diklasifikasikan menjadi *Light*, *Medium*, dan *Heavy*. Dengan demikian, parameter kendali yang diperoleh tidak hanya mempertimbangkan respons teoritis fungsi alih, tetapi juga kebutuhan respons motor pada kondisi operasi yang berbeda. Pendekatan ini diharapkan mampu menghasilkan respons kecepatan yang stabil, cepat, dan sesuai dengan karakteristik beban selama proses pembersihan lambung kapal.

Tabel 4. 8 Hasil Tuning PID menggunakan aturan ZN1 berdasarkan Konstanta Waktu

Tipe Kontroler	Parameter Hasil Tuning
Tipe P	*Kp*=0,06788
Tipe PI	*Kp*=0,06109, *Ki*=26,0700
Tipe PID	*Kp*=0,08146, *Ki*=57,9400, *Kd*=0,000029



Gambar 4. 25 Grafik Hasil Tuning PID menggunakan ZN

4.5.4 Realisasi Metode Tanpa Beban

Tahap pengujian tanpa beban difokuskan untuk mengamati linearitas perubahan *duty cycle* pada motor sikat saat menerima perintah *setpoint* otomatis dari sistem visi komputer. Pengujian ini guna menguji respon transien dari penalaan parameter PID secara nyata di darat dengan memperhatikan arus maksimal.

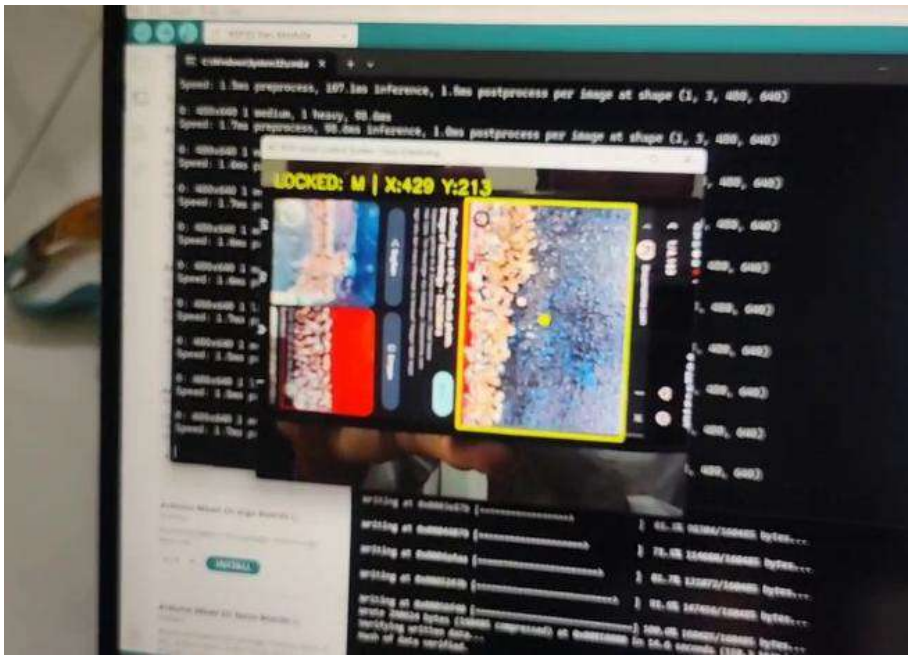
Tabel 4. 9 Perbandingan Metode PID

Parameter Kinerja	Trial & Error (Simulasi MATLAB)	Ziegler-Nichols (PID)
Nilai Parameter	*Kp* =0,01,*Ki* =0,005	*Kp* =0,1163, *Ki* =105,19, *Kd*=0,000032
Rise Time (tr*tr*)	42,52 ms	0,72 ms
Overshoot (Mp*Mp*)	4,08%	41,15%
Settling Time (ts,2%*ts*,2%)	118,40 ms	7,24 ms

Berdasarkan Tabel 4.9, metode penalaan parameter PID konvensional, yaitu Ziegler–Nichols Metode I, mampu menghasilkan respons transien yang sangat cepat dengan *rise time* sebesar 0,72 ms. Namun, penerapan parameter tersebut secara langsung pada motor sikat M200 ROV menghasilkan *overshoot* yang relatif tinggi, yaitu sebesar 41,15%. Kondisi tersebut kurang sesuai untuk pengoperasian ROV di bawah air karena percepatan motor yang terlalu agresif dapat menimbulkan lonjakan arus mula (*inrush current*) dan menyebabkan penurunan tegangan pada bus baterai (*voltage sag*). Selain itu, perubahan torsi yang terjadi secara tiba-tiba dapat menghasilkan torsi reaksi pada sasis ROV sehingga berpotensi mengganggu kestabilan posisi dan orientasi wahana ketika melakukan proses pembersihan.

Untuk mengatasi permasalahan tersebut, penelitian ini melakukan modifikasi parameter *tuning* dengan mengombinasikan batas analitis Ziegler–Nichols dengan proses *auto-tuning* menggunakan MATLAB serta pendekatan *trial and error*. Hasil penyesuaian parameter dengan nilai Kp = 0,01 dan Ki = 0,005 mampu menurunkan *overshoot* secara signifikan menjadi 4,08%. Meskipun menghasilkan respons yang lebih lambat dengan *rise time* pengujian di darat sebesar 42,52 ms, karakteristik tersebut lebih sesuai untuk aplikasi ROV karena perubahan kecepatan berlangsung secara lebih terkendali. Respons yang lebih landai membantu mengurangi lonjakan arus dan torsi reaksi sehingga kestabilan ROV selama motor sikat mulai beroperasi dapat dipertahankan. Penurunan *overshoot* juga menunjukkan bahwa parameter hasil modifikasi lebih mampu menghindari perubahan RPM yang berlebihan terhadap nilai *setpoint*.

Hasil pengujian selanjutnya menunjukkan keberhasilan penerapan algoritma *Gain Scheduling* PID. Ketika sistem visi mendeteksi tingkat *biofouling* rendah (*Light/L*), motor sikat beroperasi pada setpoint kecepatan yang lebih rendah dan stabil. Ketika kelas *Medium/M* terdeteksi, sistem memberikan perubahan setpoint yang lebih tinggi sesuai parameter kendali yang telah ditentukan. Sinkronisasi antara hasil klasifikasi YOLO, perubahan setpoint, respons RPM, dan data *feedback* yang ditampilkan melalui *Command Prompt* menunjukkan bahwa proses komunikasi dan pengendalian telah terintegrasi dengan baik. Dengan demikian, parameter PID hasil modifikasi memberikan hasil yang lebih baik antara respons, kestabilan, dan kebutuhan daya motor pada sistem pembersihan ROV.



Gambar 4. 26 Pengujian sistem kontrol *Open Loop*

Siklus aliran data dimulai ketika kamera mengambil frame lingkungan secara real-time dan mengirimkan citra tersebut ke stasiun kendali untuk diproses. Citra yang diterima kemudian dianalisis menggunakan algoritma YOLOv7 dengan kecepatan inferensi rata-rata sebesar 99,0 ms per frame. Hasil deteksi digunakan untuk mengidentifikasi tingkat kekotoran permukaan, kemudian kelas dominan ditentukan melalui mekanisme voting berdasarkan akumulasi luas area deteksi. Setelah kelas tingkat kekotoran diperoleh, informasi tersebut segera dikonversi menjadi paket data berbentuk string terformat dengan struktur pesan yang ringkas, yaitu "[cmd_char],[cx],[cy]\n". Sebagai contoh, pesan "M,446,215\n" menunjukkan kelas Medium dengan posisi koordinat pusat objek pada titik tertentu. Komunikasi antara stasiun kendali dan ESP32 menerapkan metode frame-driven communication, yaitu hanya satu paket data yang dikirim untuk setiap frame hasil pemrosesan. Pembatasan jumlah paket ini efektif mengurangi latensi komunikasi dan mencegah penumpukan data pada buffer ESP32. Dengan demikian, perubahan kondisi visual dapat segera direspons melalui perubahan perintah kontrol, sehingga perubahan frekuensi motor dan kecepatan mekanis sikat berlangsung secara simultan. Metode ini juga menjaga sinkronisasi antara proses deteksi, komunikasi, dan eksekusi kontrol secara keseluruhan.

4.5 Hasil dan Pembahasan

Sub bab ini sepenuhnya akan membahas mengenai hasil penelitian dan anlisis sistem beserta perhitungan performa dari setiap bagan.

4.5.1 Hasil Implementasi *Manuver Holding* Vertikal dan Horizontal

Pengujian kontrol *holding* pada sumbu vertikal (y-axis) dilakukan untuk mengevaluasi keandalan algoritma kendali dalam mempertahankan posisi kedalaman (*depth holding*) dan orientasi ROV agar tetap stabil menghadap vertikal. ROV melakukan operasi pembersihan lambung kapal, di mana gaya aksi-reaksi dari putaran sikat mekanis serta efek daya apung (*buoyancy*) secara konstan terus memberikan gangguan (*disturbance*) gaya dorong ke arah atas sasis. Pengujian ini memanfaatkan umpan balik data spasial dari sensor kedalaman (*depth sensor*) dan *Inertial Measurement Unit* (IMU) Pixhawk yang diproses oleh kontroler PID untuk memodulasi *duty cycle* pada seluruh *thruster* vertikal secara simultan.

Gambar 4. 27 Manuver Vertikal Trim sumbu Y

Gambar 4. 28 Manuver Vertikal Trim sumbu X

Proses implementasi menghasilkan analisis keterkaitan antara kondisi operasi ROV, mekanisme brush, dan kebutuhan gaya stabilisasi selama proses *hull cleaning* yang di Analisis berlanjut menggunakan simulasi. Model simulasi dibangun menggunakan pendekatan matematis berdasarkan prinsip dasar hidrostatika, hidrodinamika, mekanika translasi dan rotasi, serta karakteristik elektromekanik motor BLDC dan pengukuran serta percobaan ril. Hubungan antarparameter utama dapat diringkas sebagai:

$$Kondisi\ Lingkungan\ \rightarrow\ F_B, F_D \rightarrow F_N \rightarrow F_f \rightarrow T_L \rightarrow T_m, I$$

Serta:

$$T_L \rightarrow T_{reaksi} \rightarrow F_{stabilisasi}$$

Dengan demikian, model tidak hanya menghitung satu parameter secara terpisah, tetapi membentuk *coupling* antara dinamika translasi ROV, kontak brush dengan lambung, beban motor, dan kebutuhan stabilisasi. Simulator dibuat berdasarkan model suction dimana bagian dengan ketidak pastian terbesar karena bersifat empiris.

Tabel 4. 10 Model Matematis penyusun SImulator

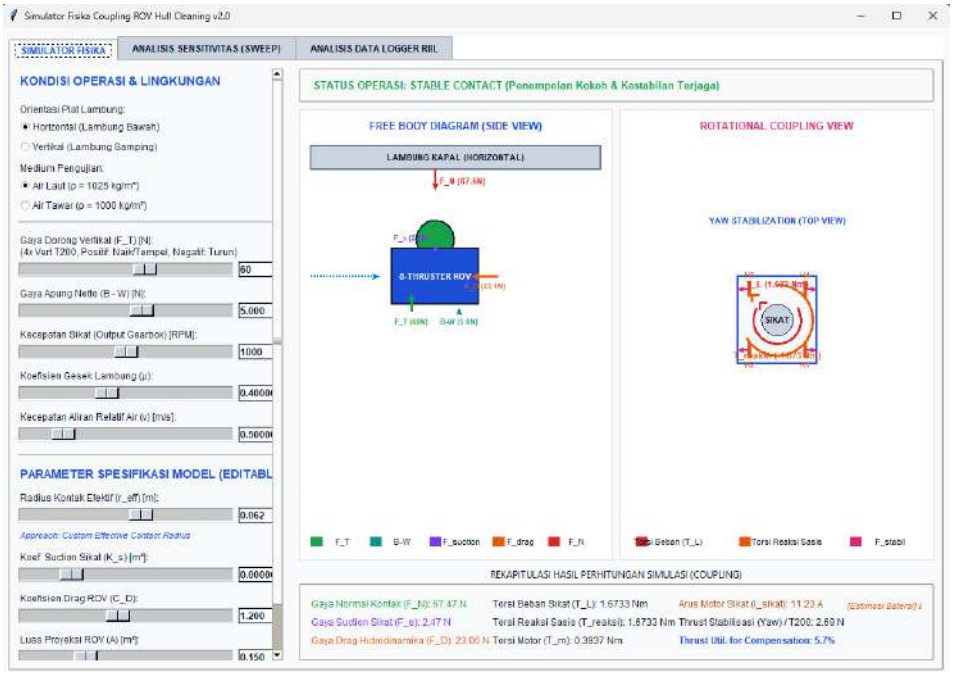
Komponen	Persamaan Dasar	Fungsi dalam Simulasi	No. Persamaan
Gaya apung	$B = \rho\ g\ V$	Menghitung gaya hidrostatik ROV	(4.3)
Berat	$W = m\ g$	Menentukan gaya gravitasi	(4.4)
Net buoyancy	$F_B = B - W$	Menentukan kecenderungan naik/turun	(4.5)
Drag	$F_D = ½\ \rho\ C_D\ A\ v^2$	Menghitung hambatan akibat aliran	(4.6)
Gaya normal	$F_N = \Sigma\ F_{\perp}$	Menentukan gaya kontak ROV–lambung	(4.7)
Gesekan brush	$F_f = \mu\ F_N$	Memperkirakan resistansi kontak	(4.8)
Torsi beban	$T_L = F_f\ r_{eff}$	Mengubah gaya kontak menjadi beban rotasi	(4.9)
Gearbox	$T_m \approx T_L / (G\ \eta_g)$	Menentukan kebutuhan torsi motor	(4.10)
BLDC	$T_m = K_t\ (I - I_0)$	Mengestimasi kebutuhan arus	(4.11)
Torsi reaksi	$T_{reaksi} \approx -T_L$	Menentukan gangguan rotasional terhadap ROV	(4.12)
Thruster	$\tau = r\ F\ \sin(\theta)$	Mengestimasi thrust kompensasi	(4.13)
Suction	$F_s = f(\rho, \omega, r, K_s, ...)$	Pendekatan gaya adhesi brush	(4.14)
Stabilitas	$\Sigma\ F, \Sigma\ M$	Menentukan keseimbangan gaya dan momen	(4.15)

Berdasarkan simulasi matematis, peningkatan kecepatan brush memberikan konsekuensi secara berantai terhadap sistem ROV. Peningkatan RPM menyebabkan perubahan gaya kontak brush terhadap lambung, yang selanjutnya meningkatkan gaya gesek dan torsi beban. Peningkatan torsi tersebut menyebabkan

kebutuhan arus motor bertambah sekaligus menghasilkan torsi reaksi yang harus dikompensasi oleh sistem thruster. Pada kondisi simulasi 1000 RPM diperoleh gaya normal sebesar 30,97 N, torsi beban brush 1,858 Nm, arus motor 12,41 A, serta kebutuhan thrust stabilisasi 2,11 N. Dengan utilisasi thrust sebesar 4,1%, model menunjukkan bahwa ROV secara teoritis masih memiliki kapasitas stabilisasi yang memadai pada titik operasi tersebut. Ketika sikat pembersih mulai beroperasi dan memicu fluktuasi gaya hidrodinamis, kontroler responsif melakukan kompensasi dinamis yang ditandai dengan minimalnya nilai kesalahan posisi tunak (*steady-state error*). Keberhasilan pengujian manuver holding vertikal ini membuktikan bahwa sistem kendali loop tertutup (*closed-loop*) pada sumbu y-axis memiliki tingkat kekokohan (*robustness*) yang optimal, sehingga mampu menjaga stabilitas posisi kerja ROV saat mengeksekusi pembersihan vertikal tanpa mengalami gejala melayang (*drifting*) atau osilasi liar yang dapat merusak plat lambung kapal.

4.5.1.1 Hasil Kondisi Operasi Simulasi

Pada kondisi operasi yang ditampilkan pada GUI, simulasi menghasilkan status *Stable Contact*, yang menunjukkan bahwa resultan gaya dan momen masih memenuhi kondisi yang ditetapkan agar ROV mampu mempertahankan kontak dengan permukaan lambung.



Gambar 4. 29 Hasil Simulasi Gaya ROV menggunakan pendekatan Fisika dan data pengujian nyata

Tabel 4. 11 Hasil Simulasi Satu Kondisi Asumsi

Parameter	Hasil
Gaya normal (F_N)	30,97 N
Gaya suction brush (F_s)	10,97 N
Drag hidrodinamika (F_D)	0,90 N
Torsi beban brush (T_L)	1,8580 Nm
Torsi reaksi chassis	1,8580 Nm
Torsi motor (T_m)	0,4372 Nm
Arus motor brush	12,41 A
Thrust stabilisasi	2,11 N
Utilisasi thrust	4,1%

Nilai torsi reaksi yang sama besar dengan torsi beban brush tetapi berlawanan arah merupakan konsekuensi Hukum III Newton. Oleh karena itu, semakin besar torsi yang dibutuhkan untuk proses pembersihan, semakin besar pula kecenderungan chassis ROV mengalami rotasi. Kondisi tersebut harus dikompensasi oleh thruster stabilisasi. Pada titik operasi tersebut, kebutuhan *thrust* stabilisasi hanya 2,11 N, sedangkan kemampuan *thrust* yang tersedia jauh lebih besar, sehingga utilisasinya sekitar 4,1%. Secara matematis kondisi ini menunjukkan bahwa aktuator stabilisasi masih memiliki *margin* yang besar.

4.5.1.2 Analisis Sensivitas RPM terhadap Gaya Normal

Hasil parameter *sweep* menunjukkan bahwa peningkatan RPM brush menyebabkan gaya normal kontak meningkat secara nonlinier. Temuan ini menunjukkan bahwa kenaikan kecepatan putar tidak hanya meningkatkan laju pembersihan, tetapi juga memengaruhi karakteristik interaksi mekanis antara brush dan permukaan lambung kapal. Pada RPM yang lebih tinggi, peningkatan gaya normal terjadi akibat perubahan dinamika putaran, defleksi bulu sikat, serta bertambahnya tekanan kontak terhadap permukaan. Kondisi tersebut dapat meningkatkan gaya gesek dan torsi beban yang harus diatasi oleh motor selama proses pembersihan. Oleh karena itu, pemilihan RPM perlu disesuaikan dengan tingkat *biofouling* yang terdeteksi agar gaya kontak yang dihasilkan tetap berada pada kondisi yang efektif dan terkendali. Pengaturan kecepatan yang tepat juga diperlukan untuk mencegah tekanan berlebih yang berpotensi menyebabkan kerusakan pada permukaan lambung. Dengan demikian, hasil *sweep* menjadi dasar dalam menentukan rentang operasi RPM brush pada masing-masing kelas.

Gambar 4. 30 Hasil RPM terhadap F_N

Bentuk kurva yang cembung menunjukkan adanya komponen yang bergantung pada kuadrat kecepatan putar. Dari Grafik pada gambar 4.26 dapat diambil beberapa informasi bahwasannya pada RPM 0-300 Gaya Normal masih dalam kisaran 65N, namun pada RPM 2000 Gaya Normal menjadi 74.89N. Oleh karena itu, peningkatan RPM pada daerah tinggi memberikan kenaikan gaya kontak yang lebih besar dibandingkan peningkatan RPM dengan besar yang sama pada daerah rendah.

4.5.1.3 Pengaruh Torsi Beban terhadap Stabilitas ROV

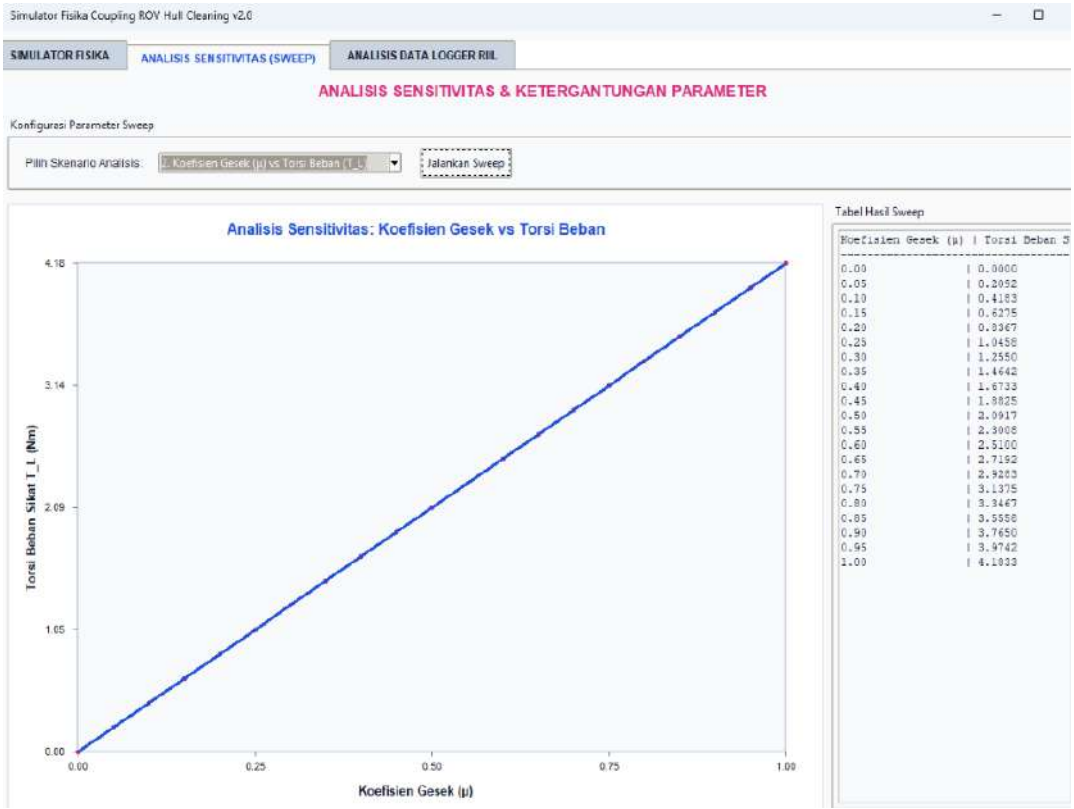
Hasil sweep torsi beban menunjukkan hubungan yang hampir linier terhadap kebutuhan thrust stabilisasi. Pada torsi beban sekitar 0,658 Nm, kebutuhan thrust stabilisasi adalah sekitar 0,748 N, sedangkan pada torsi 12,658 Nm, kebutuhan thrust meningkat hingga sekitar 14,384 N. Hubungan ini sesuai dengan

$$\tau = rF$$

Sehingga untuk konfigurasi geometris dengan lengan momen r yang tetap:

$$F_{stabil} \propto T_L$$

Dari informasi berikut ini dapat disimpulakan bahwasannya semakin besar torsi brush terhadap permukaan lambung, semakin besar pula gaya yang harus dihasilkan thruster untuk mencegah ROV mengalami perubahan orientasi akibat torsi reaksi.



Gambar 4. 31 Grafik Pengaruh Torsi Beban Terhadap Stabilitas ROV

Ini merupakan salah satu hasil simulasi yang paling relevan terhadap desain ROV *Hull Cleaning* yang mana kapasitas brush tidak boleh ditentukan hanya berdasarkan kemampuan motor. Kemampuan thruster mempertahankan orientasi juga harus diperhitungkan.

