

TUGAS AKHIR - TM184730

**PERANCANGAN SISTEM KONTROL *FUZZY LOGIC* DENGAN
MENGUNAKAN SIMULINK PADA BLUEROV 2 UNTUK
SIMULASI INSPEKSI KEADAAN LAMBUNG KAPAL PADA
LINGKUNGAN PERAIRAN AIR LAUT**

Muhammad Reza Farras
NRP 02111840000188

Dosen Pembimbing
Arif Wahjudi, S.T., M.T., Ph.D
NIP 197303222001121001

DEPARTEMEN TEKNIK MESIN
Fakultas Teknologi Industri dan Rekayasa Sistem
Institut Teknologi Sepuluh Nopember
Surabaya 2022



TUGAS AKHIR - TM184730

**PERANCANGAN SISTEM KONTROL *FUZZY LOGIC* DENGAN
MENGUNAKAN SIMULINK PADA BLUEROV 2 UNTUK
SIMULASI INSPEKSI KEADAAN LAMBUNG KAPAL PADA
LINGKUNGAN PERAIRAN AIR LAUT**

**Muhammad Reza Farras
NRP 02111840000188**

Dosen Pembimbing
Arif Wahjudi, S.T., M.T., Ph.D
NIP 197303222001121001

DEPARTEMEN TEKNIK MESIN
Fakultas Teknologi Industri dan Rekayasa Sistem
Institut Teknologi Sepuluh Nopember
Surabaya 2022

Halaman ini sengaja dikosongkan



FINAL PROJECT - TM184730

**DESIGN OF FUZZY LOGIC CONTROL SYSTEM USING SIMULINK
ON BLUEROV 2 FOR INSPECTION SIMULATION OF A HULL
SHIP IN THE SEAWATER ENVIRONMENT**

**Muhammad Reza Farras
NRP 02111840000188**

Supervisor
**Arif Wahjudi, S.T., M.T., Ph.D
NIP 197303222001121001**

DEPARTEMENT OF MECHANICAL ENGINEERING
Faculty of Industrial Technology and System Engineering
Institut Teknologi Sepuluh Nopember
Surabaya 2022

Halaman ini sengaja dikosongkan

LEMBAR PENGESAHAN




PERANCANGAN SISTEM KONTROL *FUZZY LOGIC* DENGAN MENGGUNAKAN SIMULINK PADA BLUEROV 2 UNTUK SIMULASI INSPEKSI KEADAAN LAMBUNG KAPAL PADA LINGKUNGAN PERAIRAN AIR LAUT

TUGAS AKHIR

Diajukan untuk memenuhi salah satu syarat
memperoleh gelar sarjana pada
Program Studi S-1 Teknik Mesin
Departemen Teknik Mesin
Fakultas Teknologi Industri dan Rekayasa Sistem
Institut Teknologi Sepuluh Nopember

Oleh : Muhammad Reza Farras
NRP. 02111840000188

Disetujui oleh Tim Penguji Tugas Akhir :

- | | | |
|---|--|------------|
| 1. Arif Wahjudi, S.T., M.T., Ph.D |  | Pembimbing |
| 2. Prof. Dr. Ing. I Made Londen Batan, ME |  | Penguji I |
| 3. Dinny Harnany, S.T., M.Sc |  | Penguji II |

SURABAYA
Juli, 2022

Halaman ini sengaja dikosongkan

APPROVAL SHEET

DESIGN OF FUZZY LOGIC CONTROL SYSTEM USING SIMULINK ON BLUEROV 2 FOR INSPECTION SIMULATION OF A HULL SHIP IN THE SEAWATER ENVIRONMENT




FINAL PROJECT

Submitted to fulfill one of the requirements
for obtaining a degree Bachelor Engineering at
Undergraduate Study Program of Mechanical Engineering
Department of Mechanical Engineering
Faculty of Industrial Technology and System Engineering
Institut Teknologi Sepuluh Nopember

By: Muhammad Reza Farras

NRP. 02111840000188

Approved by Final Project Examiner Team:

1. Arif Wahjudi, S.T., M.T., Ph.D  Advisor
2. Prof. Dr. Ing. I Made Londen Batan, ME  Examiner I
3. Dinny Harnany, S.T., M.Sc  Examiner II

SURABAYA
July, 2022

Halaman ini sengaja dikosongkan

PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini:

Nama mahasiswa / NRP : Muhammad Reza Farras / 02111840000188
Program studi : S1 Teknik Mesin
Dosen Pembimbing / NIP : Arif Wahjudi, S.T., M.T., Ph.D / 197303222001121001

Dengan ini menyatakan bahwa Tugas Akhir dengan judul “Perancangan Sistem Kontrol Fuzzy Logic dengan Menggunakan Simulink pada BlueROV 2 untuk Simulasi Inspeksi Keadaan Lambung Kapal pada Lingkungan Perairan Air Laut” 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.

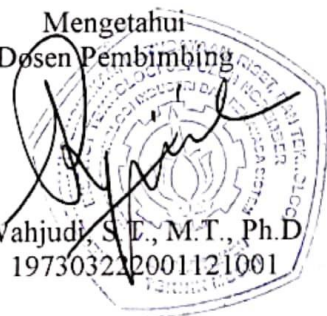
Surabaya, 8 Juli 2022

Mahasiswa



Muhammad Reza Farras
NRP. 02111840000188

Mengetahui
Dosen Pembimbing



Arif Wahjudi, S.T., M.T., Ph.D
NIP. 197303222001121001

Halaman ini sengaja dikosongkan

STATEMENT OF ORIGINALITY

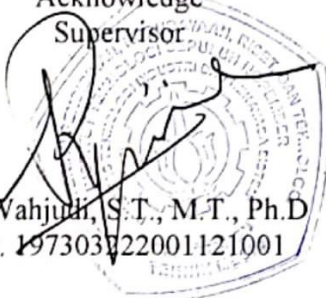
The undersigned below:

Name of student / NRP : Muhammad Reza Farras / 02111840000188
Department : Mechanical Engineering
Advisor / NIP : Arif Wahjudi, S.T., M.T., Ph.D / 197303222001121001

Hereby declare that Final Project with the title of “Design of Fuzzy Logic Control System Using Simulink on BlueROV 2 For Inspection Simulation of a Hull Ship in The Seawater Environment” is the result of my own work, is original, and is written by following the rules of scientific writing.

If in the future there is a discrepancy with this statement, then I am willing to accept sanctions in accordance with the provisions that apply at Institut Teknologi Sepuluh Nopember.

Acknowledge
Supervisor



Arif Wahjudi, S.T., M.T., Ph.D
NIP. 197303222001121001

Surabaya, 8 July 2022

Student



Muhammad Reza Farras
NRP. 02111840000188

Halaman ini sengaja dikosongkan

PERANCANGAN SISTEM KONTROL *FUZZY LOGIC* DENGAN MENGGUNAKAN SIMULINK PADA BLUE ROV 2 UNTUK SIMULASI INSPEKSI KEADAAN LAMBUNG KAPAL PADA LINGKUNGAN PERAIRAN AIR LAUT

Nama Mahasiswa : Muhammad Reza Farras
NRP : 0211184000188
Jurusan : Teknik Mesin FT-IRS ITS
Dosen Pembimbing : Arif Wahjudi ST., MT., PhD

ABSTRAK

Indonesia merupakan negara yang luas wilayahnya lebih dari 60% merupakan perairan, hal tersebut menjadi potensi besar bagi negara Indonesia dari segi kemaritiman. Penggunaan kapal untuk aktifitas di perairan menjadi hal yang sangat sering dijumpai terutama di lingkungan air laut. Semua mesin termasuk kapal pasti memerlukan perawatan secara berkala untuk menjaga performanya, termasuk perawatan dari lambung kapal. Selama ini inspeksi dari lambung kapal masih dilakukan secara manual dengan menggunakan penyelam. Oleh karena itu penelitian ini bertujuan untuk merancang sistem kontrol yang dibenamkan pada *remotely operated vehicle* (ROV) untuk misi melakukan inspeksi lambung kapal.

Pada penelitian ini dilakukan pemodelan *remotely operated vehicle*, *thruster*, dan *kontroller* dengan menggunakan bantuan simulasi *software* Simulink pada MATLAB. Pemodelan dilakukan dengan memasukkan parameter yang diketahui pada saat studi literatur. Perancangan *fuzzy logic controller* dimulai dengan mendesain *membership function* dan *rule base*. Pada proses simulasi akan dilakukan secara bertahap, mulai dari pengujian *remotely operated vehicle* pada kondisi *steady state* dan jika telah berhasil dilanjutkan dengan simulasi kontrol kedalaman dan *path tracking* lambung kapal dengan menggunakan *fuzzy logic controller* sebagai *kontroller* yang mengatur pergerakan dari *remotely operated vehicle*.

Dari penelitian yang telah dilakukan, dihasilkan tiga buah *fuzzy logic controller* untuk mengontrol pergerakan ke orientasi *surge*, *heave* dan *yaw*. Hasil menunjukkan bahwa simulasi sistem kontrol BlueROV 2 dapat mengikuti trayektori yang diinginkan meskipun memiliki keterbatasan tidak bisa digerakkan ke arah kiri. Simulasi dilakukan dua kali dengan cara meletakkan BlueROV 2 di titik koordinat ujung lambung kapal, hasil simulasi skenario pertama didapatkan hasil nilai *error* maksimum untuk kontrol *heave* sebesar -0.093138855 m dan minimum sebesar -0.00542276 m. Nilai *root square error* maksimum untuk yang dilakukan pada simulasi I berturut turut sebesar 2.67926 m dan minimum sebesar 0.261849 m. Sedangkan untuk simulasi skenario kedua didapatkan nilai *error* maksimum untuk kontrol *heave* yang dilakukan pada simulasi II sebesar -0.09304212 m dan minimum sebesar -0.00540397 m. Nilai *root square error* maksimum untuk yang dilakukan pada simulasi II sebesar 2.516389 m dan minimum sebesar 0.15128 m.

Kata kunci: *Remotely Operated Vehicle* (ROV), *Fuzzy Logic Controller* (FLC), Inspeksi Lambung Kapal, Simulink, BlueROV 2

Halaman ini sengaja dikosongkan

DESIGN OF FUZZY LOGIC CONTROL SYSTEM USING SIMULINK ON BLUE ROV 2 FOR INSPECTION SIMULATION OF A HULL SHIP IN THE SEAWATER ENVIRONMENT

Name : Muhammad Reza Farras
NRP : 02111840000188
Department : Mechanical Engineering INDSYS
Supervisor : Arif Wahjudi ST., MT., PhD

ABSTRACT

Indonesia is a country whose territory is more than 60% water, this is a big potential for the Indonesian state from a maritime perspective. The use of ships for activities in the waters is very common in the marine environment. All engines including ships definitely require regular maintenance to maintain their performance, including maintenance of the hull. So far, inspection of the ship's hull is still done manually using rescue. Therefore, this study aims to design a control system embedded in the remotely operated vehicle (ROV) for the mission of conducting hull inspections.

In this study, remotely operated vehicle, thruster, and controller modeling were carried out using the Simulink simulation software on MATLAB. Modeling is done by entering parameters that are known at the time of literature study. The fuzzy logic controller designing will begin with designing the membership function and rule base. The simulation process will be carried out in stages, starting from testing the remotely operated vehicle at steady state conditions and if successful, continuing with the simulation of depth control and hull tracking using a fuzzy logic controller as a control that regulates the movement of the remotely operated vehicle.

From the research that has been done, three fuzzy logic controllers are produced to control the movement towards the surge, heave and yaw orientations. The results show that the simulation of the BlueROV 2 control system can follow the desired trajectory even though it has the limitation of not being able to move to the left. The simulation was carried out twice by placing BlueROV 2 at the coordinates of the end of the hull, the simulation results for the first scenario obtained the maximum error value for heave control of -0.093138855 m and a minimum of -0.00542276 m. The maximum root square error value for the simulation I was 2.67926 m and the minimum was 0.261849 m, respectively. Meanwhile, for the second scenario simulation, the maximum error value for heave control carried out in simulation II is -0.09304212 m and the minimum is -0.00540397 m. The maximum root square error value for the simulation II is 2.516389 m and the minimum is 0.15128 m.

Keywords: *Remotely Operated Vehicle (ROV), Fuzzy Logic Controller (FLC), Ship Hull Inspection, Simulink, BlueROV 2*

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Dengan mengucapkan puji syukur kehadirat Allah SWT yang telah melimpahkan rahmat, taufiq, dan hidayah-Nya sehingga penulis dapat menyelesaikan tugas akhir dan menyusun laporan tugas akhir yang berjudul “Perancangan Sistem Kontrol Fuzzy Logic dengan Menggunakan Simulink pada BlueROV 2 untuk Simulasi Inspeksi Keadaan Lambung Kapal pada Lingkungan Perairan Air Laut”.

Laporan tugas akhir ini merupakan salah satu bentuk aplikasi dari teori yang telah didapatkan di bangku kuliah yang selanjutnya diharapkan dapat menjadi bekal ilmu yang bermanfaat di dunia kerja. Laporan tugas akhir ini disusun sebagai salah satu syarat untuk menyelesaikan studi strata satu dan memperoleh gelar sarjana Teknik Mesin, Institut Teknologi Sepuluh Nopember Surabaya.

Penulisan laporan tugas akhir ini tidak lepas dari bantuan, bimbingan, dan motivasi yang kuat dari berbagai pihak. Oleh karena itu penulis mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Allah SWT yang telah memberikan kesehatan dan keselamatan
2. Keluarga tercinta atas semua doa, dukungan dan kasih sayang selama ini.
3. Bapak Arif Wahjudi, S.T., M.T., Ph.D, selaku dosen pembimbing yang memberikan masukan dan bimbingan kepada penulis.
4. Segenap dosen Teknik Mesin ITS atas ilmu yang telah diberikan kepada penulis selama menuntut ilmu di Teknik Mesin.
5. Angkatan M61 yang telah memberikan pengalaman, bantuan dan suka-duka selama menuntut ilmu di Teknik Mesin ITS.
6. Rekan-rekan FX group, ParBang group dan TSM Surabaya, yang senantiasa memberikan motivasi dan semangat serta persahabatan yang erat.
7. Adit, Aul, Aziz dan Revi yang telah menemani masa sulit selama mengerjakan tugas akhir ini.

Akhir kata penulis mohon maaf apabila dalam penulisan laporan ini terdapat kata dan kalimat yang salah. Saran dan kritik akan penulis terima dengan hati yang lapang. Semoga laporan ini dapat memberikan manfaat bagi semuanya.

Surabaya, Juli 2022

Penulis

Halaman ini sengaja dikosongkan

DAFTAR ISI

HALAMAN JUDUL.....	i
COVER PAGE.....	iii
LEMBAR PENGESAHAN.....	v
APPROVAL SHEET.....	vii
PERNYATAAN ORISINALITAS	ix
STATEMENT OF ORIGINALITY	xi
ABSTRAK	xiii
ABSTRACT.....	xv
KATA PENGANTAR	xvii
DAFTAR ISI.....	xix
DAFTAR GAMBAR	xxi
DAFTAR TABEL.....	xxiii
BAB I	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan Masalah	2
1.4 Batasan Penelitian	2
1.5 Manfaat Penelitian	2
1.5.1 Bagi Penulis	2
1.5.2 Bagi Pembaca dan Pihak Terkait	2
BAB II.....	3
2.1 Penelitian Terdahulu	3
2.2 Dasar Teori.....	5
2.2.1 <i>Remotely Operated Vehicle (ROV)</i>	5
2.2.1.1 Notasi	5
2.2.1.2 <i>Frame of Reference</i>	6
2.2.1.3 Parameter BlueROV2.....	7
2.2.2 <i>ROV Motions</i>	8
2.2.2.1 <i>ROV Equations of Motions</i>	9
2.2.2.2 Model Hidrodinamis BlueROV 2	9
2.2.3 <i>Fuzzy Logic Controller</i>	10
2.2.3.1 <i>Membership Function</i>	12
2.2.4 <i>Thruster Model</i>	13
2.2.4.1 Pendorong T200.....	14
2.2.5 Profil Bentuk Lambung Kapal.....	14

2.2.6 Perencanaan Trayektori Bentuk Lambung Kapal	15
BAB III.....	17
3.1 <i>Flowchart</i> Penelitian	17
3.2 Langkah – Langkah Penelitian.....	18
3.2.1 Perumusan Masalah dan Tujuan Penelitian	18
3.2.2 Studi Literatur.....	18
3.2.3 Pemodelan Kontrol FLC dan Pembuatan Data Penelitian	18
3.2.4 Simulasi Kontrol Kedalaman.....	20
3.2.5 Simulasi <i>Path Tracking</i> pada Lambung Kapal	20
3.2.6 Kesimpulan Penelitian	20
BAB IV	21
4.1 Perancangan <i>Fuzzy Logic Controller</i>	21
4.1.1 Fuzzifikasi	21
4.1.2 <i>Rule Base</i>	24
4.1.3 <i>Fuzzy Inference</i>	27
4.1.4 Defuzzifikasi.....	29
4.2 Pengujian <i>Open Loop</i> dan <i>Close Loop</i> untuk <i>Heave</i>	29
4.3 Pengujian <i>Open Loop</i> dan <i>Close Loop</i> untuk <i>Surge</i>	30
4.4 Pengujian <i>Open Loop</i> dan <i>Close Loop</i> untuk <i>Yaw</i>	32
4.4.1 Hasil Simulasi dengan <i>Starting Point</i> $x = 0$ dan $y = 0$ dengan $e_{yaw} = -1.15^\circ$	33
4.4.2 Hasil Simulasi dengan <i>Starting Point</i> $x = 0$ dan $y = 0$ dengan $e_{yaw} = -1.2^\circ$	35
4.4.3 Hasil Simulasi dengan <i>Starting Point</i> $x = 0$ dan $y = 0$ dengan $e_{yaw} = -1.3^\circ$	37
4.4.4 Hasil Simulasi dengan <i>Starting Point</i> $x = 8.484$ dan $y = -8.484$ dengan $e_{yaw} = 1.9^\circ$	38
4.4.5 Hasil Simulasi dengan <i>Starting Point</i> $x = 8.484$ dan $y = -8.484$ dengan $e_{yaw} = 1.95^\circ$	40
4.4.6 Hasil Simulasi dengan <i>Starting Point</i> $x = 8.484$ dan $y = -8.484$ dengan $e_{yaw} = 2^\circ$	42
4.5 Perbandingan Hasil Simulasi dengan <i>Starting Point</i> $x = 0$ dan $y = 0$	43
4.6 erbandingan Hasil Simulasi dengan <i>Starting Point</i> $x = 8.484$ dan $y = -8.484$	45
BAB V.....	47
5.1 Kesimpulan	47
5.2 Saran	47
DAFTAR PUSTAKA	49
LAMPIRAN.....	51
BIODATA PENULIS	61

DAFTAR GAMBAR

Gambar 2.1 Diagram dari Kontrol Sistem	3
Gambar 2.2 Hasil <i>Path Tracking</i> Jalur Bidang XY	3
Gambar 2.3 <i>Open-loop control system block diagram</i> dari BlueROV2	4
Gambar 2.4 <i>Feedback control system block diagram</i>	4
Gambar 2.5 Koordinat BlueROV2, struktur rangka dan konfigurasi pendorong	6
Gambar 2.6 BlueROV2 <i>frame of reference</i>	6
Gambar 2.7 <i>Motion Control Systems</i>	8
Gambar 2.8 Model Hidrodinamis untuk BlueROV 2	10
Gambar 2.9 Alur <i>Fuzzy Logic Controller</i>	10
Gambar 2.10 <i>Input dan Output Membership Functions</i> untuk tipe Sugeno	10
Gambar 2.11 Diagram Skematis dari <i>Fuzzy Inference System</i>	11
Gambar 2.12 Konfigurasi Pendorong pada BlueROV2 dengan Model 6 Pendorong.....	13
Gambar 2.13 Kurva Karakteristik <i>thrust</i> vs PWM dari Pendorong T200.....	14
Gambar 2.14 Visualisasi dari Panduan <i>Line-of-Sight</i>	15
Gambar 3.1 <i>Flowchart</i> Penelitian	18
Gambar 3.2 Model Lambung Kapal 3D dan 2D tampak atas	19
Gambar 3.3 <i>Flowchart</i> Tahapan Pemodelan Kontrol FLC.....	20
Gambar 4.1 Perancangan FIS untuk BlueROV 2	21
Gambar 4.2 Fungsi keanggotaan error <i>heave</i> , <i>surge</i> , dan <i>yaw</i> logika <i>fuzzy</i>	22
Gambar 4.3 Fungsi keanggotaan <i>rate of change of error heave</i> , <i>surge</i> , dan <i>yaw</i>	23
Gambar 4.4 Fungsi keanggotaan <i>output</i> dari <i>heave</i> , <i>surge</i> , dan <i>yaw</i>	24
Gambar 4.5 <i>Rule editor</i> dari <i>fuzzy logic designer</i>	25
Gambar 4.6 Proses inferensi logika <i>fuzzy</i> untuk kontrol <i>heave</i>	27
Gambar 4.7 Proses inferensi logika <i>fuzzy</i> untuk kontrol <i>surge</i>	27
Gambar 4.8 Proses inferensi logika <i>fuzzy</i> untuk kontrol <i>yaw</i>	28
Gambar 4.9 <i>Surface viewer</i> logika <i>fuzzy</i> untuk <i>heave</i> , <i>surge</i> , dan <i>yaw</i>	28
Gambar 4.10 Grafik respon <i>open loop</i> untuk kedalaman satu meter	29
Gambar 4.11 Grafik respon <i>close loop</i> untuk kedalaman satu meter	29

Gambar 4.12 Grafik respon <i>open loop</i> untuk $x = 1$ m	31
Gambar 4.13 Grafik respon <i>close loop</i> untuk $x = 1$ m.....	31
Gambar 4.14 Visualisasi dari Bentuk Lambung Kapal	33
Gambar 4.15 Lintasan <i>desired</i> dan aktual BlueROV 2 untuk $e_{yaw} = -1.15^\circ$	34
Gambar 4.16 Respon nilai error $yaw = -1.15^\circ$	34
Gambar 4.17 Respon nilai error <i>heave</i> dengan simulasi error $yaw = -1.15^\circ$	35
Gambar 4.18 Lintasan <i>desired</i> dan aktual BlueROV 2 untuk $e_{yaw} = -1.2^\circ$	35
Gambar 4.19 Respon nilai error $yaw = -1.2^\circ$	36
Gambar 4.20 Respon nilai error <i>heave</i> dengan simulasi error $yaw = -1.2^\circ$	36
Gambar 4.21 Lintasan <i>desired</i> dan aktual BlueROV 2 untuk $e_{yaw} = -1.3^\circ$	37
Gambar 4.22 Respon nilai error $yaw = -1.3^\circ$	37
Gambar 4.23 Respon nilai error <i>heave</i> dengan simulasi error $yaw = -1.3^\circ$	38
Gambar 4.24 Lintasan <i>desired</i> dan aktual BlueROV 2 untuk $e_{yaw} = 1.9^\circ$	39
Gambar 4.25 Respon nilai error $yaw = 1.9^\circ$	39
Gambar 4.26 Respon nilai error <i>heave</i> dengan simulasi error $yaw = 1.9^\circ$	40
Gambar 4.27 Lintasan <i>desired</i> dan aktual BlueROV 2 untuk $e_{yaw} = 1.95^\circ$	40
Gambar 4.28 Respon nilai error $yaw = 1.95^\circ$	41
Gambar 4.29 Respon nilai error <i>heave</i> dengan simulasi error $yaw = 1.95^\circ$	41
Gambar 4.30 Lintasan <i>desired</i> dan aktual BlueROV 2 untuk $e_{yaw} = 2^\circ$	42
Gambar 4.31 Respon nilai error $yaw = 2^\circ$	42
Gambar 4.32 Respon nilai error <i>heave</i> dengan simulasi error $yaw = 2^\circ$	43
Gambar 4.33 Grafik Perbandingan Hasil Simulasi I	43
Gambar 4.34 Grafik Perbandingan Nilai RSE Simulasi I.....	44
Gambar 4.35 Grafik Perbandingan Nilai <i>Error Heave</i> Simulasi I.....	44
Gambar 4.36 Grafik Perbandingan Hasil Simulasi II	45
Gambar 4.37 Grafik Perbandingan Nilai RSE Simulasi II	45
Gambar 4.38 Grafik Perbandingan Nilai <i>Error Heave</i> Simulasi II	46

DAFTAR TABEL

Tabel 2.1 Perbandingan Antara Kontroller Konvensional dengan Kontroller Modern	4
Tabel 2.2 Notasi SNAME untuk <i>Marine Vessels</i>	6
Tabel 2.3 Parameter dari BlueROV2	7
Tabel 2.4 Logika Operator FLC	11
Tabel 2.5 Contoh Penarikan Kesimpulan FLC pada Ekosistem Berkendara	11
Tabel 2.6 Pendorong BlueROV2 dan Posisi Kamera dengan Referensi <i>Body Frame</i>	13
Tabel 4.1 Basis aturan untuk merancang <i>Fuzzy Logic Controller</i>	25
Tabel 4.2 Data posisi sumbu z <i>setpoint</i> dan aktual dalam pengujian <i>open loop</i>	30
Tabel 4.3 Data posisi sumbu z <i>setpoint</i> dan aktual dalam pengujian <i>close loop</i>	30
Tabel 4.4 Data posisi sumbu x <i>setpoint</i> dan aktual dalam pengujian <i>open loop</i>	31
Tabel 4.5 Data posisi sumbu x <i>setpoint</i> dan aktual dalam pengujian <i>close loop</i>	32
Tabel 4.6 Data Koordinat dari Bentuk Lambung Kapal	32

Halaman ini sengaja dikosongkan

BAB I PENDAHULUAN

1.1 Latar Belakang

Indonesia dikenal sebagai negara dengan potensi sumber daya alam yang sangat banyak, khususnya di bidang kemaritiman. Menurut data dari kementerian kelautan dan perikanan tahun 2015, Indonesia merupakan negara kepulauan terbesar di dunia terdiri lebih dari 17.000 pulau dan luas perairannya terdiri dari laut teritorial, perairan kepulauan dan perairan pedalaman yang memiliki luas total sebesar 6.400.000 km² atau mencakup lebih dari 60% dari luas wilayah NKRI. Fakta tersebut menjadikan Indonesia memiliki potensi yang sangat besar di bidang kelautan sehingga penggunaan kapal sudah menjadi hal pokok dalam bidang kelautan. Kapal sejatinya sama seperti mesin-mesin pada umumnya yaitu membutuhkan *maintenance* atau perawatan secara berkala, salah satunya adalah inspeksi pada lambung kapal. Selama ini inspeksi lambung kapal bawah air masih menggunakan penyelaman secara manual atau dilakukan oleh seorang penyelam. Namun hal ini memiliki beberapa kendala seperti pada cuaca, kedalaman, waktu dan jarak pandang. Waktu dari penyelaman di pengaruhi oleh kedalaman, semakin besar kapal yang akan di inspeksi maka waktu penyelaman yang dibutuhkan pun juga semakin lama (Tri, 2019). Seiring dengan perkembangan teknologi, banyak muncul alat-alat yang mendukung dan mempermudah kegiatan manusia, salah satunya adalah *Remotely Operated Vehicles (ROV)*. Pengembangan pada ROV diperlukan dari segi desain, efisiensi, dan juga kualitas. ROV mulai banyak digunakan untuk melakukan tugas yang berhubungan erat dengan lingkungan perairan. Mulai dari tugas untuk pengambilan sampel spesimen biologi laut, pemeriksaan infrastruktur bawah laut, inspeksi lambung kapal, dan lain sebagainya.

