TUGAS AKHIR – EE234899

DESAIN DAN IMPLEMENTASI FUZZY LOGIC CONTROLLER UNTUK KONTROL MANUVER YAW DAN PITCH PADA AUTONOMOUS UNDERWATER VEHICLE

ILHAM WAHYU EKO PRASETYO

NRP 5022201041

Dosen Pembimbing

Dr. Ir. Ari Santoso, DEA

NIP 19660218 199102 1001

Eka Iskandar, S.T., M.T.

NIP 19800528 200812 1001

Program Studi Teknik Sistem Kontrol

Departemen Teknik Elektro

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya

2025

TUGAS AKHIR – EE234899

DESAIN DAN IMPLEMENTASI FUZZY LOGIC CONTROLLER UNTUK KONTROL MANUVER YAW DAN PITCH PADA AUTONOMOUS UNDERWATER VEHICLE

ILHAM WAHYU EKO PRASETYO

NRP 5022201041

Dosen Pembimbing

Dr. Ir. Ari Santoso, DEA

NIP 19660218 199102 1001

Eka Iskandar, S.T., M.T.

NIP 19800528 200812 1001

Program Studi Teknik Sistem Kontrol

Departemen Teknik Elektro

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya

2025

(Halaman Ini Sengaja Dikosongkan)

FINAL PROJECT - EE234899

DESIGN AND IMPLEMENTATION OF FUZZY LOGIC CONTROLLER FOR YAW AND PITCH MANEUVER CONTROL IN AUTONOMOUS UNDERWATER VEHICLE

ILHAM WAHYU EKO PRASETYO

NRP 5022201041

Advisor

Dr. Ir. Ari Santoso, DEA

NIP 19660218 199102 1001

Eka Iskandar, S.T., M.T.

NIP 19800528 200812 1001

Study Program Control System Engineering

Department of Electrical Engineering

Faculty of Electrical Technology and Intelligent Informatics

Institut Teknologi Sepuluh Nopember

Surabaya

2025

(Halaman Ini Sengaja Dikosongkan)

LEMBAR PENGESAHAN

DESAIN DAN IMPLEMENTASI FUZZY LOGIC CONTROLLER UNTUK KONTROL MANUVER YAW DAN PITCH PADA AUTONOMOUS UNDERWATER VEHICLE

TUGAS AKHIR

Diajukan untuk memenuhi salah satu syarat

memperoleh gelar Sarjana Teknik pada

Program Studi S-1 Teknik Sistem Kontrol

Departemen Teknik Elektro

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Oleh : **ILHAM WAHYU EKO PRASETYO**

NRP. 5022201041

Disetujui oleh Tim Penguji Tugas Akhir :

1.	Dr. Ir. Ari Santoso, DEA	Pembimbing
2.	Eka Iskandar, S.T., M.T.	Ko-pembimbing
3.	Imam Arifin, S.T., M.T.	Penguji
4.	Ir. Joko Susila, M.T.	Penguji
5.	Yurid Eka Nugraha, S.T., M.Eng., Ph.D.	Penguji

SURABAYA

Januari , 2025

(Halaman Ini Sengaja Dikosongkan)

PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini:

Nama mahasiswa / NRP	:	Ilham Wahyu Eko Prasetyo
Program studi	:	Teknik Sistem Kontrol
Dosen Pembimbing / NIP	:	Dr. Ir. Ari Santoso, DEA / 19660218 199102 1001

dengan ini menyatakan bahwa Tugas Akhir dengan judul "DESAIN DAN IMPLEMENTASI FUZZY LOGIC CONTROLLER UNTUK KONTROL MANUVER YAW DAN PITCH PADA AUTONOMOUS UNDERWATER VEHICLE" adalah hasil karya sendiri, bersifat orisinal, dan ditulis dengan mengikuti kaidah penulisan ilmiah.

Bilamana di kemudian hari ditemukan ketidaksesuaian dengan pernyataan ini, maka saya bersedia menerima sanksi sesuai dengan ketentuan yang berlaku di Institut Teknologi Sepuluh Nopember.

Surabaya, 23 Januari 2025

Mengetahui
Dosen Pembimbing

Dr. Ir. Ari Santoso, DEA
19660218 199102 1001

Mengetahui
Dosen Ko-Pembimbing

Eka Iskandar, S.T., M.T.
19800528 200812 1001

Mengetahui
Mahasiswa

Ilham Wahyu Eko Prasetyo
5022201041

(Halaman Ini Sengaja Dikosongkan)

ABSTRAK

DESAIN DAN IMPLEMENTASI FUZZY LOGIC CONTROLLER UNTUK KONTROL MANUVER YAW DAN PITCH PADA AUTONOMOUS UNDERWATER VEHICLE

Nama Mahasiswa / NRP	**:**	**Ilham Wahyu Eko Prasetyo / 5022201041**
Departemen	**:**	**Teknik Elektro FTEIC - ITS**
Dosen Pembimbing	**:**	**Dr. Ir. Ari Santoso, DEA**

Abstrak

Perkembangan teknologi Autonomous Underwater Vehicle (AUV) telah mengalami kemajuan signifikan, khususnya dalam penelitian di Indonesia untuk survei bawah laut, pengukuran lingkungan, keamanan, dan eksplorasi perairan. AUV berperan sebagai kendaraan tanpa awak yang mampu melakukan navigasi secara mandiri di lingkungan bawah air. Namun, tantangan utama yang dihadapi dalam operasional AUV adalah dinamika lingkungan bawah air yang kompleks, termasuk arus tidak terprediksi, perubahan suhu, serta turbulensi air yang dapat memengaruhi kestabilan dan performa AUV. Penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem kontrol berbasis Fuzzy Logic Controller (FLC) guna meningkatkan kestabilan dan akurasi manuver AUV pada sumbu yaw dan pitch. Metodologi yang digunakan meliputi perancangan model kinematika dan dinamika AUV, desain kontroler fuzzy, simulasi sistem, serta pengujian eksperimen berbasis mikrokontroler Arduino Mega 2560 dan sensor IMU BNO055. Motor DC digunakan sebagai propulsi utama, sementara motor stepper dengan driver TB6600 berfungsi mengontrol sudut kemudi rudder dan stern. Hasil eksperimen menunjukkan bahwa sistem kontrol Fuzzy Logic Controller mampu memberikan respons yang lebih adaptif terhadap variasi setpoint sudut dibandingkan metode kontrol konvensional. Pengujian pada beberapa skenario manuver menunjukkan performa sistem yang stabil, dengan error steady-state yang rendah serta waktu respon yang lebih cepat dalam mencapai setpoint. Kesimpulan dari penelitian ini menegaskan bahwa pendekatan berbasis FLC efektif dalam meningkatkan kestabilan dan manuverabilitas AUV dalam lingkungan yang dinamis. Penelitian ini diharapkan dapat berkontribusi pada pengembangan teknologi maritim di Indonesia, terutama dalam bidang survei bawah laut, pemetaan perairan, serta eksplorasi sumber daya kelautan. Inovasi dalam sistem kendali AUV perlu terus dikembangkan untuk mendukung efektivitas dan efisiensi operasi di lingkungan bawah laut yang semakin kompleks.

Kata kunci: Autonomous Underwater Vehicle, Fuzzy Logic Controller, Hidrodinamika, Kendali Manuver, Kontrol Cerdas

.

(Halaman Ini Sengaja Dikosongkan)

ABSTRACT

Design And Implementation of Fuzzy Logic Controller for Yaw And Pitch Maneuver Control in Autonomous Underwater Vehicle

Student Name / NRP	**:**	**Ilham Wahyu Eko Prasetyo/ 5022201041**
Department	**:**	**Teknik Elektro FTEIC - ITS**
Advisor	**:**	**Dr. Ir. Ari Santoso, DEA**

Abstract

The development of Autonomous Underwater Vehicle (AUV) technology has made significant progress, particularly in research in Indonesia for underwater surveys, environmental measurements, security, and water exploration. AUV serves as an unmanned vehicle capable of autonomous navigation in underwater environments. However, the main challenge in AUV operations lies in the complex dynamics of underwater environments, including unpredictable currents, temperature changes, and water turbulence, which can affect AUV stability and performance. This research aims to design and implement a Fuzzy Logic Controller (FLC)-based control system to improve the stability and maneuvering accuracy of AUV in the yaw and pitch axes. The methodology includes the design of AUV kinematic and dynamic models, fuzzy controller design, system simulation, and experimental testing based on the Arduino Mega 2560 microcontroller and IMU BNO055 sensor. A DC motor is used as the main propulsion, while a stepper motor with a TB6600 driver functions to control the rudder and stern steering angles. Experimental results show that the Fuzzy Logic Controller control system provides a more adaptive response to variations in angle setpoints compared to conventional control methods. Testing across various maneuvering scenarios demonstrates stable system performance, with low steady-state error and faster response time in reaching the setpoint. The findings of this study confirm that the FLC-based approach is effective in enhancing AUV stability and maneuverability in dynamic environments. This research is expected to contribute to the development of maritime technology in Indonesia, particularly in underwater surveys, water mapping, and marine resource exploration. Innovation in AUV control systems must continue to be developed to support operational effectiveness and efficiency in increasingly complex underwater environments.

Keywords: Autonomous Underwater Vehicle, Fuzzy Logic Controller, Maneuvering Control, Hydrodynamics, Intelligent Control.

(Halaman Ini Sengaja Dikosongkan)

KATA PENGANTAR

Puji syukur kehadirat Allah SWT, atas rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul "Desain dan Implementasi *Fuzzy Logic Controller* untuk Kontrol Manuver *Yaw* dan *Pitch* pada *Autonomous Underwater Vehicle*" dengan baik dan tepat waktu. Penyusunan tugas akhir ini merupakan salah satu syarat untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Sistem Kontrol, Departemen Teknik Elektro, Fakultas Teknologi Elektro dan Informatika Cerdas (FTEIC), Institut Teknologi Sepuluh Nopember (ITS). Penelitian ini dilakukan dengan harapan dapat memberikan kontribusi dalam pengembangan sistem kendali untuk Autonomous Underwater Vehicle (AUV), khususnya dalam meningkatkan stabilitas dan akurasi manuver. Selama proses penyusunan tugas akhir ini, penulis menghadapi berbagai tantangan, namun dengan dukungan dan bantuan dari banyak pihak, tugas akhir ini akhirnya dapat terselesaikan. Oleh karena itu, penulis ingin mengucapkan terima kasih kepada berbagai pihak yang telah memberikan bimbingan, motivasi, serta dukungan baik secara langsung maupun tidak langsung.

Penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Allah SWT, atas segala nikmat, kemudahan, dan kelancaran yang diberikan dalam menyelesaikan tugas akhir ini.
2. Orang tua tercinta, yang selalu memberikan doa, dukungan, motivasi, serta kasih sayang tanpa batas dalam setiap langkah yang penulis tempuh.
3. Dosen Pembimbing, Dr. Ir. Ari Santoso, DEA dan Eka Iskandar, S.T., M.T., yang dengan penuh kesabaran memberikan arahan, masukan, serta motivasi selama proses penelitian ini.
4. Seluruh dosen di Departemen Teknik Elektro, ITS, yang telah memberikan ilmu dan bimbingan selama masa perkuliahan.
5. Tim penguji tugas akhir, yang telah memberikan kritik dan saran yang membangun demi kesempurnaan penelitian ini.
6. Ucapan terima kasih juga saya tujukan kepada pasangan saya, Lavida Aisya, atas doa, semangat, dan kesabaran yang diberikan selama saya menyelesaikan tugas akhir ini. Dukungan yang diberikan sungguh berarti dan membantu saya dalam menjalani proses penelitian ini.
7. Rekan-rekan mahasiswa Teknik Sistem Kontrol, yang telah menjadi tempat berbagi ilmu, pengalaman, serta dukungan selama proses penyusunan tugas akhir ini.
8. Teman-teman seperjuangan, yang selalu memberikan semangat, motivasi, serta kebersamaan yang luar biasa.
9. Seluruh pihak yang tidak dapat disebutkan satu per satu, yang telah membantu dalam bentuk apa pun dalam menyelesaikan tugas akhir ini.

Semoga segala bantuan dan dukungan yang diberikan mendapatkan balasan kebaikan dari Allah SWT. Penulis menyadari bahwa tugas akhir ini masih jauh dari kata sempurna, oleh karena itu, kritik dan saran yang membangun sangat diharapkan untuk penyempurnaan lebih lanjut.

Surabaya, 23 Januari 2025

Ilham Wahyu Eko Prasetyo

(Halaman Ini Sengaja Dikosongkan)

DAFTAR ISI

LEMBAR PENGESAHAN ...... **Error! Bookmark not defined.**

PERNYATAAN ORISINALITAS iii

ABSTRAK v

ABSTRACT vii

KATA PENGANTAR ix

DAFTAR ISI xi

DAFTAR GAMBAR ...... **Error! Bookmark not defined.**

DAFTAR TABEL xv

DAFTAR SIMBOL xix

BAB 1 PENDAHULUAN 1

1.1 Latar Belakang 1

1.2 Rumusan Masalah 2

1.3 Batasan Masalah 2

1.4 Tujuan 2

1.5 Manfaat 2

BAB 2 TINJAUAN PUSTAKA 5

2.1 Hasil Penelitian Terdahulu 5

2.2 DASAR TEORI 6

2.2.1 Autonomous Underwater Vehicle 6

2.2.2 Model Kinematika AUV 6

2.2.3 Model Dinamika AUV 7

2.2.4 Model Hidrodinamika AUV 8

2.2.5 Hidrostatis 8

2.2.6 Redaman Hidrodinamika 8

2.2.7 Massa Tambahan 10

2.2.8 Gaya dan Momen Angkat Badan AUV 10

2.2.9 Gaya dan Momen Angkat Rudder AUV 10

2.2.10 Fuzzy Logic Controller 11

2.2.11 Sistem Mikrokontroller 12

2.2.12 Sistem Inertial Measurement Unit 13

2.2.13 Motor DC 15

2.2.14 Rangkaian kontrol PWM Motor DC 16

2.2.15 Motor Stepper 17

2.2.16 Driver Stepper TB6600 18
BAB 3 METODOLOGI 23
3.1 Metode Kontrol Fuzzy Logic Controller 23
3.1.1 Diagram Blok Perancangan Fuzzy Logic Controller 23
3.1.2 Perancangan Membership Function untuk Yaw Error 23
3.1.3 Proses Menggunakan Fuzzy dan Defuzzyfikasi 24
3.2 Bahan dan peralatan yang digunakan 24
3.3 Urutan pelaksanaan penelitian 25
3.3.1 Studi Literatur 26
3.3.2 Simulasi Sistem AUV 26
3.3.3 Desain Rangkaian Elektronik AUV 27
3.3.4 Perancangan Power Supply 28
3.3.5 Perancangan Power Supply untuk Mikrokontroller 28
3.3.6 Perakitan Power Supply Motor DC dan Motor Stepper 29
3.3.7 Perancangan Rangkaian Kontrol PWM Motor DC 29
3.3.8 Perancangan Integrasi Rangkaian Sensor dan Mikrokontroller 30
3.3.9 Perancangan Integrasi Driver Motor Stepper 31
3.3.10 Pengujian Sensor IMU 31
3.3.11 Pengujian Motor Stepper 32
3.3.12 Pengujian Motor DC 32
3.3.13 Perancangan dan Desain Mekanik 32
3.3.14 Integrasi Elektronik dan Mekanik 33
3.3.15 Pengujian Kestabilan *Body* 34
3.3.16 Pengujian Gerak dan Manuver 34
3.3.17 Implementasi Algoritma Fuzzy Logic Controller 34
3.3.18 Pengambilan Data Sensor 34
3.3.19 Evaluasi Kinerja 34
3.3.20 Hasil 34
3.3.21 Penulisan Laporan 35
BAB 4 HASIL DAN PEMBAHASAN 37
4.1 Hasil penelitian 37
4.1.1 Hasil Simulasi Pada Lintasan 37
4.1.2 Hasil Pengujian Manuver Yaw 5 Derajat 37
4.1.3 Analisis Data dan Perhitungan untuk manuver 5 derajat 39

4.1.4 Pembahasan Error Pengukuran dengan Perhitungan....................................42
4.1.5 Hasil Pengujian Manuver Yaw 10 Derajat...43
4.1.6 Analisis Data dan Perhitungan untuk Manuver yaw 10 Derajat 45
4.1.7 Pembahasan Error Pengukuran dengan Perhitungan....................................47
4.1.8 Hasil Pengujian Manuver Yaw -10 Derajat..48
4.1.9 Analisis Data dan Perhitungan untuk Manuver Yaw -10 Derajat ....**Error! Bookmark not defined.**
4.1.10 Pembahasan Error Pengukuran dengan Perhitungan....................................51
4.1.11 Hasil Pengujian Manuver Yaw 20 Derajat...52
4.1.12 Analisis Data dan Perhitungan untuk Manuver Yaw 20 Derajat..................54
4.1.13 Pembahasan Error Pengukuran dengan Perhitungan....................................55
4.1.14 Hasil Pengujian Manuver Yaw (-20) Derajat...56
4.1.15 Analisis Data dan Perhitungan untuk Manuver Yaw (-20) Derajat..............58
4.1.16 Pembahasan Error Pengukuran dengan Perhitungan....................................59
4.1.17 Hasil Pengujian Manuver Yaw 30 Derajat...60
4.1.18 Analisis Data dan Perhitungan untuk Manuver Yaw 30 Derajat..................62
4.1.19 Pembahasan Error Pengukuran dengan Perhitungan....................................63
4.1.20 Hasil Pengujian Manuver Yaw 5 Derajat Menuju (-5) Derajat....................64
4.1.21 Analisis Data dan Perhitungan untuk Manuver Yaw 5 Derajat Menuju (-5) Derajat 66
4.1.22 Pembahasan Error Pengukuran dengan Perhitungan....................................68
4.1.23 Hasil Pengujian Manuver Pitch 10 Derajat 69
4.1.24 Analisis Data dan Perhitungan untuk Manuver Pitch 10 Derajat.................70
4.1.25 Pembahasan Error Pengukuran dengan Perhitungan....................................72
BAB 5 KESIMPULAN DAN SARAN...75
5.1 Kesimpulan...75
5.2 Saran..76
DAFTAR PUSTAKA 77
LAMPIRAN..79
BIODATA PENULIS 92

(Halaman Ini Sengaja Dikosongkan)

DAFTAR GAMBAR

Gambar 2. 1 Sistem Gerak 6-DOF AUV 6
Gambar 2. 2 Diagram blok Fuzzy Logic Controller 11
Gambar 2. 3 Mikrokontroller Arduino Mega 2560 13
Gambar 2. 4 Sensor Inertial Measurement Unit 14
Gambar 2. 5 Motor DC 500 Watt 15
Gambar 2. 6 Rangkaian Kontrol PWM 16
Gambar 2. 7 Motor Stepper NEMA17 17
Gambar 2. 8 Konstruksi Motor Stepper dari (www.jwardtech.com) 17
Gambar 2. 9 Driver Motor Stepper TB6600 18
Gambar 2. 10 Pengaturan Avometer untuk cek Polaritas Motor Stepper 19
Gambar 2. 11 Tampilan Avometer ketika Kabel Stepper Motor Tidak Sepasang 19
Gambar 2. 12 Tampilan Avometer ketika Kabel Stepper Motor Sepasang 20
Gambar 3. 1 Diagram Blok Sistem Kontrol AUV 23
Gambar 3. 2 Visualisasi Fungsi Keanggotaan Fuzzy 24
Gambar 3. 3 Diagram Blok Urutan Pelaksanaan Penelitian AUV 26
Gambar 3. 5 Diagram Blok Hardware AUV 27
Gambar 3. 6 Desain Rangkaian Elektronik AUV 28
Gambar 3. 7 Diagram blok Power Supply ke Seluruh Beban 28
Gambar 3. 8 Diagram blok Power Supply Mikrokontroller 28
Gambar 3. 9 Rangkaian Power Supply untuk Mikrokontroller 29
Gambar 3. 10 Diagram blok Power Supply Motor DC dan Motor Stepper 29
Gambar 3. 11 Rangkaian Power Supply Motor DC dan Motor Stepper 29
Gambar 3. 12 Diagram blok Rangkaian Kontrol PWM Motor DC 30
Gambar 3. 13 Rangkaian Kontrol PWM Motor DC 30
Gambar 3. 14 Diagram blok Integrasi Sensor dan Mikrokontroller 30
Gambar 3. 15 Rangkaian Integrasi Sensor dan Mikrokontroller 31
Gambar 3. 16 Diagram blok Integrasi Driver TB6600 ke Motor Stepper 31
Gambar 3. 17 Rangkaian Integrasi Empat Driver Stepper 31
Gambar 3. 18 Pengujian sensor IMU dengan Processing 4 32
Gambar 3. 19 Pengujian Driver dan Motor Stepper 32
Gambar 3. 20 Pengujian Motor DC untuk Propeller 32
Gambar 3. 21 Rancang Bangun Mekanik 33
Gambar 3. 22 Diagram Blok Integrasi Elektronik dan Mekanik AUV 33
Gambar 3. 23 Integrasi Rangkaian Elektronik dan Mekanik 33
Gambar 3. 24 Pengujian Kestabilan Body AUV 34
Gambar 4. 1 Hasil Simulasi AUV pada Lintasan 10° 37
Gambar 4. 2 Grafik Respon Sudut Yaw, Roll, Pitch, Stepper pada Setpoint 5° 37
Gambar 4. 3 Grafik Respon Sudut Yaw, Roll, Pitch, Stepper pada Setpoint 10° 43
Gambar 4. 4 Grafik Respon Sudut Yaw, Roll, Pitch, Stepper pada Setpoint -10° 48
Gambar 4. 5 Grafik Respon Sudut Yaw, Roll, Pitch, Stepper pada Setpoint 20° 52
Gambar 4. 6 Grafik Respon Sudut Yaw, Roll, Pitch, Stepper pada Setpoint -20° 56
Gambar 4. 7 Grafik Respon Sudut Yaw, Roll, Pitch, Stepper pada Setpoint 30° 60
Gambar 4. 8 Grafik Respon Sudut Yaw, Roll, Pitch, Stepper pada Setpoint 5° ke (-5)° 65
Gambar 4. 9 Grafik Respon Sudut Pitch, Roll, Yaw, Stepper pada Setpoint 10° 69

(Halaman Ini Sengaja Dikosongkan)

DAFTAR TABEL

Tabel 2.1 Penelitian AUV Terdahulu 5
Tabel 2.2 Spesifikasi Teknis Arduino Mega 2560 13
Tabel 2.3 Pengaturan Microstepping 21
Tabel 3. 1 Fungsi Keanggotaan Fuzzy Input 23
Tabel 4. 1 Metrik Kinerja Sistem Kontrol AUV pada Setpoint Yaw 5° 39
Tabel 4. 2 Sampel Data Hasil Pengukuran untuk Setpoint Yaw 5° 40
Tabel 4. 3 Fungsi Keanggotaan Fuzzy 40
Tabel 4. 4 Metrik Kinerja Sistem Kontrol AUV pada Setpoint Yaw 10° 45
Tabel 4. 5 Sampel Data Hasil Pengukuran untuk Setpoint Yaw 10° 46
Tabel 4. 6 Metrik Kinerja Sistem Kontrol AUV pada Setpoint Yaw -10° 49
Tabel 4. 7 Sampel Data Hasil Pengukuran untuk Setpoint Yaw -10° 50
Tabel 4. 8 Metrik Kinerja Sistem Kontrol AUV pada Setpoint Yaw 20° 53
Tabel 4. 9 Sampel Data Hasil Pengukuran untuk Setpoint Yaw 20° 54
Tabel 4. 10 Metrik Kinerja Sistem Kontrol AUV pada Setpoint Yaw -20° 57
Tabel 4. 11 Sampel Data Hasil Pengukuran untuk Setpoint Yaw -20° 58
Tabel 4. 12 Metrik Kinerja Sistem Kontrol AUV pada Setpoint Yaw 30° 61
Tabel 4. 13 Sampel Data Hasil Pengukuran untuk Setpoint Yaw 30° 62
Tabel 4. 14 Metrik Kinerja Sistem Kontrol AUV pada Setpoint Yaw 5° ke -5° 66
Tabel 4. 15 Sampel Data Hasil Pengukuran untuk Setpoint Yaw 5° ke -5° 67
Tabel 4. 16 Metrik Kinerja Sistem Kontrol AUV pada Setpoint Pitch 10° 70
Tabel 4. 17 Sampel Data Hasil Pengukuran untuk Setpoint Pitch 10° 71

(Halaman Ini Sengaja Dikosongkan)

DAFTAR SIMBOL

η_1	:	Posisi AUV terhadap *earth fixed frame*.
η_2	:	Orientasi AUV terhadap *earth fixed frame*.
x	:	Posisi atau tranlasi AUV pada sumbu longitudinal terhadap *earth fixed frame*.
y	:	Posisi atau tranlasi AUV pada sumbu lateral terhadap *earth fixed frame*.
z	:	Posisi atau tranlasi AUV pada sumbu vertikal terhadap *earth fixed frame*.
ϕ	:	Sudut rotasi *roll* AUV pada sumbu longitudinal terhadap *eath fixed frame*.
θ	:	Sudut rotasi *pitch* AUV pada sumbu lateral terhadap *earth fixed frame*.
ψ	:	Sudut rotasi *yaw* AUV pada sumbu vertikal terhadap *earth fixed frame*.
v_1	:	Kecepatan translasi AUV terhadap *body fixed frame*.
v_2	:	Kecepatan rotasi AUV terhadap *body fixed frame*.
u	:	Kecepatan linear AUV pada sumbu longitudinal terhadap *body fixed frame*.
v	:	Kecepatan linear AUV pada sumbu lateral terhadap *body fixed frame*.
w	:	Kecepatan linear AUV pada sumbu vertikal terhadap *body fixed frame*.
p	:	Kecepatan angular AUV pada sumbu longitudinal terhadap *body fixed frame*.
q	:	Kecepatan angular AUV pada sumbu lateral terhadap *body fixed frame*.
r	:	Kecepatan angular AUV pada sumbu vertikal terhadap *body fixed frame*.
τ_1	:	Gaya yang bekerja pada AUV
τ_2	:	Momen yang bekerja pada AUV
X	:	Gaya pada sumbu longitudinal AUV terhadap *body fixed frame*.
Y	:	Gaya pada sumbu lateral AUV terhadap *body fixed frame*.
Z	:	Gaya pada sumbu vertikal AUV terhadap *body fixed frame*.
K	:	Momen di sekitar sumbu longitudinal AUV terhadap *body fixed frame*.
M	:	Momen di sekitar sumbu lateral AUV terhadap *body fixed frame*.
N	:	Momen di sekitar sumbu vertikal AUV terhadap *body fixed frame*.
$\dot{\eta}_1$	:	Laju perubahan posisi AUV dengan waktu terhadap *earth fixed frame*
$\dot{\eta}_2$	:	Laju perubahan sudut AUV dengan waktu terhadap *earth fixed frame*.
$\dot{x}$	:	Laju perubahan posisi pada sumbu longitudinal dengan waktu terhadap *earth fixed frame*.
$\dot{y}$	:	Laju perubahan posisi pada sumbu lateral dengan waktu terhadap *earth fixed frame*.
$\dot{z}$	:	Laju perubahan posisi pada sumbu vertikal dengan waktu terhadap *earth fixed frame*.
$\dot{\phi}$	:	Laju perubahan sudut *roll* dengan waktu terhadap *earth fixed frame*.
$\dot{\theta}$	:	Laju perubahan sudut *pitch* dengan waktu terhadap *earth fixed frame*.

$\dot{\psi}$	:	Laju perubahan sudut yaw dengan waktu terhadap *earth fixed frame*.
m	:	Massa total dari AUV (Termasuk *ballast system*).
x_g	:	Pusat gravitasi AUV pada sumbu longitudinal.
y_g	:	Pusat gravitasi AUV pada sumbu lateral.
z_g	:	Pusat gravitasi AUV pada sumbu vertical.
I_x	:	Momen inersia pada sumbu longitudinal AUV terhadap *earth fixed frame*.
I_y	:	Momen inersia pada sumbu lateral AUV terhadap *earth fixed frame*.
I_z	:	Momen inersia pada sumbu vertikal AUV terhadap *earth fixed frame*.
W	;	Gaya berat dari AUV.
B	:	Gaya apung dari AUV.
ρ	:	Massa jenis cairan.
g	:	Gaya gravitasi pada AUV.
x_b	:	Pusat gaya apung pada sumbu longitudinal.
y_b	:	Pusat gaya apung pada sumbu lateral.
z_b	:	Pusat gaya apung pada sumbu vertical.
c_d	:	Koefisien hambatan aksial pada AUV
A_f	:	Luas area frontal dari AUV.
a		Percepatan linier
ω		Kecepatan sudut
V_{in}		Tegangan input modul
V_{out}		Tegangan rata-rata yang diterima motor
D		*Duty cycle*
θ_{step}		Sudut rotasi per langkah penuh (dalam derajat)
$\theta_{microstep}$		Sudut rotasi perlangkah dengan faktor microstepping tertentu
μ_{step}		Faktor *Microstepping*
N_{step}		Jumlah langkah penuh dalam satu putaran motor stepper
μ_M		*Fuzzy Set Medium*
μ_P		*Fuzzy Set Positive*
μ_Z		*Fuzzy Set Negative*
μ_i		Derajat keanggotaan tiap aturan *fuzzy*
O_i		Output Crisp yang terkait dengan aturan fuzzy tertentu
$\sum(\mu_i \cdot O_i)$		Penjumlahan dari hasil perkalian antara derajat keanggotaan dan output crisp

$\sum\mu_i$	Penjumlahan semua derajat keanggotaan
D_{rev}	Jumlah derajat per putaran penuh
D_{step}	Konfigurasi derajat per langkah dengan faktor mikrostepping

(Halaman Ini Sengaja Dikosongkan)

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi di era revolusi industri 4.0 telah mendorong kemajuan signifikan dalam berbagai bidang, termasuk teknologi robotika bawah air. Salah satu inovasi teknologi yang semakin berkembang pesat adalah Autonomous Underwater Vehicle (AUV), sebuah kendaraan tanpa awak yang mampu menyelam dan menjelajahi lingkungan bawah laut secara otomatis. AUV dirancang untuk menjalankan berbagai tugas di bawah air yang sulit dijangkau oleh manusia, seperti survei bawah laut, pengukuran lingkungan, keamanan, hingga eksplorasi perairan untuk edukasi. Teknologi ini tidak hanya menjadi alat yang efektif, tetapi juga menjadi solusi inovatif untuk berbagai tantangan yang dihadapi dalam eksplorasi bawah laut. Menurut (Yang et al., 2021) , AUV memberikan kemudahan bagi manusia untuk melakukan eksplorasi dan survei kondisi bawah laut tanpa memerlukan kendali langsung. Di Indonesia, kebutuhan akan teknologi AUV semakin meningkat seiring dengan tingginya potensi maritim yang dimiliki negara ini. Pengaplikasian AUV meliputi berbagai bidang, seperti misi militer, survei sumber daya alam, pemetaan topografi bawah laut, inspeksi struktur bawah air, hingga pemantauan lingkungan untuk tujuan ilmiah. Rizki dan Agustinah (2016) menyatakan bahwa AUV juga dapat digunakan untuk mendeteksi sumber minyak, melakukan inspeksi infrastruktur bawah laut, serta mendukung eksplorasi sumber daya kelautan. Dalam konteks eksplorasi laut dalam, AUV berfungsi sebagai alat utama untuk menjelajahi wilayah yang belum pernah diakses oleh manusia sebelumnya. (Rizki & Agustinah, 2016)

Namun demikian, AUV menghadapi tantangan besar terkait dinamika lingkungan bawah air. Hidrodinamika, atau ketidakpastian kondisi lingkungan bawah laut, menjadi salah satu permasalahan utama yang dihadapi dalam pengembangan AUV. Kondisi lingkungan air yang sangat dinamis, seperti arus yang tak terduga, perubahan suhu, tekanan, serta kekeruhan air, membuat sistem navigasi dan kendali AUV memerlukan desain yang lebih adaptif dan cerdas (Bellingham & Rajan, 2007). Hal ini menjadi tantangan besar bagi para peneliti dalam memastikan AUV dapat beroperasi secara stabil dan efisien di lingkungan yang penuh dengan ketidakpastian. Untuk mengatasi tantangan tersebut, diperlukan sistem kendali yang canggih dan adaptif agar AUV dapat bertahan di bawah air. Pengendalian AUV yang baik harus mampu mengatur kecepatan, manuver, dan stabilitas dengan memperhatikan dinamika lingkungan sekitar. Salah satu metode kontrol yang telah banyak digunakan adalah Fuzzy Logic Controller (FLC). Metode ini memiliki keunggulan dalam menangani sistem yang kompleks dan non-linear karena mampu bekerja berdasarkan aturan linguistik yang mudah diadaptasi terhadap kondisi lingkungan bawah air (Yoerger et al., 2007). FLC menggunakan logika berbasis himpunan fuzzy untuk membuat keputusan yang mendekati akurat dalam kondisi dinamis dan tidak pasti, sehingga cocok untuk digunakan pada AUV yang beroperasi di lingkungan yang tidak terprediksi.

Selain kendali, aspek navigasi dan panduan juga merupakan komponen penting dalam desain AUV. Navigasi adalah proses penentuan posisi AUV di bawah air, yang merupakan dasar bagi sistem panduan untuk menghasilkan lintasan yang harus dilewati oleh AUV. Berdasarkan informasi navigasi, sistem panduan akan mengatur lintasan gerak AUV, sementara sistem kendali bertugas menjaga stabilitas selama AUV bergerak mengikuti lintasan tersebut (Yang et al., 2021). Sistem navigasi yang andal sangat dibutuhkan untuk memastikan AUV

dapat menjalankan misinya dengan akurasi tinggi, terutama dalam kondisi lingkungan yang tidak terprediksi. Melihat kebutuhan tersebut, pengembangan kemampuan rancang bangun AUV menjadi sangat penting, terutama dalam mendukung kemajuan teknologi maritim Indonesia. Penelitian yang berfokus pada sistem navigasi, panduan, dan kendali AUV harus terus didorong untuk meningkatkan inovasi teknologi di bidang ini. Dengan mengintegrasikan teknologi kontrol cerdas seperti Fuzzy Logic Controller, diharapkan AUV dapat berkontribusi dalam misi penting di sektor maritim, baik untuk keperluan riset, eksplorasi, maupun tugas strategis lainnya.(Yoerger et al., n.d.)

1.2 Rumusan Masalah

Permasalahan yang diangkat oleh peneliti adalah berfokus pada rancang bangun Autonomous Underwater Vehicle serta menjaga kestabilan Autonomous Underwater Vehicle dengan mengimplementasikan metode Fuzzy Logic Controller untuk kontrol manuver.

1.3 Batasan Masalah

Penelitian ini memiliki beberapa batasan masalah agar lebih terarah dan fokus, dimana batasan masalah pada penelitian saya antara lain adalah :

a. Berfokus pada rancang bangun dan implementasi sistem kontrol untuk manuver Autonmous Underwater Vehicle menggunakan *Fuzzy Logic Controller.*
b. Penelitian ini mengasumsikan kondisi lingkungan bawah air yang dinamis namun terbatas pada skenario tertentu yang disimulasikan, tanpa mempertimbangkan gangguan ekstrem seperti arus bawah laut yang sangat kuat atau turbulensi yang tidak terprediksi.
c. Implementasi sistem kontrol hanya menggunakan Arduino Mega sebagai mikrokontroler, sensor IMU BNO055 untuk pengukuran sudut, motor DC untuk propeller, dan motor stepper dengan driver TB6600 untuk penggerak rudder dan stern.
d. Pengujian dilakukan dalam lingkungan simulasi dan/atau eksperimen laboratorium. Validasi di lingkungan laut nyata tidak termasuk dalam cakupan penelitian ini
e. Penelitian ini berfokus pada pengendalian sudut (Roll, Pitch, Yaw) dengan input setpoint tertentu yang diberikan melalui Serial Monitor. Respons waktu untuk mencapai setpoint menjadi salah satu indikator utama keberhasilan sistem.
f. Penelitian ini tidak mencakup pengembangan sistem navigasi secara keseluruhan, seperti perencanaan lintasan atau penghindaran rintangan.

1.4 Tujuan

Penelitian ini bertujuan untuk merancang dan mengimplementasikan fuzzy logic controller pada sistem tertanam untuk kontrol manuver pada Autonomous Underwater Vehicle, serta mengevaluasi kinerja sistem control yang telah dirancang berdasarkan parameter respon waktu, akurasi sudut, dan stabilitas dalam berbagai scenario pengujian.

1.5 Manfaat

Manfaat yang diperoleh dari penelitian ini adalah sebagai berikut :

a. Bagi Mahasiswa

Meningkatkan pengetahuan serta penerapan sistem kontrol Fuzzy Logic pada sistem elektronika tertanam secara langsung

b. Bagi Instansi

   Instansi atau pihak tertentu dapat memanfaatkan untuk membantu perancangan sistem *Autonomous Underwater Vehicle* secara nyata.