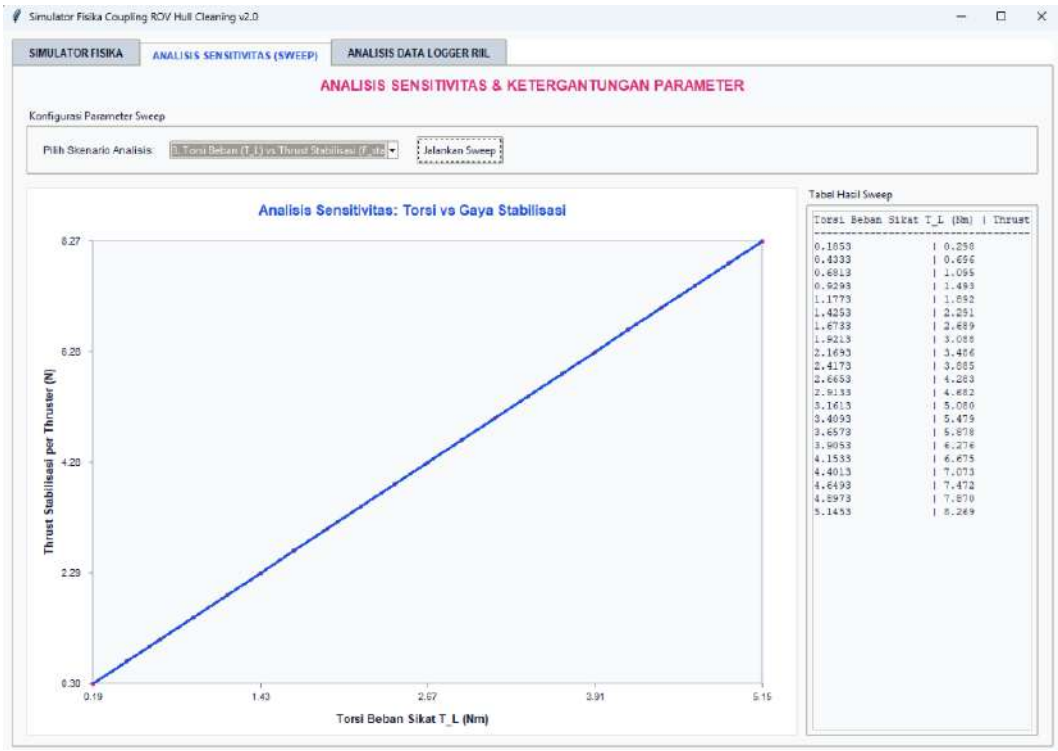
4.5.1.4 Pengaruh RPM terhadap Arus Motor Brush

Pada subbab ini dibahas hasil pengujian berdasarkan simulasi dan perbandingannya terhadap data aktual. Hasil simulasi menunjukkan bahwa peningkatan RPM menyebabkan kebutuhan arus motor meningkat secara nonlinier. Pada kondisi awal, arus motor diperkirakan sebesar 8,194 A dan meningkat hingga sekitar 25 A pada kecepatan 2000 RPM. Pada kecepatan 800 RPM, hasil perhitungan simulasi menghasilkan arus sekitar 0,289 A, sedangkan tampilan

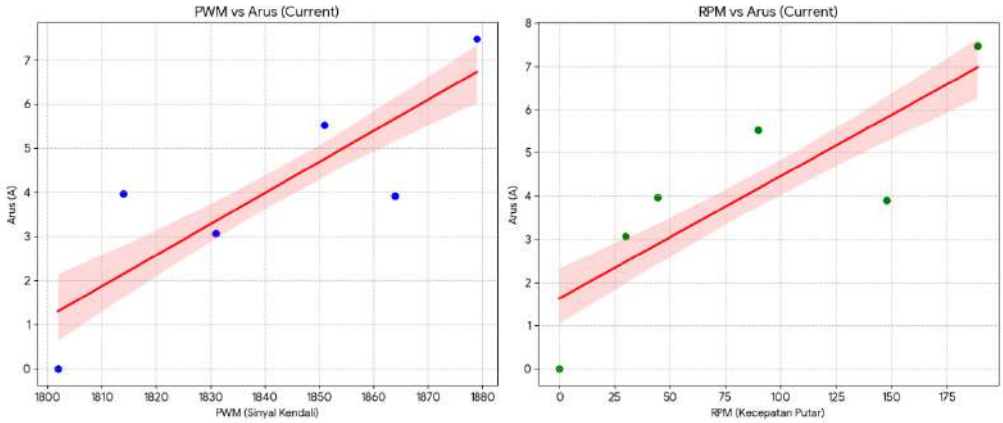
simulator utama menunjukkan nilai sebesar 8,269 A. Perbedaan tersebut menunjukkan adanya ketidaksesuaian antara model perhitungan dengan parameter atau konfigurasi yang digunakan pada simulator utama. Meskipun terdapat perbedaan nilai absolut, kecenderungan peningkatan arus terhadap kenaikan RPM tetap terlihat pada hasil simulasi. Pola yang serupa juga ditemukan pada pengujian aktual, baik saat motor beroperasi di darat maupun di dalam air. Pada kondisi nyata, peningkatan RPM menyebabkan bertambahnya gaya hambat, gaya kontak, dan torsi beban pada brush sehingga kebutuhan arus motor ikut meningkat. Beban hidrodinamika dan gesekan permukaan turut memperbesar kebutuhan daya ketika motor bekerja di dalam air. Dengan demikian, hasil pengujian aktual mendukung kecenderungan hubungan antara RPM dan arus yang diperoleh dari simulasi, meskipun masih terdapat perbedaan nilai numerik akibat penyederhanaan model dan kondisi lingkungan operasi. Torsi motor yang dibutuhkan ikut meningkat dan, berdasarkan hubungan

$$T_m = K_t(I - I_0)$$

peningkatan kebutuhan torsi menyebabkan peningkatan arus motor.



Gambar 4. 32 Hasil Simulasi Pengaruh RPM terhadap Arus Motor



Gambar 4. 33 Hasil Pengujian Pengaruh PWM terhadap arus motor (Kiri) RPM terhadap Arus Motor (Kanan)

Pita arsiran berwarna merah muda (pink) yang mengelilingi garis regresi utama pada Gambar 4.33 merupakan representasi Interval Kepercayaan 95% (*95% Confidence Interval*) dari estimasi model linier. Lebar pita menunjukkan rentang ketidakpastian atau *margin of error* dari estimasi garis regresi berdasarkan sebaran data aktual hasil pengujian. Pita yang relatif sempit pada bagian tengah grafik menunjukkan bahwa hubungan antara PWM/RPM dan arus memiliki variabilitas yang lebih rendah pada kondisi beban sedang, sehingga karakteristik operasional motor cenderung lebih stabil. Sebaliknya, pita yang semakin melebar pada bagian kanan grafik, khususnya ketika kecepatan mendekati 190 RPM, menunjukkan peningkatan variasi data arus pada kondisi beban tinggi. Kondisi tersebut dapat dipengaruhi oleh perubahan gaya hambat hidrodinamis, turbulensi air, serta variasi gaya gesek akibat perubahan tekanan normal ((FN)) antara bulu sikat nilon dan permukaan lambung. Akibatnya, konsumsi arus aktual mengalami fluktuasi yang lebih besar pada kondisi beban puncak dibandingkan kondisi operasi dengan beban lebih rendah.

4.5.2 Hasil Implementasi YOLO

Pengujian implementasi model YOLO dilakukan untuk mengetahui kemampuan model dalam mendeteksi dan mengklasifikasikan tingkat fouling pada permukaan lambung kapal. Pengujian visual menggunakan ROV dilakukan selama 10 menit pada area sekitar kapal yang sedang sandar. Dari proses pengambilan dan

filtrasi data diperoleh sebanyak 50 citra sebagai data pengujian. Hasil pengujian menunjukkan bahwa sebanyak 46 citra berhasil menghasilkan deteksi objek, sedangkan 4 citra tidak menghasilkan deteksi. Berdasarkan hasil tersebut, tingkat keberhasilan deteksi model mencapai 92%. Nilai ini menunjukkan bahwa YOLO mampu mengenali objek biofouling pada sebagian besar citra yang diperoleh selama pengujian. Hasil tersebut juga menunjukkan bahwa model memiliki performa yang cukup baik untuk digunakan dalam proses klasifikasi kondisi fouling secara visual pada sistem ROV.

Tabel 4. 12 Hasil Pengujian YOLO

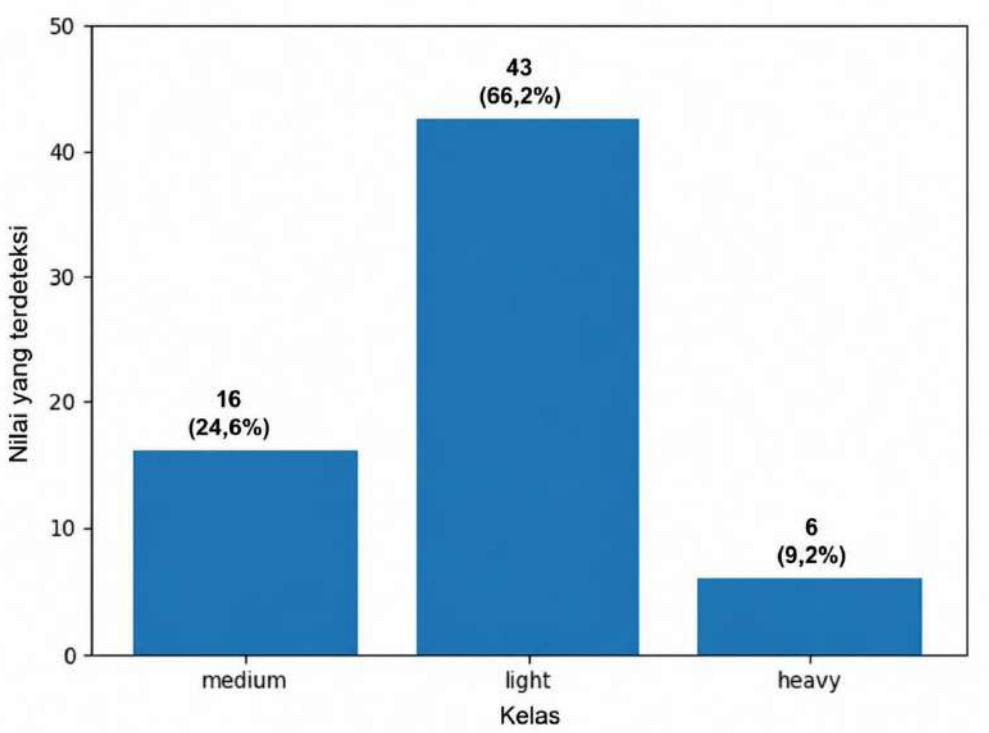
Confidence	Total data
<70	0
70<90	38
90<100	27

Img	Class	Confidence
1	medium	0.890
1	light	0.880
3	light	0.940
4	light	0.810
5	light	0.880
6	light	0.950
10	heavy	0.940
10	light	0.820
11	light	0.820
12	light	0.830
13	light	0.940
13	heavy	0.920
14	light	0.820
15	light	0.870
16	light	0.940
17	light	0.718
18	light	0.900
19	light	0.940
20	light	0.746
21	light	0.910
22	light	0.840
23	light	0.860

Img	Class	Confidence
24	medium	0.890
24	light	0.830
25	light	0.860
26	light	0.890
27	heavy	0.860
28	light	0.850
29	medium	0.890
29	heavy	0.910
30	light	0.760
31	light	0.930
31	light	0.890
32	heavy	0.850
32	medium	0.950
32	light	0.810
33	light	0.920
33	medium	0.920
34	light	0.930
34	light	0.860
35	light	0.810
36	light	0.843
37	light	0.940
37	medium	0.880

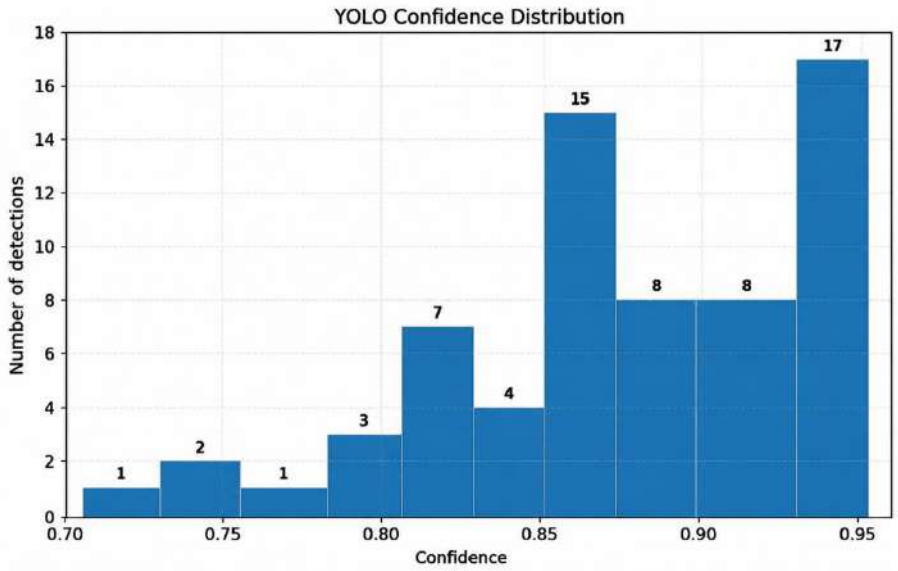
Img	Class	Confidence
37	heavy	0.840
38	light	0.860
39	light	0.930
40	light	0.880
41	light	0.920
42	medium	0.930
42	light	0.940
42	medium	0.930
43	light	0.890
44	light	0.940
45	light	0.930
45	medium	0.820
45	medium	0.870
45	medium	0.950
45	medium	0.920
45	medium	0.860
46	light	0.860
47	light	0.870
48	medium	0.900
49	medium	0.821
50	medium	0.910

Distribusi hasil deteksi berdasarkan kelas ditunjukkan pada tabel. Hasil pengujian menunjukkan bahwa kelas light merupakan kelas yang paling banyak terdeteksi, yaitu sebanyak 43 objek atau sekitar 66,2% dari keseluruhan hasil deteksi. Selanjutnya, kelas medium terdeteksi sebanyak 16 objek atau sekitar 24,6%, sedangkan kelas heavy hanya terdeteksi sebanyak 6 objek atau sekitar 9,2%.



Gambar 4. 34 Grafik Distribusi Kelas dari Hasil YOLO

Berdasarkan tabel dan grafik pada Gambar 4.34, terlihat bahwa kelas *Light* menjadi kelas yang paling dominan dibandingkan kelas *Medium* dan *Heavy*. Dominasi tersebut menunjukkan bahwa sebagian besar data hasil pengujian pada periode pengamatan memiliki tingkat biofouling ringan. Setelah seluruh hasil deteksi YOLO dikumpulkan, data diproses dengan *voting* untuk menentukan kelas dengan jumlah atau akumulasi nilai tertinggi. Kelas yang memperoleh hasil *voting* paling dominan kemudian ditetapkan sebagai keluaran pada setiap *loop* proses. Dengan demikian, sistem mampu menghasilkan keputusan kelas secara konsisten berdasarkan kondisi biofouling yang paling banyak terdeteksi.



Gambar 4. 35 Grafik YOLO Confidence Distribution

Selain jumlah deteksi, tingkat keyakinan (*confidence* score) model juga dianalisis sebagaimana ditunjukkan pada Gambar 4.26. Berdasarkan histogram distribusi *confidence*, sebagian besar hasil deteksi memiliki nilai *confidence* pada rentang 0,86 hingga 0,95, dengan beberapa prediksi mencapai nilai sekitar 0,95. Hanya sebagian kecil prediksi yang memiliki *confidence* di bawah 0,80, dengan nilai terendah sekitar 0,72.

Distribusi confidence yang terkonsentrasi pada nilai tinggi menunjukkan bahwa model memiliki tingkat keyakinan yang baik terhadap sebagian besar hasil prediksi yang dihasilkan. Nilai confidence yang tinggi mengindikasikan bahwa fitur-fitur objek fouling berhasil dipelajari dengan baik selama proses pelatihan, sehingga model mampu melakukan klasifikasi dengan tingkat kepercayaan yang relatif tinggi pada data pengujian.

Secara keseluruhan, hasil implementasi menunjukkan bahwa model YOLO mampu melakukan deteksi tingkat fouling dengan baik pada data uji. Tingkat keberhasilan deteksi sebesar 92% serta dominasi nilai confidence di atas 0,85 menunjukkan bahwa model cukup andal untuk diimplementasikan sebagai bagian dari sistem identifikasi fouling pada ROV Hull Cleaning. Meskipun demikian, distribusi data yang tidak seimbang antar kelas masih menjadi faktor yang perlu

diperhatikan agar performa model pada kelas medium dan heavy dapat lebih ditingkatkan pada penelitian selanjutnya.

Pada sistem ROV Hull Cleaning yang dikembangkan, algoritma YOLO tidak hanya berfungsi sebagai metode deteksi objek, tetapi juga sebagai dasar pengambilan keputusan (decision making) dalam sistem kendali. Setiap hasil klasifikasi yang diberikan oleh model YOLO akan diterjemahkan menjadi perintah pengaturan kecepatan putar (brush RPM) sesuai tingkat fouling yang terdeteksi.

Tabel 4. 13 Hubungan YOLO dengan Aksi Kontrol

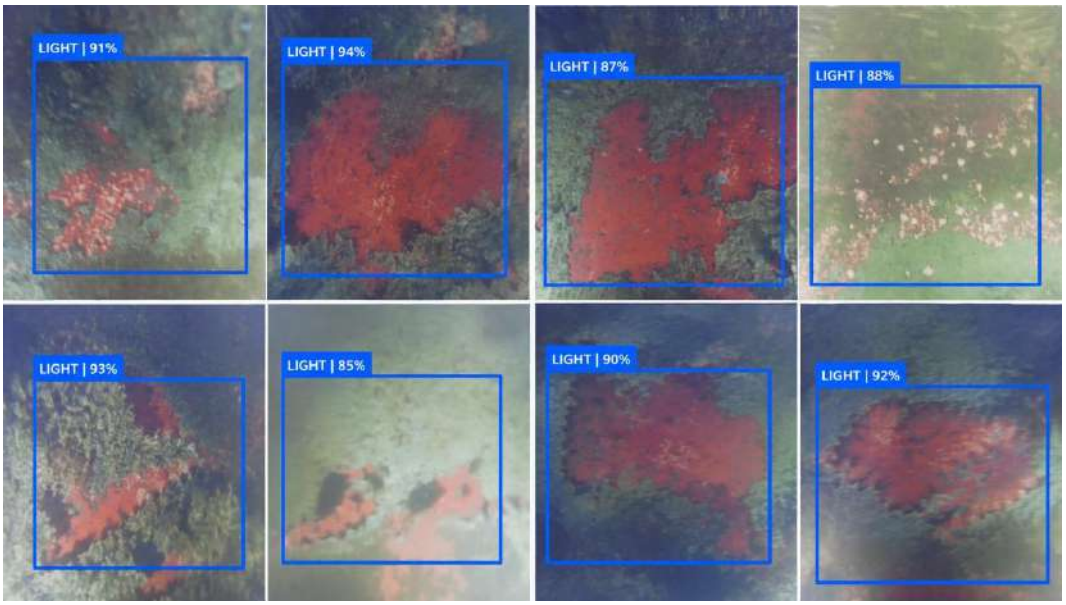
Hasil YOLO	Aksi Kontrol
Light	Brush RPM 80
Medium	Brush RPM 125
Heavy	Brush RPM 200

Berdasarkan tabel tersebut, setiap kelas hasil deteksi memiliki aksi kontrol yang berbeda. Apabila YOLO mengidentifikasi fouling dengan kategori *Light*, sistem menetapkan setpoint kecepatan brush sebesar 80 RPM. Pada kategori Medium, setpoint dinaikkan menjadi 125 RPM, sedangkan untuk kategori *Heavy*, sistem menetapkan setpoint sebesar 200 RPM. Penyesuaian ini bertujuan agar gaya gesek yang dihasilkan brush sesuai dengan tingkat fouling yang terdeteksi, sehingga proses pembersihan menjadi lebih efektif sekaligus menghindari penggunaan energi yang berlebihan.

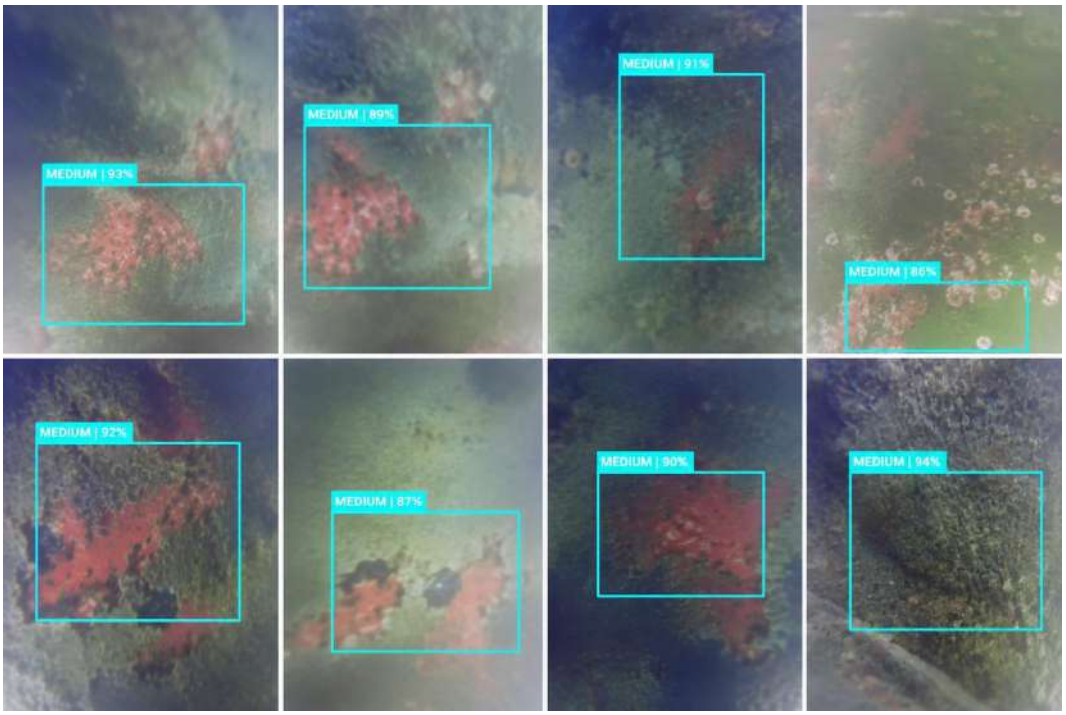
4.5.3 Analisis Pengaruh YOLO Dalam Sistem

YOLO berperan sebagai penentu *setpoint* RPM sekaligus nilai awal PWM (*starting PWM*) yang diberikan kepada motor *brush*. Hasil klasifikasi YOLO digunakan untuk menentukan kebutuhan kecepatan motor berdasarkan tingkat biofouling yang terdeteksi, yaitu *Light*, *Medium*, dan *Heavy*. Selanjutnya, hasil deteksi tersebut diproses melalui mekanisme *counter voting* dan dikombinasikan dengan data sensor gyroscope sebelum sistem menentukan apakah motor *brush* dapat mulai berputar. Keputusan sistem tidak hanya bergantung pada satu *frame*, tetapi mempertimbangkan konsistensi hasil deteksi pada beberapa *frame* secara berturut-turut. Mekanisme ini diterapkan untuk mengurangi kemungkinan *false*

detection akibat gangguan visual pada lingkungan bawah air, seperti perubahan pencahayaan, gelembung air, pergerakan kamera, maupun *noise* citra. Dengan demikian, sistem dapat menghasilkan keputusan *starting* yang lebih stabil dan sesuai kondisi biofouling aktual.