Lingkungan perairan terutama lingkungan air laut sangatlah kompleks, dengan faktor-faktor yang dapat diprediksi seperti keadaan topografi laut dan juga dengan banyak faktor yang tidak terduga seperti arus ombak, makhluk laut dan lain sebagainya (Prabhakar and Buckham, 2005). Faktor-faktor tersebut dapat mempengaruhi performa dari ROV terutama untuk tugas yang memerlukan akurasi. Maka dari itu, penelitian tentang sistem kontrol untuk ROV menjadi penting agar ROV dapat menjalankan tugasnya secara sempurna. Dalam mempertahankan posisinya (ROV) di kedalaman air, sistem kendali diterapkan pada ROV diharapkan dapat membantu ROV untuk menstabilkan posisi pada saat melakukan inspeksi pada lambung kapal. Pada saat ini, metode yang sering dijumpai untuk kontrol navigasi dari ROV antara lain PID, *adaptive control*, *fuzzy logic control*, *neural network* dan *LQR control*. Sebagai salah satu metode pengendalian yang sering digunakan, *fuzzy logic control* memiliki keunggulan utama yaitu tidak bergantung pada model matematis yang akurat ataupun model dinamis yang formal, namun tetap dapat secara efektif mengontrol sistem secara nonlinier. Ini sangat mengurangi waktu *start-up* dalam mengembangkan controller dan kompleksitasnya secara keseluruhan. Keterbatasan dari *fuzzy logic controller* adalah sulit untuk memastikan bahwa sistem kontrol stabil dan memiliki sensitivitas yang baik (Debitetto, 1994).

Pada tugas akhir ini akan berfokus pada pembuatan controller untuk ROV agar bisa melakukan *path tracking* sesuai dengan trayektori yang diinginkan yaitu bentuk lambung kapal. Pada tugas akhir ini digunakan sistem kendali *fuzzy logic controller* sebagai kontroler dari ROV untuk mempertahankan posisinya dan menjalankan *path tracking* dari bentuk lambung kapal. Pengujian dilakukan dengan mengevaluasi *close-loop* dari sistem di bawah kontrol dari *fuzzy logic controller* yang nantinya akan memberikan *output* berupa posisi dari ROV.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat pada Tugas Akhir ini adalah sebagai berikut:

1. Bagaimana perancangan pengendalian *fuzzy logic* untuk BlueROV2?
2. Bagaimana performa BlueROV2 yang dikendalikan dengan *fuzzy logic* dalam mengikuti *tracking* lambung kapal?

1.3 Tujuan Penelitian

Tujuan dari dilaksanakannya Tugas Akhir ini adalah sebagai berikut:

1. Mengetahui perancangan pengendalian *fuzzy logic* untuk BlueROV2.
2. Mengetahui performa BlueROV2 yang dikendalikan dengan *fuzzy logic* dalam mengikuti *tracking* lambung kapal.

1.4 Batasan Masalah

Batasan masalah dalam Tugas Akhir ini adalah sebagai berikut:

1. Remotely Operated Vehicle yang digunakan adalah BlueROV2 sebagai model bergerak dengan menggunakan 6 motor: 4 motor untuk bergerak ke depan dan belakang, dan 2 motor untuk bergerak ke atas dan ke bawah.
2. Pengujian dilakukan dengan kedalaman maksimum 12 meter.
3. Remotely Operated Vehicle diuji dengan menggunakan parameter air laut sebagai medium.

1.5 Manfaat Penelitian

Terdapat beberapa manfaat dari Tugas Akhir ini yang diharapkan berguna bagi penulis maupun bagi pembaca/pihak terkait.

1.5.1 Bagi Penulis

Manfaat dari Tugas Akhir ini bagi penulis adalah sebagai berikut:

1. Menerapkan ilmu pengetahuan yang dimiliki dalam rangka mendesain sistem kontrol ROV dengan menggunakan *fuzzy logic*.
2. Menambah wawasan terkait *fuzzy logic*.
3. Mengetahui pentingnya pengembangan dan optimasi produk pada suatu industri.

1.5.2 Bagi Pembaca dan Pihak Terkait

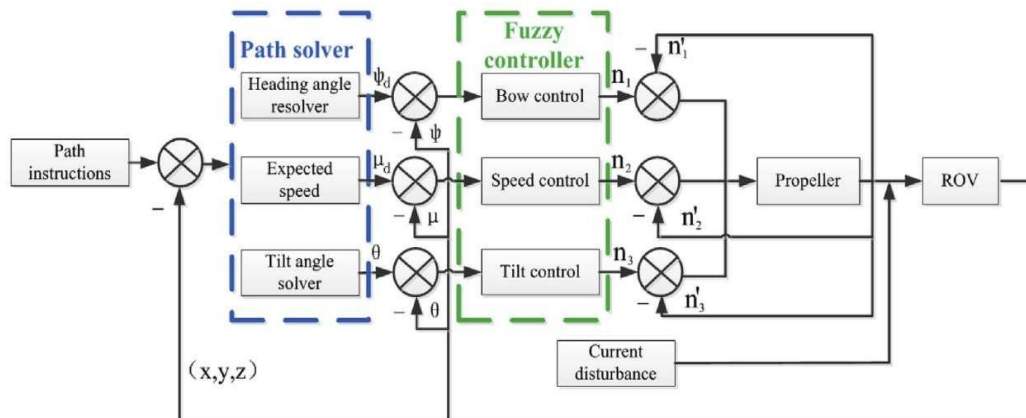
Manfaat dari Tugas Akhir ini bagi pembaca dan pihak terkait adalah sebagai berikut:

1. Sebagai penambah wawasan terkait ROV, khususnya pada perancangan sistem kontrol dengan menggunakan *fuzzy logic*.
2. Mengetahui bagaimana performa ROV dengan menggunakan *fuzzy logic* untuk melakukan inspeksi pada lambung kapal.
3. Menggerakkan perusahaan nasional, mahasiswa, dan seluruh akademisi agar dapat terus memunculkan inovasi baru pada produk-produk yang dapat membawa manfaat bagi industri nasional

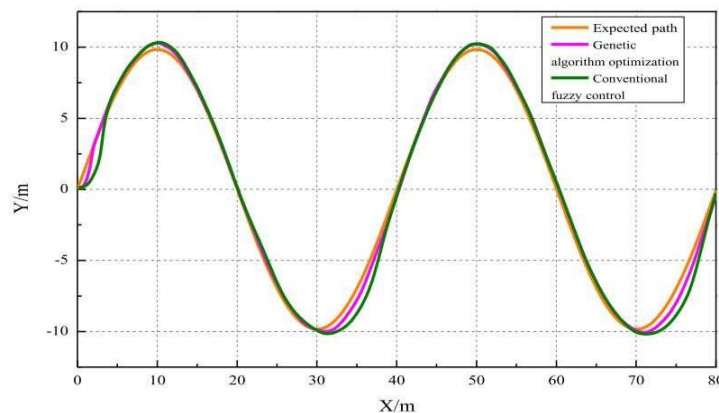
BAB II KAJIAN PUSTAKA

2.1 Penelitian Terdahulu

Pada tahun 2018, Chen dkk melakukan penelitian tentang fuzzy logic control untuk *path tracking* kendaraan bawah air dengan menggunakan *genetic algorithm* untuk optimisasinya. Kinerja *fuzzy logic control* tergantung pada apakah fungsi bawahan memiliki kinerja tinggi dan aturan fuzzy yang digunakan masuk akal. Kendaraan bawah air yang digunakan dengan nama “Sea Dog” akan di kontrol agar bisa bergerak sesuai dengan jalur yang diinginkan.



Gambar 2.1 Diagram dari Kontrol Sistem (Chen *et al.*, 2018)



Gambar 2.2 Hasil Tracking Path Jalur Bidang XY (Chen *et al.*, 2018)

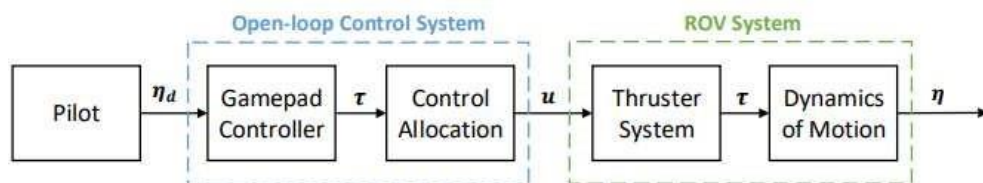
Dari hasil *tracking path* yang dilakukan, terlihat bahwa *fuzzy logic controller* dapat digunakan untuk mengontrol kendaraan bawah air untuk melakukan *tracking path*. Dengan bantuan dari *genetic algorithm* hasil yang didapatkan mampu mendekati *expected path*.

Pada tahun 2016, Aras dkk melakukan penelitian tentang *fuzzy logic controller* untuk mengontrol kedalaman pada ROV. Kontroler seperti PD memiliki masalah untuk mengontrol kedalaman ROV. PD kontroler tidak cocok untuk operasi nonlinier dari kontrol kedalaman ROV. Kontroler PID konvensional juga memiliki masalah dimana sulit untuk mencapai *zero overshoot* dalam hal respon sistem dari kontrol kedalaman. Oleh karena itu FLC atau *fuzzy logic controller* digunakan untuk meningkatkan performa dari sistem. Pendekatan yang digunakan dalam penelitian ini adalah dengan mengobservasi pergeseran *membership function* yang terjadi dalam *input membership function* lalu hasil tersebut akan dianalisis untuk melihat respon sistem dari kontrol kedalaman.

Tabel 2.1 Perbandingan Antara Kontroller Konvensional dengan Kontroller Modern (Aras *et al.*, 2016)

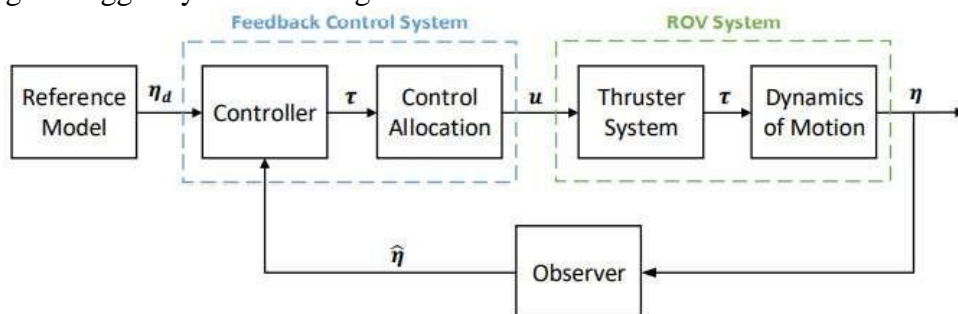
Controller	Advantages	Disadvantages
Single Input Fuzzy Logic Controller (SIFLC)	<ul style="list-style-type: none"> - Slope of linear equation give optimum performances - Able to reduce overshoot 	<ul style="list-style-type: none"> - Different control surface of piecewise linear region effect performance of depth control
PD Controller	<ul style="list-style-type: none"> - Suitable for larger gain by adding damping to the system 	<ul style="list-style-type: none"> - Not suitable for non-linear operation
PID Controller with disturbance occurrence	<ul style="list-style-type: none"> - Easy to implement 	<ul style="list-style-type: none"> - Changing parameters effect times of regulation prolongation
Fuzzy Logic Controller	<ul style="list-style-type: none"> - Suitable for non-linear operation and able to apply heuristic rules that reflect experiences of the human experts 	<ul style="list-style-type: none"> - Fine tuning process is highly time consuming

Pada tahun 2018, Wu melakukan penelitian tentang membuat model dan kontrol untuk 6 DoF ROV. BlueROV2 digunakan dalam penelitian yang dilakukan oleh Wu. Diketahui bahwa BlueROV2 dapat dikendalikan dengan operator melalui *gamepad controller* yang mengirim sinyal ke ROV. Sistem kontrol dengan tipe seperti ini memiliki rancangan yang sederhana dan memiliki akurasi yang rendah karena kurangnya mekanisme *feedback*. Oleh karena itu, sistem kontrol tipe ini hanya cocok dioperasikan secara manual.



Gambar 2.3 Open-loop control system block diagram dari BlueROV 2 (Wu, 2018)

Untuk mengoperasikan BlueROV2 secara otomatis tanpa menggunakan bantuan operator manusia sama sekali dibutuhkan mekanisme *feedback* untuk memberikan hasil yang handal. Dengan *input* model referensi yang memberikan keadaan yang diinginkan untuk ROV, dan *feedback controller* yang menggunakan perbedaan antara nilai *output* yang diinginkan dan nilai *output* yang sesungguhnya untuk menghasilkan kekuatan kontrol.



Gambar 2.4 Feedback control system block diagram dari BlueROV 2 (Wu, 2018)

Dalam penelitian yang Wu lakukan, kontroller di desain untuk menghasilkan kontrol terhadap gaya yang dibutuhkan untuk ROV sedangkan alokasi kontrol mendistribusikan kontrol gaya ke pendorong dari ROV. PID kontroller digunakan karena kesederhanaan dan fleksibilitas dari algoritma PID itu sendiri.

2.2 Dasar Teori

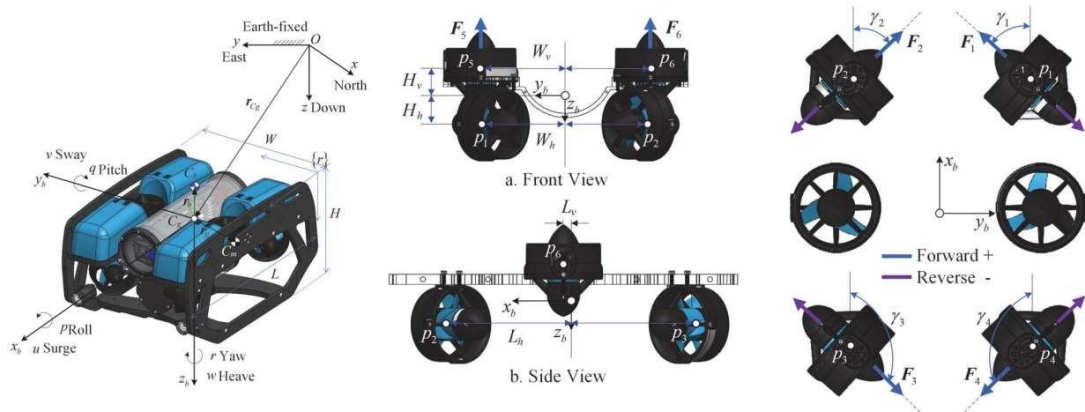
2.2.1 *Remotely Operated Vehicle (ROV)*

Pada umumnya, *underwater vehicle* dapat dibagi menjadi dua tipe: *Autonomous Underwater Vehicle (AUV)* dan *Remotely Operated Vehicle (ROV)*. AUV merupakan robot yang bergerak di dalam air tanpa membutuhkan input dari operator. Sedangkan ROV merupakan kendaraan bawah air yang pergerakannya dibatasi, biasa digunakan di industri perairan dalam seperti ekstraksi minyak lepas pantai (Rahimuddin, 2019). ROV diklasifikasikan dari ukuran, berat dan dayanya maka dapat dibagi sebagai berikut (Firmansyah, 2017) :

1. Micro ROV, tipe ROV ini memiliki berat dan ukuran yang sangat kecil. Micro ROV ada yang memiliki berat kurang dari 3kg. ROV tipe ini biasanya digunakan untuk membantu penyelam, secara spesifik untuk mengakses titik yang tidak dapat dijangkau oleh para penyelam seperti gua kecil bawah air dan pipa.
2. Mini ROV, tipe ROV ini diperkirakan memiliki berat hingga 15kg. Mini ROV biasanya memiliki daya kurang dari 5 HP dan pada umumnya dilengkapi dengan sonar untuk penggunaan survei bawah air. ROV jenis dapat mencapai kedalaman di bawah 1000 meter.
3. Light workclass, tipe ROV ini memiliki daya yang diperkirakan kurang dari 50 HP. Biasanya ROV tipe ini dibuat untuk memiliki tiga fungsi utama yang diinginkan. Tipe ini mampu bekerja sampai 2000 meter di bawah air.
4. Heavy workclass, tipe ROV ini memiliki daya sekitar 220 HP dan memiliki dua fungsi utama. ROV tipe ini mampu bekerja hingga kedalaman 3500 meter dibawah air.
5. Trenching/burial, tipe ROV ini memiliki daya diatas 200 HP dan mampu bekerja hingga kedalaman 6000 meter dibawah air.

2.2.1.1 Notasi

BlueROV2 menggunakan enam pendorong untuk dapat bergerak di dalam air. Empat diantaranya merupakan pendorong dengan orientasi *vectored* yang dapat memberikan kebebasan ROV untuk bergerak pada arah *sway*, *surge*, dan *yaw*. Sementara dua pendorong lainnya berfungsi untuk memberikan gerak mengangkat keatas. Dengan konfigurasi enam pendorong tersebut juga memberikan ROV kemampuan untuk bergerak kearah roll. Terdapat satu degree of freedom yang tidak dapat dikontrol pada BlueROV2 dengan enam pendorong yaitu gerak ke arah pitch. Dengan program bawaan yang ada ROV hanya dapat bergerak 4 DoF. Namun dengan adanya perubahan *code* pada program, pergerakan ke arah roll dapat ditambahkan. DoF dari BlueROV2 dan konfigurasi pendorong dapat dilihat pada gambar 2.3 dan notasi untuk *degrees of freedom* oleh *Society of Naval Architects and Marine Engineers (SNAME)* pada tabel 2.2



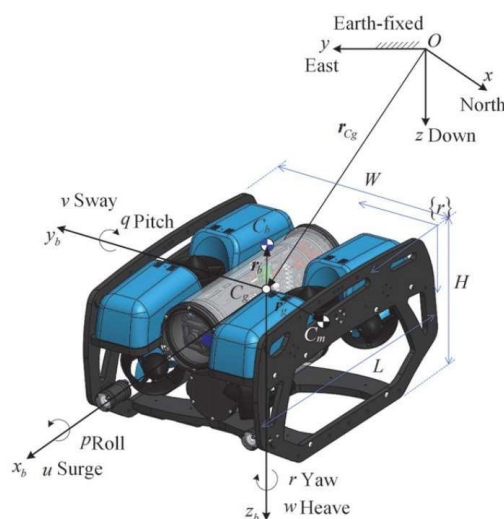
Gambar 2.5 Koordinat BlueROV2, struktur rangka dan konfigurasi pendorong (Einarsson *et al.*, 2020)

Tabel 2.2 Notasi SNAME untuk marine vessels (Einarsson *et al.*, 2020)

DoF	Movement Notation	Forces and moments	Linear and angular velocities	Positions and Euler angles
1	Motions in x direction (surge)	X	u	x
2	Motions in y direction (sway)	Y	v	y
3	Motions in z direction (heave)	Z	w	z
4	Rotation about the x axis (roll)	K	p	ϕ
5	Rotation about the y axis (pitch)	M	q	θ
6	Rotation about the z axis (yaw)	N	r	ψ

2.2.1.2 Frame of Reference

Untuk menganalisis pergerakan dari kendaraan air dalam 6 DoF, dibutuhkan minimal dua *frame of reference* (Fossen, 2011). Seperti yang ditulis oleh Fossen, dua *frame of reference* dibutuhkan untuk membangun model matematis yang dapat mewakili BlueROV2, maka pada penelitian ini akan menggunakan *frame* yang sesuai dengan referensi bacaan yaitu menggunakan *body frame* dan ditunjukkan dalam *earth fixed North-East-Down (NED) frame*.



Gambar 2.6 BlueROV2 *frame of reference* (E. M. Einarsson *et al.*, 2020)

2.2.1.3 Parameter BlueROV2

Untuk memodelkan hidrodinamik dari BlueROV2 diperlukan banyak sekali parameter yang tidak diketahui, sehingga tidak layak untuk dilakukan estimasi. Namun dengan membuat beberapa asumsi dan pertimbangan yang dapat dibuat dengan meninjau dari geometri dan operasi dari BlueROV2, menghasilkan pengurangan yang cukup besar dari parameter yang tidak diketahui (J. González-García *et al*, 2021). Kondisi dan asumsi tercantum sebagai berikut:

- Karena BlueROV2 bekerja dengan kecepatan yang relatif rendah (kurang dari 2 m/s), maka gaya *lift* dapat diabaikan
- BlueROV2 diasumsikan memiliki *port-starboard* dan *fore-aft* yang simetris, dan *center of gravity* (CG) berada dalam bidang yang simetris.
- BlueROV2 diasumsikan bekerja pada zona yang tidak terpengaruh oleh ombak. Sebagai hasilnya, gangguan ombak pada BlueROV2 dapat diabaikan.
- Alokasi pendorong pada BlueROV2 tidak mengizinkan untuk mengontrol gerakan pada orientasi *pitch*. Namun, gerakan di sekitar sumbu ini dianggap dapat mengatur diri sendiri karena momen pemulihan dari gaya apung ROV, sebagai hasilnya terjadi pengurangan pada sistem.

Posisi dari BlueROV2 disimbolkan dengan $\boldsymbol{\eta}$ dengan mengacu pada *earth-fixed frame* dan kecepatan dari BlueROV2 dilambangkan dengan \boldsymbol{v} dengan mengacu pada *body-fixed frame*. Untuk gaya dan momen dari BlueROV2 disimbolkan dengan $\boldsymbol{\tau}$ dengan mengacu pada *body-fixed frame*.

$$\boldsymbol{\eta} = [x \ y \ z \ \varphi \ \theta \ \psi]^T \quad 2.1$$

$$\boldsymbol{v} = [u \ v \ w \ p \ 0 \ r]^T \quad 2.2$$

$$\boldsymbol{\tau} = [X \ Y \ Z \ K \ M \ N]^T \quad 2.3$$

Dengan mempertimbangkan pengurangan pada sistem seperti yang telah dibahas sebelumnya, maka parameter untuk BlueROV2 dapat dilihat pada tabel 2.3

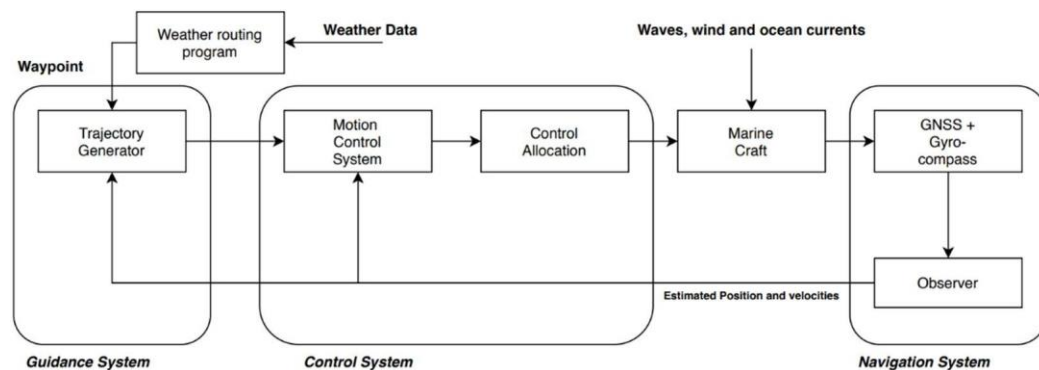
Tabel 2.3 Parameter dari BlueROV2 (J. González-García *et al.*, 2021)

Parameter	Simbol	Nilai	Satuan
Massa	m	10	kg
Buoyancy	B	100.06	N
Berat	W	98.1	N
<i>Center of gravity</i>	$r_G = (x_g, y_g, z_g)$	$r_G = (0, 0, 0)$	m
<i>Center of buoyancy</i>	$r_B = (x_b, y_b, z_b)$	$r_B = (0, 0, 0.02)$	m
Momen inersia	$I = \text{diagonal} (I_x, I_y, I_z)$	$I = \text{diagonal} (0.16, 0.16, 0.16)$	kg m ²
Parameter massa tambahan	$X_{\dot{u}}$	-5.5	kg
	$Y_{\dot{v}}$	-12.7	kg
	$Z_{\dot{w}}$	-14.57	kg
	$K_{\dot{p}}$	-0.12	kg m ² /rad
	$M_{\dot{q}}$	-0.12	kg m ² /rad
	$N_{\dot{r}}$	-0.12	kg m ² /rad

Parameter Linear Damping	X_u	-4.03	N s/m
	Y_v	-6.22	N s/m
	Z_w	-5.18	N s/m
	K_p	-0.07	N s/rad
	M_q	-0.07	N s/rad
	N_r	-0.07	N s/rad
Parameter Quadratic Damping	$X_{u u }$	-18.18	N s ² /m ²
	$Y_{v v }$	-21.66	N s ² /m ²
	$Z_{w w }$	-36.99	N s ² /m ²
	$K_{p p }$	-1.55	N s ² /rad ²
	$M_{q q }$	-1.55	N s ² /rad ²
	$N_{r r }$	-1.55	N s ² /rad ²

2.2.2 ROV Motions

Motion control system untuk kapal dan kendaraan air lainnya terdiri dari tiga blok yang independen: panduan, navigasi dan kontrol (Einarsson *et al.*, 2020).



Gambar 2.7 Motion Control Systems (Einarsson *et al.*, 2020)

Singkatnya, panduan (*guidance*) akan mengambil data cuaca, data navigasi, dan masukan dari operator untuk menghitung posisi, kecepatan dan percepatan yang diinginkan secara kontinu dari kendaraan air. Navigasi umumnya akan terlibat dengan penentuan posisi, sikap, perjalanan dan jarak tempuh. Terakhir, kontrol akan menentukan kebutuhan untuk gaya dan momen yang diperlukan untuk mencapai objektif (Fossen, 2011).

