



TESIS - TE142599

SISTEM KENDALI *UNDERWATER REMOTELY OPERATED VEHICLES* PADA TINGKAT KEDALAMAN AIR MENGGUNAKAN METODE KENDALI ADAPTIF

ALI ZAINAL ABIDIN
NRP 07111450040010

DOSEN PEMBIMBING
Ronny Mardiyanto, S.T., M.T., Ph.D
Ir. Djoko Purwanto, M. Eng., Ph.D

PROGRAM MAGISTER
BIDANG KEAHLIAN TEKNIK ELEKTRONIKA
DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI ELEKTRO
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2018



TESIS - TE142599

SISTEM KENDALI *UNDERWATER REMOTELY OPERATED VEHICLES* PADA TINGKAT KEDALAMAN AIR MENGGUNAKAN METODE KENDALI ADAPTIF

**ALI ZAINAL ABIDIN
NRP 07111450040010**

**DOSEN PEMBIMBING
Ronny Mardiyanto, S.T., M.T., Ph.D
Ir. Djoko Purwanto, M. Eng., Ph.D**

**PROGRAM MAGISTER
BIDANG KEAHLIAN TEKNIK ELEKTRONIKA
DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI ELEKTRO
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2018**

LEMBAR PENGESAHAN

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Teknik (M.T.)

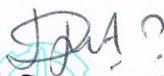
di
Institut Teknologi Sepuluh Nopember
oleh:

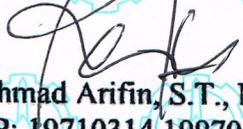
Ali Zainal Abidin
NRP. 07111450040010

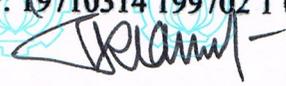
Tanggal Ujian : 03 Juli 2018
Periode Wisuda : September 2018

Disetujui oleh:


1. Ronny Mardiyanto, S.T., M.T., Ph.D (Pembimbing I)
NIP: 19810118 200312 1 003


2. Ir. Djoko Purwanto, M.Eng., Ph.D (Pembimbing II)
NIP: 19651211 199002 1 002


3. Achmad Arifin, S.T., M.Eng., Ph.D (Penguji)
NIP: 19710314 199702 1 001


4. Dr. Tri Arief Sardjono, S.T., M.T. (Penguji)
NIP: 19700212 199512 1 001


5. Dr. Ir. Hendra Kusuma, M.Eng.Sc (Penguji)
NIP: 19640902 198903 1 003


6. Astria Nur Infansyah, S.T., M.Eng., Ph.D. (Penguji)
NIP: 19810325 201012 1 002

Dekan Fakultas Teknologi Elektro


Dr. Tri Arief Sardjono, S.T., M.T.
NIP: 19700212 199512 1 001

Halaman ini sengaja dikosongkan

PERNYATAAN KEASLIAN TESIS

Dengan ini saya menyatakan bahwa isi keseluruhan Tesis saya dengan judul “**Sistem Kendali *Underwater Remotely Operated Vehicles* Pada Tingkat Kedalaman Air Menggunakan Metode Kendali Adaptif**” adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 03 Juli 2018



Ali Zainal Abidin

NRP. 07111450040010

Halaman ini sengaja dikosongkan

SISTEM KENDALI UNDERWATER REMOTELY OPERATED VEHICLES PADA TINGKAT KEDALAMAN AIR MENGGUNAKAN METODE KENDALI ADAPTIF

Nama mahasiswa : Ali Zainal Abidin
NRP : 07111450040010
Pembimbing : 1. Ronny Mardiyanto, S.T., M.T., Ph.D
2. Ir. Djoko Purwanto, M. Eng., Ph.D

ABSTRAK

Pada tingkat kedalaman air tertentu, kecepatan serta arah arus air dapat berubah tidak teratur dan memiliki tekanan yang berbeda. Faktor tersebut merupakan parameter yang mempengaruhi stabilisasi *underwater ROV (remotely operated vehicle)* pada saat di dalam air. *Underwater ROV* harus dapat bertahan (*hold position*) pada tingkat kedalaman air dan terhadap gangguan (perubahan arus air) yang ada di dalam air. Dalam menghadapi masalah tersebut, digunakan sistem kendali adaptif karena memiliki parameter kendali yang dapat beradaptasi terhadap gangguan (*disturbance*).

Pada penelitian ini, sistem kendali adaptif dirancang menggunakan neural network dan sistem kendali PID (*Proportional Integral Derivative*). Sistem kendali PID digunakan sebagai kontrol gerak *underwater ROV*, sedangkan neural network difungsikan sebagai identifikasi terhadap gangguan yang sudah dikenali (*learning*) terlebih dahulu, hasil identifikasi digunakan untuk menentukan nilai parameter dari sistem kendali PID.

Neural network terdiri dari 3 layer (*input, hidden, dan output*), *input layer* memiliki 3 neuron (*error, integral error dan derivative error*), *hidden layer* memiliki 8 neuron dan *output layer* memiliki 3 neuron (K_p , K_i dan K_d). Komputansi sistem kendali NN berbasis mikrokontroler arduino Nano 328. Pengujian dilakukan dengan memberi arus air buatan yang berasal dari Bilge Pump 1100GPH, menghasilkan gaya dorong arus atas dan bawah sebagai gangguan kepada *underwater ROV* dengan besar nilai yang bervariasi.

Pada penelitian dilakukan pengujian terhadap sistem kendali adaptif NN-PID. Pada gangguan atas memiliki nilai *overshoot* 20%, *rise time* 0,8 detik, *settling time* 2,4 detik dan *error steady state* (ESS) 6,50%. Pada gangguan bawah, *overshoot* 20%, *rise time* 2,0 detik, *settling time* 4,6 detik dan ESS 6,45%. Hasil pengujian *underwater ROV* yang dilakukan terhadap gangguan yang berubah-ubah menunjukkan sistem kendali NN-PID dapat beradaptasi dengan baik.

Kata kunci: *hold position*, NN-PID, neural network, *underwater ROV (remotely operated vehicle)*.

Halaman ini sengaja dikosongkan

CONTROL SYSTEM UNDERWATER REMOTELY OPERATED VEHICLES FOR DEPTH WATER LEVEL USING ADAPTIVE CONTROL METHODS

Student Name : Ali Zainal Abidin
Student Identity Number : 07111450040010
Supervisors : 1. Ronny Mardiyanto, S.T., M.T., Ph.D
2. Ir. Djoko Purwanto, M. Eng., Ph.D

ABSTRACT

At certain water depth levels, the velocity and direction of the water current may change irregularly and have different pressure. These factors are parameters that affect the navigation and stabilization of underwater ROV (remotely operated vehicle) in the water. Underwater ROV must be able to maintain a position (hold position) at the level of water depth and the disturbance (changes in water flow) in the water. In the face of the problem, an adaptive control system is used because it has a control parameter that can adapt to disturbance.

In this research, built NN-PID adaptive control system of two control system, that is between PID control system (Propotional Integral Derivative) and Neural Network control system. PID control functions as control over underwater motion of ROV, while Neural Network control is used as identification of previously recognized disturbances and the output value of the system is used as a tuning parameter of the PID control system.

The Nueral Network system consists of 3 layers (input, hidden, and output), the input layer has 3 neurons (error, integral error and derivative error), the hidden layer has 8 neurons and the output layer has 3 neurons (Kp, Ki and Kd). Computing NN control system based on arduino Nano 328 microcontroller. The test is carried out by giving artificial water flow from the Bilge Pump 1100GPH, resulting in an upper and lower current thrust as a disturbance to the underwater ROV with varying values.

In the experiment, we tested the NN-PID adaptive control system. At the top disturbance has a 20% overshoot value, 0.8 second rise time, 2,4 seconds settling time and error steady state (ESS) of 6,50%. On the downside, overshoot 20%, rise time 2,0 seconds, settling time 4,6 seconds and ESS 6,45%. The results of underwater ROV testing performed against the mutable disturbances indicate the NN-PID control system can adapt well.

Keywords: hold position, NN-PID, neural network, underwater ROV (remotely operated vechile).

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Alhamdulillah, puji dan syukur penulis panjatkan kepada Allah SWT, karena atas segala petunjuk dan nikmat-Nya lah tesis ini dapat diselesaikan, sholawat dan salam kepada junjungan agung nabi Muhammad SAW serta keluarganya yang suci.

Tesis berjudul “Sistem Kendali *Underwater Remotely Operated Vehicles* Pada Tingkat Kedalaman Air Menggunakan Metode Kendali Adaptif” ini disusun untuk memenuhi sebagian persyaratan memperoleh gelar Magister Teknik di Departemen Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember.

Penulis menyadari bahwa dalam penyusunan tesis ini tidak terlepas dari bantuan berbagai pihak. Oleh karena itu, dengan ketulusan dan kerendahan hati penulis menyampaikan terima kasih kepada:

1. Kedua Orang Tua Tercinta dan saudari-saudari penulis, atas segala dukungan dan doanya hingga terselesaikannya tesis ini.
2. Bapak Ronny Mardiyanto S.T., M.T., Ph.D. selaku Dosen Pembimbing, yang telah banyak memberikan saran, bantuan, serta sabar dalam membimbing penulis.
3. Bapak Ir. Djoko Purwanto, M.Eng., Ph.D. selaku Dosen Pembimbing, yang telah banyak memberikan saran, bantuan, serta sabar dalam membimbing penulis.
4. Tri Susanto sebagai teman seperjuangan dalam menyelesaikan tesis ini, dan telah banyak membantu di dalam prosesnya.
5. Rahmat Shaleh, Dhadang S.B.W. selaku teman yang selalu memberikan semangat serta bantuan yang luar biasa kepada penulis.
6. Ali Uraidhi selaku teman yang telah banyak memberikan masukan dan bantuan kepada penulis.
7. Teman-Teman di Cometronica, Jefri Hidayat, Beta Firmansyah, Victor Chandra, dan Husein Shahab.
8. Teman-Teman di Sekolah Robot Indonesia.

Penulis menyadari bahwa Tugas Akhir ini masih belum sempurna dan masih banyak hal yang perlu diperbaiki. Saran, kritik dan masukan baik dari semua pihak sangat membantu penulis terutama untuk berbagai kemungkinan pengembangan lebih lanjut. Penulis berharap semoga tesis ini dapat bermanfaat bagi pengembangan ilmu pengetahuan dan teknologi serta bagi masyarakat.

Surabaya, 03 Juli 2018

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN.....	iii
PERNYATAAN KEASLIAN TESIS.....	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xv
DAFTAR TABEL	xvii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan.....	3
1.4 Batasan Masalah	3
1.5 Kontribusi	4
BAB 2 KAJIAN PUSTAKA	5
2.1 Daya Apung	7
2.1.1 Benda Dalam Hukum Archimedes	8
2.1.2 Tekanan Hidrostatik.....	11
2.1.3 Tekanan dan Kedalaman Air	12
2.2 Sensor	13
2.2.1 Sensor IMU (<i>Inertial Measurement Unit</i>)	13
2.2.2 <i>Gyroscope</i>	14
2.2.3 <i>Accelerometer</i>	15
2.2.4 Sensor Tekanan Air	17
2.2.5 Sensor IMU MPU6050	21
2.3 Motor DC.....	22
2.4 Sistem Kendali	23
2.4.1 Sistem Kendali Adaptif.....	24
2.4.2 Sistem Kontrol PID.....	25
2.4.3 Jaringan Syaraf Tiruan (JST)	27
2.4.4 Sistem Kendali Adaptif PID.....	31

2.5	<i>Fishbone</i> Diagram Penelitian.....	32
BAB 3 METODOLOGI PENELITIAN.....		35
3.1	Studi Pustaka dan Literatur.....	36
3.2	Rancang Bangun <i>Underwater ROV</i>	37
3.2.1	Rancang Bangun Mekanik.....	38
3.2.2	Penggunaan Sensor IMU MPU6050.....	39
3.2.3	Perancangan Posisi Motor DC.....	40
3.2.4	Rancang Bangun <i>Graphical User Interface (GUI)</i>	41
3.3	Rancang Bangun Gerak <i>Underwater ROV</i>	42
3.4	Perancangan Sistem Kendali <i>Hold Position</i>	45
3.5	Pengujian Hasil.....	49
BAB 4 HASIL DAN PEMBAHASAN.....		51
4.1	Keterbatasan Penelitian.....	52
4.2	Hasil Perancangan <i>Underwater ROV</i>	52
4.3	Pengujian GUI (<i>Graphical User Interface</i>).....	53
4.4	Pengujian Sistem <i>Neural Network</i>	54
4.5	Pengujian Sistem Kendali <i>Hold Position</i>	56
4.5.1	Pengujian Sistem Kendali PID.....	57
4.5.2	Pengujian Sistem Kendali NN-PID.....	59
BAB 5 PENUTUP.....		67
5.1	Kesimpulan.....	67
5.2	Saran.....	68
DAFTAR PUSTAKA.....		69
LAMPIRAN.....		71
RIWAYAT PENULIS.....		85

DAFTAR GAMBAR

Gambar 2.1 Dasar Sistem <i>Underwater</i> ROV [4].....	5
Gambar 2.2 Daya Apung Benda.....	8
Gambar 2.3 Benda Tenggelam.....	9
Gambar 2.4 Benda Melayang.....	10
Gambar 2.5 Benda Terapung.....	11
Gambar 2.6 Sensor IMU ETHOS (ETH <i>Orientation Sensor</i>).....	14
Gambar 2.7 Sistem Mekanika Pada <i>Gyroscope</i>	15
Gambar 2.8 Struktur <i>Accelerometer</i> [10].....	17
Gambar 2.9 Diagram skematik dari <i>absolute pressure sensor</i> [12].....	18
Gambar 2.10 Rangkaian Jembatan <i>wheatstone</i> [13].....	18
Gambar 2.11 Sensor Tekanan MS5803-14BA [14].....	20
Gambar 2.12 Skematik Sensor MS5803-14BA [14].....	20
Gambar 2.13 Pemasangan Sensor Tekanan MS5803-14BA.....	21
Gambar 2.14 Sensor IMU MPU6050.....	21
Gambar 2.15 Motor DC <i>Bilge Pump</i>	23
Gambar 2.16 Sistem Kendali <i>Closed-loop</i> [17].....	24
Gambar 2.17 Diagram Blok Sistem Kendali Adaptif [18].....	25
Gambar 2.18 Diagram Blok Sistem Kontrol PID.....	26
Gambar 2.19 Diagram Blok Kontroler Standar PID.....	26
Gambar 2.20 Diagram sederhana <i>Neural Network</i>	28
Gambar 2.21 Ilustrasi fungsi sigmoid biner dengan range (0,1).....	30
Gambar 2.22 Ilustrasi fungsi sigmoid bipolar dengan range (-1,1).....	31
Gambar 2.23 Diagram Blok Sistem Kendali Adaptif PID.....	32
Gambar 2.24 <i>Fishbone</i> Diagram Penelitian.....	33
Gambar 3.1 Tahapan Penelitian.....	35
Gambar 3.2 Ilustrasi Studi Pustaka dan Literatur.....	36
Gambar 3.3 Blok Rancang Bangun <i>Underwater</i> ROV.....	37
Gambar 3.4 Rancang Bangun Mekanik <i>Underwater</i> ROV.....	38
Gambar 3.5 Posisi Peletakan Sensor IMU MPU6050.....	39
Gambar 3.6 Posisi Peletakan Motor DC Pada <i>Underwater</i> ROV.....	40
Gambar 3.7 Rancang Bangun Desain <i>Graphical User Interface</i> (GUI).....	41
Gambar 3.8 Arah Gerak <i>Underwater</i> ROV.....	42
Gambar 3.9 Arah Putar Motor DC Untuk Gerak Maju dan Mundur.....	43
Gambar 3.10 Arah Putar Motor DC Untuk Gerak Putar Kanan dan Kiri.....	44
Gambar 3.11 Arah Putar Motor DC Untuk Gerak Atas dan Bawah.....	44
Gambar 3.12 Diagram Blok Sistem <i>Hold Position Underwater</i> ROV.....	45
Gambar 3.13 Diagram Blok Kendali Adaptif NN-PID.....	46
Gambar 3.14 Diagram Blok <i>Neural Network</i>	47
Gambar 3.15 Diagram Blok Kendali Gerak <i>Underwater</i> ROV.....	48
Gambar 3.16 Ilustrasi Sistem <i>Hold Position Robot Underwater</i> ROV.....	48
Gambar 3.17 Sinyal Tanggapan Sistem Kendali.....	49

Gambar 4.1 Alat Uji Pembuat Gangguan (<i>disturbance</i>).....	51
Gambar 4.2 Hasil Perakitan Robot <i>Underwater ROV</i>	52
Gambar 4.3 GUI (<i>Graphical User Interface</i>) Comet- <i>Underwater ROV</i>	53
Gambar 4.4 GUI (<i>Graphical User Interface</i>) Menampilkan data grafik.....	54
Gambar 4.5 Pengujian <i>Neural Network</i> dengan target <i>error</i> 0.02.....	55
Gambar 4.6 Pengujian <i>Neural Network</i> dengan target <i>error</i> 0.05.....	55
Gambar 4.7 Pengujian Kendali PID Pada Kondisi Ideal	58
Gambar 4.8 Pengujian <i>Underwater ROV</i> Terhadap Gangguan Arus Atas	60
Gambar 4.9 Pengujian Terhadap Gangguan Atas.....	61
Gambar 4.10 Pengujian <i>Underwater ROV</i> Terhadap Gangguan Arus Bawah.....	62
Gambar 4.11 Pengujian Terhadap Gangguan Bawah	63
Gambar 4.12 Hasil Pengujian NN-PID Pada Pattern ke-5.....	64
Gambar 4.13 Hasil Pengujian NN-PID Pada Pattern ke-6.....	65

DAFTAR TABEL

Tabel 4.1 Input dan Output <i>Neural Network</i> dengan Nilai Target <i>error</i> 0.02	56
Tabel 4.2 Hasil Pengujian Kendali PID Konvensional.....	57
Tabel 4.3 Parameter Keluaran PID Terhadap Gangguan Atas.....	59
Tabel 4.4 Parameter Keluaran PID Terhadap Gangguan Bawah	59
Tabel 4.5 Hasil Pengujian NN-PID Terhadap Gangguan Atas	60
Tabel 4.6 Hasil Pengujian NN-PID Terhadap Gangguan Bawah	62

Halaman ini sengaja dikosongkan

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Sebagian besar dari luas bumi merupakan perairan khususnya Indonesia, secara geografis merupakan negara kepulauan yang memiliki daerah perairan lebih luas dibandingkan luas wilayah daratannya. Kegunaan teknologi sangat diperlukan untuk memberikan peranan dalam mengatasi pelbagai masalah yang ada di daerah perairan Indonesia. Metode konvensional biasanya dilakukan dengan bantuan para penyelam, sedang menggunakan teknologi digantikan oleh robot bawah laut yang dikendalikan oleh manusia.

Dalam kehidupan manusia, kecelakaan ataupun bencana alam sering terjadi di antaranya pada daerah perairan. Seringnya melakukan aktivitas yang berhubungan ataupun melintas di atas perairan, khususnya di Indonesia memiliki angka kecelakaan yang cukup tinggi.

Jalur lintas pelayaran dan jalur penerbangan merupakan aktivitas yang paling sering melintas di atas perairan, terlebih pada penerbangan dengan rute domestik di Indonesia. Data menunjukkan dari tahun 1967-2014 banyak terjadinya kecelakaan pesawat penumpang yang menyebabkan banyak korban yang terjadi di perairan dan sulit ditemukan [1].

Beberapa di antaranya pada tanggal 19 Desember 1997, SilkAir dengan kode penerbangan MI185 jurusan Jakarta - Singapura jatuh di sungai Musi Palembang, menewaskan 104 penumpang dan awak pesawat. Pada 1 Januari 2007, Adam air dengan kode penerbangan KI574 jurusan Surabaya–Manado jatuh di selat Makasar di kedalaman lebih dari 2.000 meter, seluruh 102 penumpang dan awak pesawat tewas. Pada 28 Desember 2014, Air Asia Indonesia dengan kode penerbangan QZ8501 tujuan Singapura jatuh di dasar laut Selat Karimata dekat Laut Jawa menewaskan 155 penumpang dan 7 kru pesawat, hingga hari ke-51 proses pencarian ditemukan 102 jenazah, potongan badan pesawat berhasil ditemukan oleh *underwater ROV (remotely operated vehicles)* pada tanggal 14 Januari 2015 dari RSS MV SWIFT *Rescue* kapal Singapura yang ikut dalam pencarian [1].

Pada proses evakuasi serta penyelamatan pada kecelakaan ataupun bencana alam tersebut, umumnya dilakukan dengan mencari langsung korban oleh penyelam atau penyelamat. Namun hal ini memiliki risiko yang cukup tinggi serta sering mengalami kesulitan dan hambatan untuk melakukan proses tersebut dikarenakan kondisi alam di bawah air yang cukup sulit diperkirakan kondisinya.

Perubahan arah serta kecepatan arus dalam air yang tidak teratur merupakan faktor yang sangat berpengaruh untuk melakukan proses evakuasi serta penyelamatan, di mana penyelam sulit untuk melaksanakan proses tersebut. Oleh karena itu *underwater ROV (Remotely Operated Vehicles)* dapat berperan untuk membantu hingga menggantikan penyelam selama proses itu dilakukan.

Pada penelitian ini, *underwater ROV* yang akan diaplikasikan mampu mempertahankan posisinya (*hold position*) pada kondisi atau titik di kedalaman air yang di inginkan secara vertikal (sumbu *z*) pada saat proses evakuasi serta penyelamatan, sesuai dengan acuan arah (*heading*) serta posisi *underwater ROV* dan level di ke dalam air (*depth*) terhadap perubahan arah serta kecepatan arus berdasarkan tingkat ke dalam air.

Dalam mempertahankan posisi di kedalaman air, sistem kendali adaptif diterapkan pada penelitian ini dan berfungsi sebagai pengendali gerak *underwater ROV*. Diharapkan dengan menggunakan sistem ini dapat membantu *underwater ROV* menstabilkan posisi pada saat melakukan evakuasi atau bermanuver.

Penelitian dan riset tentang *underwater ROV* dalam mempertahankan posisinya (*hold position*) juga pernah dilakukan sebelumnya, di antaranya pada penelitian [2] dengan membuat desain kontroler AUV (*Autonomous Underwater Vehicle*) agar dapat mempertahankan posisi pada level kedalaman air tertentu, peneliti mengusulkan suatu metode dengan mengindikasikan parameter dari sebuah model *close loop depth maneuver*. Adapun penelitian [3] membandingkan dengan eksperimen antara hukum kendali *Propotional Integral and Derivative (PID)* dan kendali *adative nonlinier state feedback* yang diterapkan pada *AC-ROV underwater vehicle*. Pengujian dilakukan dengan mengevaluasi perilaku *close-loop* dari sistem di bawah kontrol dari berbagai perubahan parameter dan bagaimana bisa menolak gangguan eksternal.

1.2 Rumusan Masalah

Berdasarkan tinjauan latar belakang yang telah dijelaskan dapat dilakukan perumusan masalah sebagai berikut:

1. Bagaimana merancang bangun *underwater* ROV.
2. Bagaimana mengendalikan pergerakan dan menjaga kestabilan *underwater* ROV di dalam air.
3. Bagaimana merancang sebuah sistem kendali *underwater* ROV agar dapat melakukan *hold position* di dalam air.
4. Bagaimana agar *underwater* ROV dapat beradaptasi terhadap perubahan arah arus atau gangguan (*disturbance*) yang ada di dalam air.

1.3 Tujuan

Adapun tujuan dan manfaat dilakukan penelitian ini yaitu:

1. *Underwater* ROV (*remotly operated vechile*) dapat membantu dalam proses evakuasi serta penyelamatan khususnya di dalam air.
2. Meringankan penyelam dengan memberikan informasi dan mengindikasikan posisi atau titik suatu objek.
3. Cepat tanggap dalam proses evakuasi atau mencari objek tertentu.
4. Mempertahankan posisi stabilisasi *underwater* ROV terhadap gangguan (*disturbance*) selama proses penyelamatan di dalam air.

1.4 Batasan Masalah

Batasan masalah diperlukan bila penelitian tesis terlalu luas sehingga perlu dilakukan batasan-batasan dalam penyelesaian permasalahan dalam penelitian. Batasan juga dapat berupa batasan fisik sistem bila tesis membahas implementasi metode pada sistem real.

Penelitian ini dibatasi oleh beberapa hal, yaitu:

1. *Hold position* berdasarkan posisi dari *underwater* ROV secara vertikal terhadap sumbu z di dalam air.
2. Parameter yang digunakan dalam sistem kendali yaitu tekanan air pada tingkat kedalaman air.
3. Sistem kendali yang diterapkan menggunakan metode kendali adaptif.