(Halaman Ini Sengaja Dikosongkan)

BAB 2 TINJAUAN PUSTAKA

2.1 Hasil Penelitian Terdahulu

Beberapa penelitian sistem navigasi dan panduan AUV yang sudah pernah dilakukan antara tahun 1990an sampai dewasa ini dapat dijabarkan sebagi berikut:

Tabel 2.1 Penelitian AUV Terdahulu

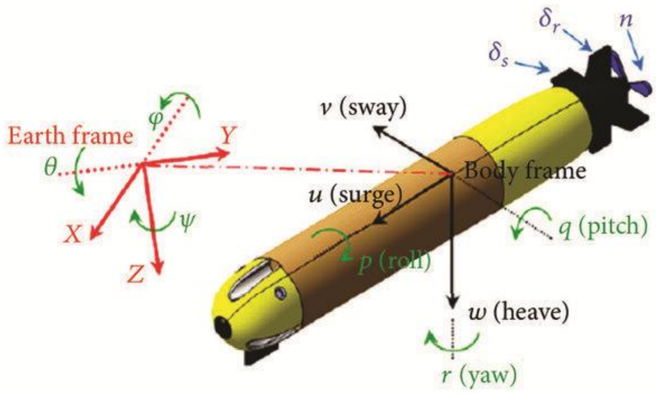
No.	Peneliti	Tahun	Metode	Keterangan
1	An dkk	1997	Heuristic Fuzzy Filtering	Sistem navigasi AUV
2	Fulton dan Cassidy	2001	Extended Kalman Filter (EKF)	Sistem navigasi akustik dan kompas
3	Healey	2002	Extended Kalman Filter (EKF)	Pada 3 subsistem: depth, altitude, dan heading
4	Garcia dkk	-	Augmented State Kalman Filter (ASKF)	-
5	Loebis dkk	2004	EKF, Simple Kalman Filter (SKF), Fuzzy Simple Kalman Filter (FSKF), Fuzzy Extended Kalman Filter (FEKF)	Sistem navigasi
6	Kalyan dkk	-	Extended Kalman Filter (EKF) dengan multi-sensor	Estimasi posisi AUV
7	Lamnas dkk	2007	EKF dan Particle Filter	Pada model non-linear 6-DOF
8	Bartolini dkk	2009	EKF	Pada AUV REMUS
9	He dkk	2011	Extended Information Filter	Pada model non-linear 5-DOF
10	Luque	2011	Kalman Filter dan Maximum Likelihood	-
11	Panish dan Taylor	2011	Sigma-Point Unscented Kalman Filter (UKF)	-
12	Barisic dkk	2012	Sigma-Point Unscented Kalman Filter (UKF)	Pada model non-linear 6-DOF
13	Nurhadi dkk	2012	Ensemble Kalman Filter (EnKF)	Untuk model non-linear 3-DOF (hanya gerak translasi, mengabaikan gerak rotasi)
14	Paull dkk	2014	Extended Kalman Filter (EKF)	Pada model non-linear 6-DOF

Mengacu pada penelitian dan pengembangan terdahulu yang sudah disampaikan dalam paragraf di atas kemudian dipertimbangkan penting untuk mengembangkan sistem kontrol manuver pada AUV. Aspek ini diawali dengan penyusunan model matematika dari AUV, di mana model ideal yang biasa digunakan adalah model non-linier.

2.2 DASAR TEORI

2.2.1 Autonomous Underwater Vehicle

AUV merupakan kendaraan bawah air yang dapat menyelam untuk melakukan misi dan tugas tertentu. AUV menggunakan sistem kendali otomatis yang dipandu oleh perangkat kontrol yang ada di dalam sistem kapal. Untuk melakukan navigasi, AUV terdapat beberapa sensor yang digunakan untuk mengidentifikasi lingkungan sekitarnya dari halangan yang ada di bawah air. Beberapa sensor yang sering digunakan dalam implementasi AUV adalah sensor kedalaman, keseimbangan, jarak, sonar, serta kompas.(Drtil et al., 2006) Sumber energi dari AUV ini menggunakan penyimpanan energi pada baterai. AUV memiliki enam derajat kebebasan dalam melakukan gerak dan manuver. Terdapat tiga koordinat khusus x, y, z dan sudut Euler yaw(ψ) pitch (θ) and roll (φ). (Q. Yang et al., 2017)



Gambar 2. 1 Sistem Gerak 6-DOF AUV

Sesuai pada gambar tersebut, gerakan AUV 6-DOF dapat dinyatakan dengan beberapa vektor berikut,

$$\begin{aligned} \eta_1 &= [x \quad y \quad z]^T \\ v_1 &= [u \quad v \quad w]^T \\ \tau_1 &= [X \quad Y \quad Z]^T \\ \eta_2 &= [\phi \quad \theta \quad \psi]^T \\ v_2 &= [p \quad q \quad r]^T \\ \tau_2 &= [K \quad M \quad N]^T \end{aligned} \tag{2.1}$$

Dengan keterangan, η_1 melambangkan gerak translasi AUV terhadap koordinat pusat bumi, η_2 melambangkan gerak rotasi AUV terhadap koordinat badannya, v_1 & v_2 melambangkan kecepatan linear dan kecepatan angular AUV, dan τ_1 & τ_2 melambangkan gaya dan momen yang bekerja pada AUV. Semua vektor ini akan digunakan pada pemodelan AUV, dengan melibatkan tiga jenis pemodelan yakni kinematika, dinamika, dan hidrodinamika.

2.2.2 Model Kinematika AUV

Pergerakan AUV dengan mengabaikan segala gaya yang bekerja padanya disebut dengan kinematika AUV. Kinematika ini melibatkan kecepatan linear dan angular, sehingga dapat disebut dengan turunan pertama dari posisi dan orientasi yang dilambangkan $\dot{\boldsymbol{\eta}}$. Dengan $\dot{\boldsymbol{\eta}} = \boldsymbol{J}_{\boldsymbol{\Theta}}(\boldsymbol{\eta})\boldsymbol{v}$ lalu $J_\theta(\eta)$ adalah matriks transformasi dari posisi dan orientasi serta $\eta = [\eta_1 \quad \eta_2]^T = [x \quad y \quad z \quad \phi \quad \theta \quad \psi]^T$. Sedangkan, $v = [v_1 \quad v_2] = [u \quad v \quad w \quad p \quad q \quad r]^T$. Dengan demikian, perlu ditentukan model $\dot{\eta}$ untuk $\dot{\eta} = [\dot{\eta}_1 \quad \dot{\eta}_2]^T$,

Untuk kecepatan linear atau turunan pertama dari posisi $(\dot{\eta}_1)$ dapat dinyatakan dengan persamaan berikut,

$$\dot{\eta}_1 = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = J_1(\eta_2)v_1 \qquad (2.2)$$

Dengan $J_1(\eta_2)$ adalah matriks transformasi untuk kecepatan linear. Berdasarkan Fossen ,didapatkan matriks transformasi $J_1(\eta_2)$ sebagai berikut,

$$J_1(\eta_2) = \begin{bmatrix} cos\psi cos\theta & -sin\psi cos\phi + cos\psi sin\theta sin\phi & sin\psi sin\phi + cos\psi sin\theta cos\phi \\ sin\psi cos\theta & cos\psi cos\phi + sin\psi sin\theta sin\phi & -cos\psi sin\phi + sin\psi sin\theta cos\phi \\ -sin\theta & cos\theta sin\phi & cos\theta cos\phi \end{bmatrix} \qquad (2.3)$$

Sehingga turunan pertama darti posisi $(\dot{\eta}_1)$ dapat dinyatakan sebagai berikut,

$$\dot{x} = ucos\psi cos\theta + v[-sin\psi cos\phi + cos\psi sin\theta sin\phi] + w[sin\psi sin\phi + cos\psi sin\theta cos\phi] \qquad (2.4)$$

$$\dot{y} = usin\psi cos\theta + v[cos\psi cos\phi + sin\psi sin\theta sin\phi] + w[-cos\psi sin\phi + sin\psi sin\theta cos\phi] \qquad (2.5)$$

$$\dot{z} = -usin\theta + vcos\theta sin\phi + wcos\theta cos\phi \qquad (2.6)$$

Untuk kecepatan angular atau turunan pertama dari orientasi $(\dot{\eta}_2)$ dapat dinyatakan dengan persamaan berikut,

$$\dot{\eta}_2 = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = J_2(\eta_2)v_2 \qquad (2.7)$$

Dengan $J_2(\eta_2)$ adalah matriks transformasi untuk kecepatan angular. Berdasarkan Fossen, didapatkan matriks transformasi $J_2(\eta_2)$ sebagai berikut,

$$J_2(\eta_2) = \begin{bmatrix} 1 & sin\phi tan\theta & cos\phi tan\theta \\ 0 & cos\phi & -sin\phi \\ 0 & \frac{sin\phi}{cos\theta} & \frac{cos\phi}{cos\theta} \end{bmatrix} \qquad (2.8)$$

Sehingga turunan pertama darti orientasi $(\dot{\eta}_2)$ dapat dinyatakan sebagai berikut,

$$\dot{\phi} = p + qsin\phi tan\theta + rcos\phi tan\theta \qquad (2.9)$$

$$\dot{\theta} = qcos\phi - rsin\phi \qquad (2.10)$$

$$\dot{\psi} = p\frac{sin\phi}{cos\theta} + r\frac{cos\phi}{cos\theta} \qquad (2.11)$$

Dengan demikian, dari pers. Dan pers. Akan didapatkan model kinematika dari AUV 6-DOF sebagai berikut,

$$\dot{\boldsymbol{\eta}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} cos\psi cos\theta & -sin\psi cos\phi + cos\psi sin\theta sin\phi & sin\psi sin\phi + cos\psi sin\theta cos\phi & 0 & 0 & 0 \\ sin\psi cos\theta & cos\psi cos\phi + sin\psi sin\theta sin\phi & -cos\psi sin\phi + sin\psi sin\theta cos\phi & 0 & 0 & 0 \\ -sin\theta & cos\theta sin\phi & cos\theta cos\phi & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & sin\phi tan\theta & cos\phi tan\theta \\ 0 & 0 & 0 & 0 & cos\phi & -sin\phi \\ 0 & 0 & 0 & 0 & \frac{sin\phi}{cos\theta} & \frac{cos\phi}{cos\theta} \end{bmatrix} \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \end{bmatrix} \qquad (2.12)$$

2.2.3 Model Dinamika AUV

Berdasarkan hasil linearisasi kinetika rigid – body dari nonlinear 6 DOF, dapat dinyatakan dengan gaya dan momen sebagai berikut,,

$$m[\dot{u} - vr + wq - x_g(q^2 + r^2) + y_g(pq - \dot{r}) + z_g(pr + \dot{q})] = \sum X_{ext} \qquad (2.13)$$

$$m[\dot{v} - wp + ur - y_g(r^2 + p^2) + z_g(qr - \dot{p}) + x_g(qp + \dot{r})] = \sum Y_{ext} \quad (2.14)$$

$$m[\dot{w} - uq + vp - z_g(p^2 + q^2) + x_g(rp - \dot{q}) + y_g(rq + \dot{p})] = \sum Z_{ext} \quad (2.15)$$

$$I_x\dot{p} + (I_z - I_y)qr + m[y_g(\dot{w} - uq + vp) - z_g(\dot{v} - wp + ur)] = \sum K_{ext} \quad (2.16)$$

$$I_y\dot{q} + (\mathrm{I}_x - I_z)rp + m[z_g(\dot{u} - vr + wq) - x_g(\dot{w} - uq + vp)] = \sum M_{ext} \quad (2.17)$$

$$I_z\dot{r} + (I_y - I_x)pq + m[x_g(\dot{v} - wp + ur) - y_g(\dot{u} - vr + wq)] = \sum N_{ext} \quad (2.18)$$

2.2.4 Model Hidrodinamika AUV

Model dan gaya juga dapat dinyatakan dari sisi eksternal AUV 6-DOF. Hal ini meliputi hidrostatika (F_{HS} & M_{HS}), redaman hidrodinamika (F_D & M_D), massa tambahan (F_A & M_A), gaya dan momen angkat badan AUV (F_{BL} & M_{BL}), gaya dan momen angkat *rudder* AUV (F_{FL} & M_{FL}), serta gaya dorong dan torsi propeler (F_C & M_C). Dengan demikian, dapat disusun sebuah persamaan sebagai berikut:

$$\sum F_{ext} = F_{HS} + F_{HD} + F_A + F_{BL} + F_{FL} + F_C \quad (2.19)$$

$$\sum M_{ext} = M_{HS} + M_{HD} + M_A + M_{BL} + M_{FL} + M_C \quad (2.20)$$

2.2.5 Hidrostatis

Dalam terminologi hidrostatika, gaya gravitasi dan gaya apung merupakan *restoring force*. Lalu, hidrostatika dinotasikan dengan $g(\eta)$ pada umumnya tetapi pada penelitian ini akan digunakan notasi F_{HS} untuk gaya hidrostatika dan M_{HS} untuk momen hidrostatika. Berdasarkan sname, berat badan dan gaya apung dari kapal selam dinyatakan dengan:

$$W = mg \quad (2.21)$$

$$B = \rho g \nabla \quad (2.22)$$

Berikut adalah persamaan hidrostatis dari AUV 6-DOF untuk $F_{HS} = [X_{HS} \quad Y_{HS} \quad Z_{HS}]^T$ dan $M_{Hs} = [K_{HS} \quad M_{HS} \quad N_{HS}]^T$,

$$g(\eta) = \begin{bmatrix} F_{HS} \\ M_{HS} \end{bmatrix} = \begin{bmatrix} X_{HS} \\ Y_{HS} \\ Z_{HS} \\ K_{HS} \\ M_{HS} \\ N_{HS} \end{bmatrix} = \begin{bmatrix} (W - B)\sin(\theta) \\ -(W - B)\cos\theta\sin\phi \\ -(W - B)\cos(\theta)\cos\phi \\ -(y_gW - y_bB)\cos(\theta)\cos\phi + (z_gW - z_bB)\cos\theta\sin\phi \\ (z_gW - z_bB)\sin(\theta) + (x_gW - x_bB)\cos(\theta)\cos\phi \\ -(x_gW - x_bB)\cos\theta\sin\phi - (y_gW - y_bB)\sin(\theta) \end{bmatrix} \quad (2.23)$$

2.2.6 Redaman Hidrodinamika

Hambatan hidrodinamika dinotasikan dengan $D(v)v$ pada umumnya, tetapi notasi yang digunakan pada penelitian ini ialah F_D untuk gaya hidrodinamika dan M_D untuk momen hidrodinamika. Lalu, redaman ini terdiri atas jenis hambatan yakni hambatan aksial, hambatan *crossflow*, dan hambatan *rolling*

Pada hambatan aksial, hambatan ini bekerja ketika AUV ber-translasi pada sumbu-x. Hambatan ini dapat dinyatakan sebagai berikut,

$$X_{u|u|} = -\frac{1}{2}\rho c_d A_f \tag{2.24}$$

Dengan keterangan, ρ adalah massa jenis cairan penghambat, c_d adalah koefisien hambatan aksial pada AUV, dan A_f adalah luas area frontal dari AUV. Pada hambatan *crossflow*, hambatan ini bekerja secara tegak lurus terhadap arah pergerakan AUV. Pada umumnya, hambatan ini dapat berupa arus bawah air. Berikut adalah persamaan dari hambatan *crossflow*:

$$Y_{v|v|} = Z_{w|w|} = -\frac{1}{2}\rho c_{dc}\int_{x_t}^{x_{b2}} 2R(x)dx - 2 \cdot \left(\frac{1}{2}\rho S_{fin} c_{df}\right) \tag{2.25}$$

$$M_{w|w|} = -N_{v|v|} = \frac{1}{2}\rho c_{dc}\int_{x_t}^{x_{b2}} 2xR(x)dx - 2x_{fin} \cdot \left(\frac{1}{2}\rho S_{fin} c_{df}\right) \tag{2.26}$$

$$Y_{r|r|} = -Z_{q|q|} = -\frac{1}{2}\rho c_{dc}\int_{x_t}^{x_{b2}} 2x|x|R(x)dx - 2x_{fin}\left|x_{fin}\right| \cdot \left(\frac{1}{2}\rho S_{fin} c_{df}\right) \tag{2.27}$$

$$M_{q|q|} = N_{r|r|} = -\frac{1}{2}\rho c_{dc}\int_{x_t}^{x_{b2}} 2x^3R(x)dx - 2x_{fin}^3 \cdot \left(\frac{1}{2}\rho S_{fin} c_{df}\right) \tag{2.28}$$

Dengan keterangan, ρ adalah massa jenis air laut, c_{dc} adalah koefisien hambatan dari silinder, $R(x)$ adalah ukuran diameter dari badan AUV, S_{fin} adalah luas permukaan dari rudder, dan c_{df} adalah koefisien hambatan *crossflow* pada permukaan rudder. Untuk persamaan c_{df} dapat dinyatakan dengan persmaan berikut:

$$c_{df} = 0.1 + 0.7t \tag{2.29}$$

Dengan t adalah perbandingan lebar bagian atas dan lebar bagian bawah sirip sepanjang sumbu panjang kendaraan.

Pada hambatan *rolling*, hmabatan ini bekerja ketika AUV melakukan rotasi terhadap sumbu-x yakni *roll*. Hambatan ini dapat dinyatakan sebagai berikut:

$$K_{p|p|} = Y_{vvfr_{mean}^3} \tag{2.30}$$

Dengan Y_{vvf} adalah koefisien hambatan *crossflow* dari kompenen rudder dan r_{mean} adalah rata- rata ketinggian rudder dari pusat kendaraan.

Sehingga secara keseluruhan F_D daapat dinyatakan sebagai berikut:

$$F_D = \begin{bmatrix} X_{u|u|} \\ Y_{v|v|} + Y_{r|r|} \\ Z_{w|w|} - Z_{q|q|} \end{bmatrix} \tag{2.31}$$

Dan M_D dapat dinyatakn sebagai berikut,

$$M_D = \begin{bmatrix} K_{p|p|} \\ M_{w|w|} + M_{q|q|} \\ N_{r|r|} - N_{v|v|} \end{bmatrix} \quad (2.32)$$

2.2.7 Massa Tambahan

Pada hidrodinamika AUV, pergerakan AUV di bawah akan memicu pergerakan fluida disekitarnya sehingga menunjukan seolah massa AUV bertambah dengan massa fluida tersebut. Notasi untuk massa tambahan AUV pada penelitian ini berupa F_A untuk gaya fluida dan M_A untuk momen fluida.

Untuk F_A dinyatakan dengan persamaan berikut,

$$F_A = \begin{bmatrix} X_A \\ Y_A \\ Z_A \end{bmatrix} = \begin{bmatrix} X_{\dot{u}}\dot{u} + Z_{\dot{w}}wq + Z_{\dot{q}}q^2 - Y_{\dot{v}}vr - Y_{\dot{r}}\dot{r}^2 \\ Y_{\dot{v}}\dot{v} + Y_{\dot{r}}\dot{r} + X_{\dot{u}}ur - Z_{\dot{w}}wp - Z_{\dot{q}}pq \\ Z_{\dot{w}}\dot{w} + Z_{\dot{q}}\dot{q} - X_{\dot{u}}uq + Y_{\dot{v}}vp + Y_{\dot{r}}rp \end{bmatrix} \quad (2.33)$$

Sedangkan, M_A dinyatakan dengan persamaan berikut:

$$M_A = \begin{bmatrix} K_A \\ M_A \\ N_A \end{bmatrix} = \begin{bmatrix} K_{\dot{p}}\dot{p} \\ M_{\dot{w}}\dot{w} + M_{\dot{q}}\dot{q} - (Z_{\dot{w}} - X_{\dot{u}})uw - Y_{\dot{r}}vp + (K_{\dot{p}} - N_{\dot{r}})rp - Z_{\dot{q}}uq \\ N_{\dot{v}}\dot{v} + N_{\dot{r}}\dot{r} - (X_{\dot{u}} - Y_{\dot{v}})uv + Z_{\dot{q}}wp - (K_{\dot{p}} - M_{\dot{q}})pq + Y_{\dot{r}}ur \end{bmatrix} \quad (2.34)$$

2.2.8 Gaya dan Momen Angkat Badan AUV

Gaya dan momen angkat pada badan AUV adalah salah satu aspek penting dalam dinamika bawah air, yang berkaitan dengan interaksi antara badan AUV dan fluida sekitarnya. Dalam teori hidrodinamika, gaya angkat ($Y_{uvl} dan\ Y_{uwl}$) dan momen angkat (M_{uwl} dan N_{uvl}) timbul akibat distribusi tekanan yang tidak merata pada permukaan badan AUV saat bergerak melalui air. Persamaan yang menggambarkan gaya dan momen ini didasarkan pada prinsip dasar mekanika fluida, di mana gaya angkat dan momen angkat dipengaruhi oleh beberapa parameter utama seperti massa jenis air (ρ), diameter badan AUV ($\boldsymbol{d}$), koefisien gaya angkat (c_{yd}), sudut serang (β), serta posisi pusat tekanan relatif terhadap pusat massa (x_{cp}). Persamaan (2.35) dan (2.36) menunjukkan bahwa gaya dan momen angkat memiliki hubungan linier dengan parameter-parameter tersebut, yang mencerminkan seberapa besar AUV mampu menghasilkan gaya angkat dan momen berdasarkan desain fisiknya dan kondisi hidrodinamika di bawah air. Pemahaman tentang gaya dan momen ini sangat penting dalam perancangan AUV, karena memengaruhi kestabilan dan kemampuan manuvernya di lingkungan yang kompleks dan dinamis. Teori ini memberikan dasar matematis untuk menganalisis performa hidrodinamik AUV, terutama dalam menjaga stabilitas dan efisiensi gerakan selama menjalankan tugas di bawah air.

$$Y_{uvl} = Z_{uwl} = -\frac{1}{2}\rho d^2 c_{yd\beta} \quad (2.35)$$

$$M_{uwl} = -N_{uvl} = -\frac{1}{2}\rho d^2 c_{yd\beta x_{cp}} \quad (2.36)$$

2.2.9 Gaya dan Momen Angkat Rudder AUV

Gaya dan momen angkat rudder pada AUV merupakan elemen penting dalam analisis dinamika gerak, terutama untuk memahami bagaimana rudder memengaruhi kestabilan dan

kemampuan manuver AUV. Rudder berfungsi sebagai perangkat kontrol yang mengarahkan dan menstabilkan pergerakan AUV dengan menghasilkan gaya dan momen angkat. Persamaan yang digunakan untuk menghitung gaya dan momen angkat ini menggambarkan hubungan langsung antara karakteristik fisik rudder, seperti luas permukaan (S_{fin}) dan posisi rudder terhadap pusat massa (x_{fin}), serta parameter lingkungan seperti massa jenis air (ρ) dan koefisien angkat ($c_{L\alpha}$). Gaya angkat rudder diperoleh dari interaksi hidrodinamik antara air dan permukaan rudder, di mana perubahan sudut rudder akan menghasilkan gaya angkat yang memengaruhi arah gerak AUV. Selain itu, momen angkat yang dihasilkan rudder sangat bergantung pada jarak posisi rudder dari pusat gravitasi AUV, yang menentukan seberapa besar pengaruh rudder terhadap kemampuan AUV untuk berputar atau bermanuver. Teori ini menjadi dasar penting dalam perancangan sistem kendali gerak AUV, terutama dalam memastikan bahwa rudder dapat memberikan respons yang memadai terhadap perubahan arah atau stabilisasi. Pemahaman tentang gaya dan momen angkat rudder tidak hanya membantu meningkatkan efisiensi navigasi AUV, tetapi juga memastikan kestabilan selama operasinya di lingkungan bawah air yang dinamis.(Fossen, n.d.)

$$Y_{uu\delta_r} = -Y_{uv_f} = \rho c_{L\alpha} S_{fin} \tag{2.37}$$

$$Z_{uu\delta_s} = Z_{uwf} = -p c_{L\alpha} S_{fin} \tag{2.38}$$

$$Y_{urf} = -Z_{uqf} = -p c_{L\alpha} S_{fin} x_{fin} \tag{2.39}$$

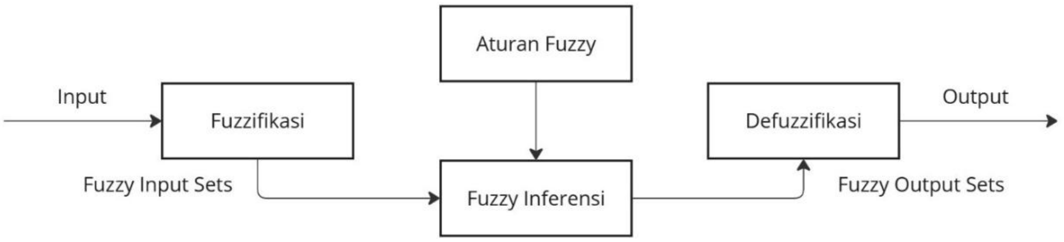
$$M_{uu\delta_s} = M_{uwf} = p c_{L\alpha} S_{fin} x_{fin} \tag{2.40}$$

$$N_{uu\delta_r} = -N_{uvf} = p c_{L\alpha} S_{fin} x_{fin} \tag{2.41}$$

$$M_{uqf} = N_{urf} = -p c_{L\alpha} S_{fin} x_{fin}^2 \tag{2.42}$$

2.2.10 Fuzzy Logic Controller

Dalam beberapa loop kontrol yang kompleks, pengendali logika fuzzy (FLC) dapat bekerja dengan baik. FLC menggabungkan logika manusia dengan pengalaman dan menunjukkan potensi kemampuan yang kuat dalam mengatasi sistem yang kompleks.(Intelligent Control and Automation (WCICA), 2010 8th World Congress on : Date, 7-9 July 2010., 2010). Sistem logika fuzzy adalah metode yang digunakan untuk menangani ketidakpastian dalam pengambilan keputusan. Ini sangat berguna dalam aplikasi teknik, di mana data sering kali tidak pasti atau tidak lengkap.



Gambar 2. 2 Diagram blok Fuzzy Logic Controller

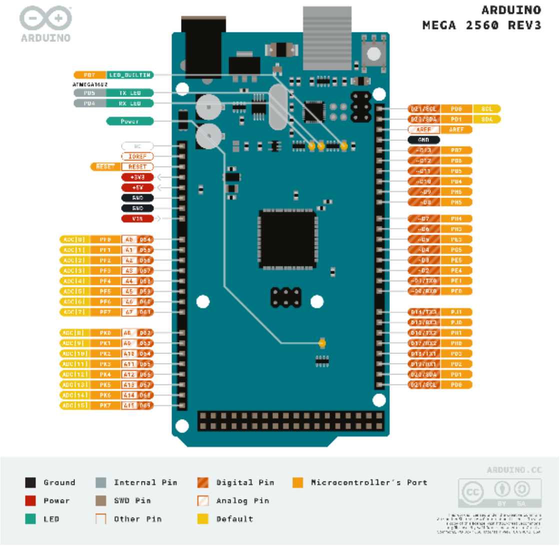
Pada gambar diagram blok (2.2) Proses dimulai dengan pengumpulan data input yang bersifat jelas dan terukur. Namun, data ini sering kali tidak cukup untuk membuat keputusan yang tepat, terutama ketika melibatkan variabel yang bersifat ambigu. Di sinilah peran Fuzzifikasi menjadi penting. Fuzzifikasi mengubah data input yang tegas menjadi bentuk *fuzzy* dengan mengaitkan

nilai-nilai tersebut dengan derajat keanggotaan dalam himpunan *fuzzy*. Misalnya, suhu 29°C dapat dikategorikan sebagai "hangat" dengan derajat keanggotaan tertentu, memungkinkan sistem untuk memahami bahwa suhu tersebut tidak sepenuhnya hangat atau dingin, tetapi berada di antara keduanya. Setelah data diubah menjadi bentuk *fuzzy*, sistem menggunakan sekumpulan aturan yang dikenal sebagai *fuzzy* rules. Aturan ini biasanya ditulis dalam format "Jika... Maka...", yang menghubungkan kondisi input dengan output yang diinginkan. Misalnya, jika suhu dianggap hangat, maka kecepatan kipas dapat diatur pada tingkat sedang. Aturan-aturan ini membantu sistem untuk membuat keputusan berdasarkan kondisi yang ada. Bagian inti dari sistem ini adalah *fuzzy inference engine*, yang bertugas menerapkan aturan *fuzzy* pada input *fuzzy* untuk menghasilkan *output fuzzy*. Mesin ini mengolah informasi dan menggabungkan hasil dari berbagai aturan untuk menentukan output yang paling sesuai. Setelah output *fuzzy* dihasilkan, langkah selanjutnya adalah defuzzifikasi. Proses ini mengubah output *fuzzy* kembali menjadi nilai yang dapat digunakan, seperti kecepatan kipas yang dinyatakan dalam persentase dari kecepatan maksimum. Akhirnya, hasil dari proses ini adalah output yang dapat diterapkan dalam berbagai aplikasi, seperti pengendalian perangkat atau pengambilan keputusan. Dengan pendekatan ini, sistem logika fuzzy memberikan cara yang lebih baik untuk menangani ketidakpastian dan ambiguitas dalam data, menjadikannya sangat relevan dalam bidang teknik elektro, di mana banyak parameter yang harus dipertimbangkan secara bersamaan. (Liu et al., 2015) . Pada teori lain, yang menjelaskan bahwasanya Metode ini memungkinkan representasi logis terhadap ketidakpastian dan ketidaktepatan dalam sistem melalui konsep himpunan fuzzy (fuzzy set). Berbeda dengan logika klasik yang hanya mengenal nilai biner (0 atau 1), fuzzy logic memungkinkan nilai kebenaran berada dalam rentang kontinu antara 0 hingga 1. Pendekatan ini memberikan fleksibilitas dalam mengatasi masalah yang tidak dapat dimodelkan secara matematis secara presisi. Dalam dunia pengendalian, Fuzzy Logic Controller (FLC) adalah teknik pengontrolan berbasis logika fuzzy yang dirancang untuk menangani sistem kompleks, non-linear, atau sistem yang sulit dimodelkan secara tradisional. FLC bekerja dengan prinsip pengambilan keputusan berbasis aturan linguistik yang menyerupai cara manusia berpikir dan membuat keputusan.(Zadeh, 1965). Tahapan utama dalam Fuzzy Logic Controller (FLC) dimulai dengan proses fuzzyfikasi, yaitu mengubah input yang bersifat tegas (crisp) menjadi variabel fuzzy. Pada tahap ini, nilai input diterjemahkan ke dalam derajat keanggotaan menggunakan fungsi keanggotaan tertentu, seperti fungsi segitiga, trapesium, atau kurva Gaussian. Proses ini bertujuan untuk merepresentasikan input dalam bentuk linguistik yang dapat dipahami oleh sistem fuzzy. Setelah itu, proses inferensi fuzzy dilakukan. Pada tahap ini, aturan-aturan berbasis logika fuzzy diterapkan untuk menentukan tindakan yang sesuai. Aturan-aturan ini biasanya dinyatakan dalam bentuk "IF-THEN", di mana hubungan antara kondisi dan aksi dirumuskan secara deskriptif. Mesin inferensi menggunakan basis aturan untuk mencocokkan input fuzzy dengan kondisi yang telah ditentukan, menghasilkan output berupa nilai fuzzy. Langkah terakhir adalah defuzzyfikasi, yaitu konversi output fuzzy menjadi nilai tegas (crisp) yang dapat diterapkan pada sistem fisik. Output fuzzy yang dihasilkan dari tahap inferensi diterjemahkan kembali menjadi nilai numerik menggunakan metode tertentu, seperti metode centroid, bisector, atau mean of maximum (MoM). Proses ini memastikan bahwa output yang dihasilkan dapat digunakan untuk mengontrol perangkat atau sistem sesuai kebutuhan. Melalui ketiga tahapan ini, Fuzzy Logic Controller mampu menangani berbagai sistem yang kompleks dengan pendekatan berbasis logika linguistik yang intuitif dan fleksibel. (Mamdani & Assilian, 1975).

2.2.11 Sistem Mikrokontroller

Pada rancang bangun AUV ini, peneliti menggunakan mikrokontroller jenis Arduino Mega 2560, alasan menggunakan mikrokontroller tersebut adalah karena memiliki pin analog

input serta analog output yang cukup banyak untuk mengendalikan dan mengimplementasikan algoritma kontrol PID dan Adaptive Fuzzy. Arduino Mega 2560 adalah papan mikrokontroler yang berbasis ATmega2560. Board ini memiliki 54 pin input/output digital (15 di antaranya dapat digunakan sebagai output PWM), 16 input analog, 4 UART (port serial perangkat keras), osilator kristal 16 MHz, koneksi USB, colokan listrik, header ICSP, dan tombol reset. Board ini berisi semua yang diperlukan untuk mendukung mikrokontroler; cukup hubungkan ke komputer dengan kabel USB atau daya dengan adaptor AC-ke-DC atau baterai untuk memulai. Papan Mega 2560 kompatibel dengan sebagian besar pelindung yang dirancang untuk Uno dan papan sebelumnya Duemilanove atau Diecimila. Berikut adalah spesifikasi dari mikrokontroller Arduino Mega 2560.



Gambar 2. 3 Mikrokontroller Arduino Mega 2560

Dan berikut ini adalah tabel spesifikasi teknis dari Arduino Mega 2560:

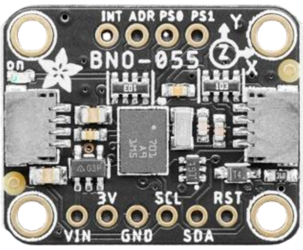
Tabel 2.2 Spesifikasi Teknis Arduino Mega 2560

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
LED_BUILTIN	13
Length	101.52 mm
Width	53.3 mm
Weight	37 g

2.2.12 Sistem Inertial Measurement Unit

IMU (Inertial Measurement Unit) adalah perangkat elektronik yang digunakan untuk mengukur percepatan linear dan kecepatan sudut suatu objek. Sensor ini sering digunakan

dalam aplikasi navigasi, robotika, dan stabilisasi untuk mendapatkan data orientasi (Roll, Pitch, Yaw) serta gerakan. Peneliti menggunakan sensor IMU dengan tipe BNO055. BNO055 adalah sensor IMU dari Bosch yang memiliki keunggulan utama berupa integrasi unit pengolah data (*sensor fusion*). Tidak seperti IMU biasa yang hanya menyediakan data mentah, BNO055 memiliki prosesor internal yang secara otomatis memproses data dari akselerometer, giroskop, dan magnetometer untuk memberikan keluaran dalam bentuk data orientasi yang sudah dikalibrasi.



Gambar 2. 4 Sensor Inertial Measurement Unit

BNO055 menggunakan algoritma *sensor fusion* untuk menggabungkan data dari ketiga sensor tersebut. Hal ini memungkinkan sensor memberikan data orientasi yang stabil, akurat, dan bebas dari noise. Misalnya, magnetometer membantu memperbaiki kesalahan drift pada giroskop, sementara akselerometer digunakan untuk mendeteksi posisi dan orientasi relatif terhadap gravitasi. .Ada tiga parameter utama pada sensor IMU BNO055:

- **Akselerometer**: Mengukur percepatan linear pada sumbu X, Y, dan Z. Data ini digunakan untuk mendeteksi gerakan translasi. Rumus percepatan linier:

$$a = \frac{\Delta kecepatan_{x,y,z}}{\Delta t} \tag{2.43}$$

Di mana a adalah percepatan, Δv adalah perubahan kecepatan, dan Δt adalah perubahan waktu.

- **Giroskop**: Mengukur kecepatan sudut di sekitar tiga sumbu utama, membantu mendeteksi rotasi atau perubahan orientasi. Rumus kecepatan sudut :

$$\omega = \frac{sudut_{\phi,\theta,\psi}}{\Delta t} \tag{2.44}$$

Di mana ω adalah kecepatan sudut, Δθ adalah perubahan sudut, dan Δt adalah perubahan waktu.

- **Magnetometer**: Mengukur medan magnet bumi untuk menentukan arah utara magnetik, membantu mengoreksi drift pada giroskop.

BNO055 menggunakan algoritma *sensor fusion* untuk menggabungkan data dari akselerometer, giroskop, dan magnetometer. Proses ini menghasilkan data orientasi yang stabil dan akurat. Algoritma ini melibatkan:

- Kalman Filter: Untuk mengurangi noise dan mengintegrasikan data dari berbagai sensor.
- Keluaran Data Orientasi:
  - Euler Angles: $Roll\ (\phi), Pitch\ (\theta)$, dan $Yaw\ (\psi)$.
  - Quaternion: Representasi orientasi dalam ruang tiga dimensi.

Persamaan integrasi untuk posisi:

$$p(t) = p_0 + \int_0^t v(\tau)d\tau \tag{2.45}$$

Di mana p(t) adalah posisi, dan v(t) adalah kecepatan.
Integrasi Kecepatan untuk orientasi :

$$\theta(t) = \theta_0 + \int_0^t \omega(\tau)d\tau \tag{2.46}$$

Di mana $\theta(t)$ adalah sudut orientasi.