Gambar 4. 36 Hasil YOLO kelas *Light*



Gambar 4. 37 Hasil YOLO kelas *Medium*

Gambar 4. 38 Hasil YOLO kelas *Heavy*

Berdasarkan Gambar 4.36 hingga Gambar 4.37, diperoleh hasil deteksi dan klasifikasi menggunakan YOLO terhadap tingkat *biofouling* pada permukaan objek berdasarkan citra bawah air. YOLO mampu mengidentifikasi keberadaan *biofouling* sekaligus mengelompokkannya ke dalam tiga kelas berdasarkan kebutuhan pembersihan, yaitu *Light*, *Medium*, dan *Heavy*. Kelas *Light* menunjukkan kondisi *biofouling* yang relatif rendah sehingga membutuhkan tingkat pembersihan ringan. Kelas *Medium* menunjukkan penumpukan *biofouling* yang lebih signifikan sehingga memerlukan pembersihan dengan intensitas sedang. Sementara itu, kelas *Heavy* menunjukkan penumpukan *biofouling* yang lebih tinggi dan dominan sehingga membutuhkan proses pembersihan yang lebih intensif. Hasil deteksi ditampilkan melalui *bounding box* disertai nilai *confidence* pada setiap objek yang teridentifikasi. Keberadaan *bounding box* pada area yang sesuai dengan karakteristik visual *biofouling* menunjukkan bahwa model dapat mengenali objek pada kondisi citra bawah air yang memiliki keterbatasan pencahayaan dan kualitas visual. Selanjutnya, hasil klasifikasi YOLO digunakan sebagai dasar penentuan tingkat pembersihan serta nilai *setpoint* kecepatan motor *brush*. Dengan demikian, proses klasifikasi tidak hanya berfungsi sebagai identifikasi visual, tetapi menjadi bagian penting dalam pengambilan keputusan otomatis sistem pembersihan.

Tabel 4. 14 Hasil Voting dari YOLO Kelas L

Kelas	Jumlah	Jumlah (%)
L (Light)	118	88.72
M (Medium)	6	4.51
H (Heavy)	9	6.77

Total Data	133
Winner	Light
Serial yang dikirim ke ESP32	L

Tabel 4. 15 Hasil Voting dari YOLO Kelas M

Kelas	Jumlah	Jumlah (%)
L (Light)	22	14.56
M (Medium)	115	76.15
H (Heavy)	7	4.63

Total Data	151
Winner	Medium
Serial yang dikirim ke ESP32	M

Tabel 4. 16 Hasil Voting dari YOLO Kelas H

Kelas	Jumlah	Jumlah (%)
L (Light)	119	73.46
M (Medium)	28	17.28
H (Heavy)	15	9.26

Total Data	162
Winner	Heavy
Serial yang dikirim ke ESP32	H

Setelah proses deteksi dan klasifikasi dilakukan oleh YOLO, sistem dilanjutkan dengan metode Voting untuk menentukan kelas akhir berdasarkan keseluruhan data hasil deteksi yang masuk. Hasil pada Tabel 4.14–4.16 menunjukkan bahwa metode voting mampu menentukan kelas dominan pada masing-masing kelompok pengujian. Presentase kemampuan YOLO di hitung melalui :

$$\% = \frac{Jumlah\ Serial\ Data\ 1\ Kelas}{Total\ Serial\ Data\ Terkumpul} \quad (4.16)$$

Pada data *Light*, sebanyak 118 dari 133 data menghasilkan kelas *Light* sebagai hasil Voting dengan persentase 88,72%. Pada data *Medium*, kelas *Medium* menjadi hasil dominan dengan 112 dari 151 data atau 74,17%. Sementara itu, pada data *Heavy*, kelas *Heavy* memperoleh 119 dari 162 data atau 73,46%. Hasil tersebut menunjukkan bahwa sistem voting berhasil mempertahankan kelas yang sesuai dengan kondisi *biofouling* yang menjadi masukan meskipun terdapat sejumlah hasil deteksi YOLO yang diklasifikasikan ke kelas lain. Dengan demikian, metode voting dapat digunakan sebagai tahap pengambilan keputusan akhir untuk mengurangi pengaruh kesalahan klasifikasi pada deteksi individual dan menghasilkan keputusan kelas yang lebih representatif terhadap kondisi *biofouling* pada setiap sesi pengujian.

4.5.3 Hasil Implementasi Metode PID

Penggunaan pengendali PID pada sistem hull cleaning tidak hanya berpengaruh terhadap kemampuan motor dalam mencapai nilai RPM referensi, tetapi juga terhadap karakteristik interaksi antara brush, biofouling, dan permukaan lambung. Pada saat brush melakukan kontak dengan biofouling, beban mekanis yang diterima motor tidak bersifat konstan. Ketebalan, kepadatan, serta karakteristik permukaan biofouling menyebabkan gaya resistif terhadap putaran brush berubah selama proses pembersihan. Perubahan beban tersebut secara langsung memengaruhi kecepatan putar dan arus motor. Pada hasil implementasi tanpa PID, kondisi tersebut dapat menyebabkan sistem berhenti bekerja atau mengalami reset/shutdown. Dengan demikian, stalling motor brush tidak hanya menjadi permasalahan pada aktuator, tetapi dapat berkembang menjadi gangguan terhadap keseluruhan sistem ROV.

Tabel 4. 17 Legenda Variabel Motor Running Mode

S	STOP
L	LIGHT
M	MEDIUM
H	HEAVY

Tabel 4. 18 Linimasa System pada kelas L

Rekap / Linimasa										
time	class	lock	system_state	setpoint	rpm	pwm	mode	current	accel_y	Keterangan
18.06.27	L	-	VOTING	0	0	1500	S	0	193	System Voting
18.06.47	L	L	WAIT_GYRO	0	0	1500	S	0	-13	Waiting Gyro
18.06.51	L	L	RUNNING	80	24	1571	L	0.8	911	Drop Point
18.06.53	L	L	RUNNING	80	79.04	1569	L	0.9	1005	Settling Time
18.06.56	L	L	RUNNING	80	40.31	1569	l	0.9	990	Drop Point
18.06.56	L	L	RUNNING	80	50.59	1577	L	1.1	970	Recovery
18.06.57	L	L	RUNNING	80	72.35	1582	L	1.3	946	Recovery
18.07.09	L	L	RUNNING	80	83.34	1569	L	0.9	998	Settling Time
18.07.10	L	-	VOTING	0	0	1500	S	0.1	450	System Voting

Tabel 4. 19 Linimasa System pada Kelas M

Rekap / Linimasa										
time	class	lock	system_state	setpoint	rpm	pwm	mode	current	accel_y	Keterangan
10.00.06	M	-	VOTING	0	0	1500	S	0	87	System Voting
10.00.26	M	M	WAIT_GYRO	0	0	1500	S	0	98	Waiting Gyro
10.00.32	M	M	RUNNING	120	26.11	1700	M	1.56	985	Drop Point
10.00.35	M	M	RUNNING	120	121.24	1748	M	3.12	905	Settling Time
10.00.38	M	M	RUNNING	120	88.7	1900	M	4.22	956	Drop Point
10.00.38	M	M	RUNNING	120	101.98	1900	M	4.25	1005	Recovery
10.00.39	M	M	RUNNING	120	119.98	1825	M	2.34	1002	Settling Time
10.00.40	M	M	RUNNING	120	103.54	1865	M	4.49	1100	Drop Point
10.00.43	M	M	RUNNING	120	97.65	1900	M	4.79	998	Drop Point
10.00.43	M	M	RUNNING	120	113.43	1900	M	4.67	1002	Recovery
10.00.45	M	M	RUNNING	120	123.32	1840	M	3.65	984	Settling Time
10.00.51	-	-	RUNNING	80	87	1570	L	1.01	923	Switch Mode

Tabel 4. 20 Linimasa System pada Kelas H

Rekap / Linimasa										
time	class	lock	system_state	setpoint	rpm	pwm	mode	current	accel_y	Keterangan
10.22.11	H	-	VOTING	0	0	1500	S	0	193	System Voting
10.22.31	H	H	WAIT_GYRO	0	0	1500	S	0	-13	Waiting Gyro
10.23.01	H	H	RUNNING	200	30.86	1806	H	2.55	971	Drop Point
10.23.05	H	H	RUNNING	200	192.54	1868	H	5.62	992	Settling Time
10.23.08	H	H	RUNNING	200	166.98	1900	H	5.99	990	Drop Point
10.23.09	H	H	RUNNING	200	173.48	1900	H	8.22	981	Drop Point
10.23.09	H	H	RUNNING	200	180.67	1900	H	8.43	970	Recovery
10.23.10	H	H	RUNNING	200	182.19	1900	H	8.65	946	Recovery
10.23.11	H	H	RUNNING	200	198.28	1900	H	7.69	998	Settling Time
10.23.35	-	-	RUNNING	120	167.11	1700	M	2.46	1002	Switch Mode

Dari hasil pengujian, untuk melakukan analisis elektro-mekanis, digunakan parameter fisik sistem yang merujuk pada spesifikasi desain simulator ROV:

1. Parameter Fisik dan permodelan sistem

Jari-jari sikat Nominal :

$$r_{\text{sikat}} = 0.0624 \text{ m } (62.4 \text{ mm} 62.4 \text{ mm})$$

Rasio Reduksi Gearbox :

(R): 5.0: 15.0: 1 (PlanetaryGearboxdenganGigiMatahari Nsun $= 12\mathrm{T}N_{\mathrm{sun}}$
$= 12T$ danGigiCincin Nring $= 48\mathrm{T}N_{\mathrm{ring}} = 48T$) R
$= 1 + \mathrm{NringNsun} = 1 + 4812 = 5.0R = 1 + N_{\mathrm{sun}}N_{\mathrm{ring}} = 1 + 1248$
$= 5.0$

Efisiensi Transmisi Gearbox :

$$(\eta_g) = 85\% \ (0.85)$$

Konstanta Torsi Motor :

$$K_t = 0.0367 \ \mathrm{Nm/A}$$

Arus Tanpa Beban : 0.5 A

Tegangan Operasi Normal : 16V

2. Metodelogi Perhitungan Elektro-Mekanis

Torsi Poros Motor :

$$T_{motor} = K_t(I - I_0)$$

Torsi Poros Sikat Keluaran :

$$T_{sikat} = T_{motor}R\eta_g$$
$$T_{sikat} = 4.25T_{motor}$$

Kecepatan Sudut Sikat :

$$\omega_{sikat} = \frac{2\pi RPM}{60}$$

Daya Mekanis Output Sistem :

$$P_{mech} = T_{sikat}\omega_{sikat}$$

Daya Listrik Input Motor :

$$P_{elec} = V_{ref}I$$

Efisiensi Sistem :

$$\eta_{sys} = \frac{P_{mech}}{P_{elec}} \times 100\backslash\%$$

Dari pendekatan matematis baik dari segi Elektro sampai mekanis dihasilkan data sebagai berikut :

Tabel 4. 21 Hasil Perhitungan Pada Kelas L

Waktu	Setpoint (RPM)	RPM Aktual	PWM (μs)	Arus (A)	Daya Listrik (Pelec)	Torsi Sikat (*T*sikat)	Daya Mekanis (*P*mech)	Efisiensi (ηsys)
18.06.51	80	24	1571	0.8	12.80 W	0.0468 Nm	0.12 W	0.90%
18.06.53	80	79.04	1569	0.9	14.40 W	0.0624 Nm	0.52 W	3.60%
18.06.56	80	40.31	1569	0.9	14.40 W	0.0624 Nm	0.26 W	1.80%
18.06.56	80	50.59	1577	1.1	17.60 W	0.0936 Nm	0.50 W	2.80%
18.06.57	80	72.35	1582	1.3	20.80 W	0.1248 Nm	0.95 W	4.50%
18.07.09	80	83.34	1569	0.9	14.40 W	0.0624 Nm	0.54 W	3.80%

Rata – rata RPM 58.27, steady state akhir 83.34, dengan Galat Steady State = +4.18%, menghasilkan nilai arus 0.98A, rata-rata torsi sikat 0.0754Nm dengan torsi yang dihasilakan maksimal 0.1248Nm, rata-rata daya mekanis yang timbul 0.48W dan menghasilkan rata-rata sebesar 2.9%.

Tabel 4. 22 Hasil Perhitungan Pada Kelas M

Waktu	Setpoint (RPM)	RPM Aktual	PWM (μs)	Arus (A)	Daya Listrik (*P*elec)	Torsi Sikat (*T*sikat)	Daya Mekanis (*P*mech)	Efisiensi (ηsys)
10.00.32	120	26.11	1700	1.56	24.96 W	0.1653 Nm	0.45 W	1.80%
10.00.35	120	121.24	1748	3.12	49.92 W	0.4087 Nm	5.19 W	10.40%
10.00.38	120	88.7	1900	4.22	67.52 W	0.5802 Nm	5.39 W	8.00%
10.00.38	120	101.98	1900	4.25	68.00 W	0.5849 Nm	6.25 W	9.20%
10.00.39	120	119.98	1825	2.34	37.44 W	0.2870 Nm	3.61 W	9.60%
10.00.40	120	103.54	1865	4.49	71.84 W	0.6223 Nm	6.75 W	9.40%
10.00.43	120	97.65	1900	4.79	76.64 W	0.6691 Nm	6.84 W	8.90%
10.00.43	120	113.43	1900	4.67	74.72 W	0.6504 Nm	7.73 W	10.30%
10.00.45	120	123.32	1840	3.65	58.40 W	0.4913 Nm	6.34 W	10.90%
10.00.51	80	87	1570	1.01	16.16 W	0.0795 Nm	0.72 W	4.50%

Rata – rata RPM 98.30, steady state akhir 123.32, dengan Galat Steady State = +2.77%, menghasilkan nilai arus 3.41A, rata-rata torsi sikat 0.4539Nm dengan torsi

yang dihasilakan maksimal 0.6691Nm, rata-rata daya mekanis yang timbul 4.93W dan menghasilkan rata-rata sebesar 8.3%.

Tabel 4. 23 Hasil Perhitungan Pada Kelas H

Waktu	Setpoint (RPM)	RPM Aktual	PWM (μs)	Arus (A)	Daya Listrik (*P*elec)	Torsi Sikat (*T*sikat)	Daya Mekanis (*P*mech)	Efisiensi (ηsys)
10.23.01	200	30.86	1806	2.55	40.80 W	0.3197 Nm	1.03 W	2.50%
10.23.05	200	192.54	1868	5.62	89.92 W	0.7986 Nm	16.10 W	17.90%
10.23.08	200	166.98	1900	5.99	95.84 W	0.8563 Nm	14.97 W	15.60%
10.23.09	200	173.48	1900	8.22	131.52 W	1.2041 Nm	21.88 W	16.60%
10.23.09	200	180.67	1900	8.43	134.88 W	1.2369 Nm	23.40 W	17.30%
10.23.10	200	182.19	1900	8.65	138.40 W	1.2712 Nm	24.25 W	17.50%
10.23.11	200	198.28	1900	7.69	123.04 W	1.1215 Nm	23.29 W	18.90%
10.23.35	120	167.11	1700	2.46	39.36 W	0.3057 Nm	5.35 W	13.60%

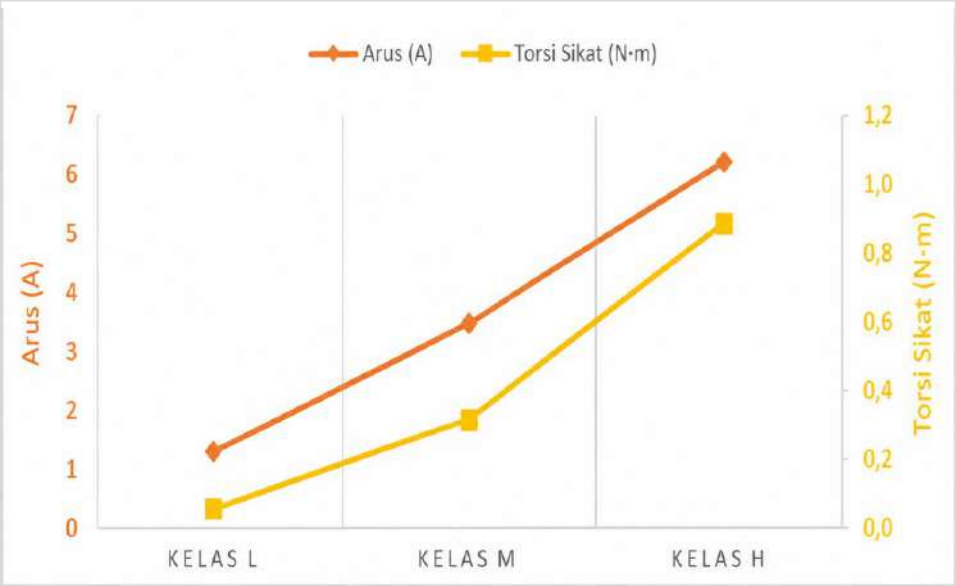
Rata – rata RPM 161.51, steady state akhir 198.28, dengan Galat Steady State = -0.86%, menghasilkan nilai arus 6.20A, rata-rata torsi sikat 0.8893Nm dengan torsi yang dihasilakan maksimal 1.2712Nm, rata-rata daya mekanis yang timbul 16.28W dan menghasilkan rata-rata sebesar 15%.

Tabel 4. 24 Perbandingan Karakteristik Kelas (summary Comparison)

Parameter Teknis	Kelas L	Kelas M	Kelas H
Rata-rata RPM Aktual	58.27 RPM	98.30 RPM	161.51 RPM
Rata-rata Konsumsi Arus	0.98 A	3.41 A	6.20 A
Konsumsi Arus Puncak	1.30 A	4.79 A	8.65 A
Rata-rata Output Torsi Sikat	0.0754 Nm	0.4539 Nm	0.8893 Nm
Torsi Puncak Sikat	0.1248 Nm	0.6691 Nm	1.2712 Nm
Daya Listrik Rata-rata (Pelec*P*elec)	15.73 W	54.56 W	99.22 W
Daya Mekanis Rata-rata (Pmech*P*mech)	0.48 W	4.93 W	16.28 W
Efisiensi Rata-rata (ηsysηsys)	2.90%	8.30%	15.00%
Rata-rata Output PWM	1572.8 μμs	1814.8 μμs	1859.2 μμs

Dari hasil percobaan dibuktikan linearitas dengan sangat baik, mulai dari rata-rata arus naik dari 0.98 A pada kondisi Light, 3.41 A pada kondisi Medium dan 6.20 A pada kondisi Heavy. Peningkatan arus ini berkorelasi langsung secara linier

terhadap peningkatan rata-rata torsi sikat dari 0.0754 Nm, 0,4539 Nm hingga 0,8893 Nm.



Gambar 4. 39 Grafik Linearitas Elektromagnetik

Dominasi arus tanpa beban pada tingkat beban yang ringan (kelas L) memakan sekitar 51% dari total arus listrik yang disuplai. Energi Listrik dihabiskan untuk memutar rotor secara internal dan mengatasi rugi besi/tembaga serta gesekan gearbox tanpa menghasilkan torsi sapu luar yang signifikan.

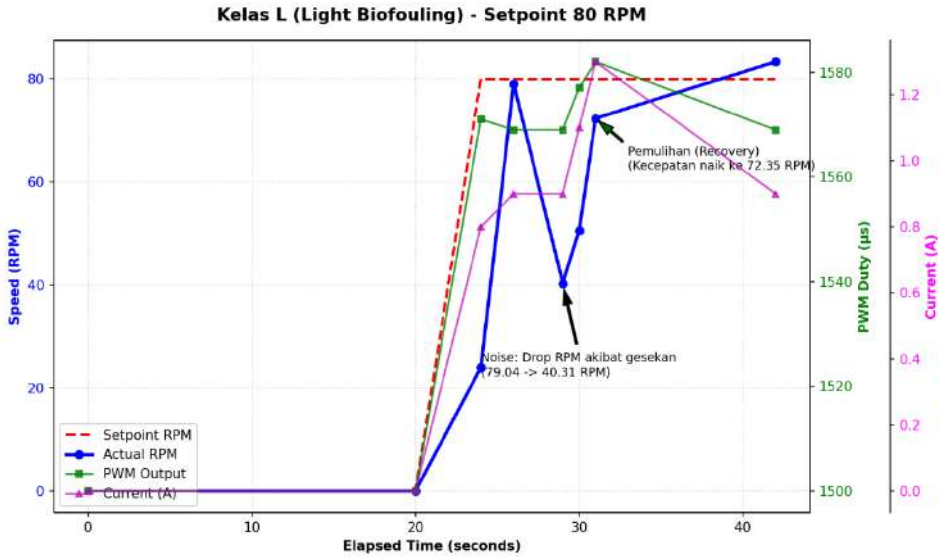
Pada kelas berat (kelas H) kontribusi arus tanpa beban turun menjadi hanya 8% dari total arus suplai. 92% energi listrik berhasil dikonversi menjadi torsi mekanis riil. Hal ini menyebabkan titik operasi motor bergeser mendekati titik efisiensi elektro-mekanis puncak. Pada beban puncak juga memberikan dampak tegangan sag, dengan asumsi arus total sistem mencapai ≈ 15 A terjadi rugi tegangan internal baterai LiPo sebesar $\Delta V = 15\ A \times 0.04\ \Omega = 0.6$ V. Penurunan tegangan ini menurunkan V_{bus} ke ≈ 16.2 V, yang membatasi kapasitas kecepatan maksimal secara teori motor BLDC dan mempercepat pengurasan baterai. Hubungan yang terjadi antara kopel gaya mekanis (gaya hisap sikat, gaya normal dan Torsi Gesek) menghasilkan efek hidrodinamika berupa gaya vakum yang menempelkan ROV ke pelat lambung kapal ($F_s = K_s \times \rho \times \omega^2$).

4.5.4 Analisis *Response System*

Pada sub bab ini disajikan analisis terperinci mengenai performa respons kendali PID terhadap gangguan (noise) mekanis ketika sikat pembersih ROV menyentuh lambung kapal dengan berbagai tingkat kekotoran.

4.5.4.1 Analisis Respon Sistem Pada Kelas L

Mengacu pada table 4.13 di dapatkan hasil sebagai berikut :



Gambar 4. 40 Grafik *Noise Response System* pada Kelas L

Pada kondisi awal, sistem mampu mempertahankan putaran sikat mendekati setpoint dengan kecepatan 79,04 RPM, arus 0,9 A, dan sinyal PWM sebesar 1569 µs (detik ke-26). Pada detik ke-29, saat sikat mulai bersentuhan dengan lapisan lumut tipis, terjadi gangguan (*step disturbance*) berupa peningkatan gaya gesek yang menyebabkan kecepatan motor turun drastis menjadi 40,31 RPM atau mengalami penurunan sebesar 49,0%.