Perbedaan yang paling penting untuk *motion control* adalah apakah kendaraan air *underactuated* atau *fully actuated*. Kendaraan air yang tergolong *underactuated* mempunyai batas yang dicapai untuk tujuan pengendalian. Sayangnya, kebanyakan dari kendaraan air merupakan tipe *underactuated*, termasuk BlueROV2 yang akan digunakan dalam simulasi penelitian kali ini yang berarti mereka tidak bisa menghasilkan kontrol gaya dan gerakan di semua DoF (Fossen, 2011).

DoF dapat diartikan sebagai bagian dari perpindahan dan rotasi yang independen yang sepenuhnya dapat menentukan posisi yang dipindahkan dan orientasi dari kendaraan air. Kendaraan air yang dapat bergerak bebas dalam ruang 3-D memiliki maksimum DoF sebanyak enam, tiga elemen gerak translasi dan tiga elemen gerak rotasi (Fossen, 2011).

Untuk mendesain sebuah kontrol sistem, penting untuk menentukan ruang kerja dimana kontrol objektif ditentukan. Semua kemungkinan posisi dan orientasi dari kendaraan air yang dapat dicapai didefinisikan sebagai ruang konfigurasi (Fossen, 2011).

Kendaraan air beroperasi dalam 6 DoF, dan perpindahan dan rotasi dideskripsikan dengan enam posisi umum dan kecepatan, dimana sudut Euler didefinisikan pada interval $S = [0, 2\pi]$ (Fossen, 2011).

2.2.2.1 ROV Equations of Motions

Model dari proses mesin dapat dilihat pada persamaan 2.4 dan 2.5 yang menjelaskan persamaan gerak Newton-Euler tentang *Center of Origin* (CO). Model ini didapatkan dari referensi bacaan pada Fossen (Fossen, 2011).

$$\dot{\eta} = J(\eta)\mathbf{v} \quad 2.4$$

$$\underbrace{\mathbf{M}_{RB}\dot{\mathbf{v}} + \mathbf{C}_{RB}(\mathbf{v})\mathbf{v}}_{\text{rigid-body forces}} + \underbrace{\mathbf{M}_A\dot{\mathbf{v}} + \mathbf{C}_A(\mathbf{v}_r)\mathbf{v}_r + \mathbf{D}(\mathbf{v}_r)\mathbf{v}_r + g(\eta)}_{\text{hydrodynamical forces}} = \boldsymbol{\tau} + \boldsymbol{\tau}_{ext} \quad 2.5$$

Dimana

$\mathbf{M}_{RB} \in \mathbb{R}^{6 \times 6}$ merupakan matriks massa body dalam CO.

$\mathbf{C}_{RB}(\mathbf{v}) \in \mathbb{R}^{6 \times 6}$ merupakan matriks rigid body coriolis dan gaya sentripetal.

$\mathbf{M}_A \in \mathbb{R}^{6 \times 6}$ merupakan massa tambahan dalam CO.

$\mathbf{C}_A(\mathbf{v}_r) \in \mathbb{R}^{6 \times 6}$ merupakan gaya coriolis untuk massa tambahan.

$\mathbf{D}(\mathbf{v}_r) \in \mathbb{R}^{6 \times 6}$ merupakan matriks damping.

$g(\eta) \in \mathbb{R}^{6 \times 1}$ merupakan vektor gaya pemulih hidrostatik.

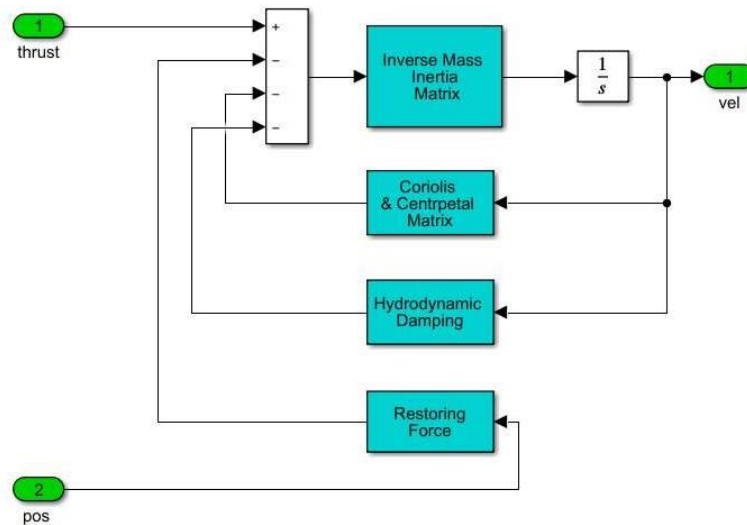
$\boldsymbol{\tau} \in \mathbb{R}^{6 \times 1}$ merupakan vektor gaya propulsi.

$\boldsymbol{\tau}_{ext} \in \mathbb{R}^{6 \times 1}$ merupakan vektor gaya eksternal yang mencakup tether.

Persamaan 2.5 adalah Hukum kedua Newton tentang gerak untuk mengekspresikan dalam koordinat rangka bergerak, dan juga dibutuhkan untuk mengkompensasi gaya coriolis dan sentripetal (Fossen, 2002). $\mathbf{v}_r \in \mathbb{R}^{6 \times 1}$ adalah vektor kecepatan relatif terhadap air dan digabungkan dari $\mathbf{v}_r = \mathbf{v} - \mathbf{v}_c$, dimana $\mathbf{v}_c \in \mathbb{R}^{6 \times 1}$ adalah arus bawah air yang terurai dalam body frame (Fossen, 2011).

2.2.2.2 Model Hidrodinamis BlueROV 2

Model hidrodinamis diperlukan sebagai model bergerak dari BlueROV 2 agar simulasi dapat berjalan mendekati keadaan yang sesungguhnya. Model yang digunakan dalam tugas akhir ini adalah model *state space* 6 DoF dari referensi. Berikut ini adalah gambaran model *state space* yang digunakan.



Gambar 2.8 Model Hidrodinamis untuk BlueROV 2 (Chin, 2013)

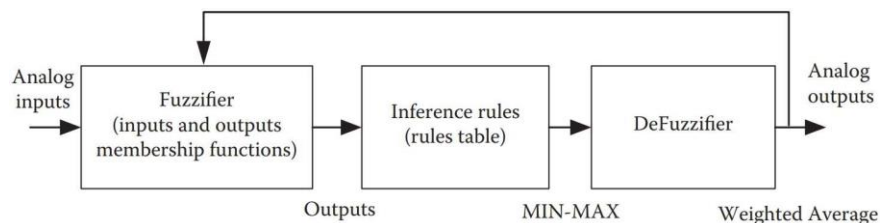
Gambar 2.8 menampilkan model hidrodinamis BlueROV 2 dalam *block_simulink*. Model hidrodinamis untuk BlueROV 2 tersusun dari komponen yang telah dituliskan pada persamaan 2.5.

2.2.3 Fuzzy Logic Controller

Sejauh ini telah banyak penelitian dan pengembangan yang dilakukan untuk kontroler ROV, termasuk linear (konvensional) dan skema kontrol cerdas (modern). Metode kontrol cerdas, dimana menggunakan *neural network* (NN), *sliding mode* (SLC), dan *fuzzy logic controller* (FLC) merupakan pilihan yang lebih kokoh dan mampu beradaptasi dengan kondisi ketidakpastian hidrodinamik. Selain itu juga mereka menunjukkan ketahanan yang sangat baik terhadap gangguan.

Contohnya, *fuzzy logic controller* (FLC) harus berurusan dengan operasi *fuzzification*, *rule-based storage*, *inference mechanism* dan *defuzzification*. Namun terlepas dari itu semua, *fuzzy logic controller* menawarkan struktur yang sederhana namun memiliki tingkat kebebasan yang tinggi dalam penyetelan parameter kontrolnya dibandingkan dengan tipe kontroler nonlinier lainnya (Chin, 2013).

Alur *fuzzy logic controller* di ilustrasikan oleh gambar 2.8



Gambar 2.9 Alur *Fuzzy Logic Controller* (Chin, 2013)

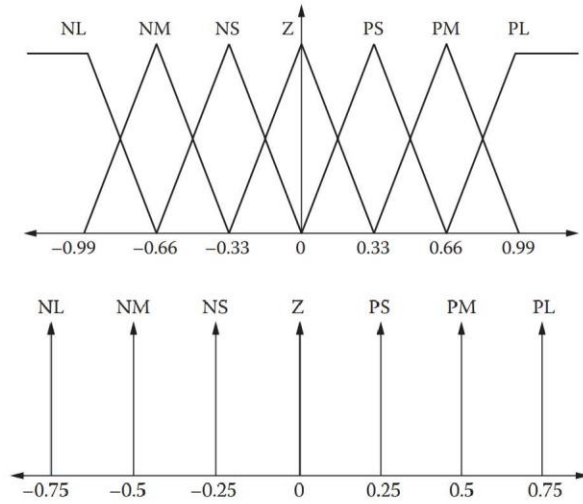
Seperti yang terlihat pada gambar 2.9, untuk mendesain fuzzy logic controller membutuhkan tiga tahapan: *fuzzification*, *inference rules*, dan *defuzzification*.

- *Fuzzification*, berfungsi untuk mengubah *input* dari sensor yang masih berupa nilai real menjadi nilai data *fuzzy*.
- *Inference rules*, berfungsi sebagai penarikan kesimpulan dari input yang telah

diberikan sebelumnya.

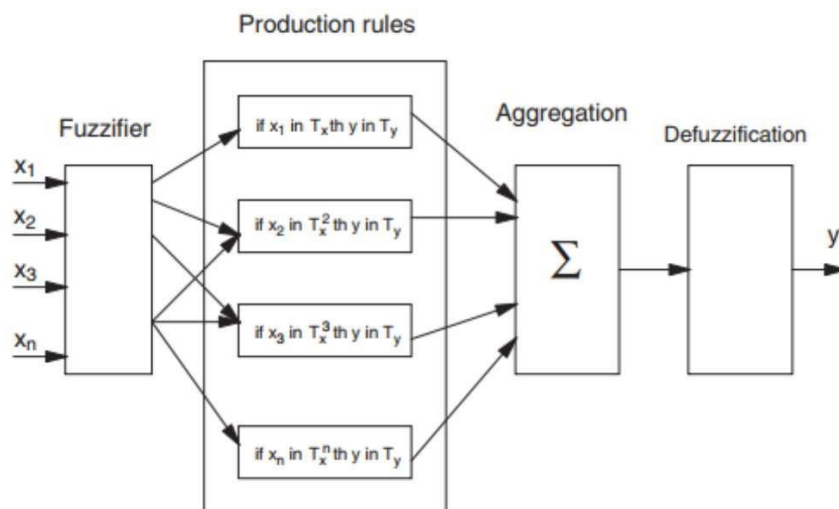
- *Defuzzification*, berfungsi untuk mengubah nilai output yang masih dalam bentuk *fuzzy* menjadi nilai output aktual.

Setelah melakukan tiga tahapan tersebut, pengontrol disetel dengan cara iteratif.



Gambar 2.10 *Input* (atas) dan *Output* (bawah) *Membership Muncions* untuk tipe Sugeno. (Chin, 2013)

Pada gambar 2.10 terlihat tujuh fungsi keanggotaan pada *fuzzy logic controller* yaitu *Negative Large* (NL), *Negative Medium* (NM), *Negative Small* (NS), *Zero* (Z), *Positive Small* (PS), *Positive Medium* (PM), *Positive Large* (PL). *Crisp set* dan *Fuzzy set* merupakan terminologi yang digunakan dalam FLC, *Crisp set* memiliki arti sebagai informasi yang jelas dan mengandung nilai asli, sedangkan *Fuzzy set* diartikan sebagai informasi yang tidak jelas dan belum mengandung nilai asli. Diagram dari *Fuzzy Inference System* dapat dilihat pada gambar 2.8



Gambar 2.11 Diagram Skematis dari *Fuzzy Inference System*.

Logika fuzzy mampu meniru dan mengaplikasikan penalaran dari pola pikir manusia secara prakiraan, hal ini yang membuat FLC unggul dari kontroller lainnya yang membutuhkan masukan yang akurat dan presisi. Karena FLC meniru pola pikir manusia maka diperlukanlah operator untuk membantu proses penarikan keputusan. Operator tersebut adalah operator *complement* (NOT), *intersection* (AND), dan *union* (OR). Logika penalaran

pada FLC berbentuk argumen variabel linguistik yang tersusun dari pernyataan bersyarat untuk klausul jika dan maka. Logika ini digunakan dengan konsep bahwa semakin banyak syarat yang terpenuhi pada premis yang telah disusun, maka semakin wajib tindakan yang akan diambil (Dernoncourt, 2013). Susunan dari logika operasi yang digunakan pada FLC dapat dilihat pada Tabel 2.4 dan contohnya pada Tabel 2.5

Tabel 2.4 Logika Operator FLC

AND			OR			NOT	
A	B	$A \cup B$	A	B	$A \cup B$	A	\bar{A}
1	1	1	1	1	1	1	0
1	0	0	1	0	1	0	1
0	1	0	0	1	1		
0	0	0	0	0	0		

Tabel 2.5 Contoh Penarikan Kesimpulan FLC pada Ekosistem Berkendara (Dernoncourt, 2013)

IF lampu merah	IF kecepatan tinggi	dan IF lampu lalu lintas dekat	THEN injak rem dengan keras
IF lampu merah	IF kecepatan rendah	dan IF lampu lalu lintas jauh	THEN jaga kecepatan
IF lampu orange	IF kecepatan sedang	dan IF lampu lalu lintas jauh	THEN injak rem perlahan
IF lampu hijau	IF kecepatan rendah	dan IF lampu lalu lintas dekat	THEN tambah kecepatan

FLC dapat diakses di MATLAB dengan menggunakan *add-on Fuzzy Logic Toolbox* (FLT), terdapat 5 jenis GUI yang tersedia untuk merancang *Fuzzy Inference System* (FIS).

1. *FIS Editor*, digunakan untuk menambahkan parameter *input* ataupun *output* yang digunakan.
2. *Membership Function Editor*, digunakan untuk merubah membership function yang digunakan seperti tipe, *range*, *display range*, dan nilai *params*.
3. *Rule Editor*, digunakan untuk memberi *rule* yang akan digunakan pada *membership function*. Rule dapat ditentukan dan di *generate* oleh *user*.
4. *Rule Viewer*, digunakan untuk melihat kinerja dari rule dengan menggunakan GUI untuk memberikan nilai *input*.
5. *Surface Viewer*, digunakan untuk melihat rule yang telah di input dalam beberapa format seperti SISO, SIMO, MISO, dan MIMO.

2.2.3.1 Membership Function

Membership function atau bisa disingkat dengan MF merupakan kurva yang menunjukkan bagaimana setiap nilai dari input yang dipetakan ke dalam nilai keanggotaan yang memiliki rentang nilai dari 0 sampai 1. Setiap MF pada *fuzzy set* pada umumnya disimbolkan dengan μ_A . Jika x adalah elemen dari A , maka $\mu_A(x)$ dapat disebut sebagai *Membership Degree* dari x pada tiap himpunan *fuzzy set*.

$$A = \{(x, \mu_A(x)) \mid x \in A\}$$

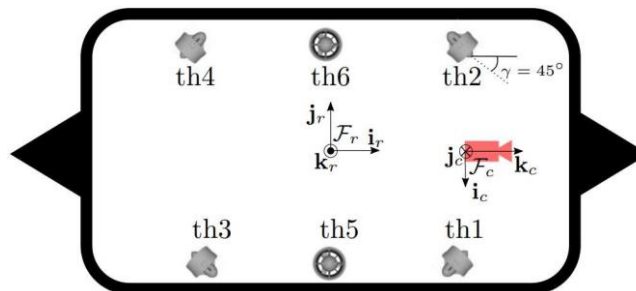
MD $\mu_A(x)$ menguantifikasi nilai keanggotaan dari elemen x kepada himpunan *fuzzy set*. Jika nilai $\mu_A(x) = \text{satu (1)}$ maka elemen x merupakan anggota dalam himpunan *fuzzy set* dan jika nilai $\mu_A(x) = \text{nol (0)}$ maka elemen x bukan merupakan anggota dalam himpunan *fuzzy set*.

Dalam pengaplikasian FLC, terdapat berbagai jenis MF yang dapat digunakan. Ada MF yang berbentuk trapezoidal, segitiga, lonceng, kurva gaussian, kurva polynomial, dan fungsi sigmoid. Semua bentuk MF yang telah disebutkan sebelumnya mengikuti persamaan matematis yang berbeda tergantung pada bentuk yang akan digunakan.

2.2.4 Thruster Model

Selain memodelkan ROV kedalam persamaan matematis, hal yang serupa juga perlu dilakukan pada konfigurasi pendorong ROV untuk menentukan berapa besar gaya yang dibutuhkan agar ROV dapat bergerak sesuai dengan input yang diinginkan.

BlueROV2 yang digunakan dalam penelitian kali ini adalah model yang memiliki konfigurasi pendorong sebanyak enam buah. Model matriks untuk pendorong BlueROV2 sudah pernah diselesaikan oleh Laranjeira, 2019. Dengan menggunakan notasi $F_r(P_r, i_r, j_r, k_r)$ koordinat *frame* terhubung pada robot r . P_r merupakan pusat dari robot, i_r, j_r, k_r , masing-masing adalah sumbu longitudinal, lateral, dan vertikal dari robot.



Gambar 2.12 Konfigurasi Pendorong pada BlueROV2 dengan Model 6 Pendorong (Laranjeira, 2019)

Terlihat pada gambar orientasi sudut $\gamma = 45^\circ$. Seperti yang telah dibahas pada subbab 2.2.1 bahwa dengan menggunakan konfigurasi enam pendorong hanya dapat memungkinkan untuk mengontrol 5 DoF. Dibutuhkan dua pendorong lagi agar ROV mampu bermanuver ke arah *pitch*.

Motor pendorong yang digunakan merupakan model T200 dari BlueRobotic (350 W, 25 A of max. current, 5.1 kgf of max. thrust, 300-3800 rev/min of rotation speed). Tabel dibawah berikut akan menampilkan pendorong, orientasi kamera, dan posisi yang sesuai untuk BlueROV2.

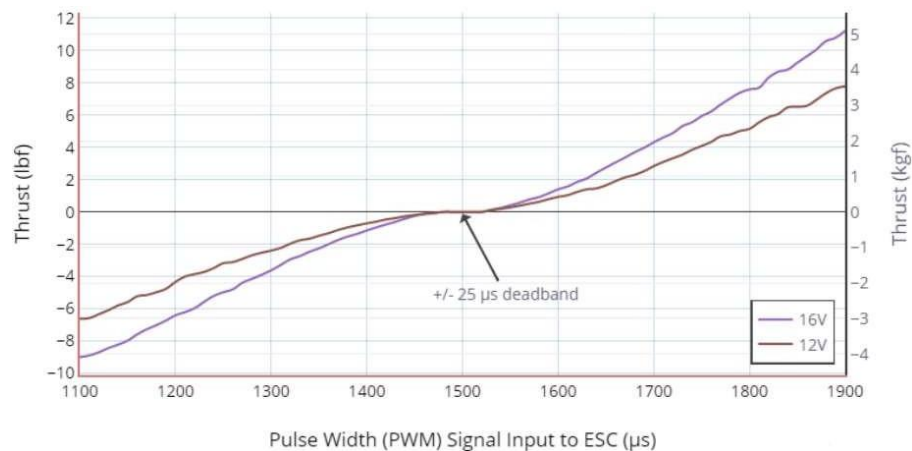
Tabel 2.6 Pendorong BlueROV2 dan posisi kamera dengan referensi body frame. Semua pengukuran menggunakan satuan meter (Laranjeira, 2019).

Positions	X	Y	Z
${}^r\overline{\mathbf{P}}_{th1}$	0.135	-0.115	0.0
${}^r\overline{\mathbf{P}}_{th2}$	0.135	0.115	0.0
${}^r\overline{\mathbf{P}}_{th3}$	-0.135	-0.115	0.0
${}^r\overline{\mathbf{P}}_{th4}$	-0.135	0.115	0.0
${}^r\overline{\mathbf{P}}_{th5}$	0	-0.115	0.070
${}^r\overline{\mathbf{P}}_{th6}$	0	0.115	0.070
${}^r\overline{\mathbf{P}}_c$	0.17	0	0.070

2.2.4.1 Pendorong T200

Pendorong yang digunakan BlueROV2 adalah pendorong T200 yang merupakan tipe pendorong yang dikontrol melalui *Pulse-Width Modulation* (PWM) yang pada dasarnya adalah sinyal gelombang yang berbentuk persegi dimana durasi waktu untuk menyala dapat diatur sesuai dengan keinginan (Ibrahim, 2014).

Sinyal PWM yang dihasilkan dari pendorong T200 BlueROV2 akan dikirim ke sirkuit *electronic speed control* (ESC) yang akan mengatur daya yang akan dikirimkan ke motor pendorong. Sinyal PWM yang dikirim ke sirkuit ESC diatur oleh karakteristik motor pendorong yang digambarkan pada gambar 2.12.



Gambar 2.13 Kurva Karakteristik *thrust* vs PWM dari Pendorong T200 (BlueRobotics, 2019)

Dari gambar 2.13 terlihat bahwa nilai dari PWM sinyal dibatasi dari *range* 1100 μ s sampai 1900 μ s. Pada kurva juga terlihat *deadband* di sekitar kecepatan nol, interpolasi kurva dapat digunakan untuk menentukan perintah PWM dari *input* yang dibutuhkan untuk mendorong (Bessa *et al.*, 2006).

2.2.5 Profil Bentuk Lambung Kapal

Bentuk lambung kapal merupakan *input* utama yang akan dijadikan acuan untuk *path tracking* dari BlueROV2. Untuk mengetahui performa dari kontrol yang akan diteliti, diperlukan profil jalan yang dapat digunakan untuk mewakili dari lambung kapal asli.

Model lambung kapal Wigley dan model S60 merupakan dua model yang populer

dalam eksperimen hidrodinamis (Hassan *et al.*, 2013). Dalam penelitian ini model Wigley akan dipilih untuk menyimulasikan kondisi yang sebenarnya. Secara matematis, lambung kapal model Wigley dapat di *generate* dengan persamaan :

$$f(x, y) = \pm \frac{B}{2} \cdot \left\{ 1 - \left(\frac{2x}{L} \right)^2 \right\} \cdot \left\{ 1 - \left(\frac{z}{T} \right)^2 \right\} \quad 2.6$$

Dengan menggunakan persamaan di atas dengan x, y, z adalah panjang, lebar dan tinggi. Persamaan di atas akan di *generate* dengan menggunakan MATLAB untuk melihat model 3D dari lambung kapal. Dimana B adalah lebar kapal, L adalah panjang kapal, dan T adalah *draft* atau jarak dari permukaan air ke dasar dari lambung kapal.

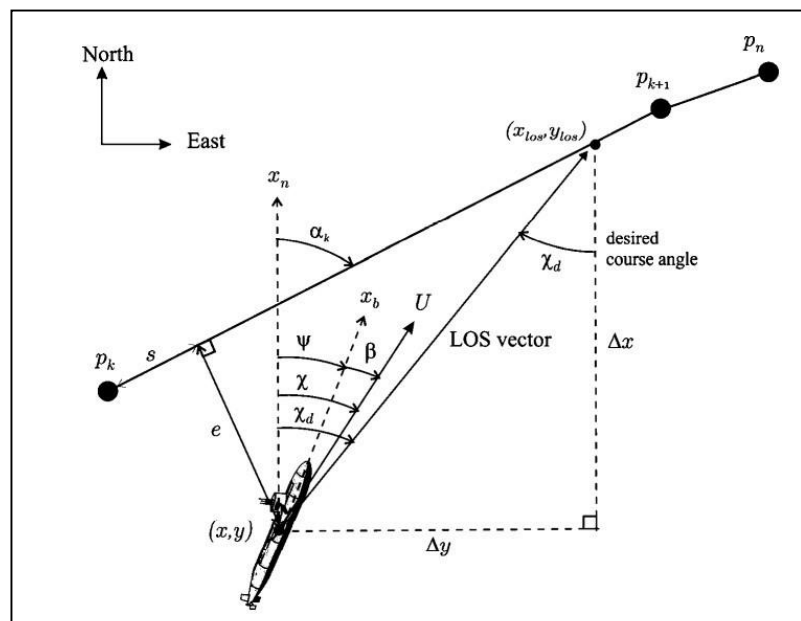
2.2.6 Perencanaan Trayektori Bentuk Lambung Kapal

Untuk memandu BlueROV 2 agar bergerak sesuai dengan lintasan yang diinginkan diperlukan jalur atau lintasan yang khusus. Dalam penelitian ini lintasan tersebut akan berupa bentuk dari lambung kapal. Manuver yang dilakukan oleh BlueROV 2 bergantung pada posisinya yang berada pada koordinat x dan y . Metode pemanduan yang dapat dilakukan agar BlueROV 2 dapat melakukan manuver sesuai dengan trayektori yang telah ditentukan dapat menggunakan metode *line-of-sight* (LOS). *Line-of-sight* akan menggunakan informasi posisi dari BlueROV 2 yang diberikan oleh GPS untuk menghitung derajat dari *heading reference* ψ (yaw) (Fossen, 2011).

Perhitungan dari *line-of-sight* dapat dilakukan dengan persamaan:

$$\psi_{ref} = \text{atan2} \cdot (y_{los} - y(t), x_{los} - x(t)) \quad 2.7$$

Pada persamaan tersebut terdapat y_{los} dan x_{los} yang mendefinisikan tujuan yang diinginkan dari BlueROV 2, sedangkan $y(t)$ dan $x(t)$ mendefinisikan posisi aktual dari BlueROV 2 pada saat (t) yang diberikan oleh GPS. Visualisasi dari *line-of-sight* dapat dilihat pada gambar 2.14 Berikut.