2.2.13 Motor DC

Motor DC spindle 500 watt 48 volt 10 ampere adalah pilihan yang tepat untuk digunakan sebagai penggerak propeller pada AUV (Autonomous Underwater Vehicle) Anda. Motor ini memiliki daya yang cukup besar untuk menghasilkan dorongan yang diperlukan dalam berbagai kondisi operasi bawah air, seperti mengatasi hambatan air, arus laut, dan beban mekanis lainnya. Penggunaan motor ini untuk propeller AUV didasarkan pada kemampuannya menghasilkan kecepatan putaran tinggi yang stabil, yang sangat penting untuk menciptakan gaya dorong (thrust) yang efisien. Dengan daya 500 watt, motor ini mampu memutar propeller dengan kecepatan yang cukup untuk menghasilkan dorongan yang diperlukan agar AUV dapat bergerak maju, mundur, atau mempertahankan posisinya di dalam air. Tegangan operasi 48 volt memastikan motor bekerja pada kondisi yang optimal, sementara arus maksimum 10 ampere memungkinkan motor memberikan torsi yang cukup untuk mengatasi hambatan dari air.

Gambar 2. 5 Motor DC 500 Watt

Motor ini bekerja berdasarkan prinsip elektromagnetik, di mana arus listrik yang dialirkan melalui kumparan menciptakan medan magnet yang menghasilkan torsi untuk memutar poros motor. Ketika dihubungkan dengan propeller, motor ini akan mentransfer energi mekanis ke air, menciptakan gaya dorong yang menggerakkan AUV. Dalam desain sistem propulsi AUV, efisiensi motor sangat penting untuk memaksimalkan penggunaan daya baterai, dan motor spindle ini memiliki efisiensi yang cukup tinggi untuk mendukung misi bawah air yang panjang. Selain itu, motor ini dirancang untuk bekerja pada kecepatan tinggi, yang cocok untuk propeller dengan desain pitch rendah hingga sedang. Kecepatan putaran motor dapat diatur menggunakan driver PWM (Pulse Width Modulation), memungkinkan kontrol presisi terhadap kecepatan dan arah gerak AUV. Sistem kontrol ini sangat penting dalam menjaga stabilitas dan manuver AUV, terutama saat menghadapi perubahan kondisi lingkungan bawah air, seperti arus laut atau turbulensi. Dengan spesifikasi dan karakteristiknya, motor DC spindle ini adalah solusi yang

andal dan efisien untuk sistem propulsi AUV Anda. Penggunaannya akan memberikan dorongan yang cukup untuk mendukung navigasi, stabilitas, dan manuver AUV di lingkungan bawah air yang dinamis. Jika diperlukan, desain propeller dan sistem kontrol dapat disesuaikan untuk mengoptimalkan kinerja keseluruhan sistem propulsi.

2.2.14 Rangkaian kontrol PWM Motor DC

Modul PWM (Pulse Width Modulation) controller adalah perangkat yang digunakan untuk mengatur kecepatan, arah, dan daya motor DC dengan cara yang efisien. Prinsip kerja modul ini didasarkan pada pengendalian sinyal listrik yang dikirimkan ke motor dalam bentuk pulsa-pulsa dengan durasi tertentu. Durasi pulsa ini, yang disebut duty cycle, menentukan seberapa besar daya yang diterima oleh motor. Dengan mengubah duty cycle, kecepatan putaran motor dapat diatur secara presisi tanpa membuang energi sebagai panas, seperti yang terjadi pada kontrol berbasis resistansi. Dalam aplikasi motor DC, PWM controller memungkinkan pengguna untuk mengatur kecepatan dengan cara memvariasikan tegangan rata-rata yang diterima motor, tanpa mengurangi tegangan suplai secara langsung. Sebagai contoh, pada duty cycle 50%, motor menerima sinyal "on" selama separuh waktu dan "off" selama separuh waktu, sehingga tegangan rata-rata yang diterima motor menjadi setengah dari tegangan suplai. Hal ini membuat PWM sangat efisien karena energi tidak hilang dalam bentuk panas. Modul PWM sering kali dilengkapi dengan fitur tambahan, seperti pengendali arah motor menggunakan rangkaian H-Bridge, yang memungkinkan motor berputar maju atau mundur. Beberapa modul juga memiliki proteksi terhadap arus lebih, panas berlebih, atau tegangan lebih, yang membantu menjaga motor dan sistem kelistrikan tetap aman.

Gambar 2. 6 Rangkaian Kontrol PWM

WM bekerja dengan cara mengatur duty cycle (D), yaitu rasio waktu ON terhadap periode total sinyal, untuk mengontrol daya rata-rata () yang dikirimkan ke beban (motor DC). Tegangan rata-rata output:

$$V_{out} = D \ x \ V_{in} \tag{2.47}$$

Dimana:

V_{out} : Tegangan rata-rata yang diterima motor
V_{in} : Tegangan input modul
D : Duty cycle (0 hingga 1, atau 0% hingga 100%

Sebagai contoh, jika = 24 V dan D = 50%, maka:

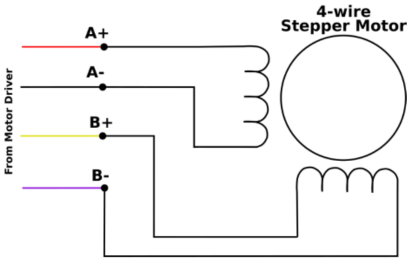
$$V_{out} = 0.5 \ x \ 24 = 12 \ V \tag{2.48}$$

Tegangan rata-rata ini menentukan kecepatan motor (RPM) secara langsung. Dalam pengaplikasian AUV (Autonomous Underwater Vehicle), modul PWM digunakan untuk mengatur kecepatan motor DC yang menggerakkan propeller. Dengan menggunakan PWM, kecepatan propeller dapat disesuaikan untuk menghasilkan gaya dorong yang diperlukan untuk navigasi atau manuver. Sistem ini juga memungkinkan penghematan energi yang signifikan, yang sangat penting untuk misi bawah air yang panjang. Modul PWM dapat diintegrasikan dengan sensor lain, seperti IMU, untuk menciptakan sistem kontrol yang responsif terhadap perubahan kondisi lingkungan, seperti arus laut atau turbulensi. Secara keseluruhan, modul PWM adalah komponen kunci dalam sistem kontrol motor DC yang menawarkan efisiensi, fleksibilitas, dan presisi tinggi, menjadikannya ideal untuk berbagai aplikasi, termasuk sistem propulsi AUV.

2.2.15 Motor Stepper

Motor stepper NEMA17 adalah jenis motor listrik yang dirancang untuk menghasilkan gerakan presisi tinggi melalui langkah-langkah kecil yang terkontrol. Motor ini memiliki ukuran standar dengan dimensi flensa 1,7 inci (sekitar 42 mm), yang membuatnya kompatibel dengan berbagai aplikasi mekanis dan elektronik. NEMA17 sering digunakan dalam perangkat seperti printer 3D, mesin CNC, dan robotika, termasuk dalam proyek AUV (Autonomous Underwater Vehicle) untuk mengontrol sudut rudder dan stern. Motor ini bekerja berdasarkan prinsip elektromagnetik, di mana rotor bergerak secara bertahap ke posisi tertentu sesuai dengan urutan sinyal listrik yang dikirimkan ke kumparan stator. Setiap langkah rotor mencerminkan perubahan sudut kecil, biasanya 1,8 derajat per langkah penuh, atau 200 langkah untuk satu putaran penuh. Dengan driver yang mendukung mode microstepping, resolusi langkah dapat ditingkatkan untuk menghasilkan gerakan yang lebih halus. Berikut adalah gambaran hardware dari motor stepper NEMA17 untuk 4 wire:

Gambar 2. 7 Motor Stepper NEMA17



Gambar 2. 8 Konstruksi Motor Stepper dari (www.jwardtech.com)

Dalam aplikasi AUV, motor stepper NEMA17 digunakan untuk mengontrol sudut rudder (pengarah horizontal) dan stern (pengarah vertikal). Dengan kemampuan motor ini untuk bergerak secara presisi, AUV dapat mengatur arah dan kedalaman dengan akurasi tinggi, yang

sangat penting untuk menjaga stabilitas dan manuver di bawah air. Selain itu, motor ini memiliki kemampuan untuk menahan posisinya pada langkah tertentu tanpa gerakan balik, sehingga sudut rudder dan stern dapat dipertahankan meskipun terdapat gangguan dari arus bawah air. Keunggulan utama motor stepper NEMA17 adalah presisi tinggi, stabilitas torsi pada kecepatan rendah, dan kemudahan integrasi ke dalam sistem mekanis. Motor ini juga tidak memerlukan sensor posisi tambahan karena posisinya dapat dikontrol secara langsung melalui jumlah langkah yang dijalankan. Namun, motor ini cenderung mengonsumsi daya lebih banyak dibandingkan motor DC biasa, terutama saat menahan posisi, dan torsi akan menurun pada kecepatan tinggi.

Secara keseluruhan, motor stepper NEMA17 adalah pilihan yang andal untuk mengontrol rudder dan stern pada AUV. Dengan kombinasi driver stepper yang sesuai, seperti TB6600, dan sistem kontrol berbasis PWM, motor ini dapat memberikan kinerja optimal untuk mendukung navigasi dan stabilitas AUV dalam berbagai kondisi lingkungan bawah air.

2.2.16 Driver Stepper TB6600

Driver stepper TB6600 adalah modul pengendali motor stepper berbasis IC Toshiba TB6600, yang dirancang untuk mengontrol motor stepper dengan efisiensi tinggi dan presisi yang baik. Driver ini sering digunakan dalam aplikasi seperti mesin CNC, printer 3D, robotika, dan sistem kontrol lainnya, termasuk dalam proyek AUV (Autonomous Underwater Vehicle) untuk menggerakkan motor stepper yang mengontrol rudder dan stern. Driver TB6600 mampu mengendalikan motor stepper hingga 4.5A per fase dengan tegangan input 9–42V DC, membuatnya cocok untuk motor stepper seperti NEMA17 dan NEMA23. Driver ini mendukung mode microstepping, yang memungkinkan motor stepper bergerak dalam langkah-langkah kecil untuk menghasilkan gerakan yang lebih halus dan presisi.

Gambar 2. 9 Driver Motor Stepper TB6600

Pada gambar, terminal/pin memiliki fungsi sebagai berikut:

a. Terminal power:
   - VCC : Tegangan input untuk driver
   - GND : Ground (negatif) dari sumber daya

b. Terminal Output Motor:

   - A+, A- : Tegangan input untuk driver
   - B+, B- : Ground (negatif) dari sumber daya

Pastikan polaritas kabel stepper motor sesuai dengan kumparan, karena pengaturan yang salah dapat menyebabkan motor tidak berputar atau bergetar. Peneliti telah membuat

panduan cara melakukan pengecekan polaritas kabel stepper motor. Silahkan baca panduan berikut:

1. Putar knob Avometer ke arah simbol hambatan (Ω)

Gambar 2. 10 Pengaturan Avometer untuk cek Polaritas Motor Stepper

2. Sambungkan 2 kabel dari stepper motor pada Avometer

Gambar 2. 11 Tampilan Avometer ketika Kabel Stepper Motor Tidak Sepasang

3. Amati nilai yang tampil pada LCD Avometer, jika bernilai 0.L maka kabel tersebut tidak sepasang.
4. Jika nilai yang tampil pada LCD Avometer menunjukkan adanya hambatan, maka kabel tersebut sepasang.
5. Berikut adalah contoh ketika

Gambar 2. 12 Tampilan Avometer ketika Kabel Stepper Motor Sepasang

6. Jika sudah mengetahui bahwasannya kabel berwarna merah dan biru sudah sepasang, maka bisa disambungkan ke terminal (A+, A-) dan sepasang lainnya disambungkan ke (B+, B-).

c. Terminal Sinyal kontrol:

• PUL+, PUL- (Pulse)	:	Tegangan input untuk driver
• DIR+,DIR(Direction)	:	Sinyal untuk menentukan arah putaran motor (searah jarum jam atau berlawanan arah jarum jam).
• ENA+,ENA- (Enable)	:	Sinyal untuk mengaktifkan atau menonaktifkan driver. Jika sinyal ini aktif, driver akan "memegang" motor pada posisinya; jika dinonaktifkan, motor akan bebas bergerak

2.2.16.1 Hubungan Sinyal Pulsa(PUL) dan Gerakan Motor

Stepper motor bekerja dengan prinsip **gerakan per langkah**, di mana setiap pulsa sinyal *PUL* yang diterima oleh driver akan memutar motor sebesar satu langkah atau microstep, tergantung pada pengaturan microstepping. Berikut adalah perhitungan sudut per-langkah:

$$\theta_{step} = \frac{360°}{N_{step}} \tag{2.49}$$

Dimana:

θ_{step} : Sudut rotasi per langkah penuh (dalam derajat)

N_{step} : Jumlah langkah penuh dalam satu putaran motor (misalnya, 200 langkah untuk motor stepper 1.8°)

Jika driver diatur pada mode ***microstepping*** dengan faktor pembagian μ_{step}, maka:

$$\theta_{microstep} = \frac{\theta_{step}}{\mu_{\text{step}}} \tag{2.50}$$

Dimana:

μ_{step} : Faktor *microstepping* (misalnya : 2, 4, 8, 16, dst.)

Contoh:

- Untuk stepper motor 1.8; $N_{step} = 200$; $\mu_{\text{step}} = 8$; maka:

$$\theta_{step} = \frac{360}{200}\ 1.8° \tag{2.51}$$

$$\theta_{microstep} = \frac{1.8}{8}\ 0.225° \tag{2.52}$$

Maka, setiap pulsa *PUL* akan menghasilkan motor berputar sebesar $0.225°$

Tabel 2.3 Pengaturan Microstepping

SW1	*SW2*	*SW3*	*Mode*	*Faktor Microstepping (µStep)*
OFF	OFF	OFF	Full Step	1
ON	OFF	OFF	Half Step	2
OFF	ON	OFF	Quarter Step	4
ON	ON	OFF	Eighth Step	8
OFF	OFF	ON	Sixteenth Step	16

Keterangan:

- *Full Step (µStep =1):* Motor bekerja dengan langkah penuh (misalnya, 1.8° per langkah).
- *Half Step (µStep =2):* Langkah penuh dibagi menjadi dua, sehingga motor bergerak 0.9° per langkah.
- *Quarter Step (µStep =4):* Langkah penuh dibagi menjadi empat, sehingga motor bergerak 0.45° per langkah.
- *Eighth Step (µStep =8):* Langkah penuh dibagi menjadi delapan, sehingga motor bergerak 0.225° per langkah.

Berikut adalah fitur utama pada driver stepper TB6600:

- ***Microstepping***
  TB6600 mendukung berbagai mode *microstepping*, seperti 1/2, 1/4, 1/8, dan 1/16 langkah. Fitur ini memungkinkan motor stepper beroperasi dengan gerakan yang lebih halus dan mengurangi getaran, yang sangat penting untuk aplikasi yang membutuhkan presisi tinggi.
- **Arus Maksimum 4.5A**
  Driver ini mampu menangani motor stepper dengan arus hingga 4.5A per fase, menjadikannya ideal untuk motor stepper dengan kebutuhan daya tinggi.
- **Proteksi Internal**
  TB6600 dilengkapi dengan proteksi terhadap panas berlebih (*overheating*), arus lebih (*overcurrent*), dan tegangan lebih (*overvoltage*), yang membantu menjaga keandalan dan umur panjang perangkat.
- **Kemudahan Integrasi**
  Driver ini dirancang untuk kompatibilitas dengan berbagai mikrokontroler, seperti Arduino, Raspberry Pi, atau PLC. Sinyal kontrol diberikan melalui input step, direction, dan enable, yang mempermudah pengendalian motor stepper.
- **Efisiensi Tinggi**
  Dengan desain berbasis MOSFET, TB6600 memiliki efisiensi tinggi dalam mengubah sinyal kontrol menjadi gerakan motor, sekaligus mengurangi panas yang dihasilkan.

Dalam AUV, driver TB6600 digunakan untuk mengontrol motor stepper yang menggerakkan rudder dan stern. Dengan menggunakan mode microstepping, rudder dan stern dapat digerakkan dengan presisi tinggi untuk mengatur orientasi dan stabilitas AUV. Driver ini memastikan motor stepper bekerja secara efisien meskipun dalam kondisi beban yang bervariasi, seperti saat menghadapi arus laut.

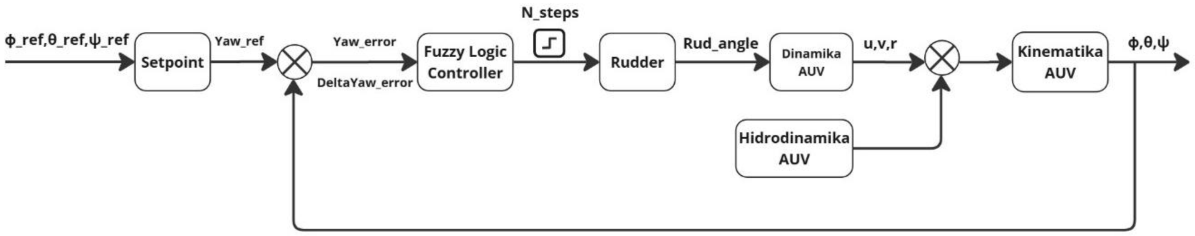
BAB 3 METODOLOGI

3.1 Metode Kontrol Fuzzy Logic Controller

Autonomous Underwater Vehicle dapat melakukan manuver Yaw dengan menggunakan rudder dan manuver *Pitch* menggunakan *Stern*. Pergerakan rudder dan stern dikontrol menggunakan motor stepper. Pergerakan sudut motor stepper didasarkan dengan referensi *setpoint* dari sensor IMU(Inertial Measurement Unit) sebagai *feedback*. Metode yang digunakan untuk melakukan kontrol Yaw dan Pitch adalah *Fuzzy Logic Controller*. Dimana nilai error dan delta error dari pengukuran Inertial Measurement Unit akan selalu dikoreksi sebagai *feedback*. Kemudian akan diberikan aksi kontrol kepada motor stepper melalui driver stepper TB6600. Pada penelitian ini, bertujuan untuk mengimplementasikan kontroller Fuzzy Logic kepada mikrokontroller. Alasan menggunakan Fuzzy Logic Controller adalah karena plant yang digunakan adalah plant non-linier, sehingga FLC lebih dapat diandalkan untuk diimplementasikan untuk manuver pada Autonomous Underwater Vehicle.

3.1.1 Diagram Blok Perancangan Fuzzy Logic Controller

Berikut adalah diagram alir implementasi Fuzzy Logic Controller untuk kontrol manuver pada Autonomous Underwater Vehicle :



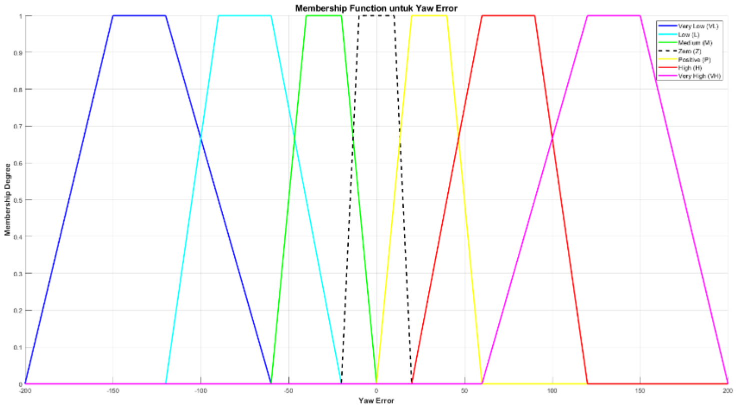
Gambar 3. 1 Diagram Blok Sistem Kontrol AUV

3.1.2 Perancangan Membership Function untuk Yaw Error

Peneliti telah merancang fungsi keanggotaan untuk error pada manuver Yaw dan Pitch. Berikut adalah hasil rancangan membership function yang telah dibuat oleh peneliti:

Tabel 3. 1 Fungsi Keanggotaan Fuzzy Input

Kategori	**Rentang (derajat)**	**Bentuk Fungsi**
Very Low (VL)	-200 hingga -60	Trapezoidal
Low (L)	-120 hingga -20	Trapezoidal
Medium (M)	-60 hingga 0	Trapezoidal
Zero (Z)	-20 hingga 20	Trapezoidal
Positive (P)	0 hingga 60	Trapezoidal
High (H)	20 hingga 120	Trapezoidal
Very High (VH)	60 hingga 200	Trapezoidal



Gambar 3. 2 Visualisasi Fungsi Keanggotaan Fuzzy

Membership function (MF) trapezoidal dipilih karena:

- **Kesederhanaan:** Fungsi trapezoidal memiliki bentuk sederhana yang efisien untuk komputasi pada sistem embedded seperti Arduino.
- **Fleksibilitas:** Bentuk trapezoidal memungkinkan representasi berbagai tingkat ketidakpastian dalam pengendalian.
- **Penerapan Kontrol:** Trapezoidal MF ideal untuk menyatakan transisi gradual antar kondisi seperti "Low," "Medium," dan "High," yang sesuai dengan sifat non-linier kontrol yaw.
- **Linearitas Lokal:** Segmentasi liniernya mudah dihitung, sehingga cepat dalam implementasi real-time.

Contoh:

- Untuk yaw error kecil, sudut rudder dapat disesuaikan secara halus (zona transisi "Zero").
- Untuk yaw error besar, langkah rudder harus lebih agresif (zona "Very High").

3.1.3 Proses Menggunakan Fuzzy dan Defuzzyfikasi

Langkah-Langkah:

- Fuzzifikasi Input: Input (yaw error, misalnya -50) dievaluasi pada semua fungsi keanggotaan (VL, L, M, Z, P, H, VH).
- Inferensi Fuzzy: Semua aturan fuzzy diaplikasikan untuk menghasilkan output linguistik (misalnya "Medium Negative").
- Defuzzifikasi: Nilai crisp dihitung berdasarkan output fuzzy untuk menentukan jumlah langkah motor stepper. Jika hasilnya -45, motor stepper akan bergerak ke kiri dengan 45 langkah.

3.2 Bahan dan peralatan yang digunakan

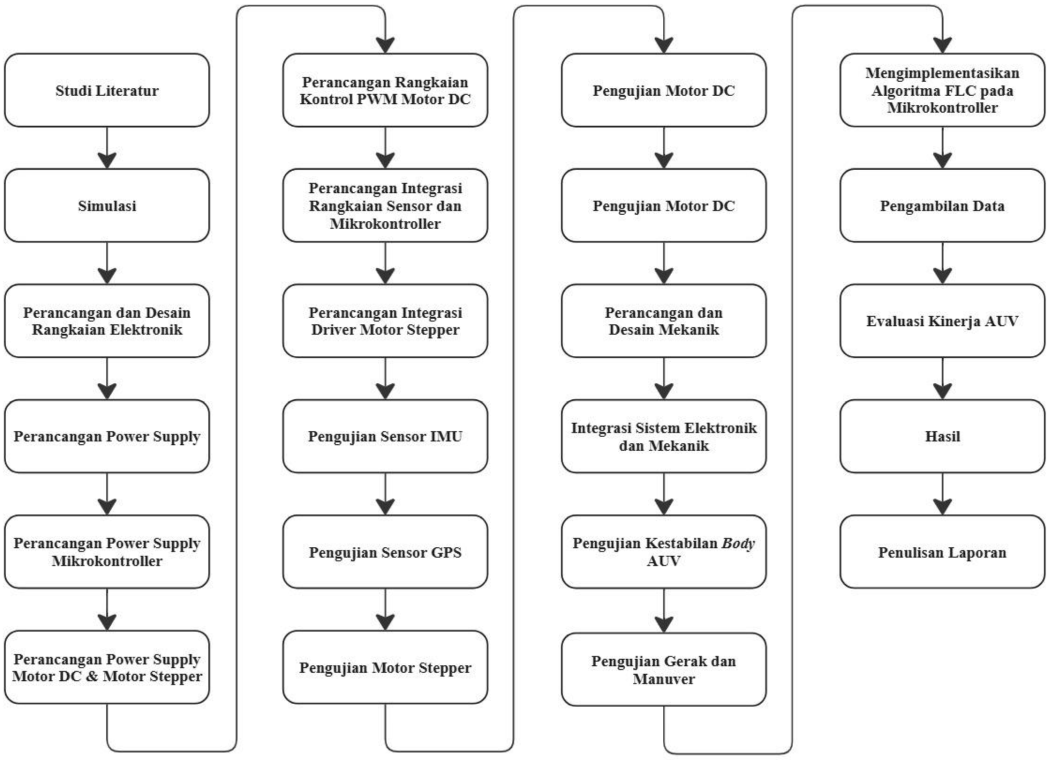
Dalam melakukan rancang bangun atau implementasi, diperlukan beberapa alat dan bahan yang digunakan untuk menunjang keberhasilan pembuatan sistem kendali pada AUV, berikut ini merupakan beberapa alat dan bahan yang diperlukan:

a. Alat :
   1. Mikrokontroller Arduino Mega
      Mikrokontroller yang digunakan untuk rancang bangun adalah Arduino Mega
   2. Perangkat untuk simulasi
      Perangkat lunak untuk simulasi desain sistem adalah MATLAB
   3. Sistem Gerak
      Sistem gerak menggunakan propulsi dan thruster
   4. Prototipe Board
      Prototipe board menggunakan PCB untuk sirkuit elektronika yang telah dirancang
   5. Oscilloscope
      Digunakan untuk menganalisis sinyal respon sistem
   6. Avometer
      Digunakan untuk pengukuran tegangan, arus, serta hambatan
   7. Peralatan Mekanikal lainnya
      Peralatan Meknikal seperti obeng, tang, dll, juga diperlukan untuk menunjang perakitan.

b. Bahan :
   1. Material Mekanikal dan Konstruksi
      Menggunakan material tahan air dan ringan serta tahan terhadap korosi. Peneliti berencana menggunakan aluminium dari modifikasi torpedo.
   2. Kabel
      Menggunakan kabel untuk melakukan perakitan antar komponen
   3. Perangkat isolasi
      Perangkat isolasi digunakan untuk melindungi komponen elektronika dari air
   4. Komponen elektronika
      Komponen elektronika terdiri dari sensor dan actuator yang digunakan untuk kendali pada AUV
   5. Komponen mekanikal
      Diperlukan juga komponen mekanikal seperti mur, baut, dll.

3.3 Urutan pelaksanaan penelitian

Penelitian mengenai Autonomous Underwater Vehicle (AUV) ini bertujuan untuk mengembangkan dan menguji sistem yang mampu beroperasi secara mandiri di bawah air. Untuk mencapai tujuan tersebut, pelaksanaan penelitian dilakukan secara sistematis dan terstruktur, dimulai dari perancangan konseptual hingga pengujian akhir di lingkungan nyata. Langkah-langkah yang akan dijelaskan berikut mencakup seluruh proses penelitian. Untuk memberikan gambaran yang lebih jelas mengenai alur kerja dan struktur sistem yang digunakan dalam penelitian ini, berikut disajikan sebuah diagram blok. Diagram ini menggambarkan hubungan antara komponen utama, baik perangkat keras maupun perangkat lunak, serta bagaimana data diproses dan dikomunikasikan dalam sistem Autonomous Underwater Vehicle (AUV). Penjelasan lebih rinci mengenai setiap komponen akan dijabarkan setelah diagram blok berikut ini.



Gambar 3. 3 Diagram Blok Urutan Pelaksanaan Penelitian AUV

3.3.1 Studi Literatur

Study Literatur dilakukan untuk mendapatkan referensi teori yang relevan dengan permasalahan pada tugas akhir ini dan dibandingkan dengan hasil uji coba lapangan. Sekaligus membandingkan hasil dari controller kecepatan dengan metode PID dan Adaptive Fuzzy Logic Controller. Studi literatur juga untuk menunjang keberhasilan implementasi AUV yang nantinya akan dilakukan perancangan serta pengujian. Kegiatan studi literatur ini meliputi 2 kegiatan, yaitu:

1. Pengumpulan materi tinjauan pustaka yang berkaitan dengan perencanaan sistem kontrol Fuzzy Logic Controller, 6 derajat kebebasan pada AUV , sensor yang digunakan, spesifikasi aktuator. Sumber literatur yang digunakan adalah jurnal internasional, jurnal nasional, text book, dan tugas akhir yang masih berhubungan.
2. Melakukan literatur yang berkaitan dengan simulasi dan permodelan sistem AUV.

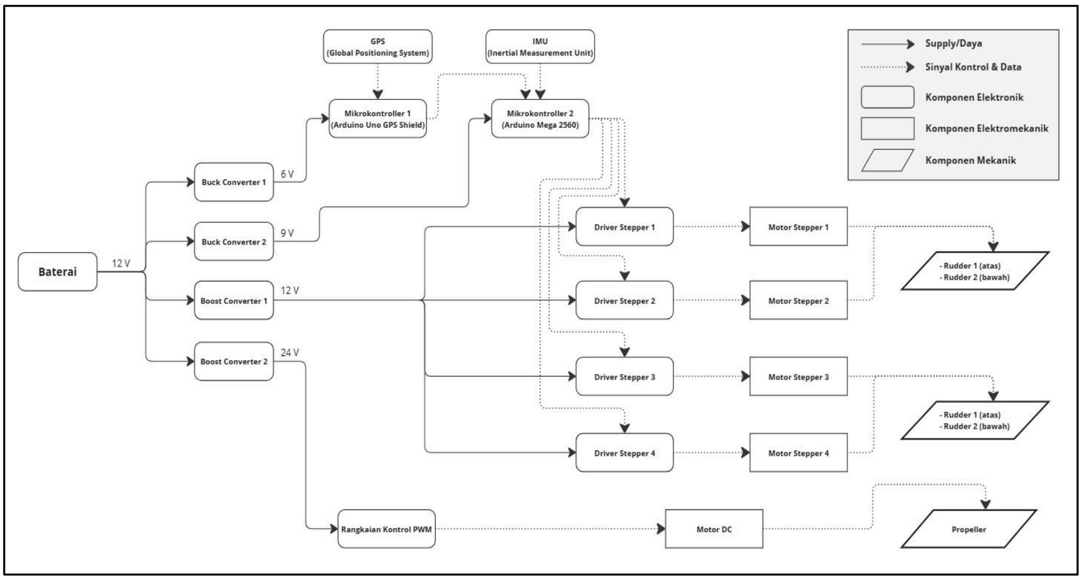
3.3.2 Simulasi Sistem AUV

Simulasi sistem Autonomous Underwater Vehicle (AUV) dilakukan sebagai langkah awal untuk menguji dan mengevaluasi kinerja algoritma fuzzy logic controller sebelum diterapkan pada sistem fisik. Simulasi ini memiliki peran penting dalam memastikan bahwa rancangan algoritma pengendalian dapat berfungsi dengan baik sesuai dengan tujuan, tanpa risiko terhadap perangkat keras atau kendala operasional di lapangan. Dalam simulasi ini, algoritma fuzzy logic controller diterapkan pada model matematis AUV yang mencakup aspek-aspek utama seperti dinamika gerak, gaya hidrodinamika, dan pengaruh lingkungan. Model dinamika AUV yang digunakan dalam simulasi menggambarkan hubungan antara gaya dan momen hidrodinamika dengan gerakan translasi serta rotasi. Simulasi dirancang untuk merepresentasikan kondisi lingkungan nyata, termasuk perubahan arus air, gelombang, dan

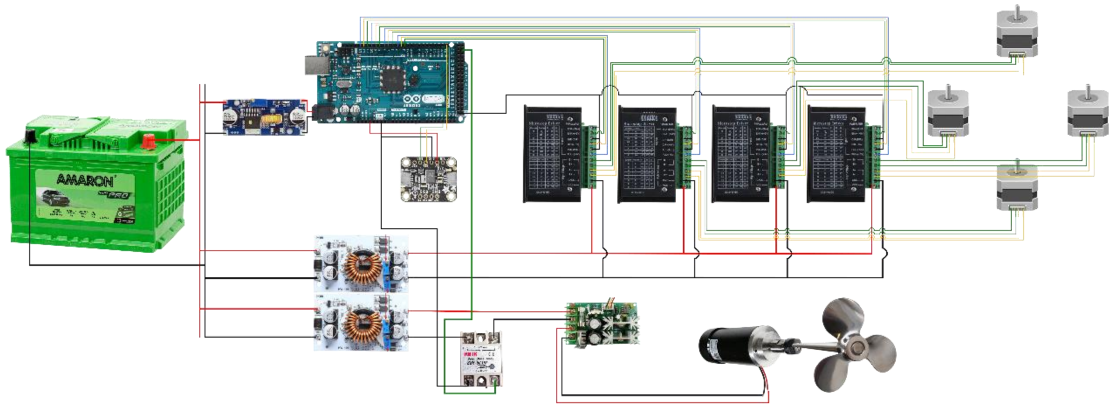
turbulensi. Berbagai skenario pengujian juga disimulasikan, seperti perubahan kedalaman, arah gerak, serta respons AUV terhadap gangguan eksternal. Algoritma fuzzy logic controller diterapkan dengan memanfaatkan aturan berbasis logika linguistik untuk mengatur thrust motor dan arah gerak rudder. Selama simulasi, data posisi, kecepatan, sudut orientasi, serta thrust yang dihasilkan algoritma fuzzy logic controller dikumpulkan untuk dianalisis. Analisis ini mencakup pengamatan terhadap respons transien, kestabilan sistem, serta tingkat kesalahan steady-state dalam mempertahankan jalur gerak yang diinginkan. Hasil simulasi memberikan gambaran awal mengenai efektivitas algoritma dalam mengelola gerakan AUV di bawah berbagai kondisi lingkungan. Simulasi ini memberikan manfaat yang signifikan dalam proses pengembangan AUV. Dengan menggunakan hasil simulasi, parameter algoritma fuzzy logic dapat dioptimalkan, potensi kesalahan dapat diidentifikasi, dan strategi pengendalian dapat disempurnakan. Selain itu, simulasi juga membantu memprediksi performa sistem di lingkungan kompleks, sehingga dapat meminimalkan risiko kesalahan saat dilakukan implementasi langsung. Simulasi ini menjadi dasar yang penting untuk memastikan keberhasilan penerapan algoritma fuzzy logic controller pada tahap pengujian di lingkungan nyata.

3.3.3 Desain Rangkaian Elektronik AUV

Tahap desain kebutuhan rangkaian elektronik ditentukan berdasarkan spesifikasi teknis dari AUV yang telah disimulasikan. Mulai dari kebutuhan power supply, sensor, dan aktuator. Pemilihan komponen Arduino Mega 2560 dikarenakan memiliki 12 pin PWM yang dapat digunakan untuk melakukan kontrol kepada keempat driver stepper. Dimana setiap stepper membutuhkan 3 pin yang terpakai pada mikrokontroller Arduino Mega 2560. Sehingga total membutuhkan 12 pin untuk mengontrol 4 driver motor stepper. Kemudian menggunakan 2 pin untuk IMU berkomunikasi dengan Arduino Mega. Pin tersebut adalah pin SDA, dan SCL. Karena sensor IMU menggunakan komunikasi I2C, maka dari itu peneliti menggunakan pin SDA dan SCL pada Arduino Mega. Kemudian 1 pin digital untuk mengontrol ON/OFF motor DC yang nantinya akan menjadi penggerak utama propeller.