1.5 Kontribusi

Dari hasil penelitian yang telah dilakukan, diharapkan dapat memberikan kontribusi dalam sistem kendali robot bawah air, khususnya pada sistem kendali *hold position* di dalam air serta dapat mempertahankan kondisi dan posisi terhadap gangguan yang ada disekitarnya.

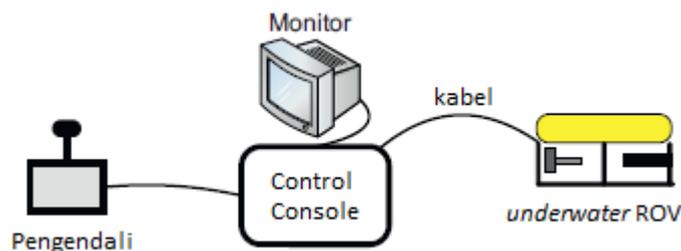
BAB 2

KAJIAN PUSTAKA

Pengertian ROV (*remotely operated vehicle*) menurut Marine Technology Society ROV Committee's "Operational Guidelines for ROVs" (1984) dan the National Research Council Committee's "Undersea Vehicles and National Needs" (1996) adalah sebuah robot bawah laut yang dikendalikan oleh operator, untuk tetap dalam kondisi yang aman, pada saat wahana bekerja di lingkungan yang berbahaya. Secara umum ROV dikenal sebagai suatu kapal selam yang biasa digunakan untuk membantu kegiatan lepas pantai pada industri minyak dan gas.

Definisi lain disampaikan oleh Christ dan Wernli (2007), dimana ROV adalah kamera yang dipasang dalam wadah tahan air, dengan pendorong untuk bermanuver, yang melekat pada kabel ke permukaan di mana sinyal video yang dikirim [4]. Sebuah ROV menerima energi dan informasi perubahan dengan panel kontrol yang terletak di permukaan melalui kabel pusat. Dapat di lihat pada Gambar 2.1 bahwa dari panel kontrol, operator dapat merencanakan pekerjaan atau menggunakan pengendali untuk manuver wahana secara langsung.

ROV pertama kali dibuat pada tahun 1953 oleh Dimitri Rebikoff dengan nama POODLE. Langkah nyata pertama diambil oleh angkatan laut AS (Amerika Serikat) untuk membuat sistem operasi ROV. Angkatan laut AS membuat ROV dengan nama *Cable-Controlled Underwater Research Vehicle* (CURV).



Gambar 2.1 Dasar Sistem *Underwater* ROV [4]

Wahana atau ROV ini dibuat untuk mengambil bom dan torpedo yang hilang di dasar laut. Kemudian angkatan laut AS membuat *Pontoon Implacement Vehicle* (PIV) dan SNOOPY, di mana SNOOPY merupakan wahana pertama yang dapat dibawa ke mana-mana. Pada tahun 1974, lebih dari 20 ROV diciptakan. Kebutuhan akan keselamatan kerja menyebabkan perusahaan menggantikan fungsi penyelam dengan menggunakan ROV. Pada tahun 1990-an, diperkirakan ada lebih dari 100 perusahaan pembuat ROV, dan lebih dari 100 operator menggunakan 3000 macam ROV yang berbeda ukuran dan kemampuannya [4].

Secara umum, wahana bawah air dibedakan menjadi dua kategori yaitu wahana berawak (*Manned Underwater Vehicle*) dan wahana tidak berawak (*Unmanned Underwater Vehicle – UUV*). *Remotely Operated Vehicle* (ROV) adalah robot bawah air yang dapat bermanuver secara tinggi, dioperasikan oleh seseorang di atas kapal. ROV dihubungkan dengan kabel yang membawa sinyal elektrik secara bolak-balik antara operator dan wahana ini. Beberapa instrumen dapat ditambahkan untuk menambahkan kemampuan ROV seperti kamera, manipulator, *water sampler*, CTD (*Conductivity, Temperature and Depth*) [5].

Sebelumnya telah ada penelitian yang berkaitan dengan *underwater* ROV ini. Di antaranya pada jurnal [6], penelitian ini memperhatikan pada sistem gerak dinamik dari *underwater* ROV dengan enam derajat kebebasan, model *non linear hydrodynamic* dengan parameternya dan melakukan simulasi numerik di Matlab/Simulink dari sebuah mode kendali *model-free second order sliding* terhadap perubahan arah arus laut.

Selanjutnya pada jurnal [7], dibahas tentang kendali *fuzzy* pada *roll*, *yaw* dan kedalaman dari *Autonomous Underwater Vehicle* (AUV), di mana sudut *roll* dan *yaw* yang diatur hanya menggunakan *error* dari AUV, namun karena kompleksitas dari *channel* kedalaman dinamik maka ditambahkan nilai kuantitas *pitch* yang digunakan untuk meningkatkan performa *depth loop*. Dengan menerapkan metode *normalized steepest descent optimization*, nilai optimal untuk parameter fungsi keanggotaan dapat ditemukan. Meskipun AUV adalah sistem yang sangat nonlinier, simulasi dirancang berdasarkan sistem kendali logika *fuzzy* pada persamaan gerak menunjukkan perilaku yang diinginkan dari AUV khususnya bila parameter fungsi keanggotaan *fuzzy* dioptimalkan.

2.1 Daya Apung

Daya apung (*buoyancy*) adalah daya tekan ke atas dari cairan (fluida) terhadap suatu benda yang sebagian atau seluruhnya dicelupkan ke dalam cairan. Hal ini terjadi disebabkan adanya reaksi dari cairan tersebut terhadap berat benda yang tercelup ke dalam cairan. Daya apung merupakan kemampuan dari suatu benda yang berada di cairan untuk mengapung dengan berat benda tersebut.

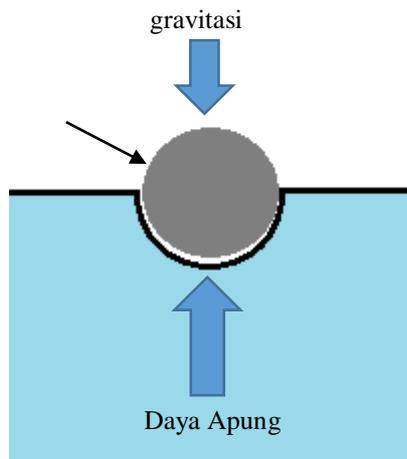
Hukum Archimedes memberikan pemahaman kepada kita tentang tekanan yang terjadi pada benda yang diletakkan pada zat cair. Hukum Archimedes ditemukan oleh ilmuwan berkebangsaan Yunani pada tahun 187-212 SM yang bernama Archimedes. Archimedes adalah seorang penemu dan ahli matematika dari Yunani yang terkenal sebagai penemu hukum hidrostatis atau yang sering disebut Hukum Archimedes.

Hukum Archimedes menyatakan, “Jika sebuah benda dicelupkan ke dalam zat cair, maka benda tersebut akan mendapat gaya yang disebut gaya apung (gaya ke atas) sebesar berat zat cair yang dipindahkannya”. Sebuah benda yang tenggelam seluruhnya atau sebagian dalam suatu fluida akan mendapatkan gaya angkat ke atas yang sama besar dengan berat fluida fluida yang dipindahkan.

Setiap benda yang berada di dalam air memiliki gaya apung ke atas. Gaya apung timbul karena adanya perluasan dari volume air yang dipindahkan. Gaya apung memiliki nilai yang sama dengan berat fluida yang dipindahkan [8]. Akibat adanya gaya apung, berat benda dalam zat cair akan berkurang. Jadi, telah jelas bahwa berat benda seakan berkurang bila benda dimasukkan ke dalam air. Hal itu karena adanya gaya ke atas yang ditimbulkan oleh air dan diterima benda. Dengan demikian maka resultan gaya antara gaya berat dengan gaya ke atas merupakan berat benda dalam air. Selanjutnya berat disebut dengan berat semu yaitu berat benda tidak sebenarnya karena benda berada dalam zat cair. Hubungan antara berat benda di udara (W), gaya ke atas (F_a) dan berat semu (W_s) dapat dilihat pada Persamaan 2.1,

$$W_s = W - F_a \quad (2.1)$$

dimana W_s adalah berat benda dalam zat cair (Kgm/S^2), W adalah berat benda sebenarnya (Kgm/S^2), dan F_a adalah gaya apung (N).



Gambar 2.2 Daya Apung Benda

Besarnya gaya apung F_a dapat dilihat pada Persamaan 2.2,

$$F_a = \rho_{cair} \times V_b \times g \quad (2.2)$$

dimana ρ_{cair} adalah massa jenis zat cair (kg/m^3), V_b adalah volume benda yang tercelup (m^3), g adalah percepatan gravitasi (ms^2).

Hukum ini bukan suatu hukum fundamental karena dapat diturunkan dari hukum Newton, jika:

- $FA = W$ maka benda akan melayang di dalam air.
- $FA > W$ maka benda akan terdorong ke atas akan melayang.
- $FA < W$ maka benda akan terdorong ke bawah dan tenggelam.

Gambar ilustrasi dari sebuah benda yang berada di air ditunjukkan pada Gambar 2.2, dimana terdapat dua gaya yang bekerja yaitu gaya apung yang mendorong benda ke atas dan gaya gravitasi dengan gaya berlawanan.

2.1.1 Benda Dalam Hukum Archimedes

Bila suatu benda dicelupkan ke dalam zat cair, maka ada 3 kemungkinan yang akan terjadi yaitu tenggelam, melayang, dan terapung.

1. Benda Tenggelam

Benda disebut tenggelam dalam zat cair apabila posisi benda selalu terletak pada dasar tempat zat cair berada, ilustrasi dan gaya yang bekerja pada

benda tenggelam dapat dilihat pada Gambar 2.3. Pada benda tenggelam terdapat tiga gaya, yaitu:

- 1) Gaya berat benda (W)
- 2) Gaya Archimedes (F_a)
- 3) Gaya normal bidang (N)

Dalam keadaan seimbang maka $W = N + F_a$ sehingga,

$$W > F_a$$

$$m \times g > \rho_{ZC} \times V_b \times g$$

$$\rho_b \times V_b \times g > \rho_{ZC} \times V_b \times g$$

$$\rho_b > \rho_{zc}$$

dimana ρ_b adalah massa jenis benda, dan ρ_{ZC} adalah massa jenis zat cair.

2. Benda Melayang

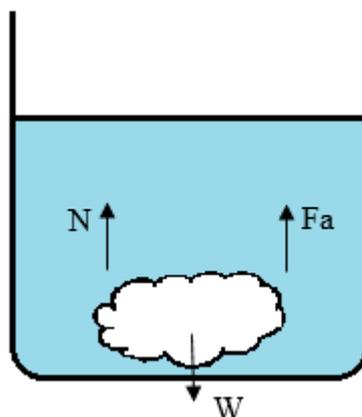
Benda disebut melayang dalam zat cair apabila posisi benda di bawah permukaan zat cair dan di atas dasar tempat zat cair, ilustrasi dan gaya yang bekerja pada benda yang melayang dapat dilihat pada Gambar 2.4.

Pada benda melayang terdapat dua gaya, yaitu gaya Archimedes (F_a) dan gaya berat benda (W). Dalam keadaan seimbang,

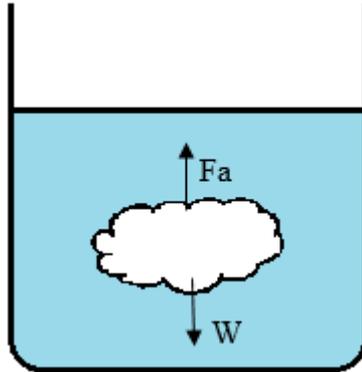
$$W = F_a$$

$$\rho_b \times V_b \times g = \rho_{ZC} \times V_b \times g$$

$$\rho_b = \rho_{zc}$$



Gambar 2.3 Benda Tenggelam



Gambar 2.4 Benda Melayang

3. Benda Terapung

Benda disebut terapung dalam zat cair apabila posisi benda sebagian muncul di permukaan zat cair dan sebagian terbenam dalam zat cair, ilustrasi dan gaya yang bekerja pada benda yang melayang dapat dilihat pada Gambar 2.5.

Pada benda terapung terdapat dua gaya, yaitu gaya Archimedes (F_a) dan gaya berat benda (W). Dalam keadaan seimbang,

$$W = F_a$$

$$\rho_b \times V_b \times g = \rho_{ZC} \times V_2 \times g$$

$$\rho_b \times V_b = \rho_{ZC} \times V_2$$

karena $V_b > V_2$, maka : $\rho_b < \rho_{ZC}$

Adapun penerapan Hukum Archimedes pada *underwater* ROV untuk menentukan massa jenis benda ditunjukkan pada Persamaan 2.3 dan massa jenis benda terhadap masa jenis air ditunjukkan pada Persamaan 2.4,

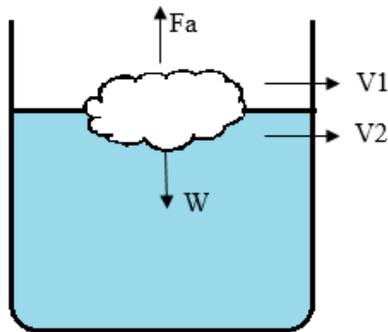
$$\rho_{benda} = \frac{m}{V_{benda}} = \frac{m}{V_{air}} \quad (2.3)$$

dimana hukum Archimedes menyatakan $V_{benda} = V_{air}$ maka,

$$\rho_{benda} = \frac{m}{m - m_s} \times \rho_{air} \quad (2.4)$$

sedangkan masa jenis air dinyatakan pada Persamaan 2.5,

$$\rho_{benda} = \frac{m - m_s}{\rho_{air}} \quad (2.5)$$



Gambar 2.5 Benda Terapung

dimana V_{air} adalah volume air yang dipindahkan, m adalah massa benda di udara, m_s adalah massa semu benda (di air), ρ_{benda} adalah massa jenis benda, dan ρ_{air} adalah massa jenis air.

Underwater ROV yang akan dirancang nanti kondisi idealnya akan dibuat melayang di dalam air, sehingga akan lebih mudah bagi *underwater* ROV untuk melakukan pergerakan dan bermanuver di dalam air. kondisi ini juga dapat meringankan kendali sistem *underwater* ROV dalam mempertahankan posisi pada tingkat kedalaman air.

2.1.2 Tekanan Hidrostatik

Tekanan hidrostatik adalah tekanan yang diakibatkan oleh gaya yang ada pada zat cair terhadap suatu luas bidang tekan pada kedalaman tertentu. Tekanan ini terjadi karena adanya berat air yang membuat cairan tersebut mengeluarkan tekanan. Sebuah benda padat yang terbenam dalam fluida akan mengalami gaya apung yang besarnya sama dengan berat fluida yang dipindahkan. Hal ini disebabkan oleh tekanan hidrostatik fluida. Besarnya tekanan ini bergantung kepada ketinggian zat cair, massa jenis dan percepatan gravitasi. Hubungan ini dikenal sebagai hukum Pascal, dirumuskan pada Persamaan 2.6,

$$p = \rho \times g \times h \quad (2.6)$$

dimana p adalah tekanan hidrostatik dalam P_a (N/m^2), ρ (rho) adalah masa jenis cairan ($1000 \text{ Kg}/m^3$), g adalah gravitasi ($9,8 \text{ m}/s$), dan h adalah tingkat

kedalaman cairan (m). Tekanan hidrostatik adalah kekuatan isotropik, yang berarti gaya yang diberikan pada objek adalah sama dari semua arah.

Tingkat kedalaman dapat dianggap terdiri dari beberapa lapis. Lapisan yang lebih atas akan membebani lapisan di bawahnya. Tekanan pada lapisan teratas hanya berasal dari tekanan udara luar, sedangkan lapisan yang di bawahnya mendapat tekanan dari udara luar dan dari berat lapisan di atasnya. Lapisan-lapisan fluida tadi diam di dalam fluida dan mendapat tekanan dari fluida yang lain baik di atas maupun di bawahnya. Karena silinder fluida diam maka resultan gaya yang dialami adalah nol.

Tekanan hidrostatik ini bekerja terhadap *underwater* ROV yang akan didesain mengapung di dalam air. Besar tekanan ini juga dimanfaatkan sensor untuk mengukur tekanan yang diberikan sehingga dapat mengetahui tingkat kedalaman *underwater* ROV di dalam air.

2.1.3 Tekanan dan Kedalaman Air

Tekanan (P) adalah satuan fisika untuk menyatakan gaya (F) per satuan luas (A) dinyatakan pada Persamaan 2.7,

$$P = \frac{F}{A} \quad (2.7)$$

dimana P adalah tekanan dengan satuan pascal (*Pressure*), F adalah gaya dengan satuan newton (*Force*), dan A adalah luas permukaan dengan satuan m^2 (*Area*).

Tekanan air laut bertambah terhadap kedalaman. Kedalaman air laut biasanya diukur dengan menggunakan *echo sounder* atau CTD (*Conductivity, Temperature, Depth*). Kedalaman yang diukur dengan menggunakan CTD didasarkan pada harga tekanan.

Tekanan didefinisikan sebagai gaya per satuan luas. Semakin ke dalam, tekanan air laut akan semakin besar. Hal ini disebabkan oleh semakin besarnya gaya yang bekerja pada lapisan yang lebih dalam. Satuan dari tekanan dalam cgs (*Centimeter Gram Second*) adalah dynes/cm², sedangkan dalam mks (*Meter Kilogram Second*) adalah Newton/m². Satu Pascal sama dengan satu Newton/m².

Dalam oseanografi, satuan tekanan yang digunakan adalah desibar (disingkat dbar), dimana $1 \text{ dbar} = 10^{-1} \text{ bar} = 10^5 \text{ dynes/cm}^2 = 10^4 \text{ Pascal}$.

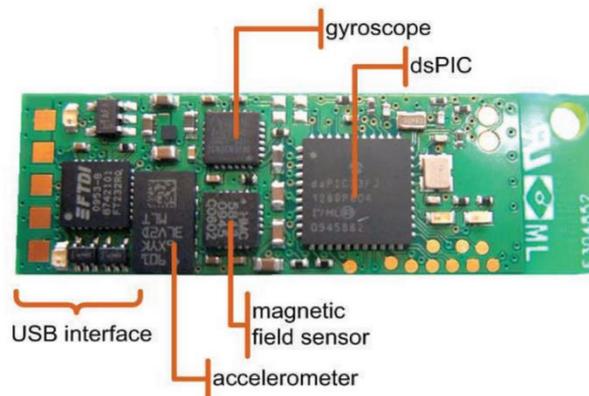
Gaya akibat tekanan bekerja dari tekanan yang berbeda pada satu titik ke titik lainnya. Gaya ini bekerja dari tekanan yang lebih tinggi ke tekanan yang lebih rendah. Di laut, gaya gravitasi yang bekerja (ke arah bawah) akan diimbangi oleh gaya akibat adanya perbedaan tekanan tersebut (ke arah atas), sehingga air yang bergerak ke bawah tidak akan mengalami percepatan.

2.2 Sensor

Sensor adalah suatu alat yang digunakan untuk mendeteksi adanya perubahan lingkungan baik secara fisik atau kimia. Pada umumnya suatu sensor dilengkapi oleh suatu rangkaian yang disebut rangkaian *signal conditioning*, yang kemudian disebut dengan istilah sensor *transmitter*. Keluaran dari sensor *transmitter* ini adalah sinyal listrik DC, yang dapat berupa tegangan atau arus listrik, untuk kemudian ditransmisikan melalui penghantar listrik dengan jangkauan jarak sesuai dengan kualitas penghantar tersebut dan kuantitas sinyal yang ditransmisikan. Pada penelitian ini akan digunakan dua jenis sensor, sensor IMU (*Inertial Measurement Units*) sebagai pendeteksi gerak rotasi dan posisi, sensor tekanan untuk mengukur tingkat kedalaman air.

2.2.1 Sensor IMU (*Inertial Measurement Unit*)

Inertial Measurement Units (IMU) adalah alat elektronik yang dapat mengukur dan mengetahui kecepatan sudut, orientasi, dan gaya gravitasi dengan menggunakan kombinasi dari pembacaan sensor *accelerometer* dan sensor *gyroscope*, terkadang juga *magnetometer*. IMU adalah komponen utama pada sistem navigasi kemiringan yang digunakan pada pesawat ataupun kapal. IMU juga telah banyak digunakan pada *smartphone* yang ada sekarang.



Gambar 2.6 Sensor IMU ETHOS (*ETH Orientation Sensor*).

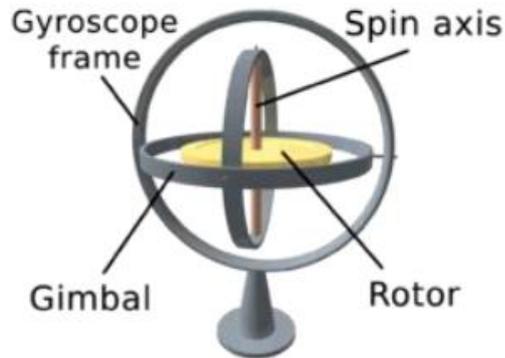
Inertial Measurement Unit (IMU) merupakan alat yang memanfaatkan sistem pengukuran seperti *gyroscope* dan *accelerometer* untuk memperkirakan posisi relatif, kecepatan, dan akselerasi dari gerakan motor. IMU adalah bagian dari navigasi sistem yang dikenal sebagai *Inertial Navigation System* atau INS.

IMU mempertahankan *6-degree-of-freedom* (DOF) yang memperkirakan gerakan yaitu posisi (X,Y,Z) dan orientasi (*roll*, *pitch*, *yaw*). Sistem seperti IMU hanya mempertahankan perhitungan terus menerus dari orientasi yang dikenal sebagai *Attitude and Heading Reference System* (AHRS) dan dipergunakan dalam cara yang sama sebagai IMU tetapi mempertahankan representasi tidak menyeluruh.

IMU sangat sensitif untuk mengukur kesalahan di *gyroscope* dan *accelerometer* yang mendasar. Penyimpangan giroskop mengarahkan pada kesalahan perhitungan dari orientasi motor, relatif terhadap gravitasi, menghasilkan kegagalan yang tidak tepat dari vektor gravitasi, seperti data *accelerometer* yang di integrasi dua kali, sisa vektor gravitasi akan menghasilkan kesalahan quadratic dalam posisi.

2.2.2 *Gyroscope*

Gyroscope adalah alat yang digunakan untuk mengukur atau mempertahankan orientasi berdasarkan prinsip momentum angular. Pada prinsipnya *mechanical gyroscope* adalah sebuah piringan (*rotor*) yang berputar pada sumbu (*axis*) yang mampu bergerak ke beberapa arah. *Gyroscope* mengukur perubahan sudut seputar sumbu tetap terhadap ruang inersia.



Gambar 2.7 Sistem Mekanika Pada *Gyroscope*

Bagian dari gyroscope terdiri dari sebuah piringan (*rotor*) yang berputar pada sumbu putar (*spin axis*). Sumbu putar ini terpasang pada suatu kerangka yang disebut gimbal (*inner-most gimbal*). *Inner-most* gimbal terpasang pada *inner* gimbal, dan *inner gimbal* terpasang pada *outer gimbal* yang merupakan kerangka terluar.

Sistem mekanik pada *gyroscope* dapat dilihat pada Gambar 2.7, dimana *gyroscope* memiliki tiga gimbal sehingga mempunyai kemampuan berputar pada tiga sumbu putar (*3 degree of rotational freedom*). Perputaran gimbal (kerangka luar) tidak mengubah posisi dari *rotor*. Prinsip inilah yang kemudian dimanfaatkan dalam instrumen *underwater ROV* untuk mendeteksi gerak *yaw*, *roll* dan *pitch*.

2.2.3 *Accelerometer*

Accelerometer adalah sebuah transduser yang berfungsi untuk mengukur percepatan, mendeteksi dan mengukur getaran, ataupun untuk mengukur percepatan akibat gravitasi bumi. *Accelerometer* juga dapat digunakan untuk mengukur getaran yang terjadi pada kendaraan, bangunan, mesin, dan juga bisa digunakan untuk mengukur getaran yang terjadi di dalam bumi, getaran mesin, jarak yang dinamis, dan kecepatan dengan ataupun tanpa pengaruh gravitasi bumi.

Ada banyak cara yang berbeda untuk membuat *accelerometer*. Beberapa *accelerometer* menggunakan piezoelektrik yang mengandung struktur kristal mikroskopis yang bisa ditekan oleh kekuatan *accelerative*, yang menyebabkan tegangan yang akan dihasilkan. Cara lain untuk melakukannya adalah dengan perubahan *penginderaan* di kapasitansi [9].