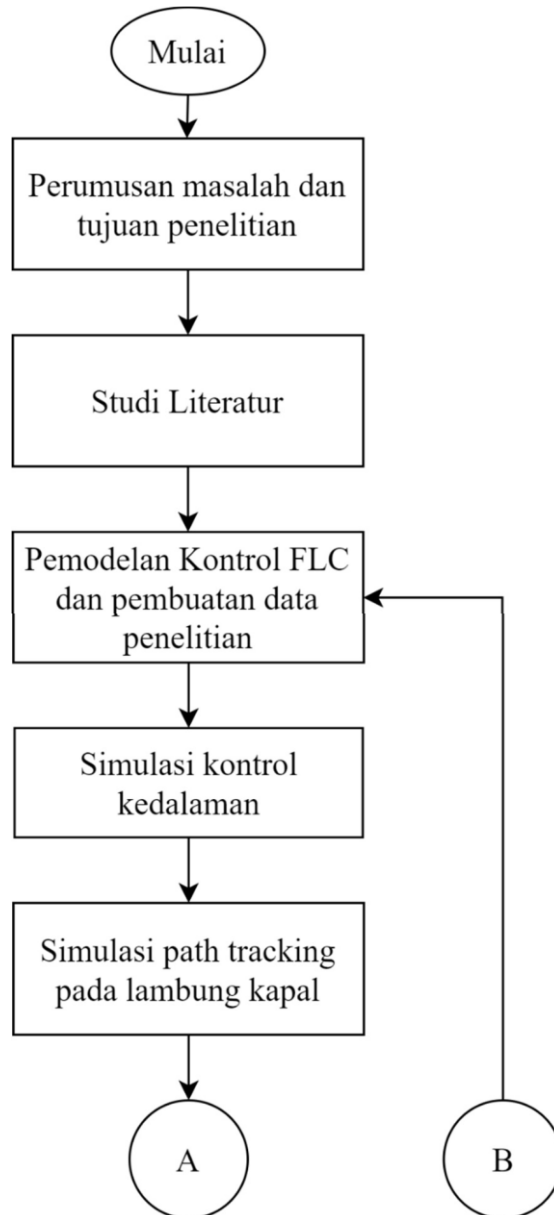
Gambar 2.14 Visualisasi dari Panduan *Line-of-Sight* (Fossen, 2011)

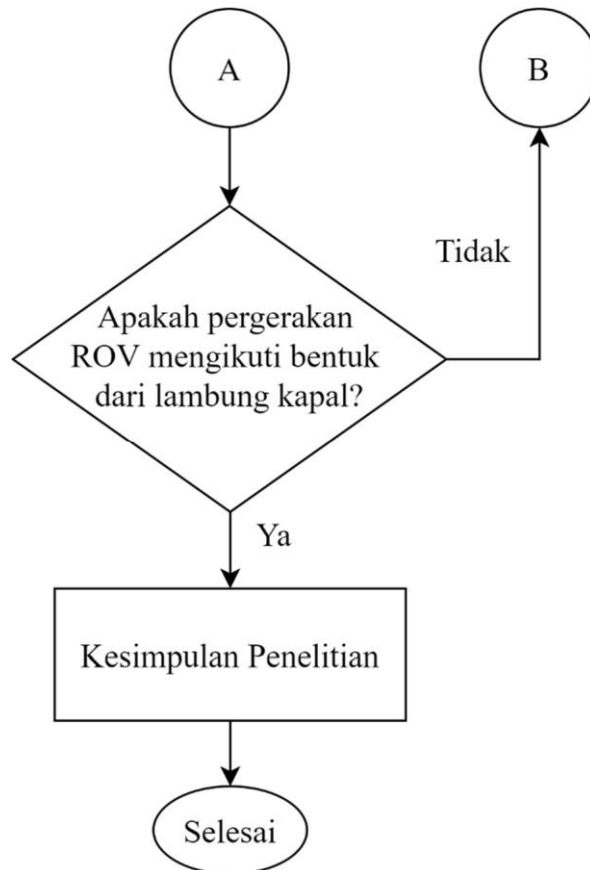
Halaman ini sengaja dikosongkan

BAB III METODOLOGI PENELITIAN

3.1 *Flowchart* Penelitian

Berikut merupakan *flowchart* dari penelitian yang akan dilakukan dalam Tugas Akhir ini.





Gambar 3.1 *Flowchart* Penelitian

3.2 Langkah – Langkah Penelitian

Berikut ini adalah langkah-langkah yang dilakukan dalam mengerjakan tugas akhir.

3.2.1 Perumusan Masalah dan Tujuan Penelitian

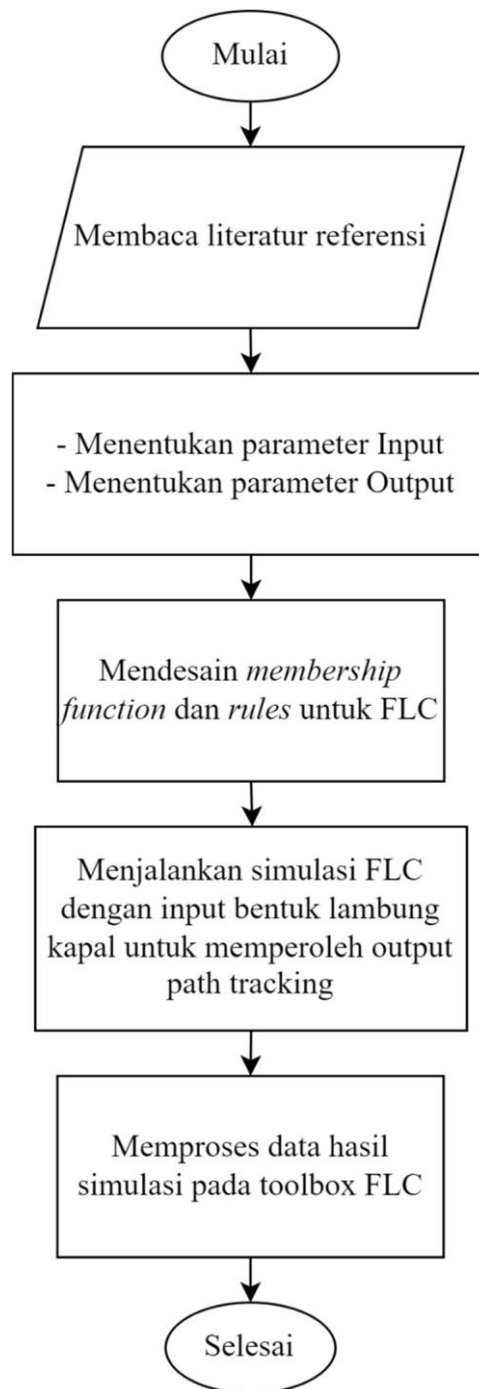
Penelitian diawali dengan perumusan masalah dimana penggunaan ROV untuk industri di bidang kelautan Indonesia belum maksimal, contohnya adalah inspeksi lambung kapal. Inspeksi lambung kapal dalam air masih dilakukan secara manual dengan menggunakan bantuan penyelam. Dari permasalahan ini penulis ingin meneliti tentang kontrol ROV untuk melakukan tugas inspeksi lambung kapal.

3.2.2 Studi Literatur

Studi literatur dilakukan dengan tujuan untuk mendalami landasan teori yang akan digunakan dalam penelitian ini. Studi literatur dilakukan dengan cara mencari jurnal – jurnal referensi yang meneliti topik serupa dengan penelitian yang akan dilakukan, jurnal dengan topik ROV dan kontrol FLC akan menjadi sasaran utama untuk mencari referensi literatur.

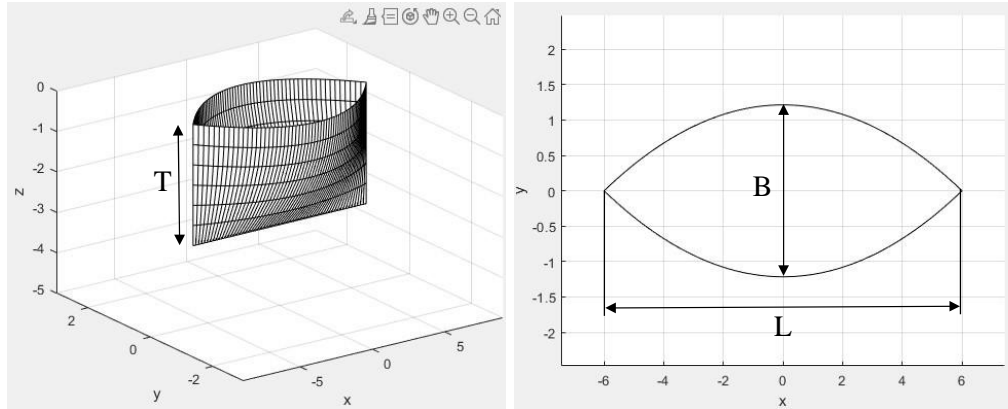
3.2.3 Pemodelan Kontrol FLC dan Pembuatan Data Penelitian

Kontrol yang digunakan dalam penelitian ini adalah *fuzzy logic controller* (FLC). FLC akan dibuat dengan menggunakan SIMULINK pada MATLAB, sistematis pemodelan FLC akan dibahas lebih lanjut pada *flowchart* berikut.



Gambar 3.2 *Flowchart* Tahapan Pemodelan Kontrol FLC

Untuk menyimulasikan bentuk dari lambung kapal sebagai salah satu *input* dari kontroler, persamaan 2.6 akan digunakan dengan ukuran $L = 12$ meter, $B = 2.5$ meter, dan $T = 3$ meter didapatkan visualisasi seperti pada gambar 3.4.



Gambar 3.3 Model Lambung Kapal 3D (kiri) dan 2D tampak atas (kanan)

Layer dengan posisi satu meter dari permukaan bentuk dari lambung kapal secara 2D tampak atas akan digunakan sebagai *input* untuk FLC. Fungsi dari bentuk lambung kapal 2D tampak atas dapat diperoleh koordinat x dan y dengan *software* Matlab. Sumber kode yang digunakan untuk menghasilkan bentuk 3D dari lambung kapal disertakan di lampiran.

3.2.4 Simulasi Kontrol Kedalaman

Simulasi kontrol kedalaman bertujuan untuk mengetahui apakah kontroler yang telah dirancang mampu untuk mengendalikan kedalaman dari ROV. Proses simulasi kontrol kedalaman dimulai dari pembuatan FLC pada *fuzzy logic toolbox* untuk mengatur pendorong ROV agar bisa mempertahankan posisi dalam *steady state*, pembuatan logika fuzzy akan berfokus pada pengendalian *output* gaya pendorong dari *thruster*. Proses simulasi kontrol kedalaman dilakukan dengan menggunakan SIMULINK pada *software* Matlab.

3.2.5 Simulasi *Path Tracking* pada Lambung Kapal

Simulasi *path tracking* dilakukan dengan tujuan untuk mengetahui apakah kontroler mampu untuk mengikuti bentuk dari profil lambung kapal. *Input* dari simulasi *path tracking* adalah jarak dari ROV ke lambung kapal dan gaya dari pendorong yang dihasilkan. Simulasi *path tracking* dianggap berhasil apabila ROV mampu bergerak mengikuti bentuk dari lambung kapal dengan menjaga jarak tertentu.

3.2.6 Kesimpulan Penelitian

Hasil simulasi kontrol kedalaman dan *path tracking* dari ROV akan di evaluasi untuk dilihat performa secara keseluruhan. Jika ROV belum berhasil mengikuti bentuk dari lambung kapal, maka akan dilakukan pengulangan pada tahap permodelan pada FLC dan pembuatan data penelitian untuk bentuk lambung kapal.

BAB IV ANALISIS DATA DAN PEMBAHASAN

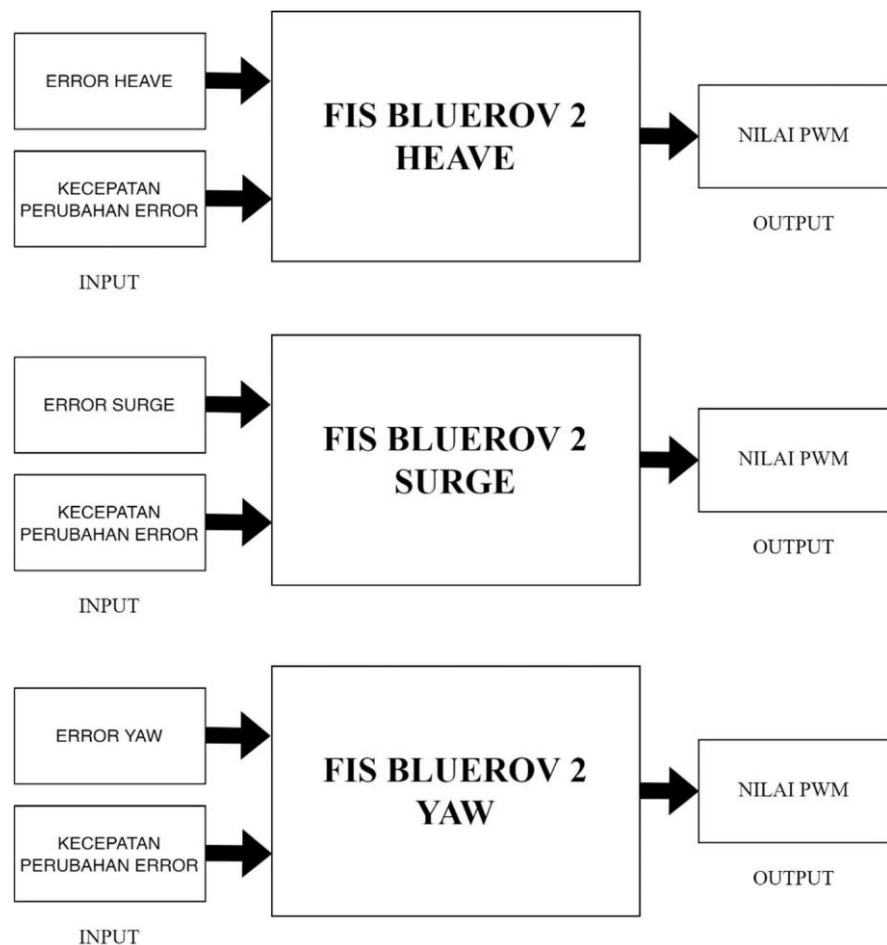
4.1 Perancangan *Fuzzy Logic Controller* (FLC)

Logika *fuzzy* adalah suatu logika yang memungkinkan nilai suatu keanggotaan berada di antara 0 sampai 1. Dengan menggunakan logika *fuzzy* suatu sistem dapat memiliki nilai keanggotaan di antara nilai 0 sampai 1. Hal ini yang membedakan logika *fuzzy* dengan logika klasik yang hanya mengizinkan nilai 0 atau 1. Maka dari itu logika *fuzzy* memiliki keunggulan yang tidak dimiliki logika lainnya.

Hal tersebut yang menjadikan logika *fuzzy* dapat diterapkan dalam pengendalian pergerakan dari BlueROV 2. Dimana logika *fuzzy* akan menjadi pengendali yang akan memberikan nilai keluaran berupa nilai PWM yang diperlukan agar BlueROV 2 dapat mencapai posisi yang diinginkan.

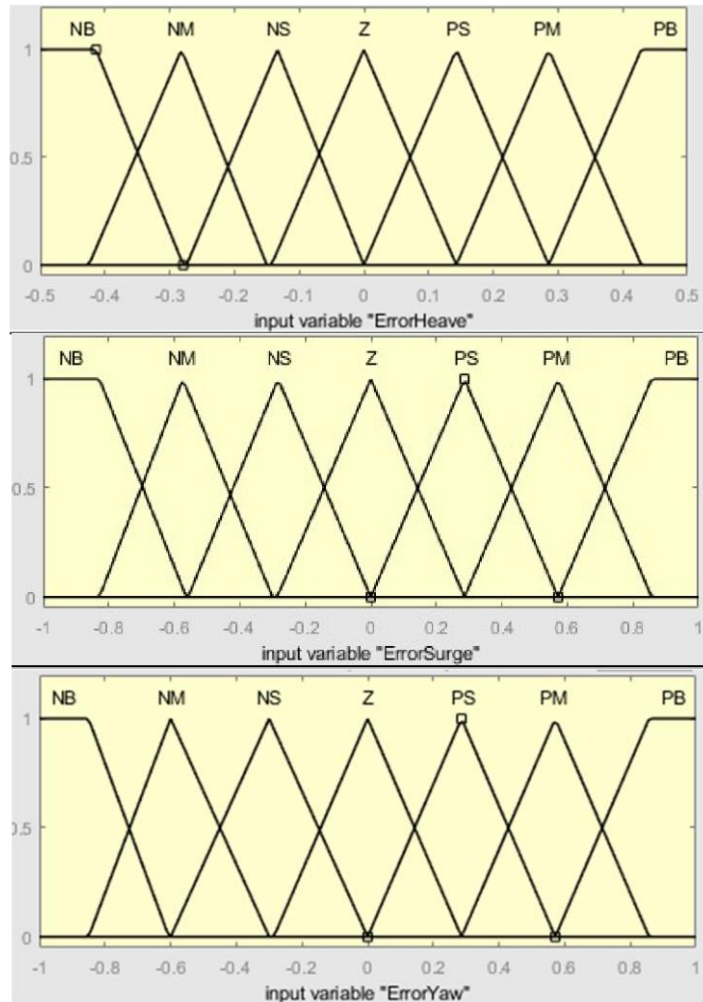
4.1.1 Fuzzifikasi

FLC yang di desain dalam tugas akhir ini ada tiga yang digunakan untuk mengontrol orientasi *surge*, *heave*, dan *yaw* dari BlueROV 2. FLC yang di desain secara umum menggunakan dua *input* yaitu nilai error dan kecepatan perubahan error. Nilai *output* dari FLC adalah nilai PWM yang berfungsi untuk mengendalikan output gaya dari *thruster* sehingga dapat digunakan untuk mengendalikan kedalaman BlueROV 2 dan arah tujuan dari BlueROV 2. Berikut ini adalah proses fuzzifikasi sistem kendali dari BlueROV2.



Gambar 4.1 Perancangan *Fuzzy Inference System* (FIS) untuk BlueROV 2

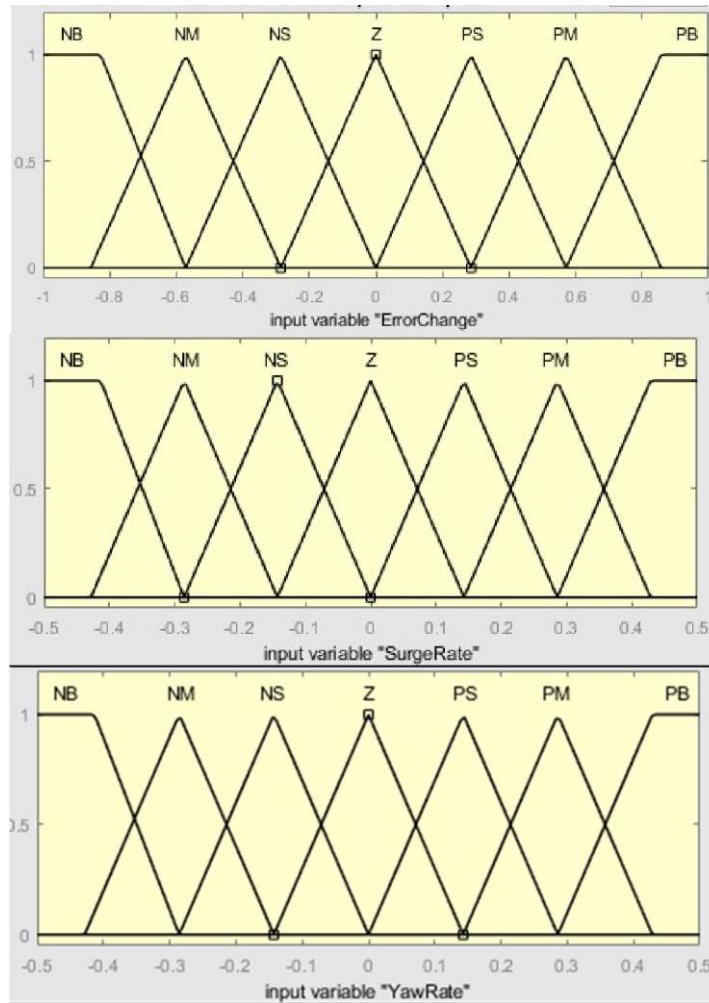
Pada gambar 4.1 Terdapat hubungan antara *input*, *rule base* dan *output* dari sistem *fuzzy* yang dirancang, dimana gambar menunjukkan dua *input* akan masuk ke dalam *fuzzy inference system* (FIS) kemudian akan dilanjutkan ke *output*. Gambar dari fungsi keanggotaan error *heave*, *surge* dan *yaw* secara berurutan ditampilkan dalam gambar 4.2 Sebagai berikut.



Gambar 4.2 Fungsi keanggotaan error *heave*, *surge*, dan *yaw* logika *fuzzy*

Gambar 4.2 menampilkan fungsi keanggotaan error *heave*, *surge*, dan *yaw* secara berurutan yang dijadikan sebagai *input* dari *fuzzy logic controller* yang dirancang. Fungsi keanggotaan error dibagi menjadi tujuh yaitu, *Negative Big* (NB), *Negative Medium* (NM), *Negative Small* (NS), *Zero* (Z), *Positive Small* (PS), *Positive Medium* (PM), *Positive Big* (PB).

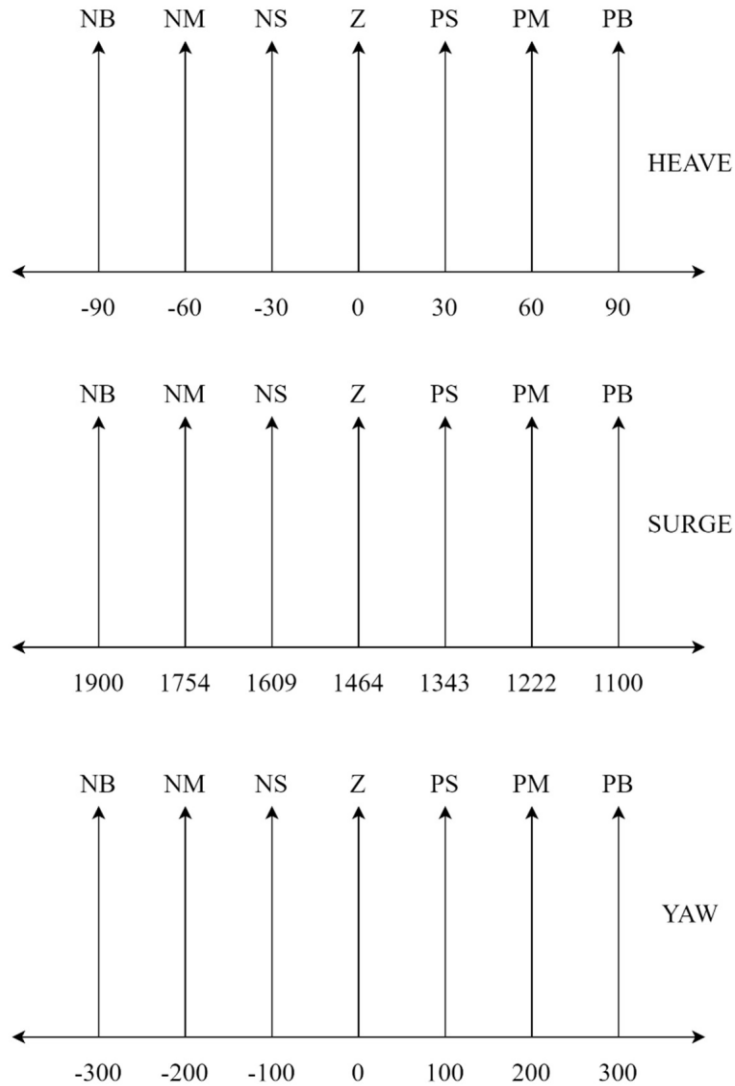
Range fungsi keanggotaan *input* dari *fuzzy logic controller* diatur sebesar -0.5 hingga 0.5 untuk *heave*, -1 hingga 1 untuk *surge*, dan -1 hingga 1 untuk *yaw*. Untuk fungsi keanggotaan NB dan PB diatur dengan tipe trapesium (*trapmf*) dan fungsi keanggotaan lainnya diatur dengan tipe segitiga (*trimf*).



Gambar 4.3 Fungsi keanggotaan *rate of change of error heave, surge, dan yaw*

Gambar 4.3 menampilkan fungsi keanggotaan *rate of change of error heave, surge, dan yaw* secara berurutan yang dijadikan sebagai *input* dari *fuzzy logic controller* yang dirancang.

Range fungsi keanggotaan *input* dari *fuzzy logic controller* diatur sebesar -1 hingga 1 untuk *heave*, -0.5 hingga 0.5 untuk *surge*, dan -0.5 hingga 0.5 untuk *yaw*. Untuk fungsi keanggotaan NB dan PB diatur dengan tipe trapesium (*trapmf*) dan fungsi keanggotaan lainnya diatur dengan tipe segitiga (*trimf*).



Gambar 4.4 Fungsi keanggotaan *output* dari *heave*, *surge*, dan *yaw*

Gambar 4.4 Menampilkan fungsi keanggotaan *output* nilai PWM untuk *heave*, *surge*, dan *yaw* secara berurutan. Fungsi keanggotaan error dibagi menjadi tujuh yaitu, *Negative Big* (NB), *Negative Medium* (NM), *Negative Small* (NS), *Zero* (Z), *Positive Small* (PS), *Positive Medium* (PM), *Positive Big* (PB). Nilai keluaran dari setiap *fuzzy logic* berbeda untuk tiap DoF. *Range* fungsi keanggotaan *output* dari *fuzzy logic controller* diatur sebesar -90 hingga 90 untuk *heave*, 1900 hingga 1100 untuk *surge*, dan -300 hingga 300 untuk *yaw*.