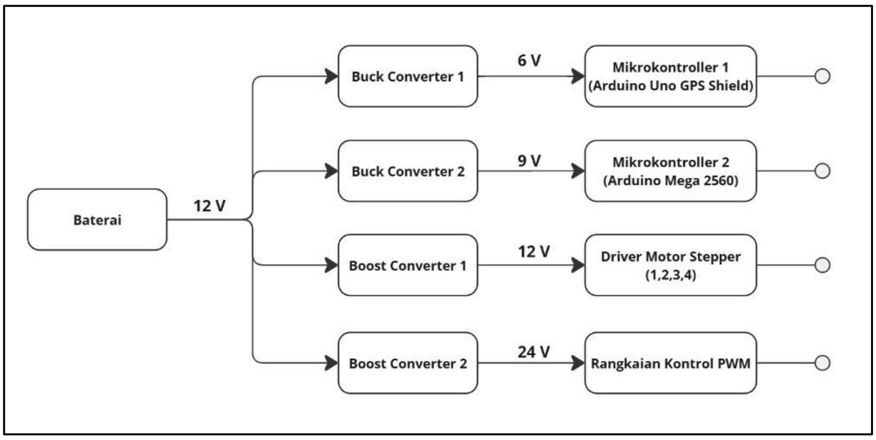
Gambar 3. 4 Diagram Blok Hardware AUV



Gambar 3. 5 Desain Rangkaian Elektronik AUV

3.3.4 Perancangan Power Supply

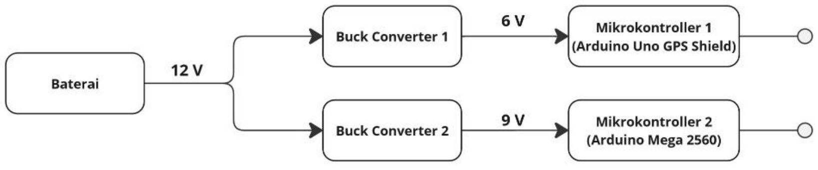
Pada tahap ini, peneliti membuat rancangan power supply untuk masing-masing pembebanan, yakni ada empat *converter* yang digunakan untuk menstabilkan daya kepada masing-masing beban. Berikut adalah diagram blok Power Supply yang telah dirancang oleh peneliti :



Gambar 3. 6 Diagram blok Power Supply ke Seluruh Beban

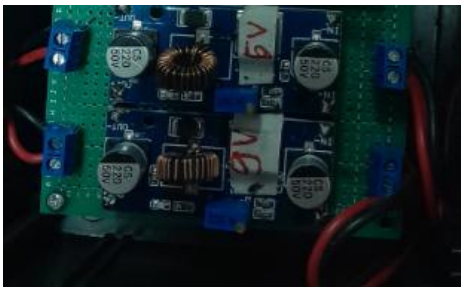
3.3.5 Perancangan Power Supply untuk Mikrokontroller

Power Supply dibutuhkan untuk memberikan catu daya pada mikrokontroller ketika keseluruhan sistem elektronik AUV berjalan secara otonom. Karena peneliti menggunakan 2 mikrokontroller, maka peneliti membuat dua rangkaian catu daya, yang pertama adalah 6V dan yang kedua adalah 9V. Berikut ini merupakan diagram blok rancangan power supply untuk mikrokontroller :



Gambar 3. 7 Diagram blok Power Supply Mikrokontroller

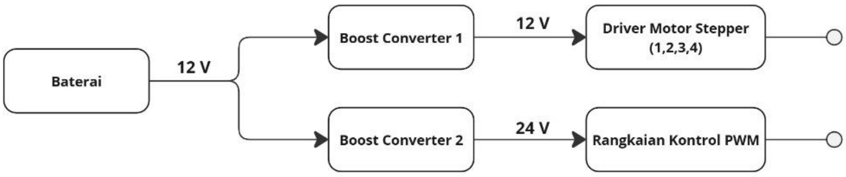
Berikut adalah hasil perakitan power supply untuk mikrokontroller :



Gambar 3. 8 Rangkaian Power Supply untuk Mikrokontroller

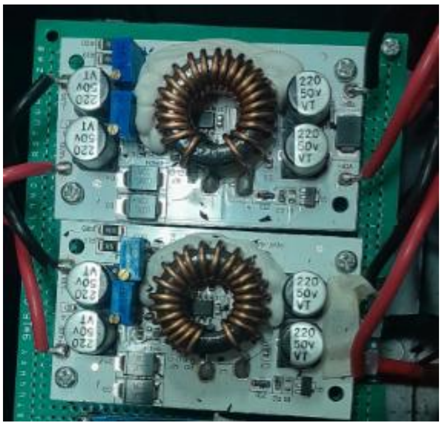
3.3.6 Perakitan Power Supply Motor DC dan Motor Stepper

Power Supply juga diberikan untuk motor dc dan juga motor stepper, pada motor DC, digunakan rangkaian power supply dengan output step-up dari 12VDC (Aki) menjadi 24VDC. Motor stepper juga akan di stabilkan dengan rangkaian power supply 12VDC ke 12VDC. Tujuan dibuatkan rangkaian penstabil tegangan ini adalah untuk menjaga kinerja motor DC dan juga keempat motor stepper yang digunakan untuk 2 Rudder dan 2 Stern, dimana Rudder dan Stern ini akan digunakan untuk manuver AUV.



Gambar 3. 9 Diagram blok Power Supply Motor DC dan Motor Stepper

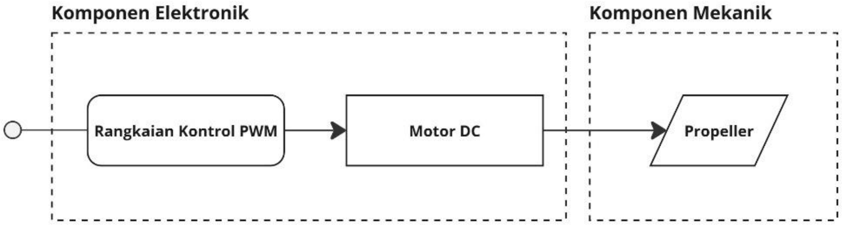
Berikut adalah hasil rangkaian penstabil tegangan Motor DC dan Motor Stepper:



Gambar 3. 10 Rangkaian Power Supply Motor DC dan Motor Stepper

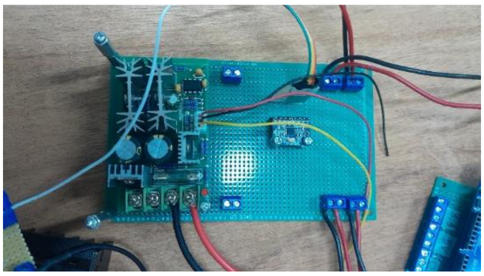
3.3.7 Perancangan Rangkaian Kontrol PWM Motor DC

Rangkaian PWM motor DC digunakan untuk mengontrol RPM dari motor dc yang akan digunakan untuk penggerak propeller. Berikut ini merupakan diagram blok rancangan rangkaian kontrol PWM untuk motor DC:



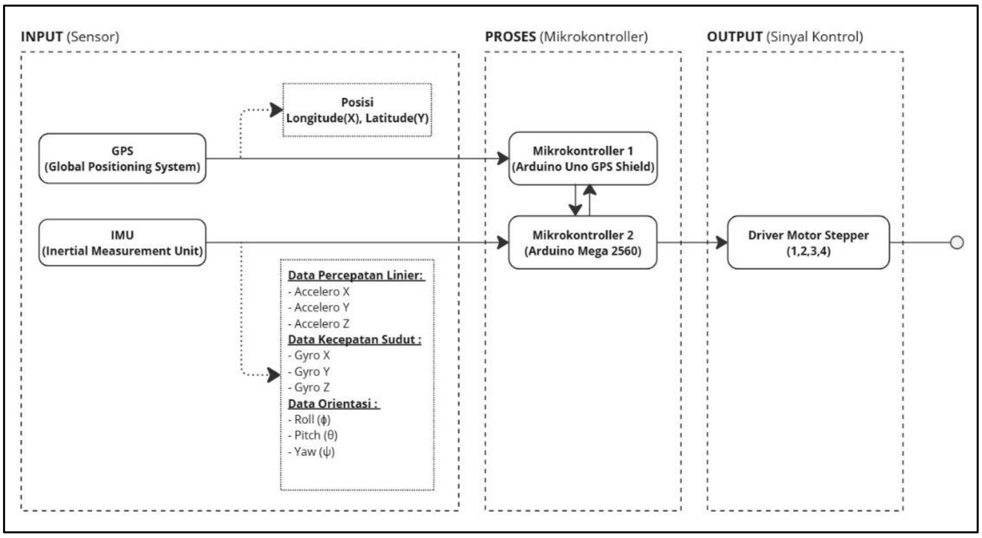
Gambar 3. 11 Diagram blok Rangkaian Kontrol PWM Motor DC

Berikut adalah hasil rangkaian dari PWM DC Motor Control yang dibuat oleh peneliti.


Gambar 3. 12 Rangkaian Kontrol PWM Motor DC

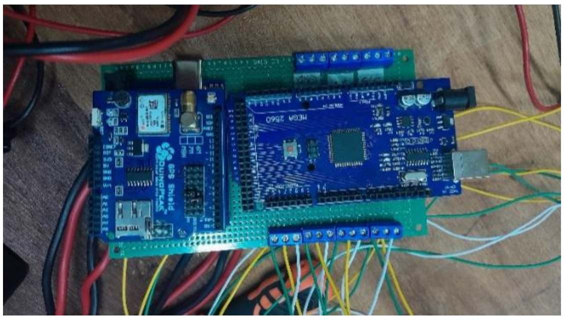
3.3.8 Perancangan Integrasi Rangkaian Sensor dan Mikrokontroller

Pada penelitian ini, menggunakan dua jenis mikrokontroller yang saling terintegrasi. Mikrokontroller pertama, digunakan untuk akuisisi data dan pemrosesan. Mikrokontroller kedua digunakan untuk memberikan sinyal kontrol ke motor. Berikut adalah diagram blok hasil integrasi kedua mikrokontroller yang dibuat oleh peneliti:



Gambar 3. 13 Diagram blok Integrasi Sensor dan Mikrokontroller

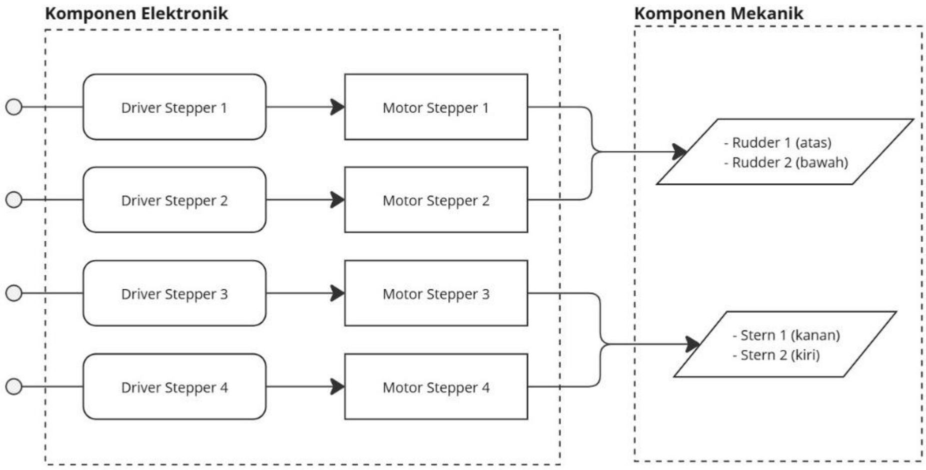
Berikut adalah hasil integrasi kedua mikrokontroller yang dibuat oleh peneliti:



Gambar 3. 14 Rangkaian Integrasi Sensor dan Mikrokontroller

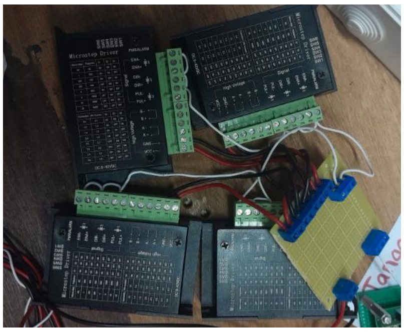
3.3.9 Perancangan Integrasi Driver Motor Stepper

Mengintegrasikan keempat driver stepper TB6600 dalam 1 papan untuk mempermudah dalam melakukan koordinator kontrol manuver pada rudder dan stern untuk pergerakan Yaw dan Pitch. Berikut ini merupakan diagram blok rancangan rangkaian integrasi empat driver stepper TB6600:



Gambar 3. 15 Diagram blok Integrasi Driver TB6600 ke Motor Stepper

Berikut adalah hasil rangkaian integrasi empat driver stepper TB6600 yang dibuat oleh peneliti:

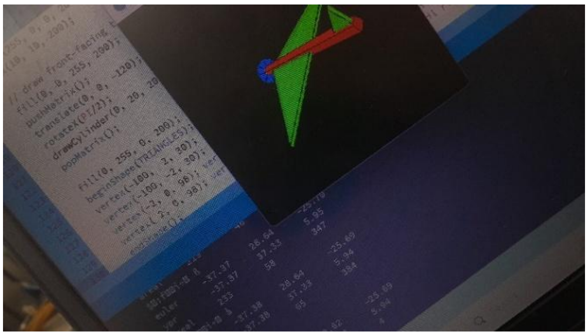


Gambar 3. 16 Rangkaian Integrasi Empat Driver Stepper

3.3.10 Pengujian Sensor IMU

Pengujian Sensor IMU dilakukan dengan menggunakan software Arduino untuk melakukan programming. Kemudian menggunakan software Processing 4 untuk melihat respon

dari sensor IMU. Berikut adalah dokumentasi pengujian sensor IMU menggunakan software Processing 4.



Gambar 3. 17 Pengujian sensor IMU dengan Processing 4

3.3.11 Pengujian Motor Stepper

Pengujian Motor stepper dilakukan untuk mengetahui keluaran sinyal yang diberikan dalam 1 waktu pada beberapa variasi microstepping. Berikut adalah proses pengujian sinyal keluaran dari driver yang diberikan kepada motor stepper.

Gambar 3. 18 Pengujian Driver dan Motor Stepper

3.3.12 Pengujian Motor DC

Pengujian Motor DC digunakan untuk mengetahui RPM dari motor DC pada kondisi sebelum gearbox dan juga sesudah gearbox. Berikut adalah dokumentasi pengujian motor dc yang akan digunakan untuk penggerak propeller.

Gambar 3. 19 Pengujian Motor DC untuk Propeller

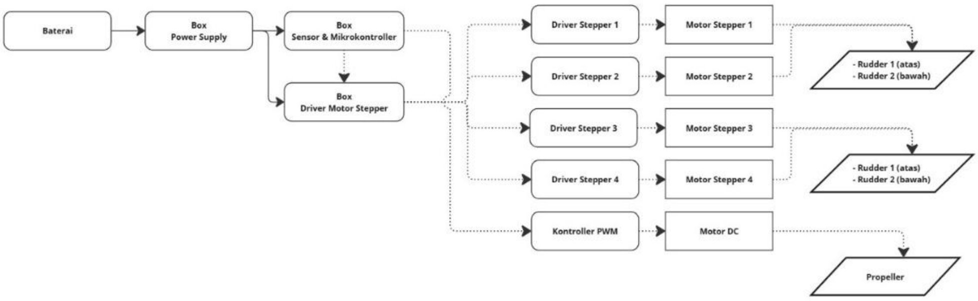
3.3.13 Perancangan dan Desain Mekanik

Pada tahap perancangan dan desain mekanik, dilakukan dengan menggunakan software Fusion 360.

Gambar 3. 20 Rancang Bangun Mekanik

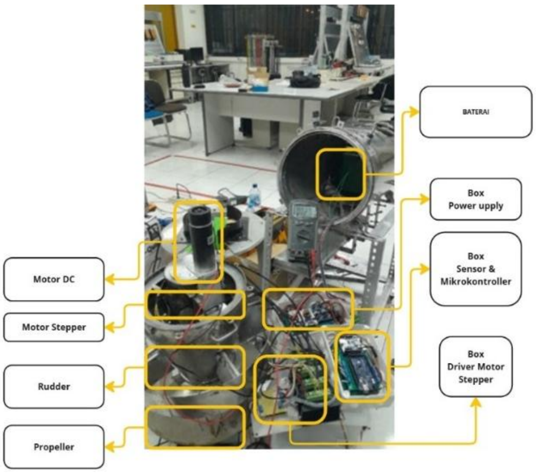
3.3.14 Integrasi Elektronik dan Mekanik

Pada subbab ini, akan dijelaskan proses integrasi antara sistem elektronik dan mekanik dalam perancangan Autonomous Underwater Vehicle (AUV). Integrasi ini merupakan tahap krusial yang memastikan seluruh komponen, baik perangkat keras elektronik seperti sensor, aktuator, dan kontroler, maupun perangkat mekanik seperti rangka, propulsi, dan sistem hidrodinamik, dapat berfungsi secara harmonis. Dengan pendekatan yang sistematis, integrasi ini bertujuan untuk mengoptimalkan performa AUV dalam menjalankan tugas-tugas yang telah dirancang, baik di lingkungan simulasi maupun dalam uji coba di bawah air. Untuk mendukung penjelasan pada subbab ini, berikut disajikan sebuah diagram blok yang menggambarkan struktur dari elektronik dan mekanik AUV. Berikut adalah diagram blok integrasi elektronik dan mekanik AUV:



Gambar 3. 21 Diagram Blok Integrasi Elektronik dan Mekanik AUV

Dan disajikan pula dokumentasi ketika melakukan integrasi sistem elektronik dan mekanik. Berikut adalah dokumentasi proses integrasi elektronik dan mekanik AUV:



Gambar 3. 22 Integrasi Rangkaian Elektronik dan Mekanik

Pada proses integrasi elektronik dan mekanik, dipastikan semua sambungan kabel harus kuat agar ketika AUV diuji, tidak ada kegagalan pada sistem elektronik maupun mekanik.

3.3.15 Pengujian Kestabilan *Body*

Pengujian kestabilan body dilakukan untuk menguji apakah posisi AUV pada kondisi tenggelam, mengapung, atau melayang di dalam air. Karena yang diharapkan oleh peneliti adalah posisi AUV berada pada kondisi melayang di dalam air.

Gambar 3. 23 Pengujian Kestabilan Body AUV

3.3.16 Pengujian Gerak dan Manuver

Pengujian gerak dan manuver bertujuan untuk memastikan bahwa sistem mekanik pada AUV mampu berfungsi dengan baik sebelum algoritma kontrol fuzzy diterapkan. Pengujian ini dilakukan secara manual dengan mengamati kinerja rudder yang digerakkan oleh motor stepper dalam kondisi tekanan hidrodinamika air. Hasil dari pengujian ini akan menjadi dasar evaluasi awal apakah sistem mekanik dapat memenuhi kebutuhan stabilitas dan manuver AUV.

3.3.17 Implementasi Algoritma Fuzzy Logic Controller

Setelah memastikan bahwa sistem mekanik bekerja dengan baik, langkah selanjutnya adalah merancang dan mengimplementasikan algoritma fuzzy logic controller pada mikrokontroler. Algoritma ini dirancang untuk memungkinkan AUV beroperasi secara otonom, dengan memanfaatkan data dari sensor untuk mengambil keputusan pengendalian. Proses implementasi mencakup pengkodean algoritma, pengujian perangkat lunak, dan integrasi dengan perangkat keras AUV.

3.3.18 Pengambilan Data Sensor

Pengambilan data sensor dilakukan setelah pengujian mekanik dan elektronik berhasil. Pada tahap ini, AUV dioperasikan di lingkungan nyata untuk mengumpulkan data seperti posisi, kecepatan, sudut orientasi (roll, pitch, yaw), dan parameter lingkungan. Dalam penelitian ini, pengujian dilakukan di Laboratorium Hidrodinamika milik Badan Riset dan Inovasi Nasional, yang menyediakan kondisi lingkungan terkontrol untuk pengujian hidrodinamika.

3.3.19 Evaluasi Kinerja

Tahap evaluasi kinerja dilakukan untuk mengukur sejauh mana algoritma fuzzy logic controller mampu mengendalikan AUV secara efektif di lingkungan nyata. Pengujian ini mencakup analisis respons sistem terhadap berbagai skenario seperti perubahan arah, kecepatan, dan tekanan hidrodinamika. Hasil evaluasi digunakan untuk menentukan apakah sistem pengendalian memenuhi kriteria performa yang telah ditentukan.

3.3.20 Hasil

Tahap ini mendokumentasikan hasil pengujian sistem, termasuk analisis kinerja kontroler fuzzy dan rancang bangun AUV. Data yang diperoleh dianalisis untuk

mengidentifikasi kelebihan, kekurangan, dan potensi perbaikan. Jika hasil tidak sesuai dengan harapan, perbaikan akan dilakukan berdasarkan temuan dari dokumentasi pengujian.

3.3.21 Penulisan Laporan

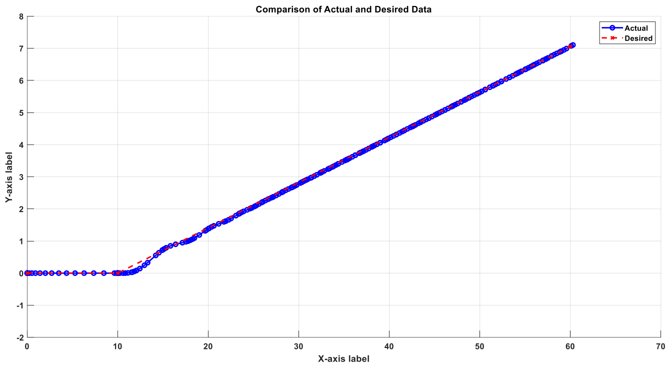
Langkah terakhir dalam penelitian ini adalah penulisan laporan yang mencakup seluruh proses penelitian, dari perancangan hingga pengujian. Laporan ini meliputi data hasil percobaan, analisis kinerja, dokumentasi proses pengujian, dan kesimpulan. Laporan ini bertujuan untuk menyediakan dokumentasi menyeluruh sebagai acuan untuk pengembangan lebih lanjut.

(Halaman Ini Sengaja Dikosongkan)

BAB 4 HASIL DAN PEMBAHASAN

4.1 Hasil Pengujian dan Pembahasan

4.1.1 Hasil Simulasi Pada Lintasan

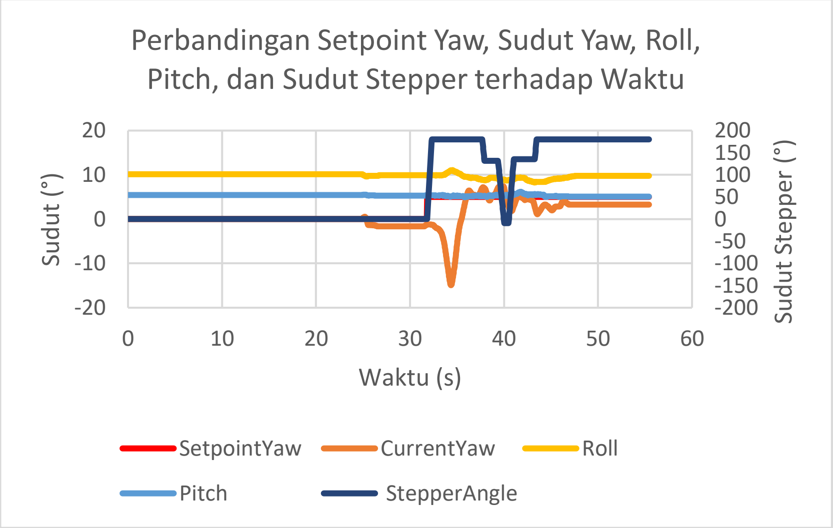


Gambar 4. 1 Hasil Simulasi AUV pada Lintasan 10°

Gambar 4.1 menunjukkan perbandingan antara jalur aktual dan jalur yang diharapkan. Dari hasil pengujian, dapat dilihat bahwa data aktual hampir sepenuhnya mengikuti data yang diharapkan, yang ditunjukkan oleh tumpang tindih yang erat antara kedua garis pada grafik. Meskipun demikian, nilai RMSE (*Root Mean Square Error*) rata-rata yang mencapai 0.081502 menunjukkan adanya deviasi yang minimum antara nilai aktual dan nilai yang diharapkan secara numerik. Berdasarkan hasil dari simulasi pada lintasan dengan kemiringan 10° dan kecepatan dari 0 hingga 4.4 m/s, sistem mencapai nilai akhir yang diinginkan dalam waktu 17.4 detik.

4.1.2 Hasil Pengujian Manuver Yaw 5°

Pada pengujian ini, AUV diberikan setpoint sebesar 5° dengan kecepatan propeller konstan senilai 300 RPM. Untuk mengamati kinerja sistem dalam melakukan manuver Yaw, data yang diperoleh dari pengujian telah diolah dan divisualisasikan. Grafik yang dihasilkan memberikan gambaran dinamis mengenai perubahan parameter utama sistem terhadap waktu, seperti “SetpointYaw”, ”CurrentYaw”, ”Roll”, ”Pitch”, dan ”StepperAngle”. Berikut adalah grafik hasil pengujian yang telah didapat:



Gambar 4. 2 Grafik Respon Sudut Yaw, Roll, Pitch, Stepper pada Setpoint 5°

Grafik (4.2) menggambarkan bagaimana sistem kontrol yaw beroperasi dengan memantau dan mengatur beberapa parameter utama, yaitu **SetpointYaw**, **CurrentYaw**, **Roll**, **Pitch**, dan **StepperAngle**, terhadap waktu. Sumbu horizontal menunjukkan waktu, sementara sumbu vertikal merepresentasikan nilai-nilai parameter tersebut. Pada grafik ini, **SetpointYaw** yang ditunjukkan oleh garis merah mencerminkan target yaw sistem. Sepanjang waktu, nilainya tetap stabil, menandakan bahwa sistem diharapkan mempertahankan sudut yaw tertentu tanpa fluktuasi. Hal ini menunjukkan tujuan sistem adalah menjaga stabilitas yaw dengan konsistensi yang tinggi. **CurrentYaw**, yang direpresentasikan oleh garis oranye, menunjukkan nilai yaw aktual yang diukur oleh sistem. Pada awalnya, CurrentYaw terlihat stabil, tetapi mulai mengalami perubahan ketika sistem mendeteksi yaw error, yaitu perbedaan antara nilai aktual dan nilai target. Proses koreksi yang dilakukan oleh sistem menyebabkan fluktuasi yang signifikan, termasuk overshoot, saat sistem berusaha dengan cepat mendekati nilai SetpointYaw. Setelah proses koreksi selesai, CurrentYaw akhirnya menjadi stabil dan mendekati nilai SetpointYaw. Parameter **Roll** dan **Pitch**, yang masing-masing ditunjukkan oleh garis kuning dan biru, menggambarkan orientasi sistem dalam sumbu lainnya. Sepanjang grafik, nilai Roll dan Pitch relatif stabil dengan sedikit fluktuasi, yang menunjukkan bahwa sistem mampu menjaga keseimbangan orientasi meskipun sedang melakukan koreksi yaw yang intens. **StepperAngle**, yang direpresentasikan oleh garis hitam, mencerminkan sudut motor stepper yang digunakan untuk melakukan koreksi yaw error. Sudut ini berubah secara signifikan selama proses koreksi, terutama saat yaw error besar, mencerminkan respons aktif dari sistem untuk mengurangi error tersebut. Setelah CurrentYaw mendekati SetpointYaw, StepperAngle kembali stabil, menandakan bahwa sistem tidak lagi membutuhkan koreksi besar.

Secara keseluruhan, grafik ini mencerminkan tiga fase utama dalam proses pengendalian yaw. **Fase Awal (Initial Phase)**: Sistem dimulai dalam keadaan stabil, di mana CurrentYaw mendekati SetpointYaw. Tidak ada perubahan signifikan pada StepperAngle, karena yaw error masih berada dalam batas toleransi. **Fase Koreksi (Correction Phase)**: Ketika perbedaan antara SetpointYaw dan CurrentYaw meningkat, sistem mendeteksi error dan memulai proses koreksi. Hal ini ditandai dengan perubahan besar pada StepperAngle dan fluktuasi CurrentYaw. Fluktuasi ini menunjukkan respons sistem dalam mengurangi yaw error, meskipun ada overshoot sebelum akhirnya mendekati nilai target. **Fase Stabilitas (Steady-State Phase)**: Setelah koreksi selesai, CurrentYaw stabil dan mendekati SetpointYaw. StepperAngle juga kembali konstan, mencerminkan bahwa sistem telah mencapai kondisi stabil. Selama fase ini, Roll dan Pitch tetap stabil, menunjukkan bahwa sistem berhasil menjaga orientasi yang baik selama proses pengendalian yaw. Grafik ini menunjukkan bagaimana sistem kontrol yaw mampu merespons secara efektif terhadap perubahan dan mencapai stabilitas akhir, dengan menjaga keseimbangan orientasi meskipun terdapat dinamika koreksi.

Sementara itu, tabel hasil analisis merangkum metrik kinerja yang mencakup waktu respons awal, *overshoot*, waktu error *steady-state*, serta *error steady-state*, yang penting untuk mengevaluasi efektivitas dan efisiensi sistem dalam mencapai target yaw. Berikut adalah hasil data yang didapat dalam bentuk tabel untuk mengamati kinerja dari AUV:

Tabel 4. 1 Metrik Kinerja Sistem Kontrol AUV pada Setpoint Yaw 5°

Parameter	Nilai	Satuan
Rise Time	10.75	s
Settling Time	41.836	s
Steady-State Error	1.75	°
Overshoot (%)	47.6	%
Overshoot Time	39.899	s
Overshoot Value	2.38	°
Integral Absolute Error (IAE)	80.42986	-
Integral Squared Error (ISE)	554.981386	-
Mean Square Error (MSE)	9.758474741	-
Root Mean Squared Error (RMSE)	3.123855749	-

Tabel ini menunjukkan hasil analisis kinerja sistem kontrol yaw berdasarkan beberapa metrik penting. Sistem memiliki waktu respons awal rise time sebesar 10.75 detik, yang mencerminkan respons awal sistem yang relatif lambat. Waktu yang dibutuhkan untuk mencapai stabilitas penuh (settling time) adalah 41.836 detik, menunjukkan bahwa sistem memerlukan waktu yang cukup lama untuk stabilisasi. Error steady-state sebesar 1.75° menunjukkan bahwa sistem memiliki akurasi yang cukup baik dalam mencapai target yaw. Overshoot sistem mencapai 47.6% dengan nilai maksimum sebesar 8.5°, yang terjadi pada 39.899 detik, menunjukkan respons koreksi yang cukup agresif. Selain itu, metrik error seperti Integral Absolute Error (IAE) sebesar 80.416 dan Integral Squared Error (ISE) sebesar 554.96 menunjukkan akumulasi error yang cukup besar selama proses. Root Mean Squared Error (RMSE) sebesar 3.12 memberikan gambaran rata-rata deviasi sistem dari target yaw. Secara keseluruhan, sistem menunjukkan respons steady-state yang baik dan akurasi yang memadai, tetapi membutuhkan optimisasi pada pengurangan overshoot, waktu respons awal, dan waktu stabilisasi agar lebih efisien dan presisi.

4.1.2.1 Analisis Data dan Perhitungan untuk manuver *5°*

Untuk menganalisis data pada variasi manuver dengan setpoint Yaw sebesar 5 derajat, dilakukan evaluasi terhadap respons sistem menggunakan metode fuzzy logic. Analisis ini bertujuan untuk memahami kinerja sistem dalam mengoreksi yaw error berdasarkan data pengujian, meliputi tahap fuzzifikasi, inferensi, hingga defuzzifikasi. Selain itu, parameter seperti kehalusan koreksi, respons progresif, dan keandalan sistem akan dievaluasi untuk menentukan efektivitas pendekatan fuzzy logic dalam mengatur pergerakan motor stepper. Penyesuaian pada fungsi keanggotaan dan aturan fuzzy juga akan dipertimbangkan untuk mencapai target koreksi sudut yang optimal. Peneliti mengambil sampel data sensor yang telah terbaca oleh mikrokontroller dan telah diproses secara realtime menggunakan Fuzzy Logic Controller.

Tabel 4. 2 Sampel Data Hasil Pengukuran untuk Setpoint Yaw 5°

Time (s)	Setpoint	Current Yaw	Yaw Error	Stepper Angle	Stepper Enabled
31.734	0	-1.31	1.31	0	OFF
31.758	0	-1.25	1.25	0	OFF
31.780	0	-1.25	1.25	0	OFF
31.803	0	-1.19	1.19	0	OFF
31.826	5	-1.19	6.19	0	OFF
31.864	5	-1.19	6.19	12.6	ON
31.898	5	-1.19	6.19	25.2	ON
31.933	5	-1.19	6.19	37.8	ON
31.965	5	-1.19	6.19	50.4	ON
31.999	5	-1.19	6.19	63	ON

Perhitungan pada sistem kontrol dimulai ketika Stepper Enabled berstatus "ON" atau ketika terjadi perubahan nilai Setpoint yang diterima oleh mikrokontroler. Pada tabel ini, perubahan Setpoint terjadi pada waktu 31.826 detik, dengan nilai Setpoint berubah dari 0 menjadi 5. Namun, pada waktu tersebut, Stepper Angle masih bernilai 0 dan Stepper Enabled masih berstatus "OFF". Pada iterasi berikutnya, yaitu pada waktu 31.864 detik, Stepper Angle mulai merespons dengan memberikan nilai sudut sebesar 12.6 derajat, dan Stepper Enabled berubah menjadi "ON". Hal ini menunjukkan bahwa sistem mulai aktif mengoreksi error yaw yang terdeteksi. Pada setiap langkah berikutnya, sistem menghasilkan keluaran kontrol pada Stepper Angle yang terus meningkat untuk mengoreksi Yaw Error hingga sistem mendekati target yaw yang diinginkan. Data ini menunjukkan bagaimana algoritma kontrol pada mikrokontroler memproses data secara real-time untuk mengatur pergerakan stepper motor dalam mengoreksi error yaw berdasarkan nilai setpoint yang diberikan.

Pada tabel (4.2), diambil 1 sampel Yaw Error untuk dilakukan perhitungan secara manual sesuai dengan teori Fuzzy Logic Controller yang telah diimplementasikan pada mikrokontroller. Berikut adalah langkah-langkah perhitungan matematis secara manual:

- Diketahui:
  - Input Yaw Error = 6.19°

1. **Fungsi Keanggotaan Fuzzy:**

Tabel 4. 3 Fungsi Keanggotaan Fuzzy

Kategori	*a*	*b*	*c*	*d*
Very Low (VL)	-200	-150	-120	-60
Low (L)	-120	-90	-60	-20
Medium (M)	-60	-40	-20	0
Zero (Z)	-20	-10	10	20
Positive (P)	0	20	40	60
High (H)	20	60	90	120
Very High (VH)	60	120	150	200

2. **Proses Fuzzifikasi:**

   Fuzzifikasi bertujuan untuk menghitung derajat kenaggotaan (μ) untuk input Yaw Error = 6.19°

- **Fuzzy set Z (Zero):**

$$\mu Z(x)=\begin{cases}\frac{x-a}{b-a}, & jika\ a \le x \le b\\ \frac{d-x}{d-c}, & jika\ c \le x \le d\\ 0, & lainnya\end{cases} \tag{4.1}$$

Substitusi nilai $a = -20, b = -10, c = 10, d = 20$;

$$\mu z(6.19) = \frac{20 - 6.19}{20 - 10} = \frac{13.81}{10} = 0.381 \tag{4.2}$$

- **Fuzzy set P (Zero):**

$$\mu P(x)=\begin{cases}\frac{x-a}{b-a}, & jika\ a \le x \le b\\ \frac{d-x}{d-c}, & jika\ c \le x \le d\\ 0, & lainnya\end{cases} \tag{4.3}$$

Substitusi nilai $a = 0, b = 20, c = 40, d = 60$;

$$\mu z(6.19) = \frac{6.19 - 0}{20 - 0} = \frac{6.19}{20} = 0.3095 \tag{4.4}$$

- **Fuzzy Set Lainnya** (VL, L, M, H, VH):

Karena $x = 6.19°$ berada di luar rentang fuzzy set lainnya, maka:

$$\mu = 0$$

- **Hasil Fuzzifikasi:**
  - $\mu_Z = 0.1381$
  - $\mu_P = 0.3095$
  - $\mu = 0$

3. **Inferensi:**

   Gunakan aturan linguistik:
   1. **If Z THEN Output = 0 steps**
   2. **IF P THEN Output = +60 steps**

   Kontribusi setiap aturan dihitung dengan:

   $$Kontribusi\ = \mu \cdot Output$$

   - **Kontribusi Z**:

     $\mu_Z \cdot O_Z = 0.1381 \cdot 0 = 0$
   - **Kontribusi P**:

     $\mu_P \cdot O_P = 0.3095 \cdot 60 = 18.57$

4. **Defuzzifikasi:**

   Pada proses defuzzifikasi, digunakan metpde *Weighted Average:*

   $$Output\ Crisp = \frac{\sum(\mu_i \cdot O_i)}{\sum \mu_i} \tag{4.5}$$

   Substitusi nilai:

   $$Output\ Crisp = \frac{(0.1381 \cdot 0) + (0.3095 \cdot 60)}{0.1381 + 0.3095} \tag{4.6}$$

   $$Output\ Crisp = \frac{(0) + 18.57}{0.44476} = 41.5° \tag{4.7}$$

5. **Hasil Akhir**

Hasil output crisp bernilai 41.5° untuk mengurangi yaw error 6.19°

6. **Menghitung Jumlah Langkah**

Untuk menghitung Output Langkah atau *Step* dari motor stepper berdasarkan hasil fuzzy logic (41.5°), kita memerlukan informasi tentang:

1. Jumlah langkah per revolusi stepper: $N_{steps}/_{rev}$
2. Jumlah derajat per putaran penuh: $D_{rev} = 360°$

Peneliti menggunakan konfigurasi 400 step pada algoritma program di sistem mikrokontroller

$$N_{steps}/_{rev} = 400\ steps/rev$$

Langkah ini setara dengan $D_{step}(derajat\ per\ langkah)$ dihitung sebagai:

$$D_{step} = \frac{D_{rev}}{N_{steps}/_{rev}}$$

$$D_{step} = \frac{360°}{400} = 0.9°$$

Jumlah langkah yang dibutuhkan dhitung sebagai:

$$Jumlah\ Langkah = \frac{Output\ Crisp}{D_{step}} \quad (4.8)$$

Substitusi nilai:

$$Jumlah\ Langkah = \frac{41.5°}{0.9°} = 46.11\ Langkah \quad (4.9)$$

Karena jumlah langkah harus berupa bilangan bulat, nilai ini dibulatkan menjadi 46 Langkah

4.1.2.2 Pembahasan Error Pengukuran dengan Perhitungan Yaw 5°

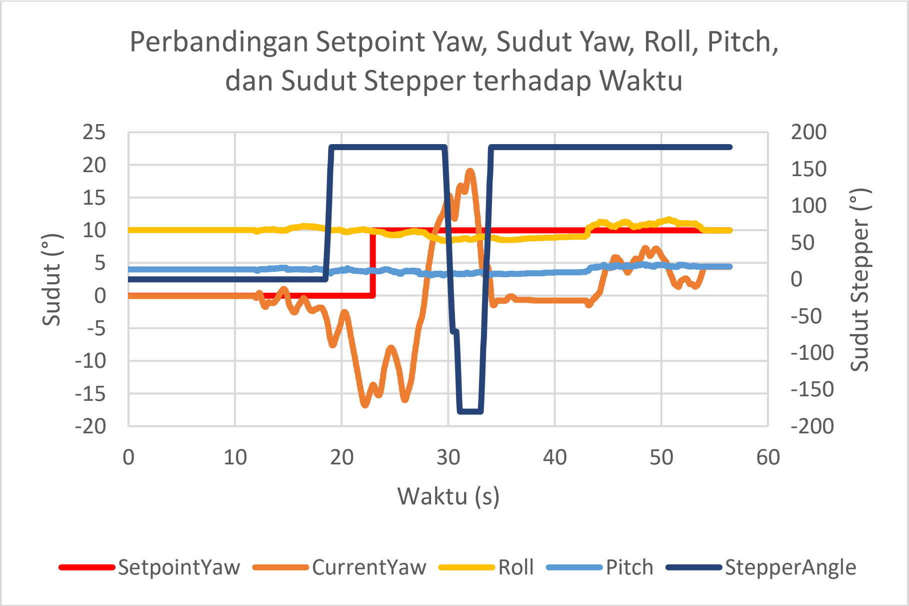
Perbedaan antara hasil perhitungan Crisp Output dan nilai yang tercatat pada tabel dapat terjadi karena beberapa faktor yang berhubungan dengan karakteristik sistem kontrol dan perangkat keras yang digunakan. Salah satu faktor utamanya adalah **respons incremental** dari motor stepper. Motor stepper bergerak secara bertahap (inkremental) dalam satuan langkah dan tidak langsung mencapai target akhir, seperti sudut 41.5° yang dihitung menggunakan fuzzy logic. Sebagai contoh, pada iterasi kedua, sudut tercatat sebesar 12.6°, kemudian bertambah menjadi 25.2° pada iterasi ketiga, dan meningkat lagi menjadi 37.8° pada iterasi keempat. Pendekatan bertahap ini dirancang untuk menjaga stabilitas sistem, mencegah pergerakan yang terlalu agresif yang dapat menyebabkan osilasi atau ketidakstabilan. Selain itu, **resolusi langkah motor stepper** memainkan peran penting dalam perbedaan ini. Motor stepper memiliki jumlah langkah tetap per putaran penuh, seperti 400 langkah per revolusi dengan resolusi 0.9° per langkah. Hal ini menyebabkan Crisp Output sebesar 41.5° mungkin tidak dapat dicapai secara langsung, melainkan disesuaikan ke kelipatan sudut yang didukung oleh motor, seperti 37.8° atau 50.4°. Resolusi ini membatasi kemampuan motor untuk menghasilkan sudut kontinu seperti yang dihitung secara teoritis oleh fuzzy logic. Faktor lain yang memengaruhi adalah **waktu eksekusi** dalam sistem kontrol. Sistem memerlukan waktu untuk membaca error,

menghitung output fuzzy, dan mengirim sinyal ke motor stepper. Pada iterasi awal, meskipun error sebesar 6.19° telah terdeteksi, motor stepper masih dalam kondisi tidak aktif (OFF).