Antarmuka kapasitif memiliki beberapa fitur menarik. Dalam kebanyakan teknologi *micromachining* tidak ada atau minimal pengolahan tambahan yang diperlukan. Kapasitor dapat beroperasi baik sebagai sensor dan aktuator. Penginderaan kapasitif independen dari bahan dasar dan bergantung pada variasi kapasitansi ketika geometri kapasitor berubah. Mengabaikan efek *fringing* yang dekat tepi, adapun kapasitansi pelat sejajar ditunjukkan pada Persamaan 2.8,

$$C_0 = \epsilon_0 \epsilon_A \frac{A}{d} = \epsilon_A \frac{1}{d} \quad (2.8)$$

dimana ϵ_A adalah $\epsilon_0 \epsilon_A$, A adalah luas elektroda, d jarak antara kapasitor dan ϵ adalah permitivitas bahan yang memisahkan [10]. Perubahan pada salah satu parameter tersebut akan diukur sebagai perubahan kapasitansi dan variasi dari masing-masing tiga variabel telah digunakan dalam MEMS.

Jenis *accelerometer* MEMS terdiri dari perubahan pergerakan masa dengan piring yang terpasang melalui sistem suspensi mekanis dengan kerangka acuan, perpindahan plat dan piring luar tetap merupakan kapasitor [10]. Ruang bebas (udara) kapasitansi antara piring bergerak dan dua piring luar stasioner C_1 dan C_2 adalah fungsi yang sesuai perpindahan x_1 dan x_2 , besar kapasitansi tersebut ditunjukkan pada Persamaan 2.9,

$$C_1 = \epsilon_A \frac{1}{x_1} = \epsilon_A \frac{1}{d+x} = C_0 - \Delta C \quad , \quad C_2 = \epsilon_A \frac{1}{x_2} = \epsilon_A \frac{1}{d+x} = C_0 - \Delta C \quad (2.9)$$

dimana jika akselerasi adalah nol, kapasitansi C_1 dan C_2 adalah sama karena $x_1 = x_2$, buktinya perpindahan massa x karena hasil percepatan. Jika $x \neq 0$, maka perbedaan kapasitansi dapat dicari melalui Persamaan 2.10 dan 2.11,

$$C_2 - C_1 = 2\Delta C = 2\epsilon_A \frac{x}{d^2 - x^2} \quad (2.10)$$

$$\Delta C x^2 + \epsilon_A x - \Delta C d^2 = 0 \quad (2.11)$$

dan dapat disederhanakan menjadi Persamaan 2.12.

$$x \approx \frac{d^2}{\epsilon_A} \Delta C = d \frac{\Delta C}{C_0} \quad (2.12)$$

sensing element. Pengukuran tekanan absolut yang ditunjukkan pada Gambar 2.9, merupakan pengukuran tekanan relatif pada P_1 terhadap tekanan vakum P_0 ($P_0 = 0 \text{ bar}$) [12].

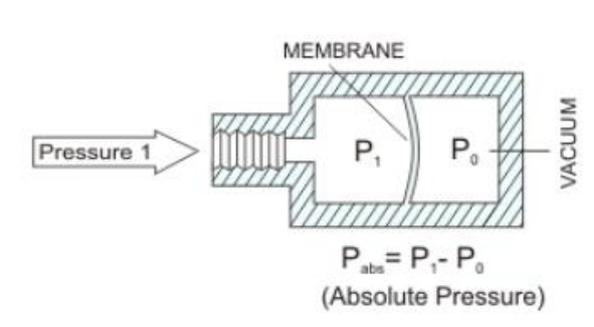
Sensor tekanan telah dikembangkan menggunakan efek *piezoresistive* pada beberapa material, meliputi silikon dan pol-Si. Sensor tekanan *piezoresistive* menggunakan efek *piezoresistive* dari silikon, sehingga dapat diketahui nilai resistansi dari Persamaan 2.15,

$$R = \rho_1 \frac{l}{A} \tag{2.15}$$

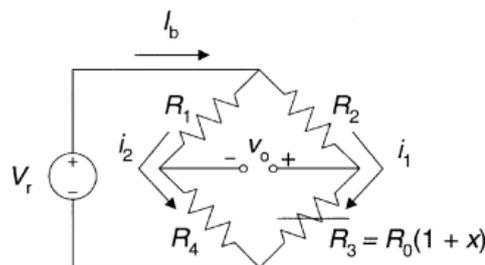
nilai dari ρ_1 dapat dilihat pada Persamaan 2.16,

$$\rho_1 = \frac{\partial \rho}{\varepsilon} \tag{2.16}$$

dimana R adalah resistansi, ρ adalah *piezoresistivitas*, $\partial \rho$ adalah perubahan resistivitas, ρ_1 adalah resistivitas awal, ε adalah tekanan, l adalah panjang konduktor, A adalah luas penampang.



Gambar 2.9 Diagram skematik dari *absolute pressure* sensor [12]



Gambar 2.10 Rangkaian Jembatan *wheatstone* [13]

Dari Persamaan 2.15 dan Persamaan 2.16 dapat diketahui bahwa resistansi yang diperoleh dipengaruhi oleh ketahanan, panjang dan luas penampang dari material yang digunakan. Perubahan panjang dan luas penampang mempengaruhi nilai resistansi pada material tersebut. Akan terjadi perubahan nilai hambatan pada material secara elektrik pada silikon yang digunakan ketika regangan (*strain*) ditekan atau digunakan. Hal ini merupakan efek dari *piezoresistive* dengan bahan semikonduktor berupa silikon. Perubahan tekanan dan tegangan yang dihasilkan, memiliki hubungan yang linier. Konfigurasi pada rangkaian jembatan *wheatstone* yang digunakan *piezoresistive* dapat dilihat pada Gambar 2.10, rangkaian ini digunakan untuk meningkatkan sensitivitas dari keluaran sensor yang diperoleh.

Berdasarkan dari rangkaian jembatan *wheatstone* maka dapat dihitung besar nilai tegangan *output* dari sensor terhadap perubahan besar tekanan yang diberikan terhadap sensor, Persamaan 2.17 dapat digunakan untuk menghitung perubahan tegangan tersebut [13].

$$V_o = V_r \left(\frac{R_3}{R_2 + R_3} - \frac{R_4}{R_1 + R_4} \right) \quad (2.17)$$

Pada saat sensor menerima tekanan, diafragma akan tertekan, dan menekan semikonduktor berupa silikon. Hal itu menyebabkan piezoresistor mengubah nilai tahanan pada resistor. Ketika terdapat tekanan maka nilai resistansi akan naik. Ketika nilai tahanan naik maka jika sensor tersebut diberi tegangan maka nilai tegangan yang dihasilkan pada sensor tersebut akan naik pula. Sehingga hubungan antara hambatan dan tegangan pada sensor ini berbanding lurus.

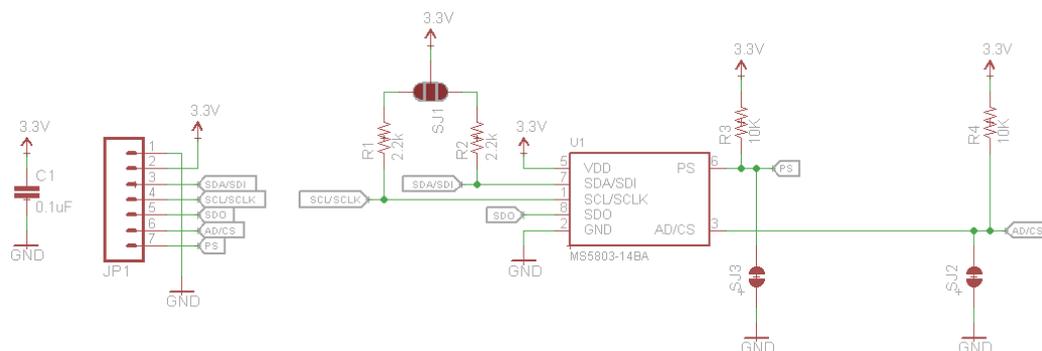
Pada penelitian ini akan digunakan sensor tekanan untuk mengetahui tingkat kedalaman air. Sensor tersebut adalah Sensor MS-5803-14BA yang merupakan salah satu jenis sensor tekanan *piezo-resistive* yang dapat diaplikasikan di dalam air, bentuk fisik dari sensor tersebut dapat dilihat pada Gambar 2.11. Sensor MS-5803-14BA adalah generasi baru dari sensor tekanan resolusi tinggi dengan antarmuka SPI dan I2C bus. Hal ini dioptimalkan untuk sistem pengukuran kedalaman dengan resolusi kedalaman air dari 1 cm. Modul sensor termasuk sebuah sensor linier tekanan tinggi. Prinsip kerja sensor ini dalam pengukuran tingkat kedalaman air berdasarkan pada tekanan hidrostatik.

Tekanan pada satu kedalaman bergantung pada massa air yang berada di atasnya. Persamaan yang digunakan untuk mengukur harga kedalaman dari harga tekanan adalah persamaan hidrostatis, yaitu $p = \rho \times g \times h$, di mana $p =$ perubahan tekanan, $\rho =$ densitas air laut, $g =$ percepatan gravitasi, dan $h =$ perubahan kedalaman. Jadi, jika tekanan berubah sebesar 100 dbar, dengan harga percepatan gravitasi $g = 9.8 \text{ m/s}^2$ dan densitas air laut $\rho = 1025 \text{ Kg/m}^3$, maka perubahan kedalamannya adalah 99,55 meter.

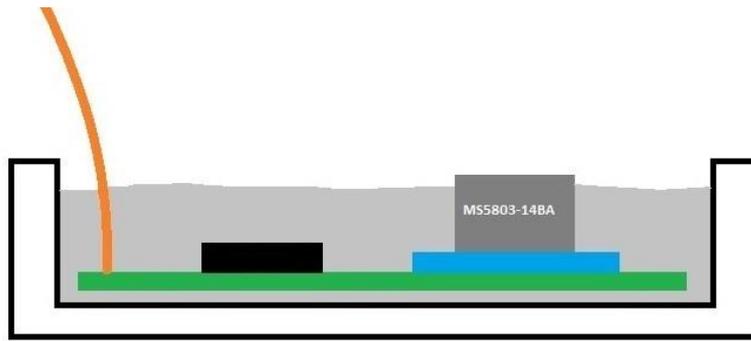
Desain gambar skematik dari sensor MS-5803-14BA dapat dilihat pada Gambar 2.12, sensor ini dapat dihubungkan ke mikrokontroler. Protokol komunikasi sederhana, tanpa perlu pemrograman register internal dalam perangkat. Sensor ini dilengkapi pelindung gel dan topi stainless baja anti magnetik yang dapat melindungi dari tekanan lebih dari 30 bar. Generasi modul sensor ini telah mengadopsi teknologi MEMS [14].



Gambar 2.11 Sensor Tekanan MS5803-14BA [14]



Gambar 2.12 Skematik Sensor MS5803-14BA [14]

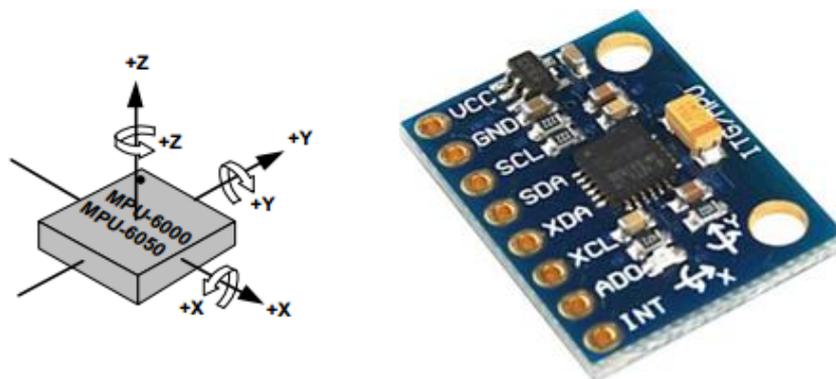


Gambar 2.13 Pemasangan Sensor Tekanan MS5803-14BA

Ilustrasi dari cara pemasangan atau peletakan sensor yang akan dipasang pada *underwater* ROV ditunjukkan seperti pada Gambar 2.13, bagian atas dari sensor yang akan kontak langsung ke air sehingga dapat mengukur tekanan air untuk mengetahui tingkat kedalaman air.

2.2.5 Sensor IMU MPU6050

Sensor MPU6050 adalah sensor yang terintegrasi dengan 6 sumbu *Motion Tracking* dengan penggabungan perangkat 3 sumbu *gyroscope*, 3 sumbu *accelerometer* dan sebuah *Digital Motion Processor (DMP)*. Dengan sistem komunikasi I2C sensor bus, sensor dapat langsung menerima masukan dari 3 sumbu sensor kompas sehingga dapat menghasilkan 9 sumbu *Motion Fusion*. Dengan sistem komunikasi tersebut maka komunikasi ke semua *register* dapat dilakukan dengan baik pada kecepatan 400KHz.



Gambar 2.14 Sensor IMU MPU6050

Bentuk fisik dan ilustrasi sumbu kerja dari sensor IMU MPU6050 dapat dilihat pada Gambar 2.14. Sensor IMU MPU6050 ini memiliki beberapa fitur, yaitu:

1. Gyroscope

Sensor MEMS 3 sumbu *gyroscope* MPU6050 memiliki fitur sebagai berikut:

- Digital *output* X-, Y-, dan Z- sumbu sudut sensor dengan tingkat *full programmable* berkisar pada ± 250 , ± 500 , ± 1000 , dan $\pm 2000^\circ/s$.
- Terintegrasi 16-bit ADC.
- Peningkatan bias dan sensitivitas suhu dapat distabilkan saat kalibrasi.
- Memiliki kemampuan untuk meredam *noise* pada frekuensi rendah (*Digital-Programmable Low Pass Filter*).

2. Accelerometer

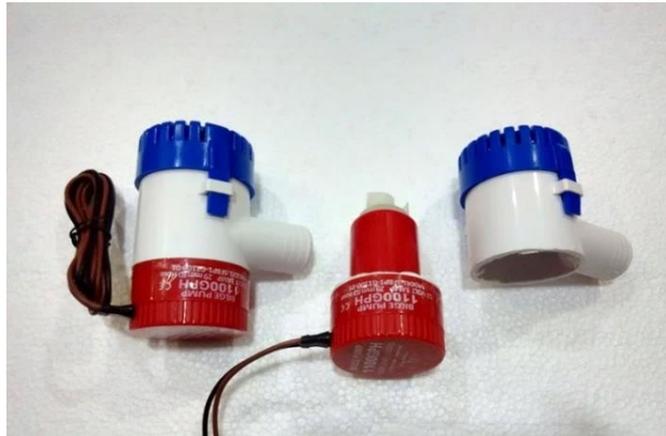
Sensor MEMS 3 sumbu *accelerometer* MPU6050 memiliki fitur sebagai berikut:

- *Digital – output* 3 sumbu *accelerometer* dengan *programmable full scale* berkisar dari $\pm 2g$, $\pm 4g$, dan $\pm 16g$.
- Terintegrasi 16-bit ADC.
- Mampu mendeteksi orientasi dan pensinyalan.
- Mampu mendeteksi sentuhan.
- *High-G interrupt*.

2.3 Motor DC

Jenis motor dapat dibedakan menjadi dua yakni motor AC dan motor DC. Perkembangan *underwater* ROV sejauh ini banyak yang menggunakan motor DC karena kekuatan, keragaman, kehandalan, dan kemudahan antarmuka (*interface*). Motor DC biasa digunakan karena memiliki kecepatan dan torsi yang bagus serta mudah dikendalikan arah putar dan kecepatannya.

Pada penelitian ini akan digunakan motor DC sebagai penggerak atau pendorong (*thruster*) *underwater* ROV. Jenis motor dc yang digunakan merupakan motor DC untuk pompa air pada lambung kapal (*bilge pumps* 1100 GPH) yang telah dimodifikasi.



Gambar 2.15 Motor DC *Bilge Pump*

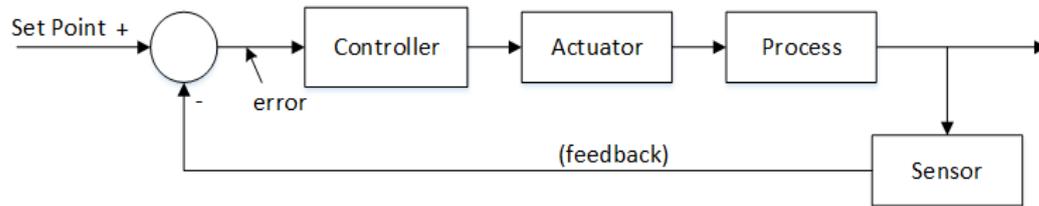
Motor DC jenis ini dibuat khusus tahan air dan beroperasi pada tegangan 12 VDC dengan arus sesuai dengan tipe dari motor itu sendiri. Bentuk dari jenis motor tersebut dapat dilihat pada Gambar 2.15. Motor DC yang digunakan akan dilengkapi dengan baling-baling (*propeller*), sehingga menghasilkan gaya dorong terhadap *underwater ROV* yang dimanfaatkan untuk membantu melakukan pergerakan atau bermanuver di dalam air.

Penempatan atau peletakan posisi motor tersebut pada *underwater ROV* dapat mempengaruhi stabilisasi. Pada *underwater ROV*, stabilisasi diperlukan agar ROV mudah dikendalikan dan dapat melakukan *hold position* pada tingkat kedalaman air.

2.4 Sistem Kendali

Sistem adalah susunan komponen fisik yang terhubung atau terkait sedemikian rupa sehingga membentuk atau bertindak sebagai seluruh unit dalam satu kesatuan. Sedangkan kata kontrol atau kendali biasanya diartikan mengatur, mengarahkan, atau perintah.

Dari kedua makna kata sistem dan kontrol/kendali, sistem kendali adalah suatu susunan komponen fisik yang terhubung atau terkait sedemikian rupa sehingga dapat memerintah, mengarahkan, atau mengatur diri sendiri atau sistem lain [15]. Sistem kendali terdiri dari subsistem dan proses (atau *plants*) yang disusun untuk mendapatkan keluaran (*output*) dan kinerja yang diinginkan dari masukan yang diberikan [16].



Gambar 2.16 Sistem Kendali *Closed-loop* [17]

Sistem kendali umumnya terbagi atas dua kategori yaitu *open-loop* dan *closed-loop*. Perbedaan ini berdasarkan dari tindakan atau aksi dari kontrol tersebut untuk mengaktifkan sistem untuk menghasilkan *output* [15].

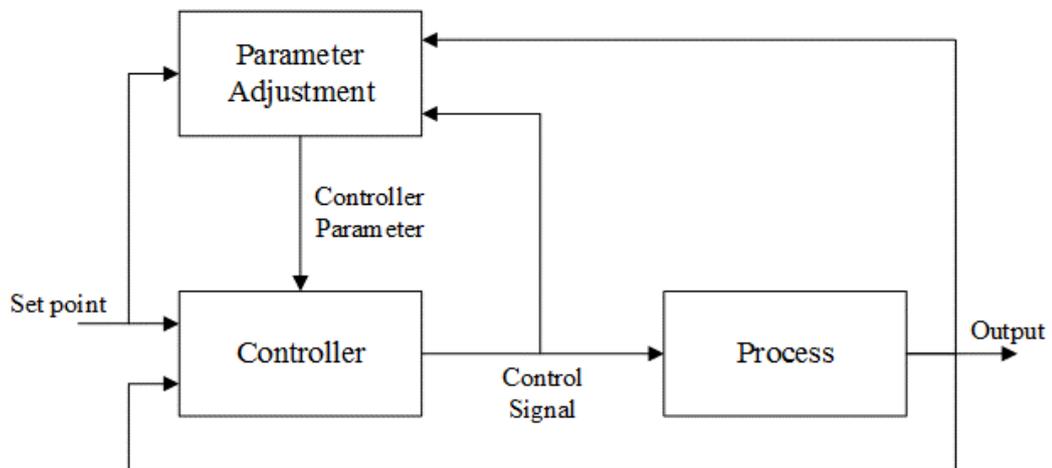
Blok diagram dari sistem kendali *closed-loop* atau juga dikenal sebagai sistem kontrol umpan balik dapat dilihat pada Gambar 2.16, keluaran dari proses ini terus dipantau oleh sensor. Hasil dari pembacaan sensor ini diumpan balikan ke *controller*, untuk menjaga *output* dari sistem sesuai dengan set poin yang diinginkan.

Sistem kendali pada penelitian ini merupakan sistem kendali *closed-loop*, dimana hasil pembacaan sensor untuk membaca tingkat kedalaman air diumpan balikan ke *controller* sehingga sistem dapat menjaga nilai *output* yang diinginkan.

2.4.1 Sistem Kendali Adaptif

Sistem kendali adaptif merupakan sistem kendali yang mempunyai parameter kendali yang dapat beradaptasi. Parameter kendali tersebut dapat beradaptasi terhadap perubahan kondisi lingkungan di sekitarnya, seperti adanya gangguan serta perubahan karakter internal dari sistem yang dikendalikan. Penggunaan sistem kendali adaptif menunjukkan peningkatan kinerja sistem karena suatu sistem umumnya berada dalam situasi yang mengandung derau dan gangguan serta kondisi internal dan eksternalnya mengandung ketidakpastian.

Sistem kendali adaptif terdiri dari dua bagian *loop* tertutup, *loop* pertama adalah *normal feedback Control* terhadap *process* dan *controller* dan *loop* yang kedua adalah *loop* dengan *parameter adjustment*. Blok diagram dari sistem kendali adaptif dapat dilihat pada Gambar 2.17.

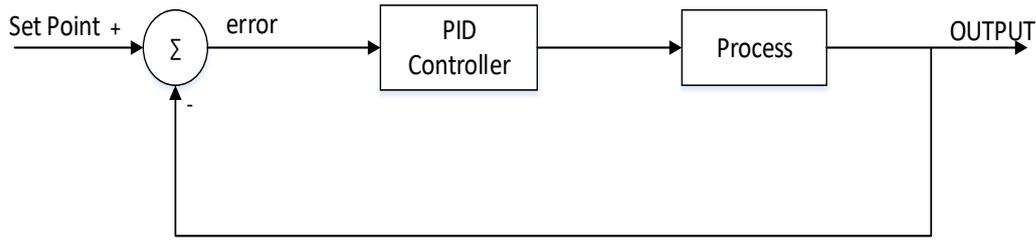


Gambar 2.17 Diagram Blok Sistem Kendali Adaptif [18]

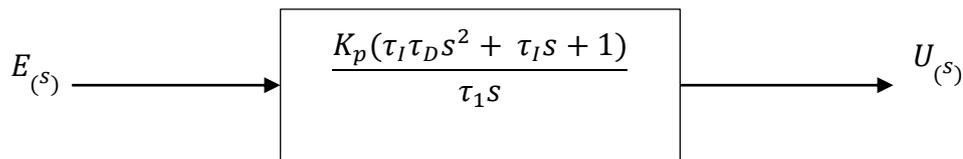
Dalam pengambilan keputusan (*decision function*), pada saat performansi sistem telah ditentukan, fungsi berikutnya adalah menetapkan perubahan apa yang harus dibuat untuk memperbaiki performansi. Fungsi pengambilan keputusan dilaksanakan dengan memakai algoritma terprogram dari sistem adaptif. Berdasarkan algoritma tersebut, keputusan mungkin mengubah satu atau lebih parameter *input* ke proses, untuk mengubah beberapa parameter internal kendali, atau perubahan-perubahan yang lain.

2.4.2 Sistem Kontrol PID

Sistem kontrol PID (*Proportional Integral Derivative*) merupakan kendali untuk menentukan presisi suatu sistem instrumentasi dengan karakteristik adanya umpan balik (*Feedback*) pada sistem tersebut. Kontrol PID merupakan algoritma kontrol yang banyak digunakan di industri proses karena strukturnya yang sederhana dan kemampuan yang tinggi dalam memecahkan banyak masalah kendali praktis. Blok diagram dari sistem kontrol PID secara umum ditunjukkan pada Gambar 2.18, dimana hasil *output* diumpan balik ke PID *controller* dan dihitung selisih eror dari nilai *set point* dan nilai *output*.



Gambar 2.18 Diagram Blok Sistem Kontrol PID



Gambar 2.19 Diagram Blok Kontroler Standar PID

Sistem kontrol PID terdiri dari tiga buah cara pengaturan yaitu kontrol P (*Proportional*), I (*Integral*), dan D (*Derivative*) dengan masing-masing memiliki kelebihan dan kekurangan. Dalam implementasinya masing-masing cara dapat bekerja sendiri maupun gabungan di antaranya. Dalam perancangan sistem kontrol PID yang perlu dilakukan adalah mengatur parameter P, I atau D agar tanggapan sinyal keluaran sistem terhadap masukan tertentu sebagaimana yang diinginkan.