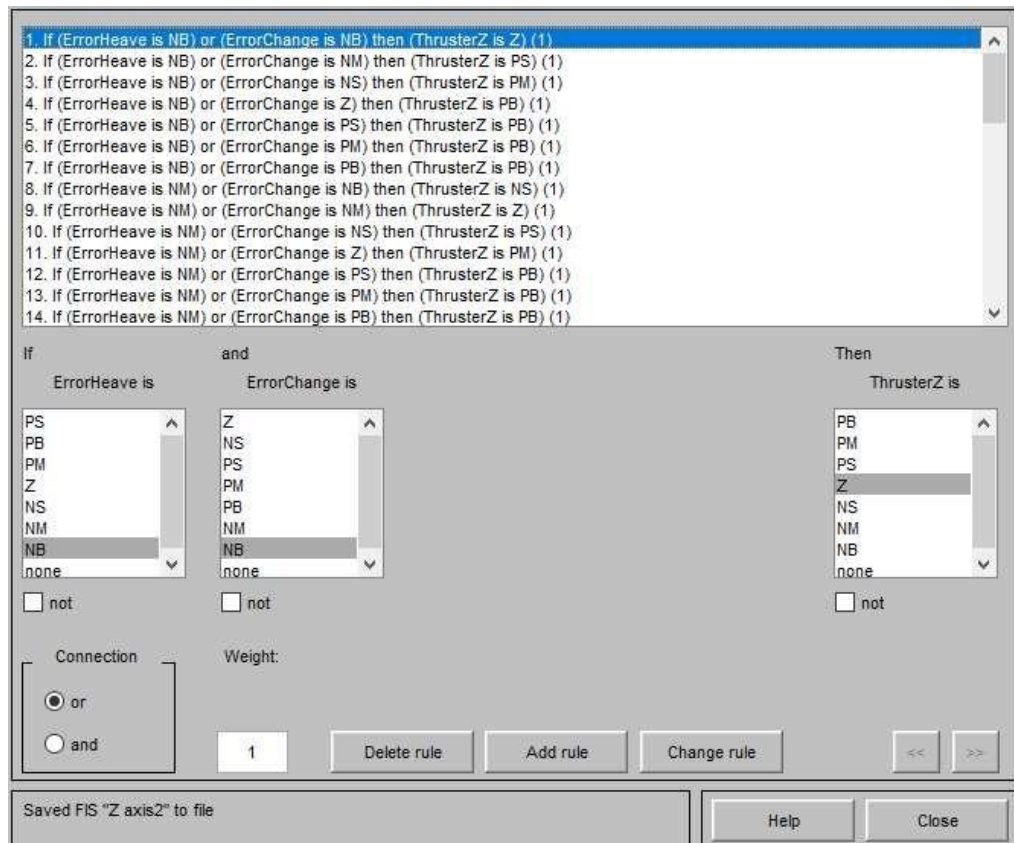
4.1.2 Rule Base

Rule base adalah kumpulan aturan dari FLC yang dibangun dengan logika “*if-then*” yang digunakan untuk menyatakan hubungan antara *input* dan *output* sistem agar tujuan dari pengendalian dapat dicapai. Berikut ini adalah tabel *rule base* yang digunakan dalam merancang FLC.

Tabel 4.1 Basis aturan untuk merancang *Fuzzy Logic Controller*

c/e	NB	NM	NS	Z	PS	PM	PB
NB	Z	NS	NM	NB	NB	NB	NB
NM	PS	Z	NS	NM	NB	NB	NB
NS	PM	PS	Z	NS	NM	NB	NB
Z	PB	PM	PS	Z	NS	NM	NB
PS	PB	PB	PM	PS	Z	NS	NM
PM	PB	PB	PB	PM	PS	Z	NS
PB	PB	PB	PB	PB	PM	PS	Z

Tabel 4.1 menampilkan tentang hubungan basis aturan antara *input* dan *output* logika *fuzzy* sebagai logika yang digunakan untuk mengendalikan BlueROV 2. Tabel 4.1 dijadikan sebagai dasar penentuan hubungan “*if-then*” pada *fuzzy logic controller*. Gambar 4.5 Dibawah merupakan *rule editor* dari *fuzzy logic controller*.



Gambar 4.5 *Rule editor* dari *fuzzy logic designer*

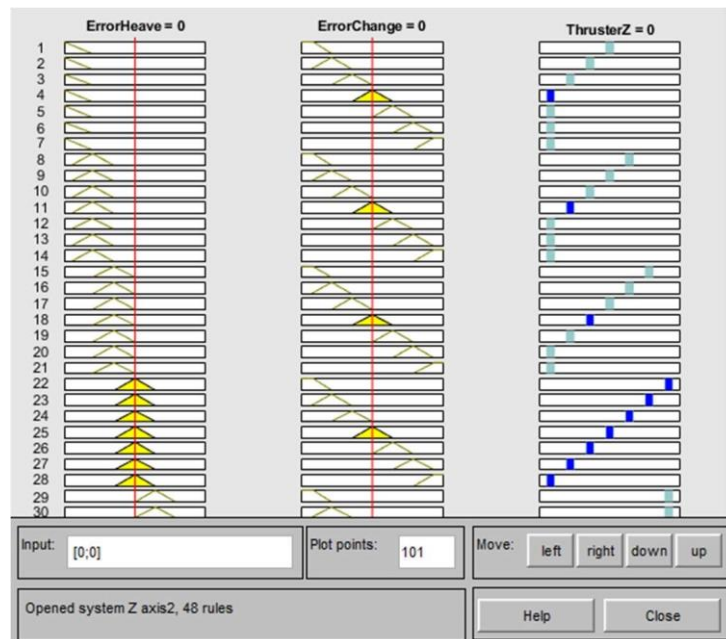
Setelah membuat basis aturan pada tabel 4.1, maka aturan-aturan tersebut diaplikasikan kedalam *rule editor* dari *fuzzy logic designer* seperti pada gambar 4.5. Pada *rule editor*, hubungan antara *input* dan *output* diatur secara individual dengan menggunakan aturan yang telah dibuat pada tabel 4.1. Berikut ini adalah contoh algoritma *fuzzy* untuk *heave* yang telah dirancang berisi total 49 aturan :

1. If (ErrorHeave is NB) or (ErrorChange is NB) then (ThrusterZ is Z) (1)
2. If (ErrorHeave is NB) or (ErrorChange is NM) then (ThrusterZ is PS) (1)
3. If (ErrorHeave is NB) or (ErrorChange is NS) then (ThrusterZ is PM) (1)
4. If (ErrorHeave is NB) or (ErrorChange is Z) then (ThrusterZ is PB) (1)
5. If (ErrorHeave is NB) or (ErrorChange is PS) then (ThrusterZ is PB) (1)
6. If (ErrorHeave is NB) or (ErrorChange is PM) then (ThrusterZ is PB) (1)
7. If (ErrorHeave is NB) or (ErrorChange is PB) then (ThrusterZ is PB) (1)
8. If (ErrorHeave is NM) or (ErrorChange is NB) then (ThrusterZ is NS) (1)
9. If (ErrorHeave is NM) or (ErrorChange is NM) then (ThrusterZ is Z) (1)
10. If (ErrorHeave is NM) or (ErrorChange is NS) then (ThrusterZ is PS) (1)
11. If (ErrorHeave is NM) or (ErrorChange is Z) then (ThrusterZ is PM) (1)
12. If (ErrorHeave is NM) or (ErrorChange is PS) then (ThrusterZ is PB) (1)
13. If (ErrorHeave is NM) or (ErrorChange is PM) then (ThrusterZ is PB) (1)
14. If (ErrorHeave is NM) or (ErrorChange is PB) then (ThrusterZ is PB) (1)
15. If (ErrorHeave is NS) or (ErrorChange is NB) then (ThrusterZ is NM) (1)
16. If (ErrorHeave is NS) or (ErrorChange is NM) then (ThrusterZ is NS) (1)
17. If (ErrorHeave is NS) or (ErrorChange is NS) then (ThrusterZ is Z) (1)
18. If (ErrorHeave is NS) or (ErrorChange is Z) then (ThrusterZ is PS) (1)
19. If (ErrorHeave is NS) or (ErrorChange is PS) then (ThrusterZ is PM) (1)
20. If (ErrorHeave is NS) or (ErrorChange is PM) then (ThrusterZ is PB) (1)
21. If (ErrorHeave is NS) or (ErrorChange is PB) then (ThrusterZ is PB) (1)
22. If (ErrorHeave is Z) or (ErrorChange is NB) then (ThrusterZ is NB) (1)
23. If (ErrorHeave is Z) or (ErrorChange is NM) then (ThrusterZ is NM) (1)
24. If (ErrorHeave is Z) or (ErrorChange is NS) then (ThrusterZ is NS) (1)
25. If (ErrorHeave is Z) or (ErrorChange is Z) then (ThrusterZ is Z) (1)
26. If (ErrorHeave is Z) or (ErrorChange is PS) then (ThrusterZ is PS) (1)
27. If (ErrorHeave is Z) or (ErrorChange is PM) then (ThrusterZ is PM) (1)
28. If (ErrorHeave is Z) or (ErrorChange is PB) then (ThrusterZ is PB) (1)
29. If (ErrorHeave is PS) or (ErrorChange is NB) then (ThrusterZ is NB) (1)
30. If (ErrorHeave is PS) or (ErrorChange is NM) then (ThrusterZ is NB) (1)
31. If (ErrorHeave is PS) or (ErrorChange is NS) then (ThrusterZ is NM) (1)
32. If (ErrorHeave is PS) or (ErrorChange is Z) then (ThrusterZ is NS) (1)
33. If (ErrorHeave is PS) or (ErrorChange is PS) then (ThrusterZ is Z) (1)
34. If (ErrorHeave is PS) or (ErrorChange is PM) then (ThrusterZ is PS) (1)
35. If (ErrorHeave is PS) or (ErrorChange is PB) then (ThrusterZ is PM) (1)
36. If (ErrorHeave is PM) or (ErrorChange is NB) then (ThrusterZ is NB) (1)
37. If (ErrorHeave is PM) or (ErrorChange is NM) then (ThrusterZ is NB) (1)
38. If (ErrorHeave is PM) or (ErrorChange is NS) then (ThrusterZ is NB) (1)
39. If (ErrorHeave is PM) or (ErrorChange is Z) then (ThrusterZ is NM) (1)
40. If (ErrorHeave is PM) or (ErrorChange is PS) then (ThrusterZ is NS) (1)
41. If (ErrorHeave is PM) or (ErrorChange is PM) then (ThrusterZ is Z) (1)
42. If (ErrorHeave is PM) or (ErrorChange is PB) then (ThrusterZ is PS) (1)
43. If (ErrorHeave is PB) or (ErrorChange is NB) then (ThrusterZ is NB) (1)
44. If (ErrorHeave is PB) or (ErrorChange is NM) then (ThrusterZ is NB) (1)
45. If (ErrorHeave is PB) or (ErrorChange is NS) then (ThrusterZ is NB) (1)
46. If (ErrorHeave is PB) or (ErrorChange is Z) then (ThrusterZ is NB) (1)
47. If (ErrorHeave is PB) or (ErrorChange is PS) then (ThrusterZ is NM) (1)
48. If (ErrorHeave is PB) or (ErrorChange is PM) then (ThrusterZ is NS) (1)
49. If (ErrorHeave is PB) or (ErrorChange is PB) then (ThrusterZ is Z) (1)

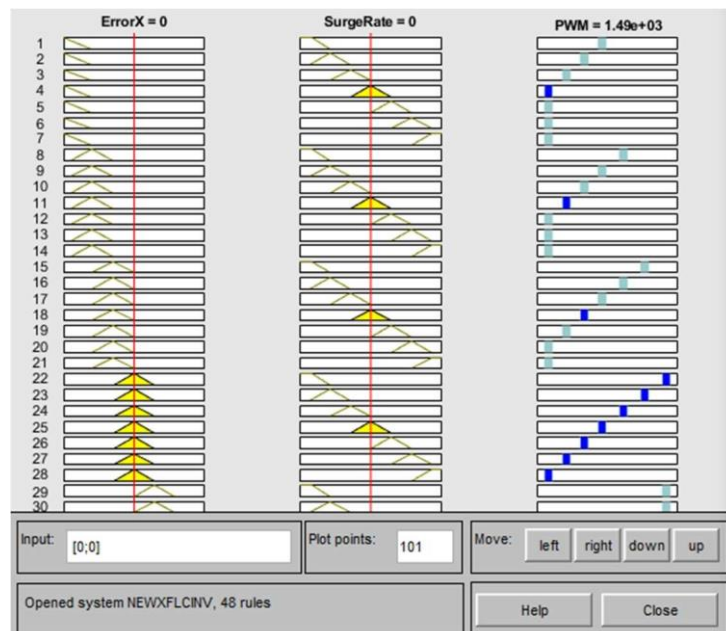
4.1.3 Fuzzy Inference System

Fuzzy inference system atau sistem inferensi *fuzzy* adalah suatu proses merumuskan pemetaan dari *input* yang diberikan menuju ke *output* dengan menggunakan logika *fuzzy*. Hasil dari pemetaan tersebut akan memberikan dasar dari penarikan keputusan atau membedakan pola-pola.

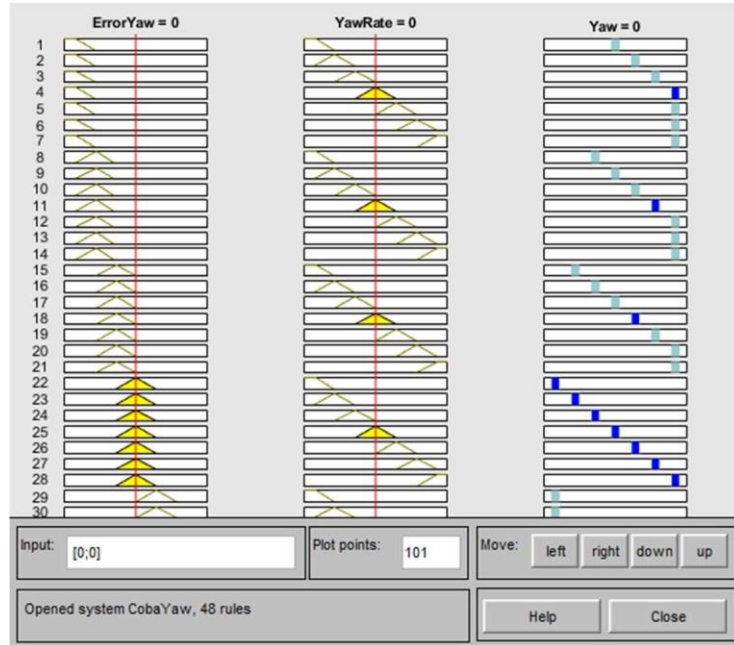
Dalam tugas akhir ini, proses inferensi *fuzzy* berguna untuk mendapatkan *output* berupa sinyal kendali dari logika *fuzzy* untuk memberikan nilai PWM berdasarkan basis aturan yang telah dibuat pada tabel 4.1. Berikut ini adalah gambar *rule viewer* dari FLC yang telah dirancang.



Gambar 4.6 Proses inferensi logika *fuzzy* untuk kontrol *heave*

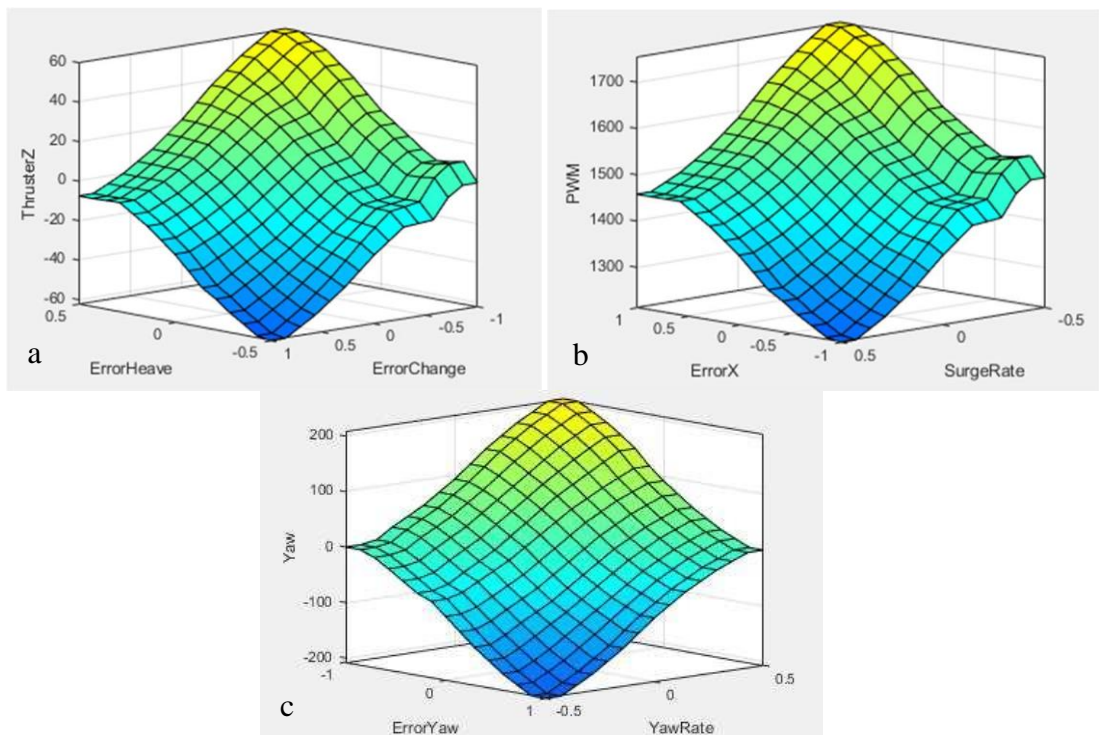


Gambar 4.7 Proses inferensi logika *fuzzy* untuk kontrol *surge*



Gambar 4.8 Proses inferensi logika *fuzzy* untuk kontrol *yaw*

Rule viewer pada gambar diatas akan bergerak sesuai dengan aturan yang telah dibuat. *Rule base* dari logika *fuzzy* untuk BlueROV 2 dapat ditampilkan dalam bidang 3D. Untuk menampilkan visualisasi 3D dari *rule base* logika *fuzzy* dapat menggunakan bantuan dari *surface viewer* pada *fuzzy logic designer*. Berikut ini adalah gambar dari *surface* logika *fuzzy*.



Gambar 4.9 *Surface viewer* logika *fuzzy* untuk *heave* (a), *surge* (b), dan *yaw* (c)

Pada gambar 4.9 menampilkan *surface viewer* dari sistem logika *fuzzy* yang telah dirancang. Dalam bidang 3D *surface viewer* menampilkan hubungan antara *input* (*error* dan *rate of change of error*) dengan *output* (nilai PWM). Warna pada *surface* memberikan informasi berupa seberapa besar nilai PWM akan diberikan ke sistem. Dengan urutan dari yang terkecil

hingga terbesar sesuai urutan warna: biru, biru muda, hijau, kuning.

4.1.4 Defuzzifikasi

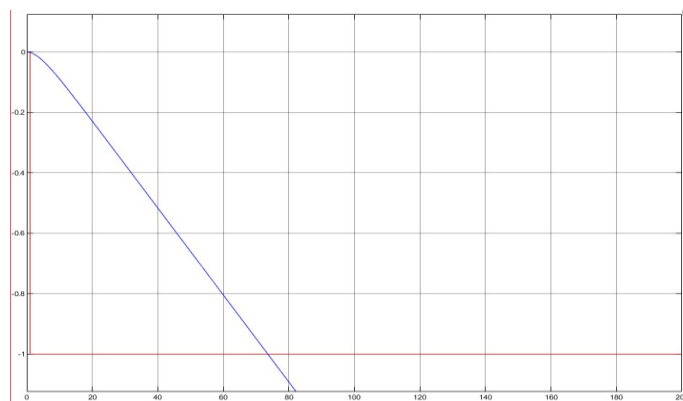
Defuzzifikasi adalah proses pengubahan hasil dari inferensi *fuzzy set* pada *inference engine* menjadi hasil yang *crisp*. Proses defuzzifikasi merupakan proses yang penting karena aktuator berupa pendorong hanya dapat menerima masukan berupa besaran real dari controller untuk melakukan perintah *path tracking*.

Metode defuzzifikasi yang digunakan adalah metode wtaver (*weighted average*). Pada metode ini nilai *crisp output* didapatkan setelah merata-rata hasil perkalian setiap komponen dengan faktor yang mencerminkan kepentingannya.

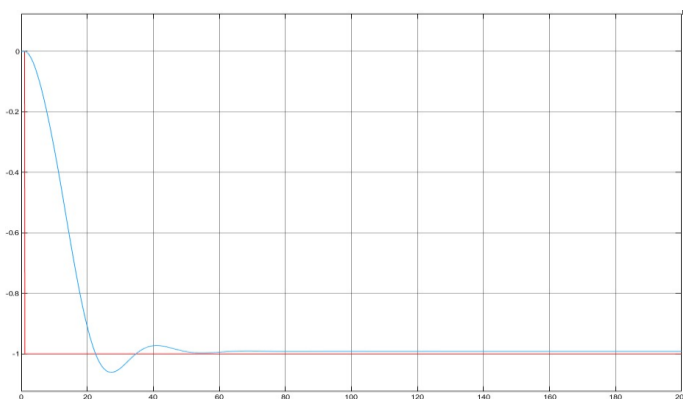
4.2 Pengujian Open Loop dan Close Loop untuk Heave

Pengujian performa dari *fuzzy logic controller* (FLC) untuk mengontrol pergerakan translasi di sumbu z atau nama lainnya gerak *heave* untuk mengetahui respon ROV terhadap waktu dan posisi. Pengujian *open loop* dilakukan untuk mengetahui respon ROV tanpa menggunakan pengendali. Sedangkan uji *close loop* dilakukan untuk mengetahui respon ROV apabila diberikan pengendali FLC.

Uji *open loop* dan *close loop* dilakukan terhadap model *state-space* dari BlueROV 2 yang digunakan sesuai dengan referensi dengan diberikan nilai *set point* berupa sinyal *step* sebesar satu meter di bawah air untuk mempresentasikan kedalaman yang diinginkan. Grafik respon sistem ditampilkan pada gambar 4.10 dan 4.11.



Gambar 4.10 Grafik respon *open loop* untuk kedalaman satu meter



Gambar 4.11 Grafik respon *close loop* untuk kedalaman satu meter

Uji *open loop* pada sistem dengan memasukkan nilai PWM sebesar $1665 \mu s$ menghasilkan respon dari BlueROV 2 terus menurun dan melebihi nilai dari *set point* setelah 74 detik. Hal ini dikarenakan tidak ada pengendali yang dapat mempertahankan nilai respon agar tetap *isteady*.

Untuk uji *close loop* dengan memasukkan nilai PWM sebesar $1665 \mu s$ menghasilkan respon posisi kedalaman yang mampu mencapai nilai *set point* setelah 22,7 detik. Dari grafik terlihat adanya *overshoot* sebesar 6,16%. Sistem dapat mempertahankan respon yang stabil setelah 50,6 detik. Sistem memiliki *error steady state* paling besar 0,95%. Hal ini menandakan bahwa penggunaan FLC dapat memberikan respon yang stabil pada BlueROV 2 untuk menjaga posisi kedalaman tertentu. Berikut ini adalah tabel perbandingan posisi pada sumbu z yang diinginkan dengan posisi aktual dari BlueROV 2.

Tabel 4.2 Data posisi sumbu z *setpoint* dan aktual dalam pengujian *open loop*

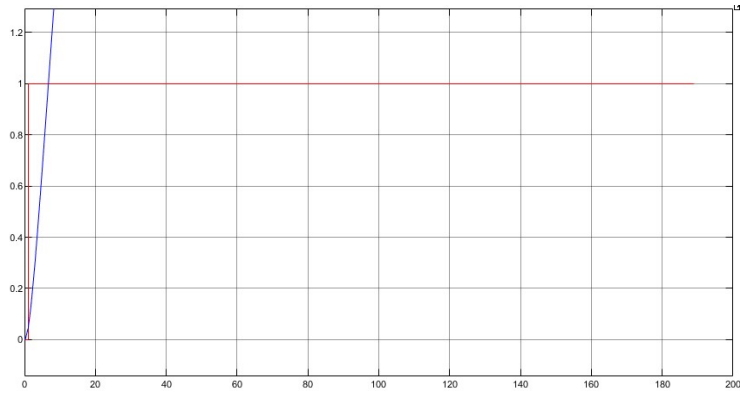
No.	Z desired	Z actual	Time (second)
1	-1	0	0
2	-1	-0.229	20
3	-1	-0.516	40
4	-1	-0.804	60
5	-1	-1.092	80
6	-1	-1.380	100
7	-1	-1.669	120
8	-1	-1.957	140
9	-1	-2.245	160
10	-1	-2.533	180

Tabel 4.3 Data posisi sumbu z *setpoint* dan aktual dalam pengujian *close loop*

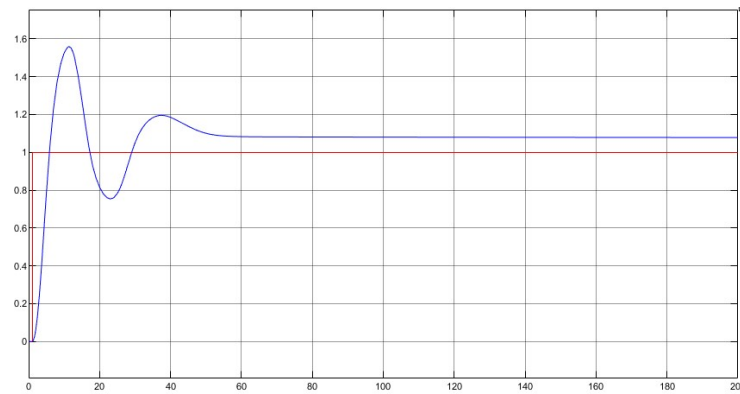
No.	Z desired	Z actual	Time (second)
1	-1	0	0
2	-1	-0.910	20
3	-1	-0.910	40
4	-1	-0.994	60
5	-1	-0.992	80
6	-1	-0.991	100
7	-1	-0.992	120
8	-1	-0.992	140
9	-1	-0.992	160
10	-1	-0.992	180

4.3 Pengujian *Open Loop* dan *Close Loop* untuk *Surge*

Pengujian performa dari *fuzzy logic controller* (FLC) untuk mengontrol pergerakan translasi di sumbu x atau nama lainnya gerak *surge* untuk mengetahui respon ROV terhadap waktu dan posisi. Uji *open loop* dan *close loop* dilakukan terhadap model *state-space* dari BlueROV2 yang digunakan sesuai dengan referensi dengan diberikan nilai *set point* berupa sinyal *step* sebesar satu meter. Grafik respon sistem ditampilkan pada gambar 4.12 dan 4.13.



Gambar 4.12 Grafik respon *open loop* untuk $x = 1$ m



Gambar 4.13 Grafik respon *close loop* untuk $x = 1$ m

Dari gambar 4.12 diketahui sistem memerlukan waktu 7 detik untuk mencapai *set point* sebelum pada akhirnya pergerakan dari ROV pada sumbu x akan terus meningkat dikarenakan tidak adanya pengendali yang mampu mempertahankan nilainya setelah mencapai *set point*. Pengujian dilakukan dengan memberi input nilai PWM sebesar $1665 \mu\text{s}$ pada pendorong 1 dan 2, sedangkan input nilai PWM pada pendorong 3 dan 4 diberikan sebesar $1300 \mu\text{s}$.

Untuk pengujian *close loop* menghasilkan respon posisi x yang mampu mencapai *set point* dalam waktu 6 detik. Dari grafik terlihat adanya *overshoot* dengan nilai sebesar 55,9%. Sistem dapat mempertahankan respon yang stabil setelah 54 detik. Sistem memiliki *error steady state* sebesar 8,5%.