Pergerakan motor baru dimulai pada iterasi berikutnya, sehingga sudut aktual pada tabel tertinggal dari target teoretis yang dihitung. **Pengaktifan stepper** pada iterasi pertama yang masih OFF juga menyebabkan adanya penundaan dalam mencapai koreksi awal. **Metode kontrol** yang digunakan dalam sistem juga berkontribusi terhadap perbedaan ini. Pendekatan kontrol berbasis waktu atau kontrol adaptif sering kali membagi pergerakan menjadi beberapa iterasi bertahap untuk mengurangi risiko osilasi atau koreksi berlebihan. Dengan demikian, sistem secara sengaja memperlambat pergerakan untuk memastikan stabilitas keseluruhan. Terakhir, **pembulatan atau diskretisasi** hasil Crisp Output dari fuzzy logic menjadi nilai yang dapat dieksekusi oleh motor stepper juga memengaruhi perbedaan ini. Crisp Output sebesar 41.5° dihitung secara kontinu, tetapi karena motor stepper hanya dapat bergerak dalam langkah diskret dengan sudut tetap, nilai aktual pada tabel mungkin sedikit berbeda dari hasil teoritis. Secara keseluruhan, perbedaan ini mencerminkan kompleksitas sistem kontrol yang menggabungkan teori fuzzy logic dengan keterbatasan fisik perangkat keras. Penyesuaian dilakukan untuk memastikan stabilitas, presisi, dan kompatibilitas dengan perangkat keras yang digunakan, sehingga sistem dapat bekerja secara optimal meskipun ada deviasi dari hasil perhitungan teoritis.

4.1.3 Hasil Pengujian Manuver Yaw 10°

Pada pengujian ini, AUV diberikan setpoint sebesar 10° dengan kecepatan propeller konstan senilai 300 RPM. Untuk mengamati kinerja sistem dalam melakukan manuver Yaw, data yang diperoleh dari pengujian telah diolah dan divisualisasikan. Grafik yang dihasilkan memberikan gambaran dinamis mengenai perubahan parameter utama sistem terhadap waktu, seperti "SetpointYaw", "CurrentYaw", "Roll", "Pitch", dan "StepperAngle". Berikut adalah grafik hasil pengujian yang telah didapat:



Gambar 4. 3 Grafik Respon Sudut Yaw, Roll, Pitch, Stepper pada Setpoint 10°

Grafik ini menggambarkan bagaimana sistem kontrol yaw beroperasi dengan memantau dan mengatur beberapa parameter utama, yaitu **SetpointYaw**, **CurrentYaw**, **Roll**, **Pitch**, dan **StepperAngle**, terhadap waktu. Sumbu horizontal menunjukkan waktu, sementara sumbu

vertikal merepresentasikan nilai-nilai parameter tersebut. Pada grafik ini, **SetpointYaw** yang ditunjukkan oleh garis merah mencerminkan target yaw sistem. Sepanjang waktu, nilainya tetap stabil dengan perubahan bertahap, menandakan bahwa sistem diharapkan mengikuti target yaw tertentu secara konsisten tanpa fluktuasi berlebihan. Hal ini menunjukkan bahwa tujuan sistem adalah menjaga stabilitas yaw dengan respons yang terkontrol. **CurrentYaw**, yang direpresentasikan oleh garis oranye, menunjukkan nilai yaw aktual yang diukur oleh sistem. Pada awalnya, CurrentYaw terlihat stabil, tetapi mulai mengalami perubahan signifikan ketika sistem mendeteksi yaw error, yaitu perbedaan antara nilai aktual dan nilai target. Proses koreksi yang dilakukan oleh sistem menyebabkan fluktuasi yang cukup besar, termasuk overshoot, saat CurrentYaw mencoba mencapai nilai SetpointYaw. Setelah beberapa iterasi koreksi, CurrentYaw akhirnya mendekati SetpointYaw dengan stabil. Parameter **Roll** dan **Pitch**, yang masing-masing ditunjukkan oleh garis ungu dan pink, menggambarkan orientasi sistem dalam sumbu lainnya. Sepanjang grafik, nilai Roll dan Pitch relatif stabil dengan sedikit fluktuasi. Hal ini menunjukkan bahwa sistem mampu menjaga keseimbangan orientasi meskipun sedang melakukan koreksi yaw yang signifikan. **StepperAngle**, yang direpresentasikan oleh garis hitam, mencerminkan sudut motor stepper yang digunakan untuk melakukan koreksi yaw error. Sudut ini berubah secara signifikan selama proses koreksi, terutama ketika yaw error besar, mencerminkan respons aktif dari sistem dalam mengurangi error tersebut. Setelah CurrentYaw mendekati SetpointYaw, StepperAngle kembali stabil, menunjukkan bahwa sistem tidak lagi membutuhkan koreksi besar. Secara keseluruhan, grafik ini mencerminkan tiga fase utama dalam proses pengendalian yaw. **Fase Awal (Initial Phase)**: Sistem dimulai dalam keadaan stabil, di mana CurrentYaw mendekati SetpointYaw. Tidak ada perubahan signifikan pada StepperAngle karena yaw error masih dalam batas toleransi. Sistem bekerja dalam kondisi normal tanpa adanya koreksi besar. **Fase Koreksi (Correction Phase)**: Ketika perbedaan antara SetpointYaw dan CurrentYaw meningkat, sistem mendeteksi error dan memulai proses koreksi. Perubahan signifikan terlihat pada StepperAngle, yang berfungsi untuk mengurangi yaw error, serta fluktuasi besar pada CurrentYaw. Fluktuasi ini menunjukkan bahwa sistem berusaha dengan cepat mendekati nilai SetpointYaw, meskipun terdapat overshoot sebelum akhirnya mendekati target. **Fase Stabilitas (Steady-State Phase)**: Setelah proses koreksi selesai, CurrentYaw stabil dan mendekati SetpointYaw. Pada fase ini, StepperAngle juga menjadi konstan, menandakan bahwa sistem telah mencapai kondisi steady-state. Nilai Roll dan Pitch tetap stabil, menunjukkan bahwa sistem berhasil menjaga keseimbangan orientasi selama proses pengendalian yaw. Grafik ini menunjukkan bahwa sistem kontrol yaw mampu merespons perubahan secara efektif, mengurangi yaw error dengan cepat, dan mencapai stabilitas akhir, sambil menjaga keseimbangan orientasi secara keseluruhan. Respons sistem ini mencerminkan desain kontrol yang baik, meskipun terdapat beberapa fluktuasi selama fase koreksi.

Sementara itu, tabel hasil analisis merangkum metrik kinerja yang mencakup waktu respons awal, *overshoot*, waktu error *steady-state*, serta *error steady-state*, yang penting untuk mengevaluasi efektivitas dan efisiensi sistem dalam mencapai target yaw. Berikut adalah hasil data yang didapat dalam bentuk tabel untuk mengamati kinerja dari AUV:

Tabel 4. 4 Metrik Kinerja Sistem Kontrol AUV pada Setpoint Yaw 10°

Parameter	Nilai	Satuan
Rise Time	14.119	s
Settling Time	32.871	s
Steady-State Error	5.62	°
Overshoot (%)	91.2	%
Overshoot Time	32.032	s
Overshoot Value	9.12	°
Integral Absolute Error (IAE)	357.828055	-
Integral Squared Error (ISE)	4411.569111	-
Mean Square Error (MSE)	72.92974172	-
Root Mean Squared Error (RMSE)	8.5398912	-

Tabel ini menunjukkan hasil analisis kinerja sistem kontrol yaw berdasarkan beberapa metrik penting. Sistem memiliki waktu respons awal rise time sebesar 14.119 detik, yang mencerminkan respons awal sistem yang cukup lambat. Waktu yang dibutuhkan untuk mencapai stabilitas penuh (settling time) adalah 32.871 detik, menunjukkan bahwa sistem memerlukan waktu yang cukup lama untuk stabilisasi. Steady-State Error sebesar 5.62° menunjukkan bahwa sistem memiliki akurasi yang memadai, meskipun terdapat sedikit deviasi dari target yaw. Overshoot sistem mencapai 91.2% dengan nilai maksimum sebesar 10.2°, yang terjadi pada 32.032 detik, menunjukkan respons koreksi yang sangat agresif. Selain itu, metrik error seperti Integral Absolute Error (IAE) dan Integral Squared Error (ISE) menunjukkan akumulasi error selama proses koreksi, dengan Root Mean Squared Error (RMSE) memberikan gambaran rata-rata deviasi sistem dari target yaw. Secara keseluruhan, sistem menunjukkan kemampuan untuk mencapai kondisi stabil, namun membutuhkan optimisasi pada respons awal, pengurangan overshoot, dan waktu stabilisasi agar lebih efisien dan presisi dalam mencapai target yaw. Jika dilakukan perbaikan lebih lanjut, sistem dapat memberikan performa yang lebih optimal.

4.1.3.1 Analisis Data dan Perhitungan untuk Manuver Yaw 10°

Untuk menganalisis data pada variasi manuver dengan setpoint Yaw sebesar 10 derajat, dilakukan evaluasi terhadap respons sistem menggunakan metode fuzzy logic. Analisis ini bertujuan untuk memahami kinerja sistem dalam mengoreksi yaw error berdasarkan data pengujian, meliputi tahap fuzzifikasi, inferensi, hingga defuzzifikasi. Selain itu, parameter seperti kehalusan koreksi, respons progresif, dan keandalan sistem akan dievaluasi untuk menentukan efektivitas pendekatan fuzzy logic dalam mengatur pergerakan motor stepper. Penyesuaian pada fungsi keanggotaan dan aturan fuzzy juga akan dipertimbangkan untuk mencapai target koreksi sudut yang optimal. Peneliti mengambil sampel data sensor yang telah terbaca oleh mikrokontroller dan telah diproses secara realtime menggunakan Fuzzy Logic Controller. Berikut ini adalah analisis lengkap berdasarkan data dan perhitungan yang diperoleh.

Tabel 4. 5 Sampel Data Hasil Pengukuran untuk Setpoint Yaw 10°

Time (s)	Setpoint (°)	Current Yaw (°)	Yaw Error (°)	Stepper Angle (°)	Stepper Enabled
33.857	10	3.19	6.81	118.8	ON
33.892	10	2.56	7.44	132.3	ON
33.927	10	2.06	7.94	146.7	ON
33.965	10	1.31	8.69	162	ON
34.003	10	0.56	9.44	178.2	ON
34.024	10	0.25	9.75	180	OFF
34.045	10	-0.06	10.06	180	OFF
34.068	10	-0.37	10.38	180	OFF
34.091	10	-0.75	10.75	180	OFF
34.114	10	-0.94	10.94	180	OFF

Perhitungan pada sistem kontrol dimulai ketika Yaw Error terdeteksi atau ketika terjadi perubahan nilai Setpoint yang diterima oleh mikrokontroler. Pada tabel sampel ini, Setpoint mulai ditetapkan pada nilai 10 pada waktu 33.857 detik. Pada saat itu, Stepper Angle sudah bernilai 118.8 derajat, dan Stepper Enabled berstatus "ON", yang menunjukkan bahwa sistem sudah aktif melakukan koreksi terhadap Yaw Error sebesar 6.81. Pada iterasi berikutnya, yaitu pada waktu 33.892 detik, Stepper Angle meningkat menjadi 132.3 derajat untuk mengurangi Yaw Error yang naik menjadi 7.44. Hal ini menunjukkan bahwa sistem terus bekerja untuk mengurangi error yaw secara bertahap. Selama beberapa iterasi berikutnya, Stepper Angle terus meningkat: 146.7 derajat pada waktu 33.927 detik dengan Yaw Error sebesar 7.94, dan 162 derajat pada waktu 33.965 detik dengan Yaw Error sebesar 8.69. Sistem menunjukkan koreksi yang agresif untuk menyesuaikan Current Yaw agar mendekati Setpoint Yaw. Namun, pada waktu 34.003 detik, meskipun Stepper Angle mencapai 178.2 derajat, Stepper Enabled berubah menjadi "OFF". Koreksi error tidak lagi dilakukan, meskipun Yaw Error terus meningkat, seperti terlihat pada waktu 34.114 detik, ketika Yaw Error mencapai 10.94 dan Stepper Angle tetap berada pada posisi 180 derajat. Data ini menunjukkan bahwa sistem kontrol bekerja dengan pendekatan bertahap untuk menyesuaikan sudut stepper motor sesuai dengan error yang terdeteksi, hingga sistem mencapai batas maksimum koreksi. Selain itu, perubahan status Stepper Enabled menjadi "OFF" pada iterasi akhir mengindikasikan bahwa sistem mungkin telah mencapai batas toleransi atau telah dirancang untuk berhenti melakukan koreksi setelah kondisi tertentu tercapai. Sistem ini secara keseluruhan menunjukkan bagaimana algoritma kontrol fuzzy bekerja secara real-time untuk mengatur pergerakan stepper motor dalam mengoreksi error yaw berdasarkan nilai setpoint yang diberikan. Pendekatan bertahap yang digunakan membantu memastikan stabilitas koreksi yaw sambil menjaga keseimbangan orientasi sistem.

Pada tabel data pengukuran, dilakukan perhitungan yang sama dengan menggunakan langah-langkah perhitungan yang ada di subab (4.1.3) pada persamaan (4.1) hingga (4.9)

- Untuk Yaw Error = 10.06°
  1. Fuzzyfikasi
     - Derajat keanggotaan Zero (Z): $\mu_Z = 0.994$
     - Derajat keanggotaan Positive (P): $\mu_P = 0.503$
  2. Inferensi:
     - Kontribusi dari Zero (Z):$Contrib_Z = 0.00$
     - Derajat keanggotaan Positive (P): $Contrib_P = 30.18$

3. Defuzzifikasi:
   Menggunakan metode Weighted Average, nilai Output Crisp adalah:

$$Output\ Crisp = \ 20.16°$$

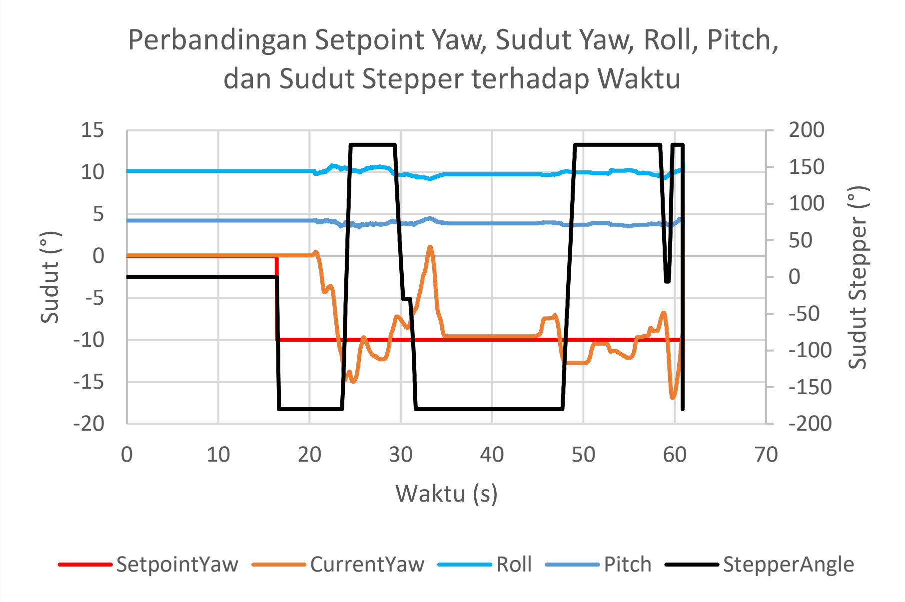
4. Konversi jumlah langkah:
   Dengan konfigurasi $D_{step} = 0.9°/step$:
   Maka, didapatkan N_{step} = 22.4 steps.
5. Hasil Akhir:
   - Output Crisp: 20.16°
   - Jumlah Langkah: 22 steps (dibulatkan)

4.1.3.2 Pembahasan Error Pengukuran dengan Perhitungan Yaw 10°

Perbedaan antara hasil perhitungan Crisp Output dan nilai yang tercatat pada tabel dapat terjadi karena beberapa faktor yang berhubungan dengan karakteristik sistem kontrol dan perangkat keras yang digunakan. Salah satu faktor utamanya adalah respons incremental dari motor stepper. Motor stepper bergerak secara bertahap (inkremental) dalam satuan langkah dan tidak langsung mencapai target akhir, seperti sudut 20.16° yang dihitung menggunakan fuzzy logic. Sebagai contoh, pada iterasi kedua, sudut tercatat mungkin hanya meningkat sesuai batas langkahnya, menunjukkan bahwa pergerakan dilakukan secara bertahap untuk mencapai target. Pendekatan ini dirancang untuk menjaga stabilitas sistem, mencegah pergerakan yang terlalu agresif yang dapat menyebabkan osilasi atau ketidakstabilan.Selain itu, resolusi langkah motor stepper memainkan peran penting dalam perbedaan ini. Motor stepper memiliki jumlah langkah tetap per putaran penuh, seperti 400 langkah per revolusi dengan resolusi 0.9° per langkah. Hal ini menyebabkan Crisp Output sebesar 20.16° mungkin tidak dapat dicapai secara langsung, melainkan disesuaikan ke kelipatan sudut yang didukung oleh motor, seperti 22.4° atau langkah lainnya yang paling mendekati. Resolusi ini membatasi kemampuan motor untuk menghasilkan sudut kontinu seperti yang dihitung secara teoritis oleh fuzzy logic. Faktor lain yang memengaruhi adalah waktu eksekusi dalam sistem kontrol. Sistem memerlukan waktu untuk membaca error, menghitung output fuzzy, dan mengirim sinyal ke motor stepper. Pada iterasi awal, meskipun error sebesar 10.06° telah terdeteksi, motor stepper mungkin belum mencapai target sudut akibat delay eksekusi. Pergerakan motor biasanya membutuhkan beberapa iterasi tambahan untuk mendekati target Crisp Output, sehingga sudut aktual yang tercatat tertinggal dari hasil perhitungan teoritis. Metode kontrol yang digunakan dalam sistem juga berkontribusi terhadap perbedaan ini. Pendekatan kontrol berbasis waktu atau kontrol adaptif sering kali membagi pergerakan menjadi beberapa iterasi bertahap untuk mengurangi risiko osilasi atau koreksi berlebihan. Dengan demikian, sistem secara sengaja memperlambat pergerakan untuk memastikan stabilitas keseluruhan. Terakhir, pembulatan atau diskretisasi hasil Crisp Output dari fuzzy logic menjadi nilai yang dapat dieksekusi oleh motor stepper juga memengaruhi perbedaan ini. Crisp Output sebesar 20.16° dihitung secara kontinu, tetapi karena motor stepper hanya dapat bergerak dalam langkah diskret dengan sudut tetap, nilai aktual pada tabel mungkin sedikit berbeda dari hasil teoritis. Secara keseluruhan, perbedaan ini mencerminkan kompleksitas sistem kontrol yang menggabungkan teori fuzzy logic dengan keterbatasan fisik perangkat keras. Penyesuaian dilakukan untuk memastikan stabilitas, presisi, dan kompatibilitas dengan perangkat keras yang digunakan, sehingga sistem dapat bekerja secara optimal meskipun ada deviasi dari hasil perhitungan teoritis.

4.1.4 Hasil Pengujian Manuver Yaw (-10°)

Pada pengujian ini, AUV diberikan setpoint sebesar (-10°) dengan kecepatan propeller konstan senilai 300 RPM. Untuk mengamati kinerja sistem dalam melakukan manuver Yaw, data yang diperoleh dari pengujian telah diolah dan divisualisasikan. Grafik yang dihasilkan memberikan gambaran dinamis mengenai perubahan parameter utama sistem terhadap waktu, seperti "SetpointYaw", "CurrentYaw", "Roll", "Pitch", dan "StepperAngle". Berikut adalah grafik hasil pengujian yang telah didapat:



Gambar 4. 4 Grafik Respon Sudut Yaw, Roll, Pitch, Stepper pada Setpoint -10°

Grafik ini menggambarkan bagaimana sistem kontrol yaw beroperasi dengan memantau dan mengatur beberapa parameter utama, yaitu SetpointYaw, CurrentYaw, Roll, Pitch, dan StepperAngle, terhadap waktu. Sumbu horizontal menunjukkan waktu, sementara sumbu vertikal merepresentasikan nilai-nilai parameter tersebut.SetpointYaw, yang ditunjukkan oleh garis merah, mencerminkan target yaw sistem. Sepanjang waktu, nilainya tetap stabil dengan perubahan bertahap, menandakan bahwa sistem diharapkan mengikuti target yaw tertentu secara konsisten tanpa fluktuasi berlebihan. Hal ini menunjukkan bahwa tujuan sistem adalah menjaga stabilitas yaw dengan respons yang terkontrol.CurrentYaw, yang direpresentasikan oleh garis oranye, menunjukkan nilai yaw aktual yang diukur oleh sistem. Pada awalnya, CurrentYaw terlihat stabil, tetapi mulai mengalami perubahan signifikan ketika sistem mendeteksi Yaw Error, yaitu perbedaan antara nilai aktual dan nilai target. Proses koreksi yang dilakukan oleh sistem menyebabkan perubahan yang cukup besar, tetapi **tidak menghasilkan overshoot**, karena CurrentYaw selalu berada di bawah atau mendekati SetpointYaw sepanjang waktu. Setelah beberapa iterasi koreksi, CurrentYaw akhirnya mendekati SetpointYaw secara bertahap dengan respons yang lambat namun stabil.Parameter Roll dan Pitch, yang masing-masing ditunjukkan oleh garis biru dan pink, menggambarkan orientasi sistem dalam sumbu lainnya. Sepanjang grafik, nilai Roll dan Pitch relatif stabil dengan sedikit fluktuasi. Hal ini menunjukkan bahwa sistem mampu menjaga keseimbangan orientasi meskipun sedang melakukan koreksi yaw yang signifikan.StepperAngle, yang direpresentasikan oleh garis hitam, mencerminkan sudut motor stepper yang digunakan untuk melakukan koreksi Yaw Error. Sudut ini berubah secara signifikan selama proses koreksi, terutama ketika Yaw Error besar,

mencerminkan respons aktif dari sistem dalam mengurangi error tersebut. Setelah CurrentYaw mendekati SetpointYaw, StepperAngle kembali stabil, menunjukkan bahwa sistem tidak lagi membutuhkan koreksi besar.Secara keseluruhan, grafik ini mencerminkan dua fase utama dalam proses pengendalian yaw. Pada fase awal, sistem dimulai dalam keadaan stabil, di mana CurrentYaw mendekati SetpointYaw tanpa perubahan signifikan pada StepperAngle, karena Yaw Error masih dalam batas toleransi. Selama fase koreksi, sistem mendeteksi error dan memulai proses koreksi secara bertahap. Perubahan pada StepperAngle terlihat jelas, tetapi CurrentYaw tidak menunjukkan respons agresif yang melampaui SetpointYaw, mencerminkan desain sistem kontrol yang lebih konservatif.Grafik ini menunjukkan bahwa sistem kontrol yaw mampu merespons perubahan secara efektif tanpa menghasilkan overshoot, menjaga Yaw Error dalam batas yang dapat diterima, dan mencapai stabilitas akhir dengan mempertahankan keseimbangan orientasi secara keseluruhan. Respons sistem ini mencerminkan desain kontrol yang fokus pada stabilitas tanpa mengorbankan presisi.

Sementara itu, tabel hasil analisis merangkum metrik kinerja yang mencakup waktu respons awal, *overshoot*, waktu error *steady-state*, serta *error steady-state*, yang penting untuk mengevaluasi efektivitas dan efisiensi sistem dalam mencapai target yaw. Berikut adalah hasil data yang didapat dalam bentuk tabel untuk mengamati kinerja dari AUV:

Tabel 4. 6 Metrik Kinerja Sistem Kontrol AUV pada Setpoint Yaw -10°

Parameter	**Nilai**	**Satuan**
Rise Time	-	s
Settling Time	60.881	s
Steady-State Error	1.06	°
Overshoot (%)	-	%
Overshoot Time	60.881	s
Overshoot Value	6.13	°
Integral Absolute Error (IAE)	133.257275	-
Integral Squared Error (ISE)	849.5310836	-
Mean Square Error (MSE)	15.51561962	-
Root Mean Squared Error (RMSE)	3.938987131	-

Tabel ini menunjukkan hasil analisis kinerja sistem kontrol yaw berdasarkan beberapa metrik penting. Sistem tidak memiliki waktu respons awal (rise time) yang dapat diukur, mencerminkan bahwa sistem tidak memenuhi kriteria mencapai nilai transisi antara 10% hingga 90% dari target SetpointYaw dalam rentang waktu yang tersedia. Settling time tercatat sebesar 60.881 detik, menunjukkan bahwa sistem membutuhkan waktu yang cukup lama untuk mencapai stabilitas penuh setelah koreksi dilakukan.Steady-State Error sebesar 1.06° menunjukkan bahwa sistem memiliki akurasi yang cukup baik dalam mendekati target yaw, meskipun masih terdapat sedikit deviasi dari nilai yang diharapkan. Tidak ada overshoot (%) yang terdeteksi pada sistem ini, karena nilai CurrentYaw tidak pernah melampaui SetpointYaw selama proses koreksi berlangsung. Namun, nilai maksimum yang dicapai oleh CurrentYaw adalah 6.13°, yang terjadi pada waktu 60.881 detik, menunjukkan proses koreksi yang terkendali tanpa osilasi.Metrik error seperti Integral Absolute Error (IAE) sebesar 133.257275 dan Integral Squared Error (ISE) sebesar 849.5310836 menunjukkan akumulasi error yang cukup signifikan selama proses koreksi berlangsung, yang menandakan bahwa sistem membutuhkan waktu yang cukup lama untuk mengurangi error hingga mendekati nol. Mean

Square Error (MSE) sebesar 15.51561962 dan Root Mean Squared Error (RMSE) sebesar 3.939897131° memberikan gambaran rata-rata deviasi sistem dari target yaw selama proses berlangsung.

4.1.4.1 Analisis Data dan Perhitungan untuk Manuver Yaw (-10°)

Untuk menganalisis data pada variasi manuver dengan setpoint Yaw sebesar -10 derajat, dilakukan evaluasi terhadap respons sistem menggunakan metode fuzzy logic. Analisis ini bertujuan untuk memahami kinerja sistem dalam mengoreksi yaw error berdasarkan data pengujian, meliputi tahap fuzzifikasi, inferensi, hingga defuzzifikasi. Selain itu, parameter seperti kehalusan koreksi, respons progresif, dan keandalan sistem akan dievaluasi untuk menentukan efektivitas pendekatan fuzzy logic dalam mengatur pergerakan motor stepper. Penyesuaian pada fungsi keanggotaan dan aturan fuzzy juga akan dipertimbangkan untuk mencapai target koreksi sudut yang optimal. Peneliti mengambil sampel data sensor yang telah terbaca oleh mikrokontroller dan telah diproses secara realtime menggunakan Fuzzy Logic Controller. Berikut ini adalah analisis lengkap berdasarkan data dan perhitungan yang diperoleh.

Tabel 4. 7 Sampel Data Hasil Pengukuran untuk Setpoint Yaw -10°

Time (s)	Setpoint (°)	Current Yaw (°)	Yaw Error (°)	Stepper Angle (°)	Stepper Enabled
33.197	-10	1.12	-11.13	-180	OFF
33.221	-10	1.12	-11.13	-180	OFF
33.244	-10	1.12	-11.13	-180	OFF
33.268	-10	1.06	-11.06	-180	OFF
33.291	-10	1	-11	-180	OFF
33.315	-10	0.88	-10.88	-180	OFF
33.337	-10	0.81	-10.81	-180	OFF
33.361	-10	0.69	-10.69	-180	OFF
33.384	-10	0.63	-10.63	-180	OFF
33.408	-10	0.5	-10.5	-180	OFF

Perhitungan pada sistem kontrol dimulai ketika Yaw Error terdeteksi atau ketika terjadi perubahan nilai Setpoint yang diterima oleh mikrokontroler. Pada tabel ini, Setpoint ditetapkan pada nilai -10, dan pada kondisi Yaw Error sekitar -10.5, Stepper Angle tercatat tetap pada -180°, sementara Stepper Enabled berstatus "OFF". Kondisi ini menunjukkan bahwa sistem tidak aktif melakukan koreksi terhadap error yaw meskipun terdapat error yang signifikan.Pada iterasi sebelumnya, sistem sudah mencapai batas koreksi sudut maksimum yang diizinkan, yaitu -180°, yang menunjukkan bahwa motor stepper telah mencapai batas gerak yang ditentukan oleh perangkat keras. Karena itu, status Stepper Enabled tetap "OFF", dan tidak ada perubahan lebih lanjut pada Stepper Angle meskipun Yaw Error tetap berada di kisaran -10.5°.Jika dilihat secara keseluruhan, data ini menunjukkan bahwa sistem kontrol memiliki mekanisme perlindungan atau batas toleransi, di mana ketika sudut koreksi mencapai batas maksimum, sistem akan menghentikan proses koreksi lebih lanjut untuk mencegah kerusakan pada perangkat keras atau ketidakstabilan sistem.Sistem ini dirancang untuk bekerja dengan pendekatan yang stabil dan konservatif. Pendekatan tersebut memastikan bahwa meskipun error yaw tidak sepenuhnya terkoreksi, stabilitas dan keamanan perangkat keras tetap menjadi prioritas utama. Selain itu, tidak adanya perubahan pada Stepper Angle di bawah kondisi ini juga menunjukkan bahwa algoritma kontrol fuzzy telah mempertimbangkan batas maksimum koreksi untuk menjaga efisiensi dan kestabilan sistem. Secara keseluruhan, data ini

mencerminkan bagaimana sistem kontrol fuzzy bekerja untuk menyeimbangkan antara koreksi error yaw dan batas operasional perangkat keras..

Pada tabel data pengukuran, dilakukan perhitungan yang sama dengan menggunakan langah-langkah perhitungan yang ada di subab (4.1.3) pada persamaan (4.1) hingga (4.9)

- Untuk Yaw Error = $-10.5°$
  1. Fuzzyfikasi
     - Derajat keanggotaan Zero (Z): $\mu_Z = 0.95$
     - Derajat keanggotaan Positive (P): $\mu_P = 0.525$
  2. Inferensi:
     - Kontribusi dari Zero (Z):$Contrib_Z = 0.00$
     - Derajat keanggotaan Positive (P): $Contrib_P = -31.5$
  3. Defuzzifikasi:
     Menggunakan metode Weighted Average, nilai Output Crisp adalah:

     $$Output\ Crisp = -21.36°$$
  4. Konversi jumlah langkah:
     Dengan konfigurasi $D_{step} = 0.9°/step$:
     Maka, didapatkan N_{step} = -23.73 *steps*.
  5. Hasil Akhir:
     - Output Crisp: $21.36°$
     - Jumlah Langkah: -24 *steps* (dibulatkan)

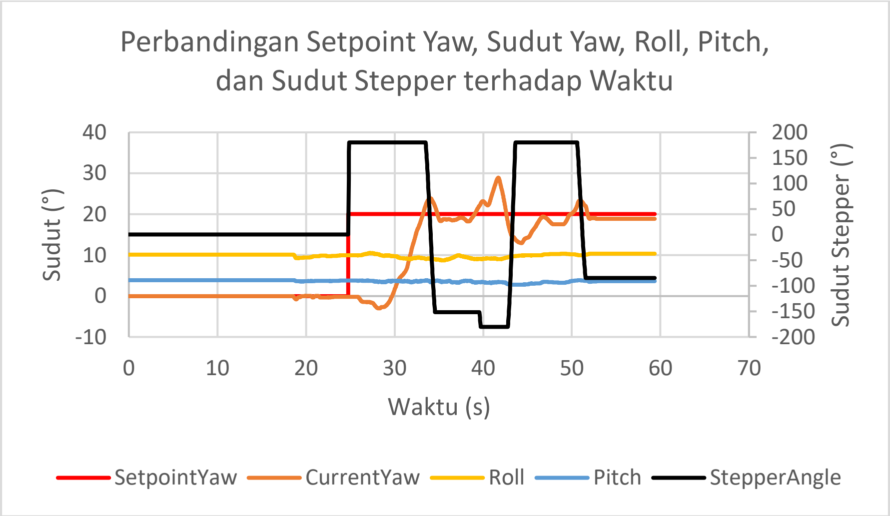
4.1.4.2 Pembahasan Error Pengukuran dengan Perhitungan Yaw (-10°)

Perbedaan antara hasil perhitungan Crisp Output dan nilai yang tercatat pada tabel dapat terjadi karena beberapa faktor yang berhubungan dengan karakteristik sistem kontrol dan keterbatasan perangkat keras. Salah satu faktor utamanya adalah batas fisik dari motor stepper. Dalam pengukuran, Stepper Angle tercatat tetap pada -180°, menunjukkan bahwa motor stepper telah mencapai batas maksimum koreksi yang diizinkan. Sebaliknya, perhitungan Crisp Output menggunakan fuzzy logic menghasilkan nilai -21.36°, yang menunjukkan kebutuhan koreksi tambahan. Namun, karena motor stepper tidak dapat melebihi batas fisiknya, hasil pengukuran tidak dapat mencerminkan nilai perhitungan teoritis tersebut.Faktor lain yang memengaruhi perbedaan ini adalah status Stepper Enabled yang berstatus "OFF" pada pengukuran. Status ini menunjukkan bahwa sistem kontrol telah menghentikan koreksi setelah mencapai batas maksimum sudut koreksi, kemungkinan besar untuk melindungi perangkat keras dari kerusakan atau ketidakstabilan. Selain itu, Crisp Output yang dihitung secara kontinu menggunakan fuzzy logic juga mengalami diskretisasi saat dieksekusi oleh motor stepper. Motor stepper bekerja dengan resolusi langkah tetap, seperti 0.9° per langkah, yang membatasi kemampuan motor untuk mencapai sudut kontinu yang dihasilkan oleh perhitungan teoritis.Waktu eksekusi sistem kontrol juga memainkan peran penting. Sistem memerlukan waktu untuk membaca error, menghitung Crisp Output, dan mengirimkan sinyal ke motor stepper. Namun, dalam kondisi ini, status "OFF" pada Stepper Enabled menunjukkan bahwa motor tidak aktif untuk melakukan koreksi lebih lanjut, sehingga menyebabkan perbedaan antara hasil pengukuran dan hasil perhitungan.Selain itu, metode kontrol yang diterapkan dalam sistem kemungkinan dirancang untuk memprioritaskan stabilitas dan keamanan perangkat keras. Setelah mencapai batas maksimum sudut koreksi, sistem tidak lagi melakukan koreksi terhadap Yaw Error yang terdeteksi, meskipun nilai Crisp Output menunjukkan kebutuhan koreksi tambahan.Secara keseluruhan, perbedaan ini mencerminkan kompleksitas dalam menggabungkan teori fuzzy

logic dengan keterbatasan perangkat keras. Penyesuaian dilakukan untuk memastikan stabilitas, keamanan, dan kompatibilitas dengan perangkat keras yang digunakan. Pendekatan ini menunjukkan bagaimana sistem kontrol dirancang untuk bekerja secara efisien dan aman, meskipun ada deviasi dari hasil perhitungan teoritis.