Pengendali PID mendeteksi error sebesar 39,69 RPM dan merespons dengan meningkatkan sinyal PWM secara bertahap dari 1569 µs menjadi 1577 µs dan 1582 µs, yang diikuti peningkatan arus motor dari 0,9 A menjadi 1,1 A dan 1,3 A untuk menghasilkan torsi tambahan. Sistem kemudian berhasil memulihkan kecepatan hingga 83,34 RPM pada detik ke-42, dengan PWM kembali ke 1569 µs dan arus stabil di 0,9 A. Waktu pemulihan sekitar 12 detik menunjukkan respons yang relatif lambat, yang dipengaruhi oleh nilai penguatan integral yang kecil (Ki

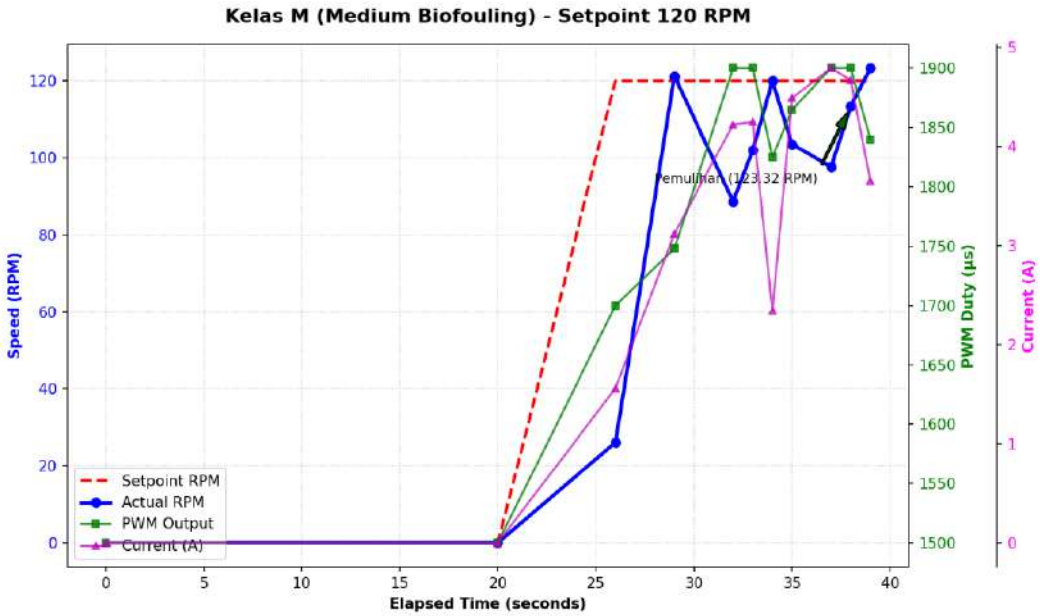
= 0,0001), sehingga koreksi terhadap error dilakukan secara bertahap namun tetap stabil

Tabel 4. 25 Metrik Peforma Sistem Pada Kelas L

Parameter Pengujian	Kelas L (Light - 80 RPM)
Setpoint Target (RPM)	80
Rise Time (tr*tr*) Awal	2.0 detik
Overshoot (Mp*Mp*) Awal	0.0% (Maks: 79.04 RPM)
Settling Time (ts*ts*) Awal	2.0 detik
Pre-Noise Speed (RPM)	79.04
Undershoot Terendah (RPM)	40.31
Kuat Noise (Penurunan RPM)	38.73 RPM (49.0%)
Settling Time (ts_rec*ts_rec*) Pemulihan	12.0 detik
PWM Maks. Saat Gangguan (μ*μ*s)	1582
Arus Puncak Listrik (Ipeak*Ipeak*)	1.3 A

4.5.4.2 Analisis Respon Sistem Pada Kelas M

Mengacu pada table 4.14 di dapatkan hasil sebagai berikut :



Gambar 4. 41 Grafik *Noise Response System* pada Kelas M

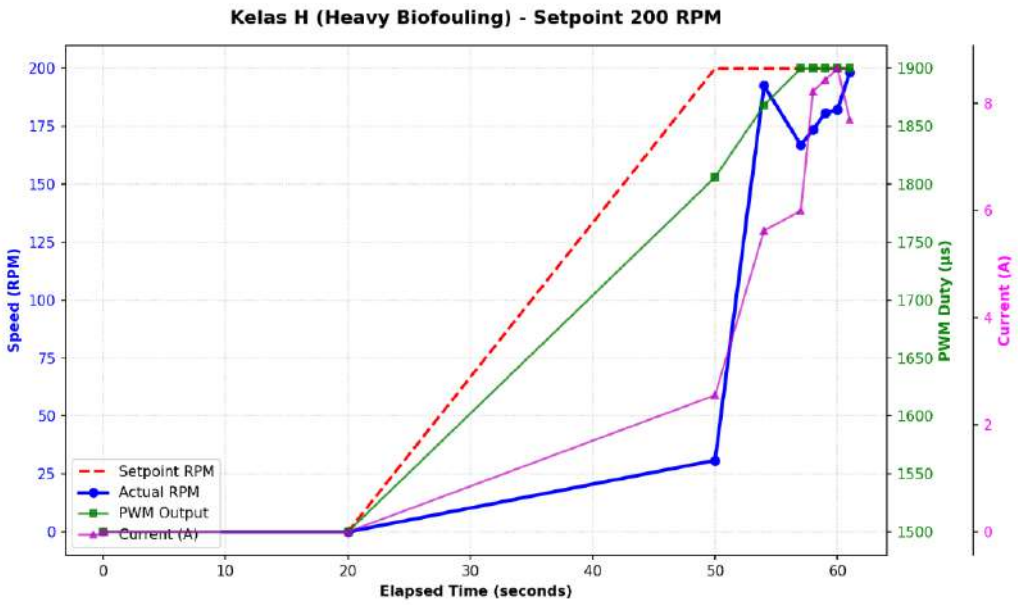
Pada kondisi awal, sistem mempertahankan putaran sikat pada 121,24 RPM dengan arus 3,12 A dan sinyal PWM 1748 µs (detik ke-29). Pada detik ke-32, saat sikat berkontak dengan kerak biofouling kategori sedang, terjadi gangguan (*step disturbance*) yang menyebabkan kecepatan turun menjadi 88,70 RPM atau berkurang sebesar 26,1%. Pengendali PID segera mendeteksi galat dan meningkatkan sinyal PWM hingga batas saturasi 1900 µs, sehingga arus motor meningkat dari 3,12 A menjadi 4,22 A dan 4,25 A untuk menghasilkan torsi tambahan. Respons tersebut memungkinkan sistem memulihkan kecepatan menjadi 119,98 RPM hanya dalam waktu sekitar 2 detik (detik ke-34), kemudian PWM diturunkan kembali menjadi 1825 µs dan arus berkurang menjadi 2,34 A. Selanjutnya, pada detik ke-35 hingga ke-37 terjadi gangguan berulang yang kembali menurunkan kecepatan hingga 103,54 RPM dan 97,65 RPM. PID kembali merespons dengan memberikan PWM maksimum 1900 µs, sehingga arus meningkat hingga 4,79 A dan kecepatan berhasil dipulihkan menjadi 123,32 RPM pada detik ke-39. Hasil ini menunjukkan bahwa PID mampu mempertahankan kestabilan putaran brush terhadap gangguan berulang dengan memanfaatkan kapasitas maksimum aktuator saat diperlukan.

Tabel 4. 26 Metrik Peforma Sistem Pada Kelas M

Parameter Pengujian	Kelas M (Medium - 120 RPM)
Setpoint Target (RPM)	120
Rise Time (tr*tr*) Awal	3.0 detik
Overshoot (Mp*Mp*) Awal	1.03% (Maks: 121.24 RPM)
Settling Time (ts*ts*) Awal	3.0 detik
Pre-Noise Speed (RPM)	121.24
Undershoot Terendah (RPM)	88.7
Kuat Noise (Penurunan RPM)	32.54 RPM (26.8%)
Settling Time (ts_rec*ts_rec*) Pemulihan	2.0 detik
PWM Maks. Saat Gangguan (µ*µ*s)	1900 (Jenuh)
Arus Puncak Listrik (Ipeak*Ipeak*)	4.79 A

4.5.4.3 Analisis Respon Sistem Pada Kelas H

Mengacu pada table 4.15 di dapatkan hasil sebagai berikut :



Gambar 4. 42 Grafik *Noise Response System* pada Kelas H

Pada kondisi awal, sistem mampu mempertahankan putaran sikat pada 192,54 RPM dengan arus 5,62 A dan sinyal PWM sebesar 1868 µs pada detik ke-54. Selanjutnya, pada detik ke-57 terjadi gangguan (*step disturbance*) akibat kontak brush dengan kerak *biofouling* keras berupa teritip yang menghasilkan torsi lawan cukup tinggi. Kondisi tersebut menyebabkan kecepatan motor turun menjadi 166,98 RPM atau mengalami penurunan sebesar 16,5%. Pengendali PID mendeteksi galat sebesar 33,02 RPM dan merespons dengan meningkatkan sinyal PWM hingga mencapai batas saturasi 1900 µs. Peningkatan beban mekanis tersebut juga menyebabkan arus motor meningkat dari 5,99 A menjadi 8,22 A, kemudian 8,43 A, hingga mencapai 8,65 A. Kondisi ini menunjukkan bahwa motor membutuhkan torsi yang lebih besar untuk mempertahankan putaran terhadap hambatan pembersihan. Meskipun menerima beban ekstrem, motor BLDC berhasil mengatasi torsi lawan dan meningkatkan kembali kecepatannya hingga 198,28 RPM pada detik ke-61, dengan arus sebesar 7,69 A. Waktu pemulihan sekitar 4 detik menunjukkan bahwa pengendali PID mampu mengembalikan kecepatan

mendekati setpoint serta menjaga kestabilan sistem pada kondisi *biofouling* dengan resistansi mekanis tinggi. Namun, kestabilan tersebut dicapai dengan peningkatan konsumsi arus sebagai konsekuensi kebutuhan torsi motor yang lebih besar.

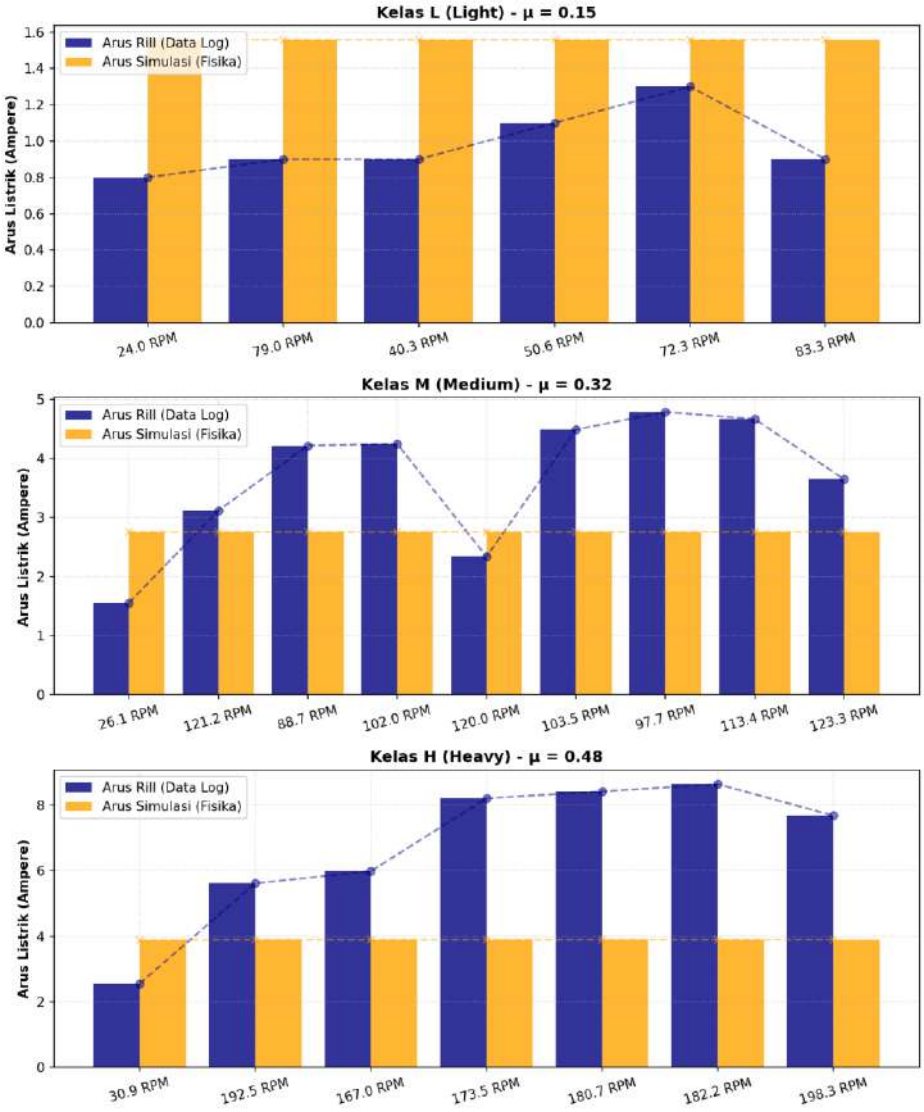
Tabel 4. 27 Metrik Peforma Sistem Pada Kelas H

Parameter Pengujian	Kelas H (Heavy - 200 RPM)
Setpoint Target (RPM)	200
Rise Time (tr*tr*) Awal	4.0 detik
Overshoot (Mp*Mp*) Awal	0.0% (Maks: 192.54 RPM)
Settling Time (ts*ts*) Awal	4.0 detik
Pre-Noise Speed (RPM)	192.54
Undershoot Terendah (RPM)	166.98
Kuat Noise (Penurunan RPM)	25.56 RPM (13.3%)
Settling Time (ts_rec*ts_rec*) Pemulihan	4.0 detik
PWM Maks. Saat Gangguan (μ*μ*s)	1900 (Jenuh)
Arus Puncak Listrik (Ipeak*Ipeak*)	8.65 A

4.5.4.4 Perbandingan Metode

Subbab ini membahas proses validasi model simulasi dengan membandingkan hasil simulasi matematis terhadap hasil implementasi pada sistem ROV *Hull Cleaning*. Validasi dilakukan dengan membandingkan karakteristik arus motor *brush* pada tiga kondisi biofouling, yaitu *Light*, *Medium*, dan *Heavy*. Parameter tersebut dipilih karena arus motor berkaitan langsung dengan perubahan beban dan kebutuhan torsi selama proses pembersihan. Tingkat kesesuaian antara model dan sistem aktual kemudian dievaluasi menggunakan *Mean Absolute Error* (MAE) serta persentase galat. Hasil perbandingan digunakan untuk mengetahui kemampuan model dalam merepresentasikan kondisi operasi aktual sekaligus mengidentifikasi faktor yang menyebabkan perbedaan antara simulasi dan implementasi. Dengan demikian, validasi tidak hanya menilai tingkat akurasi

model, tetapi juga menunjukkan keterbatasan model dalam menggambarkan kondisi dinamis selama proses *brushing*.



Gambar 4. 43 Grafik Perbandingan Riil & Simulasi

Pada Kelas *Light*, hasil simulasi cenderung mengestimasi arus lebih tinggi (*overestimation*) dibandingkan hasil pengujian aktual. Pada kecepatan 24,0 RPM, arus riil tercatat sebesar 0,80 A, sedangkan simulasi memprediksi 1,56 A dengan galat +0,76 A. Kondisi serupa terjadi pada kecepatan 79,04 RPM dan 83,34 RPM, ketika arus riil sebesar 0,90 A, sementara simulasi tetap menghasilkan 1,56 A dengan galat +0,66 A. Nilai *Mean Absolute Error* (MAE) sebesar 0,575 A atau 62,6%

menunjukkan bahwa model masih memberikan estimasi yang terlalu tinggi pada kondisi beban ringan. Perbedaan tersebut dipengaruhi oleh arus tanpa beban motor ((I_0=0,5) A) dan asumsi gaya gesek minimum pada model yang lebih besar dibandingkan kondisi aktual. Pada kondisi nyata, lumut tipis memiliki gaya adhesi dan gesekan yang relatif rendah ketika berada di dalam air, sehingga kebutuhan arus motor lebih kecil. Selain itu, model belum sepenuhnya mempertimbangkan karakteristik aliran air dan perubahan gaya kontak yang terjadi selama brush berputar.

Pada Kelas *Medium*, simulasi menghasilkan arus sebesar 2,76 A dan menunjukkan kesesuaian yang lebih baik pada kondisi tunak. Pada kecepatan 121,24 RPM, arus riil sebesar 3,12 A menghasilkan galat −0,36 A atau sekitar 11,5%. Namun, ketika terjadi gangguan akibat kontak dengan biofouling, arus aktual meningkat meskipun RPM mengalami penurunan. Pada kecepatan 88,70 RPM dan 103,54 RPM, arus riil masing-masing mencapai 4,22 A dan 4,49 A, sedangkan simulasi tetap menghasilkan 2,76 A. Kondisi tersebut menghasilkan galat −1,46 A dan −1,73 A. Nilai MAE sebesar 1,277 A atau 35,8% menunjukkan bahwa model cukup merepresentasikan kondisi tunak, tetapi belum mampu menggambarkan peningkatan beban dinamis selama kontak brush dengan biofouling. Kenaikan arus pada kondisi tersebut menunjukkan adanya perubahan torsi beban yang tidak direpresentasikan oleh model statis.

Pada Kelas *Heavy*, perbedaan antara simulasi dan pengujian semakin besar karena beban mekanis yang diterima brush lebih tinggi. Pada kondisi awal 192,54 RPM, arus riil sebesar 5,62 A, sedangkan simulasi hanya memprediksi 3,90 A dengan galat −1,72 A. Ketika RPM turun menjadi 166,98 RPM akibat gangguan, arus aktual meningkat menjadi 5,99 A dengan galat −2,09 A. Deviasi terbesar terjadi pada fase pemulihan ketika arus mencapai 8,65 A, sedangkan simulasi tetap berada pada 3,90 A sehingga menghasilkan galat −4,75 A. Nilai MAE sebesar 3,221 A atau 46,9% menunjukkan bahwa model gaya gesek statis belum mampu merepresentasikan *peak dynamic torque* akibat interaksi brush dengan kerak keras seperti teritip. Beban tersebut menyebabkan pengendali meningkatkan PWM untuk mempertahankan RPM, sehingga terjadi lonjakan arus yang tidak muncul pada simulasi statis.

Secara keseluruhan, hasil validasi menunjukkan bahwa kesenjangan antara model dan sistem aktual meningkat seiring bertambahnya tingkat biofouling. Model relatif mampu menggambarkan kondisi operasi tunak, tetapi kurang akurat ketika terjadi perubahan beban secara tiba-tiba. Hal ini menunjukkan bahwa model simulasi masih perlu dikembangkan dengan memasukkan komponen beban dinamis, perubahan gaya kontak, karakteristik hidrodinamika, serta respons kontrol tertutup. Dengan pengembangan tersebut, model diharapkan dapat merepresentasikan kondisi transien secara lebih realistis dan menghasilkan prediksi arus yang lebih mendekati hasil implementasi pada sistem ROV *Hull Cleaning*.

4.5.5 Hasil dari Brushing System

Dari sistem yang sudah diaplikasikan dan diuji dengan media plat besi seluas 60 cm x 60 cm dengan biofouling buatan yang didesain semirip mungkin dengan kondisi kapal.

Gambar 4. 44 Capture Pic sebelum Brushing kondisi *Heavy*

Gambar 4. 45 Hasil Plat sesudah Brushing kondisi *Heavy*

Berdasarkan lima data terakhir pengujian deteksi, diperoleh rata-rata koordinat pusat objek sebesar (623,40; 577,40) piksel dengan standar deviasi sebesar 0,49 piksel pada masing-masing sumbu X dan Y. Nilai standar deviasi yang rendah menunjukkan bahwa posisi objek hasil deteksi relatif konsisten pada setiap pengambilan citra. Selain itu, rata-rata luas area objek yang terdeteksi mencapai 40.376,60 piksel² dengan standar deviasi 54,38 piksel². Variasi yang kecil tersebut menunjukkan bahwa sistem pengolahan citra mampu mempertahankan konsistensi dalam menentukan posisi dan luas area biofouling. Dengan demikian, kedua parameter tersebut dapat digunakan sebagai informasi pendukung dalam identifikasi tingkat fouling dan pengambilan keputusan sistem kontrol ROV.

Pengujian dilakukan menggunakan pelat uji berukuran 40 cm × 30 cm dengan luas total 1200 cm² atau 0,12 m² sebagai representasi permukaan lambung kapal. Sistem pembersihan menggunakan sikat dengan diameter efektif 12 cm dan jari-jari efektif reffr_{eff} sebesar 0,06 m yang ditentukan berdasarkan panjang radial bulu sikat terluar. Dimensi pelat dan ukuran sikat menjadi parameter utama dalam mengevaluasi cakupan area pembersihan, distribusi gaya kontak, serta efektivitas mekanisme brushing terhadap permukaan biofouling.

Hasil pengukuran menunjukkan lebar area biofouling sebesar 200,94 piksel. Dengan faktor skala 0,05 cm/piksel, lebar tersebut setara dengan sekitar 10,05 cm. Nilai ini lebih kecil daripada diameter efektif sapuan sikat sebesar 12 cm, sehingga area target dapat tercakup dalam satu lintasan (*single pass*). Kesesuaian antara dimensi area fouling dan diameter sikat menunjukkan bahwa sistem memiliki cakupan pembersihan yang memadai. Hasil tersebut juga sesuai dengan Gambar 4.39, ketika noda biofouling pada bagian tengah pelat dapat dibersihkan secara menyeluruh dalam satu kali proses penyikatan tanpa memerlukan lintasan tambahan.

Analisis efektivitas pembersihan pada Gambar 4.39 menunjukkan perbedaan kinerja berdasarkan zona kontak sikat. Pada zona inti (*Center Core Zone*) dengan radius 0–4 cm dari pusat poros, tingkat kebersihan mencapai 100%. Kondisi ini dipengaruhi oleh gaya tekan normal yang lebih terpusat dan kerapatan bulu sikat yang tinggi, sehingga lapisan biofouling dapat terkikis hingga permukaan logam terekspos. Pada zona periferi (*Periphery Zone*) dengan radius 4–6 cm, efektivitas

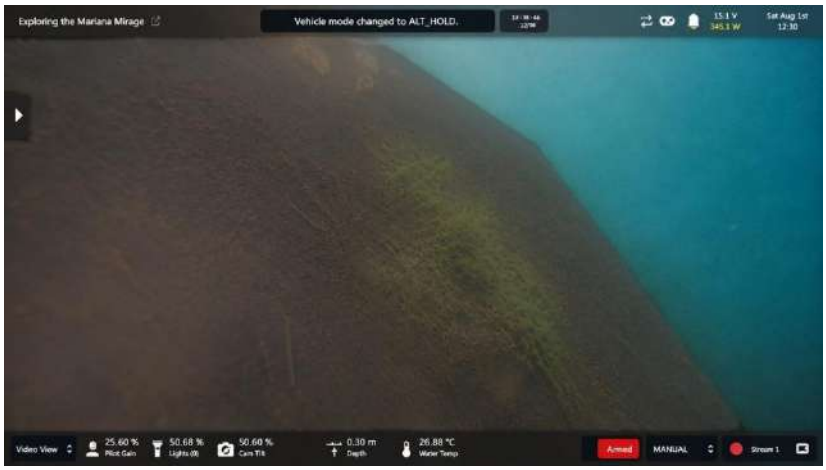
pembersihan berada pada kisaran 80–85%. Penurunan tersebut disebabkan oleh defleksi elastis bulu sikat selama rotasi yang mengurangi tekanan kontak lokal pada bagian tepi. Akibatnya, masih terdapat sedikit residu biofouling pada batas jalur sapuan. Secara keseluruhan, efisiensi pembersihan rata-rata dalam satu lintasan (*single-pass point efficiency*) mencapai 92,86%. Nilai tersebut menunjukkan bahwa sistem brushing mampu membersihkan sebagian besar area target dalam satu lintasan, sehingga mengurangi kebutuhan *multi-pass*, mempercepat proses pembersihan, dan meningkatkan efisiensi operasi ROV.