Berikut ini adalah tabel perbandingan posisi pada sumbu x yang diinginkan dengan posisi aktual dari BlueROV 2.

Tabel 4.4 Data posisi sumbu x *setpoint* dan aktual dalam pengujian *open loop*

No.	X desired	X actual	Time (second)
1	1	0	0
2	1	3.593	20
3	1	7.512	40
4	1	11.430	60
5	1	15.349	80
6	1	19.268	100
7	1	23.186	120
8	1	27.105	140
9	1	31.024	160
10	1	34.942	180

Tabel 4.5 Data posisi sumbu x *setpoint* dan aktual dalam pengujian *close loop*

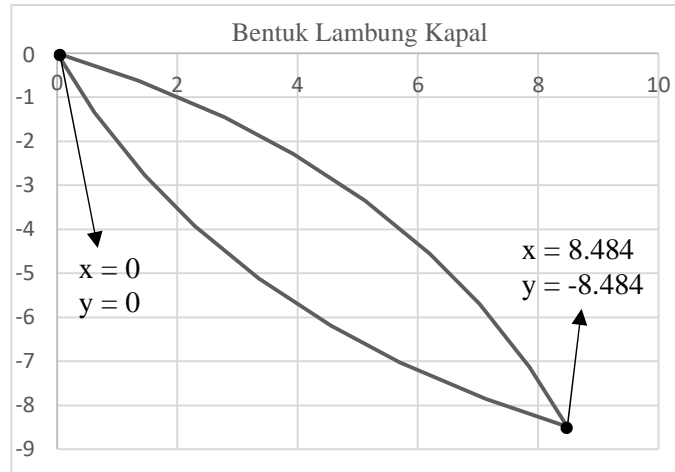
No.	X desired	X actual	Time (second)
1	1	0	0
2	1	0.816	20
3	1	1.184	40
4	1	1.082	60
5	1	1.081	80
6	1	1.080	100
7	1	1.079	120
8	1	1.079	140
9	1	1.078	160
10	1	1.078	180

4.4 Pengujian Performansi Sistem *Path Tracking* BlueROV2

Pengujian performansi sistem *tracking path* BlueROV2 dilakukan dengan tujuan untuk mengetahui respon dinamik dari sistem pengendali FLC yang telah dirancang. Uji performansi dilakukan dengan memasukkan nilai *set point* berupa koordinat jalur yang akan menghasilkan bentuk dari lambung kapal yang telah diubah menjadi koordinat x dan y. Berikut ini adalah data dari koordinat bentuk lambung kapal yang telah dibuat.

Tabel 4.6 Data Koordinat dari Bentuk Lambung Kapal

No.	Koordinat x	Koordinat y
1	0	0
2	1.354117	-0.62548
3	2.783813	-1.45819
4	3.931698	-2.2899
5	5.12575	-3.35825
6	6.194098	-4.5523
7	7.025813	-5.70019
8	7.858517	-7.12988
9	8.484	-8.484
10	7.129883	-7.85852
11	5.700188	-7.02581
12	4.552302	-6.1941
13	3.35825	-5.12575
14	2.289902	-3.9317
15	1.458188	-2.78381
16	0.625483	-1.35412
17	0	0



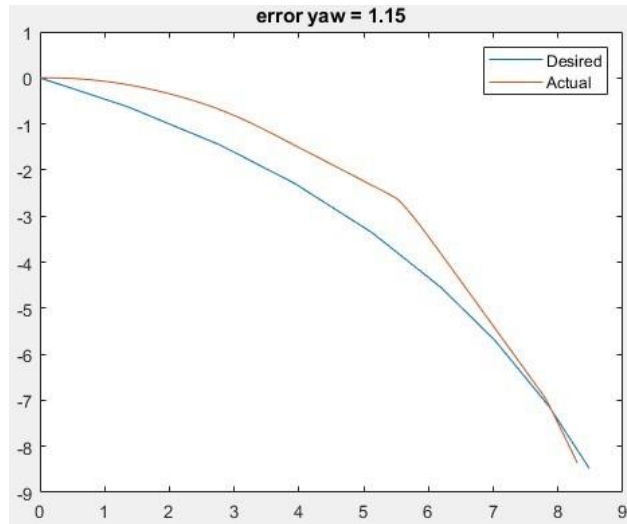
Gambar 4.14 Visualisasi dari Bentuk Lambung Kapal

Simulasi *path tracking* dilakukan sebanyak dua kali. Simulasi pertama dilakukan dengan memposisikan BlueROV 2 pada koordinat $x = 0$ dan $y = 0$. Simulasi kedua dilakukan dengan memposisikan BlueROV 2 pada koordinat $x = 8.484$ dan $y = -8.484$. Hal ini dilakukan karena saat simulasi berjalan terdapat *singularity* pada model *state space* dari referensi yang digunakan. Salah satunya adalah apabila BlueROV 2 diberi perintah untuk berbelok ke arah kiri maka nilai kecepatan dari BlueROV 2 menjadi tak hingga. Oleh sebab itu simulasi dijalankan dua kali dengan mengatur posisi BlueROV 2 agar hanya perlu untuk belok ke kanan saja. Jika BlueROV 2 mengalami perubahan posisi *yaw* yang sangat cepat ke arah kanan maka akan mengakibatkan juga kecepatan BlueROV 2 akan menjadi tak hingga, oleh sebab itu kecepatan untuk bergerak ke arah *yaw* positif maksimum adalah sebesar $0,015$ rad/s. Pada model yang diberikan oleh referensi, pengendalian DoF yang diizinkan hanya satu di waktu yang sama, sehingga untuk melakukan manuver belok BlueROV 2 harus diam dahulu lalu melakukan manuver belok sesuai sudut yang dibutuhkan.

Karena ada beberapa limitasi yang berasal dari model referensi maka bentuk dari lambung kapal disesuaikan dengan kebutuhan yaitu dengan cara menempatkan bentuk lambung kapal pada kuadran empat di koordinat kartesian sehingga BlueROV 2 hanya perlu bergerak ke arah kanan saja. Diperlukan *switch* untuk mengatur BlueROV 2 yang berfungsi untuk menjaga posisi dengan memanfaatkan *error yaw* agar BlueROV 2 tidak bermanuver ke arah kiri.

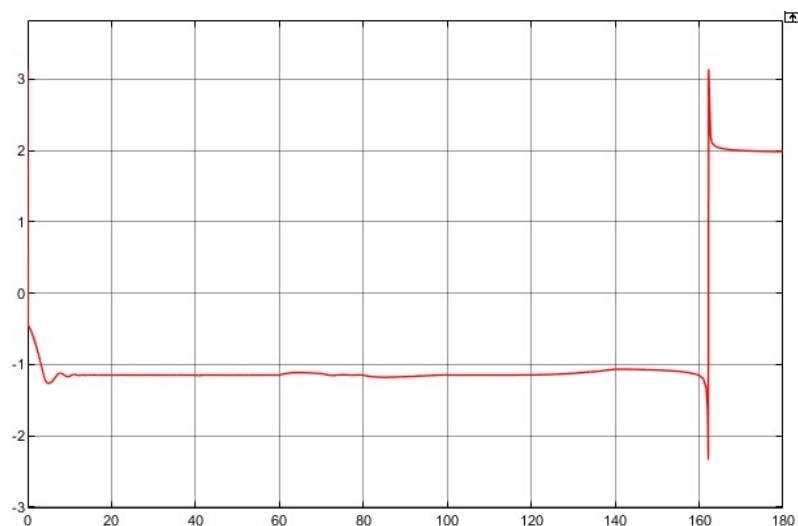
4.4.1 Hasil Simulasi dengan *Starting Point* $x = 0$ dan $y = 0$ dengan $e_{yaw} = -1.15^\circ$

Berikut ini adalah hasil simulasi sistem pengendalian BlueROV 2 dengan *input* berupa koordinat bentuk lambung kapal. Nilai *error yaw* dibutuhkan sebagai *threshold* agar BlueROV 2 dapat menjaga posisinya disebelah kiri dari *set point* untuk menghindari terjadinya *output* tak hingga dari model.

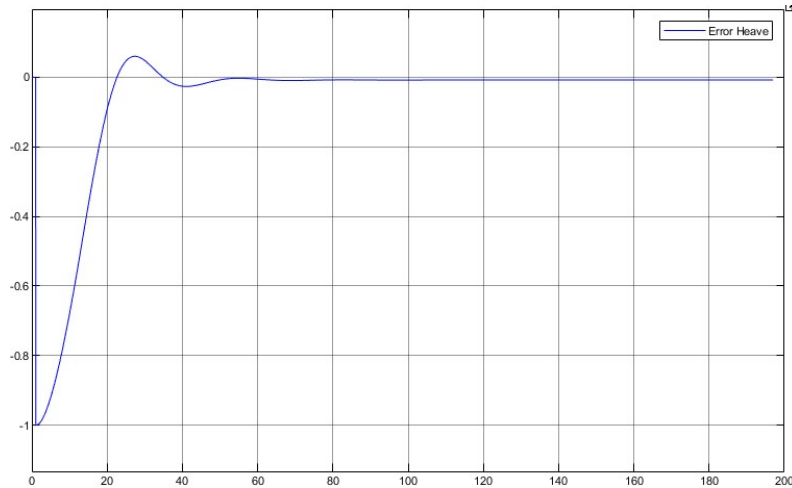


Gambar 4.15 Lintasan *desired* dan aktual BlueROV 2 untuk $e_{yaw} = -1.15^\circ$

Gambar 4.15 adalah grafik hasil simulasi sistem pengendalian gerak BlueROV 2 dengan menggunakan FLC sebagai kontroler. Simulasi yang dilakukan didapatkan hasil bahwa sistem yang telah dibuat dapat mengikuti lintasan berupa bentuk dari lambung kapal yang dijadikan sebagai *set point*. Selama BlueROV 2 melintasi trayektori tersebut, terdapat *root square error* (RSE) maksimum sebesar 2.67926 m dan minimum 0.3821 m. Selisih simpangan antar trayektori *desired* dan aktual disebabkan oleh keterbatasan model dinamik dari BlueROV 2 yang hanya mengizinkan untuk mengontrol satu DoF dalam satu waktu, sehingga BlueROV 2 tertinggal dari *set point* yang telah ditetapkan.



Gambar 4.16 Respon nilai error $yaw = -1.15^\circ$

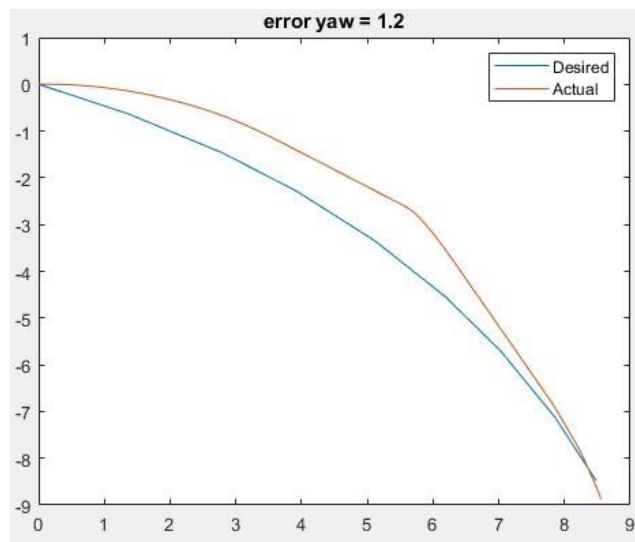


Gambar 4.17 Respon nilai error *heave* dengan simulasi error *yaw* = -1.15°

Gambar 4.16 dan 4.17 menampilkan grafik untuk error *yaw* dan *heave*. Selama simulasi dilakukan dalam kurun waktu 180 detik, *error* sudut *yaw* dapat terjaga stabil pada -1.15° , namun pada detik ke-163 *error yaw* langsung mengalami *spike* yang diakibatkan oleh BlueROV 2 yang posisinya berada di sebelah kanan dari *desired* trayektori dan tidak mampu untuk mengembalikan posisinya agar berada di sebelah kiri dari *desired* trayektori. Terdapat *error heave* maksimum sebesar -0.09313885 m dan minimum sebesar -0.00542276 m.

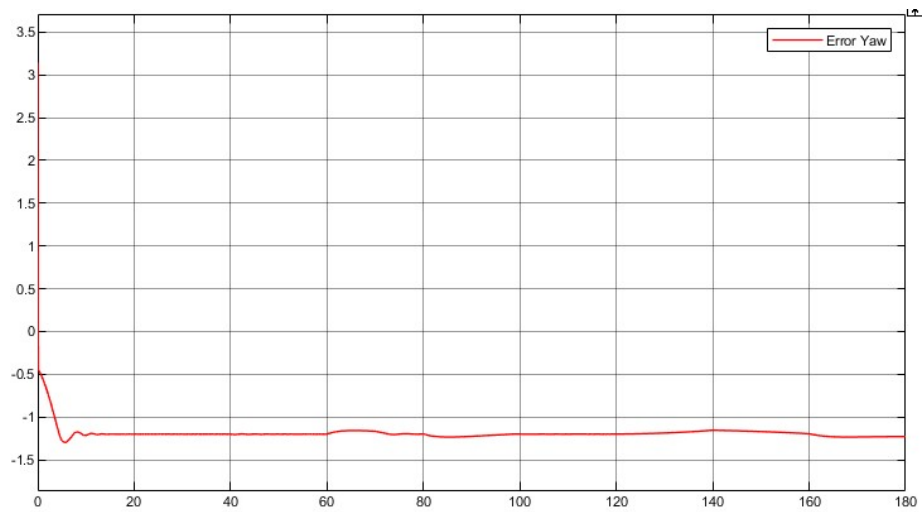
4.4.2 Hasil Simulasi dengan *Starting Point* $x = 0$ dan $y = 0$ dengan $e_{yaw} = -1.2^\circ$

Berikut ini adalah hasil simulasi *path tracking* BlueROV 2 dengan menggunakan *thresholder switch* berupada *error yaw* sebesar -1.2° .

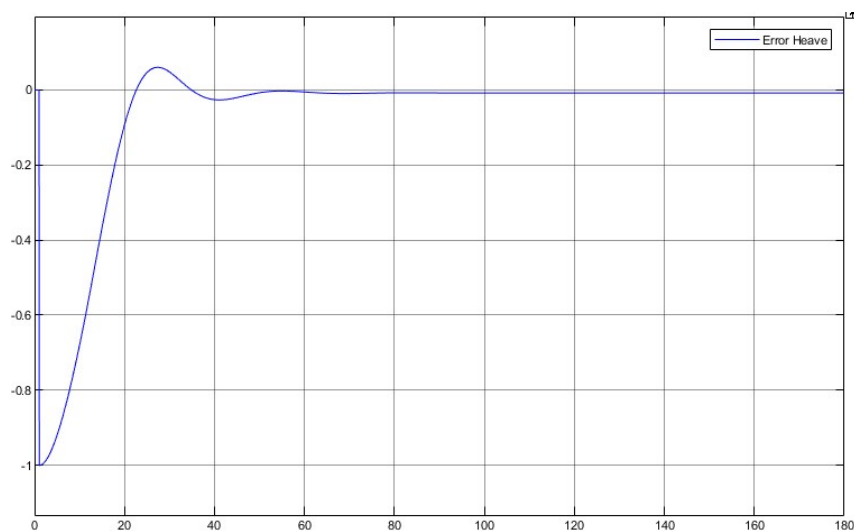


Gambar 4.18 Lintasan *desired* dan aktual BlueROV 2 untuk $e_{yaw} = -1.2^\circ$

Gambar 4.18 menunjukkan perbandingan dari *desired* trayektori dengan *actual* trayektori. Dapat dilihat bahwa BlueROV 2 mampu mengikuti *path tracking* dengan nilai *error yaw* sebesar -1.2° . Selama BlueROV 2 melintasi trayektori tersebut, terdapat *root square error* (RSE) maksimum sebesar 2.604283 m dan minimum sebesar 0.261849 m. Hasil RSE yang didapat lebih kecil dibandingkan dengan menggunakan *threshold error yaw* sebesar -1.15° .



Gambar 4.19 Respon nilai error *yaw* = -1.2°

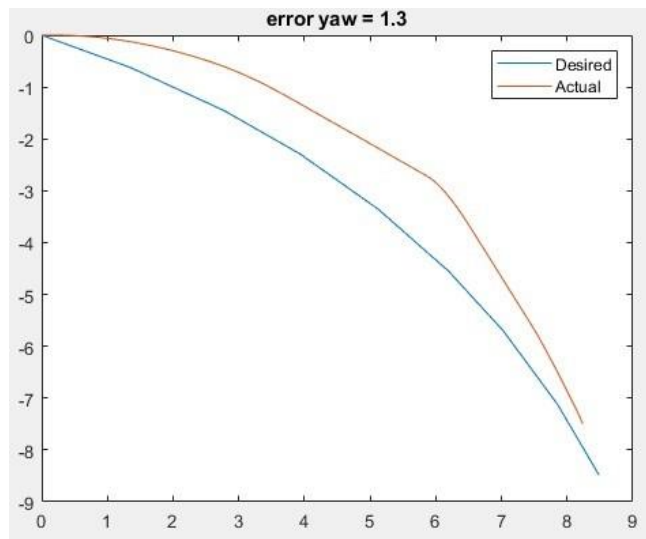


Gambar 4.20 Respon nilai error *heave* dengan simulasi error *yaw* = -1.2°

Pada gambar 4.19 dan 4.20 menampilkan grafik nilai error untuk *yaw* dan *heave*. Selama simulasi dilakukan dalam 180 detik, *error* sudut *yaw* dapat terjaga stabil pada -1.2° . Nilai *error yaw* terbesar yang terjadi pada detik ke 180 yaitu sebesar -1.22809° . Terdapat *error heave* maksimum sebesar -0.09304719 m dan minimum sebesar -0.0054242 m.

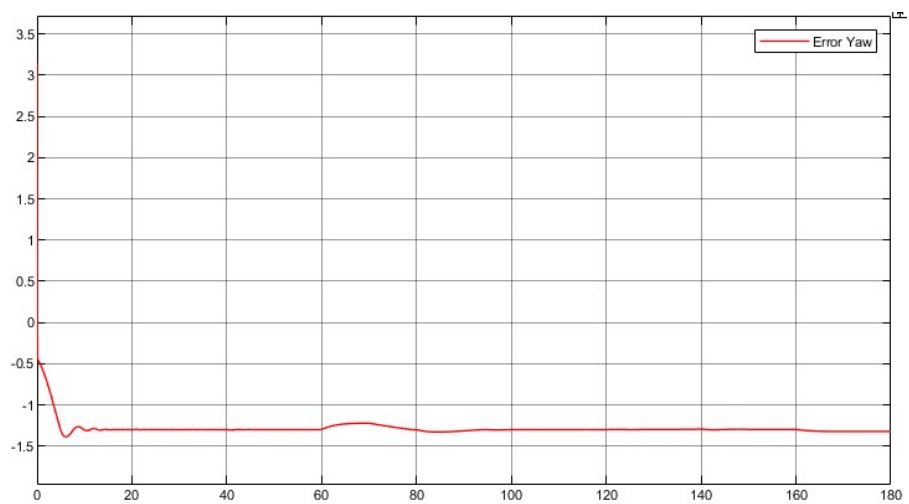
4.4.3 Hasil Simulasi dengan *Starting Point* $x = 0$ dan $y = 0$ dengan $e_{yaw} = -1.3^\circ$

Berikut ini adalah hasil simulasi *path tracking* BlueROV 2 dengan menggunakan *thresholder switch* berupada *error yaw* sebesar -1.3° .

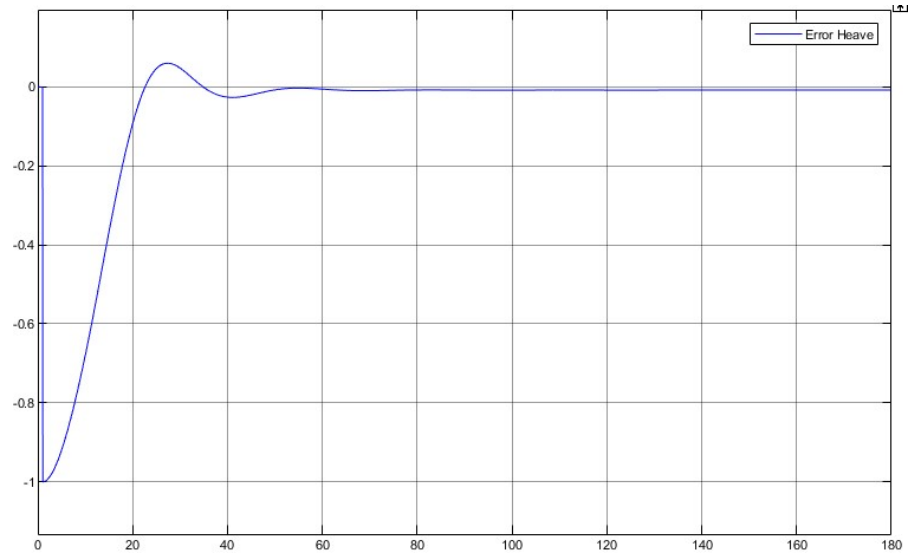


Gambar 4.21 Lintasan *desired* dan aktual BlueROV 2 untuk $e_{yaw} = -1.3^\circ$

Gambar 4.21 menunjukkan perbandingan dari *desired* trayektori dengan *actual* trayektori. Dapat dilihat BlueROV 2 mampu mengikuti *path tracking* dengan nilai *error yaw* sebesar -1.3° . Selama BlueROV 2 melintasi trayektori tersebut, terdapat *root square error* (RSE) maksimum sebesar 2.398793 m dan minimum sebesar 0.549204 m.



Gambar 4.22 Respon nilai *error yaw* = -1.3°



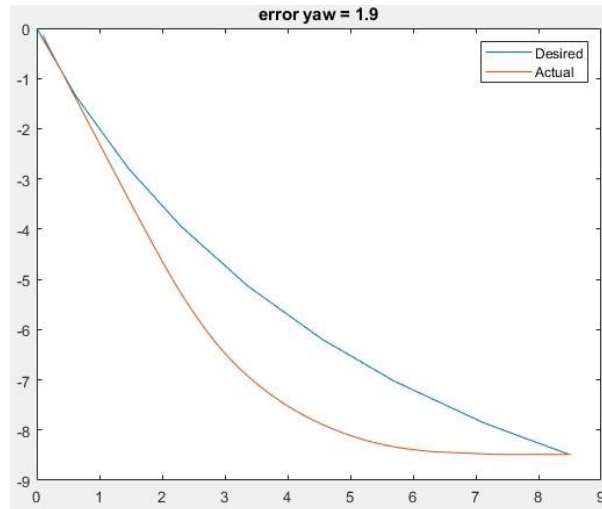
Gambar 4.23 Respon nilai error *heave* dengan simulasi error *yaw* = -1.3°

Pada gambar 4.22 dan 4.23 menampilkan grafik nilai error untuk *yaw* dan *heave*. Selama simulasi dilakukan dalam 180 detik, *error* sudut *yaw* dapat terjaga stabil pada -1.3° . Nilai *error yaw* terbesar yang terjadi pada detik ke 180 yaitu sebesar -1.32196° . Terdapat *error heave* maksimum sebesar -0.09304212 m dan minimum sebesar -0.005430348 m.

4.4.4 Hasil Simulasi dengan *Starting Point* $x = 8.484$ dan $y = -8.484$ dengan $e_{yaw} = 1.9^\circ$

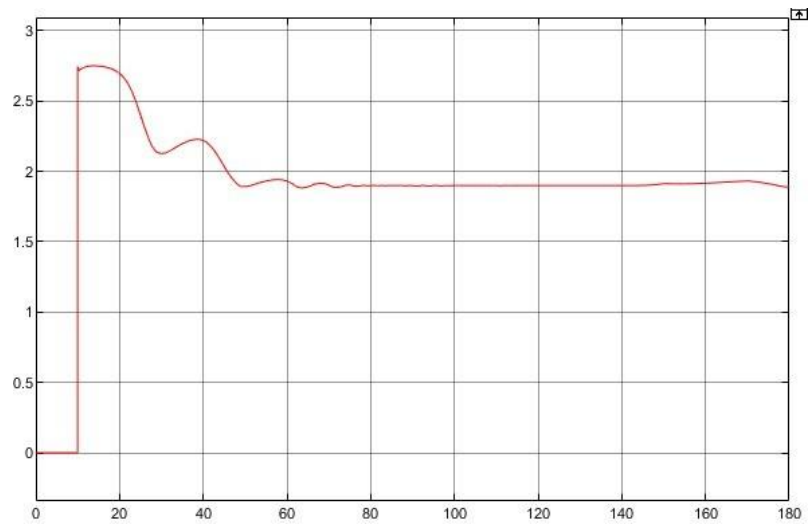
Dalam skenario simulasi kedua terdapat sedikit perbedaan, dimana pada sepuluh detik pertama digunakan untuk memutar drone sebesar 180° . Perbedaan nilai *error yaw* pada simulasi skenario pertama dan kedua terdapat perbedaan yang cukup besar, dimana pada skenario pertama pembacaan *error yaw* sesaat setelah *initial condition* adalah sebesar -0.4237° sedangkan pada simulasi skenario kedua pembacaan *error yaw* sesaat setelah *initial condition* adalah sebesar 2.76° . Oleh karena itu nilai *threshold* yang digunakan untuk simulasi skenario kedua berbeda dengan simulasi skenario pertama.

Berikut ini adalah hasil simulasi skenario kedua yang dimulai dari $x = 8.484$ dan $y = -8.484$. Masih sama seperti simulasi dengan skenario yang pertama, nilai *error yaw* dibutuhkan sebagai *threshold* agar BlueROV 2 dapat menjaga posisinya disebelah kiri dari *set point* untuk menghindari terjadinya *output* tak hingga dari model.