Adapun hubungan sinyal eror dan sinyal kontrol pada kontroler tipe PID standar dapat dinyatakan pada Persamaan 2.18, dimana E adalah eror yang merupakan selisih antara set poin dengan umpan balik keluaran atau output, K_p adalah konstanta kendali *proportional*, $1/T_I$ adalah konstanta kendali *integral*, T_D adalah konstanta kendali *derivative* [17].

$$Output_{PID} = K_p \left[E + \frac{1}{T_I} \int E dt + T_D \frac{dE}{dt} \right] \quad (2.18)$$

Dalam bentuk diagram blok kontroler PID tipe ini dapat dilihat pada Gambar 2.19, sedangkan dalam bentuk *transfer function* dapat dinyatakan pada Persamaan 2.19,

$$\frac{(U)_s}{E(s)} = K_p \left(1 + \frac{1}{\tau_1 s} + \tau_D s \right) \quad (2.19)$$

atau bentuk lain dapat dilihat pada Persamaan 2.20,

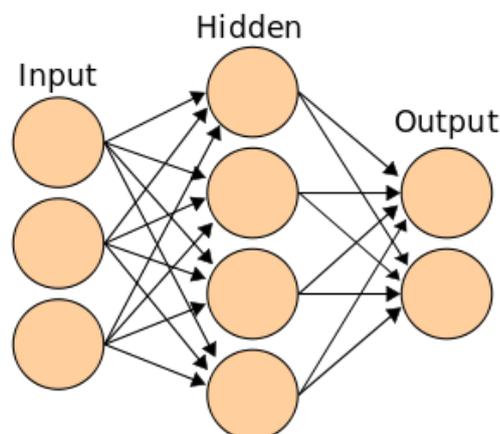
$$\frac{U(s)}{E(s)} = \frac{K_p(\tau_1 \tau_D s^2 + \tau_1 s + 1)}{\tau_1 s} \quad (2.20)$$

Pada kondisi operasi tertentu (seperti misalnya sering terjadi gangguan pada proses atau parameter proses yang berubah-ubah), parameter kontrol ini harus sering di atur agar kinerjanya tetap baik. Salah satu teknik dalam sistem kontrol yang sering dilakukan untuk mengatasi permasalahan ini adalah dengan menggunakan metode kendali adaptif PID, dimana parameter kontrol diubah secara otomatis jika terjadi perubahan kondisi operasi yang menyebabkan kinerja kontrol menurun.

Penggunaan kendali PID telah diusulkan pada banyak karya penelitian. Dapat dilihat pada jurnal [19], sebuah desain baru dari kinerja adaptif kendali PID dengan struktur prediksi model telah diusulkan. Sebuah desain kendali PID untuk sebuah kelas sistem PWM *feedback time-varying* dengan menggabungkan pendekatan fungsi ortogonal dan algoritma genetika telah dipelajari pada penelitian [20].

2.4.3 Jaringan Syaraf Tiruan (JST)

Jaringan syaraf tiruan (*artificial neural network*) atau juga disebut *simulated neural network* (SNN), atau umumnya hanya disebut *neural network* (NN) adalah sistem komputasi yang arsitektur dan operasinya diilhami dari pengetahuan tentang sel syaraf biologis di dalam otak. Jaringan syaraf tiruan merupakan salah satu representasi buatan dari otak manusia yang selalu mencoba menstimulasi proses pembelajaran pada otak manusia tersebut. Jaringan syaraf tiruan dapat digambarkan sebagai model matematis dan komputasi untuk fungsi aproksimasi non-linear, klasifikasi data cluster dan regresi non-parametrik atau sebuah simulasi dari koleksi model jaringan syaraf biologi.



Gambar 2.20 Diagram sederhana *Neural Network*

Model jaringan syaraf ditunjukkan dengan kemampuannya dalam emulasi, analisis, prediksi dan asosiasi. Pada Gambar 2.20 merupakan ilustrasi dari perancangan *neural network*, kemampuan yang dimiliki jaringan syaraf tiruan dapat digunakan untuk belajar dan menghasilkan aturan atau operasi dari beberapa contoh atau input yang dimasukkan dan membuat prediksi tentang kemungkinan output yang akan muncul atau menyimpan karakteristik input yang diberikan kepada jaringan syaraf tiruan.

Secara sederhana, JST adalah sebuah alat pemodelan data statistik non-linier. JST dapat digunakan untuk memodelkan hubungan yang kompleks antara input dan output untuk menemukan pola-pola pada data.

2.4.3.1 Konsep Dasar Jaringan Syaraf Tiruan (JST)

Jaringan syaraf tiruan terdiri dari beberapa neuron dan ada hubungan antar neuron-neuron seperti pada otak manusia. Neuron/sel saraf adalah sebuah unit pemroses informasi yang merupakan dasar operasi jaringan syaraf tiruan.

Jaringan syaraf tiruan terdiri atas beberapa elemen penghitung tak linier yang masing-masing dihubungkan melalui suatu pembobot dan tersusun secara paralel. Pembobot inilah yang nantinya akan berubah (beradaptasi) selama proses pelatihan.

Pelatihan perlu dilakukan pada suatu jaringan syaraf tiruan sebelum digunakan untuk menyelesaikan masalah. Hasil pelatihan jaringan syaraf tiruan dapat diperoleh tanggapa yang benar (yang diinginkan) terhadap masukan yang

diberikan. Jaringan syaraf tiruan dapat memberikan tanggapan yang benar walaupun masukan yang diberikan terkena gangguan atau berubah oleh suatu keadaan.

Metode pelatihan yang digunakan untuk meminimalkan eror pada keluaran adalah metode *backpropagation*. *Backpropagation* biasanya menggunakan jaringan multilayer. Penggunaan *backpropagation* terdiri dari beberapa tahap yaitu tahap belajar atau pelatihan dan tahap pengujian atau penggunaan

Tahap pelatihan *backpropagation* dilakukan dengan memasukkan data input ke jaringan (*feedforward*), menghitung propagasi balik dari eror dan pembaharuan bobot serta bias.

2.4.3.2 Karakteristik Jaringan Syaraf Tiruan

Penyelesaian masalah dengan jaringan syaraf tiruan tidak memerlukan pemrograman. Jaringan syaraf tiruan menyelesaikan masalah melalui proses belajar dari contoh-contoh pelatihan yang diberikan. Biasanya pada jaringan syaraf tiruan diberikan sebuah himpunan pola pelatihan yang terdiri dari sekumpulan contoh pola. Selama proses belajar itu pola masukan disajikan bersama-sama dengan pola keluaran yang diinginkan. Jaringan akan menyesuaikan nilai bobotnya sebagai tanggapan atas pola masukan dan sasaran yang disajikan tersebut.

- Faktor Bobot

Bobot merupakan suatu nilai yang mendefinisikan tingkat atau kepentingan hubungan antara suatu node dengan node yang lain. Semakin besar bobot suatu hubungan menandakan semakin pentingnya hubungan kedua node tersebut. Bobot merupakan suatu hubungan berupa bilangan real maupun integer, tergantung dari jenis permasalahan dan model yang digunakan. Bobot-bobot tersebut bisa ditentukan untuk berada didalam interval tertentu. selama proses pelatihan, bobot tersebut dapat menyesuaikan dengan pola-pola input.

Jaringan dengan sendirinya akan memperbaiki diri terus-menerus karena adanya kemampuan untuk belajar. Setiap ada suatu masalah baru, jaringan dapat belajar dari masalah baru tersebut, yaitu dengan mengatur kembali nilai bobot untuk menyesuaikan karakter nilai.

- Fungsi Aktivasi

Setiap neuron mempunyai keadaan internal yang disebut level aktivasi atau level aktivitas yang merupakan fungsi input yang diterima. Secara tipikal suatu neuron mengirimkan aktivitasnya kebeberapa neuron lain sebagai sinyal. Yang perlu diperhatikan adalah bahwa neuron hanya dapat mengirimkan satu sinyal sesaat, walaupun sinyal tersebut dapat dipancarkan ke beberapa neuron yang lain.

Ada beberapa pilihan fungsi aktivasi yang digunakan dalam metode *backpropagation*, seperti fungsi sigmoid biner, dan sigmoid bipolar. Karakteristik yang harus dimiliki fungsi aktivasi tersebut adalah kontinu, diferensiabel, dan tidak menurun secara monoton. Fungsi aktivasi diharapkan dapat mendekati nilai-nilai maksimum dan minimum secara baik. Berikut ini adalah fungsi aktivasi yang sering digunakan yaitu:

1. Fungsi Sigmoid Biner

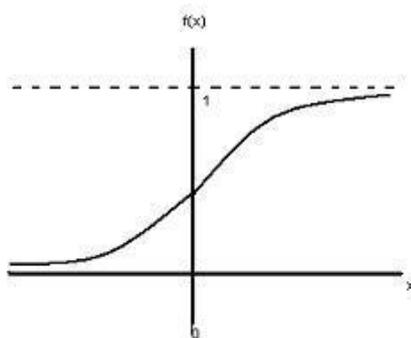
Fungsi ini digunakan untuk jaringan syaraf yang dilatih dengan menggunakan metode *backpropagation*. Fungsi sigmoid biner memiliki nilai pada range 0 sampai 1. Fungsi ini sering digunakan untuk jaringan syaraf yang membutuhkan nilai output yang terletak pada interval 0 sampai 1. Definisi fungsi sigmoid biner adalah sebagai berikut:

$$f_1(x) = \frac{1}{(1+e^{-x})} \tag{2.21}$$

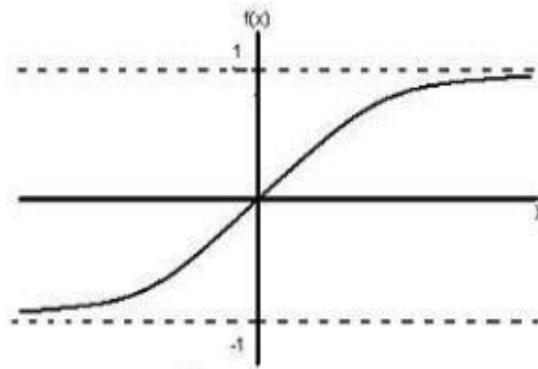
dengan turunan

$$f_1'(x) = f_1(x)(1 - f_1(x)) \tag{2.22}$$

dapat dilihat pada Gambar 2.21, yang merupakan ini ilustrasi fungsi sigmoid biner.



Gambar 2.21 Ilustrasi fungsi sigmoid biner dengan range (0,1)



Gambar 2.22 Ilustrasi fungsi sigmoid bipolar dengan range $(-1,1)$

1. Fungsi Sigmoid Bipolar

Fungsi sigmoid bipolar hampir sama dengan fungsi sigmoid biner, hanya saja output dari fungsi ini memiliki range antara 1 sampai -1. Definisi fungsi sigmoid bipolar adalah sebagai berikut:

$$f_2(x) = 2f_1(x) - 1 \quad (2.23)$$

dengan turunan

$$f_2'(x) = \frac{1}{2}(1 + f_2(x))(1 - f_2(x)) \quad (2.24)$$

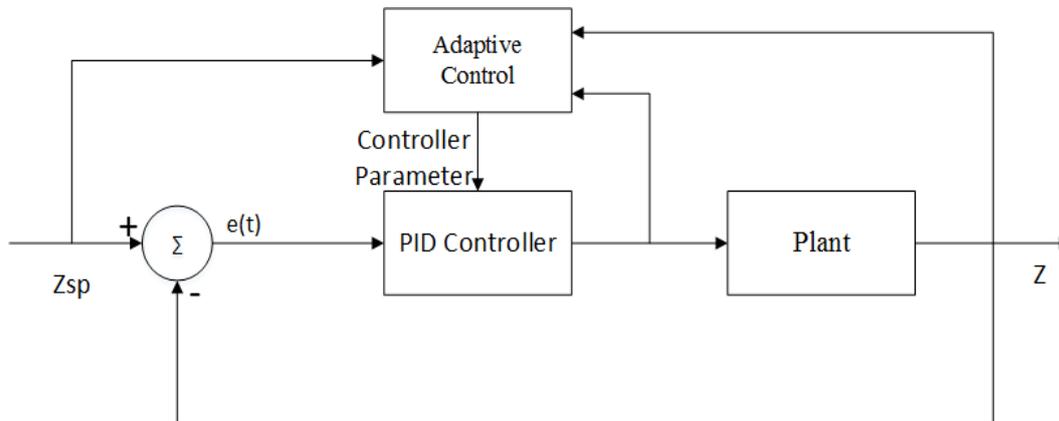
pada Gambar 2.22, merupakan ilustrasi fungsi sigmoid bipolar.

2.4.4 Sistem Kendali Adaptif PID

Pada sistem ini kendali *Neural Network* digunakan untuk menentukan besar nilai tiap parameter dari kontrol PID, prosedur *tuning* pada kontrol PID dapat dilakukan secara manual dengan metode *trial and error* namun belum tentu mendapatkan nilai yang benar.

Pada saat terjadi perubahan parameter terhadap *plan* atau objek, untuk mengatasi permasalahan tersebut dibutuhkan suatu pendekatan alternatif yang dapat mencapai suatu tingkat automasi dari proses tuning atau disebut *auto-tuning*.

Dalam penelitian ini metode alternatif yang digunakan adalah sistem kendali adaptif. Kendali adaptif diimplementasikan untuk mendapatkan nilai tuning pada tiap parameter K_p, K_i , dan K_d dari kontroler PID.



Gambar 2.23 Diagram Blok Sistem Kendali Adaptif PID

Penggunaan sistem kendali adaptif memungkinkan penggunaan berbagai metode tuning PID. Strategi kontrol ini juga menggabungkan keunggulan dari kontroler PID dan kendali adaptif sehingga sistem dapat mencapai respons yang lebih cepat, kekokohan yang bagus, akurasi *steady-state* yang bagus, dan kemampuan beradaptasi terhadap perubahan parameter *plant* yang diharapkan.

Blok diagram dari sistem kendali adaptif PID dapat dilihat pada Gambar 2.23, dimana *input* terdiri dari tiga variabel yaitu error, *output* sinyal kontrol PID, dan *output* dari *plant* yang digunakan sebagai variabel *input* pada sistem kendali adaptif, sedangkan variabel output dari hasil proses sistem kendali adaptif berupa nilai dari tiap parameter pada kontrol PID. Hasil *output* tersebut selanjutnya digunakan sebagai nilai parameter K_p, K_i , dan K_d yang baru.

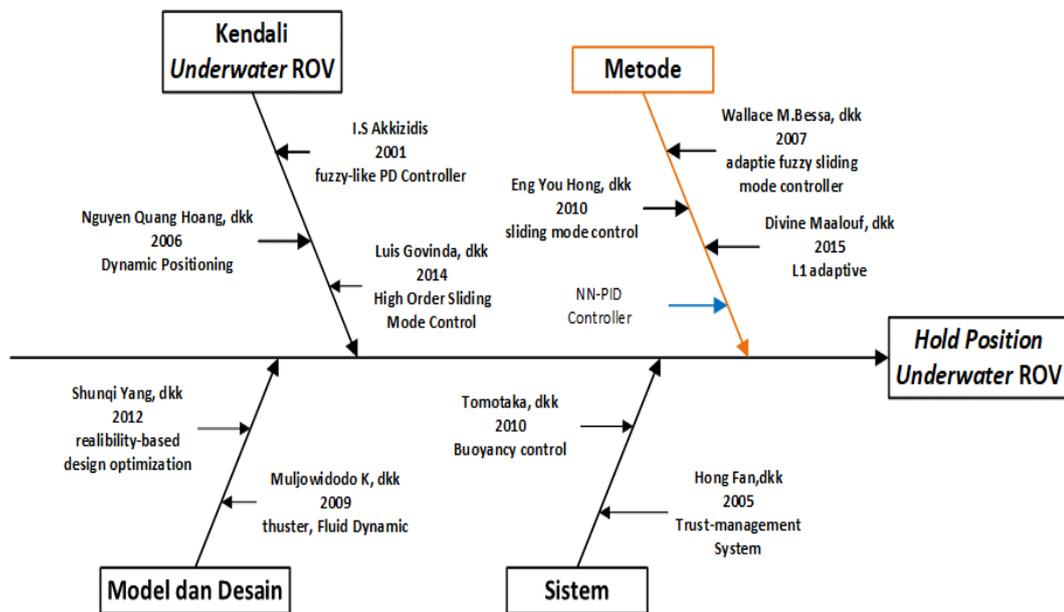
2.5 Fishbone Diagram Penelitian

Kerangka diagram *fishbone* pada penelitian yang akan dilakukan ditunjukkan pada Gambar 2.24. *Fishbone diagram* digunakan ketika ingin mengidentifikasi kemungkinan penyebab masalah dan terutama ketika sebuah *team* cenderung jatuh berpikir pada rutinitas [21].

Suatu tindakan dan langkah perbaikan akan lebih mudah dilakukan jika masalah dan akar penyebab masalah sudah ditemukan. Manfaat *fishbone* diagram ini dapat menolong untuk menemukan akar penyebab masalah secara *user friendly*.

Perancangan *fishbone* pada penelitian ini terdiri dari empat kategori tema dengan masalah utama yaitu kendali kedalaman pada *underwater* ROV. Tema tersebut di antaranya tentang metode, model dan disain, metode, sistem dan kendali *underwater* ROV.

Dari beberapa jurnal yang telah dikaji, maka pada akhir *fishbone* diagram didapatkan suatu permasalahan yang akan dilakukan dalam penelitian. Bagaimana sistem kendali kedalaman dari *underwater* ROV di dalam air, sehingga dapat mempertahankan posisi (*hold position*) pada tingkat kedalaman air.



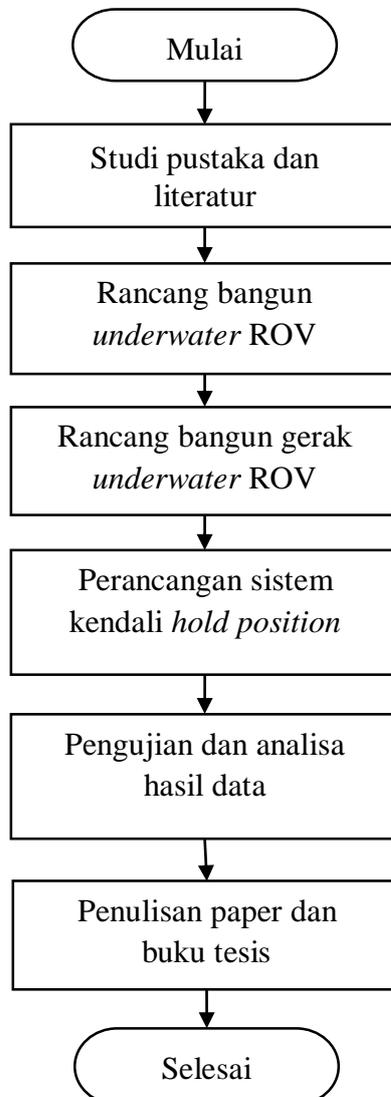
Gambar 2.24 *Fishbone* Diagram Penelitian

Halaman ini sengaja dikosongkan

BAB 3

METODOLOGI PENELITIAN

Pada penelitian ini akan dirancang sistem *hold position* pada *underwater* ROV untuk mempertahankan posisi pada tingkat kedalaman air dengan menggunakan metode kendali *neural network* PID (NN-PID). Penelitian dilakukan dengan beberapa tahapan di antaranya, studi pustaka dan literatur, rancang bangun *underwater* ROV, rancang bangun gerak *underwater* ROV, perancangan sistem kendali *hold position*, pengambilan data, analisa data. Tahapan penelitian dapat dilihat pada Gambar 3.1.

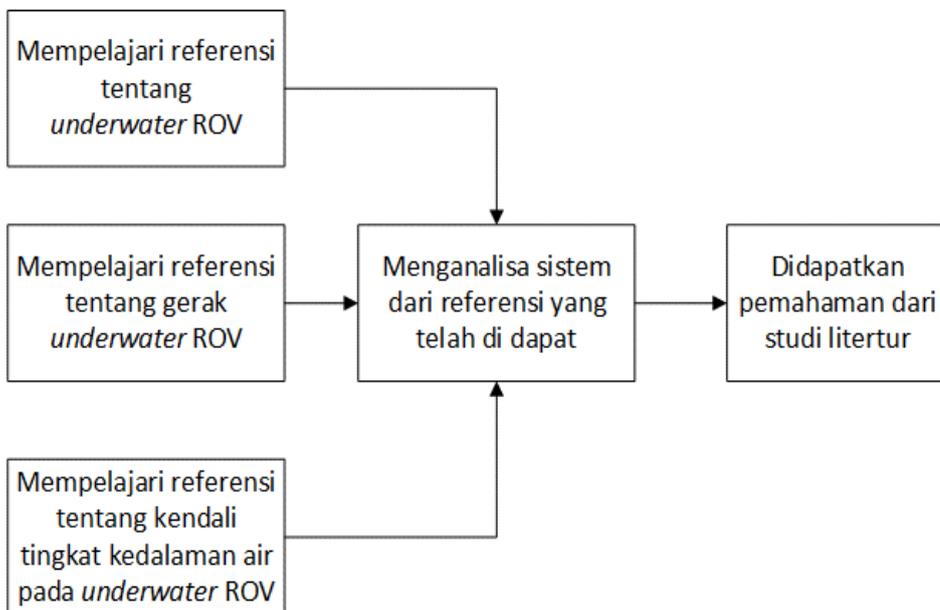


Gambar 3.1 Tahapan Penelitian

3.1 Studi Pustaka dan Literatur

Pada tahapan ini penulis akan mencari dan menelaah serta mempelajari dari beberapa pustaka dan literatur yang relevan dengan penelitian yang akan dikerjakan. Sebelumnya telah dilakukan pemetaan terhadap literatur di bab sebelumnya pada Gambar 2.24, beberapa di antaranya mempelajari referensi tentang *underwater* ROV secara umum, sistem gerakanya, dan terlebih terhadap sistem kendali yang dirancang menggunakan kombinasi antara *neural network* dan sistem kendali PID dalam mempertahankan posisi (*hold position*) pada tingkat kedalaman air.

Adapun Tujuan studi pustaka ini dilakukan untuk dijadikan dasar pemikiran dalam pengerjaan penelitian dan pembuatan laporan tesis, serta mengetahui peta domain tentang penelitian yang akan dilakukan. Ilustrasi studi pustaka dan literatur serta persiapan yang dilakukan sebelum pada tahap perancangan *underwater* ROV itu sendiri dapat dilihat pada Gambar 3.2.

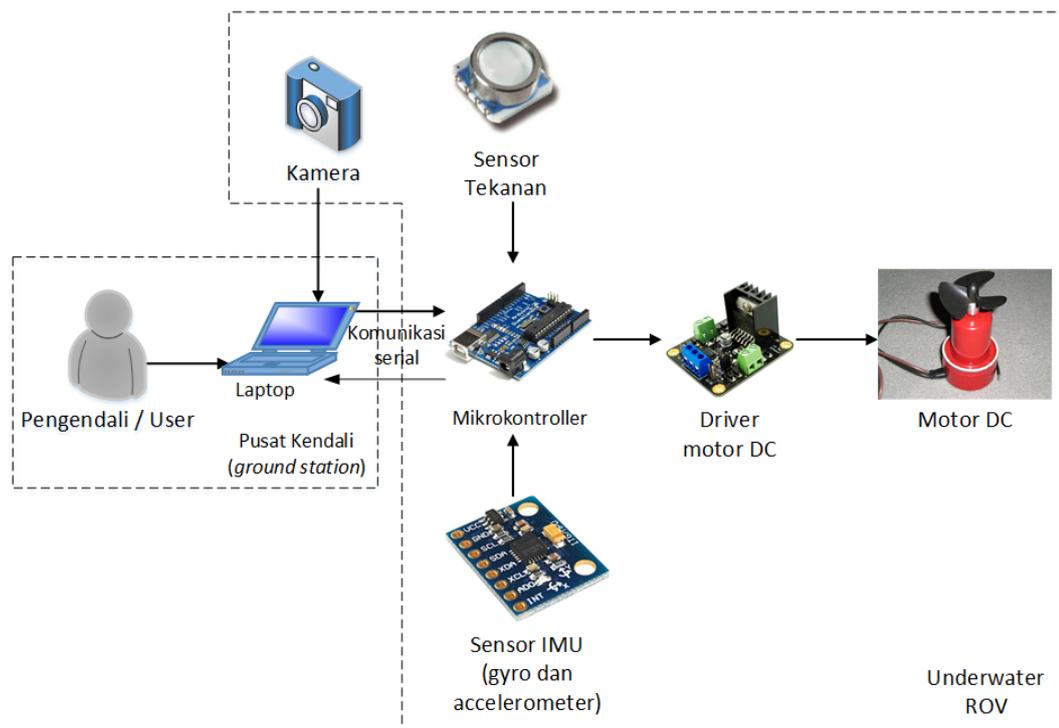


Gambar 3.2 Ilustrasi Studi Pustaka dan Literatur

3.2 Rancang Bangun *Underwater ROV*

Pada tahapan ini akan dilakukan rancang bangun *underwater ROV*, adapun blok rancang bangun dari *underwater ROV* dapat dilihat pada Gambar 3.3. Blok rancang bangun ini terbagi dua bagian, yaitu pusat kendali (*ground station*) dan *underwater ROV* itu sendiri. Pada bagian blok pusat kendali terdiri dari *user* atau pengguna sebagai pengendali melalui komputer dengan fitur GUI (*graphical user interface*) yang dibangun menggunakan perangkat lunak Visual Studio 2010, serta dapat mengetahui informasi kondisi dari *underwater ROV*.