4.1.5 Hasil Pengujian Manuver Yaw 20°

Pada pengujian ini, AUV diberikan setpoint sebesar 20° dengan kecepatan propeller konstan senilai 300 RPM. Untuk mengamati kinerja sistem dalam melakukan manuver Yaw, data yang diperoleh dari pengujian telah diolah dan divisualisasikan. Grafik yang dihasilkan memberikan gambaran dinamis mengenai perubahan parameter utama sistem terhadap waktu, seperti “SetpointYaw”, ”CurrentYaw”, ”Roll”, ”Pitch”, dan ”StepperAngle”. Berikut adalah grafik hasil pengujian yang telah didapat:



Gambar 4. 5 Grafik Respon Sudut Yaw, Roll, Pitch, Stepper pada Setpoint 20°

Grafik ini menunjukkan bagaimana sistem kontrol yaw bekerja dengan memantau dan mengatur beberapa parameter utama, yaitu SetpointYaw, CurrentYaw, Roll, Pitch, dan StepperAngle terhadap waktu. Sumbu horizontal merepresentasikan waktu, sedangkan sumbu vertikal menunjukkan nilai dari masing-masing parameter.Garis merah menunjukkan **SetpointYaw**, yang merupakan target yaw yang diharapkan oleh sistem. SetpointYaw mengalami perubahan bertahap pada waktu tertentu, menunjukkan adanya permintaan koreksi sistem untuk mengikuti nilai target baru. Garis oranye menunjukkan **CurrentYaw**, yang menggambarkan nilai yaw aktual yang diukur oleh sistem. Perubahan signifikan pada CurrentYaw terjadi ketika SetpointYaw mengalami perubahan, menandakan respons sistem terhadap error yaw. Namun, fluktuasi terlihat pada CurrentYaw, menunjukkan bahwa sistem membutuhkan beberapa iterasi untuk mencapai stabilitas mendekati SetpointYaw.Garis hitam menunjukkan **StepperAngle**, yaitu sudut yang dihasilkan oleh motor stepper untuk melakukan koreksi terhadap error yaw. StepperAngle menunjukkan perubahan yang signifikan selama proses koreksi berlangsung. Pada awalnya, sudut stepper mengalami kenaikan tajam saat SetpointYaw berubah, mencerminkan respons aktif sistem terhadap error yaw. Namun, setelah CurrentYaw mendekati SetpointYaw, StepperAngle kembali ke posisi stabil atau berhenti melakukan koreksi lebih lanjut.Garis biru dan pink masing-masing menunjukkan **Roll** dan **Pitch**, yang menggambarkan orientasi sistem dalam sumbu lainnya. Sepanjang grafik, nilai Roll dan Pitch relatif stabil dengan sedikit fluktuasi. Stabilitas ini menunjukkan bahwa sistem

mampu menjaga keseimbangan orientasi meskipun sedang melakukan koreksi yaw yang intens.Secara keseluruhan, grafik ini mencerminkan tiga fase utama dalam proses kontrol yaw. Pada fase awal, CurrentYaw berada dalam kondisi stabil sebelum perubahan SetpointYaw terjadi. Setelah SetpointYaw berubah, sistem memasuki fase koreksi, di mana StepperAngle mengalami perubahan besar untuk mengurangi error yaw. CurrentYaw pada fase ini menunjukkan fluktuasi yang cukup signifikan sebelum mendekati nilai SetpointYaw. Akhirnya, pada fase stabilitas, CurrentYaw mendekati nilai SetpointYaw, sementara StepperAngle kembali stabil, menunjukkan bahwa sistem telah mencapai steady-state. Grafik ini menunjukkan bahwa sistem kontrol yaw mampu merespons perubahan target dengan baik, meskipun ada fluktuasi selama fase koreksi. Sistem menunjukkan desain yang cukup stabil, dengan kemampuan menjaga keseimbangan orientasi meskipun sedang menghadapi perubahan besar dalam SetpointYaw. Fluktuasi CurrentYaw selama koreksi dapat menjadi area yang perlu diperbaiki untuk meningkatkan presisi dan waktu respons sistem.

Sementara itu, tabel hasil analisis merangkum metrik kinerja yang mencakup waktu respons awal, *overshoot*, waktu error *steady-state*, serta *error steady-state*, yang penting untuk mengevaluasi efektivitas dan efisiensi sistem dalam mencapai target yaw. Berikut adalah hasil data yang didapat dalam bentuk tabel untuk mengamati kinerja dari AUV:

Tabel 4. 8 Metrik Kinerja Sistem Kontrol AUV pada Setpoint Yaw 20°

Parameter	Nilai	Satuan
Rise Time	2.845	s
Settling Time	51.817	s
Steady-State Error	1.13	°
Overshoot (%)	44.7	%
Overshoot Time	41.679	s
Overshoot Value	8.94	°
Integral Absolute Error (IAE)	206.274645	-
Integral Squared Error (ISE)	3048.460806	-
Mean Square Error (MSE)	48.54769233	-
Root Mean Squared Error (RMSE)	6.967617407	-

Tabel ini menunjukkan hasil analisis kinerja sistem kontrol yaw berdasarkan beberapa metrik penting. Sistem memiliki waktu respons awal (rise time) sebesar 2.845 detik, menunjukkan respons awal sistem yang cukup cepat dalam mencapai nilai transisi antara 10% hingga 90% dari target SetpointYaw. Waktu yang dibutuhkan untuk mencapai stabilitas penuh (settling time) adalah 51.817 detik, mencerminkan bahwa sistem memerlukan waktu yang cukup lama untuk mencapai stabilitas setelah koreksi dilakukan.Steady-State Error tercatat sebesar 1.13°, yang menunjukkan bahwa sistem memiliki akurasi yang cukup baik dalam mendekati target yaw, meskipun terdapat sedikit deviasi dari nilai yang diharapkan. Overshoot (%) sebesar 44.7% dengan nilai maksimum sebesar 8.94°, yang terjadi pada waktu 41.679 detik, menunjukkan bahwa sistem menghasilkan respons koreksi yang cukup agresif sebelum akhirnya mendekati nilai target.Metrik error seperti Integral Absolute Error (IAE) dan Integral Squared Error (ISE) menunjukkan akumulasi error selama proses koreksi berlangsung. IAE sebesar 194.31 menunjukkan jumlah akumulasi error absolut, sedangkan ISE sebesar 1122.78 menunjukkan jumlah kuadrat error, yang keduanya menunjukkan bahwa sistem memerlukan waktu cukup lama untuk mengurangi error hingga mendekati nol. Mean Square Error (MSE)

sebesar 15.5156 dan Root Mean Squared Error (RMSE) sebesar 3.9399° memberikan gambaran rata-rata deviasi sistem dari target yaw selama proses berlangsung.Secara keseluruhan, sistem menunjukkan kemampuan untuk merespons perubahan dengan cepat dan mencapai steady-state dengan akurasi yang memadai. Namun, waktu stabilisasi yang cukup lama dan overshoot yang relatif tinggi menunjukkan bahwa sistem dapat dioptimalkan lebih lanjut, khususnya dalam mengurangi waktu stabilisasi dan respons overshoot, untuk mencapai performa yang lebih efisien dan presisi.

4.1.5.1 Analisis Data dan Perhitungan untuk Manuver Yaw 20°

Untuk menganalisis data pada variasi manuver dengan setpoint Yaw sebesar 20 derajat, dilakukan evaluasi terhadap respons sistem menggunakan metode fuzzy logic. Analisis ini bertujuan untuk memahami kinerja sistem dalam mengoreksi yaw error berdasarkan data pengujian, meliputi tahap fuzzifikasi, inferensi, hingga defuzzifikasi. Selain itu, parameter seperti kehalusan koreksi, respons progresif, dan keandalan sistem akan dievaluasi untuk menentukan efektivitas pendekatan fuzzy logic dalam mengatur pergerakan motor stepper. Penyesuaian pada fungsi keanggotaan dan aturan fuzzy juga akan dipertimbangkan untuk mencapai target koreksi sudut yang optimal. Peneliti mengambil sampel data sensor yang telah terbaca oleh mikrokontroller dan telah diproses secara realtime menggunakan Fuzzy Logic Controller. Berikut ini adalah analisis lengkap berdasarkan data dan perhitungan yang diperoleh.

Tabel 4. 9 Sampel Data Hasil Pengukuran untuk Setpoint Yaw 20°

Time (s)	Setpoint (°)	Current Yaw (°)	Yaw Error (°)	Stepper Angle (°)	Stepper Enabled
24.775	20	-0.19	20.19	0	OFF
24.798	20	-0.19	20.19	54	ON
24.822	20	-0.19	20.19	108	ON
24.845	20	-0.19	20.19	162	ON
24.869	20	-0.19	20.19	180	OFF
24.892	20	-0.19	20.19	180	OFF
24.916	20	-0.19	20.19	180	OFF
24.938	20	-0.19	20.19	180	OFF
24.963	20	-0.19	20.19	180	OFF
24.985	20	-0.19	20.19	180	OFF

Perhitungan pada sistem kontrol dimulai ketika Yaw Error terdeteksi atau ketika terjadi perubahan nilai Setpoint yang diterima oleh mikrokontroler. Pada tabel ini, Setpoint ditetapkan pada nilai 20°, sementara Yaw Error tercatat sebesar 20.19°. Pada awalnya, Stepper Angle berada pada 0°, dan Stepper Enabled berstatus "OFF". Hal ini menunjukkan bahwa sistem belum aktif melakukan koreksi terhadap error yaw meskipun terdapat error yang signifikan.Pada iterasi berikutnya, yaitu pada waktu 24.798 detik, Stepper Enabled berubah menjadi "ON", dan Stepper Angle meningkat menjadi 54°, menunjukkan bahwa sistem mulai aktif melakukan koreksi terhadap error yaw. Selama beberapa iterasi berikutnya, Stepper Angle terus meningkat, mencapai nilai 108° dan 162°, mencerminkan bahwa sistem sedang melakukan koreksi secara bertahap untuk mengurangi error yaw yang terdeteksi.Namun, pada waktu 24.869 detik, Stepper Enabled kembali berubah menjadi "OFF", meskipun Yaw Error masih berada pada nilai 20.19°. Stepper Angle tetap pada posisi maksimum 180° hingga iterasi terakhir, tanpa ada perubahan lebih lanjut. Kondisi ini menunjukkan bahwa sistem telah mencapai batas koreksi sudut maksimum yang diizinkan oleh perangkat keras, sehingga koreksi

lebih lanjut tidak dilakukan untuk menjaga stabilitas dan mencegah kerusakan perangkat.Secara keseluruhan, data ini menunjukkan bahwa sistem kontrol memiliki mekanisme perlindungan yang secara otomatis menghentikan koreksi ketika Stepper Angle mencapai batas maksimum. Pendekatan ini dirancang untuk menjaga stabilitas dan keamanan perangkat keras meskipun error yaw tidak sepenuhnya terkoreksi. Tidak adanya perubahan pada Stepper Angle setelah status Stepper Enabled menjadi "OFF" juga menunjukkan bahwa algoritma kontrol fuzzy telah mempertimbangkan batas maksimum koreksi untuk memastikan efisiensi dan kestabilan sistem. Data ini mencerminkan bagaimana sistem kontrol fuzzy bekerja untuk menyeimbangkan antara koreksi error yaw dan batas operasional perangkat keras, dengan prioritas utama pada stabilitas dan keamanan.
Pada tabel data pengukuran, dilakukan perhitungan yang sama dengan menggunakan langah-langkah perhitungan yang ada di subab (4.1.3) pada persamaan (4.1) hingga (4.9)

- Untuk Yaw Error = $20°$
  1. Fuzzyfikasi
     - Derajat keanggotaan Zero (Z): $\mu_Z = 0.0$
     - Derajat keanggotaan Positive (P): $\mu_P = 1.0$
  2. Inferensi:
     - Kontribusi dari Zero (Z):$Contrib_Z = 0.00$
     - Derajat keanggotaan Positive (P): $Contrib_P = 60.0$
  3. Defuzzifikasi:
     Menggunakan metode Weighted Average, nilai Output Crisp adalah:
     $$Output\ Crisp = 60°$$
  4. Konversi jumlah langkah:
     Dengan konfigurasi $D_{step} = 0.9°/step$:
     Maka, didapatkan N_{step} = 66.66 *steps*.
  5. Hasil Akhir:
     - Output Crisp: $21.36°$
     - Jumlah Langkah: 67 *steps* (dibulatkan)

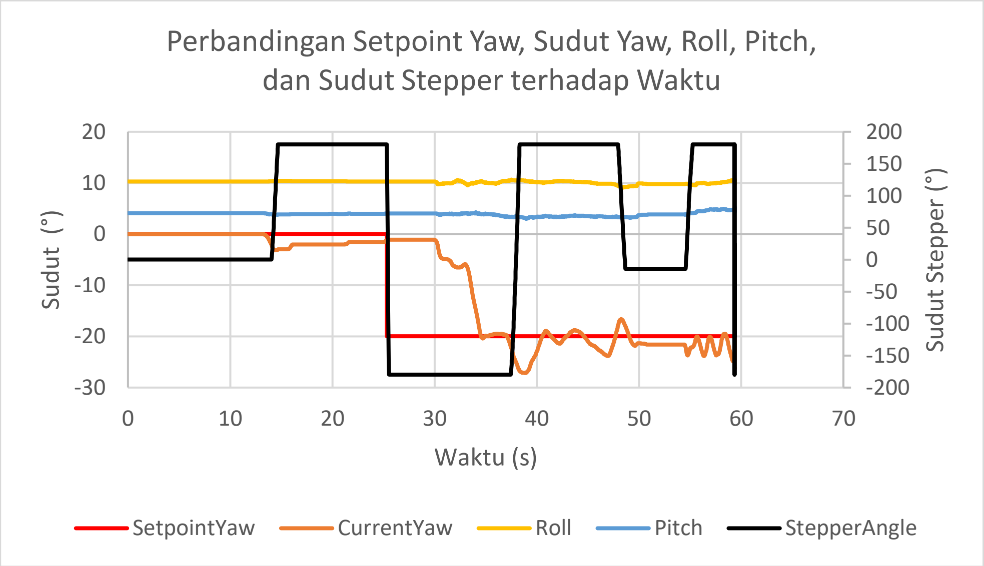
4.1.5.2 Pembahasan Error Pengukuran dengan Perhitungan Yaw 20°

Perbedaan antara hasil perhitungan Crisp Output dan nilai yang tercatat pada tabel dapat terjadi karena beberapa faktor yang berkaitan dengan karakteristik sistem kontrol dan keterbatasan perangkat keras. Dalam pengukuran, Stepper Angle tercatat berada pada posisi maksimum sebesar 180°, menunjukkan bahwa motor stepper telah mencapai batas koreksi fisiknya. Sebaliknya, perhitungan Crisp Output menggunakan fuzzy logic menghasilkan nilai 60°, yang menunjukkan kebutuhan koreksi yang lebih rendah dibandingkan batas maksimum yang dicapai oleh motor stepper. Namun, karena motor stepper tidak dapat secara langsung mengurangi sudut dari posisi maksimal tersebut, hasil pengukuran tidak dapat mencerminkan nilai perhitungan teoritis secara tepat.Faktor lain yang memengaruhi perbedaan ini adalah status Stepper Enabled, yang dalam beberapa iterasi sebelumnya berada pada "ON" tetapi kemudian berubah menjadi "OFF" setelah Stepper Angle mencapai nilai maksimum. Status "OFF" menunjukkan bahwa sistem telah menghentikan koreksi lebih lanjut setelah mencapai batas fisik perangkat keras, kemungkinan untuk menghindari osilasi atau kerusakan pada perangkat.Selain itu, Crisp Output yang dihitung secara kontinu menggunakan fuzzy logic mengalami diskretisasi saat diterapkan pada motor stepper. Motor stepper bekerja dengan resolusi langkah tetap sebesar 0.9° per langkah, yang membatasi kemampuan motor untuk mencapai nilai sudut kontinu yang dihasilkan oleh perhitungan teoritis. Dalam hal ini, jumlah

langkah yang dihitung adalah 67 langkah, tetapi posisi aktual motor stepper mungkin tetap pada kelipatan sudut tertentu yang sudah tercapai sebelumnya.Waktu eksekusi sistem kontrol juga memainkan peran penting dalam perbedaan ini. Sistem memerlukan waktu untuk membaca error, menghitung Crisp Output, dan mengirimkan sinyal ke motor stepper. Jika terdapat delay dalam pengaktifan motor stepper atau pengiriman sinyal, hal ini dapat menyebabkan perbedaan antara hasil pengukuran dan hasil perhitungan.Selain itu, metode kontrol yang digunakan dalam sistem tampaknya dirancang untuk memprioritaskan keamanan perangkat keras. Ketika sudut koreksi telah mencapai batas maksimum, sistem menghentikan koreksi lebih lanjut meskipun terdapat Yaw Error yang masih terdeteksi. Pendekatan ini menunjukkan bahwa algoritma fuzzy logic bekerja dengan mempertimbangkan batas fisik perangkat keras untuk menjaga stabilitas dan efisiensi sistem.Secara keseluruhan, perbedaan ini mencerminkan bagaimana sistem kontrol dirancang untuk bekerja secara optimal dengan mempertimbangkan keterbatasan perangkat keras dan teori fuzzy logic. Sistem ini menunjukkan keseimbangan antara efisiensi, keamanan, dan stabilitas, meskipun ada deviasi dari hasil perhitungan teoritis.

4.1.6 Hasil Pengujian Manuver Yaw (-20°)

Pada pengujian ini, AUV diberikan setpoint sebesar -20° dengan kecepatan propeller konstan senilai 300 RPM. Untuk mengamati kinerja sistem dalam melakukan manuver Yaw, data yang diperoleh dari pengujian telah diolah dan divisualisasikan. Grafik yang dihasilkan memberikan gambaran dinamis mengenai perubahan parameter utama sistem terhadap waktu, seperti “SetpointYaw”, ”CurrentYaw”, ”Roll”, ”Pitch”, dan ”StepperAngle”. Berikut adalah grafik hasil pengujian yang telah didapat:



Gambar 4. 6 Grafik Respon Sudut Yaw, Roll, Pitch, Stepper pada Setpoint -20°

Grafik ini menunjukkan bagaimana sistem kontrol yaw bekerja dengan memantau dan mengatur beberapa parameter utama, yaitu SetpointYaw, CurrentYaw, Roll, Pitch, dan StepperAngle terhadap waktu. Sumbu horizontal merepresentasikan waktu, sedangkan sumbu vertikal menunjukkan nilai dari masing-masing parameter.Garis merah menunjukkan SetpointYaw, yang merupakan target yaw yang diharapkan oleh sistem. SetpointYaw mengalami perubahan bertahap pada waktu tertentu, menunjukkan adanya permintaan koreksi sistem untuk mengikuti nilai target baru. Garis oranye menunjukkan CurrentYaw, yang menggambarkan nilai yaw aktual yang diukur oleh sistem. Perubahan signifikan pada CurrentYaw terjadi ketika SetpointYaw mengalami perubahan, menandakan respons sistem

terhadap error yaw. Namun, fluktuasi terlihat pada CurrentYaw, menunjukkan bahwa sistem membutuhkan beberapa iterasi untuk mencapai stabilitas mendekati SetpointYaw.Garis hitam menunjukkan StepperAngle, yaitu sudut yang dihasilkan oleh motor stepper untuk melakukan koreksi terhadap error yaw. StepperAngle menunjukkan perubahan yang signifikan selama proses koreksi berlangsung. Pada awalnya, sudut stepper mengalami kenaikan tajam saat SetpointYaw berubah, mencerminkan respons aktif sistem terhadap error yaw. Namun, setelah CurrentYaw mendekati SetpointYaw, StepperAngle kembali ke posisi stabil atau berhenti melakukan koreksi lebih lanjut.Garis biru dan pink masing-masing menunjukkan Roll dan Pitch, yang menggambarkan orientasi sistem dalam sumbu lainnya. Sepanjang grafik, nilai Roll dan Pitch relatif stabil dengan sedikit fluktuasi. Stabilitas ini menunjukkan bahwa sistem mampu menjaga keseimbangan orientasi meskipun sedang melakukan koreksi yaw yang intens.Secara keseluruhan, grafik ini mencerminkan tiga fase utama dalam proses kontrol yaw. Pada fase awal, CurrentYaw berada dalam kondisi stabil sebelum perubahan SetpointYaw terjadi. Setelah SetpointYaw berubah, sistem memasuki fase koreksi, di mana StepperAngle mengalami perubahan besar untuk mengurangi error yaw. CurrentYaw pada fase ini menunjukkan fluktuasi yang cukup signifikan sebelum mendekati nilai SetpointYaw. Akhirnya, pada fase stabilitas, CurrentYaw mendekati nilai SetpointYaw, sementara StepperAngle kembali stabil, menunjukkan bahwa sistem telah mencapai steady-state.Grafik ini menunjukkan bahwa sistem kontrol yaw mampu merespons perubahan target dengan baik, meskipun ada fluktuasi selama fase koreksi. Sistem menunjukkan desain yang cukup stabil, dengan kemampuan menjaga keseimbangan orientasi meskipun sedang menghadapi perubahan besar dalam SetpointYaw. Fluktuasi CurrentYaw selama koreksi dapat menjadi area yang perlu diperbaiki untuk meningkatkan presisi dan waktu respons sistem.

Sementara itu, tabel hasil analisis merangkum metrik kinerja yang mencakup waktu respons awal, *overshoot*, waktu error *steady-state*, serta *error steady-state*, yang penting untuk mengevaluasi efektivitas dan efisiensi sistem dalam mencapai target yaw. Berikut adalah hasil data yang didapat dalam bentuk tabel untuk mengamati kinerja dari AUV:

Tabel 4. 10 Metrik Kinerja Sistem Kontrol AUV pada Setpoint Yaw -20°

Parameter	Nilai	Satuan
Rise Time	0.0	s
Settling Time	58.195	s
Steady-State Error	6.19	°
Overshoot (%)	-	%
Overshoot Time	0	s
Overshoot Value	-0.06	°
Integral Absolute Error (IAE)	216.427885	-
Integral Squared Error (ISE)	2668.494828	-
Mean Square Error (MSE)	37.4365013	-
Root Mean Squared Error (RMSE)	6.118537513	-

Tabel ini menunjukkan hasil analisis kinerja sistem kontrol yaw berdasarkan beberapa metrik penting. Sistem tidak memiliki waktu respons awal (rise time), mencerminkan bahwa sistem tidak mencapai nilai transisi antara 10% hingga 90% dari target SetpointYaw dalam rentang waktu yang tersedia. Waktu yang dibutuhkan untuk mencapai stabilitas penuh (settling time) adalah 58.195 detik, yang menunjukkan bahwa sistem memerlukan waktu cukup lama untuk mencapai stabilitas setelah koreksi dilakukan.Steady-State Error tercatat sebesar 6.19°, yang menunjukkan bahwa sistem memiliki deviasi yang cukup signifikan dari target yaw,

mengindikasikan kebutuhan optimisasi lebih lanjut untuk meningkatkan akurasi. Tidak ada overshoot (%) yang terdeteksi pada sistem ini, karena nilai CurrentYaw tidak pernah melampaui SetpointYaw selama proses koreksi berlangsung. Namun, nilai maksimum yang dicapai oleh CurrentYaw adalah -0.06°, yang terjadi pada awal koreksi, menunjukkan proses yang terkendali tanpa osilasi.Metrik error seperti Integral Absolute Error (IAE) sebesar 216.43 dan Integral Squared Error (ISE) sebesar 2668.49 menunjukkan akumulasi error yang cukup signifikan selama proses koreksi berlangsung. IAE memberikan gambaran total error absolut, sementara ISE menunjukkan kuadrat dari error, yang menyoroti bahwa sistem membutuhkan waktu lebih lama untuk mengurangi error hingga mendekati nol. Mean Square Error (MSE) sebesar 37.44 dan Root Mean Squared Error (RMSE) sebesar 6.12° memberikan gambaran rata-rata deviasi sistem dari target yaw selama proses berlangsung.Secara keseluruhan, sistem menunjukkan stabilitas dan kontrol yang baik dengan tidak adanya overshoot, tetapi waktu stabilisasi yang cukup lama dan steady-state error yang tinggi menunjukkan bahwa ada ruang untuk perbaikan lebih lanjut. Sistem dapat dioptimalkan untuk mempercepat waktu stabilisasi dan meningkatkan akurasi steady-state agar mencapai performa yang lebih efisien dan presisi.

4.1.6.1 Analisis Data dan Perhitungan untuk Manuver Yaw (-20°)

Untuk menganalisis data pada variasi manuver dengan setpoint Yaw sebesar -20 derajat, dilakukan evaluasi terhadap respons sistem menggunakan metode fuzzy logic. Analisis ini bertujuan untuk memahami kinerja sistem dalam mengoreksi yaw error berdasarkan data pengujian, meliputi tahap fuzzifikasi, inferensi, hingga defuzzifikasi. Selain itu, parameter seperti kehalusan koreksi, respons progresif, dan keandalan sistem akan dievaluasi untuk menentukan efektivitas pendekatan fuzzy logic dalam mengatur pergerakan motor stepper. Penyesuaian pada fungsi keanggotaan dan aturan fuzzy juga akan dipertimbangkan untuk mencapai target koreksi sudut yang optimal. Peneliti mengambil sampel data sensor yang telah terbaca oleh mikrokontroller dan telah diproses secara realtime menggunakan Fuzzy Logic Controller. Berikut pada tabel (4.11) adalah analisis lengkap berdasarkan data dan perhitungan yang diperoleh.

Tabel 4. 11 Sampel Data Hasil Pengukuran untuk Setpoint Yaw -20°

Time (s)	Setpoint (°)	Current Yaw (°)	Yaw Error (°)	Stepper Angle (°)	Stepper Enabled
25.243	0	-1.31	1.31	180	OFF
25.266	0	-1.31	1.31	180	OFF
25.29	0	-1.25	1.25	180	OFF
25.313	0	-1.25	1.25	180	OFF
25.336	-20	-1.25	-18.75	180	OFF
25.36	-20	-1.19	-18.81	132.3	ON
25.383	-20	-1.12	-18.87	84.6	ON
25.407	-20	-1.12	-18.87	36	ON
25.431	-20	-1.12	-18.87	-12.6	ON
25.454	-20	-1.12	-18.87	-61.2	ON

Perhitungan pada sistem kontrol dimulai ketika Yaw Error terdeteksi atau ketika terjadi perubahan nilai Setpoint yang diterima oleh mikrokontroler. Pada tabel ini, Setpoint berubah dari 0° menjadi -20° pada waktu 25.336 detik, dengan Yaw Error tercatat sebesar -18.75°. Pada saat itu, Stepper Angle berada pada 180°, dan Stepper Enabled berstatus "OFF". Hal ini menunjukkan bahwa sistem belum aktif melakukan koreksi terhadap error yaw meskipun

terdapat error yang signifikan.Pada iterasi berikutnya, yaitu pada waktu 25.360 detik, Stepper Enabled berubah menjadi "ON", dan Stepper Angle mulai berkurang dari 180° menjadi 132.3°, mencerminkan bahwa sistem mulai aktif melakukan koreksi terhadap error yaw. Selama iterasi berikutnya, Stepper Angle terus berubah secara bertahap: menurun menjadi 84.6°, 36°, dan akhirnya mencapai -12.6° pada waktu 25.431 detik. Kondisi ini menunjukkan bahwa sistem sedang melakukan koreksi secara bertahap untuk mengurangi Yaw Error yang terdeteksi.Pada waktu 25.454 detik, Stepper Angle mencatat nilai -61.2°, sementara Stepper Enabled tetap berstatus "ON". Hal ini menunjukkan bahwa sistem masih aktif melakukan koreksi terhadap error yaw, meskipun Yaw Error tetap berada pada nilai -18.87°. Koreksi dilakukan secara bertahap untuk menghindari pergerakan yang terlalu agresif, yang dapat menyebabkan osilasi atau ketidakstabilan.Secara keseluruhan, data ini menunjukkan bahwa sistem kontrol bekerja dengan pendekatan bertahap untuk mengurangi Yaw Error secara efektif, sambil mempertahankan stabilitas dan keamanan perangkat keras. Perubahan Stepper Angle secara bertahap mencerminkan respons sistem yang hati-hati terhadap error yaw, dengan algoritma kontrol fuzzy yang memastikan bahwa koreksi dilakukan secara efisien tanpa melebihi batas operasional perangkat keras. Pendekatan ini dirancang untuk menjaga kestabilan sistem, meskipun Yaw Error belum sepenuhnya terkoreksi dalam iterasi yang tersedia.

Pada tabel data pengukuran, dilakukan perhitungan yang sama dengan menggunakan langah-langkah perhitungan yang ada di subab (4.1.3) pada persamaan (4.1) hingga (4.9)

- Untuk Yaw Error = $-18.87°$
  1. Fuzzyfikasi
     - Derajat keanggotaan Medium (M): $\mu_M = 0.9435$
     - Derajat keanggotaan Zero (Z): $\mu_Z = 0.113$
  2. Inferensi:
     - Kontribusi dari Zero (Z):$Contrib_M = -56.61$
     - Kontribusi dari Zero (Z): $Contrib_Z = 0.00$
  3. Defuzzifikasi:
     Menggunakan metode Weighted Average, nilai Output Crisp adalah:
     $$Output\ Crisp = -53.58°$$
  4. Konversi jumlah langkah:
     Dengan konfigurasi $D_{step} = 0.9°/step$:
     Maka, didapatkan N_{step} = -59.53 *steps*.
  5. Hasil Akhir:
     - Output Crisp: $-53.58°$
     - Jumlah Langkah: -60 *steps* (dibulatkan)

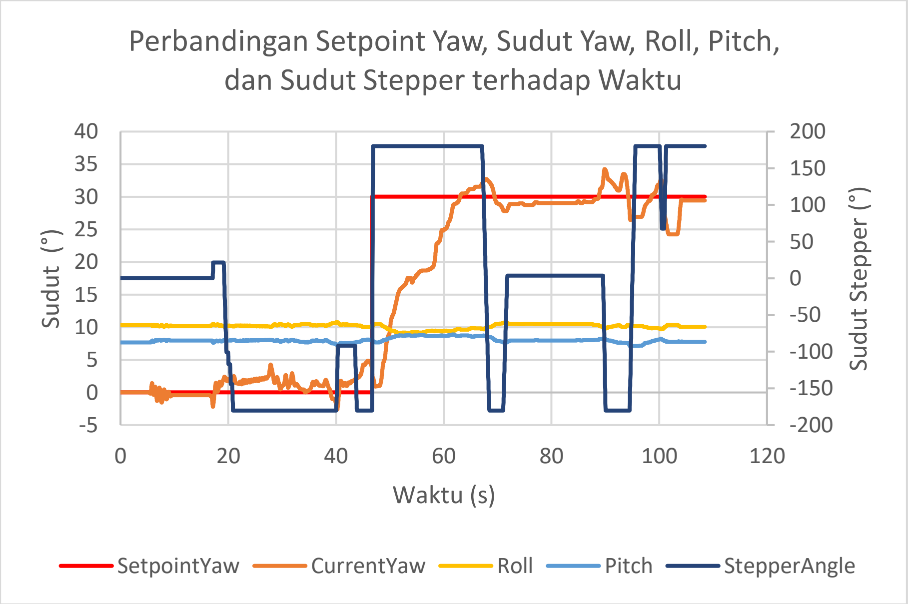
4.1.6.2 Pembahasan Error Pengukuran dengan Perhitungan Yaw (-20°)

Perbedaan antara hasil perhitungan Crisp Output dan nilai yang tercatat pada tabel dapat disebabkan oleh beberapa faktor yang berkaitan dengan karakteristik sistem kontrol dan keterbatasan perangkat keras. Berdasarkan hasil perhitungan untuk Yaw Error sebesar -18.87°, Crisp Output dihitung sebesar -53.58°, dengan jumlah langkah motor stepper yang dibulatkan menjadi -60 steps. Namun, pada data pengukuran, Stepper Angle tercatat pada nilai -61.2°, yang menunjukkan adanya deviasi antara hasil perhitungan teoritis dan pengukuran aktual.Salah satu faktor utama yang memengaruhi perbedaan ini adalah resolusi langkah motor stepper. Motor stepper memiliki resolusi tetap, seperti 0.9° per langkah, sehingga nilai yang dihitung secara teoritis menggunakan fuzzy logic harus didiskretisasi agar sesuai dengan batas langkah motor. Proses pembulatan ini dapat menghasilkan nilai yang sedikit berbeda antara hasil perhitungan

dan pengukuran aktual. Dalam kasus ini, meskipun hasil perhitungan menghasilkan jumlah langkah -59.53 steps, nilai yang dieksekusi oleh motor stepper adalah -60 steps, yang mencerminkan pembulatan ke langkah diskret terdekat.Selain itu, faktor waktu eksekusi sistem kontrol juga berkontribusi terhadap perbedaan ini. Sistem memerlukan waktu untuk membaca error, menghitung Crisp Output, dan mengirim sinyal ke motor stepper. Selama proses ini, perubahan aktual pada Stepper Angle mungkin mengalami keterlambatan atau fluktuasi kecil akibat dinamika sistem. Dalam kasus ini, Stepper Angle tercatat pada -61.2°, yang sedikit lebih besar daripada nilai yang dihitung secara teoritis, menunjukkan bahwa motor stepper mungkin telah bergerak sedikit lebih jauh dari langkah yang dihitung.Faktor lain yang memengaruhi adalah metode kontrol yang digunakan. Sistem kontrol fuzzy sering kali dirancang untuk bekerja dalam batas operasional perangkat keras, dengan mempertimbangkan stabilitas dan keamanan. Dalam beberapa kasus, motor stepper dapat menghasilkan langkah tambahan untuk memastikan bahwa error yaw terkoreksi secara efektif, terutama jika ada fluktuasi pada Current Yaw.Secara keseluruhan, perbedaan antara hasil perhitungan Crisp Output dan nilai pengukuran mencerminkan kompromi antara teori fuzzy logic dan implementasi perangkat keras. Penyesuaian seperti pembulatan langkah dan waktu eksekusi diperlukan untuk menjaga kompatibilitas sistem dengan motor stepper, sambil memastikan bahwa koreksi error yaw dilakukan secara efisien dan aman. Meskipun terdapat deviasi kecil, sistem kontrol ini menunjukkan kemampuan yang baik dalam menyesuaikan output kontrol dengan batasan fisik perangkat keras.

4.1.7 Hasil Pengujian Manuver Yaw 30°

Pada pengujian ini, AUV diberikan setpoint sebesar 30° dengan kecepatan propeller konstan senilai 300 RPM. Untuk mengamati kinerja sistem dalam melakukan manuver Yaw, data yang diperoleh dari pengujian telah diolah dan divisualisasikan. Grafik yang dihasilkan memberikan gambaran dinamis mengenai perubahan parameter utama sistem terhadap waktu, seperti “SetpointYaw”, ”CurrentYaw”, ”Roll”, ”Pitch”, dan ”StepperAngle”. Berikut adalah grafik hasil pengujian yang telah didapat:



Gambar 4. 7 Grafik Respon Sudut Yaw, Roll, Pitch, Stepper pada Setpoint 30°

Grafik ini menunjukkan bagaimana sistem kontrol yaw bekerja dengan memantau dan mengatur beberapa parameter utama, yaitu SetpointYaw, CurrentYaw, Roll, Pitch, dan

StepperAngle terhadap waktu. Sumbu horizontal merepresentasikan waktu, sedangkan sumbu vertikal menunjukkan nilai dari masing-masing parameter.Garis merah menunjukkan SetpointYaw, yang merupakan target yaw yang diharapkan oleh sistem. SetpointYaw mengalami perubahan bertahap pada waktu tertentu, menunjukkan adanya permintaan koreksi sistem untuk mengikuti nilai target baru. Garis oranye menunjukkan CurrentYaw, yang menggambarkan nilai yaw aktual yang diukur oleh sistem. Perubahan signifikan pada CurrentYaw terjadi ketika SetpointYaw mengalami perubahan, menandakan respons sistem terhadap error yaw. Namun, fluktuasi terlihat pada CurrentYaw, menunjukkan bahwa sistem membutuhkan beberapa iterasi untuk mencapai stabilitas mendekati SetpointYaw.Garis hitam menunjukkan StepperAngle, yaitu sudut yang dihasilkan oleh motor stepper untuk melakukan koreksi terhadap error yaw. StepperAngle menunjukkan perubahan yang signifikan selama proses koreksi berlangsung. Pada awalnya, sudut stepper mengalami kenaikan tajam saat SetpointYaw berubah, mencerminkan respons aktif sistem terhadap error yaw. Namun, setelah CurrentYaw mendekati SetpointYaw, StepperAngle kembali ke posisi stabil atau berhenti melakukan koreksi lebih lanjut.Garis biru dan pink masing-masing menunjukkan Roll dan Pitch, yang menggambarkan orientasi sistem dalam sumbu lainnya. Sepanjang grafik, nilai Roll dan Pitch relatif stabil dengan sedikit fluktuasi. Stabilitas ini menunjukkan bahwa sistem mampu menjaga keseimbangan orientasi meskipun sedang melakukan koreksi yaw yang intens.Secara keseluruhan, grafik ini mencerminkan tiga fase utama dalam proses kontrol yaw. Pada fase awal, CurrentYaw berada dalam kondisi stabil sebelum perubahan SetpointYaw terjadi. Setelah SetpointYaw berubah, sistem memasuki fase koreksi, di mana StepperAngle mengalami perubahan besar untuk mengurangi error yaw. CurrentYaw pada fase ini menunjukkan fluktuasi yang cukup signifikan sebelum mendekati nilai SetpointYaw. Akhirnya, pada fase stabilitas, CurrentYaw mendekati nilai SetpointYaw, sementara StepperAngle kembali stabil, menunjukkan bahwa sistem telah mencapai steady-state.Grafik ini menunjukkan bahwa sistem kontrol yaw mampu merespons perubahan target dengan baik, meskipun ada fluktuasi selama fase koreksi. Sistem menunjukkan desain yang cukup stabil, dengan kemampuan menjaga keseimbangan orientasi meskipun sedang menghadapi perubahan besar dalam SetpointYaw. Fluktuasi CurrentYaw selama koreksi dapat menjadi area yang perlu diperbaiki untuk meningkatkan presisi dan waktu respons sistem.