Gambar 4. 46 Plat sebelum *Brushing* kondisi *Medium*

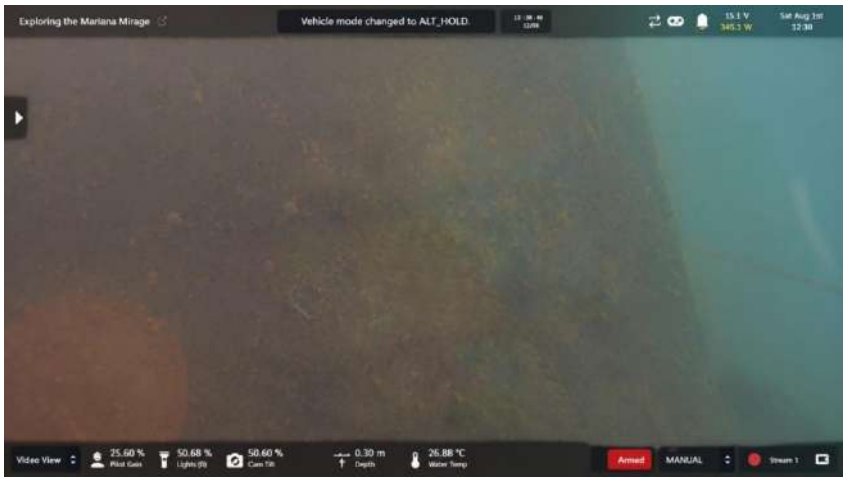
Gambar 4. 47 Plat sesudah *Brushing* kondisi

Berdasarkan hasil deteksi YOLO pada kelas *Heavy*, diperoleh rata-rata koordinat pusat target sebesar (838,00; 180,00) piksel dengan luas *bounding box* sebesar 55.374,00 piksel². Dengan asumsi bentuk *bounding box* berupa persegi, panjang sisi area terdeteksi diperkirakan sebesar 235,32 piksel. Menggunakan faktor kalibrasi 0,08 cm/piksel, ukuran tersebut setara dengan 18,83 cm dan menghasilkan estimasi luas area biofouling sebesar 354,39 cm². Lebar area biofouling sebesar 18,83 cm lebih besar dibandingkan diameter efektif sapuan sikat sebesar 12 cm. Kondisi tersebut menyebabkan satu kali lintasan (*single pass*) belum mampu mencakup seluruh permukaan fouling. Sikat hanya menjangkau bagian tengah area target, sedangkan pengotor masih dapat tertinggal pada sisi kiri dan kanan. Oleh karena itu, dimensi area fouling menjadi salah satu faktor penting dalam menentukan kebutuhan lintasan pembersihan.

Hasil evaluasi pada Gambar 4.41 menunjukkan bahwa sistem memiliki efektivitas tinggi dalam menghilangkan tonjolan tiga dimensi akibat kerak teritip (*barnacles*). Struktur cangkang yang menjadi salah satu penyebab peningkatan hambatan hidrodinamika berhasil diratakan dengan tingkat pembersihan mencapai 100%. Meskipun demikian, lapisan sisa kerak berupa *barnacle adhesive base* masih tertinggal pada permukaan logam. Kondisi tersebut menyebabkan tingkat kebersihan secara visual setelah satu kali lintasan hanya mencapai sekitar 65%. Perbedaan efektivitas juga terlihat pada distribusi zona kontak sikat. Zona inti dengan radius 0–4 cm dari pusat poros menghasilkan tingkat kebersihan sekitar 75–80% karena didukung gaya tekan normal sebesar 104,89 N yang menghasilkan tekanan kontak lebih besar terhadap permukaan. Sebaliknya, zona periferi dengan radius 4–6 cm hanya mencapai 45–55% akibat defleksi elastis bulu sikat selama proses rotasi. Defleksi tersebut menyebabkan tekanan kontak pada bagian tepi lebih rendah sehingga residu perekat lebih sulit dihilangkan. Secara keseluruhan, efisiensi pembersihan *single-pass* mencapai 65%. Hasil ini menunjukkan bahwa sistem mampu merontokkan dan meratakan kerak keras secara efektif, tetapi belum mampu menghilangkan seluruh residu perekat teritip. Dengan demikian, *multi-pass* masih diperlukan apabila target pembersihan mengharuskan permukaan logam terbebas dari *adhesive base*. Kondisi ini juga menunjukkan bahwa strategi lintasan harus mempertimbangkan ukuran area fouling dan diameter sapuan sikat.



Gambar 4. 48 Kondisi Sebelum *Brushing* kondisi *Light*



Gambar 4. 49 Kondisi Sesudah *Bruhsing* kondisi *Light*

Berdasarkan hasil deteksi YOLO, diperoleh rata-rata koordinat pusat target sebesar (967,00; 552,00) piksel dengan luas bounding box sebesar 24.869,00 piksel². Dengan asumsi bentuk persegi, diperoleh estimasi panjang sisi kotak pembatas sebesar 157,70 piksel. Menggunakan faktor kalibrasi 0,05 cm/piksel, dimensi fisik area biofouling diperkirakan memiliki panjang sisi 7,88 cm dengan luas 62,17 cm², sedangkan pada faktor kalibrasi 0,08 cm/piksel diperoleh panjang sisi 12,62 cm dan luas 159,16 cm². Berdasarkan kondisi operasi sistem, nilai kalibrasi 0,08 cm/piksel digunakan sebagai acuan karena merepresentasikan jarak kerja kamera terhadap permukaan lambung saat proses pembersihan. Hasil tersebut menunjukkan bahwa lebar area biofouling sebesar 12,62 cm sangat mendekati diameter efektif sikat sebesar 12 cm, sehingga seluruh area target dapat dicakup secara optimal dalam satu kali lintasan (single pass).

Hasil evaluasi menunjukkan bahwa efisiensi pembersihan visual mencapai sekitar 100% atau minimal 99,5%, ditandai dengan terlepasnya seluruh lapisan lumut hijau dari permukaan pelat logam setelah satu kali proses penyikatan. Tingginya tingkat keberhasilan ini disebabkan oleh karakteristik lumut lunak yang memiliki gaya adhesi mekanis relatif rendah terhadap permukaan baja, sehingga kecepatan sikat 80 RPM dengan tekanan normal yang dihasilkan sistem sudah cukup untuk melepaskan biofouling secara menyeluruh. Analisis distribusi area kontak menunjukkan bahwa zona inti dengan radius 0–4 cm dari pusat poros mencapai tingkat kebersihan 100%, di mana seluruh lapisan lumut berhasil dikikis hingga permukaan logam terekspos. Sementara itu, pada zona periferi dengan radius 4–6 cm, tingkat kebersihan tetap berada pada kisaran 99–100%. Meskipun ujung bulu sikat mengalami defleksi elastis selama proses rotasi, gaya gesek yang dihasilkan masih lebih besar daripada gaya adhesi lumut lunak, sehingga tidak ditemukan residu maupun pola cincin biofouling pada area tepi lintasan sikat. Hasil ini menunjukkan bahwa sistem brushing memiliki efektivitas yang sangat tinggi untuk membersihkan biofouling kategori Light hanya dalam satu kali lintasan.

BAB 5
KESIMPULAN DAN SARAN

5.1 Kesimpulan

1. Perencanaan Desain dan Sistem Brushing: Sistem pembersih (*brushing*) lambung kapal dirancang menggunakan sikat (*brush*) yang terintegrasi dengan *planetary gearbox* rasio 5:1. Desain ini terbukti mampu meningkatkan torsi sikat hingga 4,25 kali lipat dibandingkan dengan torsi nominal motor, sehingga putaran *brush* dapat dipertahankan secara stabil pada variasi setpoint kecepatan (80 RPM, 120 RPM, dan 200 RPM) tanpa mengalami kondisi macet (*stall*). Selain meningkatkan kemampuan mekanis *brush*, rotasi sikat juga menghasilkan gaya hisap dinamis (*dynamic suction force*) yang berkontribusi terhadap peningkatan gaya tekan normal (FN) ROV pada permukaan lambung kapal, sehingga meningkatkan kestabilan kontak mekanis sikat dengan permukaan lambung kapal agar proses pembersihan berjalan lebih efektif.
2. Pengaruh YOLO dalam Penentuan Kelas Sistem Kontrol: Implementasi algoritma *You Only Look Once* (YOLO) berpengaruh sebagai sistem identifikasi dan klasifikasi tingkat kekotoran lambung kapal (*biofouling*) menjadi tiga kelas (Light, Medium, dan Heavy) secara *real-time* dengan rata-rata waktu inferensi sebesar 99 ms. Hasil klasifikasi ini digunakan sebagai acuan dinamis untuk menentukan setpoint kecepatan putaran motor *brush*. Guna meningkatkan keandalan proses klasifikasi dan mencegah terjadinya fluktuasi setpoint (*chattering*) akibat gangguan deteksi sesaat (*noise*), diimplementasikan algoritma *counter voting* dengan syarat minimal lima frame berturut-turut menghasilkan kelas yang sama sebelum keputusan transisi setpoint diambil. Selain itu, integrasi data IMU Pixhawk berhasil memastikan bahwa *brush* hanya diaktifkan ketika orientasi ROV telah stabil dan menempel pada lambung kapal.
3. Optimasi Kecepatan dan Performa Respons Motor: Optimasi pengaturan kecepatan sikat dan performa respons motor dicapai menggunakan

pengendali PI (*Proportional-Integral*) secara adaptif untuk mempertahankan kecepatan sikat sesuai kelas *biofouling*. Pengendali PI secara aktif mengoreksi sinyal PWM yang diberikan ke ESC untuk mempertahankan putaran *brush* ketika terjadi peningkatan torsi akibat gaya gesek *biofouling*. Karakteristik respons transien sistem sikat dioptimalkan berdasarkan karakteristik beban masing-masing kelas kekotoran dengan rincian sebagai berikut:

a. Kelas Light: Menghasilkan respons bertipe *overdamped* dengan waktu pemulihan sekitar 12 detik untuk menjaga kestabilan operasi dengan rata-rata konsumsi arus rendah sebesar 1,2 A.
b. Kelas Medium: Menghasilkan respons *underdamped* moderat dengan waktu pemulihan sekitar 2 detik melalui pemanfaatan saturasi PWM hingga 1900 μs dan arus rata-rata 2,5 A.
c. Kelas Heavy: Menghasilkan respons yang lebih agresif dengan mengizinkan *overshoot* awal guna menghasilkan momentum mekanis yang cukup untuk merontokkan *biofouling* keras (seperti teritip) dengan arus rata-rata 5,8 A (arus puncak 8,65 A) dan waktu pemulihan sekitar 4 detik.
d. Validasi Model: Hasil validasi model simulasi terhadap implementasi riil menunjukkan adanya deviasi pada kondisi dinamis dengan nilai *Mean Absolute Error* (MAE) masing-masing sebesar 0,575 A pada kelas Light, 1,277 A pada kelas Medium, dan 3,221 A pada kelas Heavy. Deviasi ini terjadi karena efek tumbukan dinamis bulu sikat dan perubahan gaya gesek selama pembersihan yang belum sepenuhnya terwakili dalam model simulasi statis.

5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, pengembangan model simulasi dapat dilakukan dengan mengintegrasikan persamaan pengendali PI dan fungsi transfer motor ke dalam modul simulasi gaya rov dan hasil pengukuran riil. Pendekatan ini diharapkan mampu merepresentasikan karakteristik sistem secara lebih realistis, termasuk fenomena *overshoot*, *transient load*, serta respons dinamis

motor selama proses pembersihan, sehingga hasil simulasi memiliki tingkat kesesuaian yang lebih tinggi terhadap kondisi implementasi.

Pada aspek sistem kendali, kinerja pengendali dapat ditingkatkan dengan menerapkan metode *anti-windup* untuk mencegah kejenuhan integrator ketika sinyal PWM mencapai batas maksimum sebesar 1900 μs. Selain itu, penerapan *dynamic gain scheduling* dengan penyesuaian parameter Kp dan Ki pada setiap kelas biofouling diharapkan mampu meningkatkan kecepatan respons sistem, khususnya pada kelas Light yang masih menunjukkan waktu pemulihan relatif lebih lambat dibandingkan kelas lainnya.

Dari sisi perangkat keras, sistem dapat dikembangkan dengan menambahkan rangkaian *current limiter* pada jalur ESC atau ESP32 sebagai mekanisme proteksi terhadap kondisi arus berlebih. Proteksi ini diperlukan untuk mengurangi risiko kerusakan pada baterai, motor BLDC, maupun komponen elektronika ketika motor mengalami beban berlebih atau kondisi *stall*, yang pada penelitian ini menghasilkan arus puncak lebih dari 8 A.

Pada sistem visi, performa algoritma YOLO masih dapat ditingkatkan melalui proses optimasi model menggunakan TensorRT atau ONNX sehingga waktu inferensi dapat diturunkan hingga di bawah 30 ms. Selain itu, penambahan kelas deteksi "Clean Hull" disarankan agar sistem mampu mengidentifikasi area lambung yang telah bersih. Dengan adanya kelas tersebut, brush dapat dihentikan secara otomatis setelah proses pembersihan selesai sehingga mengurangi konsumsi energi, memperpanjang umur sikat, serta meminimalkan risiko kerusakan pada lapisan cat *anti-fouling* kapal.

DAFTAR PUSTAKA

Akram, W., Casavola, A., Kapetanović, N., & Miškovic, N. (2022). A Visual Servoing Scheme for Autonomous Aquaculture Net Pens Inspection Using ROV. *Sensors*, *22*(9). https://doi.org/10.3390/s22093525

Angga, A. T. N., Ravi, A. M., Priyambodo, D., Realdo, A. M., Ivanuri, F., Safitri, L. N., Zakariz, N. P., & Messiah, M. S. (2021). Use Of ACS 712ELC-5A Current Sensor on Overloaded Load Installation Safety System. *Applied Technology and Computing Science Journal*, *4*(1), 47–55. https://doi.org/10.33086/atcsj.v4i1.2088

Direktorat Jenderal Perhubungan Laut. (2017). *Hk-103-1-3-Djpl-17-Prosedur-Perlimbungan-Kapal-Berbendera-Indonesia*.

Eltayeb, A., Ahmed, G., Imran, I. H., Alyazidi, N. M., & Abubaker, A. (2024). Comparative Analysis: Fractional PID vs. PID Controllers for Robotic Arm Using Genetic Algorithm Optimization. *Automation*, *5*(3), 230–245. https://doi.org/10.3390/automation5030014

Feng, W., Ma, C., & Liang, J. (2023). Overview of Underwater Robots for Hull Cleaning. *Journal of Engineering Research and Reports*, *25*(9), 134–141. https://doi.org/10.9734/jerr/2023/v25i9987

Kabanov, A., Kramar, V., & Ermakov, I. (2021). Design and modeling of an experimental rov with six degrees of freedom. *Drones*, *5*(4). https://doi.org/10.3390/drones5040113

Kuo, C. L., Pu, Y. C., & Chen, Q. A. (2022). Position Tracking of an Underwater Robot Based on Floating-Downing PI Control. *Processes*, *10*(11). https://doi.org/10.3390/pr10112346

Kuruvilla, J., Joseph, M., & Stalin, K. (2023). Underwater Inspection and Exploration Rov. *International Journal of Advances in Engineering and Management (IJAEM)*, *5*(6), 472. https://doi.org/10.35629/5252-0506472477

Mannix, E. J., Wei, S., A. Woodham, B., Wilkinson, P., & Robinson, A. P. (2021). Automating the assessment of biofouling in images using expert agreement as

a gold standard. *Scientific Reports*, *11*(1). https://doi.org/10.1038/s41598-021-81011-2

Muhammad Nabel Al Fayyed, Rifdah Mayhasna Nur Alayya, Wildan Abu Bakar Sidiq, & Tatyantoro Andrasto. (2025). Optimasi PID Kontrol BLDC Menggunakan Metode Ziegler-Nichols. *Jurnal Ilmiah Teknik Mesin, Elektro dan Komputer*, *5*(2), 01–09. https://doi.org/10.51903/juritek.v5i2.4212

Palanisamy, R. P., Pyun, D. K., & Findikoglu, A. T. (2024). Accurate Ultrasonic Thickness Measurement for Arbitrary Time-Variant Thermal Profile. *Sensors*, *24*(16). https://doi.org/10.3390/s24165304

Redmon, J., Farhadi, A., Seitz, S., & Tatlock, Z. (2021). *Practical Computer Vision*.

S, S. K., Kumar, N. N., T, Y. N., & Kumar, H. (2021). Automated Hull Cleaning and Monitoring Robot. *International Research Journal of Engineering and Technology*. www.irjet.net

Song, C., & Cui, W. (2021). *Review of Underwater Ship Hull Cleaning Technologies*. https://doi.org/10.1007/s11804-020-00157-z/Published

Song, P., Zhao, L., Li, H., Xue, X., & Liu, H. (2024). RSE-YOLOv8: An Algorithm for Underwater Biological Target Detection. *Sensors*, *24*(18). https://doi.org/10.3390/s24186030

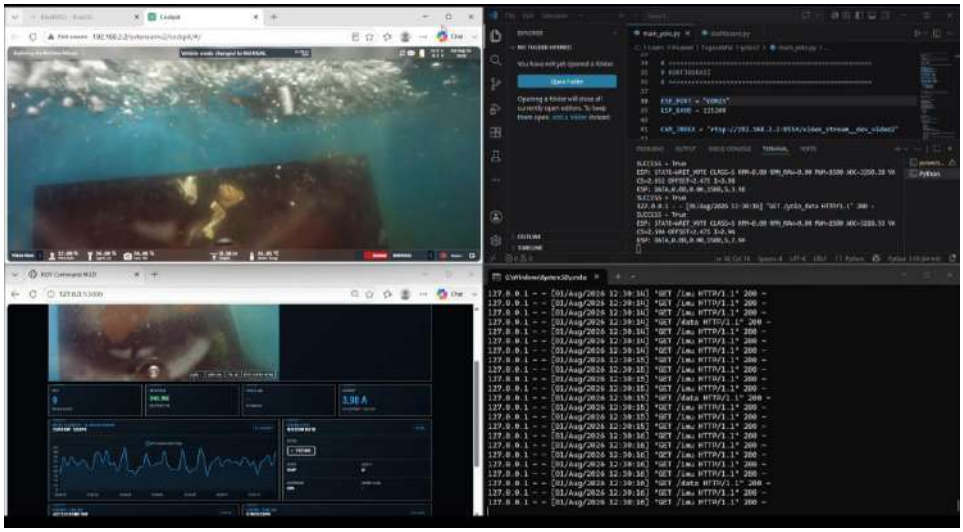
Triyatna, T., & Ardiansyah, S. (t.t.). *PROTOTYPE SISTEM ABSENSI SISWA/I DENGAN MENGGUNAKAN SENSOR RFID BERBASIS ARDUINO UNO*.

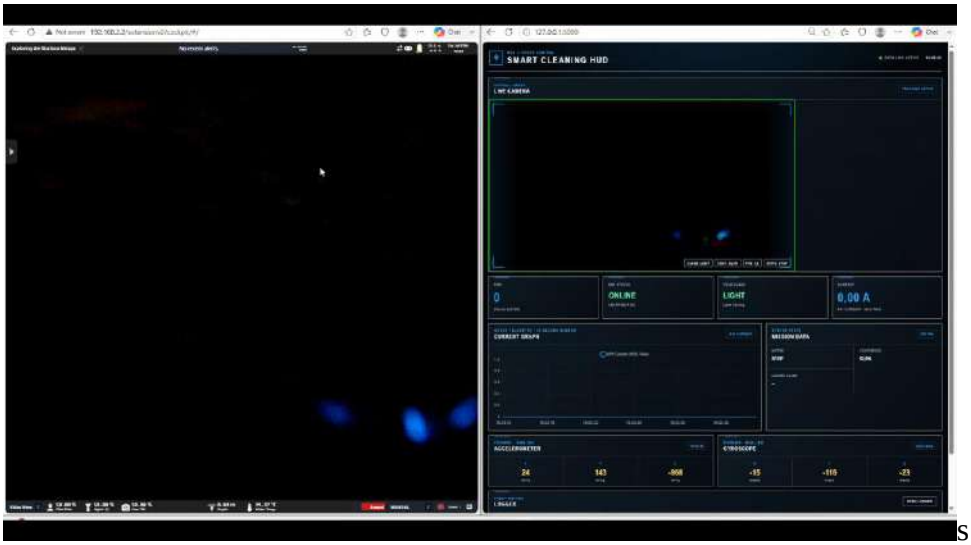
Wei, K., Zhang, Z., & Zhou, Z. (2024). Design of a new composite underwater hull cleaning robot. *Journal of Sea Research*, *198*. https://doi.org/10.1016/j.seares.2024.102488

Yamamoto, H., Nishida, Y., Matsuo, T., & Ishii, K. (2022). *Motion Control of a Ship Hull Cleaning Robot*.