Pada bagian blok *underwater ROV* terdiri dari mikrokontroler sebagai otak kendali, sensor tekanan yang digunakan untuk mengetahui tingkat kedalaman air, sensor IMU digunakan untuk membaca posisi dan orientasi sudut dari *underwater ROV* di dalam air, *driver* motor DC sebagai kendali motor DC, dan motor DC sebagai penggerak atau pendorong (*thruster*) yang diletakkan pada sisi-sisi sumbu x dan y pada badan robot untuk bernavigasi di dalam air. Terdapat juga kamera yang akan dipasang pada sisi bagian depan *underwater ROV*, digunakan untuk melihat objek-objek tertentu di dalam air.

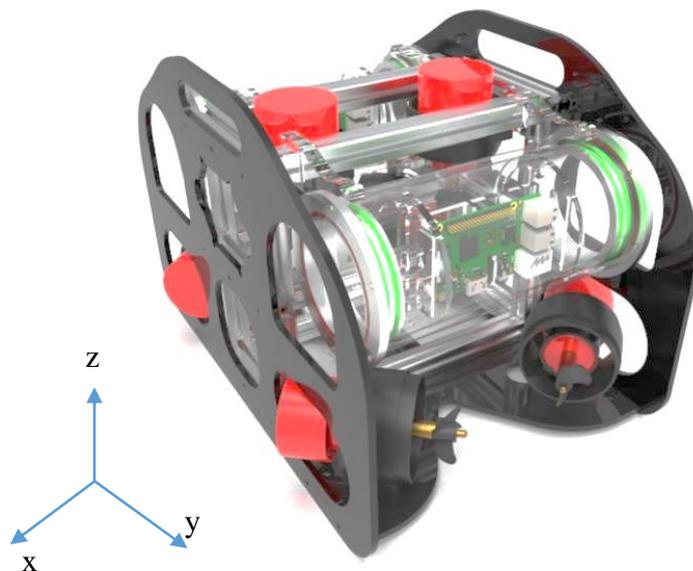


Gambar 3.3 Blok Rancang Bangun *Underwater ROV*

3.2.1 Rancang Bangun Mekanik

Rancang bangun desain mekanik dengan tampilan 3 dimensi dari *underwater* ROV dapat dilihat pada Gambar 3.4. Berbentuk persegi dan dirancang dengan enam buah motor DC, 2 buah dipasang vertikal searah sumbu z dan 4 buah dipasang horizontal searah sumbu x dan y. Kotak untuk tempat komponen elektronika dan sensor diletakkan di bagian depan dan belakang, didesain kedap air dengan memiliki tahanan terhadap tekanan yang cukup tinggi pada tingkat kedalaman air.

Desain mekanik ini dirancang agar *underwater* ROV dapat melayang (daya apung) di dalam air pada saat kondisi diam (*default position*), sehingga mempermudah dalam bermanuver di dalam air dan dapat mempertahankan kondisi *hold position* pada tingkat kedalaman air tertentu. Terdapat juga kamera yang akan dipasang pada sisi bagian depan *underwater* ROV, digunakan untuk melihat objek-objek tertentu di dalam air, dimana selanjutnya akan ditampilkan pada *ground station* sehingga *user* dapat melihat kondisi bawah air melalui GUI (*Graphical User Interface*) yang akan dibuat nantinya.



Gambar 3.4 Rancang Bangun Mekanik *Underwater* ROV

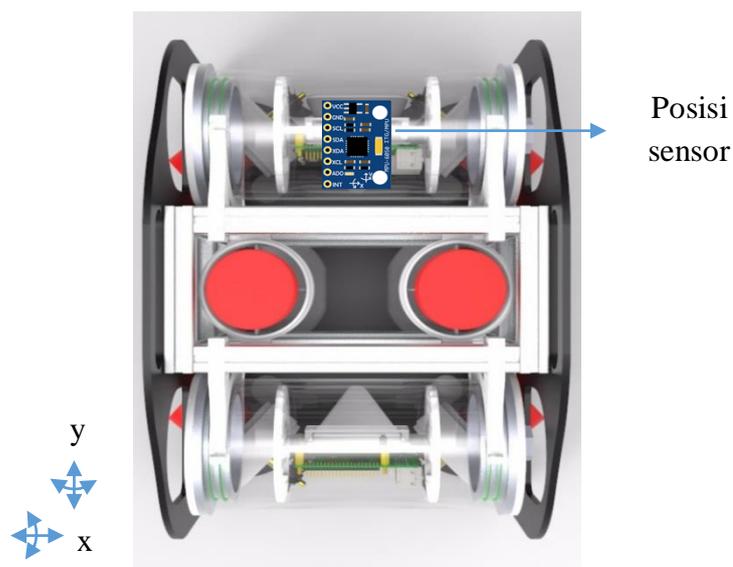
3.2.2 Penggunaan Sensor IMU MPU6050

Penggunaan sensor MPU6050 merupakan hal yang penting, dikarenakan sensor ini digunakan untuk membaca sudut posisi dari robot. Posisi peletakan sensor sangat penting untuk mendapatkan hasil pembacaan yang akurat. Dalam hal ini sensor sangat berpengaruh dalam menyeimbangkan dan mempercepat waktu tanggap terhadap perubahan sudut.

Sensor IMU akan diletakkan tepat di tengah-tengah bagian dari *underwater* ROV yang merupakan titik pusat koordinat dari sumbu posisi (x,y,z) dan orientasi (*roll, pitch, yaw*) sehingga dalam pengendalian posisi ROV memiliki keakuratan yang cukup tinggi dan tercapainya kestabilan ROV di dalam air.

Sensor berada pada titik *Center Of Gravity* (COG) dari *underwater* ROV, dapat dilihat pada Gambar 3.5 merupakan ilustrasi posisi peletakan sensor pada *underwater* ROV. Getaran yang dihasilkan dari pergerakan motor DC diyakini dapat mengganggu sensor, sehingga pada sensor diberi karet redaman untuk meminimalisasikan kesalahan.

Pembacaan sudut oleh sensor MPU6050 yang terdiri dari *gyro* dan *accelerometer* dapat diperoleh melalui perhitungan dari tiap sensor, dan memungkinkan untuk menggabungkan kedua sensor untuk menghasilkan pembacaan sudut.

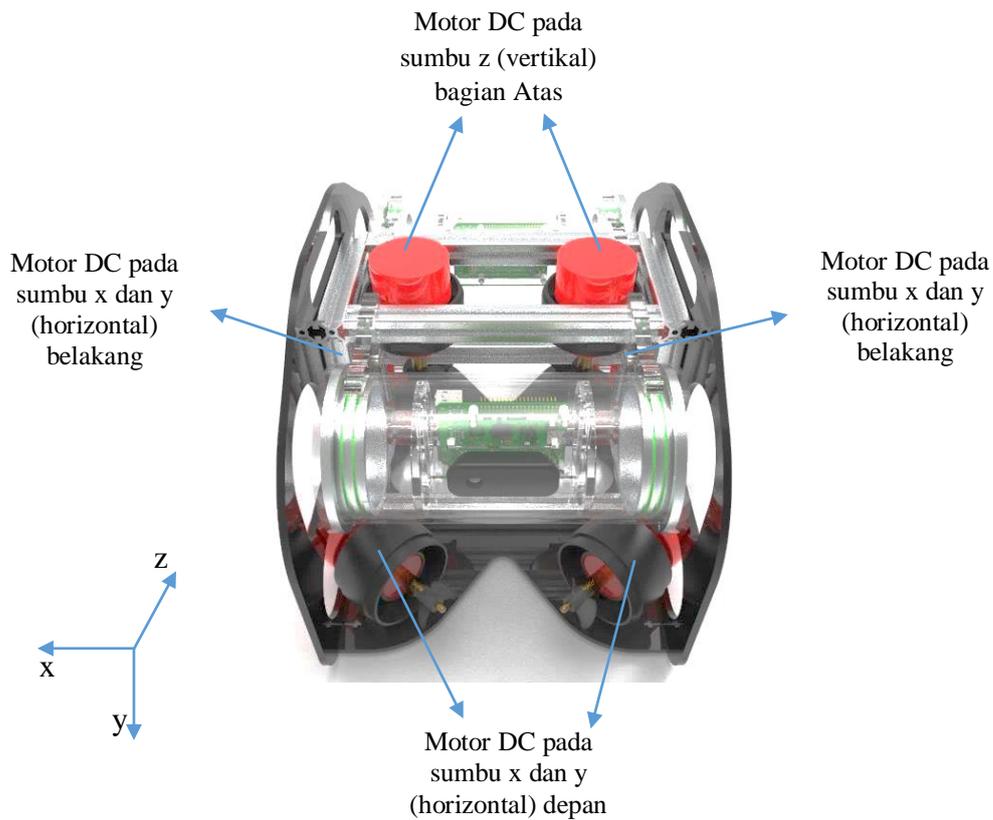


Gambar 3.5 Posisi Peletakan Sensor IMU MPU6050

3.2.3 Perancangan Posisi Motor DC

Perancangan peletakan posisi motor DC ini sangat penting, khususnya pada motor DC yang akan diletakkan secara vertikal sebagai *thruster* (pendorong) untuk mengendalikan tingkat kedalaman air yang akan digunakan pada penelitian ini. Posisi 4 buah motor DC akan diletakkan pada tiap sudut sisi bagian atas dan 2 pada tiap sisi samping (kanan dan kiri) dari *underwater* ROV, posisi dan arah motor DC tersebut dapat dilihat pada Gambar 3.6.

Besar gaya dorong yang dihasilkan oleh motor DC yang dipasang secara vertikal dapat memberikan gerak ke atas dan ke bawah, sehingga *underwater* ROV dapat menstabilkan posisi di kedalaman air pada tingkat kedalaman tertentu atau pada saat melakukan *hold position*. Motor DC yang dipasang secara horizontal menghasilkan gerak maju, mundur, putar kanan dan putar kiri.



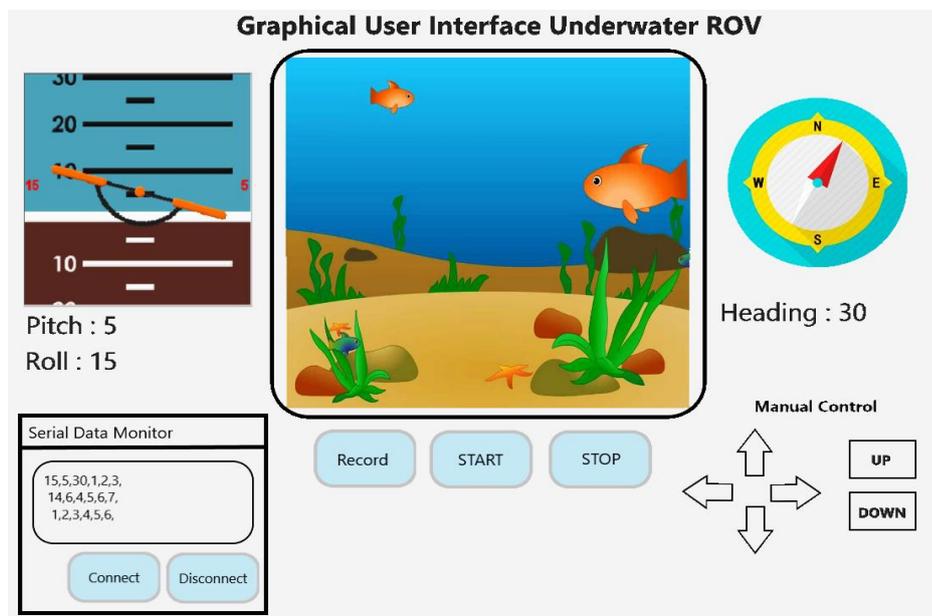
Gambar 3.6 Posisi Peletakan Motor DC Pada *Underwater* ROV

3.2.4 Rancang Bangun *Graphical User Interface* (GUI)

Robot *underwater* ROV bekerja di kedalaman air, sehingga pengguna (*user*) yang berada di darat (permukaan) membutuhkan informasi dan kondisi secara langsung (*real time*) *underwater* ROV tersebut serta keadaan lingkungan sekitar *underwater* ROV pada saat berada di dalam air.

Dibutuhkan antarmuka berupa perangkat lunak yang dapat menghubungkan antara pengguna terhadap *underwater* ROV. Pada Gambar 3.7, merupakan rancang bangun desain GUI yang memiliki beberapa fitur, diantaranya dapat menampilkan gambar dari kamera yang ada di *underwater* ROV secara langsung, fitur pengendali gerak untuk bermanuver di dalam air, dan menampilkan informasi data hasil pembacaan sensor tingkat kedalaman, sensor kompas serta gyro dan accelerometer sensor.

Rancang bangun GUI menggunakan perangkat lunak VB.Net 2010 yang nantinya akan terhubung ke *underwater* ROV melalui komunikasi serial. Hasil paket data yang dikirim oleh *underwater* ROV diolah menjadi tampilan yang menarik dan mudah dibaca oleh pengguna. Pengaturan sitem kendali *underwater* ROV juga dapat dilakukan melalui GUI ini.



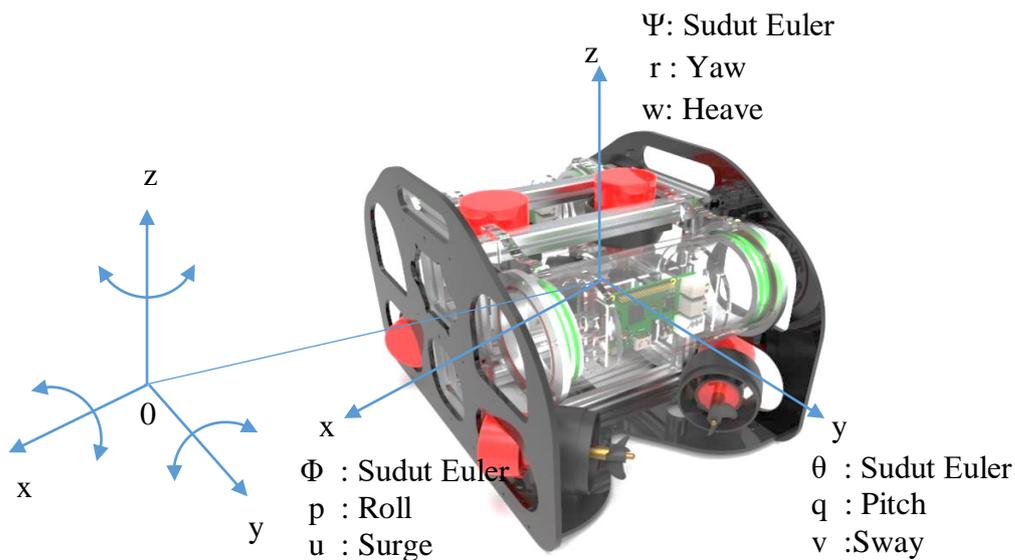
Gambar 3.7 Rancang Bangun Desain *Grapichal User Interface* (GUI)

3.3 Rancang Bangun Gerak *Underwater* ROV

Pada dasarnya *underwater* ROV dapat bergerak atau bermanuver disebabkan oleh gaya atau dorongan yang dihasilkan oleh putaran baling-baling yang digerakkan motor DC. Karena ada gaya tersebut, maka sistem kesetimbangan benda tidak lagi berlaku. *Underwater* ROV memiliki 6 derajat kebebasan (6 DOF), 3 gerak translasi dan 3 gerak rotasi.

Dapat dilihat pada Gambar 3.8 bahwa terdapat 3 sumbu kerja (x,y,z) yang merupakan arah dari pergerakan *underwater* ROV. Pada tiap-tiap sumbu memiliki pergerakan translasi dan rotasi, di antaranya:

1. Pada sumbu x terdapat
 - *Surge* : arah gerak *underwater* ROV ke samping searah sumbu x.
 - *Roll* : merupakan gerak rotasi terhadap sumbu x.
2. Pada sumbu y terdapat
 - *Sway* : arah gerak *underwater* ROV maju dan mundur searah sumbu y.
 - *Pitch* : merupakan gerak rotasi terhadap sumbu y.
3. Pada sumbu z terdapat
 - *Heave* : arah gerak *underwater* ROV keatas atau bawah searah sumbu y.
 - *Yaw* : merupakan gerak rotasi terhadap sumbu z.



Gambar 3.8 Arah Gerak *Underwater* ROV

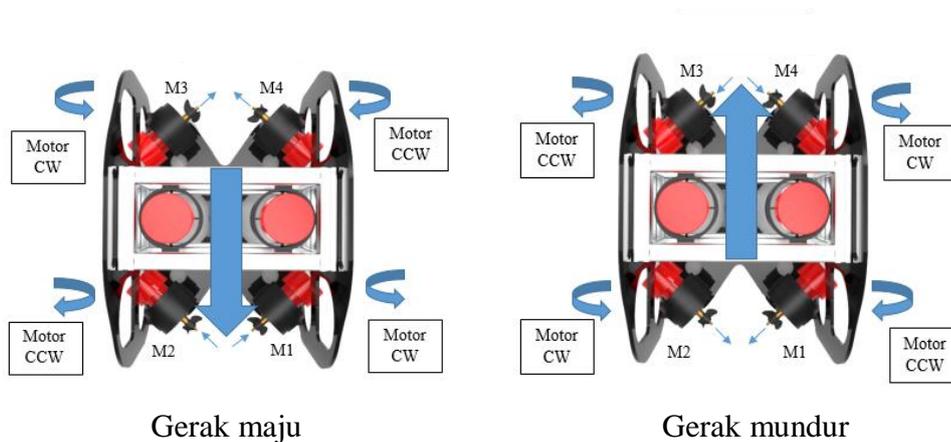
Berdasarkan posisi motor terhadap sumbu kerja, maka dapat dirancang pola pergerakan *underwater* ROV untuk dapat bermanuver di dalam air. Pergerakan ini juga nantinya tergantung pada bentuk bilah dari baling-baling yang digunakan. Berikut beberapa pergerakan yang dapat dilakukan, di antaranya:

1. Gerak maju dan mundur

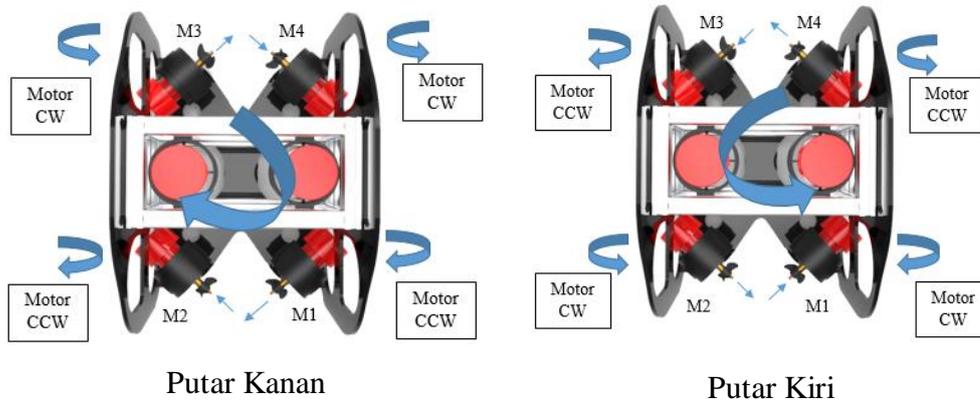
Gerak ini memanfaatkan pergerakan 4 buah motor horizontal pada sumbu x dan y, gaya dorong yang dihasilkan motor dari perputaran baling-baling saling berlawanan sehingga dapat menyebabkan pergerakan maju dan mundur. Ilustrasi dari pergerakan arah putar motor DC untuk melakukan pergerakan maju dan mundur ditunjukkan pada Gambar 3.9.

2. Gerak memutar ke kanan dan ke kiri

Gerak ini memanfaatkan pergerakan 4 buah motor horizontal pada sumbu x dan y seperti pada gerak maju dan mundur, akan tetapi gaya dorong yang dihasilkan motor dari perputaran baling-baling searah jarum jam atau sebaliknya sehingga dapat menyebabkan pergerakan memutar ke kanan dan ke kiri. Ilustrasi dari pergerakan arah putar motor DC untuk melakukan pergerakan putar kanan dan putar kiri ditunjukkan pada Gambar 3.10.



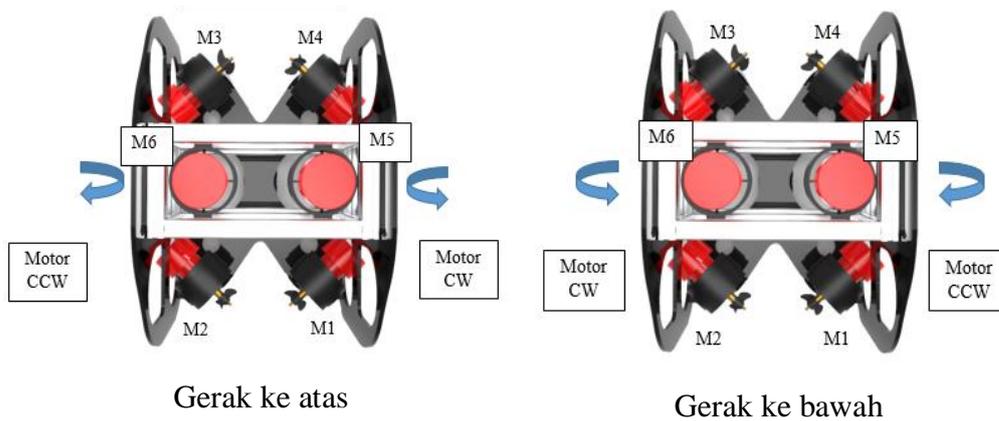
Gambar 3.9 Arah Putar Motor DC Untuk Gerak Maju dan Mundur



Gambar 3.10 Arah Putar Motor DC Untuk Gerak Putar Kanan dan Kiri

3. Gerak Atas dan Bawah

Gerak ini memanfaatkan pergerakan 2 buah motor vertikal pada sumbu z, gaya dorong yang dihasilkan motor dari perputaran baling-baling dapat menyebabkan pergerakan ke atas dan ke bawah. Ilustrasi dari pergerakan arah putar motor DC untuk melakukan pergerakan ke atas dan ke bawah ditunjukkan pada Gambar 3.11.



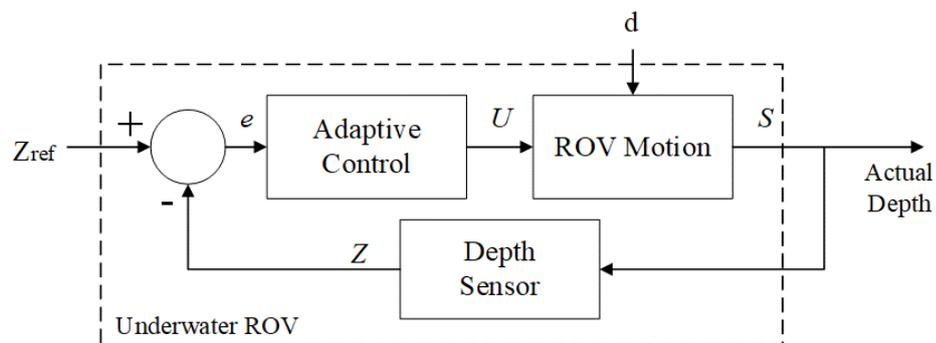
Gambar 3.11 Arah Putar Motor DC Untuk Gerak Atas dan Bawah

3.4 Perancangan Sistem Kendali *Hold Position*

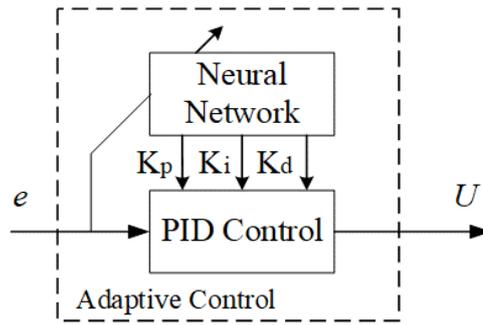
Pada tahap ini dirancang sistem kendali *neural network self-tuning* PID (NN-PID) yang akan digunakan untuk *hold position* pada tingkat kedalaman air secara vertikal terhadap sumbu z. Perancangan blok diagram dari keseluruhan sistem kendali NN-PID yang nantinya akan digunakan untuk sistem *hold position* pada tingkat kedalaman air oleh *underwater* ROV dapat dilihat pada Gambar 3.12. Parameter masukkan sebagai *set point* berupa tingkat kedalaman dari posisi *underwater* ROV secara vertikal terhadap sumbu z, sedangkan keluaran yang diharapkan dari perancangan sistem ini agar dapat menjaga *hold position* *underwater* ROV tersebut pada tingkat kedalaman air.