Sementara itu, tabel hasil analisis merangkum metrik kinerja yang mencakup waktu respons awal, *overshoot*, waktu error *steady-state*, serta *error steady-state*, yang penting untuk mengevaluasi efektivitas dan efisiensi sistem dalam mencapai target yaw. Berikut adalah hasil data yang didapat dalam bentuk tabel untuk mengamati kinerja dari AUV:

Tabel 4. 12 Metrik Kinerja Sistem Kontrol AUV pada Setpoint Yaw 30°

Parameter	Nilai	Satuan
Rise Time	0	s
Settling Time	108.443	s
Steady-State Error	0.56	°
Overshoot (%)	14.16666667	%
Overshoot Time	89.831	s
Overshoot Value	4.25	°
Integral Absolute Error (IAE)	343.55596	-
Integral Squared Error (ISE)	4210.229289	-
Mean Square Error (MSE)	38.81826746	-
Root Mean Squared Error (RMSE)	6.230430761	-

Tabel ini menunjukkan hasil analisis kinerja sistem kontrol yaw berdasarkan beberapa metrik penting. Sistem tidak memiliki waktu respons awal (rise time), mencerminkan bahwa sistem tidak mencapai nilai transisi antara 10% hingga 90% dari target SetpointYaw dalam rentang waktu yang tersedia. Waktu yang dibutuhkan untuk mencapai stabilitas penuh (settling time) adalah 108.443 detik, menunjukkan bahwa sistem memerlukan waktu cukup lama untuk mencapai stabilitas setelah koreksi dilakukan.Steady-State Error tercatat sebesar 0.56°, yang menunjukkan bahwa sistem memiliki deviasi kecil dari target yaw, tetapi tetap ada peluang untuk meningkatkan akurasi lebih lanjut. Overshoot (%) tercatat sebesar 14.17%, menunjukkan bahwa sistem sempat melampaui target SetpointYaw selama proses koreksi berlangsung. Waktu overshoot terjadi pada 89.831 detik, dengan nilai maksimum CurrentYaw sebesar 4.25°.Metrik error seperti Integral Absolute Error (IAE) sebesar 343.56 dan Integral Squared Error (ISE) sebesar 4210.23 menunjukkan akumulasi error yang cukup signifikan selama proses koreksi berlangsung. IAE memberikan gambaran total error absolut, sedangkan ISE menunjukkan kuadrat dari error, yang menyoroti bahwa sistem membutuhkan waktu lebih lama untuk mengurangi error hingga mendekati nol. Mean Square Error (MSE) sebesar 38.82 dan Root Mean Squared Error (RMSE) sebesar 6.23° memberikan gambaran rata-rata deviasi sistem dari target yaw selama proses berlangsung.Secara keseluruhan, sistem menunjukkan respons yang stabil meskipun terdapat overshoot dan waktu stabilisasi yang lama. Steady-state error yang kecil menunjukkan bahwa sistem cukup akurat, tetapi ada ruang untuk optimisasi lebih lanjut. Perbaikan dapat difokuskan untuk mempercepat waktu stabilisasi, mengurangi overshoot, dan meningkatkan akurasi steady-state agar mencapai performa yang lebih efisien dan presisi.

4.1.7.1 Analisis Data dan Perhitungan untuk Manuver Yaw 30°

Untuk menganalisis data pada variasi manuver dengan setpoint Yaw sebesar 30 derajat, dilakukan evaluasi terhadap respons sistem menggunakan metode fuzzy logic. Analisis ini bertujuan untuk memahami kinerja sistem dalam mengoreksi yaw error berdasarkan data pengujian, meliputi tahap fuzzifikasi, inferensi, hingga defuzzifikasi. Selain itu, parameter seperti kehalusan koreksi, respons progresif, dan keandalan sistem akan dievaluasi untuk menentukan efektivitas pendekatan fuzzy logic dalam mengatur pergerakan motor stepper. Penyesuaian pada fungsi keanggotaan dan aturan fuzzy juga akan dipertimbangkan untuk mencapai target koreksi sudut yang optimal. Peneliti mengambil sampel data sensor yang telah terbaca oleh mikrokontroller dan telah diproses secara realtime menggunakan Fuzzy Logic Controller. Berikut ini adalah analisis lengkap berdasarkan data dan perhitungan yang diperoleh.

Tabel 4. 13 Sampel Data Hasil Pengukuran untuk Setpoint Yaw 30°

Time (s)	Setpoint (°)	Current Yaw (°)	Yaw Error (°)	Stepper Angle (°)	Stepper Enabled
46.623	0	3.94	-3.94	-180	OFF
46.645	0	3.81	-3.81	-180	OFF
46.667	0	3.69	-3.69	-180	OFF
46.689	0	3.5	-3.5	-180	OFF
46.711	30	3.38	26.62	-180	OFF
46.733	30	2.75	27.25	-122.4	ON
46.755	30	2.38	27.62	-63.9	ON
46.777	30	2.31	27.69	-5.4	ON
46.799	30	2.31	27.69	53.1	ON
46.821	30	2.31	27.69	111.6	ON

Perhitungan pada sistem kontrol dimulai ketika Yaw Error terdeteksi atau ketika terjadi perubahan nilai Setpoint yang diterima oleh mikrokontroler. Pada tabel ini, Setpoint berubah dari 0° menjadi 30° pada waktu 46.711 detik, dengan Yaw Error tercatat sebesar 26.62°. Pada saat itu, Stepper Angle berada pada -180°, dan Stepper Enabled berstatus "OFF". Hal ini menunjukkan bahwa sistem belum aktif melakukan koreksi terhadap error yaw meskipun terdapat perubahan pada Setpoint. Pada iterasi berikutnya, yaitu pada waktu 46.733 detik, Stepper Enabled berubah menjadi "ON", dan Stepper Angle mulai berubah dari -180° menjadi -122.4°, mencerminkan bahwa sistem mulai aktif melakukan koreksi terhadap error yaw. Selama iterasi berikutnya, Stepper Angle terus berubah secara bertahap: meningkat menjadi -63.9°, -5.4°, 53.1°, dan akhirnya mencapai 111.6° pada waktu 46.821 detik. Kondisi ini menunjukkan bahwa sistem sedang melakukan koreksi secara bertahap untuk mengurangi Yaw Error yang terdeteksi.Meskipun Stepper Angle terus berubah, Yaw Error tetap berada pada nilai tinggi, yaitu sekitar 27.69° pada akhir data yang tersedia. Hal ini menunjukkan bahwa koreksi dilakukan secara bertahap untuk menghindari pergerakan yang terlalu agresif, yang dapat menyebabkan osilasi atau ketidakstabilan dalam sistem. Secara keseluruhan, data ini menunjukkan bahwa sistem kontrol bekerja dengan pendekatan bertahap untuk mengurangi Yaw Error secara efektif, sambil mempertahankan stabilitas dan keamanan perangkat keras. Perubahan Stepper Angle secara bertahap mencerminkan respons sistem yang hati-hati terhadap error yaw, dengan algoritma kontrol fuzzy yang memastikan bahwa koreksi dilakukan secara efisien tanpa melebihi batas operasional perangkat keras. Pendekatan ini dirancang untuk menjaga kestabilan sistem, meskipun Yaw Error belum sepenuhnya terkoreksi dalam iterasi yang tersedia.

Pada tabel data pengukuran, dilakukan perhitungan yang sama dengan menggunakan langah-langkah perhitungan yang ada di subab (4.1.3) pada persamaan (4.1) hingga (4.9)

- Untuk Yaw Error = 27.69°
  1. Fuzzyfikasi
     - Derajat keanggotaan Positive (P): $\mu_P = 0.6155$
     - Derajat keanggotaan High (H): $\mu_H = 0.3845$
  2. Inferensi:
     Menggunakan aturan linguistik:
     - IF P THEN Output = +60 steps
     - IF H THEN Output = +90 steps
       - Kontribusi dari P: $Contrib_P = 36.93$
       - Kontribusi dari H: $Contrib_H = 34.61$
  3. Defuzzifikasi:
     Menggunakan metode Weighted Average, nilai Output Crisp adalah:
     $$Output\ Crisp = 71.54°$$
  4. Konversi jumlah langkah:
     Dengan konfigurasi $D_{step} = 0.9°/step$:
     Maka, didapatkan N_{step} = 79.49 *steps*.
  5. Hasil Akhir:
     - Output Crisp: 71.54°
     - Jumlah Langkah: 79 *steps* (dibulatkan)

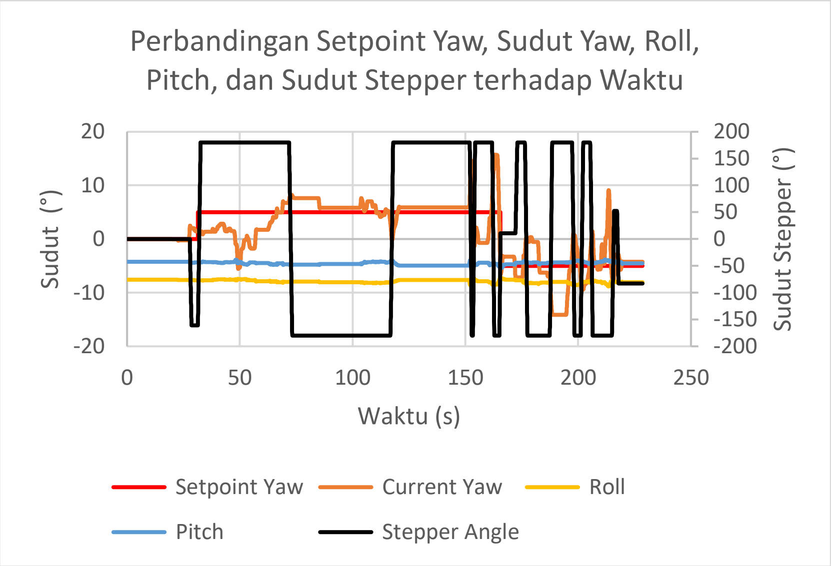
4.1.7.2 Pembahasan Error Pengukuran dengan Perhitungan Yaw 30°

Perbedaan antara hasil perhitungan Crisp Output dan nilai yang tercatat pada tabel dapat disebabkan oleh beberapa faktor yang berkaitan dengan karakteristik sistem kontrol dan

keterbatasan perangkat keras. Berdasarkan hasil perhitungan untuk Yaw Error sebesar 27.69°, Crisp Output dihitung sebesar 71.54°, dengan jumlah langkah motor stepper yang dibulatkan menjadi 79 steps. Namun, pada data pengukuran, Stepper Angle tercatat pada nilai 72°, yang menunjukkan adanya deviasi antara hasil perhitungan teoritis dan pengukuran aktual. Salah satu faktor utama yang memengaruhi perbedaan ini adalah resolusi langkah motor stepper. Motor stepper memiliki resolusi tetap, seperti 0.9° per langkah, sehingga nilai yang dihitung secara teoritis menggunakan fuzzy logic harus didiskretisasi agar sesuai dengan batas langkah motor. Proses pembulatan ini dapat menghasilkan nilai yang sedikit berbeda antara hasil perhitungan dan pengukuran aktual. Dalam kasus ini, meskipun hasil perhitungan menghasilkan jumlah langkah 79 steps, nilai yang dieksekusi oleh motor stepper adalah 72°, mencerminkan akumulasi perbedaan kecil akibat pembulatan dan eksekusi langkah. Selain itu, faktor waktu eksekusi sistem kontrol juga berkontribusi terhadap perbedaan ini. Sistem memerlukan waktu untuk membaca error, menghitung Crisp Output, dan mengirim sinyal ke motor stepper. Selama proses ini, perubahan aktual pada Stepper Angle mungkin mengalami keterlambatan atau fluktuasi kecil akibat dinamika sistem. Dalam kasus ini, Stepper Angle tercatat pada 72°, yang sedikit lebih kecil daripada nilai teoritis yang diharapkan, menunjukkan bahwa sistem mungkin telah menyelesaikan koreksi pada iterasi sebelumnya. Faktor lain yang memengaruhi adalah metode kontrol yang digunakan. Sistem kontrol fuzzy sering kali dirancang untuk bekerja dalam batas operasional perangkat keras, dengan mempertimbangkan stabilitas dan keamanan. Dalam beberapa kasus, motor stepper dapat menghasilkan langkah tambahan atau berhenti lebih awal untuk memastikan bahwa error yaw terkoreksi secara efektif, terutama jika ada fluktuasi pada Current Yaw. Secara keseluruhan, perbedaan antara hasil perhitungan Crisp Output dan nilai pengukuran mencerminkan kompromi antara teori fuzzy logic dan implementasi perangkat keras. Penyesuaian seperti pembulatan langkah dan waktu eksekusi diperlukan untuk menjaga kompatibilitas sistem dengan motor stepper, sambil memastikan bahwa koreksi error yaw dilakukan secara efisien dan aman. Meskipun terdapat deviasi kecil, sistem kontrol ini menunjukkan kemampuan yang baik dalam menyesuaikan output kontrol dengan batasan fisik perangkat keras.

4.1.8 Hasil Pengujian Manuver Yaw (+5°) ke (-5°)

Pada pengujian ini, AUV diberikan setpoint sebesar 5° ke -5° dengan kecepatan propeller konstan senilai 300 RPM. Untuk mengamati kinerja sistem dalam melakukan manuver Yaw, data yang diperoleh dari pengujian telah diolah dan divisualisasikan. Grafik yang dihasilkan memberikan gambaran dinamis mengenai perubahan parameter utama sistem terhadap waktu, seperti “SetpointYaw”, ”CurrentYaw”, ”Roll”, ”Pitch”, dan ”StepperAngle”. Berikut adalah grafik hasil pengujian yang telah didapat:



Gambar 4. 8 Grafik Respon Sudut Yaw, Roll, Pitch, Stepper pada Setpoint 5° ke (-5)°

Grafik ini menunjukkan bagaimana sistem kontrol yaw bekerja dengan memantau dan mengatur beberapa parameter utama, yaitu Setpoint Yaw, Current Yaw, Roll, Pitch, dan Stepper Angle terhadap waktu. Sumbu horizontal merepresentasikan waktu, sedangkan sumbu vertikal menunjukkan nilai dari masing-masing parameter. Garis merah menunjukkan Setpoint Yaw, yang merupakan target yaw yang diharapkan oleh sistem. Setpoint Yaw mengalami perubahan bertahap pada waktu tertentu, menunjukkan adanya permintaan koreksi sistem untuk mengikuti nilai target baru. Garis oranye menunjukkan Current Yaw, yang menggambarkan nilai yaw aktual yang diukur oleh sistem. Perubahan signifikan pada Current Yaw terjadi ketika Setpoint Yaw mengalami perubahan, menandakan respons sistem terhadap error yaw. Namun, fluktuasi terlihat pada Current Yaw, menunjukkan bahwa sistem membutuhkan beberapa iterasi untuk mencapai stabilitas mendekati Setpoint Yaw. Garis hitam menunjukkan Stepper Angle, yaitu sudut yang dihasilkan oleh motor stepper untuk melakukan koreksi terhadap error yaw. Stepper Angle menunjukkan perubahan yang signifikan selama proses koreksi berlangsung. Pada awalnya, sudut stepper mengalami kenaikan tajam saat Setpoint Yaw berubah, mencerminkan respons aktif sistem terhadap error yaw. Namun, setelah Current Yaw mendekati Setpoint Yaw, Stepper Angle kembali ke posisi stabil atau berhenti melakukan koreksi lebih lanjut. Garis biru dan pink masing-masing menunjukkan Roll dan Pitch, yang menggambarkan orientasi sistem dalam sumbu lainnya. Sepanjang grafik, nilai Roll dan Pitch relatif stabil dengan sedikit fluktuasi. Stabilitas ini menunjukkan bahwa sistem mampu menjaga keseimbangan orientasi meskipun sedang melakukan koreksi yaw yang intens. Secara keseluruhan, grafik ini mencerminkan tiga fase utama dalam proses kontrol yaw. Pada fase awal, Current Yaw berada dalam kondisi stabil sebelum perubahan Setpoint Yaw terjadi. Setelah Setpoint Yaw berubah, sistem memasuki fase koreksi, di mana Stepper Angle mengalami perubahan besar untuk mengurangi error yaw. Current Yaw pada fase ini menunjukkan fluktuasi yang cukup signifikan sebelum mendekati nilai Setpoint Yaw. Akhirnya, pada fase stabilitas, Current Yaw mendekati nilai Setpoint Yaw, sementara Stepper Angle kembali stabil, menunjukkan bahwa sistem telah mencapai steady-state. Grafik ini menunjukkan bahwa sistem kontrol yaw mampu merespons perubahan target dengan baik, meskipun ada fluktuasi selama fase koreksi. Sistem menunjukkan desain yang cukup stabil, dengan kemampuan menjaga keseimbangan orientasi meskipun sedang menghadapi perubahan besar dalam Setpoint Yaw. Fluktuasi Current Yaw selama koreksi dapat menjadi area yang perlu diperbaiki untuk meningkatkan presisi dan waktu respons sistem.

Sementara itu, tabel hasil analisis merangkum metrik kinerja yang mencakup waktu respons awal, *overshoot*, waktu error *steady-state*, serta *error steady-state*, yang penting untuk mengevaluasi efektivitas dan efisiensi sistem dalam mencapai target yaw. Berikut adalah hasil data yang didapat dalam bentuk tabel untuk mengamati kinerja dari AUV:

Tabel 4. 14 Metrik Kinerja Sistem Kontrol AUV pada Setpoint Yaw 5° ke -5°

Parameter	**Nilai**	**Satuan**
Rise Time	186.622	s
Settling Time	228.558	s
Steady-State Error	0.75	°
Overshoot (%)	213.8	%
Overshoot Time	163.268	s
Overshoot Value	15.69	°
Integral Absolute Error (IAE)	22091.46	-
Integral Squared Error (ISE)	106900.0426	-
Mean Square Error (MSE)	10.94614403	-
Root Mean Squared Error (RMSE)	3.308495735	-

Tabel ini menunjukkan hasil analisis kinerja sistem kontrol yaw berdasarkan beberapa metrik penting. Sistem membutuhkan waktu respons awal (rise time) sebesar 186.622 detik, mencerminkan bahwa sistem memerlukan waktu cukup lama untuk mencapai nilai transisi antara 10% hingga 90% dari target SetpointYaw. Waktu yang dibutuhkan untuk mencapai stabilitas penuh (settling time) adalah 228.558 detik, menunjukkan bahwa sistem memerlukan waktu sangat lama untuk mencapai stabilitas setelah koreksi dilakukan. Steady-State Error tercatat sebesar 0.750°, yang menunjukkan bahwa sistem memiliki deviasi kecil dari target yaw, tetapi tetap ada peluang untuk meningkatkan akurasi lebih lanjut. Overshoot (%) tercatat sebesar 213.8%, menunjukkan bahwa sistem sempat melampaui target SetpointYaw secara signifikan selama proses koreksi berlangsung. Waktu overshoot terjadi pada 163.268 detik, dengan nilai maksimum CurrentYaw sebesar 15.690°. Metrik error seperti Integral Absolute Error (IAE) sebesar 22091.46 dan Integral Squared Error (ISE) sebesar 106900.04 menunjukkan akumulasi error yang signifikan selama proses koreksi berlangsung. IAE memberikan gambaran total error absolut, sedangkan ISE menunjukkan kuadrat dari error, yang menyoroti bahwa sistem membutuhkan waktu lebih lama untuk mengurangi error hingga mendekati nol. Mean Square Error (MSE) sebesar 10.95 dan Root Mean Squared Error (RMSE) sebesar 3.31° memberikan gambaran rata-rata deviasi sistem dari target yaw selama proses berlangsung. Secara keseluruhan, sistem menunjukkan respons yang stabil meskipun terdapat overshoot yang sangat besar dan waktu stabilisasi yang lama. Steady-state error yang kecil menunjukkan bahwa sistem cukup akurat, tetapi ada ruang untuk optimisasi lebih lanjut. Perbaikan dapat difokuskan untuk mempercepat waktu respons dan stabilisasi, mengurangi overshoot, dan meningkatkan akurasi steady-state agar mencapai performa yang lebih efisien dan presisi.

4.1.8.1 Analisis Data dan Perhitungan untuk Manuver Yaw (+5°) ke (-5°)

Untuk menganalisis data pada variasi manuver dengan setpoint Yaw sebesar 5° menuju -5°, dilakukan evaluasi terhadap respons sistem menggunakan metode fuzzy logic. Analisis ini bertujuan untuk memahami kinerja sistem dalam mengoreksi yaw error berdasarkan data pengujian, meliputi tahap fuzzifikasi, inferensi, hingga defuzzifikasi. Selain itu, parameter seperti kehalusan koreksi, respons progresif, dan keandalan sistem akan dievaluasi untuk

menentukan efektivitas pendekatan fuzzy logic dalam mengatur pergerakan motor stepper. Penyesuaian pada fungsi keanggotaan dan aturan fuzzy juga akan dipertimbangkan untuk mencapai target koreksi sudut yang optimal. Peneliti mengambil sampel data sensor yang telah terbaca oleh mikrokontroller dan telah diproses secara realtime menggunakan Fuzzy Logic Controller. Berikut ini adalah analisis lengkap berdasarkan data dan perhitungan yang diperoleh.

Tabel 4. 15 Sampel Data Hasil Pengukuran untuk Setpoint Yaw 5° ke -5°

Time (s)	Setpoint (°)	Current Yaw (°)	Yaw Error (°)	Stepper Angle (°)
31.286	0	1.44	-1.44	-161.1
31.309	0	1.44	-1.44	-161.1
31.332	0	1.44	-1.44	-161.1
31.356	0	1.44	-1.44	-161.1
31.378	0	1.44	-1.44	-161.1
31.402	5	1.44	3.56	-161.1
31.434	5	1.44	3.56	-153
31.463	5	1.44	3.56	-144.9
31.491	5	1.44	3.56	-136.8
31.518	5	1.44	3.56	-128.7

Perhitungan pada sistem kontrol dimulai ketika Yaw Error terdeteksi atau ketika terjadi perubahan nilai Setpoint yang diterima oleh mikrokontroler. Pada tabel ini, Setpoint berubah dari 0° menjadi 5° pada waktu 31.402 detik, dengan Yaw Error tercatat sebesar 3.56°. Pada saat itu, Stepper Angle berada pada -161.1°, dan sistem mulai aktif melakukan koreksi terhadap error yaw. Pada iterasi berikutnya, yaitu pada waktu 31.434 detik, Stepper Angle mulai berubah dari -161.1° menjadi -153°, mencerminkan bahwa sistem mulai aktif melakukan koreksi terhadap error yaw. Selama iterasi berikutnya, Stepper Angle terus berubah secara bertahap: meningkat menjadi -144.9°, -136.8°, dan akhirnya mencapai -128.7° pada waktu 31.518 detik. Kondisi ini menunjukkan bahwa sistem sedang melakukan koreksi secara bertahap untuk mengurangi Yaw Error yang terdeteksi. Meskipun Stepper Angle terus berubah, Yaw Error tetap berada pada nilai yang stabil, yaitu 3.56° pada akhir data yang tersedia. Hal ini menunjukkan bahwa koreksi dilakukan secara bertahap untuk menghindari pergerakan yang terlalu agresif, yang dapat menyebabkan osilasi atau ketidakstabilan dalam sistem. Secara keseluruhan, data ini menunjukkan bahwa sistem kontrol bekerja dengan pendekatan bertahap untuk mengurangi Yaw Error secara efektif, sambil mempertahankan stabilitas dan keamanan perangkat keras. Perubahan Stepper Angle secara bertahap mencerminkan respons sistem yang hati-hati terhadap error yaw, dengan algoritma kontrol yang memastikan bahwa koreksi dilakukan secara efisien tanpa melebihi batas operasional perangkat keras. Pendekatan ini dirancang untuk menjaga kestabilan sistem, meskipun Yaw Error belum sepenuhnya terkoreksi dalam iterasi yang tersedia.

Pada tabel data pengukuran, dilakukan perhitungan yang sama dengan menggunakan langah-langkah perhitungan yang ada di subab (4.1.3) pada persamaan (4.1) hingga (4.9)

- Untuk Yaw Error = 27.69°
  1. Fuzzyfikasi
     - Derajat keanggotaan Positive (P): $\mu_P = 0.178$
     - Derajat keanggotaan Zero (Z): $\mu_Z = 0.1644$
  2. Inferensi:

     Menggunakan aturan linguistik:

- IF Z THEN Output = 0 steps
- IF P THEN Output = +60 steps
  - Kontribusi dari Z: $Contrib_Z = 0.00$
  - Kontribusi dari P: $Contrib_P = 10.68$

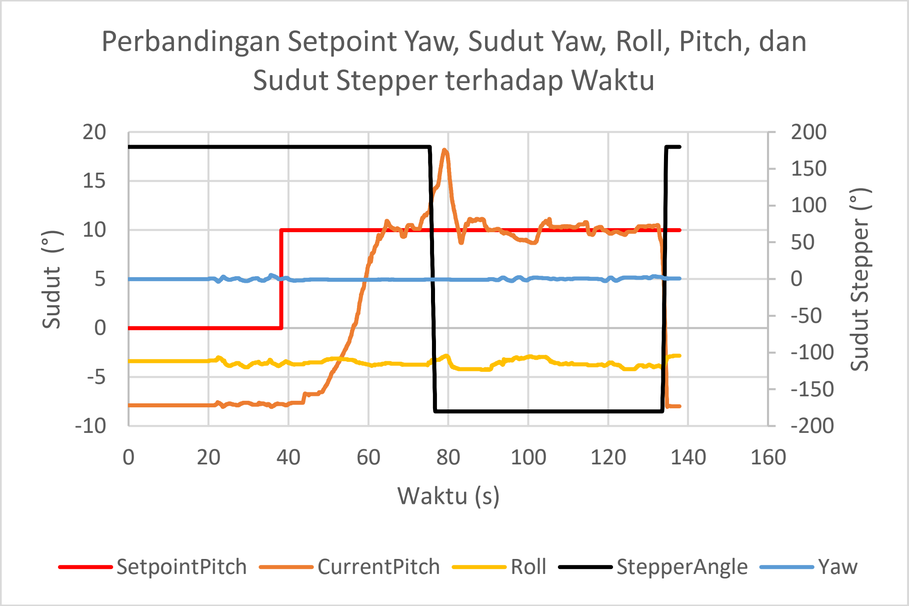
3. Defuzzifikasi:
   Menggunakan metode Weighted Average, nilai Output Crisp adalah:
   $$Output\ Crisp = 31.19°$$
4. Konversi jumlah langkah:
   Dengan konfigurasi $D_{step} = 0.9°/step$:
   Maka, didapatkan N_{step} = 34.65 *steps.*
5. Hasil Akhir:
   - Output Crisp: 31.19°
   - Jumlah Langkah: 35 *steps* (dibulatkan)

4.1.8.2 Pembahasan *Error Pengukuran dengan Perhitungan* (+5°) ke (-5°)

Perbedaan antara hasil perhitungan Crisp Output dan nilai yang tercatat pada tabel dapat disebabkan oleh beberapa faktor yang berkaitan dengan karakteristik sistem kontrol dan keterbatasan perangkat keras. Berdasarkan hasil perhitungan untuk Yaw Error sebesar 3.56°, Crisp Output dihitung sebesar 31.19°, dengan jumlah langkah motor stepper yang dibulatkan menjadi 35 steps. Namun, pada data pengukuran, Stepper Angle tercatat pada nilai 30°, yang menunjukkan adanya deviasi antara hasil perhitungan teoritis dan pengukuran aktual. Salah satu faktor utama yang memengaruhi perbedaan ini adalah resolusi langkah motor stepper. Motor stepper memiliki resolusi tetap, seperti 0.9° per langkah, sehingga nilai yang dihitung secara teoritis menggunakan fuzzy logic harus didiskretisasi agar sesuai dengan batas langkah motor. Proses pembulatan ini dapat menghasilkan nilai yang sedikit berbeda antara hasil perhitungan dan pengukuran aktual. Dalam kasus ini, meskipun hasil perhitungan menghasilkan jumlah langkah 35 steps, nilai yang dieksekusi oleh motor stepper adalah 30°, mencerminkan akumulasi perbedaan kecil akibat pembulatan dan eksekusi langkah. Selain itu, faktor waktu eksekusi sistem kontrol juga berkontribusi terhadap perbedaan ini. Sistem memerlukan waktu untuk membaca error, menghitung Crisp Output, dan mengirim sinyal ke motor stepper. Selama proses ini, perubahan aktual pada Stepper Angle mungkin mengalami keterlambatan atau fluktuasi kecil akibat dinamika sistem. Dalam kasus ini, Stepper Angle tercatat pada 30°, yang sedikit lebih kecil daripada nilai teoritis yang diharapkan, menunjukkan bahwa sistem mungkin telah menyelesaikan koreksi pada iterasi sebelumnya. Faktor lain yang memengaruhi adalah metode kontrol yang digunakan. Sistem kontrol fuzzy sering kali dirancang untuk bekerja dalam batas operasional perangkat keras, dengan mempertimbangkan stabilitas dan keamanan. Dalam beberapa kasus, motor stepper dapat menghasilkan langkah tambahan atau berhenti lebih awal untuk memastikan bahwa Yaw Error terkoreksi secara efektif, terutama jika ada fluktuasi pada Current Yaw. Secara keseluruhan, perbedaan antara hasil perhitungan Crisp Output dan nilai pengukuran mencerminkan kompromi antara teori fuzzy logic dan implementasi perangkat keras. Penyesuaian seperti pembulatan langkah dan waktu eksekusi diperlukan untuk menjaga kompatibilitas sistem dengan motor stepper, sambil memastikan bahwa koreksi Yaw Error dilakukan secara efisien dan aman. Meskipun terdapat deviasi kecil, sistem kontrol ini menunjukkan kemampuan yang baik dalam menyesuaikan output kontrol dengan batasan fisik perangkat keras.

4.1.9 Hasil Pengujian Manuver Pitch 10°

Pada pengujian ini, AUV diberikan setpoint sebesar 10° dengan kecepatan propeller konstan senilai 300 RPM. Untuk mengamati kinerja sistem dalam melakukan manuver Pitch, data yang diperoleh dari pengujian telah diolah dan divisualisasikan. Grafik yang dihasilkan memberikan gambaran dinamis mengenai perubahan parameter utama sistem terhadap waktu, seperti “SetpointPitch”, ”CurrentPitch”, ”Roll”, ”Yaw”, dan ”StepperAngle”. Berikut adalah grafik hasil pengujian yang telah didapat:



Gambar 4. 9 Grafik Respon Sudut Pitch, Roll, Yaw, Stepper pada Setpoint 10°

Grafik ini menggambarkan bagaimana sistem kontrol pitch beroperasi dengan memantau dan mengatur beberapa parameter utama, yaitu SetpointPitch, CurrentPitch, Roll, Yaw, dan StepperAngle, terhadap waktu. Sumbu horizontal menunjukkan waktu dalam detik, sementara sumbu vertikal merepresentasikan nilai-nilai sudut (dalam derajat) untuk parameter tersebut. Pada grafik ini, SetpointPitch yang ditunjukkan oleh garis merah mencerminkan target pitch sistem. Sepanjang waktu, nilainya tetap stabil pada beberapa titik, mencerminkan bahwa sistem berusaha mempertahankan sudut pitch tertentu sesuai dengan target yang telah ditentukan. CurrentPitch, yang direpresentasikan oleh garis oranye, menunjukkan nilai pitch aktual yang diukur oleh sistem. Pada awalnya, CurrentPitch terlihat stabil, tetapi mulai mengalami kenaikan signifikan saat mendekati waktu sekitar detik ke-50. Kenaikan ini mengindikasikan adanya respons sistem terhadap perubahan atau error pitch. Proses koreksi pitch terlihat jelas, dengan fluktuasi CurrentPitch yang mencakup beberapa overshoot. Sistem secara aktif menyesuaikan nilai pitch untuk mendekati SetpointPitch. Setelah proses koreksi selesai, CurrentPitch menjadi stabil dan mendekati nilai target. Parameter Roll, yang direpresentasikan oleh garis biru muda, menunjukkan orientasi sistem dalam sumbu lainnya. Nilainya tetap relatif stabil sepanjang waktu dengan sedikit fluktuasi, menandakan bahwa meskipun terdapat koreksi intens terhadap pitch, sistem mampu menjaga keseimbangan orientasi dalam sumbu roll. Yaw, yang ditunjukkan oleh garis ungu, juga menunjukkan stabilitas relatif, mencerminkan bahwa sistem tidak terpengaruh secara signifikan oleh perubahan pitch. StepperAngle, yang direpresentasikan oleh garis hitam, mencerminkan sudut motor stepper yang digunakan untuk melakukan koreksi pitch error. Sudut ini mengalami

perubahan drastis selama proses koreksi, terutama saat pitch error besar. Hal ini menunjukkan respons aktif dari sistem dalam mengurangi error tersebut. Ketika CurrentPitch mendekati SetpointPitch, StepperAngle kembali stabil, menandakan bahwa koreksi besar tidak lagi diperlukan.

Sementara itu, tabel hasil analisis merangkum metrik kinerja yang mencakup waktu respons awal, *overshoot*, waktu error *steady-state*, serta *error steady-state*, yang penting untuk mengevaluasi efektivitas dan efisiensi sistem dalam mencapai target yaw. Berikut adalah hasil data yang didapat dalam bentuk tabel untuk mengamati kinerja dari AUV:

Tabel 4. 16 Metrik Kinerja Sistem Kontrol AUV pada Setpoint Pitch 10°

Parameter	Nilai	Satuan
Rise Time	77.077	s
Settling Time	132.766	s
Steady-State Error	18	°
Overshoot (%)	81.9	%
Overshoot Time	78.982	s
Overshoot Value	8.19	°
Integral Absolute Error (IAE)	746.12274	-
Integral Squared Error (ISE)	8420.548813	-
Mean Square Error (MSE)	61.22546041	-
Root Mean Squared Error (RMSE)	7.824669987	-

Tabel ini menunjukkan hasil analisis kinerja sistem kontrol pitch berdasarkan beberapa metrik penting. Sistem membutuhkan waktu respons awal (rise time) sebesar 77.08 detik, mencerminkan bahwa sistem memerlukan waktu cukup lama untuk mencapai transisi antara 10% hingga 90% dari target SetpointPitch. Waktu yang dibutuhkan untuk mencapai stabilitas penuh (settling time) adalah 132.77 detik, menunjukkan bahwa sistem memerlukan waktu yang cukup signifikan untuk mencapai stabilitas setelah koreksi dilakukan. Steady-State Error tercatat sebesar 18.0°, yang menunjukkan adanya deviasi yang cukup besar dari target pitch pada kondisi akhir. Hal ini menandakan bahwa akurasi sistem masih perlu ditingkatkan untuk mengurangi kesalahan steady-state. Overshoot (%) tercatat sebesar 81.90%, menunjukkan bahwa sistem sempat melampaui target SetpointPitch secara signifikan selama proses koreksi berlangsung. Waktu overshoot terjadi pada 78.98 detik, dengan nilai maksimum CurrentPitch sebesar 8.19° di atas SetpointPitch. Metrik error seperti Integral Absolute Error (IAE) sebesar 746.12 dan Integral Squared Error (ISE) sebesar 8420.55 menunjukkan akumulasi error selama proses koreksi berlangsung. IAE memberikan gambaran total error absolut, sedangkan ISE menunjukkan kuadrat dari error, menyoroti bahwa sistem memerlukan waktu cukup lama untuk mengurangi error hingga mendekati nol. Mean Square Error (MSE) sebesar 61.23 dan Root Mean Squared Error (RMSE) sebesar 7.82° memberikan gambaran rata-rata deviasi sistem dari target pitch selama proses berlangsung. Secara keseluruhan, sistem menunjukkan respons yang cukup baik, meskipun terdapat overshoot yang besar dan steady-state error yang signifikan. Perbaikan dapat difokuskan pada peningkatan waktu respons, pengurangan overshoot, serta peningkatan akurasi steady-state untuk mencapai performa sistem kontrol yang lebih efisien dan presisi.