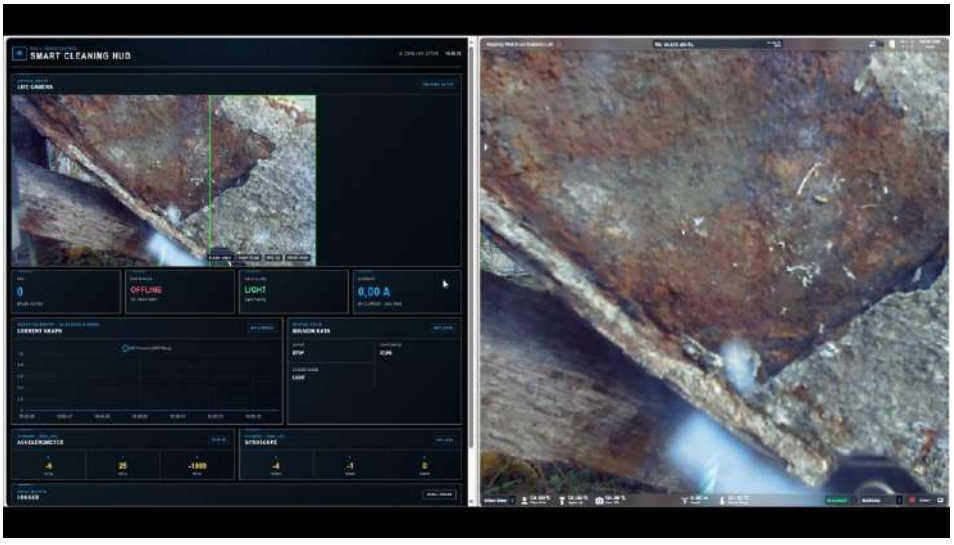
Zhafirah, T. A., Agus Khumaidi, Zindhu Maulana Ahmad Putra, Dwi Sasmita Aji Pambudi, Ryan Yudha Adhitya, & Didik Sukoco. (2024). Perancangan Keputusan Cut Off Baterai ROV menggunakan Metode Decision Tree. *Jurnal Elektronika dan Otomasi Industri*, *11*(1), 287–298. https://doi.org/10.33795/elkolind.v11i1.5469

LAMPIRAN





s



time	class	locked_cl	system_st	setpoint	rpm	pwm	mode	current
07/19/26:2	H	H	WAIT_DEL	0	0	1500	H	0
07/19/26:2	H	H	RUNNING	0	0	1500	H	0
07/19/26:2	H	H	RUNNING	0	0	1500	H	0
07/19/26:2	H	H	RUNNING	0	0	1500	H	0
07/19/26:2	L	H	RUNNING	0	0	1500	H	0
07/19/26:2	H	H	RUNNING	0	0	1500	H	0
07/19/26:2	H	H	RUNNING	0	0	1500	H	0
07/19/26:2	L	H	RUNNING	0	0	1500	H	0
07/19/26:2	L	H	RUNNING	0	0	1500	H	0
07/19/26:2	H	H	RUNNING	200	44.45	1814	H	3.97
07/19/26:2	H	H	RUNNING	200	44.45	1814	H	3.97
07/19/26:2	H	H	RUNNING	200	44.45	1814	H	3.97
07/19/26:2	H	H	RUNNING	200	44.45	1814	H	3.97
07/19/26:2	H	H	RUNNING	200	44.45	1814	H	3.97
07/19/26:2	H	H	RUNNING	200	44.45	1814	H	3.97
07/19/26:2	H	H	RUNNING	200	44.45	1814	H	3.97
07/19/26:2	H	H	RUNNING	200	29.99	1831	H	3.06
07/19/26:2	M	H	RUNNING	200	29.99	1831	H	3.06
07/19/26:2	H	H	RUNNING	200	29.99	1831	H	3.06
07/19/26:2	H	H	RUNNING	200	29.99	1831	H	3.06
07/19/26:2	H	H	RUNNING	200	29.99	1831	H	3.06
07/19/26:2	H	H	RUNNING	200	29.99	1831	H	3.06
07/19/26:2	H	H	RUNNING	200	29.99	1831	H	3.06
07/19/26:2	H	H	RUNNING	200	89.74	1851	H	5.52
07/19/26:2	H	H	RUNNING	200	89.74	1851	H	5.52
07/19/26:2	-	H	RUNNING	200	89.74	1851	H	5.52
07/19/26:2	-	H	RUNNING	200	89.74	1851	H	5.52
07/19/26:2	-	H	RUNNING	200	89.74	1851	H	5.52
07/19/26:2	H	H	RUNNING	200	89.74	1851	H	5.52
07/19/26:2	H	H	RUNNING	200	89.74	1851	H	5.52
07/19/26:2	H	H	RUNNING	200	147.84	1864	H	3.91
07/19/26:2	H	H	RUNNING	200	147.84	1864	H	3.91
07/19/26:2	H	H	RUNNING	200	147.84	1864	H	3.91
07/19/26:2	H	H	RUNNING	200	147.84	1864	H	3.91
07/19/26:2	H	H	RUNNING	200	147.84	1864	H	3.91
07/19/26:2	H	H	RUNNING	200	147.84	1864	H	3.91
07/19/26:2	H	H	RUNNING	200	147.84	1864	H	3.91
07/19/26:2	H	H	RUNNING	200	189	1879	H	7.47
07/19/26:2	H	H	RUNNING	200	189	1879	H	7.47
07/19/26:2	H	H	RUNNING	200	189	1879	H	7.47
07/19/26:2	H	H	RUNNING	200	189	1879	H	7.47
07/19/26:2	H	H	RUNNING	200	189	1879	H	7.47
07/19/26:2	H	H	RUNNING	200	189	1879	H	7.47
07/19/26:2	H	H	RUNNING	200	189	1879	H	7.47
07/19/26:2	H	H	RUNNING	200	209.77	1894	H	5.68
07/19/26:2	H	H	RUNNING	200	204.91	1900	H	8.51
07/19/26:2	H	H	RUNNING	200	204.91	1900	H	8.51
07/19/26:2	H	H	RUNNING	200	204.91	1900	H	8.51
07/19/26:2	H	H	RUNNING	200	204.91	1900	H	8.51
07/19/26:2	H	H	RUNNING	200	204.91	1900	H	8.51
07/19/26:2	H	H	RUNNING	200	204.91	1900	H	8.51

BlueRobotics

M200 and M500 Motor User Guide

The M200 and M500 are subsea motors designed to run in harsh ocean conditions, making them perfect for use on USVs (like the BlueBoat), AUVs, and in custom projects such as actuators and subsea tooling. This guide explains how to use and service the M200 and M500.

M200 Motor

Subsea motor for USVs, AUV, and more

$185.00 – $195.00

Select options

M500 Motor

High-power subsea motor for vehicles, tooling, and more

$650.00

Add to Cart

Important Safety Notes

Always practice caution when working with electricity in water.

Keep body parts away from a spinning motor to avoid injury.

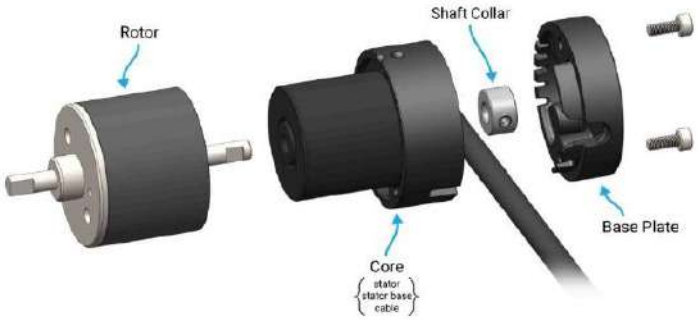
Do not operate the motor for more than 10 seconds outside of water (while dry) at a time. The bearings require water for lubrication and may be damaged if the motor is operated for an extended period of time while dry.

Most threadlockers are not chemically compatible with polycarbonate and will damage the motor if used on any of the screws. Only use a threadlocker that is chemically compatible with polycarbonate on the mounting screws if necessary.

A clicking sound or noisy motor is normal, especially when operated while dry. Most noise and vibration go away once the motor is operated in water.

Parts of a Motor

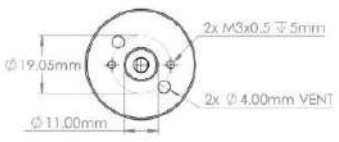
The M200 and M500 are outrunner motors. This means the entire outer rotor spins while the inner stator remains stationary. The following diagram shows the components of the motors.



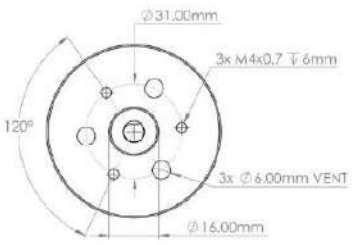
Rotor Attachments

The top of the rotor has threaded holes that can be used to attach propellers, tools, or other driven components. The two larger unthreaded holes are vents for drainage. Refer to the diagrams below for thread sizes, hole patterns, and maximum screw engagement depths for each motor.

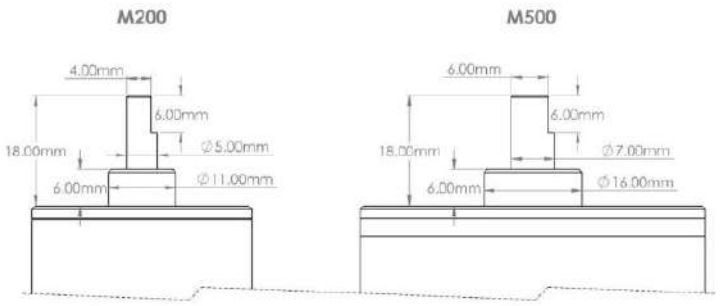
M200



M500



The rotor shaft has a drive flat—a flat section milled into the shaft that provides a positive surface for a fastener to bear against, preventing an attached component from slipping during operation.

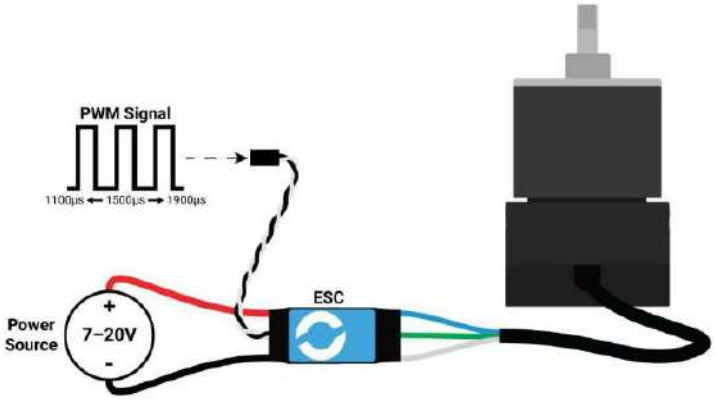


Operating the Motor

To operate a motor you need:

- The motor
- A brushless electronic speed controller (ESC) – always required, even if you don't need variable speed or direction control. Unlike brushed motors, brushless motors need an ESC to energize the motor phases at the correct timing.
- A power source – a battery or power supply within the operating voltage range of your motor.
- A signal to control the ESC – any device that can output a PWM signal, such as a microcontroller, RC receiver, or the Thruster Commander. The signal tells the ESC how fast and in which direction to run the motor.

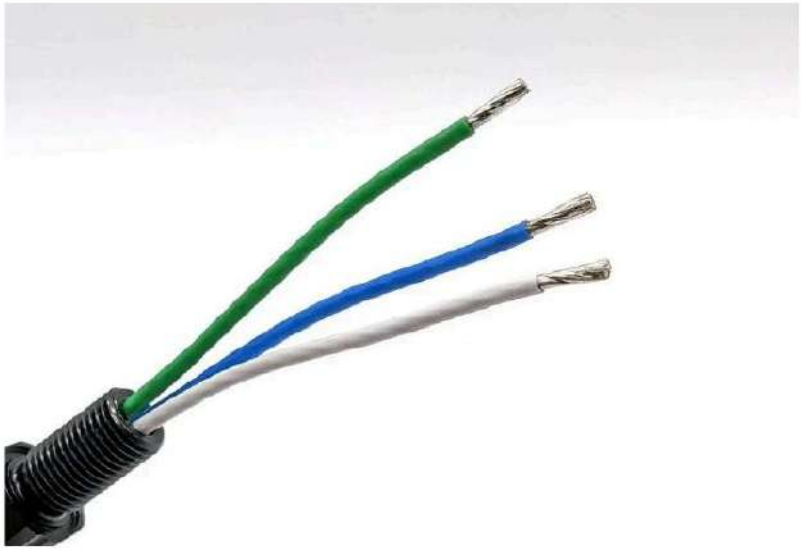
The following diagram shows how everything is connected:



The following instructions cover operation with a Blue Robotics ESC. If you're using a third-party ESC, the wiring is the same but refer to your ESC's documentation for initialization and signal details.

Connecting to an ESC

The motor cable has three conductors that terminate in tinned wire ends.



Connect these to the three motor phase wires on your ESC. The wires can be soldered together or connected using bullet connectors or screw terminal blocks. The order does not matter—the motor will spin regardless of how the three wires are connected. If the motor spins in the wrong direction, swap any two of the three wires.

Connecting to a Power Source

Connect the red (positive) and black (ground) ESC power wires to your power source. Make sure the voltage is within the operating range for your motor:

M200: 7–20V

M500: 7–24V

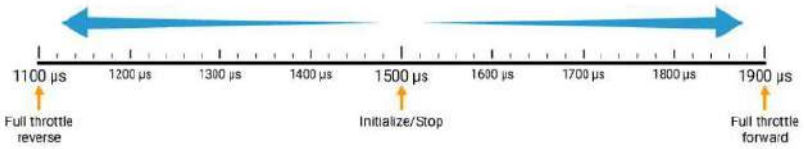
When the ESC is connected to power you should hear three beeps in rising pitch indicating the ESC is on and ready.

Sending a Signal

Connect the white ESC signal wire to your signal source, and the black (ground) ESC wire to a ground pin on the signal source. Send a PWM signal of 1500 µs to initialize the ESC. When initialized you will hear two more beeps—the first confirms a throttle signal is detected, the second confirms the correct 1500 µs signal and that the ESC is fully initialized.

Once initialized, the motor speed and direction is controlled by varying the PWM signal:

- **1500 µs to stop**
- **1525–1900 µs for forward**
- **1475–1100 µs for reverse**



Do not run the motor at high speed for more than 10 seconds outside of water. The bearings require water for lubrication and may be damaged if run dry for extended periods.

time	class	locked_class		system_state		setpoint	rpm	pwm	mode	current	accel_x	
	accel_y	accel_z	gyro_x	gyro_y	gyro_z							
07/19/26:23:26	H	H		WAIT_DELAY		0	0	1500	H	-0.99	9	-3
-1004	-2	1		0								
07/19/26:23:26	H	H		WAIT_DELAY		0	0	1500	H	-0.99	0	-4
-1007	-1	3		-1								
07/19/26:23:26	H	H		WAIT_DELAY		0	0	1500	H	-0.99	3	-7
-1008	-4	3		0								
07/19/26:23:26	H	H		WAIT_DELAY		0	0	1500	H	-0.99	8	-4
-1008	-4	33		0								
07/19/26:23:26	H	H		WAIT_DELAY		0	0	1500	H	-0.99	9	-4
-1009	-29	-14		8								
07/19/26:23:26	H	H		WAIT_DELAY		0	0	1500	H	-0.99	9	-1
-1004	-9	40		-14								
07/19/26:23:26	H	H		WAIT_DELAY		0	0	1500	H	-1.33	9	-2
-1000	1	-728		294								
07/19/26:23:26	H	H		WAIT_DELAY		0	0	1500	H	-1.33	-47	
73	-1110	-919		99	-291							
07/19/26:23:26	H	H		WAIT_DELAY		0	0	1500	H	-1.33	-21	
150	-975	-1930		168	-120							
07/19/26:23:26	H	H		WAIT_DELAY		0	0	1500	H	-1.33	7	
213	-997	-2060		327	-371							
07/19/26:23:26	H	H		WAIT_DELAY		0	0	1500	H	-1.33	-35	
506	-865	-2152		-229	526							
07/19/26:23:26	H	H		WAIT_DELAY		0	0	1500	H	-1.33	-52	
592	-838	-1430		-7	188							
07/19/26:23:26	H	H		RUNNING		0	0	1500	H	-1.33	-5	
737	-682	-1528		113	-144							
07/19/26:23:26	H	H		RUNNING		200	0	1802	H	-1.1	-29	
870	-497	-1769		482	-263							
07/19/26:23:26	H	H		RUNNING		200	0	1802	H	-1.1	-6	
900	-467	-267		0	5							
07/19/26:23:26	L	H		RUNNING		200	0	1802	H	-1.1	-5	
896	-477	238		3	13							
07/19/26:23:26	H	H		RUNNING		200	0	1802	H	-1.1	1	
875	-499	372		35	-16							
07/19/26:23:26	H	H		RUNNING		200	0	1802	H	-1.1	-2	
854	-538	71		5	4							
07/19/26:23:26	L	H		RUNNING		200	0	1802	H	-1.1	1	
861	-531	136		10	-2							
07/19/26:23:26	L	H		RUNNING		200	0	1802	H	-1.1	1	
842	-549	-24		-10	35							

07/19/26:23:26 H H RUNNING 200 44.45 1814 H 3.97 4
822 -582 524 104 -117
07/19/26:23:26 H H RUNNING 200 44.45 1814 H 3.97 21
805 -607 571 -206 60
07/19/26:23:26 H H RUNNING 200 44.45 1814 H 3.97 2
789 -623 915 -117 31
07/19/26:23:26 H H RUNNING 200 44.45 1814 H 3.97 -27
722 -716 803 -58 -40
07/19/26:23:26 H H RUNNING 200 44.45 1814 H 3.97 4
635 -770 1157 -74 -6
07/19/26:23:26 H H RUNNING 200 44.45 1814 H 3.97 -49
647 -786 1015 107 -406
07/19/26:23:26 H H RUNNING 200 44.45 1814 H 3.97 -21
472 -858 1959 -2 -13
07/19/26:23:26 H H RUNNING 200 29.99 1831 H 3.06 -32
416 -916 45 0 -45
07/19/26:23:26 M H RUNNING 200 29.99 1831 H 3.06 -1
376 -899 155 -117 141
07/19/26:23:26 H H RUNNING 200 29.99 1831 H 3.06 -79
405 -933 849 -123 259
07/19/26:23:26 H H RUNNING 200 29.99 1831 H 3.06 -31
265 -974 904 41 -449
07/19/26:23:26 H H RUNNING 200 29.99 1831 H 3.06 -10
139 -991 1100 16 -56
07/19/26:23:26 H H RUNNING 200 29.99 1831 H 3.06 -37
114 -998 76 27 -12
07/19/26:23:26 H H RUNNING 200 29.99 1831 H 3.06 -69
147 -1003 86 18 -164
07/19/26:23:26 H H RUNNING 200 89.74 1851 H 5.52 -19
127 -1009 191 5 8
07/19/26:23:26 H H RUNNING 200 89.74 1851 H 5.52 -50
88 -1021 -351 75 56
07/19/26:23:26 - H RUNNING 200 89.74 1851 H 5.52 -70
243 -975 -1553 92 85
07/19/26:23:26 - H RUNNING 200 89.74 1851 H 5.52 25
439 -852 -2431 621 191
07/19/26:23:26 - H RUNNING 200 89.74 1851 H 5.52 19
585 -758 -2445 -172 110
07/19/26:23:26 H H RUNNING 200 89.74 1851 H 5.52 25
753 -635 -1695 3 -55
07/19/26:23:26 H H RUNNING 200 89.74 1851 H 5.52 -31
839 -518 -1624 277 -158

07/19/26:23:26	H	H	RUNNING	200	147.84	1864	H	3.91	-5
968	-333	-1379	148	-90					
07/19/26:23:26	H	H	RUNNING	200	147.84	1864	H	3.91	25
959	-282	-1155	128	-53					
07/19/26:23:26	H	H	RUNNING	200	147.84	1864	H	3.91	-29
975	-191	-1035	-141	-7					
07/19/26:23:26	H	H	RUNNING	200	147.84	1864	H	3.91	-9
1014	-114	-744	-63	-16					
07/19/26:23:26	H	H	RUNNING	200	147.84	1864	H	3.91	-2
989	-108	-586	-255	-43					
07/19/26:23:26	H	H	RUNNING	200	147.84	1864	H	3.91	-3
1016	-9	-755	41	15					
07/19/26:23:26	H	H	RUNNING	200	147.84	1864	H	3.91	-15
976	-15	399	9	14					
07/19/26:23:26	H	H	RUNNING	200	189	1879	H	7.47	-13
963	-47	325	12	30					
07/19/26:23:26	H	H	RUNNING	200	189	1879	H	7.47	-7
953	-111	827	39	-28					
07/19/26:23:26	H	H	RUNNING	200	189	1879	H	7.47	-17
1031	-51	134	9	17					
07/19/26:23:26	H	H	RUNNING	200	189	1879	H	7.47	-15
1051	-53	565	66	7					
07/19/26:23:26	H	H	RUNNING	200	189	1879	H	7.47	-13
911	-285	399	61	-57					
07/19/26:23:26	H	H	RUNNING	200	189	1879	H	7.47	-37
1010	-184	11	-96	-2					
07/19/26:23:26	H	H	RUNNING	200	189	1879	H	7.47	-1
995	-128	-226	6	-1					
07/19/26:23:26	H	H	RUNNING	200	209.77	1894	H	5.68	-8
998	-133	13	16	10					
07/19/26:23:26	M	H	RUNNING	200	209.77	1894	H	5.68	-31
1018	-108	374	-189	58					
07/19/26:23:26	M	H	RUNNING	200	209.77	1894	H	5.68	10
990	-147	691	-2	-8					
07/19/26:23:26	M	H	RUNNING	200	209.77	1894	H	5.68	-26
1020	-140	-151	14	-4					
07/19/26:23:26	M	H	RUNNING	200	209.77	1894	H	5.68	-9
978	-190	-89	3	-4					
07/19/26:23:26	H	H	RUNNING	200	209.77	1894	H	5.68	-13
993	-159	-52	-6	1					
07/19/26:23:26	H	H	RUNNING	200	209.77	1894	H	5.68	-3
1006	-148	36	-19	-1					

07/19/26:23:26 H H RUNNING 200 204.91 1900 H 8.51 -23
999 -149 -7 5 2
07/19/26:23:26 H H RUNNING 200 204.91 1900 H 8.51 -13
1003 -141 152 -4 0
07/19/26:23:26 H H RUNNING 200 204.91 1900 H 8.51 -2
997 -146 7 7 -2
07/19/26:23:26 H H RUNNING 200 204.91 1900 H 8.51 -9
983 -181 -87 -65 44
07/19/26:23:26 H H RUNNING 200 204.91 1900 H 8.51 -4
998 -158 37 22 -24
07/19/26:23:26 H H RUNNING 200 204.91 1900 H 8.51 -8
997 -161 21 34 -7
07/19/26:23:26 H H RUNNING 200 204.91 1900 H 8.51 2
992 -143 -23 -18 9
07/19/26:23:26 H H RUNNING 200 203.31 1900 H 7.36 -4
985 -175 42 139 -11
07/19/26:23:26 H H RUNNING 200 203.31 1900 H 7.36 0
989 -38 -598 85 -2
07/19/26:23:26 H H RUNNING 200 203.31 1900 H 7.36 -12
1024 -126 -257 -2 -34
07/19/26:23:26 H H RUNNING 200 203.31 1900 H 7.36 -9
980 -180 -18 -7 -22
07/19/26:23:26 H H RUNNING 200 203.31 1900 H 7.36 -6
991 -154 -14 -23 -48
07/19/26:23:26 H H RUNNING 200 203.31 1900 H 7.36 -16
984 -163 17 55 -12
07/19/26:23:26 H H RUNNING 200 203.31 1900 H 7.36 -1
993 -142 -72 3 -3
07/19/26:23:26 H H RUNNING 200 203.31 1900 H 7.36 -4
994 -148 -4 8 -1
07/19/26:23:26 H H RUNNING 200 204.09 1900 H 4.76 -11
993 -149 12 5 -5
07/19/26:23:26 H H RUNNING 200 204.09 1900 H 4.76 -9
1002 -125 -23 -1 -2
07/19/26:23:26 H H RUNNING 200 204.09 1900 H 4.76 -11
998 -139 15 7 -2
07/19/26:23:26 H H RUNNING 200 204.09 1900 H 4.76 -7
995 -143 -62 2 -1
07/19/26:23:26 H H RUNNING 200 204.09 1900 H 4.76 -10
991 -144 4 -1 -1
07/19/26:23:26 H H RUNNING 200 204.09 1900 H 4.76 -10
988 -147 12 3 -1