Skema dari sistem kendali adaptif pada penelitian ini digunakan sistem *tuning* parameter PID yang ditentukan oleh sistem *neural network*, tipe ini merupakan salah satu skema kendali adaptif dimana performasi keluaran sistem (proses) mengikuti performasi keluaran (*output*). Parameter kontrol diatur melalui mekanisme pengaturan yang didasarkan pada *error*, *integral error* dan *diferensial error* yang merupakan perhitungan nilai antara keluaran proses dengan keluaran model referensinya.

Blok diagram pada Gambar 3.12 terdapat bagian dari blok *underwater* ROV yang terdiri dari sistem kendali adaptif dan kendali gerak ROV, *depth* sensor digunakan sebagai *feedback* dari hasil pembacaan posisi tingkat kedalaman *underwater* ROV secara aktual didalam air.



Gambar 3.12 Diagram Blok Sistem *Hold Position Underwater* ROV



Gambar 3.13 Diagram Blok Kendali Adaptif NN-PID

Sistem kendali adaptif dibangun dari sistem *neural network* dan sistem kendali PID. Pada Gambar 3.13 merupakan blok diagram dari sistem kendali adaptif NN-PID yang telah di rancang. Sistem neural network digunakan sebagai penala tiap parameter dari sistem kendali PID.

Pada Persamaan 3.10 merupakan perhitungan besar nilai selisih antara *set point* terhadap nilai aktual,

$$e(t) = Z_{ref}(t) - Z(t) \quad (3.10)$$

dimana $e(t)$ adalah nilai *error*, $Z_{ref}(t)$ adalah nilai *set point* kedalaman, dan $Z(t)$ adalah kedalaman air secara aktual. Selanjutnya setelah mendapatkan hasil *error* maka dihitung besar nilai integral eror menggunakan Persamaan 3.11 – 3.14,

$$x_2(t) = \int e(t)dt \quad (3.11)$$

$$\frac{dx_2(t)}{dt} = e(t) \quad (3.12)$$

$$x_2(t) - x_2(t - 1) = e(t)dt \quad (3.13)$$

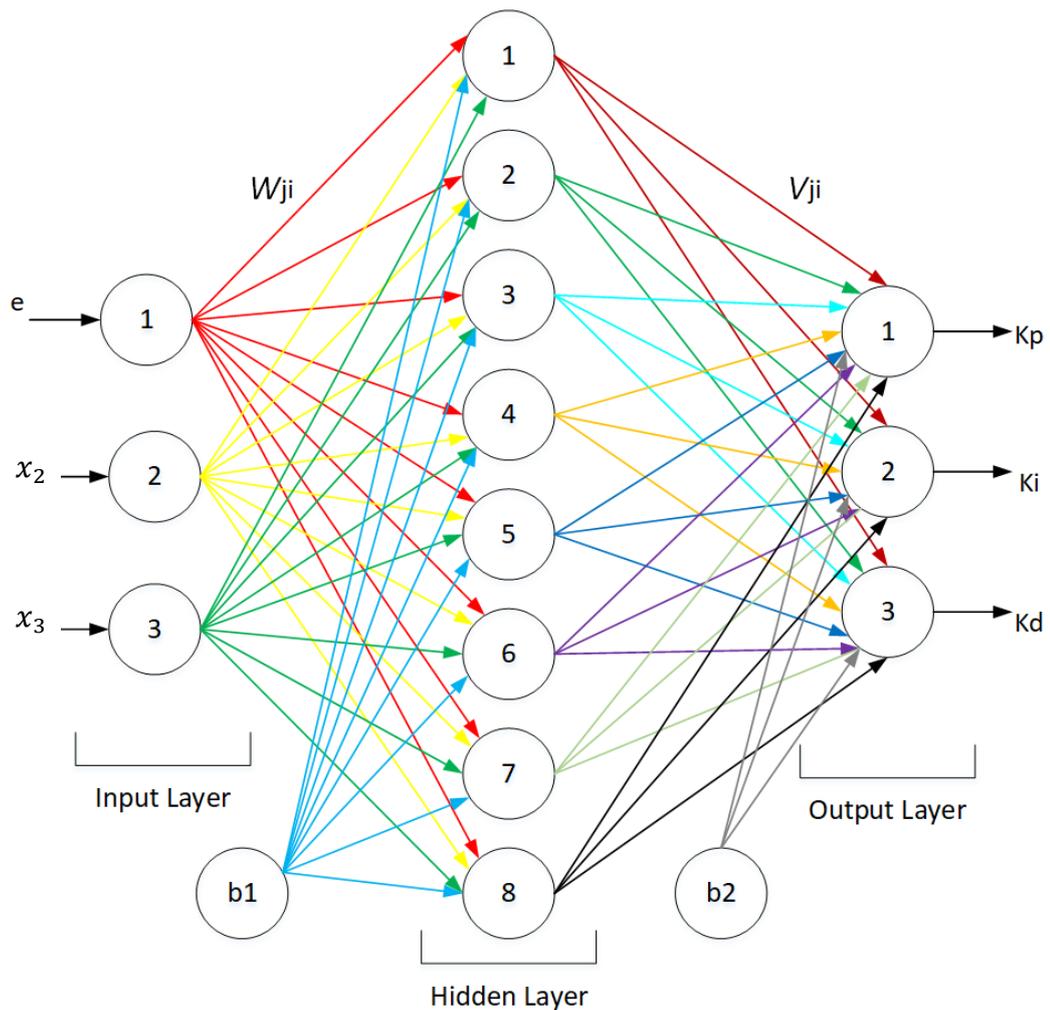
$$x_2(t) = x_2(t - 1) + e(t)dt \quad (3.14)$$

dimana x_2 adalah integral error, t adalah waktu, dan e adalah nilai *error*. Persamaan 3.15 dan Persamaan 3.16 digunakan untuk mencari nilai diferensial eror,

$$x_3(t) = de(t) \quad (3.15)$$

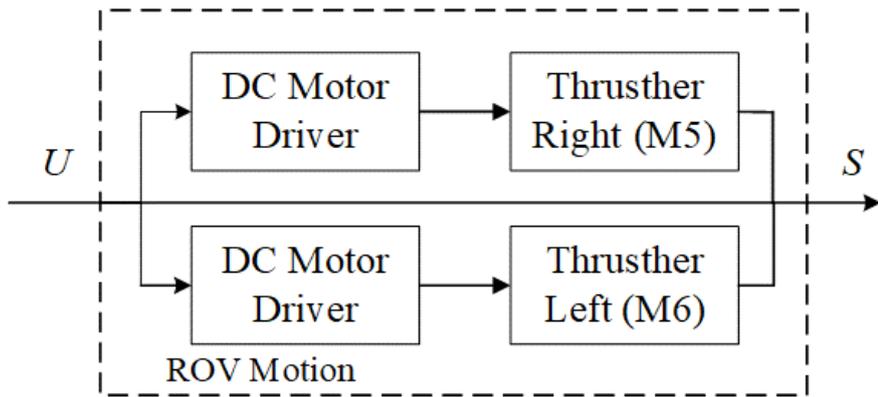
$$x_3(t) = e(t) - e(t - 1) \quad (3.16)$$

dimana x_3 adalah diferensial eror, e adalah *error* dan t adalah waktu.



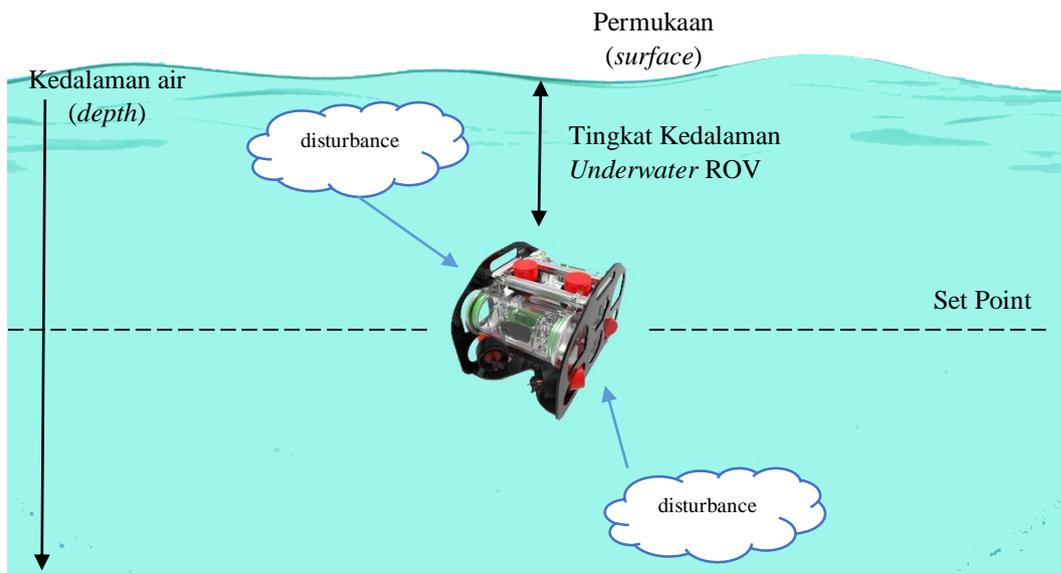
Gambar 3.14 Diagram Blok *Neural Network*

Pada Gambar 3.14, merupakan blok diagram perancangan *neural network* yang terdiri dari 3 layer (*input layer*, *hidden layer* dan *output layer*). Adapun pada *input* layer memiliki 3 neuron masukkan yang terdiri dari error, integral error dan differensial error, pada *hidden* layer memiliki 8 neuron yang telah dikalkulasikan dengan besar nilai bobot 1 (b_1), dan *output* layer terdiri dari 3 neuron (K_p , K_i , dan K_d) yang akan digunakan oleh parameter kendali PID setelah dikalkulsikan terhadap besar nilai pada bobot 2 (b_2).



Gambar 3.15 Diagram Blok Kendali Gerak *Underwater* ROV

Blok diagram dari sistem kendali gerak *underwater* ROV untuk dapat melakukan aksi didalam air dalam melakukan *hold position* dapat dilihat pada Gambar 3.15. Masukkan didapat dari hasil keluaran sistem kendali adaptif NN-PID untuk mengendalikan *driver* motor dc, dimana gerak motor pendorong (*thruster*) ditentukan arah dan kecepatannya. Pergerakan atau aksi yang dihasilkan untuk gerak ke atas dan ke bawah dalam mempertahankan posisi secara vertikal terhadap sumbu z.

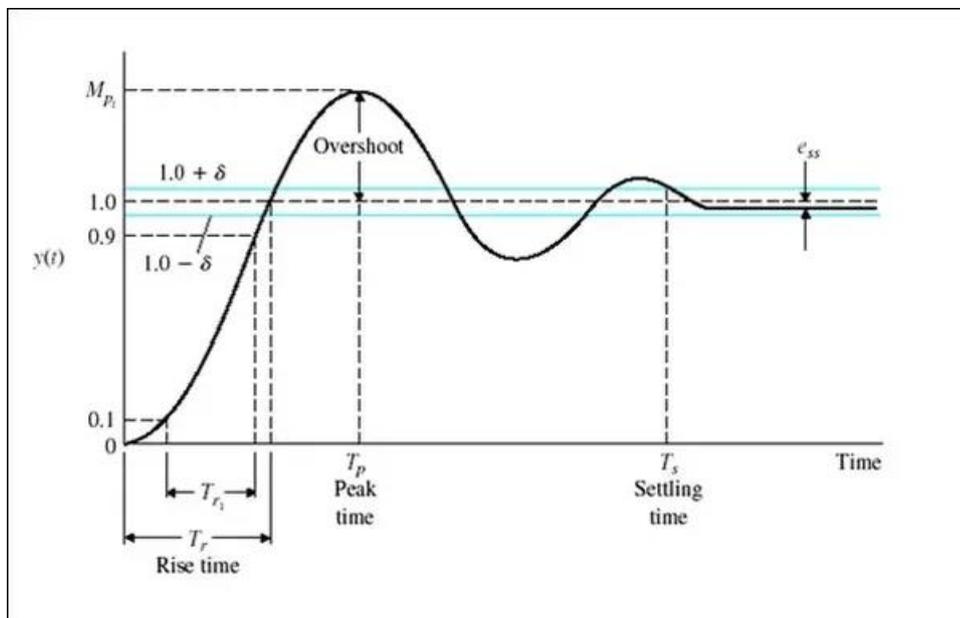


Gambar 3.16 Ilustrasi Sistem *Hold Position* Robot *Underwater* ROV

Pada Gambar 3.16 merupakan ilustrasi dari sistem *hold position* robot *underwater* ROV pada saat berada di dalam air. *Underwater* ROV mempertahankan posisi pada nilai kedalaman set point yang terukur dari permukaan (*surface*). *Disturbance* (gangguan) yang berada disekitar *underwater* ROV memberikan pengaruh besar, untuk itu sistem kendali *hold position* harus dapat beradaptasi terhadap terjadinya perubahan *disturbance* yang tidak menentu dalam menjaga tingkat kedalaman *underwater* ROV secara vertikal.

3.5 Pengujian Hasil

Dalam penelitian ini akan dilakukan pengujian hasil data yang didapat pada saat percobaan. Pengujian dilakukan terhadap keluaran aksi dari sistem kendali yang telah dirancang dengan meninjau beberapa faktor, diantaranya faktor maksimum respon (*overshoot*), waktu naik (*rise time*), waktu tunak (*settling time*), *error steady state* (ESS), dan *root mean square error* (RSME).



Gambar 3.17 Sinyal Tanggapan Sistem Kendali

Pada Gambar 3.17, merupakan sinyal tanggapan dari sistem kendali pada umumnya. *Overshoot* merupakan nilai relatif yang menyatakan perbandingan harga maksimum respon yang melampaui harga *steady state* dibanding dengan nilai *steady state*. *Rise time* adalah ukuran waktu yang di ukur mulai tanggapan $t=0$ s/d tanggapan memotong sumbu *steady state* yang pertama. *Settling time* yaitu ukuran waktu yang menyatakan tanggapan telah masuk $\pm 10\%$ dari *steady state*. *Error steady state* merupakan kondisi dimana saat sistem mencapai kondisi stabilnya, sinyal respon akan berhenti pada nilai dikisaran $\pm 10\%$ dari target (*setpoint*). Persamaan 3.17 dan Persamaan 3.18 merupakan cara menghitung untuk mencari besar nilai *error* pada saat kondisi telah mencapai *steady state*,

$$e_{ss} = \lim_{t \rightarrow 0} e(t) = \lim_{s \rightarrow 0} sE(s) \quad (3.17)$$

$$e_{ss} = \lim_{s \rightarrow 0} \frac{sZ_{ref}}{1+U_s} \quad (3.18)$$

dimana e_{ss} adalah *error steady state*, Z_{ref} adalah nilai target, dan U_s adalah nilai keluaran dari sistem kendali.

Dalam meninjau keberhasilan suatu sistem kendali, maka dilakukan perhitungan nilai akar kesalahan kuadrat rata-rata (*root mean square error*) yang merupakan suatu ukuran kesalahan berdasarkan pada selisih antara dua buah nilai yang bersesuaian. Persamaan 3.19 merupakan perhitungan besar nilai RMSE,

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (P_i - O_i)^2}{n}} \quad (3.19)$$

dimana *RMSE* adalah *root mean square error*, P_i adalah nilai target (*predicted value*), O_i adalah nilai aktual (*observed value*), dan n adalah banyak data.

BAB 4

HASIL DAN PEMBAHASAN

Pada bab ini dibahas tentang pengujian dan pengambilan data robot *underwater* ROV dalam mempertahankan posisi (*hold position*) pada tingkat kedalaman air. Teknik pengujian dilakukan dengan menguji robot di dalam air dengan diberikan gangguan (*disturbance*) dari luar atau diciptakan arus buatan dengan menggunakan motor bilge pump berkekuatan 1100GPH, dioperasikan pada level tegangan 12,5 volt sehingga menghasilkan gaya dorong air yang dikeluarkan oleh pompa tersebut.

Pada Gambar 4.1 merupakan alat uji yang telah dibuat, dilengkapi dengan *driver* motor untuk mengendalikan kecepatan pompa dalam mendorong keluar air sehingga dapat mengatur besar gangguan yang dihasilkan, terdapat 4 tombol pengatur besar gangguan yang terdiri dari 25%, 50%, 75% dan 100% (*full speed*). *Underwater* ROV harus dapat mempertahankan dan beradaptasi terhadap gangguan tersebut pada tingkat kedalaman air yang telah ditentukan.



Gambar 4.1 Alat Uji Pembuat Gangguan (*disturbance*)

4.1 Keterbatasan Penelitian

Dalam Penelitian yang telah dilakukan, terdapat beberapa keterbatasan yang mempengaruhi kondisi dari hasil penelitian. Adapun beberapa keterbatasan tersebut adalah:

1. Penelitian ini difokuskan pada sistem kendali tingkat kedalaman air yang dapat beradaptasi terhadap gangguan (*disturbance*).
2. Kondisi *underwater* ROV hanya dapat mempertahankan posisi pada tingkat kedalaman air secara vertikal.
3. Sistem Kendali *Neural Network* digunakan sebagai penentu (*selector*) nilai untuk menyala (*tuning*) parameter *Proportional*, *Integral*, *Derivative* pada sistem kendali PID.

4.2 Hasil Perancangan *Underwater* ROV

Pada Gambar 4.2 merupakan foto hasil robot *Underwater* ROV yang telah dirancang atau dirakit dari desain rancang bangun yang ditunjukkan pada bab sebelumnya. Pengujian hasil ini menunjukkan bahwa desain rancang bangun setiap bagian robot *underwater* ROV memiliki kecocokan bentuk dan desain, dan hasil perakitan menunjukkan bahwa ukuran setiap bagian sudah sesuai dengan rancang bangun yang diinginkan.



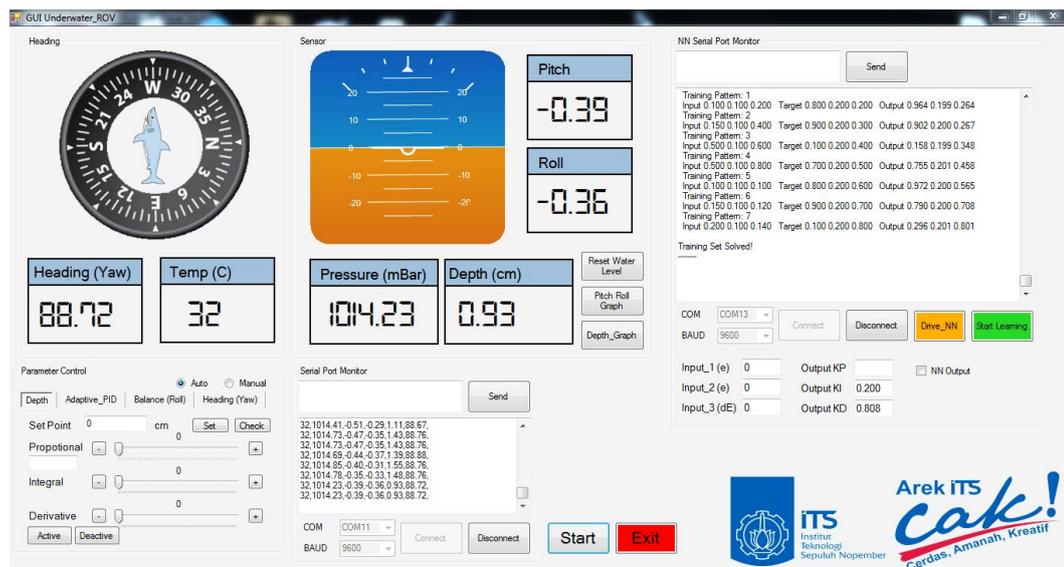
Gambar 4.2 Hasil Perakitan Robot *Underwater* ROV

Pengujian dari hasil perakitan ini dilakukan dengan menguji coba pergerakan robot *underwater* ROV pada saat navigasi di dalam air. Hasil dari uji coba tersebut menunjukkan robot *underwater* ROV dapat melakukan pergerakan ke atas, ke bawah, belok kanan, belok kiri, maju dan mundur dengan baik.

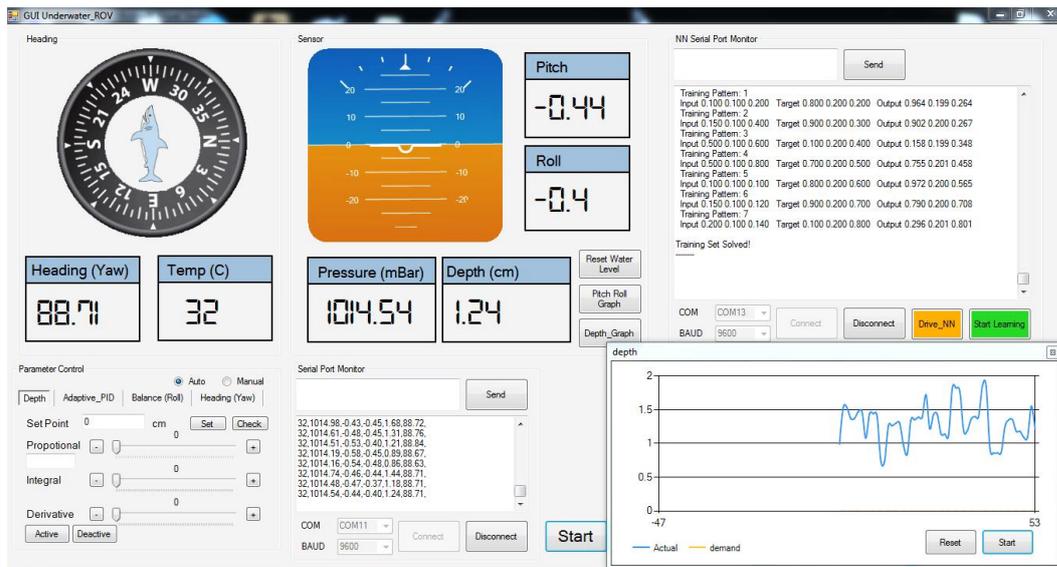
4.3 Pengujian GUI (*Graphical User Interface*)

GUI (*Graphical User Interface*) berperan penting dalam membaca dan memberikan informasi data kepada pengguna (*user*) serta mengendalikan robot *Underwater* ROV saat berada didalam air. Dilakukan pengujian terhadap GUI untuk memastikan semua fitur dapat bekerja dengan baik agar dapat menampilkan informasi dengan benar.

Pada Gambar 4.3 merupakan hasil pengujian komunikasi antara GUI dengan robot *underwater* ROV, GUI dapat menampilkan informasi data dengan baik berupa hasil pembacaan sensor-sensor pada robot saat berada didalam air dan juga mengirimkan data ke robot *underwater* ROV. GUI dilengkapi beberapa fitur, diantaranya dapat menampilkan data hasil pembacaan kedalaman bentuk grafik serta dapat menyimpan data tersebut ke dalam bentuk file berekstensi, hasil uji coba salah satu fitur tersebut dapat dilihat pada Gambar 4.4.



Gambar 4.3 GUI (*Graphical User Interface*) Comet-Underwater ROV



Gambar 4.4 GUI (*Graphical User Interface*) Menampilkan data grafik

4.4 Pengujian Sistem *Neural Network*

Penggunaan *Neural Network* diharapkan dapat memproses serta merespon data masukan dengan baik dan cepat agar memberikan data hasil keluaran yang mendekati nilai target yang telah ditentukan. Pada Gambar 4.5 merupakan hasil pengujian *Neural Network* dengan nilai target sebesar 0.02 memiliki nilai *training cycle* sebesar 7907 kali untuk mencapai nilai target, dimana jika nilai kesalahan (*error*) lebih kecil dari nilai target maka proses pelatihan dihentikan atau sudah berhasil mencapai target.