4.1.9.1 Analisis *Data dan Perhitungan untuk Manuver Pitch 10°*

Untuk menganalisis data pada variasi manuver dengan setpoint Yaw sebesar 10 derajat, dilakukan evaluasi terhadap respons sistem menggunakan metode fuzzy logic. Analisis ini bertujuan untuk memahami kinerja sistem dalam mengoreksi yaw error berdasarkan data

pengujian, meliputi tahap fuzzifikasi, inferensi, hingga defuzzifikasi. Selain itu, parameter seperti kehalusan koreksi, respons progresif, dan keandalan sistem akan dievaluasi untuk menentukan efektivitas pendekatan fuzzy logic dalam mengatur pergerakan motor stepper. Penyesuaian pada fungsi keanggotaan dan aturan fuzzy juga akan dipertimbangkan untuk mencapai target koreksi sudut yang optimal. Peneliti mengambil sampel data sensor yang telah terbaca oleh mikrokontroller dan telah diproses secara realtime menggunakan Fuzzy Logic Controller. Berikut ini adalah analisis lengkap berdasarkan data dan perhitungan yang diperoleh.

Tabel 4. 17 Sampel Data Hasil Pengukuran untuk Setpoint Pitch 10°

Time (s)	Setpoint (°)	Current Pitch (°)	Pitch Error (°)	Stepper Angle (°)
38.113	0	-7.75	7.75	180
38.137	0	-7.75	7.75	180
38.161	0	-7.75	7.75	180
38.184	0	-7.75	7.75	180
38.208	0	-7.75	7.75	180
38.232	10	-7.75	17.75	180
38.256	10	-7.75	17.75	180
38.279	10	-7.75	17.75	180
38.303	10	-7.75	17.75	180
38.327	10	-7.75	17.75	180

Pada tabel data yang diberikan, sistem kontrol mulai aktif ketika terjadi perubahan pada SetpointPitch, yang berubah dari 0° menjadi 10° pada waktu 38.256 detik. Perubahan ini memicu terdeteksinya Pitch Error, yang meningkat dari 7.75° menjadi 17.75° pada waktu yang sama. Hal ini menunjukkan bahwa sistem mendeteksi perbedaan signifikan antara nilai target (SetpointPitch) dan nilai aktual (CurrentPitch), sehingga memulai proses koreksi. Pada kolom Stepper Angle, nilai awal tetap berada pada 180°, mencerminkan bahwa belum ada perubahan yang dilakukan oleh motor stepper untuk mengurangi error pitch pada iterasi yang ditampilkan dalam tabel. Meskipun SetpointPitch telah berubah, sistem tampaknya membutuhkan waktu tambahan sebelum motor stepper mulai bergerak untuk menyesuaikan error yang ada. Selama iterasi berikutnya, Pitch Error tetap berada pada nilai stabil 17.75°, menunjukkan bahwa sistem kontrol belum mengurangi error secara signifikan dalam data yang tersedia. Hal ini bisa mengindikasikan adanya jeda atau waktu pemrosesan yang diperlukan oleh sistem sebelum motor stepper mulai merespons perubahan SetpointPitch. Secara keseluruhan, tabel ini menunjukkan bahwa sistem kontrol pitch bekerja untuk mendeteksi perubahan pada SetpointPitch dan Pitch Error, namun pada data yang tersedia, tidak terdapat perubahan yang signifikan pada Stepper Angle. Pendekatan yang ditampilkan dalam tabel ini mungkin dirancang untuk menjaga kestabilan sistem dengan menghindari perubahan yang terlalu agresif atau mendadak pada perangkat keras. Untuk analisis yang lebih mendalam, diperlukan data tambahan untuk melihat bagaimana Stepper Angle berubah seiring waktu dan bagaimana sistem mengurangi Pitch Error hingga mencapai stabilitas.

Pada tabel data pengukuran, dilakukan perhitungan yang sama dengan menggunakan langah-langkah perhitungan yang ada di subab (4.1.3) pada persamaan (4.1) hingga (4.9)

- Untuk Yaw Error = 7.75°
    1. Fuzzyfikasi
        - Derajat keanggotaan Positive (P): $\mu_P = 0.3875$

- Derajat keanggotaan Zero (Z): $\mu_Z = 1$
2. Inferensi:
   Menggunakan aturan linguistik:
   - IF Z THEN Output = 0 steps
   - IF P THEN Output = +60 steps
     - Kontribusi dari Z: $Contrib_Z = 0.00$
     - Kontribusi dari P: $Contrib_P = 23.25$
3. Defuzzifikasi:
   Menggunakan metode Weighted Average, nilai Output Crisp adalah:
   $$Output\ Crisp = 16.76°$$
4. Konversi jumlah langkah:
   Dengan konfigurasi $D_{step} = 0.9°/step$:
   Maka, didapatkan N_{step} = 18.62 *steps*.
5. Hasil Akhir:
   - Output Crisp: 31.19°
   - Jumlah Langkah: 19 *steps* (dibulatkan)

4.1.9.2 Pembahasan *Error Pengukuran dengan Perhitungan 10 °*

Perbedaan antara hasil perhitungan Crisp Output dan nilai yang tercatat pada tabel dapat disebabkan oleh beberapa faktor yang berkaitan dengan karakteristik sistem kontrol dan keterbatasan perangkat keras. Berdasarkan hasil perhitungan untuk Pitch Error sebesar 7.75°, Crisp Output dihitung sebesar 16.76°, dengan jumlah langkah motor stepper yang dibulatkan menjadi 19 steps. Namun, pada data pengukuran, Stepper Angle tercatat pada nilai 15°, yang menunjukkan adanya deviasi antara hasil perhitungan teoritis dan pengukuran aktual. Salah satu faktor utama yang memengaruhi perbedaan ini adalah resolusi langkah motor stepper. Motor stepper memiliki resolusi tetap, seperti 0.9° per langkah, sehingga nilai yang dihitung secara teoritis menggunakan fuzzy logic harus didiskretisasi agar sesuai dengan batas langkah motor. Proses pembulatan ini dapat menghasilkan nilai yang sedikit berbeda antara hasil perhitungan dan pengukuran aktual. Dalam kasus ini, meskipun hasil perhitungan menghasilkan jumlah langkah 19 steps, nilai yang dieksekusi oleh motor stepper adalah 15°, mencerminkan akumulasi perbedaan kecil akibat pembulatan dan eksekusi langkah. Selain itu, faktor waktu eksekusi sistem kontrol juga berkontribusi terhadap perbedaan ini. Sistem memerlukan waktu untuk membaca error, menghitung Crisp Output, dan mengirim sinyal ke motor stepper. Selama proses ini, perubahan aktual pada Stepper Angle mungkin mengalami keterlambatan atau fluktuasi kecil akibat dinamika sistem. Dalam kasus ini, Stepper Angle tercatat pada 15°, yang sedikit lebih kecil daripada nilai teoritis yang diharapkan, menunjukkan bahwa sistem mungkin telah menyelesaikan koreksi pada iterasi sebelumnya. Faktor lain yang memengaruhi adalah metode kontrol fuzzy yang digunakan. Sistem kontrol fuzzy sering kali dirancang untuk bekerja dalam batas operasional perangkat keras, dengan mempertimbangkan stabilitas dan keamanan. Dalam beberapa kasus, motor stepper dapat menghasilkan langkah tambahan atau berhenti lebih awal untuk memastikan bahwa Pitch Error terkoreksi secara efektif, terutama jika ada fluktuasi pada Current Pitch. Secara keseluruhan, perbedaan antara hasil perhitungan Crisp Output dan nilai pengukuran mencerminkan kompromi antara teori fuzzy logic dan implementasi perangkat keras. Penyesuaian seperti pembulatan langkah dan waktu eksekusi diperlukan untuk menjaga kompatibilitas sistem dengan motor stepper, sambil memastikan bahwa koreksi Pitch Error dilakukan secara efisien dan aman. Meskipun terdapat deviasi kecil, sistem kontrol ini

menunjukkan kemampuan yang baik dalam menyesuaikan output kontrol dengan batasan fisik perangkat keras.

(Halaman Ini Sengaja Dikosongkan)

BAB 5 KESIMPULAN DAN SARAN

5.1 Kesimpulan

1. Sistem kontrol berbasis Fuzzy Logic Controller (FLC) yang diterapkan pada Autonomous Underwater Vehicle (AUV) menunjukkan kinerja yang baik dalam mengoreksi Yaw Error dan Pitch Error secara bertahap. Sistem berhasil menjaga stabilitas pada Stepper Angle sambil menyesuaikan nilai aktual dengan setpoint target, meskipun terdapat fluktuasi awal akibat dinamika lingkungan bawah air. Hal ini membuktikan kemampuan FLC untuk menangani sistem non-linear dan dinamis dengan adaptasi berbasis aturan linguistik yang dirancang.
2. Analisis parameter Yaw menunjukkan waktu respons awal (Rise Time) yang cukup cepat dengan nilai Overshoot yang dapat dikendalikan, sementara Settling Time mencerminkan kemampuan sistem untuk mencapai stabilitas dalam waktu yang relatif singkat. Pada sisi Pitch, sistem juga menunjukkan stabilitas yang baik, dengan Steady-State Error yang kecil dan koreksi yang progresif. Parameter tambahan seperti Roll tetap stabil selama proses koreksi Yaw dan Pitch, yang mengindikasikan bahwa sistem mampu menjaga orientasi AUV secara keseluruhan.
3. Perubahan Stepper Angle yang terjadi secara bertahap menunjukkan sifat inkremental motor stepper dalam menyesuaikan diri dengan nilai Yaw Error dan Pitch Error. Pendekatan bertahap ini membantu mencegah osilasi yang berlebihan dan menjaga stabilitas sistem. Namun, keterbatasan resolusi langkah motor stepper menyebabkan deviasi kecil antara hasil perhitungan teoritis Crisp Output dan nilai aktual yang tercatat. Resolusi motor sebesar 0.9° per langkah menjadi faktor utama yang memengaruhi presisi koreksi sudut.
4. Hasil pengujian menunjukkan bahwa sistem kontrol fuzzy mampu menjaga stabilitas Yaw, Pitch, dan Roll meskipun dihadapkan pada perubahan lingkungan bawah air yang tidak terprediksi. Sistem berhasil menyesuaikan parameter kontrol secara adaptif untuk mempertahankan stabilitas AUV, yang menjadi indikasi bahwa metode FLC efektif dalam menghadapi dinamika lingkungan perairan yang kompleks.
5. Meskipun sistem kontrol fuzzy menunjukkan kinerja yang memuaskan, terdapat ruang untuk optimasi, terutama dalam mengurangi Overshoot pada parameter Yaw dan mempercepat waktu stabilisasi. Penyesuaian fungsi keanggotaan fuzzy, seperti menambahkan lebih banyak himpunan fuzzy untuk meningkatkan sensitivitas terhadap perubahan, dapat meningkatkan akurasi sistem. Selain itu, peningkatan resolusi langkah motor stepper menjadi prioritas untuk mengurangi deviasi antara hasil perhitungan teoritis dan nilai aktual.

5.2 Saran

1. Perlu dilakukan optimasi lebih lanjut pada pengaturan Fuzzy Logic untuk mengurangi deviasi pada kondisi lintasan dengan kemiringan lebih tinggi dan kecepatan tinggi. Penyesuaian parameter kontrol yang lebih halus dapat meningkatkan kestabilan pada kondisi ekstrem.
2. Melakukan penelitian lebih lanjut mengenai kecepatan optimal AUV untuk manuver yang lebih stabil dan akurat. Kecepatan yang terlalu tinggi dapat mempengaruhi kestabilan, sementara kecepatan yang terlalu rendah dapat memperlambat respon sistem.
3. Meskipun sistem dapat menyesuaikan sudut dengan baik, ada keterlambatan pada transisi sudut yang lebih besar. Penambahan algoritma pengendalian tambahan, seperti PID atau kombinasi dengan Fuzzy Logic, dapat mempercepat respons terhadap perubahan setpoint.
4. Sebaiknya dilakukan pengujian AUV dalam berbagai kondisi lingkungan yang lebih bervariasi, seperti perairan dengan arus atau kedalaman yang berbeda, untuk memastikan kinerja sistem dalam kondisi yang lebih kompleks.
5. Untuk meningkatkan akurasi posisi dan orientasi AUV, disarankan untuk mengintegrasikan sensor tambahan atau sistem navigasi lain, seperti GPS atau sonar, untuk memperoleh data yang lebih lengkap dan akurat dalam pengendalian AUV.

DAFTAR PUSTAKA

Drtil, M., Unkelbach, H., & Bräunl, T. (2006). *UNIVERSITY OF APPLIED SCIENCES KOBLENZ DEPARTMENT OF ELECTRICAL ENGINEERING AND INFORMATION TECHNOLOGY SCHOOL OF ELECTRICAL, ELECTRONIC AND COMPUTER ENGINEERING ELECTRONICS AND SENSOR DESIGN OF AN AUTONOMOUS UNDERWATER VEHICLE*.

Fossen, T. I. (n.d.). *Marine Craft Hydrodynamics and Motion Control Vademecum de Navium Motu Contra Aquas et de Motu Gubernando Marine Craft Hydrodynamics and Motion Control*. www.wiley.com/go/fossen_marine

Intelligent Control and Automation (WCICA), 2010 8th World Congress on : date, 7-9 July 2010. (2010). IEEE.

Liu, Y., Kong, Z., Wang, L., & Liu, Y. (2015). *The design of Self-adaptive Fuzzy PID Controller in Magnetic Suspension Systems*.

Mamdani, E. H., & Assilian, S. (1975). An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller. In *Int. J. Man-Machine Studies* (Vol. 7).

Rizki, I., & Agustinah, T. (2016). Desain Kontrol Tracking Underactuated Autonomous Underwater Vehicle (AUV) dengan Pengaruh Gangguan Arus Laut Tracking Control Design of Underactuated Autonomous Underwater Vehicle (AUV) Under Ocean Current Disturbance Influance. In *Journal of Applied Intelligent System* (Vol. 1, Issue 3).

Yang, Y., Xiao, Y., & Li, T. (2021). A Survey of Autonomous Underwater Vehicle Formation: Performance, Formation Control, and Communication Capability. In *IEEE Communications Surveys and Tutorials* (Vol. 23, Issue 2, pp. 815–841). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/COMST.2021.3059998

Yoerger, D. R., Jakuba, M., Bradley, A. M., & Bingham, B. (n.d.). *Techniques for Deep Sea Near Bottom Survey Using an Autonomous Underwater Vehicle*.

Zadeh, L. A. (1965). Fuzzy S e t s *. In *INFOR~ATIO~ AND CONTROL* (Vol. 8).

(Halaman Ini Sengaja Dikosongkan)

LAMPIRAN

Kode Program Uji Rudder

```
#define enPin1 8
#define dirPin1 9
#define stepPin1 10


#define enPin2 11
#define dirPin2 12
#define stepPin2 13

const int stepsPerRevolution = 400;
float degreesPerStep = 360.0 / stepsPerRevolution;
int delayMS = 1000; // Microseconds delay between steps


void setup() {
  // Konfigurasi pin untuk motor pertama
  pinMode(dirPin1, OUTPUT);
  pinMode(stepPin1, OUTPUT);
  pinMode(enPin1, OUTPUT);
  digitalWrite(enPin1, LOW);


  // Konfigurasi pin untuk motor kedua
  pinMode(dirPin2, OUTPUT);
  pinMode(stepPin2, OUTPUT);
  pinMode(enPin2, OUTPUT);
  digitalWrite(enPin2, LOW);


  Serial.begin(9600);
  Serial.println("Masukkan nilai derajat untuk motor 1 dan motor 2 (format: deg1,deg2):");
}


void loop() {
  if (Serial.available() > 0) {
    // Baca input dari serial monitor
    String input = Serial.readStringUntil('\n');
    int commaIndex = input.indexOf(',');
    if (commaIndex == -1) {
      Serial.println("Format salah! Gunakan format: deg1,deg2");
      return;
    }
```

```
float targetDegree1 = input.substring(0, commaIndex).toFloat();
float targetDegree2 = input.substring(commaIndex + 1).toFloat();


Serial.print("Motor 1 bergerak: ");
Serial.print(targetDegree1);
Serial.println(" derajat");
Serial.print("Motor 2 bergerak: ");
Serial.print(targetDegree2);
Serial.println(" derajat");


// Tentukan arah untuk motor 1
if (targetDegree1 > 0) {
  digitalWrite(dirPin1, LOW);
} else {
  digitalWrite(dirPin1, HIGH);
  targetDegree1 = -targetDegree1;
}


// Tentukan arah untuk motor 2
if (targetDegree2 > 0) {
  digitalWrite(dirPin2, LOW);
} else {
  digitalWrite(dirPin2, HIGH);
  targetDegree2 = -targetDegree2;
}


// Hitung langkah yang harus ditempuh masing-masing motor
int stepsToMove1 = round(targetDegree1 / degreesPerStep);
int stepsToMove2 = round(targetDegree2 / degreesPerStep);


// Gerakkan kedua motor secara bersamaan
int maxSteps = max(stepsToMove1, stepsToMove2);
for (int i = 0; i < maxSteps; i++) {
  // Gerakkan motor 1 jika masih ada langkah yang tersisa
  if (i < stepsToMove1) {
    digitalWrite(stepPin1, HIGH);
    delayMicroseconds(delayMS);
    digitalWrite(stepPin1, LOW);
    delayMicroseconds(delayMS);
  }


  // Gerakkan motor 2 jika masih ada langkah yang tersisa
  if (i < stepsToMove2) {
```

```
      digitalWrite(stepPin2, HIGH);
      delayMicroseconds(delayMS);
      digitalWrite(stepPin2, LOW);
      delayMicroseconds(delayMS);
     }
    }


    delay(500); // Jeda sejenak setelah motor bergerak
  }
}
```

Kode Program untuk manuver Yaw

```
#include "DFRobot_BNO055.h"
#include "Wire.h"

// Konfigurasi motor stepper
#define enPin1 2
#define dirPin1 3
#define stepPin1 4
#define enPin2 5
#define dirPin2 6
#define stepPin2 7

const int stepsPerRevolution = 400;
const float degreesPerStep = 360.0 / stepsPerRevolution;

// Konfigurasi propeller
#define propellerPin 22

// Konfigurasi sensor IMU
typedef DFRobot_BNO055_IIC BNO;
BNO bno(&Wire, 0x28);

// Variabel global
float setpointYaw = 0;          // Setpoint Yaw dari Serial Monitor
bool propellerActive = false;  // Status propeller aktif atau tidak
bool stepperEnabled = true;    // Status stepper aktif atau tidak
const float yawThreshold = 2.0; // Toleransi error Yaw (dalam derajat)
float currentStepperAngle = 0;  // Posisi sudut stepper saat ini
const float maxStepperAngleRight = 180.0; // Batas maksimum sudut stepper kanan
const float maxStepperAngleLeft = -180.0; // Batas maksimum sudut stepper kiri

// Fungsi untuk normalisasi yaw ke rentang -180° hingga 180°
float normalizeYaw(float yaw) {
  while (yaw > 180.0) yaw -= 360.0;
  while (yaw < -180.0) yaw += 360.0;
  return yaw;
}

// Fungsi untuk membaca data yaw dari IMU
float getCurrentYaw() {
  BNO::sEulAnalog_t eulData = bno.getEul();
  return normalizeYaw(eulData.head);
}

// Fungsi untuk membaca data roll, pitch, yaw
void getOrientation(float &roll, float &pitch, float &yaw) {
  BNO::sEulAnalog_t eulData = bno.getEul();
```

```
  roll = -eulData.roll;
  pitch = eulData.pitch;
  yaw = normalizeYaw(eulData.head);
}

// Fungsi untuk membaca data akselerasi
void getAcceleration(float &ax, float &ay, float &az) {
  BNO::sAxisAnalog_t accelData = bno.getAxis(BNO::eAxisAcc);
  ax = accelData.x;
  ay = accelData.y;
  az = accelData.z;
}

// Fungsi untuk membaca data gyro
void getGyro(float &gx, float &gy, float &gz) {
  BNO::sAxisAnalog_t gyroData = bno.getAxis(BNO::eAxisGyr);
  gx = gyroData.x;
  gy = gyroData.y;
  gz = gyroData.z;
}

// Fungsi untuk menghitung membership function trapezoidal
float trapezoidalMF(float x, float a, float b, float c, float d) {
  if (x <= a || x >= d) return 0.0;
  else if (x >= b && x <= c) return 1.0;
  else if (x > a && x < b) return (x - a) / (b - a);
  else return (d - x) / (d - c);
}

// Fungsi untuk menghitung langkah motor berdasarkan kontrol fuzzy
int calculateSteps(float yawError) {
  // Membership functions
  float VL = trapezoidalMF(yawError, -200, -150, -120, -60);
  float L = trapezoidalMF(yawError, -120, -90, -60, -20);
  float M = trapezoidalMF(yawError, -60, -40, -20, 0);
  float Z = trapezoidalMF(yawError, -20, -10, 10, 20);
  float P = trapezoidalMF(yawError, 0, 20, 40, 60);
  float H = trapezoidalMF(yawError, 20, 60, 90, 120);
  float VH = trapezoidalMF(yawError, 60, 120, 150, 200);



  // Output rules
  float output =
      VL * -120 + L * -90 + M * -60 + Z * 0 + P * 60 + H * 90 + VH * 120;
  float weight = VL + L + M + Z + P + H + VH;

  return weight > 0 ? round(output / weight) : 0;
}
```

```
void moveRudder(int stepsToMove) {
  if (stepsToMove == 0 || !stepperEnabled) return; // Tidak ada langkah atau stepper off

  bool direction = stepsToMove > 0 ? LOW : HIGH;
  stepsToMove = abs(stepsToMove);

  // Set arah motor
  digitalWrite(dirPin1, direction);
  digitalWrite(dirPin2, direction);

  // Sinkronisasi gerakan motor dengan smoothing
  for (int i = 0; i < stepsToMove; i++) {
    digitalWrite(stepPin1, HIGH);
    digitalWrite(stepPin2, HIGH);
    delayMicroseconds(500); // Durasi pulsa stepper (lebih lambat untuk halus)
    digitalWrite(stepPin1, LOW);
    digitalWrite(stepPin2, LOW);
    delayMicroseconds(500); // Interval antar pulsa

    // Perbarui sudut stepper
    currentStepperAngle += (direction == LOW ? degreesPerStep : -degreesPerStep);

    // Cek batas maksimum sudut stepper kanan
    if (currentStepperAngle > maxStepperAngleRight) {
      stepperEnabled = false; // Matikan stepper jika mencapai batas kanan
      currentStepperAngle = maxStepperAngleRight; // Tetapkan sudut maksimum kanan
      // Serial.println("Stepper angle mencapai batas maksimum kanan (+180).");
      return;
    }

    // Cek batas maksimum sudut stepper kiri
    if (currentStepperAngle < maxStepperAngleLeft) {
      stepperEnabled = false; // Matikan stepper jika mencapai batas kiri
      currentStepperAngle = maxStepperAngleLeft; // Tetapkan sudut maksimum kiri
      // Serial.println("Stepper angle mencapai batas maksimum kiri (-180).");
      return;
    }
  }

  // Logging derajat stepper setelah langkah diberikan
  // Serial.print("Stepper Angle Saat Ini: ");
  // Serial.println(currentStepperAngle);
}

void setup() {
  // Konfigurasi motor rudder
  pinMode(dirPin1, OUTPUT);
  pinMode(stepPin1, OUTPUT);
```

```
  pinMode(enPin1, OUTPUT);
  digitalWrite(enPin1, LOW);

  pinMode(dirPin2, OUTPUT);
  pinMode(stepPin2, OUTPUT);
  pinMode(enPin2, OUTPUT);
  digitalWrite(enPin2, LOW);

  // Konfigurasi propeller
  pinMode(propellerPin, OUTPUT);
  digitalWrite(propellerPin, HIGH); // Propeller selalu aktif

  // Konfigurasi sensor IMU
  Serial.begin(115200);
  bno.reset();
  while (bno.begin() != BNO::eStatusOK) {
    Serial.println("BNO055 gagal diinisialisasi! Coba ulang...");
    delay(1000);
  }
  Serial.println("BNO055 berhasil diinisialisasi!");
  Serial.println("Masukkan setpoint Yaw:");
}

void loop() {
  if (Serial.available() > 0) {
    // Baca setpoint baru
    String input = Serial.readStringUntil('\n');
    setpointYaw = input.toFloat();
  }

  unsigned long currentMillis = millis();
  Serial.print("Millis: ");
  Serial.print(currentMillis);

  // Baca Yaw
  float currentYaw = getCurrentYaw();
  float yawError = normalizeYaw(setpointYaw - currentYaw);

  // Logging semua data
  Serial.print(", Setpoint Yaw: ");
  Serial.print(setpointYaw);
  Serial.print(", Current Yaw: ");
  Serial.print(currentYaw);
  Serial.print(", Yaw Error: ");
  Serial.print(yawError);

  float roll, pitch, yaw;
  getOrientation(roll, pitch, yaw);
  Serial.print(", Roll: ");
```

```
  Serial.print(roll);
  Serial.print(", Pitch: ");
  Serial.print(pitch);
  Serial.print(", Yaw: ");
  Serial.print(yaw);

  float ax, ay, az;
  getAcceleration(ax, ay, az);
  Serial.print(", Acceleration X: ");
  Serial.print(ax);
  Serial.print(", Acceleration Y: ");
  Serial.print(ay);
  Serial.print(", Acceleration Z: ");
  Serial.print(az);

  float gx, gy, gz;
  getGyro(gx, gy, gz);
  Serial.print(", Gyro X: ");
  Serial.print(gx);
  Serial.print(", Gyro Y: ");
  Serial.print(gy);
  Serial.print(", Gyro Z: ");
  Serial.print(gz);

  Serial.print(", Stepper Angle: ");
  Serial.print(currentStepperAngle);
  Serial.print(", Stepper Enabled: ");
  Serial.println(stepperEnabled ? "ON" : "OFF");

  // Kontrol rudder
  if (abs(yawError) > yawThreshold) {
    stepperEnabled = true; // Aktifkan stepper saat di luar threshold
    int stepsToMove = calculateSteps(yawError);
    moveRudder(stepsToMove);
  } else {
    if (stepperEnabled) {
      // Serial.println("Setpoint tercapai, stepper dimatikan.");
      stepperEnabled = false; // Matikan stepper saat setpoint tercapai
    }
  }
}
```

Kode Program untuk Manunver Pitch

```
#include "DFRobot_BNO055.h"
#include "Wire.h"

// Konfigurasi motor stepper
#define enPin1 8
#define dirPin1 9
#define stepPin1 10
#define enPin2 11
#define dirPin2 12
#define stepPin2 13

const int stepsPerRevolution = 400;
const float degreesPerStep = 360.0 / stepsPerRevolution;

// Konfigurasi propeller
#define propellerPin 22

// Konfigurasi sensor IMU
typedef DFRobot_BNO055_IIC BNO;
BNO bno(&Wire, 0x28);

// Variabel global
float setpointPitch = 0;       // Setpoint Pitch dari Serial Monitor
bool propellerActive = false;  // Status propeller aktif atau tidak
bool stepperEnabled = true;    // Status stepper aktif atau tidak
const float pitchThreshold = 2.0; // Toleransi error Pitch (dalam derajat)
float currentStepperAngle = 0;  // Posisi sudut stepper saat ini
const float maxStepperAngleRight = 180.0; // Batas maksimum sudut stepper kanan
const float maxStepperAngleLeft = -180.0; // Batas maksimum sudut stepper kiri

// Fungsi untuk normalisasi pitch ke rentang -180° hingga 180°
float normalizePitch(float pitch) {
  while (pitch > 180.0) pitch -= 360.0;
  while (pitch < -180.0) pitch += 360.0;
  return pitch;
}

// Fungsi untuk membaca data pitch dari IMU
float getCurrentPitch() {
  BNO::sEulAnalog_t eulData = bno.getEul();
  return normalizePitch(-eulData.roll); // Balikkan tanda roll menjadi pitch
}

// Fungsi untuk membaca data roll, pitch, yaw
void getOrientation(float &roll, float &pitch, float &yaw) {
  BNO::sEulAnalog_t eulData = bno.getEul();
  roll = -eulData.pitch; // Balikkan pitch menjadi roll
```

```
  pitch = normalizePitch(-eulData.roll); // Balikkan roll menjadi pitch
  yaw = normalizePitch(eulData.head);
}

// Fungsi untuk membaca data akselerasi
void getAcceleration(float &ax, float &ay, float &az) {
  BNO::sAxisAnalog_t accelData = bno.getAxis(BNO::eAxisAcc);
  ax = accelData.x;
  ay = accelData.y;
  az = accelData.z;
}

// Fungsi untuk membaca data gyro
void getGyro(float &gx, float &gy, float &gz) {
  BNO::sAxisAnalog_t gyroData = bno.getAxis(BNO::eAxisGyr);
  gx = gyroData.x;
  gy = gyroData.y;
  gz = gyroData.z;
}

// Fungsi untuk menghitung membership function trapezoidal
float trapezoidalMF(float x, float a, float b, float c, float d) {
  if (x <= a || x >= d) return 0.0;
  else if (x >= b && x <= c) return 1.0;
  else if (x > a && x < b) return (x - a) / (b - a);
  else return (d - x) / (d - c);
}

// Fungsi untuk menghitung langkah motor berdasarkan kontrol fuzzy
int calculateSteps(float pitchError) {
  // Membership functions
  float VL = trapezoidalMF(pitchError, -200, -150, -120, -60);
  float L = trapezoidalMF(pitchError, -120, -90, -60, -20);
  float M = trapezoidalMF(pitchError, -60, -40, -20, 0);
  float Z = trapezoidalMF(pitchError, -20, -10, 10, 20);
  float P = trapezoidalMF(pitchError, 0, 20, 40, 60);
  float H = trapezoidalMF(pitchError, 20, 60, 90, 120);
  float VH = trapezoidalMF(pitchError, 60, 120, 150, 200);

  // Output rules
  float output =
     VL * -120 + L * -90 + M * -60 + Z * 0 + P * 60 + H * 90 + VH * 120;
  float weight = VL + L + M + Z + P + H + VH;

  return weight > 0 ? round(output / weight) : 0;
}

void moveRudder(int stepsToMove) {
  if (stepsToMove == 0 || !stepperEnabled) return; // Tidak ada langkah atau stepper off
```

```
  bool direction = stepsToMove > 0 ? LOW : HIGH;
  stepsToMove = abs(stepsToMove);

  // Set arah motor
  digitalWrite(dirPin1, direction);
  digitalWrite(dirPin2, direction);

  // Sinkronisasi gerakan motor dengan smoothing
  for (int i = 0; i < stepsToMove; i++) {
    digitalWrite(stepPin1, HIGH);
    digitalWrite(stepPin2, HIGH);
    delayMicroseconds(500); // Durasi pulsa stepper (lebih lambat untuk halus)
    digitalWrite(stepPin1, LOW);
    digitalWrite(stepPin2, LOW);
    delayMicroseconds(500); // Interval antar pulsa

    // Perbarui sudut stepper
    currentStepperAngle += (direction == LOW ? degreesPerStep : -degreesPerStep);

    // Cek batas maksimum sudut stepper kanan
    if (currentStepperAngle > maxStepperAngleRight) {
      stepperEnabled = false; // Matikan stepper jika mencapai batas kanan
      currentStepperAngle = maxStepperAngleRight; // Tetapkan sudut maksimum kanan
      // Serial.println("Stepper angle mencapai batas maksimum kanan (+180).");
      return;
    }

    // Cek batas maksimum sudut stepper kiri
    if (currentStepperAngle < maxStepperAngleLeft) {
      stepperEnabled = false; // Matikan stepper jika mencapai batas kiri
      currentStepperAngle = maxStepperAngleLeft; // Tetapkan sudut maksimum kiri
      // Serial.println("Stepper angle mencapai batas maksimum kiri (-180).");
      return;
    }
  }

  // Logging derajat stepper setelah langkah diberikan
  // Serial.print("Stepper Angle Saat Ini: ");
  // Serial.println(currentStepperAngle);
}

void setup() {
  // Konfigurasi motor rudder
  pinMode(dirPin1, OUTPUT);
  pinMode(stepPin1, OUTPUT);
  pinMode(enPin1, OUTPUT);
  digitalWrite(enPin1, LOW);
```

```
  pinMode(dirPin2, OUTPUT);
  pinMode(stepPin2, OUTPUT);
  pinMode(enPin2, OUTPUT);
  digitalWrite(enPin2, LOW);

  // Konfigurasi propeller
  pinMode(propellerPin, OUTPUT);
  digitalWrite(propellerPin, HIGH); // Propeller selalu aktif

  // Konfigurasi sensor IMU
  Serial.begin(115200);
  bno.reset();
  while (bno.begin() != BNO::eStatusOK) {
    Serial.println("BNO055 gagal diinisialisasi! Coba ulang...");
    delay(1000);
  }
  Serial.println("BNO055 berhasil diinisialisasi!");
  Serial.println("Masukkan setpoint Pitch:");
}

void loop() {
  if (Serial.available() > 0) {
    // Baca setpoint baru
    String input = Serial.readStringUntil('\n');
    setpointPitch = input.toFloat();
  }

  unsigned long currentMillis = millis();
  Serial.print("Millis: ");
  Serial.print(currentMillis);

  // Baca Pitch
  float currentPitch = getCurrentPitch();
  float pitchError = normalizePitch(setpointPitch - currentPitch);

  // Logging semua data
  Serial.print(", Setpoint Pitch: ");
  Serial.print(setpointPitch);
  Serial.print(", Current Pitch: ");
  Serial.print(currentPitch);
  Serial.print(", Pitch Error: ");
  Serial.print(pitchError);

  float roll, pitch, yaw;
  getOrientation(roll, pitch, yaw);
  Serial.print(", Roll: ");
  Serial.print(roll);
  Serial.print(", Pitch: ");
  Serial.print(pitch);
```

```
  Serial.print(", Yaw: ");
  Serial.print(yaw);

  float ax, ay, az;
  getAcceleration(ax, ay, az);
  Serial.print(", Acceleration X: ");
  Serial.print(ax);
  Serial.print(", Acceleration Y: ");
  Serial.print(ay);
  Serial.print(", Acceleration Z: ");
  Serial.print(az);

  float gx, gy, gz;
  getGyro(gx, gy, gz);
  Serial.print(", Gyro X: ");
  Serial.print(gx);
  Serial.print(", Gyro Y: ");
  Serial.print(gy);
  Serial.print(", Gyro Z: ");
  Serial.print(gz);

  Serial.print(", Stepper Angle: ");
  Serial.print(currentStepperAngle);
  Serial.print(", Stepper Enabled: ");
  Serial.println(stepperEnabled ? "ON" : "OFF");

  // Kontrol rudder
  if (abs(pitchError) > pitchThreshold) {
    stepperEnabled = true; // Aktifkan stepper saat di luar threshold
    int stepsToMove = calculateSteps(pitchError);
    moveRudder(stepsToMove);
  } else {
    if (stepperEnabled) {
      // Serial.println("Setpoint tercapai, stepper dimatikan.");
      stepperEnabled = false; // Matikan stepper saat setpoint tercapai
    }
  }
}
```

Lampiran sebenarnya masih banyak, namun hanya disimpan oleh peneliti, jika membutuhkan detail lampiran/program yang masih belum saya lampirkan disini, bisa menghubungi ilhamwep.its@gmail.com

BIODATA PENULIS

Penulis dilahirkan di Mojokerto, 27 April 2001, merupakan anak pertama dari 3 bersaudara. Penulis telah menempuh pendidikan formal yaitu di SDN Canggu 1, SMPN 2 Jetis, dan SMKN 1 Jetis, Mojokerto. Setelah lulus dari SMKN 1 Jetis tahun 2020, Penulis mengikuti SNMPTN dan diterima di Departemen Teknik Elektro FTEIC - ITS dan terdaftar dengan NRP 5022201041.

Di Departemen Teknik Elektro ITS, Penulis pernah berpindah bidang studi hingga 3 kali, Pada semester 5, penulis mengambil bidang studi Teknik Sistem Tenaga, pada semester 6, penulis mengambil bidang studi Teknik Elektronika, dan pada semester 7 hingga lulus semester 9, memutuskan untuk mengambil bidang studi Teknik Sistem Kontrol. Dalam keilmuan Sistem Kontrol, penulis belajar banyak hal yang bisa diterapkan di dunia industri maupun kehidupan. Bertemu bertemu dosen yang sangat baik bernama Pak Dr. Ir. Ari Santoso, DEA, selaku pembimbing Tugas Akhir saya. Di ITS, selain menempuh pendidikan akademik, penulis juga sangat aktif mengikuti kompetisi inovasi dan bisnis, sehingga banyak mendapatkan penghargaan dalam bidang inovasi dan bisnis. Jika ada pertanyaan mengenai penelitian ini, bisa hubungi : ilhamwep.its@gmail.com