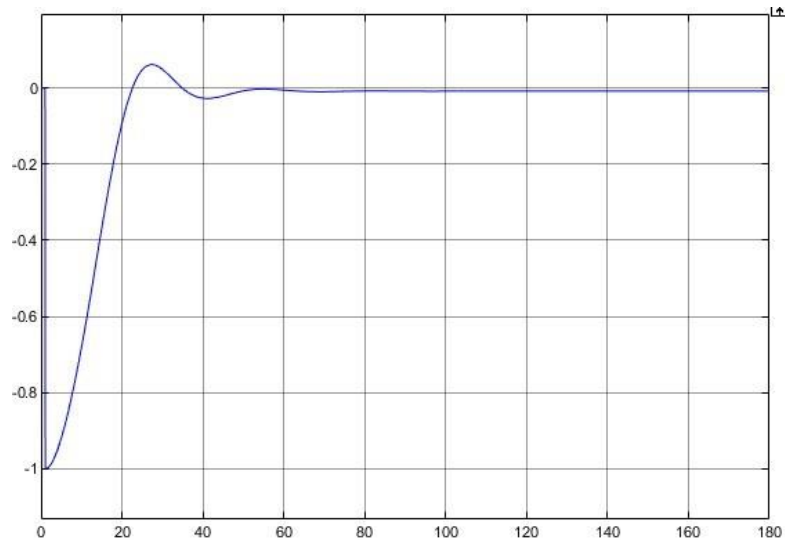


Gambar 4.24 Lintasan *desired* dan aktual BlueROV 2 untuk $e_{yaw} = 1.9^\circ$

Gambar 4.24 menunjukkan perbandingan dari *desired* trayektori dengan *actual* trayektori. Dapat dilihat bahwa BlueROV 2 mampu mengikuti *path tracking* dengan nilai *error yaw* sebesar 1.9° . Selama BlueROV 2 melintasi trayektori tersebut, terdapat *root square error* (RSE) maksimum sebesar 1.542101 m dan minimum sebesar 0.467576 m. Dalam simulasi menggunakan.



Gambar 4.25 Respon nilai error yaw = 1.9°

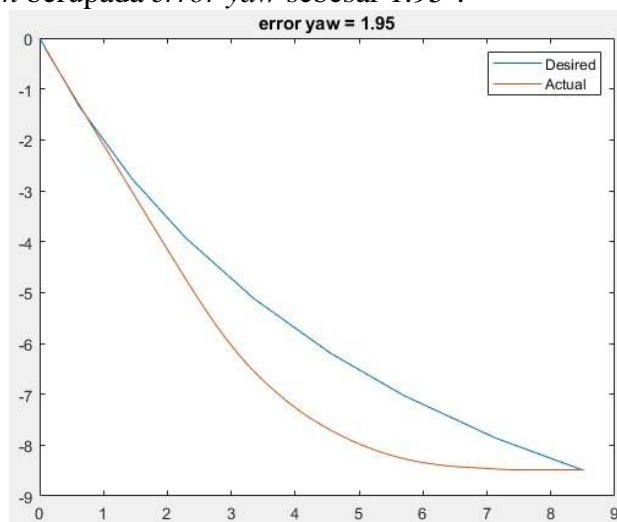


Gambar 4.26 Respon nilai error *heave* dengan simulasi error *yaw* = 1.9°

Pada gambar 4.25 dan 4.26 menampilkan grafik nilai error untuk *yaw* dan *heave*. Selama simulasi dilakukan dalam 180 detik, *error* sudut *yaw* dapat terjaga stabil pada 1.9° . Nilai *error yaw* terbesar yang terjadi pada detik ke 20 yaitu sebesar 2.698146. Terdapat *error heave* maksimum sebesar -0.09304212 m dan minimum sebesar -0.00543035 m.

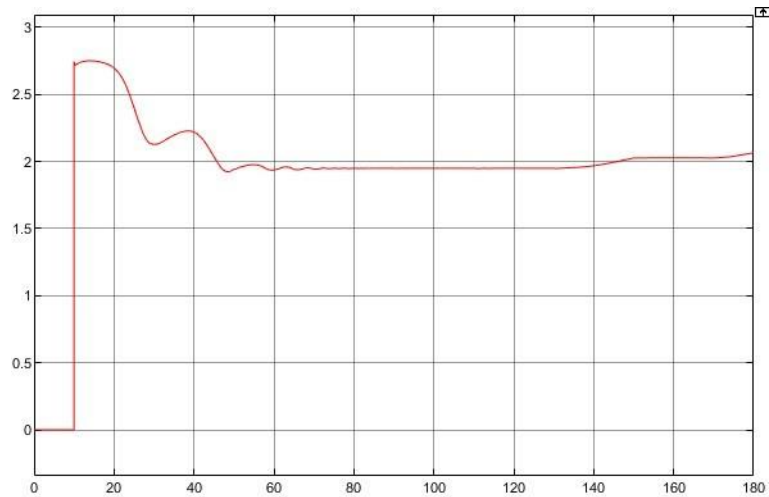
4.4.5 Hasil Simulasi dengan *Starting Point* $x = 8.484$ dan $y = -8.484$ dengan $e_{yaw} = 1.95^\circ$

Berikut ini adalah hasil simulasi *path tracking* BlueROV 2 dengan menggunakan *thresholder switch* berupakan *error yaw* sebesar 1.95° .

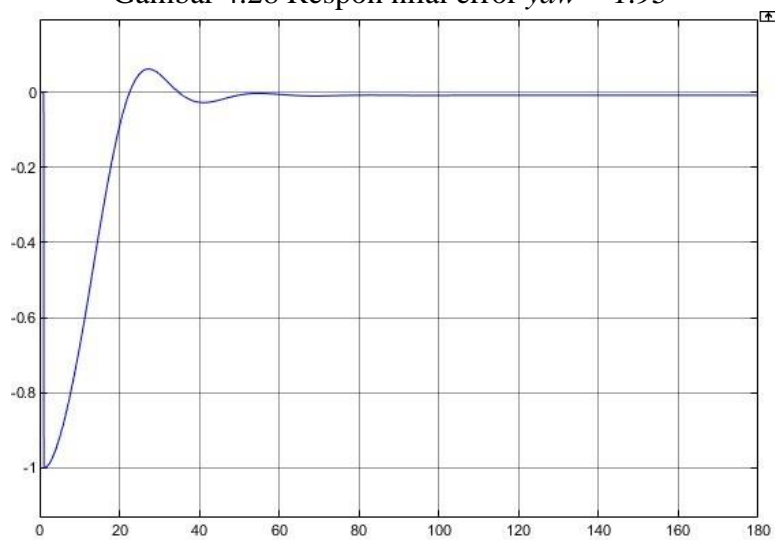


Gambar 4.27 Lintasan *desired* dan aktual BlueROV 2 untuk $e_{yaw} = 1.95^\circ$

Gambar 4.27 menunjukkan perbandingan dari *desired* trayektori dengan *actual* trayektori. Dapat dilihat bahwa BlueROV 2 mampu mengikuti *path tracking* dengan nilai *error yaw* sebesar 1.95° . Selama BlueROV 2 melintasi trayektori tersebut, terdapat *root square error* (RSE) maksimum sebesar 2.398793 m dan minimum sebesar 0.196904 m.



Gambar 4.28 Respon nilai error $yaw = 1.95^\circ$

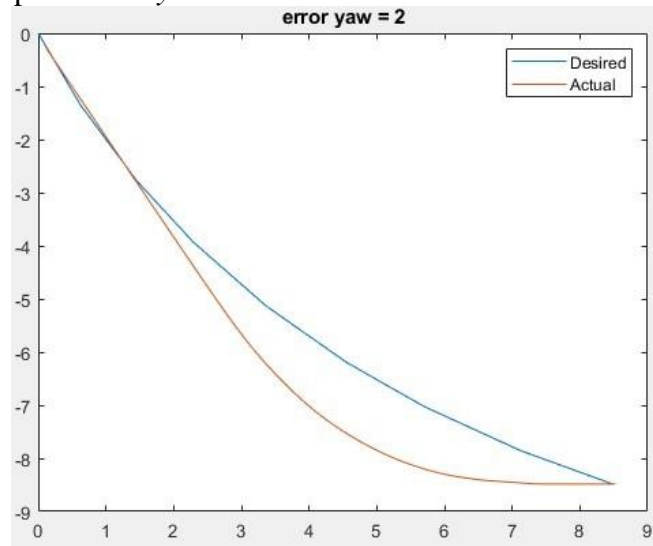


Gambar 4.29 Respon nilai error *heave* dengan simulasi error $yaw = 1.95^\circ$

Pada gambar 4.28 dan 4.29 menampilkan grafik nilai error untuk *yaw* dan *heave*. Selama simulasi dilakukan dalam 180 detik, *error* sudut *yaw* dapat terjaga stabil pada 1.95° . Nilai *error yaw* terbesar yang terjadi pada detik ke 180 yaitu sebesar 2.698146° . Terdapat *error heave* maksimum sebesar -0.0928315 m dan minimum sebesar -0.005419 m.

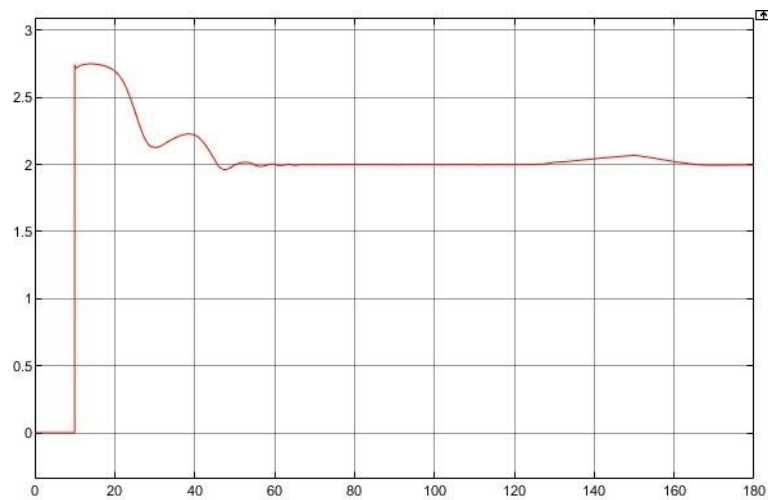
4.4.6 Hasil Simulasi dengan Starting Point $x = 8.484$ dan $y = -8.484$ dengan $e_{yaw} = 2^\circ$

Berikut ini adalah hasil simulasi *path tracking* BlueROV 2 dengan menggunakan *thresholder switch* berupada *error yaw* sebesar 2° .

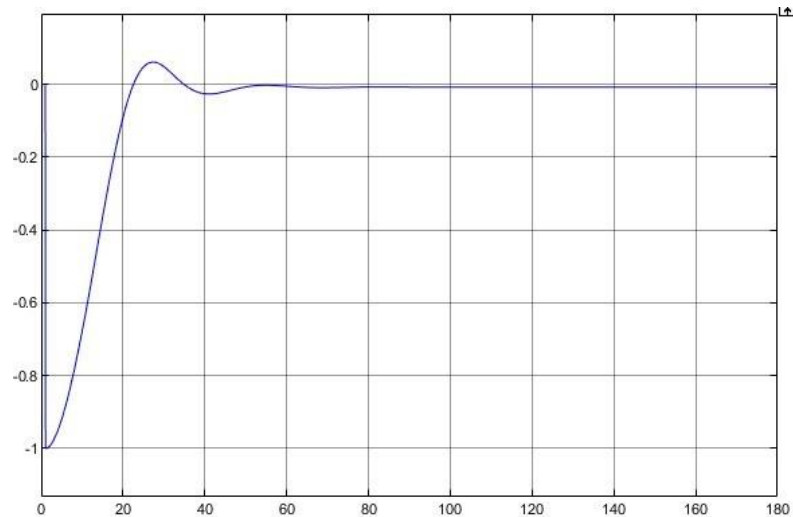


Gambar 4.30 Lintasan *desired* dan aktual BlueROV 2 untuk $e_{yaw} = 2^\circ$

Gambar 4.30 menunjukkan perbandingan dari *desired* trayektori dengan *actual* trayektori. Dapat dilihat bahwa BlueROV 2 mampu mengikuti *path tracking* dengan nilai *error yaw* sebesar 2° . Selama BlueROV 2 melintasi trayektori tersebut, terdapat *root square error* (RSE) maksimum sebesar 2.516389 m dan minimum sebesar 0.15128 m. Dalam simulasi menggunakan.



Gambar 4.31 Respon nilai *error yaw* = 2°

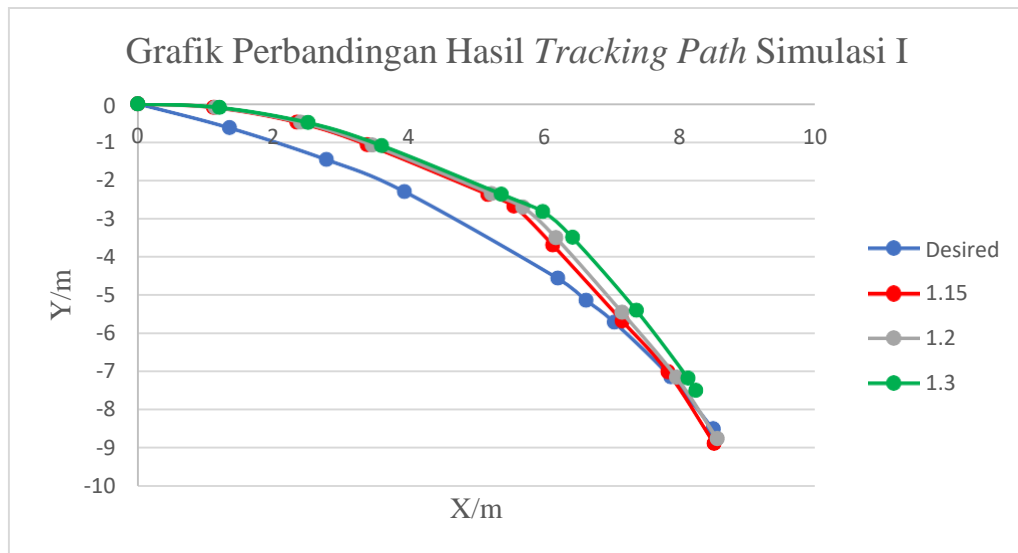


Gambar 4.32 Respon nilai error *heave* dengan simulasi error *yaw* = 2°

Pada gambar 4.31 dan 4.32 menampilkan grafik nilai error untuk *yaw* dan *heave*. Selama simulasi dilakukan dalam 180 detik, *error* sudut *yaw* dapat terjaga stabil pada 2°. Nilai *error yaw* terbesar yang terjadi pada detik ke 180 yaitu sebesar 2.698146°. Terdapat *error heave* maksimum sebesar -0.09058658 m dan minimum sebesar -0.00540397 m.

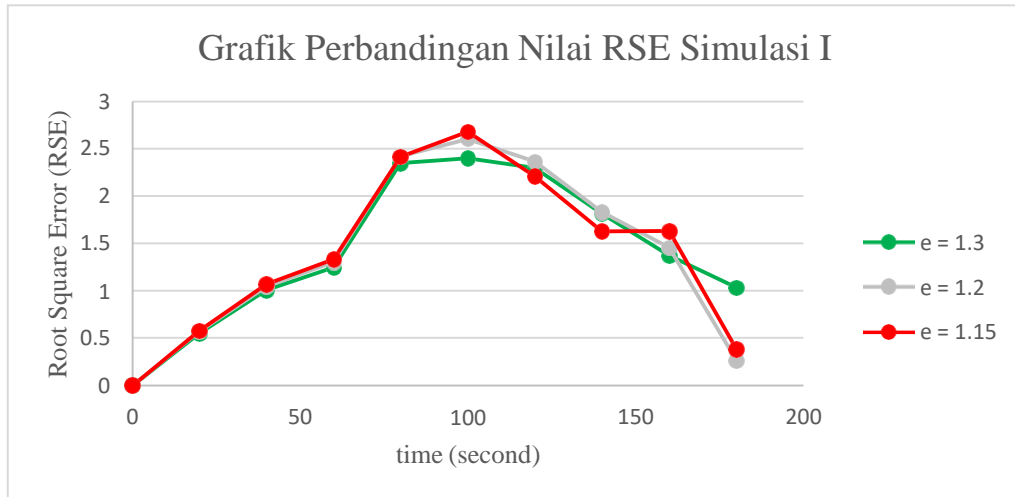
4.5 Perbandingan Hasil Simulasi dengan *Starting Point* x = 0 dan y = 0

Berikut ini adalah gambar grafik perbandingan hasil simulasi I yang dilakukan pada *starting point* x = 0 dan y = 0.



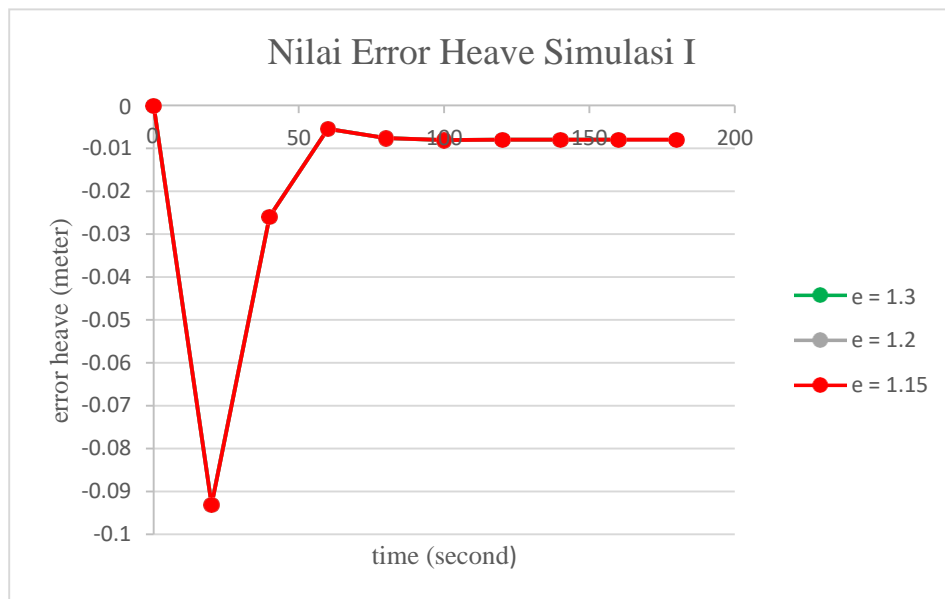
Gambar 4.33 Grafik Perbandingan Hasil *Tracking Path* Simulasi I

Dari gambar 4.33 terlihat bahwa hasil simulasi I dengan menggunakan ketiga *threshold error yaw* dapat mengikuti bentuk dari lambung kapal. Dengan menggunakan *threshold error yaw* sebesar 1.2° menghasilkan *path tracking* yang paling mendekati *desired path*.



Gambar 4.34 Grafik Perbandingan Nilai RSE Simulasi I

Dari gambar 4.34 dapat dilihat nilai RSE maksimum terdapat pada simulasi dengan *error yaw* sebesar 1.15° pada detik ke-100 dengan nilai 2.67926 dan untuk nilai RSE minimum paling kecil terdapat pada simulasi dengan *error yaw* sebesar 1.2° pada detik ke-180 dengan nilai 0.2618. Untuk hasil selengkapnya dapat dilihat di lampiran B1 sampai B3.

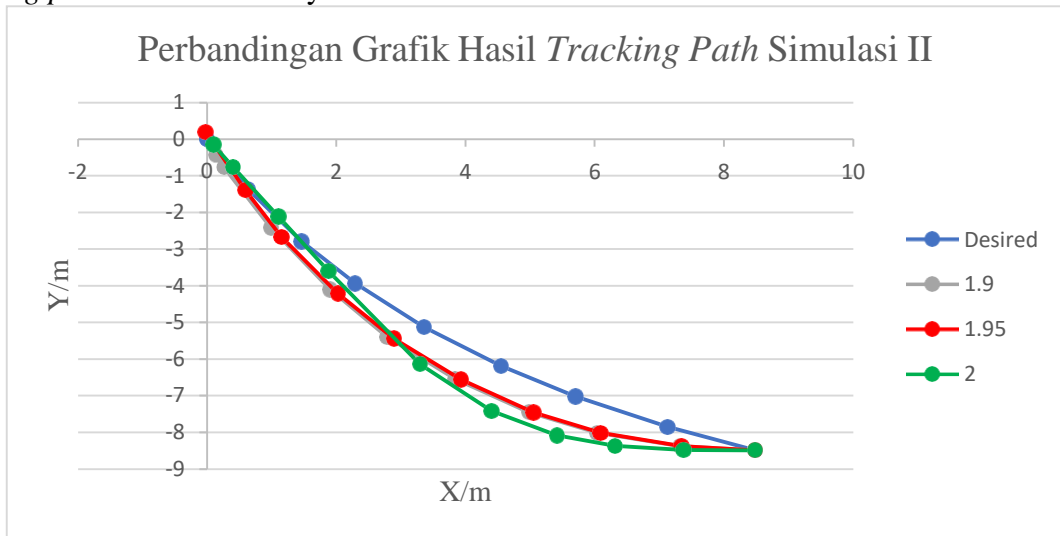


Gambar 4.35 Grafik Perbandingan Nilai *Error Heave* Simulasi I

Pada gambar 4.35 dapat dilihat gambar grafik perbandingan nilai *error heave* simulasi I dengan menggunakan tiga nilai *threshold* yang berbeda. Pada grafik terlihat hampir tidak tampak perbedaan yang besar dari ketiga *threshold* yang berbeda. Nilai *error heave* maksimum terdapat pada simulasi dengan *error yaw* sebesar 1.15° pada detik ke-20 dengan nilai -0.093189 m dan untuk nilai *error heave* minimum terdapat pada simulasi dengan *error yaw* sebesar 1.2° pada detik ke-60 dengan nilai -0.00542 m.

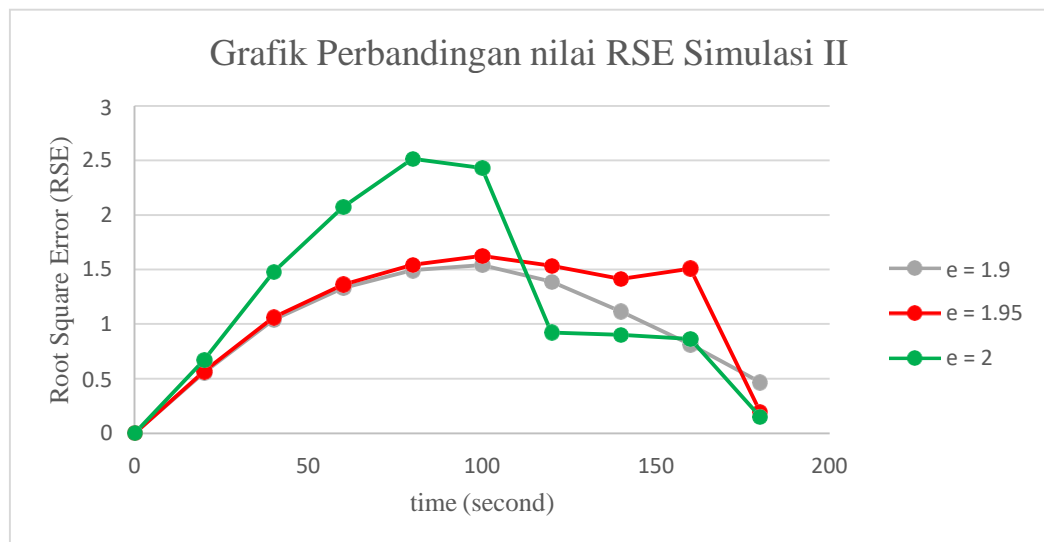
4.6 Perbandingan Hasil Simulasi dengan *Starting Point* $x = 8.484$ dan $y = -8.484$

Berikut ini adalah gambar grafik perbandingan hasil simulasi II yang dilakukan pada *starting point* $x = 8.484$ dan $y = -8.484$.



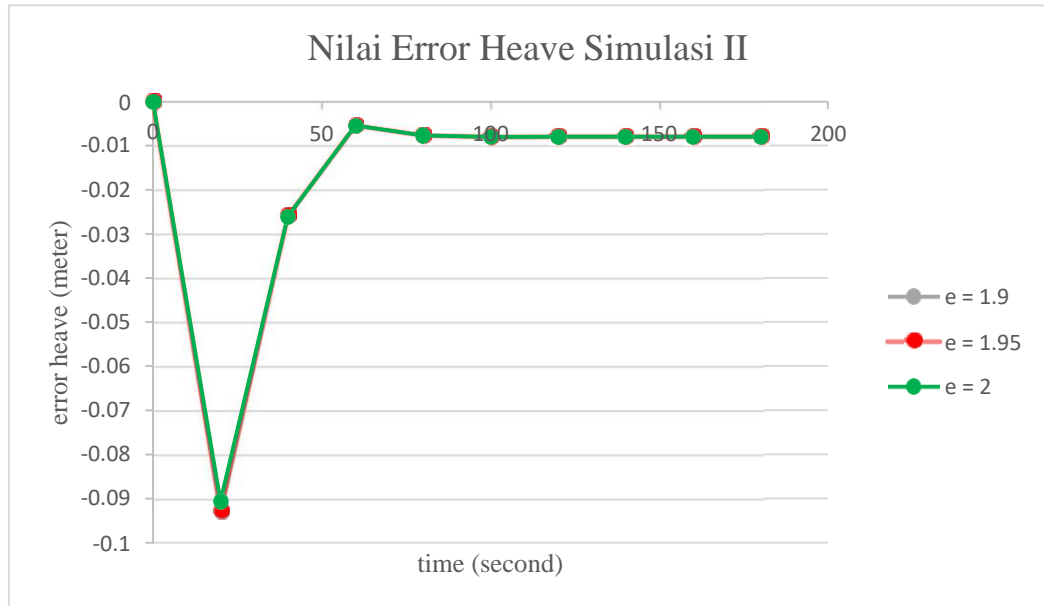
Gambar 4.36 Grafik Perbandingan Hasil *Tracking Path* Simulasi II

Dari gambar 4.36 terlihat bahwa hasil simulasi II dengan menggunakan ketiga variasi *threshold error yaw* sudah dapat mengikuti *desired path* dari lambung kapal. Dengan menggunakan *threshold error yaw* sebesar 1.9° menghasilkan *path tracking* yang paling mendekati *desired path*.



Gambar 4.37 Grafik Perbandingan Nilai RSE Simulasi II

Dari gambar 4.37 dapat dilihat nilai RSE maksimum terdapat pada simulasi dengan *error yaw* sebesar 2° pada detik ke-80 dengan nilai 2.516389 dan untuk nilai RSE minimum terdapat pada simulasi dengan *error yaw* sebesar 2° pada detik ke-180 dengan nilai 0.15128. Untuk hasil selengkapnya dapat dilihat di lampiran B4 sampai B6.