07/19/26:23:26	H	H	RUNNING		200	204.09	1900	H	4.76	-8
982	-155	15	1	-1						
07/19/26:23:26	H	H	RUNNING		200	190.7	1900	H	7.3	-7
996	-137	-18	1	-1						
07/19/26:23:26	H	H	RUNNING		200	190.7	1900	H	7.3	-12
989	-146	-1	3	-2						
07/19/26:23:26	H	H	RUNNING		200	190.7	1900	H	7.3	-10
990	-142	-29	3	-2						
07/19/26:23:26	H	H	RUNNING		200	190.7	1900	H	7.3	-10
995	-139	-13	0	0						
07/19/26:23:26	H	H	RUNNING		200	190.7	1900	H	7.3	1
1007	-115	9	11	0						
07/19/26:23:26	H	H	RUNNING		200	190.7	1900	H	7.3	-2
984	-148	29	-1	-2						
07/19/26:23:26	H	H	RUNNING		200	190.7	1900	H	7.3	-13
996	-135	39	-1	0						
07/19/26:23:26	H	H	RUNNING		200	191.97	1900	H	8.9	-9
995	-136	16	0	0						
07/19/26:23:26	H	H	RUNNING		200	191.97	1900	H	8.9	-12
1000	-126	51	1	-2						
07/19/26:23:26	H	H	RUNNING		200	191.97	1900	H	8.9	-11
993	-143	-1	2	-2						
07/19/26:23:26	H	H	RUNNING		200	191.97	1900	H	8.9	-9
989	-147	10	3	-1						
07/19/26:23:26	H	H	RUNNING		200	191.97	1900	H	8.9	-11
992	-144	-17	0	-3						
07/19/26:23:26	H	H	RUNNING		200	191.97	1900	H	8.9	-6
996	-138	-15	1	-2						
07/19/26:23:26	-	H	RUNNING		200	191.97	1900	H	8.9	-9
997	-133	8	1	-1						
07/19/26:23:26	-	H	RUNNING		200	183.04	1900	H	7.75	-9
1000	-129	14	2	-2						
07/19/26:23:26	-	H	RUNNING		200	183.04	1900	H	7.75	-9
995	-139	-17	3	-2						
07/19/26:23:26	-	H	RUNNING		200	183.04	1900	H	7.75	-10
997	-137	-8	5	-1						
07/19/26:23:26	-	H	RUNNING		200	183.04	1900	H	7.75	-11
992	-139	-26	5	0						
07/19/26:23:26	-	H	RUNNING		200	183.04	1900	H	7.75	-10
997	-133	-29	2	-2						
07/19/26:23:26	-	H	RUNNING		200	183.04	1900	H	7.75	-11
997	-130	-29	4	-2						

07/19/26:23:26	H	H	RUNNING		200	183.04	1900	H	7.75	-10
992	-139	-16	0	-1						
07/19/26:23:26	H	H	RUNNING		200	196.22	1900	H	6.37	-13
998	-128	14	8	0						
07/19/26:23:26	H	H	RUNNING		200	196.22	1900	H	6.37	2
987	-141	0	3	0						
07/19/26:23:26	H	H	RUNNING		200	196.22	1900	H	6.37	-3
992	-132	-12	5	-1						
07/19/26:23:26	H	H	RUNNING		200	196.22	1900	H	6.37	-15
1009	-118	25	2	-3						
07/19/26:23:26	H	H	RUNNING		200	196.22	1900	H	6.37	-12
1004	-123	-42	-10	0						
07/19/26:23:26	H	H	RUNNING		200	196.22	1900	H	6.37	-18
992	-145	51	-4	-3						
07/19/26:23:26	H	H	RUNNING		200	196.22	1900	H	6.37	-9
999	-123	18	8	-1						
07/19/26:23:26	H	H	RUNNING		200	201.77	1900	H	7.56	-6
1009	-106	-33	1	0						
07/19/26:23:26	H	H	RUNNING		200	201.77	1900	H	7.56	-2
992	-132	51	1	-3						
07/19/26:23:26	H	H	RUNNING		200	201.77	1900	H	7.56	-14
998	-129	9	-5	-5						
07/19/26:23:26	H	H	RUNNING		200	201.77	1900	H	7.56	-14
993	-113	-78	-59	3						
07/19/26:23:26	H	H	RUNNING		200	201.77	1900	H	7.56	-2
983	-106	-168	-128	-12						
07/19/26:23:26	H	H	RUNNING		200	201.77	1900	H	7.56	-11
1006	-39	-325	-4	16						
07/19/26:23:26	H	H	RUNNING		200	201.77	1900	H	7.56	-13
1021	-35	622	-115	39						
07/19/26:23:26	H	H	RUNNING		200	175.88	1900	H	8.47	-17
1032	-89	1161	303	-53						
07/19/26:23:26	H	H	RUNNING		200	175.88	1900	H	8.47	-28
977	-294	929	908	-305						
07/19/26:23:26	H	H	RUNNING		200	175.88	1900	H	8.47	-4
886	-454	1951	-52	-36						
07/19/26:23:26	H	H	RUNNING		200	175.88	1900	H	8.47	1
728	-700	720	-166	168						
07/19/26:23:26	H	H	RUNNING		200	175.88	1900	H	8.47	-7
688	-739	248	52	-161						
07/19/26:23:26	H	H	RUNNING		200	175.88	1900	H	8.47	-37
596	-823	1055	70	-119						

07/19/26:23:26	H	H	RUNNING	200	175.88	1900	H	8.47	-35
543	-860	215	106	456					
07/19/26:23:26	H	H	RUNNING	200	175.88	1900	H	8.47	-16
675	-767	-1171	215	245					
07/19/26:23:26	-	H	RUNNING	200	204.17	1900	H	8.24	22
710	-724	-1030	-260	71					
07/19/26:23:26	-	H	RUNNING	200	204.17	1900	H	8.24	76
842	-573	-765	-12	-51					
07/19/26:23:26	H	H	RUNNING	200	204.17	1900	H	8.24	73
899	-510	516	-316	150					
07/19/26:23:26	H	H	RUNNING	200	204.17	1900	H	8.24	51
826	-644	731	-49	227					
07/19/26:23:26	H	H	RUNNING	200	204.17	1900	H	8.24	40
837	-541	-721	51	-56					
07/19/26:23:26	H	H	RUNNING	200	204.17	1900	H	8.24	-3
863	-517	-1305	-44	-54					
07/19/26:23:26	H	H	RUNNING	200	204.17	1900	H	8.24	-4
1043	-105	-1585	36	26					
07/19/26:23:26	H	H	RUNNING	200	193.41	1889	H	6.56	-1
937	-291	1109	27	23					
07/19/26:23:26	H	H	RUNNING	200	193.41	1874	H	6.56	-7
932	-361	956	-5	-14					
07/19/26:23:26	H	H	RUNNING	200	193.41	1869	H	6.56	2
914	-411	-453	-2	-7					
07/19/26:23:26	H	H	RUNNING	200	193.41	1869	H	6.56	-4
946	-362	-226	-17	-12					
07/19/26:23:26	H	H	RUNNING	200	193.41	1877	H	6.56	-4
978	-247	-402	15	11					
07/19/26:23:26	H	H	RUNNING	200	193.41	1877	H	6.56	-10
981	-249	500	14	2					
07/19/26:23:26	H	H	RUNNING	200	193.41	1877	H	6.56	2
936	-328	112	-11	-13					
07/19/26:23:26	H	H	RUNNING	200	246.87	1883	H	5.83	-5
982	-243	-395	3	5					
07/19/26:23:26	H	H	RUNNING	200	246.87	1900	H	5.83	-4
987	-225	214	9	12					
07/19/26:23:26	H	H	RUNNING	200	246.87	1900	H	5.83	-4
967	-273	506	2	-6					
07/19/26:23:26	H	H	RUNNING	200	246.87	1900	H	5.83	-7
943	-330	-165	0	-5					
07/19/26:23:26	H	H	RUNNING	200	246.87	1888	H	5.83	-4
994	-241	-101	6	7					

07/19/26:23:26	H	H	RUNNING		200	246.87	1888	H	5.83	-7
979	-265	324	-5	-8						
07/19/26:23:26	H	H	RUNNING		200	246.87	1887	H	5.83	-3
946	-323	-176	3	-7						
07/19/26:23:26	H	H	RUNNING		200	241.37	1886	H	6.23	8
981	-261	-151	13	7						
07/19/26:23:26	H	H	RUNNING		200	241.37	1861	H	6.23	-10
949	-320	257	-7	-9						
07/19/26:23:26	H	H	RUNNING		200	241.37	1861	H	6.23	-2
954	-314	-313	0	-12						
07/19/26:23:26	H	H	RUNNING		200	241.37	1861	H	6.23	-2
987	-243	-307	8	11						
07/19/26:23:26	H	H	RUNNING		200	241.37	1861	H	6.23	-6
969	-276	475	9	0						
07/19/26:23:26	H	H	RUNNING		200	241.37	1852	H	6.23	-2
940	-328	48	-6	-17						
07/19/26:23:26	H	H	RUNNING		200	241.37	1852	H	6.23	-6
965	-271	-532	4	6						
07/19/26:23:26	-	H	RUNNING		200	141.81	1852	H	7.5	-6
977	-250	300	11	9						
07/19/26:23:26	-	H	RUNNING		200	141.81	1852	H	7.5	-2
954	-298	345	-4	-13						
07/19/26:23:26	-	H	RUNNING		200	141.81	1867	H	7.5	-3
963	-284	-337	-2	0						
07/19/26:23:26	-	H	RUNNING		200	141.81	1867	H	7.5	-7
987	-243	18	3	-7						
07/19/26:23:26	H	H	RUNNING		200	141.81	1871	H	7.5	-7
962	-270	-183	-21	-50						
07/19/26:23:26	H	H	RUNNING		200	141.81	1875	H	7.5	-6
972	-188	-1001	-16	-30						
07/19/26:23:26	H	H	RUNNING		200	141.81	1875	H	7.5	-12
1001	-54	-666	0	-9						
07/19/26:23:26	H	H	RUNNING		200	143.24	1877	H	6.7	-14
1014	10	-180	0	-9						
07/19/26:23:26	H	H	RUNNING		200	143.24	1877	H	6.7	-17
1016	50	-69	3	2						
07/19/26:23:26	H	H	RUNNING		200	143.24	1879	H	6.7	-17
999	39	-7	1	17						
07/19/26:23:26	H	H	RUNNING		200	143.24	1879	H	6.7	-13
1003	43	-36	15	18						
07/19/26:23:26	H	H	RUNNING		200	143.24	1879	H	6.7	-5
996	-132	743	19	14						

07/19/26:23:26	H	H	RUNNING		200	143.24	1890	H	6.7	-4
982	-203	584	15	14						
07/19/26:23:26	-	H	RUNNING		200	143.24	1891	H	6.7	-2
944	-321	609	9	8						
07/19/26:23:26	-	H	RUNNING		200	159.16	1891	H	7.39	0
945	-351	386	13	-2						
07/19/26:23:26	-	H	RUNNING		200	159.16	1891	H	7.39	0
911	-418	-5	0	-10						
07/19/26:23:26	-	H	RUNNING		200	159.16	1892	H	7.39	-3
931	-378	-303	-7	-34						
07/19/26:23:26	-	H	RUNNING		200	159.16	1892	H	7.39	-1
941	-303	-885	-14	-17						
07/19/26:23:26	-	H	RUNNING		200	159.16	1890	H	7.39	-7
1023	-165	-424	9	5						
07/19/26:23:26	-	H	RUNNING		200	159.16	1890	H	7.39	-10
975	-225	179	-10	-9						
07/19/26:23:26	-	H	RUNNING		200	159.16	1891	H	7.39	-8
970	-229	-230	-10	-12						
07/19/26:23:26	-	H	RUNNING		200	136.02	1900	H	6.07	-10
986	-163	-351	-2	-15						
07/19/26:23:26	-	H	RUNNING		200	136.02	1900	H	6.07	-8
992	-126	-461	-5	-22						
07/19/26:23:26	-	H	RUNNING		200	136.02	1900	H	6.07	-11
1005	-45	-397	11	18						
07/19/26:23:26	-	H	RUNNING		200	136.02	1900	H	6.07	-14
1016	-10	353	16	24						
07/19/26:23:26	-	H	RUNNING		200	136.02	1900	H	6.07	-13
1002	-82	751	18	26						
07/19/26:23:26	-	H	RUNNING		200	136.02	1900	H	6.07	-6
992	-191	1019	29	23						
07/19/26:23:26	-	H	RUNNING		200	136.02	1900	H	6.07	-4
926	-343	962	0	2						
07/19/26:23:26	-	H	RUNNING		200	173.74	1900	H	5.65	-1
933	-383	170	1	0						
07/19/26:23:26	-	H	RUNNING		200	173.74	1900	H	5.65	0
926	-394	43	6	-11						
07/19/26:23:26	H	H	RUNNING		200	173.74	1900	H	5.65	1
909	-415	-304	-26	-32						
07/19/26:23:26	H	H	RUNNING		200	173.74	1900	H	5.65	1
945	-312	-1142	-28	-49						
07/19/26:23:26	H	H	RUNNING		200	173.74	1900	H	5.65	-10
989	-156	-1522	8	42						

07/19/26:23:26	H	H	RUNNING		200	173.74	1900	H	5.65	-23
1047	178	-134	17	1						
07/19/26:23:26	H	H	RUNNING		200	173.74	1900	H	5.65	-16
1004	35	-25	22	36						
07/19/26:23:26	H	H	RUNNING		200	160.52	1900	H	7.22	-18
1001	-21	947	22	32						
07/19/26:23:26	H	H	RUNNING		200	160.52	1900	H	7.22	-8
991	-108	1014	5	11						
07/19/26:23:26	-	H	RUNNING		200	160.52	1900	H	7.22	-6
995	-196	473	13	3						
07/19/26:23:26	-	H	RUNNING		200	160.52	1900	H	7.22	-6
946	-288	228	49	-8						
07/19/26:23:26	-	H	RUNNING		200	160.52	1900	H	7.22	1
870	-472	371	-2	-11						
07/19/26:23:26	-	H	RUNNING		200	160.52	1900	H	7.22	1
907	-420	-303	-46	-82						
07/19/26:23:26	-	H	RUNNING		200	160.52	1900	H	7.22	-3
940	-203	-2779	40	-26						
07/19/26:23:26	-	H	RUNNING		200	197.76	1900	H	7.87	1
1052	38	217	10	6						
07/19/26:23:26	-	H	RUNNING		200	197.76	1900	H	7.87	-12
1023	33	233	29	70						
07/19/26:23:26	-	H	RUNNING		200	197.76	1877	H	7.87	-14
992	-97	2285	-91	90						
07/19/26:23:26	-	H	RUNNING		200	197.76	1877	H	7.87	7
812	-515	782	129	-20						
07/19/26:23:26	-	H	RUNNING		200	197.76	1877	H	7.87	7
910	-379	-1462	-28	-62						
07/19/26:23:26	-	H	RUNNING		200	197.76	1877	H	7.87	-11
984	-117	-1707	-45	-25						
07/19/26:23:26	-	H	RUNNING		200	197.76	1877	H	7.87	-20
1000	54	37	15	19						
07/19/26:23:26	-	H	RUNNING		200	197.76	1877	H	7.87	-20
1048	79	344	16	18						
07/19/26:23:26	-	H	RUNNING		200	160.98	1900	H	6.85	-10
986	-90	563	6	0						
07/19/26:23:27	H	H	RUNNING		200	160.98	1900	H	6.85	-5
983	-139	122	-2	-8						
07/19/26:23:27	H	H	RUNNING		200	160.98	1900	H	6.85	-8
996	-122	-178	0	-4						
07/19/26:23:27	H	H	RUNNING		200	160.98	1900	H	6.85	-9
993	-118	-53	0	1						

07/19/26:23:27	H	H	RUNNING		200	160.98	1900	H	6.85	-9
997	-116	89	0	-2						
07/19/26:23:27	H	H	RUNNING		200	160.98	1900	H	6.85	-8
1005	-109	4	3	0						
07/19/26:23:27	H	H	RUNNING		200	160.98	1900	H	6.85	-9
992	-131	24	0	-3						
07/19/26:23:27	H	H	RUNNING		200	196.82	1821	H	4.7	-7
1000	-123	-23	4	1						
07/19/26:23:27	H	H	RUNNING		200	196.82	1821	H	4.7	-7
1007	-122	114	11	2						
07/19/26:23:27	H	H	RUNNING		200	196.82	1835	H	4.7	-7
985	-178	137	0	0						
07/19/26:23:27	H	H	RUNNING		200	196.82	1835	H	4.7	-11
1002	-159	111	11	2						
07/19/26:23:27	H	H	RUNNING		200	196.82	1835	H	4.7	-6
969	-226	118	-2	2						
07/19/26:23:27	H	H	RUNNING		200	196.82	1835	H	4.7	-5
1001	-193	162	5	3						
07/19/26:23:27	H	H	RUNNING		200	196.82	1835	H	4.7	-5
946	-338	202	3	-12						
07/19/26:23:27	H	H	RUNNING		200	194.51	1866	H	5.76	0
937	-340	-340	-8	-14						
07/19/26:23:27	H	H	RUNNING		200	194.51	1866	H	5.76	-5
981	-266	-437	1	0						
07/19/26:23:27	H	H	RUNNING		200	194.51	1866	H	5.76	-7
994	-232	91	23	5						
07/19/26:23:27	H	H	RUNNING		200	194.51	1866	H	5.76	-3
943	-326	259	-4	-7						
07/19/26:23:27	H	H	RUNNING		200	194.51	1866	H	5.76	-4
962	-300	-188	1	0						
07/19/26:23:27	H	H	RUNNING		200	194.51	1866	H	5.76	-6
974	-280	88	2	0						
07/19/26:23:27	H	H	RUNNING		200	194.51	1866	H	5.76	-2
955	-311	43	0	0						
07/19/26:23:27	H	H	RUNNING		200	194.51	1866	H	5.76	-7
958	-316	67	10	0						
07/19/26:23:27	H	H	RUNNING		200	138.88	1900	H	5.59	-3
935	-352	64	4	-3						
07/19/26:23:27	H	H	RUNNING		200	138.88	1900	H	5.59	-4
948	-327	-67	-5	-4						
07/19/26:23:27	H	H	RUNNING		200	138.88	1900	H	5.59	-5
966	-300	-83	-2	-2						

07/19/26:23:27	H	H	RUNNING	200	138.88	1900	H	5.59	-9
959	-307	-22	3	0					
07/19/26:23:27	H	H	RUNNING	200	138.88	1900	H	5.59	-3
959	-310	58	0	1					
07/19/26:23:27	H	H	RUNNING	200	138.88	1900	H	5.59	0
964	-311	45	5	-2					
07/19/26:23:27	H	H	RUNNING	200	138.88	1900	H	5.59	0
944	-338	-35	2	0					
07/19/26:23:27	H	H	RUNNING	200	120.38	1900	H	7.81	-3
955	-320	37	0	-4					
07/19/26:23:27	H	H	RUNNING	200	120.38	1900	H	7.81	-3
962	-308	-49	-1	-2					
07/19/26:23:27	H	H	RUNNING	200	120.38	1900	H	7.81	-5
956	-314	-27	0	-1					
07/19/26:23:27	H	H	RUNNING	200	120.38	1900	H	7.81	-4
961	-306	0	6	-5					
07/19/26:23:27	H	H	RUNNING	200	120.38	1900	H	7.81	0
955	-314	-82	4	0					
07/19/26:23:27	H	H	RUNNING	200	120.38	1900	H	7.81	-3
953	-314	42	4	0					
07/19/26:23:27	H	H	RUNNING	200	120.38	1900	H	7.81	-2
959	-313	10	3	-3					
07/19/26:23:27	H	H	RUNNING	200	119.41	1900	H	5.38	0
955	-314	-11	3	-1					
07/19/26:23:27	H	H	RUNNING	200	119.41	1900	H	5.38	-3
955	-318	7	0	-2					
07/19/26:23:27	H	H	RUNNING	200	119.41	1900	H	5.38	-3
958	-309	-28	5	-3					
07/19/26:23:27	H	H	RUNNING	200	119.41	1900	H	5.38	-1
957	-311	-10	1	-2					
07/19/26:23:27	-	H	RUNNING	200	119.41	1900	H	5.38	-2
959	-309	-41	0	-2					
07/19/26:23:27	-	H	RUNNING	200	119.41	1900	H	5.38	-7
961	-309	-31	-3	0					
07/19/26:23:27	-	H	RUNNING	200	119.41	1900	H	5.38	-9
956	-315	73	0	-2					
07/19/26:23:27	-	H	RUNNING	200	132.5	1900	H	8.92	-11
963	-308	-7	-2	0					
07/19/26:23:27	-	H	RUNNING	200	132.5	1900	H	8.92	-9
953	-323	49	4	0					
07/19/26:23:27	H	H	RUNNING	200	132.5	1900	H	8.92	-5
956	-315	38	3	-1					

07/19/26:23:27	H	H	RUNNING	200	132.5	1900	H	8.92	-2	
955	-320	-2	2	-2						
07/19/26:23:27	H	H	RUNNING	200	132.5	1900	H	8.92	0	
952	-322	-15	4	-3						
07/19/26:23:27	H	H	RUNNING	200	132.5	1900	H	8.92	0	
955	-314	-11	0	-3						
07/19/26:23:27	H	H	RUNNING	200	132.5	1900	H	8.92	-4	
958	-316	-26	-1	-2						
07/19/26:23:27	H	H	RUNNING	200	134.86	1900	H	6.28	-4	
959	-310	-21	4	-2						
07/19/26:23:27	H	H	RUNNING	200	134.86	1900	H	6.28	-4	
954	-318	-8	1	-1						
07/19/26:23:27	H	H	RUNNING	200	134.86	1900	H	6.28	-4	
954	-323	27	1	0						
07/19/26:23:27	-	H	RUNNING	200	134.86	1900	H	6.28	0	
955	-321	59	7	0						
07/19/26:23:27	-	H	RUNNING	200	134.86	1900	H	6.28	-5	
951	-325	26	1	-1						
07/19/26:23:27	-	H	RUNNING	200	134.86	1900	H	6.28	-2	
959	-313	1	0	-3						
07/19/26:23:27	-	H	RUNNING	200	150.88	1900	H	6.82	-5	
953	-320	-21	1	-1						
07/19/26:23:27	-	H	RUNNING	200	150.88	1900	H	6.82	0	
954	-315	-3	2	-3						
07/19/26:23:27	-	H	RUNNING	200	150.88	1900	H	6.82	-1	
955	-320	-64	-3	-2						
07/19/26:23:27	M	H	RUNNING	200	150.88	1900	H	6.82	-8	
955	-321	-14	4	-2						
07/19/26:23:27	M	H	RUNNING	200	150.88	1900	H	6.82	-5	
954	-325	9	3	-1						
07/19/26:23:27	H	H	RUNNING	200	150.88	1900	H	6.82	-4	
949	-333	30	2	-2						
07/19/26:23:27	H	H	RUNNING	200	150.88	1900	H	6.82	-1	
959	-317	22	-2	-4						
07/19/26:23:27	H	H	RUNNING	200	150.88	1900	H	6.82	2	
870	-506	-63	-1	-1						
07/19/26:23:27	H	H	RUNNING	200	173.27	1900	H	4.84	1	
879	-499	-14	0	-3						
07/19/26:23:27	H	H	RUNNING	200	173.27	1900	H	4.84	2	
880	-495	-54	1	-1						
07/19/26:23:27	H	H	RUNNING	200	173.27	1900	H	4.84	2	
876	-496	-34	0	-2						