Pengujian dilakukan dengan memberikan nilai target kesalahan yang berbeda-beda. Pada Gambar 4.6 merupakan hasil dengan nilai target sebesar 0,05 mencapai nilai *training cycle* sebesar 2797 kali untuk mencapai nilai target. Dari hasil pengujian dengan nilai yang berbeda, menunjukkan bahwa sistem *neural network* yang telah dirancang dapat berjalan dengan baik dan dapat digunakan kedalam sistem *hold position* pada robot *underwater ROV*.

```

COM11
Training Pattern: 2
Input 0.150 0.100 0.400 Target 0.900 0.200 0.300 Output 0.862 0.202 0.291
Training Pattern: 3
Input 0.500 0.100 0.600 Target 0.100 0.200 0.400 Output 0.120 0.198 0.353
Training Pattern: 4
Input 0.500 0.100 0.800 Target 0.700 0.200 0.500 Output 0.740 0.201 0.463
Training Pattern: 5
Input 0.100 0.100 0.100 Target 0.800 0.200 0.600 Output 0.956 0.199 0.596
Training Pattern: 6
Input 0.150 0.100 0.120 Target 0.900 0.200 0.700 Output 0.819 0.200 0.651
Training Pattern: 7
Input 0.200 0.100 0.140 Target 0.100 0.200 0.800 Output 0.150 0.201 0.799 Output 0.150,0

TrainingCycle: 7907 Error = 0.020

Training Pattern: 0
Input 0.500 0.100 0.100 Target 0.700 0.200 0.100 Output 0.681 0.200 0.105
Training Pattern: 1
Input 0.100 0.100 0.200 Target 0.800 0.200 0.200 Output 0.850 0.198 0.178
Training Pattern: 2
Input 0.150 0.100 0.400 Target 0.900 0.200 0.300 Output 0.859 0.201 0.289
Training Pattern: 3
Input 0.500 0.100 0.600 Target 0.100 0.200 0.400 Output 0.107 0.198 0.373
Training Pattern: 4
Input 0.500 0.100 0.800 Target 0.700 0.200 0.500 Output 0.685 0.200 0.500
Training Pattern: 5
Input 0.100 0.100 0.100 Target 0.800 0.200 0.600 Output 0.954 0.199 0.590
Training Pattern: 6
Input 0.150 0.100 0.120 Target 0.900 0.200 0.700 Output 0.804 0.200 0.650
Training Pattern: 7
Input 0.200 0.100 0.140 Target 0.100 0.200 0.800 Output 0.131 0.200 0.802

Training Set Solved!
-----
Autoscroll Both NL & CR 9600 baud Clear output

```

Gambar 4.5 Pengujian *Neural Network* dengan target *error* 0.02

```

COM11
Training Pattern: 2
Input 0.150 0.100 0.400 Target 0.900 0.200 0.300 Output 1.000 0.200 0.259
Training Pattern: 3
Input 0.500 0.100 0.600 Target 0.100 0.200 0.400 Output 0.101 0.201 0.397
Training Pattern: 4
Input 0.500 0.100 0.800 Target 0.700 0.200 0.500 Output 0.716 0.199 0.506
Training Pattern: 5
Input 0.100 0.100 0.100 Target 0.800 0.200 0.600 Output 0.995 0.199 0.564
Training Pattern: 6
Input 0.150 0.100 0.120 Target 0.900 0.200 0.700 Output 0.862 0.200 0.696
Training Pattern: 7
Input 0.200 0.100 0.140 Target 0.100 0.200 0.800 Output 0.128 0.200 0.810 Output 0.128,0

TrainingCycle: 2797 Error = 0.050

Training Pattern: 0
Input 0.500 0.100 0.100 Target 0.700 0.200 0.100 Output 0.711 0.200 0.101
Training Pattern: 1
Input 0.100 0.100 0.200 Target 0.800 0.200 0.200 Output 0.999 0.201 0.278
Training Pattern: 2
Input 0.150 0.100 0.400 Target 0.900 0.200 0.300 Output 1.000 0.200 0.256
Training Pattern: 3
Input 0.500 0.100 0.600 Target 0.100 0.200 0.400 Output 0.102 0.201 0.388
Training Pattern: 4
Input 0.500 0.100 0.800 Target 0.700 0.200 0.500 Output 0.722 0.199 0.503
Training Pattern: 5
Input 0.100 0.100 0.100 Target 0.800 0.200 0.600 Output 0.996 0.199 0.575
Training Pattern: 6
Input 0.150 0.100 0.120 Target 0.900 0.200 0.700 Output 0.872 0.200 0.701
Training Pattern: 7
Input 0.200 0.100 0.140 Target 0.100 0.200 0.800 Output 0.133 0.200 0.811

Training Set Solved!
-----
Autoscroll Both NL & CR 9600 baud Clear output

```

Gambar 4.6 Pengujian *Neural Network* dengan target *error* 0.05

Tabel 4.1 Input dan Output *Neural Network* dengan Nilai Target *error* 0.02

Input NN			Output NN			Target Output		
Input 1 (<i>error</i>)	Input 2 (Integral <i>error</i>)	Input 3 (Derivatif <i>error</i>)	Output 1	Output 2	Output 3	Target 1	Target 2	Target 3
1,38	13,9	0,36	0,68	0,06	0,45	0,70	0,07	0,42
0,82	9,89	0,93	0,69	0,07	0,42	0,7	0,07	0,42
1,68	9,55	1,57	0,80	0,14	0,49	0,8	0,15	0,48
0,78	2,35	0,17	0,84	0,22	0,57	0,85	0,22	0,58
0,36	7,74	0,48	0,74	0,28	0,46	0,75	0,29	0,49
0,97	6,92	0,6	1,03	0,30	0,84	1,00	0,29	0,86
1,26	8,48	0,92	0,96	0,15	0,39	0,95	0,14	0,38
1,48	9,36	0,74	0,95	0,28	0,35	0,95	0,27	0,38

Setelah dilakukan pengujian terhadap pelatihan (*training*) pada sistem neural network, maka dilanjutkan pengujian dengan memberikan nilai masukan sehingga dapat dilihat apakah sistem dapat memberikan nilai keluaran sesuai dengan target keluaran yang diinginkan. Pada Tabel 4.1 merupakan hasil dari pengujian beberapa nilai masukan (*input NN*) yang diberikan dengan hasil nilai keluaran (*output NN*) dengan nilai target kesalahan sebesar 0.02.

4.5 Pengujian Sistem Kendali *Hold Position*

Dalam menguji sistem kendali *hold position* pada robot *underwater ROV*, maka dilakukan pengujian dengan kondisi gangguan (*disturbance*) dan kondisi ideal, adapun kondisi ideal dilakukan pada saat robot dapat beraksi tanpa unsur gangguan dari luar, sedangkan kondisi gangguan dilakukan dengan membuat arus buatan dengan media atau alat yang telah dijelaskan pada sub bab sebelumnya.

Pengujian dilakukan terhadap dua sistem kendali, kendali PID pada kondisi ideal dan kendali NN-PID terhadap gangguan. Kondisi gangguan dibagi dua keadaan, yaitu gangguan atas dan gangguan bawah. Kondisi gangguan atas merupakan arus gangguan yang diberikan dari posisi atas dari *underwater ROV*, sedangkan gangguan bawah diberikan dari posisi bawah *underwater ROV*.

4.5.1 Pengujian Sistem Kendali PID

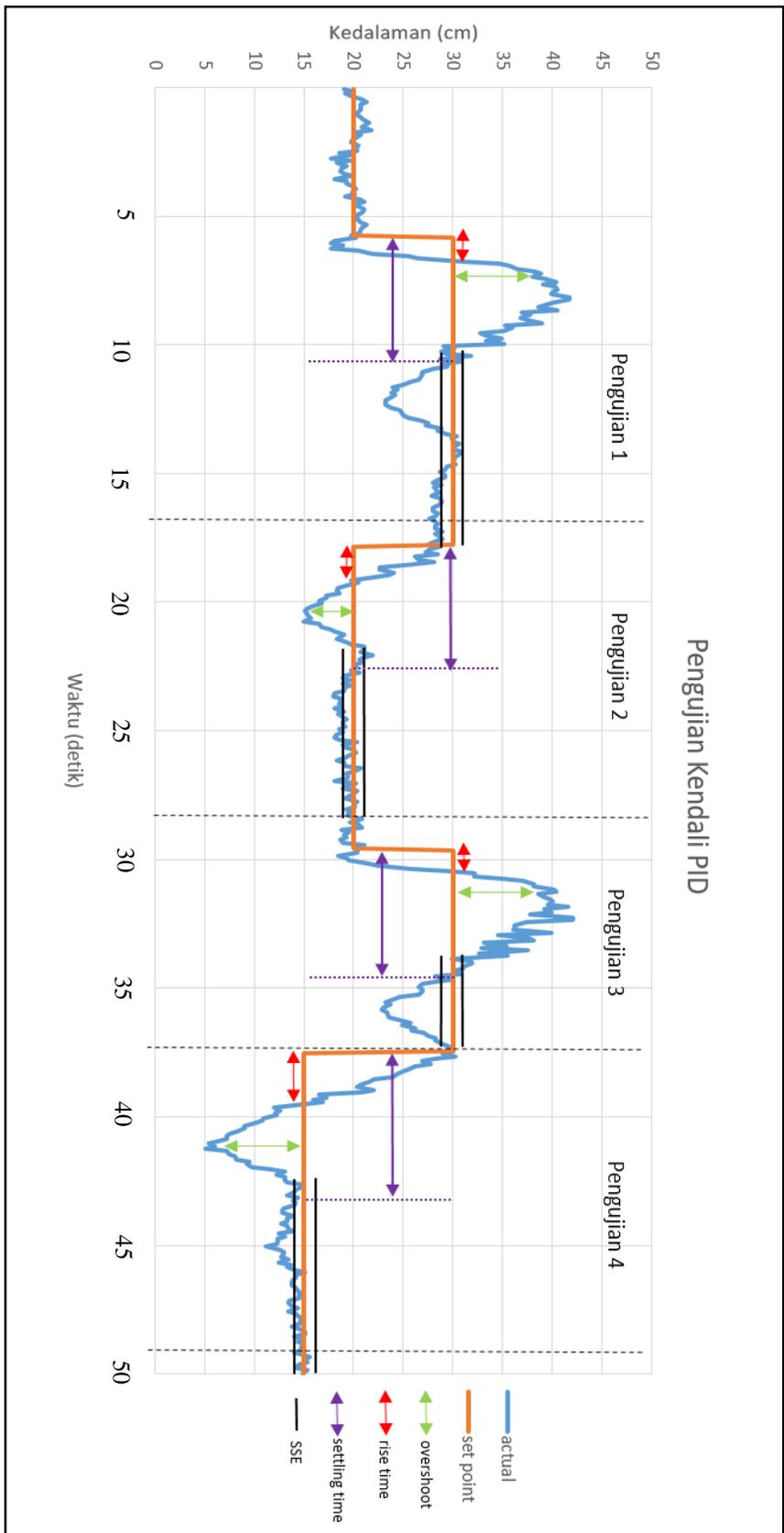
Pengujian sistem kendali PID dilakukan dengan menyalakan tiap parameter kendali PID secara manual (Konvensional). Besar nilai tiap parameter yang didapat saat pengujian yaitu $K_p = 60$, $K_i = 0,3$ dan $K_d = 0,1$. Penentuan nilai tiap parameter disesuaikan terhadap respon *underwater* ROV agar dapat menjaga nilai *set point* yang diinginkan.

Ada beberapa faktor yang ditinjau dari hasil pengujian pada sistem kendali ini, diantaranya besar nilai hasil terhadap *over shoot*, *rise time*, *settling time* dan *steady state error* (SSE). Dapat dilihat pada Tabel 4.2, merupakan hasil pengujian dari respon sistem kendali PID yang telah dilakukan dengan beberapa kali pengujian. Pada pengujian 1 dan pengujian 3 tidak mencapai kondisi SSE, dikarenakan *underwater* ROV mengalami kesulitan dalam merespon gerak, dimana hal ini ditunjukkan juga oleh nilai *over shoot* yang mencapai hingga 100%.

Pada Gambar 4.7 merupakan grafik hasil respon pengujian sistem kendali PID pada kondisi ideal (tanpa gangguan). Pengujian dilakukan sebanyak 4 kali dengan perubahan nilai setpoint untuk melihat respon dari sistem kendali dalam mempertahankan kondisi yang diinginkan. Hasil pengujian menunjukkan bahwa sistem kendali PID yang digunakan dalam mempertahankan kondisi terhadap nilai setpoint kurang baik, dilihat dari beberapa nilai faktor yang didapat dari proses pengujian yang telah dilakukan.

Tabel 4.2 Hasil Pengujian Kendali PID Konvensional

Pengujian	Over Shoot (%)	Rise Time (detik)	Settling Time (detik)	e_{ss} (%)	RMSE (%)
Pengujian 1	100%	1,4	8,7	6,50	4,70
Pengujian 2	55 %	1,7	5,0	4,35	2,70
Pengujian 3	100%	1,2	5,3	-	5,95
Pengujian 4	35%	2,0	3,9	7,30	5,41



Gambar 4.7 Pengujian Kendali PID Pada Kondisi Ideal

4.5.2 Pengujian Sistem Kendali NN-PID

Sistem kendali NN-PID merupakan sistem kontrol utama pada robot *underwater* ROV dalam melakukan *hold position*. Pada sistem kendali ini dilakukan pengujian dengan *tunning* tiap parameter PID secara manual yang bertujuan agar dapat bertahan terhadap gangguan yang berubah-ubah dengan besaran gangguan yang telah ditentukan sebelumnya.

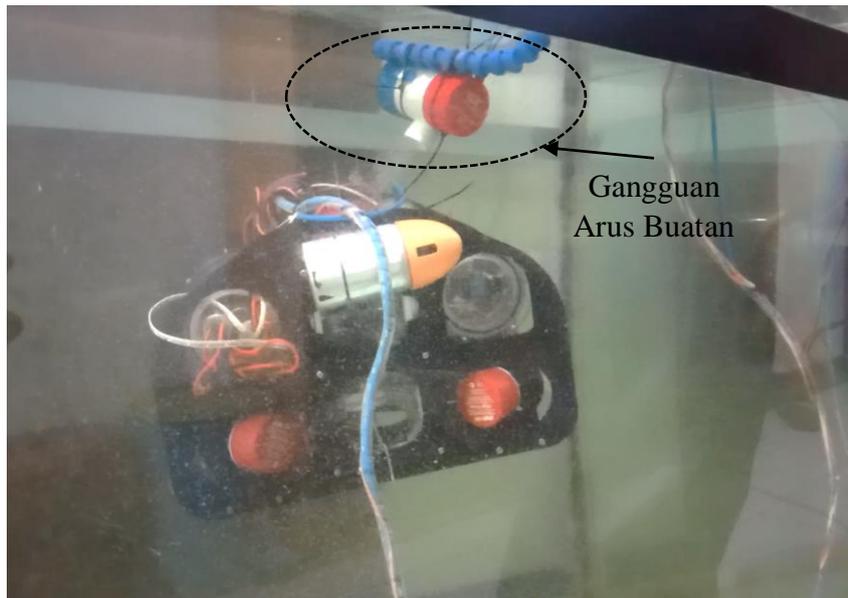
Data pada Tabel 4.3 dan Tabel 4.4 merupakan hasil *tunning* parameter PID yang didapat saat uji coba dengan berbagai kondisi gangguan atas dan bawah, dimana tiap kondisi memiliki nilai yang berbeda agar robot *underwater* ROV dapat selalu mempertahankan kondisi terhadap nilai *set point* yang diberikan.

Tabel 4.3 Parameter Keluaran PID Terhadap Gangguan Atas

Kondisi Saat Pengujian	Parameter PID		
	Kp	Ki	Kd
Ideal	60	0,3	0,1
Gangguan 25%	70	0,7	4,2
Gangguan 50%	70	0,7	4,2
Gangguan 75%	80	1,5	4,8
Gangguan 100%	85	2,2	5,8

Tabel 4.4 Parameter Keluaran PID Terhadap Gangguan Bawah

Kondisi Saat Pengujian	Parameter PID		
	Kp	Ki	Kd
Ideal	60	0,3	0,1
Gangguan 25%	75	2,9	4,9
Gangguan 50%	100	2,9	8,6
Gangguan 75%	95	1,4	3,8
Gangguan 100%	95	2,7	3,8



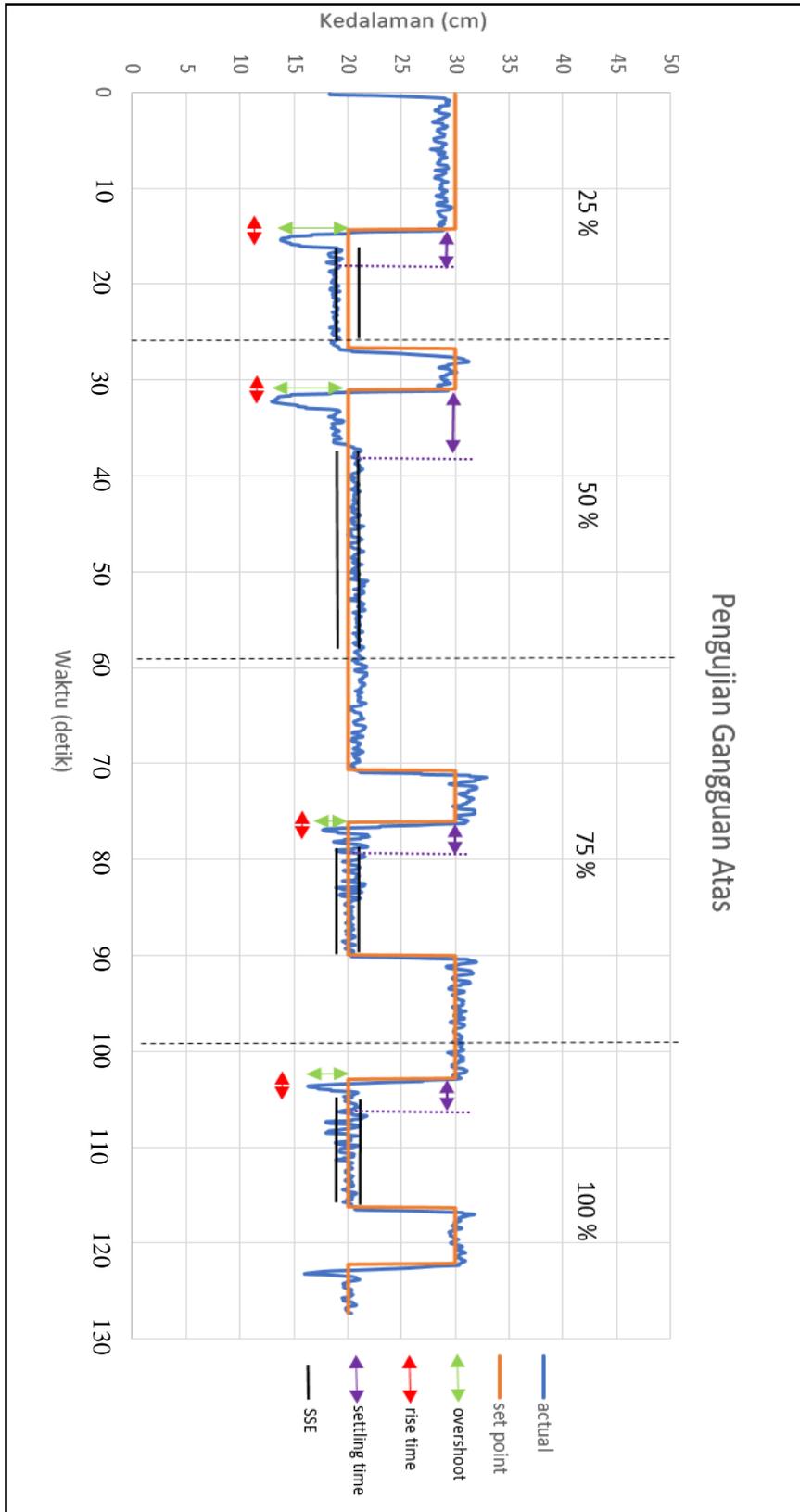
Gambar 4.8 Pengujian *Underwater ROV* Terhadap Gangguan Arus Atas

Pengujian dilakukan dengan memberikan gangguan dengan arus buatan yang menimbulkan efek gaya dorong terhadap *underwater ROV*, dapat dilihat pada Gambar 4.8 merupakan cara pengujian dengan menimbulkan gangguan arus dari bagian atas *underwater ROV*, besar gangguan yang ditimbulkan oleh pompa memiliki variasi nilai keluaran.

Beberapa faktor yang ditinjau dari hasil pengujian sistem kendali NN-PID sama seperti pada pengujian sistem kendali PID. Pada Tabel 4.5 merupakan hasil pengujian sistem kendali NN-PID terhadap gangguan atas, adapun hasil grafik respon pengujian *underwater ROV* terhadap gangguan atas dapat dilihat pada Gambar 4.9.

Tabel 4.5 Hasil Pengujian NN-PID Terhadap Gangguan Atas

Pengujian	Over Shoot (%)	Rise Time (detik)	Settling Time (detik)	e_{ss} (%)	RMSE (%)
Gangguan 25%	10%	1,5	3,2	5,65	2,72
Gangguan 50%	20%	0,7	6,9	4,30	2,11
Gangguan 75%	20%	0,8	2,4	6,50	1,82
Gangguan 100%	30%	0,7	2,3	4,85	1,83



Gambar 4.9 Pengujian Terhadap Gangguan Atas



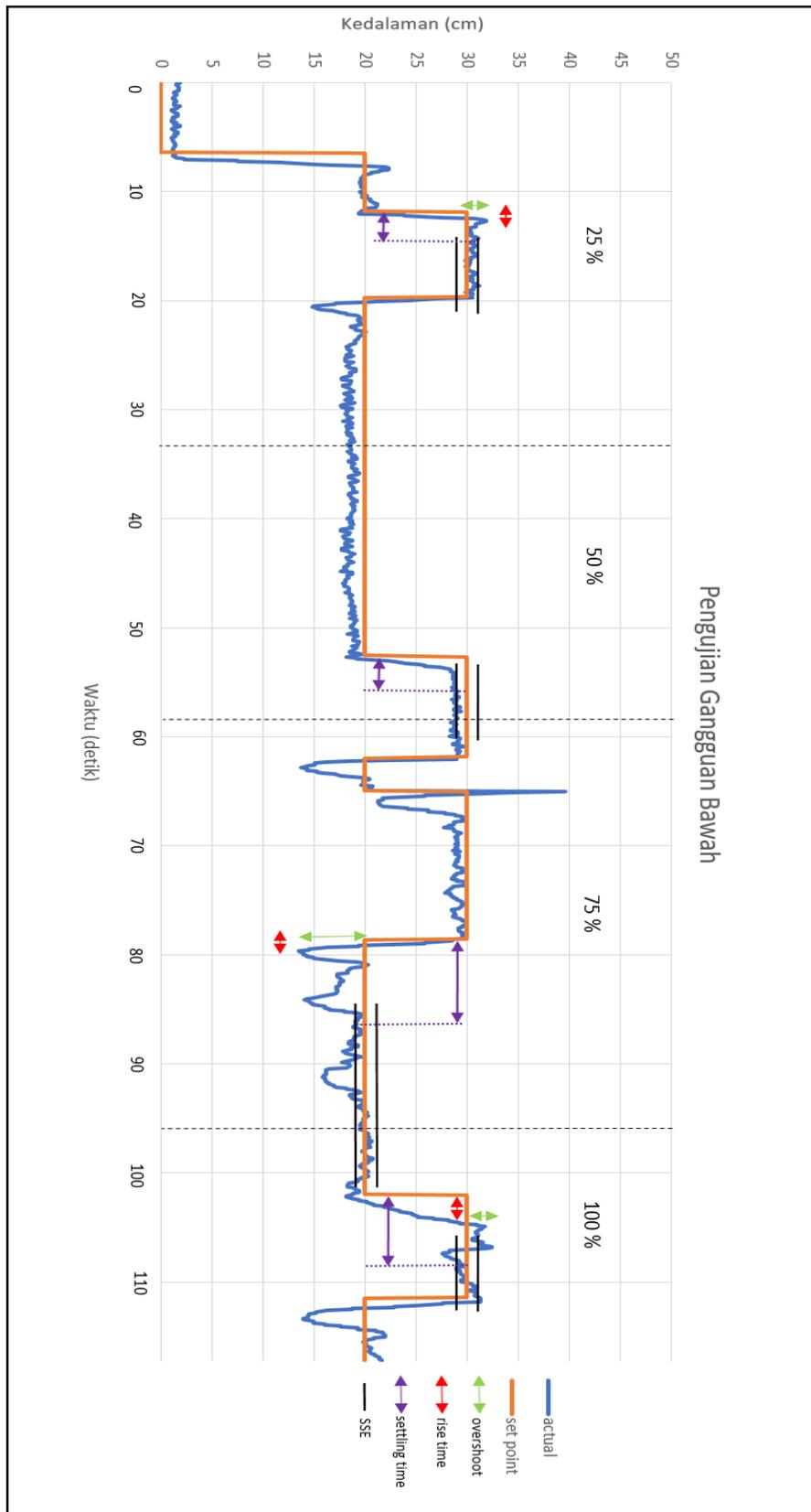
Gambar 4.10 Pengujian *Underwater* ROV Terhadap Gangguan Arus Bawah

Pengujian juga dilakukan dengan memberikan gangguan arus bawah, dimana gangguan ini diberikan dari sisi bagian bawah *underwater* ROV dengan besaran nilai gangguan yang telah ditentukan. Pada Gambar 4.10 merupakan ilustrasi pengujian *underwater* ROV dengan memberikan gangguan arus buatan dari sisi bagian bawah.