Gambar 4.38 Grafik Perbandingan Nilai *Error Heave* Simulasi I

Pada gambar 4.38 dapat dilihat gambar grafik perbandingan nilai *error heave* simulasi II dengan menggunakan tiga nilai *threshold* yang berbeda. Pada grafik terlihat hampir tidak tampak perbedaan yang besar dari ketiga *threshold* yang berbeda. Nilai *error heave* maksimum terdapat pada simulasi dengan *error yaw* sebesar 1.9° pada detik ke-20 dengan nilai -0.09304212 m dan untuk nilai *error heave* minimum terdapat pada simulasi dengan *error yaw* sebesar 2° pada detik ke-60 dengan nilai -0.00540397 m.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil analisis penelitian tugas akhir, didapatkan kesimpulan seperti berikut :

1. Perancangan kontroller berbasis logika *fuzzy* untuk pengendalian BlueROV 2 telah dirancang dan bekerja dengan performansi yang baik. Sebanyak total tiga buah *fuzzy logic controller* telah dibuat untuk orientasi *surge*, *heave* dan *yaw* dengan parameter sebagai berikut :
 - Untuk kontrol *surge* digunakan dua parameter *input* yaitu *error surge* dengan *range* -1 hingga 1 dan *rate of change of error* dengan *range* -0.5 hingga 0.5 untuk memberikan satu parameter *output* berupa nilai PWM dengan *range* -90 hingga 90.
 - Untuk kontrol *heave* digunakan dua *input* yaitu *error heave* dengan *range* -0.5 hingga 0.5 dan *rate of change of error* dengan *range* -1 hingga 1 untuk memberikan satu *output* berupa nilai PWM dengan *range* 1900 hingga 1100.
 - Untuk kontrol *yaw* digunakan dua *input* yaitu *error yaw* dengan *range* -1 hingga 1 dan *rate of change of error* dengan *range* -0.5 hingga 0.5 untuk memberikan satu *output* berupa nilai PWM dengan *range* -300 hingga 300.
2. Performa dari BlueROV 2 dalam melakukan *path tracking* didapatkan data sebagai berikut :
 - Simulasi 1
 - RSE maksimum 2.67926 m.
 - RSE minimum 0.261849 m.
 - *Error heave* maksimum -0.093138855 m.
 - *Error heave* minimum -0.00542276 m.
 - Simulasi 2
 - RSE maksimum 2.516389 m.
 - RSE minimum 0.15128 m.
 - *Error heave* maksimum -0.09304212 m.
 - *Error heave* minimum -0.00540397 m.

Terdapat nilai *root square error* yang cukup besar dikarenakan orientasi dari *surge* dan *yaw* BlueROV 2 hanya dapat dikendalikan satu saja dalam satu waktu.

5.2 Saran

Saran untuk mengembangkan penelitian selanjutnya dari penulis ada sebagai berikut :

1. Merancang model hidrodinamis spesifik untuk BlueROV 2 dengan konfigurasi enam pendorong untuk menghindari terjadinya *error* saat simulasi.
2. Membuat algoritma untuk optimisasi *tuning gain* kontroller agar mendapatkan hasil yang lebih akurat.
3. Menyempurnakan kontroller dengan menyatukan beberapa kontroller MISO menjadi sebuah kontroller MIMO.

Halaman ini sengaja dikosongkan

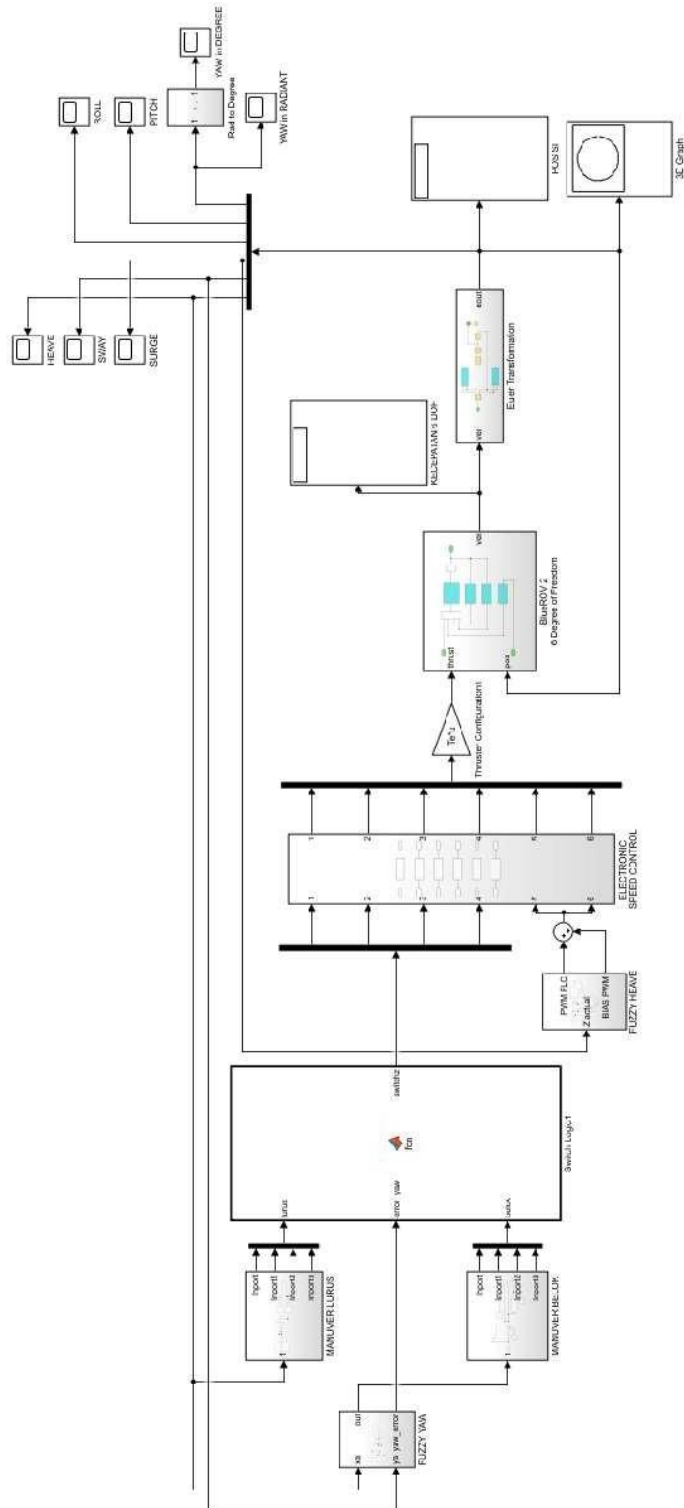
DAFTAR PUSTAKA

- [1] Blue Robotics Inc, “Bluerov2.” <https://www.bluerobotics.com/store/rov/bluerov2/bluerov2/>, Accessed April 2022.
- [2] Blue Robotics Inc, “Bluerov2.” <https://www.bluerobotics.com/store/thrusters/t200-thrusters>, Accessed April 2022.
- [3] C. S. Chin, *Computer-Aided Control Systems Design*. Taylor & Francis Group, 2013.
- [4] D. Ibrahim, “Designing Embedded Systems with 32-Bit PIC Microcontrollers and MikroC,” 1st ed., 2014, p. 371.
- [5] E. M. Einarsson and A. Lipenitis, “Model Predictive Control for the BlueROV2 Theory and Implementation,” 2020, [Online]. Available: <https://www.et.aau.dk/>
- [6] F. Derroncourt, “Introduction to fuzzy logic,” Jan. 2013.
- [7] H. Zakerdoost, H. Ghassemi, and M. Ghiasi, “An evolutionary optimization technique applied to resistance reduction of the ship hull form,” *J. Nav. Archit. Mar. Eng.*, vol. 10, no. 1, pp. 1–12, 2013, doi: 10.3329/jname.v10i1.12927.
- [8] J. González-García *et al.*, “Model-free high order sliding mode control with finite-time tracking for unmanned underwater vehicles,” *Appl. Sci.*, vol. 11, no. 4, pp. 1–22, 2021, doi: 10.3390/app11041836.
- [9] J. W. Chen, H. Zhu, L. Zhang, and Y. Sun, “Research on fuzzy control of path tracking for underwater vehicle based on genetic algorithm optimization,” *Ocean Eng.*, vol. 156, no. February, pp. 217–223, 2018, doi: 10.1016/j.oceaneng.2018.03.010.
- [10] Kementerian Kelautan dan Perikanan, KKP | Kementerian Kelautan dan Perikanan 2019, Accessed April 2022.
- [11] M. S. M. Aras *et al.*, “Fuzzy logic controller for depth control of underwater remotely operated vehicle,” *J. Theor. Appl. Inf. Technol.*, vol. 91, no. 2, pp. 275–288, 2016.
- [12] P. A. DeBitetto, “Fuzzy logic for depth control of Unmanned Undersea Vehicles,” in *IEEE Journal of Oceanic Engineering*, vol. 20, no. 3, pp. 242–248, July 1995, doi: 10.1109/48.393079.
- [13] S. Prabhakar and B. Buckham, “Dynamics modeling and control of a variable length remotely operated vehicle tether,” *Proceedings of OCEANS 2005 MTS/IEEE*, 2005, pp. 1255–1262 Vol. 2, doi: 10.1109/OCEANS.2005.1639927.
- [14] S. Tri, “Rancang Bangun Metode Tracking Object Pada Underwater Remotely Operated Vehicle,” Institut Teknologi Sepuluh Nopember, 2019.

- [15] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*. 2011. doi: 10.1002/9781119994138.
- [16] T. I. Fossen, *Marine Control Systems: Guidance, Navigation and Control of Ships, Rigs and Underwater Vehicles*. Marine Cybernetics, Trondheim, Norway, 2002
- [17] C. Wu, *6-DoF Modelling and Control of a Remotely Operated Vehicle*. Master's Thesis, Flinders University, Adelaide, Australia, 2018.
- [18] W. M. Bessa, M. . Dutra, and E. Kreuzer, "Thruster dynamics compensation for the positioning of underwater robotic vehicles through a fuzzy sliding mode based approach," *ABCM Symp. Mechatronics*, vol. 2, no. 1990, pp. 605–612, 2006.

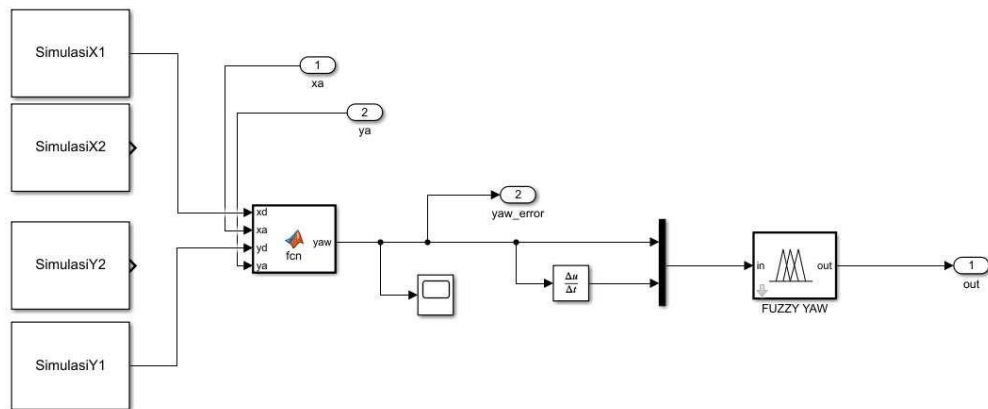
LAMPIRAN A

SIMULINK SISTEM KONTROL FUZZY LOGIC BLUEROV 2

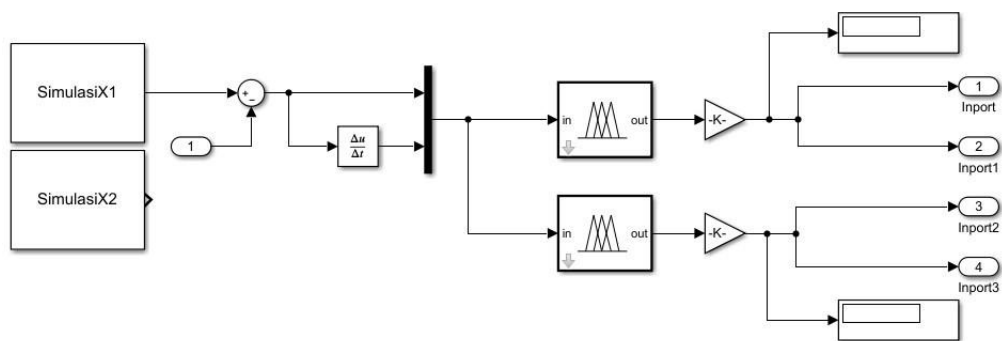


Subsystem Simulink :

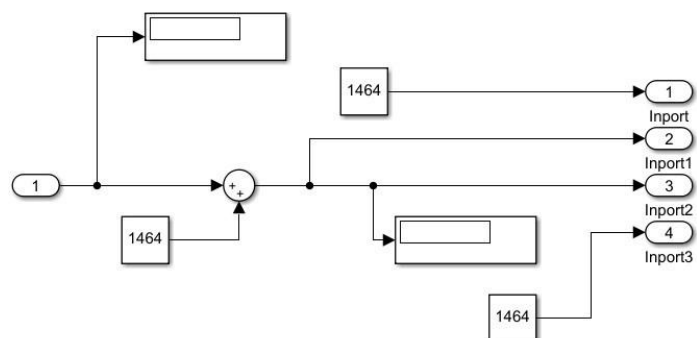
1. Fuzzy Yaw



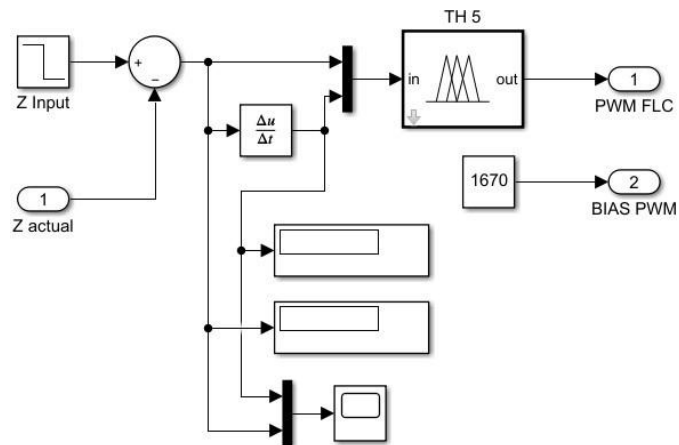
2. Manuver Lurus



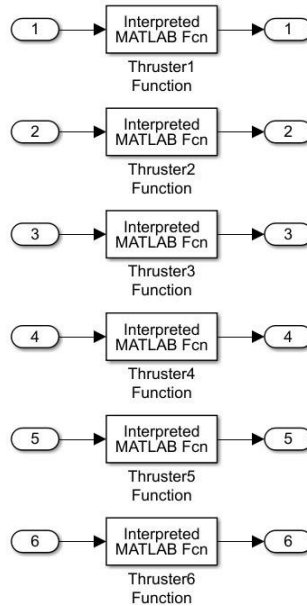
3. Manuver Belok



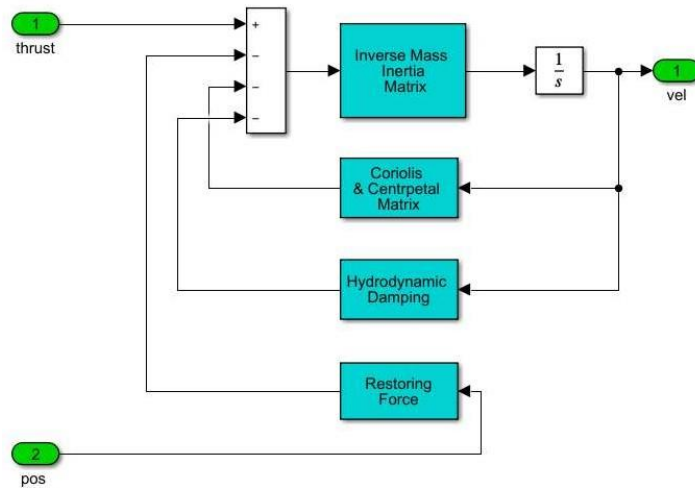
4. Fuzzy Heave



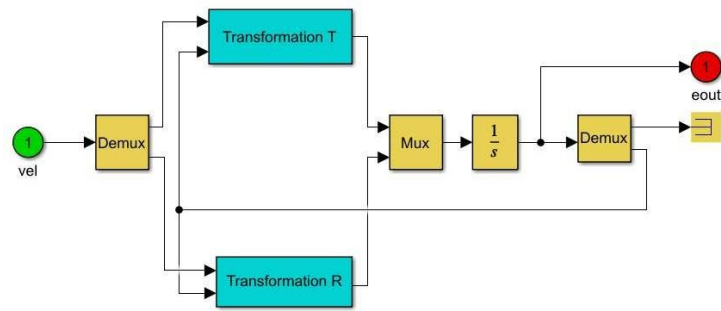
5. Electronic Speed Control



6. BlueROV 2 6 Degree of Freedom



7. Euler Transformation



LAMPIRAN B
DATA HASIL SIMULASI

Tabel B1. Perbandingan trayektori dan kedalaman BlueROV 2 dengan $e_{yaw} = 1.15^\circ$
dan *starting point* $x = 0, y = 0$

<i>Error yaw = 1.15°</i>								
t(s)	X _d (m)	Y _d (m)	Z _d (m)	X _a (m)	Y _a (m)	Z _a (m)	RSE	<i>error heave</i> (m)
0	0	0	0	0	0	0	0	0
20	1.354117	-0.62548	-1	1.117288	-0.09623	-0.90686	0.579824	-0.09313885
40	2.783813	-1.45819	-1	2.347244	-0.48237	-0.97411	1.069025	-0.02589137
60	3.931698	-2.2899	-1	3.386658	-1.07203	-0.99458	1.334273	-0.00542276
80	6.194098	-4.5523	-1	5.163438	-2.36871	-0.99235	2.414609	-0.0076472
100	6.609955	-5.12624	-1	5.545243	-2.66762	-0.99193	2.67926	-0.00806521
120	7.025813	-5.70019	-1	6.120866	-3.68414	-0.99202	2.209839	-0.00797897
140	7.858517	-7.12988	-1	7.144113	-5.6661	-0.99203	1.628815	-0.00797258
160	8.484	-8.484	-1	7.81615	-6.99626	-0.99202	1.63076	-0.00797536
180	8.484	-8.484	-1	8.499229	-8.86586	-0.99202	0.382165	-0.00797537

Tabel B2. Perbandingan trayektori dan kedalaman BlueROV 2 dengan $e_{yaw} = 1.2^\circ$
dan *starting point* $x = 0, y = 0$

<i>Error yaw = 1.2°</i>								
t(s)	X _d (m)	Y _d (m)	Z _d (m)	X _a (m)	Y _a (m)	Z _a (m)	RSE	<i>error heave</i> (m)
0	0	0	0	0	0	0	0	0
20	1.354117	-0.62548	-1	1.148326	-0.09657	-0.90695	0.567541	-0.09305
40	2.783813	-1.45819	-1	2.405739	-0.48528	-0.9741	1.043782	-0.0259
60	3.931698	-2.2899	-1	3.459929	-1.07639	-0.99458	1.301989	-0.00542
80	6.194098	-4.5523	-1	5.214403	-2.34079	-0.99235	2.418797	-0.00765
100	6.609955	-5.12624	-1	5.678426	-2.69426	-0.99193	2.604283	-0.00807
120	7.025813	-5.70019	-1	6.168196	-3.49488	-0.99202	2.366193	-0.00798
140	7.858517	-7.12988	-1	7.141749	-5.44419	-0.99203	1.83175	-0.00797
160	8.484	-8.484	-1	7.94361	-7.13396	-0.99202	1.454176	-0.00798
180	8.484	-8.484	-1	8.544336	-8.7388	-0.99202	0.261849	-0.00798

Tabel B3. Perbandingan trayektori dan kedalaman BlueROV 2 dengan $e_{yaw} = 1.3^\circ$
dan *starting point* $x = 0, y = 0$

<i>Error yaw = 1.3°</i>								
t(s)	X _d (m)	Y _d (m)	Z _d (m)	X _a (m)	Y _a (m)	Z _a (m)	RSE	<i>error heave</i> (m)
0	0	0	0	0	0	0	0	0
20	1.354117	-0.62548	-1	1.207582	-0.09619	-0.90696	0.549204	-0.09304212
40	2.783813	-1.45819	-1	2.515176	-0.49028	-0.97411	1.004496	-0.02589181
60	3.931698	-2.2899	-1	3.598502	-1.08981	-0.99457	1.245485	-0.00543035

80	6.194098	-4.5523	-1	5.363638	-2.35788	-0.99235	2.346306	-0.00764625
100	6.609955	-5.12624	-1	5.971573	-2.81396	-0.99193	2.398793	-0.00806533
120	7.025813	-5.70019	-1	6.411827	-3.48808	-0.99202	2.295737	-0.00797898
140	7.858517	-7.12988	-1	7.35246	-5.38845	-0.99203	1.813473	-0.00797261
160	8.484	-8.484	-1	8.117006	-7.16213	-0.99202	1.371873	-0.00797535
180	8.484	-8.484	-1	8.228861	-7.48254	-0.99202	1.033449	-0.00797537

Tabel B4. Perbandingan trayektori dan kedalaman BlueROV 2 dengan $e_{yaw} = 1.9^\circ$
dan *starting point* $x = 8.484, y = -8.484$

<i>Error yaw = 1.9°</i>								
t(s)	X _d (m)	Y _d (m)	Z _d (m)	X _a (m)	Y _a (m)	Z _a (m)	RSE	<i>error heave</i> (m)
0	8.484	-8.484	0	8.484	-8.484	0	0	0
20	7.129883	-7.85852	-1	7.320358	-8.37899	-0.90696	0.554234	-0.09304
40	5.700188	-7.02581	-1	6.03668	-8.01003	-0.97411	1.040151	-0.02589
60	4.552302	-6.1941	-1	4.982049	-7.45149	-0.99457	1.328806	-0.00543
80	3.35825	-5.12575	-1	3.841882	-6.54027	-0.99235	1.494912	-0.00765
100	2.289902	-3.9317	-1	2.788939	-5.39082	-0.99193	1.542101	-0.00807
120	1.458188	-2.78381	-1	1.906834	-4.09699	-0.99202	1.3877	-0.00798
140	0.625483	-1.35412	-1	0.986737	-2.41131	-0.99203	1.11721	-0.00797
160	0	0	-1	0.263411	-0.77009	-0.99202	0.81389	-0.00798
180	0	0	-1	0.140264	-0.44604	-0.99202	0.467576	-0.00798

Tabel B5. Perbandingan trayektori dan kedalaman BlueROV 2 dengan $e_{yaw} = 1.95^\circ$
dan *starting point* $x = 8.484, y = -8.484$

<i>Error yaw = 1.95°</i>								
t(s)	X _d (m)	Y _d (m)	Z _d (m)	X _a (m)	Y _a (m)	Z _a (m)	RSE	<i>error heave</i> (m)
0	8.484	-8.484	0	8.484	-8.484	0	0	0
20	7.129883	-7.85852	-1	7.349168	-8.38154	-0.90717	0.56713	-0.09283
40	5.700188	-7.02581	-1	6.093412	-8.01337	-0.97402	1.062964	-0.02598
60	4.552302	-6.1941	-1	5.057207	-7.46005	-0.99458	1.362923	-0.00542
80	3.35825	-5.12575	-1	3.930267	-6.56122	-0.99235	1.545244	-0.00765
100	2.289902	-3.9317	-1	2.892422	-5.44347	-0.99193	1.627413	-0.00807
120	1.458188	-2.78381	-1	2.026603	-4.21022	-0.99202	1.535495	-0.00798
140	0.625483	-1.35412	-1	1.150763	-2.66911	-0.99203	1.416028	-0.00797
160	0	0	-1	0.594078	-1.38584	-0.99202	1.507809	-0.00798
180	0	0	-1	-0.02345	-0.19550	-0.99202	0.196904	-0.00798

Tabel B6. Perbandingan trayektori dan kedalaman BlueROV 2 dengan $e_{yaw} = 2^\circ$
 dan *starting point* $x = 8.484, y = -8.484$

<i>Error yaw = 2°</i>								
t(s)	X _d (m)	Y _d (m)	Z _d (m)	X _a (m)	Y _a (m)	Z _a (m)	RSE	<i>error heave</i> (m)
0	8.484	-8.484	0	8.484	-8.484	0	0	0
20	7.129883	-7.85852	-1	7.375353	-8.48117	-0.90941	0.66929	-0.09059
40	5.700188	-7.02581	-1	6.315826	-8.36848	-0.97386	1.477084	-0.02614
60	4.552302	-6.1941	-1	5.412857	-8.0778	-0.9946	2.070962	-0.0054
80	3.35825	-5.12575	-1	4.405137	-7.41403	-0.99234	2.516389	-0.00766
100	2.289902	-3.9317	-1	3.300394	-6.14045	-0.99194	2.428923	-0.00806
120	1.458188	-2.78381	-1	1.879541	-3.60405	-0.99202	0.922133	-0.00798
140	0.625483	-1.35412	-1	1.107885	-2.11618	-0.99203	0.901914	-0.00797
160	0	0	-1	0.405522	-0.76088	-0.99202	0.862198	-0.00798
180	0	0	-1	0.099357	-0.15128	-0.99202	0.15128	-0.00798

Halaman ini sengaja dikosongkan

LAMPIRAN C
CODE TO GENERATE HULL SHIP

```
clear all;
clc;
B = 2.5; % B value
L = 12; % L value
T = 3; % T value

%x vector
x = -6:1:6;
%z vector
z=0:-0.5:-0.5;

%meshgrid init
[X,Z]=meshgrid(x,z);
%y function declaration
Y=B/2*(1-(2*X/L).^2).*(1-(Z/T).^2);
C=[0 0 0];
mesh(X,Y,Z)
xlabel('x'); ylabel('y'); zlabel('z')
axis([-9 9 -3 3 -5 0])
hold on

Y=-B/2*(1-(2*X/L).^2).*(1-(Z/T).^2);
mesh(X,Y,Z)
colormap(C)

%end of code
```

Halaman ini sengaja dikosongkan

BIODATA PENULIS



Penulis dilahirkan di Sidoarjo, 4 Agustus 2000, merupakan anak pertama dari 3 bersaudara. Penulis telah menempuh pendidikan formal yaitu di TK Pelita Hati Lampung, SDIT Labs School Kaizen Bogor, SMPN 12 Bekasi dan SMAN 5 Bekasi. Setelah lulus dari sekolah menengah atas, penulis melanjutkan pendidikan di Jurusan Teknik Mesin ITS. Di Departemen Teknik Mesin Penulis aktif di beberapa kegiatan seminar yang diselenggarakan oleh Departemen, Media Komunikasi Teknik Mesin ITS (DIMENSI), aktif sebagai Asisten Praktikum Mekanika Fluida I dan II, dan mengikuti program Merdeka Belajar Kampus Merdeka – BANGKIT.