07/19/26:23:27	H	H	RUNNING		200	173.27	1900	H	4.84	1
892	-478	-39	1	-3						
07/19/26:23:27	H	H	RUNNING		200	173.27	1900	H	4.84	0
878	-497	-33	3	-2						
07/19/26:23:27	H	H	RUNNING		200	173.27	1900	H	4.84	2
883	-488	-8	0	-3						
07/19/26:23:27	H	H	RUNNING		200	202.13	1900	H	4.51	-1
879	-492	0	6	-2						
07/19/26:23:27	H	H	RUNNING		200	202.13	1900	H	4.51	2
877	-494	7	9	-2						
07/19/26:23:27	H	H	RUNNING		200	202.13	1900	H	4.51	1
889	-480	-17	4	-2						
07/19/26:23:27	H	H	RUNNING		200	202.13	1900	H	4.51	0
890	-477	-23	4	-1						
07/19/26:23:27	H	H	RUNNING		200	202.13	1900	H	4.51	1
884	-485	-5	3	-2						
07/19/26:23:27	H	H	RUNNING		200	202.13	1900	H	4.51	0
887	-481	-11	2	-2						
07/19/26:23:27	H	H	RUNNING		200	202.13	1900	H	4.51	3
883	-486	0	5	-2						
07/19/26:23:27	H	H	RUNNING		200	178.18	1889	H	6.92	4
883	-485	-6	3	-2						
07/19/26:23:27	H	H	RUNNING		200	178.18	1894	H	6.92	2
879	-490	-25	2	-2						
07/19/26:23:27	H	H	RUNNING		200	178.18	1894	H	6.92	1
885	-481	-33	8	-3						
07/19/26:23:27	H	H	RUNNING		200	178.18	1896	H	6.92	2
879	-490	-37	-3	-2						
07/19/26:23:27	H	H	RUNNING		200	178.18	1900	H	6.92	1
891	-475	-33	-1	-4						
07/19/26:23:27	H	H	RUNNING		200	178.18	1900	H	6.92	0
893	-468	-33	0	-2						
07/19/26:23:27	H	H	RUNNING		200	178.18	1900	H	6.92	1
894	-472	-14	6	-4						
07/19/26:23:27	H	H	RUNNING		200	178.18	1900	H	6.92	0
882	-484	-29	0	-1						
07/19/26:23:27	H	H	RUNNING		200	181.38	1900	H	7.18	2
886	-479	21	-1	-2						
07/19/26:23:27	H	H	RUNNING		200	181.38	1900	H	7.18	2
890	-473	-11	0	-1						
07/19/26:23:27	H	H	RUNNING		200	181.38	1900	H	7.18	0
885	-483	2	5	0						

07/19/26:23:27	H	H	RUNNING		200	181.38	1900	H	7.18	1
885	-483	47	1	-1						
07/19/26:23:27	H	H	RUNNING		200	181.38	1900	H	7.18	0
886	-481	9	4	-1						
07/19/26:23:27	H	H	RUNNING		200	181.38	1900	H	7.18	2
891	-478	-3	5	-1						
07/19/26:23:27	H	H	RUNNING		200	181.38	1900	H	7.18	-1
886	-481	-18	-2	-2						
07/19/26:23:27	H	H	RUNNING		200	185.06	1900	H	7.55	7
894	-470	9	0	-2						
07/19/26:23:27	H	H	RUNNING		200	185.06	1900	H	7.55	1
893	-466	-43	5	-3						
07/19/26:23:27	H	H	RUNNING		200	185.06	1900	H	7.55	1
883	-480	-15	0	-4						
07/19/26:23:27	H	H	RUNNING		200	185.06	1900	H	7.55	0
890	-472	-35	0	-1						
07/19/26:23:27	H	H	RUNNING		200	185.06	1900	H	7.55	1
890	-473	6	0	-2						
07/19/26:23:27	H	H	RUNNING		200	185.06	1900	H	7.55	0
892	-474	6	-2	-2						
07/19/26:23:27	H	H	RUNNING		200	185.06	1900	H	7.55	5
891	-469	-37	0	-4						
07/19/26:23:27	H	H	RUNNING		200	213.54	1877	H	5.2	3
895	-469	-78	6	-3						
07/19/26:23:27	H	H	RUNNING		200	213.54	1877	H	5.2	5
891	-475	-31	-1	-1						
07/19/26:23:27	H	H	RUNNING		200	213.54	1877	H	5.2	0
891	-473	-12	0	-2						
07/19/26:23:27	H	H	RUNNING		200	213.54	1879	H	5.2	0
893	-475	0	0	0						
07/19/26:23:27	H	H	RUNNING		200	213.54	1879	H	5.2	3
893	-471	-1	6	-2						
07/19/26:23:27	H	H	RUNNING		200	213.54	1879	H	5.2	0
887	-474	-29	1	-1						
07/19/26:23:27	H	H	RUNNING		200	213.54	1877	H	5.2	0
890	-473	30	0	0						
07/19/26:23:27	H	H	RUNNING		200	186.11	1891	H	6.43	1
890	-477	9	3	0						
07/19/26:23:27	H	H	RUNNING		200	186.11	1891	H	6.43	2
887	-478	2	9	-1						
07/19/26:23:27	H	H	RUNNING		200	186.11	1891	H	6.43	0
888	-480	37	1	0						

07/19/26:23:27	L	H	RUNNING		200	186.11	1891	H	6.43	-2
885	-481	31	0	-1						
07/19/26:23:27	L	H	RUNNING		200	186.11	1891	H	6.43	2
892	-475	39	-3	0						
07/19/26:23:27	L	H	RUNNING		200	186.11	1900	H	6.43	-3
888	-480	88	-229	157						
07/19/26:23:27	L	H	RUNNING		200	186.11	1900	H	6.43	-19
811	-561	756	-4	6						
07/19/26:23:27	L	H	RUNNING		200	186.11	1900	H	6.43	0
860	-531	214	-29	28						
07/19/26:23:27	L	H	RUNNING		200	186.18	1900	H	5.84	4
855	-543	191	-54	49						
07/19/26:23:27	L	H	RUNNING		200	186.18	1900	H	5.84	-2
832	-567	357	-16	7						
07/19/26:23:27	L	H	RUNNING		200	186.18	1900	H	5.84	9
808	-601	150	0	0						
07/19/26:23:27	L	H	RUNNING		200	186.18	1900	H	5.84	2
803	-611	41	4	-2						
07/19/26:23:27	H	H	RUNNING		200	186.18	1900	H	5.84	10
810	-603	3	11	0						
07/19/26:23:27	L	H	RUNNING		200	186.18	1900	H	5.84	10
815	-594	-43	4	-1						
07/19/26:23:27	L	H	RUNNING		200	186.18	1900	H	5.84	1
807	-609	-14	-1	1						
07/19/26:23:27	H	H	RUNNING		200	206.45	1900	H	6.02	5
820	-589	24	2	0						
07/19/26:23:27	H	H	RUNNING		200	206.45	1900	H	6.02	7
819	-592	16	-3	0						
07/19/26:23:27	H	H	RUNNING		200	206.45	1900	H	6.02	4
818	-590	2	4	-1						
07/19/26:23:27	H	H	RUNNING		200	206.45	1900	H	6.02	4
817	-588	7	2	-2						
07/19/26:23:27	H	H	RUNNING		200	206.45	1891	H	6.02	5
819	-591	3	9	-1						
07/19/26:23:27	H	H	RUNNING		200	206.45	1891	H	6.02	3
814	-597	-8	1	-1						
07/19/26:23:27	H	H	RUNNING		200	206.45	1891	H	6.02	4
819	-589	-7	-1	-1						
07/19/26:23:27	H	H	RUNNING		200	208.26	1789	H	6.81	4
820	-590	4	5	-1						
07/19/26:23:27	H	H	RUNNING		200	208.26	1789	H	6.81	3
820	-588	-17	-6	0						

07/19/26:23:27	H	H	RUNNING		200	208.26	1789	H	6.81	6
821	-583	-19	0	-1						
07/19/26:23:27	H	H	RUNNING		200	208.26	1789	H	6.81	3
816	-596	-3	3	-2						
07/19/26:23:27	H	H	RUNNING		200	208.26	1789	H	6.81	2
818	-590	5	1	-2						
07/19/26:23:27	M	H	RUNNING		200	208.26	1789	H	6.81	7
821	-587	1	3	-2						
07/19/26:23:27	M	H	RUNNING		200	208.26	1789	H	6.81	2
816	-596	-4	-2	-1						
07/19/26:23:27	M	H	RUNNING		200	210.17	1770	H	7.94	7
816	-592	14	-1	0						
07/19/26:23:27	M	H	RUNNING		200	210.17	1770	H	7.94	5
823	-583	11	-6	1						
07/19/26:23:27	M	H	RUNNING		200	210.17	1770	H	7.94	2
823	-588	13	4	-5						
07/19/26:23:27	M	H	RUNNING		200	210.17	1759	H	7.94	1
827	-582	11	12	-9						
07/19/26:23:27	M	H	RUNNING		200	210.17	1759	H	7.94	1
814	-594	-20	0	-3						
07/19/26:23:27	M	H	RUNNING		200	210.17	1759	H	7.94	3
822	-585	8	2	-1						
07/19/26:23:27	-	H	RUNNING		200	210.17	1759	H	7.94	7
817	-593	-2	5	-1						
07/19/26:23:27	H	H	RUNNING		200	141.4	1900	H	7.54	7
816	-592	-5	3	-1						
07/19/26:23:27	H	H	RUNNING		200	141.4	1900	H	7.54	5
820	-588	2	-3	2						
07/19/26:23:27	H	H	RUNNING		200	141.4	1900	H	7.54	3
817	-597	13	2	2						
07/19/26:23:27	H	H	RUNNING		200	141.4	1900	H	7.54	7
814	-596	-4	6	0						
07/19/26:23:27	H	H	RUNNING		200	141.4	1900	H	7.54	6
816	-596	-11	4	-1						
07/19/26:23:27	H	H	RUNNING		200	141.4	1900	H	7.54	3
813	-597	-2	0	-1						
07/19/26:23:27	H	H	RUNNING		200	141.4	1900	H	7.54	1
814	-593	-8	4	-1						
07/19/26:23:27	H	H	RUNNING		200	153.92	1900	H	7.94	6
820	-588	-22	-71	27						
07/19/26:23:27	-	H	RUNNING		200	153.92	1900	H	7.94	4
790	-628	214	-23	-1						

07/19/26:23:27	-	H	RUNNING		200	153.92	1900	H	7.94	0
776	-646	132	18	-13						
07/19/26:23:27	-	H	RUNNING		200	153.92	1900	H	7.94	1
784	-639	-22	-7	-2						
07/19/26:23:27	-	H	RUNNING		200	153.92	1900	H	7.94	-1
779	-640	33	1	-1						
07/19/26:23:27	-	H	RUNNING		200	153.92	1900	H	7.94	15
780	-639	25	-22	11						
07/19/26:23:27	-	H	RUNNING		200	153.92	1900	H	7.94	-4
780	-641	85	-87	27						
07/19/26:23:27	-	H	RUNNING		200	153.92	1900	H	7.94	-1
775	-647	322	-40	18						
07/19/26:23:27	-	H	RUNNING		200	182.88	1900	H	7.21	-10
752	-672	75	-24	3						
07/19/26:23:27	-	H	RUNNING		200	182.88	1900	H	7.21	-16
750	-680	64	1	2						
07/19/26:23:27	-	H	RUNNING		200	182.88	1900	H	7.21	-3
741	-685	1	-21	2						
07/19/26:23:27	-	H	RUNNING		200	182.88	1900	H	7.21	-7
739	-689	90	4	-2						
07/19/26:23:27	-	H	RUNNING		200	182.88	1900	H	7.21	-7
740	-686	-13	6	-4						
07/19/26:23:27	-	H	RUNNING		200	182.88	1900	H	7.21	-2
750	-678	-19	0	0						
07/19/26:23:27	-	H	RUNNING		200	183.75	1900	H	7.13	-5
753	-671	-9	-1	-3						
07/19/26:23:27	-	H	RUNNING		200	183.75	1900	H	7.13	-9
743	-683	13	2	-4						
07/19/26:23:27	H	H	RUNNING		200	183.75	1900	H	7.13	-5
751	-674	10	-3	-5						
07/19/26:23:27	H	H	RUNNING		200	183.75	1900	H	7.13	-6
750	-675	33	1	-3						
07/19/26:23:27	H	H	RUNNING		200	183.75	1900	H	7.13	-2
745	-683	26	3	-1						
07/19/26:23:27	L	H	RUNNING		200	183.75	1900	H	7.13	-4
740	-687	0	9	0						
07/19/26:23:27	L	H	RUNNING		200	183.75	1900	H	7.13	-4
738	-688	-15	0	0						
07/19/26:23:27	L	H	RUNNING		200	147.75	1900	H	5.31	-8
738	-693	-5	6	0						
07/19/26:23:27	L	H	RUNNING		200	147.75	1900	H	5.31	-4
730	-698	-10	3	-2						

07/19/26:23:27	L	H	RUNNING		200	147.75	1900	H	5.31	-6
737	-688	-9	1	0						
07/19/26:23:27	L	H	RUNNING		200	147.75	1900	H	5.31	0
747	-681	-7	13	8						
07/19/26:23:27	L	H	RUNNING		200	147.75	1900	H	5.31	-2
740	-690	-47	-12	21						
07/19/26:23:27	-	H	RUNNING		200	147.75	1900	H	5.31	-13
742	-689	-29	-25	24						
07/19/26:23:27	-	H	RUNNING		200	147.75	1900	H	5.31	6
753	-672	-55	-32	54						
07/19/26:23:27	-	H	RUNNING		200	173.17	1900	H	5.08	-40
766	-654	-206	0	-9						
07/19/26:23:27	-	H	RUNNING		200	173.17	1900	H	5.08	-1
751	-673	83	17	-7						
07/19/26:23:27	-	H	RUNNING		200	173.17	1900	H	5.08	-15
756	-673	-33	43	-25						
07/19/26:23:27	-	H	RUNNING		200	173.17	1900	H	5.08	0
772	-654	-113	26	-2						
07/19/26:23:27	-	H	RUNNING		200	173.17	1900	H	5.08	-9
759	-668	-149	3	2						
07/19/26:23:27	-	H	RUNNING		200	173.17	1900	H	5.08	1
774	-646	-71	0	-3						
07/19/26:23:27	-	H	RUNNING		200	159.25	1900	H	6.02	0
764	-654	6	-3	-2						
07/19/26:23:27	-	H	RUNNING		200	159.25	1900	H	6.02	0
769	-659	63	4	-13						
07/19/26:23:27	-	H	RUNNING		200	159.25	1900	H	6.02	-3
764	-659	148	-109	48						
07/19/26:23:27	-	H	RUNNING		200	159.25	1900	H	6.02	-17
704	-727	535	499	-513						
07/19/26:23:27	-	H	RUNNING		200	159.25	1900	H	6.02	0
847	-574	169	-31	-43						
07/19/26:23:27	-	H	RUNNING		200	159.25	1900	H	6.02	17
668	-715	184	3	-4						
07/19/26:23:27	-	H	RUNNING		200	153.01	1900	H	8.05	-15
688	-735	7	3	-5						
07/19/26:23:27	-	H	RUNNING		200	153.01	1900	H	8.05	-7
684	-742	0	0	-3						
07/19/26:23:27	-	H	RUNNING		200	153.01	1900	H	8.05	-22
675	-750	0	1	-1						
07/19/26:23:27	-	H	RUNNING		200	153.01	1900	H	8.05	-16
684	-743	-1	-1	0						

07/19/26:23:27	L	H	RUNNING	200	153.01	1900	H	8.05	-8
685	-744	0	0	0					
07/19/26:23:27	L	H	RUNNING	200	153.01	1900	H	8.05	-12
684	-750	0	2	-1					
07/19/26:23:27	L	H	RUNNING	200	153.01	1900	H	8.05	-36
689	-737	-2	0	0					
07/19/26:23:27	L	H	RUNNING	200	177.74	1900	H	6.58	-7
684	-746	-1	1	-3					
07/19/26:23:27	L	H	RUNNING	200	177.74	1900	H	6.58	-14
697	-727	0	3	-3					
07/19/26:23:27	L	H	RUNNING	200	177.74	1900	H	6.58	-15
679	-753	-3	4	-3					
07/19/26:23:27	L	H	RUNNING	200	177.74	1900	H	6.58	-16
681	-744	-3	1	-2					
07/19/26:23:27	-	H	RUNNING	200	177.74	1900	H	6.58	-19
685	-743	-2	2	-1					
07/19/26:23:27	-	H	RUNNING	200	177.74	1900	H	6.58	-15
684	-744	-3	1	-1					
07/19/26:23:27	-	H	RUNNING	200	154.56	1900	H	7.82	-15
684	-744	-1	1	-2					
07/19/26:23:27	-	H	RUNNING	200	154.56	1900	H	7.82	-15
684	-742	-2	1	-1					
07/19/26:23:27	-	H	RUNNING	200	154.56	1900	H	7.82	-17
686	-741	-1	1	-2					
07/19/26:23:27	-	H	RUNNING	200	154.56	1900	H	7.82	-15
684	-744	-1	2	-3					
07/19/26:23:27	H	H	RUNNING	200	154.56	1900	H	7.82	-13
685	-742	-1	1	-2					
07/19/26:23:27	H	H	RUNNING	200	154.56	1900	H	7.82	-23
686	-742	-1	2	-2					
07/19/26:23:27	H	H	RUNNING	200	143.7	1900	H	6.31	-21
683	-743	-2	2	-2					
07/19/26:23:27	H	H	RUNNING	200	143.7	1900	H	6.31	-13
683	-743	-2	3	-1					
07/19/26:23:27	H	H	RUNNING	200	143.7	1900	H	6.31	-18
684	-744	-1	2	-2					
07/19/26:23:27	H	H	RUNNING	200	143.7	1900	H	6.31	-16
685	-744	-2	3	-2					
07/19/26:23:27	H	H	RUNNING	200	143.7	1900	H	6.31	-17
685	-740	-2	1	-2					
07/19/26:23:27	-	H	RUNNING	200	143.7	1900	H	6.31	-15
685	-741	-2	2	-2					

07/19/26:23:27	-	H	RUNNING	200	143.7	1900	H	6.31	-15	
683	-745	-2	2	-2						
07/19/26:23:27	-	H	RUNNING	200	72.55	1900	H	6.15	-16	
684	-746	-2	2	-1						
07/19/26:23:27	-	H	RUNNING	200	72.55	1900	H	6.15	-15	
682	-743	-1	2	-2						
07/19/26:23:27	-	H	RUNNING	200	72.55	1900	H	6.15	-15	
685	-743	-2	3	-3						
07/19/26:23:27	-	H	RUNNING	200	72.55	1900	H	6.15	-17	
686	-743	-1	2	-1						
07/19/26:23:27	-	H	RUNNING	200	72.55	1900	H	6.15	-16	
685	-739	-2	1	-2						
07/19/26:23:27	-	H	RUNNING	200	72.55	1900	H	6.15	-18	
682	-746	-3	1	-1						
07/19/26:23:27	-	H	RUNNING	200	72.55	1900	H	6.15	-11	
680	-745	-2	1	-2						
07/19/26:23:27	-	H	RUNNING	200	113.1	1900	H	6.7	-18	
683	-745	-1	2	-2						
07/19/26:23:27	-	H	RUNNING	200	113.1	1900	H	6.7	-16	
684	-741	-2	2	-2						
07/19/26:23:27	-	H	RUNNING	200	113.1	1900	H	6.7	-16	
685	-744	-2	0	-1						
07/19/26:23:27	-	H	RUNNING	200	113.1	1900	H	6.7	-14	
682	-744	-2	2	0						
07/19/26:23:27	-	H	RUNNING	200	113.1	1900	H	6.7	-17	
684	-742	-3	1	-1						
07/19/26:23:27	-	H	RUNNING	200	113.1	1900	H	6.7	-16	
683	-745	-2	1	-1						
07/19/26:23:27	-	H	RUNNING	200	153.41	1900	H	7.25	-19	
685	-738	-2	2	-1						
07/19/26:23:27	-	H	RUNNING	200	153.41	1900	H	7.25	-19	
683	-746	-2	0	-1						
07/19/26:23:27	-	H	RUNNING	200	153.41	1900	H	7.25	-15	
684	-745	-2	1	-2						
07/19/26:23:27	-	H	RUNNING	200	153.41	1900	H	7.25	-13	
682	-743	-2	1	-1						
07/19/26:23:27	-	H	RUNNING	200	153.41	1900	H	7.25	-18	
683	-744	-2	2	-1						
07/19/26:23:27	-	H	RUNNING	200	153.41	1900	H	7.25	-10	
686	-742	-2	2	-3						
07/19/26:23:27	-	H	RUNNING	200	177.94	1900	H	6.93	-16	
682	-744	-2	2	-2						

07/19/26:23:27	-	H	RUNNING		200	177.94	1900	H	6.93	-13
680	-745	-2	2	-2						
07/19/26:23:27	-	H	RUNNING		200	177.94	1900	H	6.93	-13
682	-744	-2	2	-1						
07/19/26:23:27	M	H	RUNNING		200	177.94	1900	H	6.93	-17
683	-743	-2	2	-2						
07/19/26:23:27	M	H	RUNNING		200	177.94	1900	H	6.93	-16
683	-743	-2	2	-1						
07/19/26:23:27	M	H	RUNNING		200	177.94	1900	H	6.93	-16
683	-745	-2	3	-2						
07/19/26:23:27	M	H	RUNNING		200	177.94	1900	H	6.93	-16
681	-744	-1	2	-2						
07/19/26:23:27	M	H	RUNNING		200	195.22	1900	H	6.72	-16
683	-746	-2	2	-3						
07/19/26:23:27	M	H	RUNNING		200	195.22	1900	H	6.72	-14
683	-743	-3	2	-2						
07/19/26:23:27	M	H	RUNNING		200	195.22	1881	H	6.72	-19
685	-741	-3	2	-2						
07/19/26:23:27	-	H	RUNNING		200	195.22	1881	H	6.72	-16
683	-744	-3	2	-2						
07/19/26:23:27	-	H	RUNNING		200	195.22	1881	H	6.72	-21
682	-744	-2	0	0						
07/19/26:23:27	-	H	RUNNING		200	195.22	1883	H	6.72	-26
680	-746	-1	2	-1						
07/19/26:23:27	L	H	RUNNING		200	195.22	1883	H	6.72	-14
684	-741	-2	8	-1						
07/19/26:23:27	H	H	RUNNING		200	157.97	1883	H	6.83	-18
681	-750	-8	12	51						
07/19/26:23:27	H	H	RUNNING		200	157.97	1900	H	6.83	-5
636	-788	-787	190	27						
07/19/26:23:27	H	H	RUNNING		200	157.97	1900	H	6.83	-20
745	-668	-1285	248	-165						
07/19/26:23:27	H	H	RUNNING		200	157.97	1900	H	6.83	31
853	-464	-1489	-28	-42						
07/19/26:23:27	H	H	RUNNING		200	157.97	1900	H	6.83	-5
923	-376	-1543	27	-33						
07/19/26:23:27	H	H	RUNNING		200	157.97	1900	H	6.83	-1
972	-233	-921	70	-58						
07/19/26:23:27	H	H	RUNNING		200	189.91	1900	H	6.54	-8
1032	-81	-763	7	-44						
07/19/26:23:27	H	H	RUNNING		200	189.91	1900	H	6.54	-9
995	-101	-187	3	-13						

07/19/26:23:27	H	H	RUNNING		200	189.91	1900	H	6.54	-14
1011	-68	-74	0	-56						
07/19/26:23:27	H	H	RUNNING		200	189.91	1900	H	6.54	-18
996	-74	-197	1	0						
07/19/26:23:27	H	H	RUNNING		200	189.91	1900	H	6.54	-23
1006	-42	-6	5	-78						
07/19/26:23:27	H	H	RUNNING		200	189.91	1900	H	6.54	-26
989	-48	-291	3	-116						
07/19/26:23:27	H	H	RUNNING		200	194.01	1821	H	6.52	-37
996	-1	-457	1	-5						
07/19/26:23:27	H	H	RUNNING		200	194.01	1821	H	6.52	-48
1001	41	1	2	-1						
07/19/26:23:27	H	H	RUNNING		200	194.01	1821	H	6.52	-45
1000	44	-5	1	-1						
07/19/26:23:27	H	H	RUNNING		200	194.01	1821	H	6.52	-43
998	44	-4	2	0						
07/19/26:23:27	H	H	RUNNING		200	194.01	1821	H	6.52	-45
1000	45	-5	0	-2						
07/19/26:23:27	H	H	RUNNING		200	194.01	1821	H	6.52	-48
1000	40	-2								