Pada Tabel 4.6 merupakan hasil pengujian sistem kendali NN-PID yang melakukan adaptasi terhadap gangguan bawah. Grafik data hasil pengujian NN-PID terhadap gangguan bawah dapat dilihat pada Gambar 4.11. Hasil pengujian menunjukkan bahwa sistem kendali NN-PID bekerja dengan baik, dilihat dari beberapa hasil faktor respon kendali yang didapat pada saat pengujian terhadap gangguan.

Tabel 4.6 Hasil Pengujian NN-PID Terhadap Gangguan Bawah

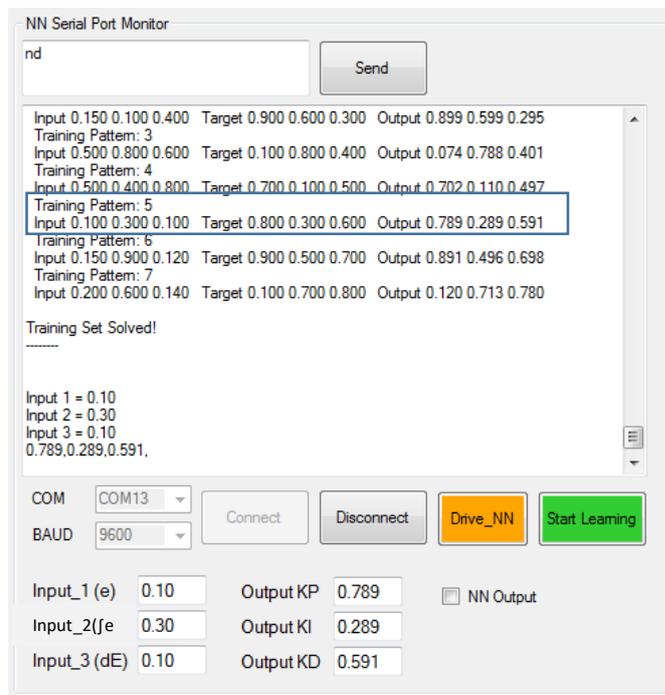
Pengujian	Over Shoot (%)	Rise Time (detik)	Settling Time (detik)	e_{ss} (%)	RMSE (%)
Gangguan 25%	20%	2,0	4,6	6,45	4,12
Gangguan 50%	-	0,8	3,5	-	2,07
Gangguan 75%	50%	0,6	3,2	4,55	2,95
Gangguan 100%	60%	0,2	1,9	5,30	3,35



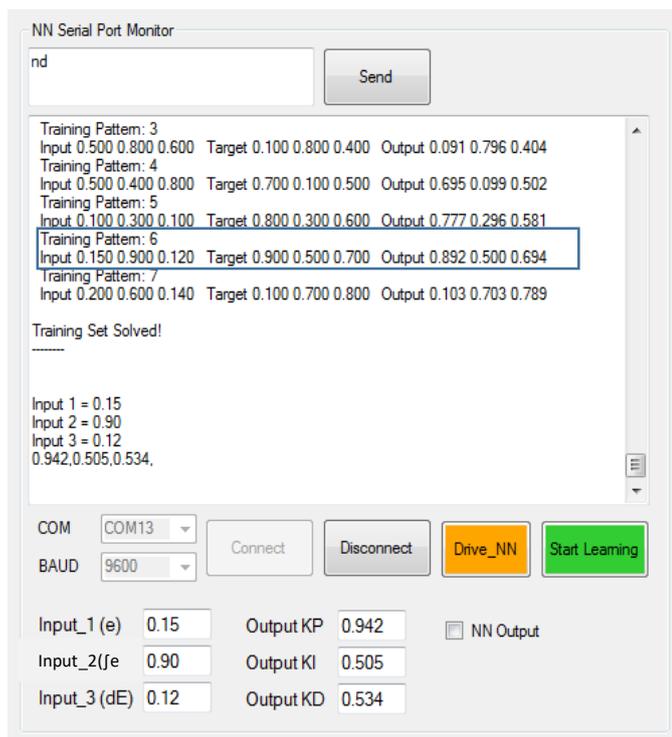
Gambar 4.11 Pengujian Terhadap Gangguan Bawah

Pengujian sistem kendali *neural network* dalam menentukan parameter PID sangat diperlukan, di mana parameter ini akan digunakan dalam sistem kendali PID untuk melakukan sistem *hold position underwater* ROV. Pada Gambar 4.12 merupakan hasil *training* dengan target eror 0.002 dengan memiliki nilai *training cycle* sebesar 1012, pengujian diberikan nilai masukan pada pattern ke-5 dengan input_1 = 0,10, input_2 = 0,30 dan input_3 = 0.10 yang didapat dari hasil eror, integral eror dan derivatif eror. Hasil keluaran nilai pada Output KP = 0,789, Output KI = 0,289 dan Output KD = 0,591 menunjukkan bahwa nilai-nilai tersebut sudah mendekati nilai target yang diinginkan, dengan tujuan *underwater* ROV dapat melakukan *hold position* sesuai dengan besar gangguan yang diberikan.

Pengujian dilakukan dengan memberikan besaran nilai masukan yang berbeda-beda, dengan tujuan agar hasil nilai keluaran sesuai dengan keluaran nilai target yang diinginkan. Pada Gambar 4.13 merupakan hasil dari masukan pada pattern ke-6 dengan nilai input_1 = 0,15, input_2 = 0,90 dan input_3 = 0.12 dengan besar nilai keluaran pada Output KP = 0,942, Output KI = 0,505 dan Output KD = 0,534. Hasil ini menunjukkan juga bahwa nilai tersebut mendekati nilai target yang diinginkan.



Gambar 4.12 Hasil Pengujian NN-PID Pada Pattern ke-5



Gambar 4.13 Hasil Pengujian NN-PID Pada Pattern ke-6

Halaman ini sengaja dikosongkan

BAB 5

PENUTUP

Berdasarkan dari hasil penelitian yang telah dilakukan, dengan judul “Sistem Kendali *Underwater Remotely Operated Vehicles* Pada Tingkat Kedalaman Air Menggunakan Metode Kendali Adaptif”, maka penulis dapat memberikan kesimpulan dan saran.

5.1 Kesimpulan

Pengujian sistem kendali *hold position* pada *underwater* ROV dilakukan terhadap sistem kendali PID secara konvensional dan sistem kendali adaptif NN-PID yang telah dirancang. Dari hasil pengujian sistem kendali PID tanpa adanya gangguan yang diberikan, dengan nilai parameter $K_p = 60$, $K_i = 0,3$ dan $K_d = 0,1$ menghasilkan nilai *overshoot* 100%, *rise time* 1,2 detik, *settling time* 5,3 detik, tidak mencapai *error steady state* (ESS), dan memiliki RMSE sebesar 5,95%.

Hasil pengujian sistem kendali adaptif NN-PID terhadap gangguan atas dengan besar gangguan yang beragam serta perubahan nilai set point, mendapatkan nilai terbaik saat pengujian dengan besar *overshoot* 10%, *rise time* 1,5 detik, *settling time* 3,2 detik, ESS 5,65%, dan RMSE sebesar 2,72%. Sistem kendali adaptif NN-PID terhadap gangguan bawah mendapat nilai terbaik saat pengujian dengan besar *overshoot* 20%, *rise time* 2,0 detik, *settling time* 4,6 detik, ESS 6,45%, dan RMSE sebesar 4,12%.

Dari hasil pengujian yang dilakukan terhadap kedua sistem kendali antara sistem kendali PID secara konvensional dan sistem kendali adaptif NN-PID, dapat diketahui bahwa kendali adaptif yang telah dirancang bekerja lebih baik dari pada sistem kendali secara konvensional, ditinjau dari besar nilai *overshoot*, *rise time*, *settling time* dan *SSE* saat pengujian dalam melakukan *hold position* oleh *underwater* ROV pada saat di dalam air dengan atau tanpa gangguan.

5.2 Saran

Pada penelitian yang telah dilakukan, terdapat masalah yaitu khususnya pada sistem kendali yang dapat mempelajari kondisi terhadap gangguan (*disturbance*) yang baru, sehingga robot *underwater* ROV dapat mengantisipasi dan memperbaiki sistem kendali *hold position*. Sistem *hold position* tidak hanya pada tingkat kedalaman air, akan tetapi dapat melakukan sistem *hold position* pada sumbu atau bidang kerja lainnya dari robot *underwater* ROV tersebut.

DAFTAR PUSTAKA

- [1] T. Wikipedia, "Daftar kecelakaan dan insiden pesawat penumpang di Indonesia," wikipedia.org, 11 april 2015. [Online]. Available: http://id.wikipedia.org/wiki/Daftar_kecelakaan_dan_insiden_pesawat_penumpang_di_Indonesia. [Accessed 10 juni 2015].
- [2] E. Y. Hong, H. G. Soon and M. Chitre, "Depth Control of an Autonomous Underwater," *OCEANS 2010 IEEE - Sydney*, pp. 1-6, 2010.
- [3] D. maalouf, V. Creuze and A. Chemori, "Real-Time Experimental Comparison of Two Depth Control Schemes for Underwater Vehicles," *International Journal of Advanced Robotic Systems*, no. 12, 2015.
- [4] Christ, R. D and R. L. W. Sr, *The ROV Manual : A User Guide for Observation-Class Remotely Operated Vehicle*, Elsevier Ltd. Oxford, 2007.
- [5] N. O. a. A. Administration, "Remotely Operated Vehicles (ROV)," 2010. [Online]. Available: <http://oceanexplorer.noaa.gov/technology/subs/rov.html>. [Accessed 10 juni 2015].
- [6] L. G. García-Valdovinos, T. Salgado-Jiménez, M. Bandala-Sánchez and e. all, "Modelling, Design and Robust Control of a Remotely Operated Underwater Vehicle," *International Journal of Advanced Robotic Systems*, no. 11, 2014.
- [7] B. Raeisy, A. A. Safavi and A. R. Khayatian, "Optimized Fuzzy Control Design Of An Autonomous Underwater Vehicle," *Iranian Journal Of Fuzzy Systems* , vol. 9, pp. 25-41, 2012.
- [8] V. Streeter, W. E and B. K, *Fluid Mechanics Boston: WCB*, 1998.
- [9] S. Beeby, G. Ensell, M. Kraft and N.White, *MEMS mechanical sensors*, USA: Artech house inc., 2004.
- [10] S. E. Lyshevski, *Mems and Nems: systems, devices and structures*, USA: CRC Press LLC., 2002.
- [11] B. E. Boser, "Electronics for micromachined inertial sensors," *Transducers Dig. of Tech*, pp. 1169-1172, Juni 1997.
- [12] AMSYS, "How to measure absolute," AMSYS GmbH & Co. KG, juli 2009. [Online]. Available: <http://www.amsys.info>. [Accessed 2 juli 2015].

- [13] R. Pallas-Areny and J. G. Webster, *Sensor and Signal Conditioning*, 2nd ed., New York: A Wiley-Interscience Publication, 2001.
- [14] M. Specialties, "Pressure Sensor - MS5803-14BA," *Measurement Specialties*, 2012.
- [15] J. DiStefano, A. Stubberud and I. Williams, *Schaum's Outline of Feedback and Control Systems*, 2nd Edition, McGraw-Hill, 2011.
- [16] N. S. Nise, *Control Systems Engineering, International Student Version*, 6th Edition, John Wiley & Sons, Inc, 2010.
- [17] C. T. Kilian, *Modern Control Technology: Components and Systems*, 2nd ed., Delmar Thomson Learning, 2001.
- [18] K. J. Astrom, *Adaptive Control*, 2nd ed., Addison-Wesley, 1995.
- [19] H. Yagi, J. Shimonishi, T. Yamamoto and T. Hinamoto, "Design of a performance-adaptive PID controller with a model prediction structure," *International Journal of Innovative Computing, Information and Control*, vol. 5, no. 10(A), pp. 2969-2983, 2009.
- [20] C.-H. Hsieh, W.-H. Ho and J.-H. Chou, "Optimal PID controllers design of PWM feedback timevarying systems using orthogonal-functions approach and genetic algorithm," *International Journal of Innovative Computing, Information and Control*, vol. 5, pp. 387-397, 2009.
- [21] N. Tague, *The quality toolbox*. (2th ed.), Milwaukee: Wisconsin: ASQ Quality Press, 2005.
- [22] F. Instruments, *Digital Pendulum: Control in a Matlab Environment*, Sussex, UK: Feedback Instruments Ltd., 2006.
- [23] K. T. a. M. Sugeno, "Stability analysis and design of fuzzy control," *Fuzzy Sets and Systems*, vol. 45, pp. 135-156, 1992.

LAMPIRAN

adaptive_PID.ino

```
/*
*****
* Hold Position and motion control for Underwater_ROV in water levels
* using pressure sensor MS5803 14B and IMU MPU6050
* alizainal_shahab@yahoo.com
*
* update 23 April 2018
* IDE : Arduino 1.6.8
* Hardware Platform : Arduino Mega2560
*****/

#include "Wire.h"
#include "SparkFun_MS5803_I2C.h"
#include "I2Cdev.h"
#include "EEPROM.h"
#include "MPU6050.h"
#include "KalmanFilter.h"
#include "PID_v1.h"
#include "RunningAverage.h"

#define cw 0
#define ccw 1
#define brake 2

#define Gry_offset 0
#define Gyr_Gain 0.00763358
RunningAverage myRA(10);
//motor_dc control
int inApin[6] = {22, 24, 26, 28, 30, 32}; // INA: Clockwise input
int inBpin[6] = {23, 25, 27, 29, 31, 33}; // INB: Counter-clockwise input
int pwmpin[6] = {2, 3, 5, 6, 7, 8}; // PWM input
//int enpin[2] = {0, 1}; // EN: Status of switches output (Analog pin)
//compass GY-271
double angle;
MS5803 sensor(ADDRESS_HIGH); //ADDRESS_HIGH = 0x76 ;
ADDRESS_LOW = 0x77

//MPU6050 config
MPU6050 mpu;

volatile float pitch = 0;
volatile float roll = 0;
volatile float acc_z = 0;
```

```

volatile float kalPitch = 0;
volatile float kalRoll = 0;
volatile float kalpres = 0;
KalmanFilter kalmanX(0.001, 0.003, 0.03);
KalmanFilter kalmanY(0.001, 0.003, 0.03);
KalmanFilter kalmandepth(0.001, 0.003, 0.03);
unsigned int suhu;
double pressure_abs;
double water_level;
double out_depth;
volatile double set_level,balance,head_set;
double reset_level = 0, nilai;
double avg;
double base_altitude = 1655.0;

//PID Library parameter
double Setpoint, Input, Output;
double de_Setpoint, de_Input, de_Output;
double head_Setpoint, head_Input, head_Output;
PID myPID(&Input, &Output, &Setpoint,6,2,1, DIRECT);
PID depth_PID(&de_Input, &de_Output, &de_Setpoint,6,2,1, DIRECT);
PID head_PID(&head_Input, &head_Output, &head_Setpoint,6,2,1, DIRECT);
double Speed_ka = 0, Speed_ki = 0;
float kp,ki,kd;
float depth_kp,depth_ki,depth_kd;
float adapt_kp,adapt_ki,adapt_kd;
float head_kp,head_ki,head_kd;

String inputString = "";
boolean mode = true;
boolean control = true;
boolean onoff = true;
boolean adaptive = true;

void setup() {
  // put your setup code here, to run once:

  Serial.begin(9600);
  Wire.begin();

  while(!mpu.begin(MPU6050_SCALE_2000DPS, MPU6050_RANGE_16G))
  {
    Serial.println("Could not find a valid MPU6050 sensor, check wiring!");
    delay(500);
  }
  for (int i=0; i<7; i++)

```

```

{
  pinMode(inApin[i], OUTPUT);
  pinMode(inBpin[i], OUTPUT);
  pinMode(pwmpin[i], OUTPUT);
}
// Initialize braked
for (int i=0; i<7; i++)
{
  digitalWrite(inApin[i], LOW);
  digitalWrite(inBpin[i], LOW);
}

mpu.setAccelPowerOnDelay(MPU6050_DELAY_3MS);
mpu.calibrateGyro();
mpu.setThreshold(0);
sensor.reset();
sensor.begin();
Setpoint = 0;
Input = kalPitch;
de_Setpoint = 0;
de_Input = pressure_abs;
head_Setpoint = 0;
head_Input = avg;
myPID.SetMode(AUTOMATIC);
myPID.SetOutputLimits(-255,255);
depth_PID.SetMode(AUTOMATIC);
depth_PID.SetOutputLimits(-255,255);
head_PID.SetMode(AUTOMATIC);
head_PID.SetOutputLimits(-255,255);

kp = EEPROM.read(1); ki = EEPROM.read(2)*0.1; kd = EEPROM.read(3)*0.1;
depth_kp = EEPROM.read(4); depth_ki = EEPROM.read(5)*0.1; depth_kd =
EEPROM.read(6)*0.1;
head_kp = EEPROM.read(15); head_ki = EEPROM.read(16)*0.1; head_kd =
EEPROM.read(17)*0.1;
adapt_kp = EEPROM.read(20); adapt_ki = EEPROM.read(21)*0.1; adapt_kd =
EEPROM.read(22)*0.1;
EEPROM_readAnything( 7, nilai);
reset_level = nilai;
Serial.println("OK_Ready");
Serial.print("Depth Kp = ");
Serial.print(EEPROM.read(4));
Serial.print(',');
Serial.print("Depth Ki = ");
Serial.print(EEPROM.read(5)*0.1);
Serial.print(',');
Serial.print("Depth Kd = ");

```

```

Serial.println(EEPROM.read(6)*0.1);
Serial.print("Pressure Reset level ");
Serial.println(reset_level);
}

void loop() {
// put your main code here, to run repeatedly:
mpu_sensor();
kompas();
suhu= sensor.getTemperature(CELSIUS, ADC_1024);
pressure_abs = sensor.getPressure(ADC_1024);
myRA.addValue(pressure_abs);
avg = myRA.getFastAverage();
water_level = avg - reset_level;
if (mode)
{
SetPID(kp, ki, kd, kalPitch);
adaptivePID_Depth(depth_kp, depth_ki, depth_kd, avg);
if (control)
{
//Balancing output
motorGo(4,Output>0?ccw:cw,abs(Output));
motorGo(5,Output>0?ccw:cw,abs(Output));
}
else
{
//Depth Output
motorGo(4,de_Output>0?cw:ccw,abs(de_Output + Output));
motorGo(5,de_Output>0?ccw:cw,abs(de_Output - Output));
}
}
else
{
if(!onoff)
{
on_off();
}
}
}

template <class T> int EEPROM_writeAnything(int ee, const T& value)
{
const byte* p = (const byte*)(const void*)&value;
int i;
for (i = 0; i < sizeof(value); i++)
EEPROM.write(ee++, *p++);
return i;
}

```

```

}
template <class T> int EEPROM_readAnything(int ee, T& value)
{
  byte* p = (byte*)(void*)&value;
  int i;
  for (i = 0; i < sizeof(value); i++)
    *p++ = EEPROM.read(ee++);
  return i;
}

```

control.ino

```

void SetPID(float kp, float ki, float kd, float set_input)
{
  Setpoint = balance;
  Input = set_input;
  myPID.SetTunings(kp,ki,kd);
  myPID.Compute();
}

```

```

void PID_Depth(float depth_kp, float depth_ki, float depth_kd, float depth)
{
  de_Setpoint = reset_level + set_level;
  de_Input = depth;
  depth_PID.SetTunings(depth_kp,depth_ki,depth_kd);
  depth_PID.Compute();
}

```

```

void adaptivePID_Depth(float depth_kp, float depth_ki, float depth_kd, float
depth)
{
  de_Setpoint = reset_level + set_level;
  de_Input = depth;
  double gap = abs(de_Setpoint - depth);
  if (!adaptive)
  {
    if (gap < 5)
    {
      depth_PID.SetTunings(adapt_kp,adapt_ki,adapt_kd);
    }
    else
    {
      depth_PID.SetTunings(depth_kp,depth_ki,depth_kd);
    }
  }
  else

```

```

{
  depth_PID.SetTunings(depth_kp,depth_ki,depth_kd);
}

depth_PID.Compute();
}

void PID_Heading(float head_kp, float head_ki, float head_kd, float heading)
{
  head_Setpoint = head_set;
  head_Input = heading;
  head_PID.SetTunings(head_kp,head_ki,head_kd);
  head_PID.Compute();

  motorGo(2,head_Output>0?cw:ccw,abs(head_Output));
  motorGo(3,head_Output>0?ccw:cw,abs(head_Output));

}

void on_off()
{
  double setpoint = set_level;
  double gap = setpoint - water_level;
  motorGo(4,gap>0?cw:ccw,255);
  motorGo(5,gap>0?ccw:cw,255);
}

```

motor.ino

```

void motorGo(int motor, int direct, int pwm)
{
  if (motor <= 6)
  {
    if (direct == 0)
    {
      digitalWrite(inApin[motor], HIGH);
      digitalWrite(inBpin[motor], LOW);
    }
    else if (direct == 1)
    {
      digitalWrite(inApin[motor], LOW);
      digitalWrite(inBpin[motor], HIGH);
    }
  }
  analogWrite(pwmpin[motor], pwm);
}

```

```

}

void up()
{
  motorGo(4,cw,255);
  motorGo(5,ccw,255);
}
void down()
{
  motorGo(4,ccw,255);
  motorGo(5,cw,255);
}
void forward()
{
  motorGo(2,cw,255);
  motorGo(3,ccw,255);
}
void back()
{
  motorGo(2,ccw,255);
  motorGo(3,cw,255);
}
void right()
{
  motorGo(2,cw,255);
  motorGo(3,cw,255);
}
void left()
{
  motorGo(2,ccw,255);
  motorGo(3,ccw,255);
}
void berhenti()
{
  motorGo(2,cw,0);
  motorGo(3,cw,0);
  motorGo(4,cw,0);
  motorGo(5,cw,0);
}

```

sensor.ino

```
#define RAW_DATA_CSV
#define RAW_DATA_ARRAY
#define HMC5883_WriteAddress 0x1E // i.e 0x3C >> 1
#define HMC5883_ModeRegisterAddress 0x02
#define HMC5883_ContinuousModeCommand (uint8_t)0x00 // cast to uint8_t
added to get code to compile under Arduino v1.0
#define HMC5883_DataOutputXMSBAddress 0x03

//=====alamat i2c kompas=====
int regb=0x01;
int regbdata=0x40;
int outputData[6];
const int x_offset = 0;
const int y_offset = 0;
const int z_offset = 0;
#ifdef SCALE
const int x_scale = 1;
const int y_scale = 1;
const int z_scale = 1;
#endif
#define I2C_TX write
#define I2C_RX read

double sealevel(double P, double A)
{
  return(P/pow(1-(A/44330.0),5.255));
}

double altitude(double P, double P0)
{
  return(44330.0*(1-pow(P/P0,1/5.255)));
}

void mpu_sensor()
{
  Vector normAccel = mpu.readNormalizeAccel();
  Vector gyr = mpu.readNormalizeGyro();
  acc_z = normAccel.ZAxis;
  // Calculate Pitch, Roll and Yaw
  pitch = -(atan2(normAccel.XAxis, sqrt(normAccel.YAxis*normAccel.YAxis +
normAccel.ZAxis*normAccel.ZAxis))*180.0)/M_PI;
  roll = (atan2(normAccel.YAxis, normAccel.ZAxis)*180.0)/M_PI;
  kalPitch = kalmanY.update(pitch, gyr.YAxis);
  kalRoll = kalmanX.update(roll, gyr.XAxis);
}
```

```

void kompas()
{
int i,x,y,z;
Wire.beginTransmission(HMC5883_WriteAddress);
Wire.I2C_TX(regb);
Wire.I2C_TX(regbdata);
Wire.endTransmission();
Wire.beginTransmission(HMC5883_WriteAddress); //Initiate a transmission
with HMC5883 (Write address).
Wire.I2C_TX(HMC5883_ModeRegisterAddress); //Place the Mode Register
Address in send-buffer.
Wire.I2C_TX(HMC5883_ContinuousModeCommand); //Place the command
for Continuous operation Mode in send-buffer.
Wire.endTransmission(); //Send the send-buffer to HMC5883 and
end the I2C transmission.
Wire.beginTransmission(HMC5883_WriteAddress); //Initiate a transmission
with HMC5883 (Write address).
Wire.requestFrom(HMC5883_WriteAddress,6);
if(6 <= Wire.available())
{
for(i=0;i<6;i++)
{
outputData[i]=Wire.I2C_RX(); //Store the data in outputData buffer
}
}
x=outputData[0] << 8 | outputData[1]; //Combine MSB and LSB of X Data
output register
z=outputData[2] << 8 | outputData[3]; //Combine MSB and LSB of Z Data
output register
y=outputData[4] << 8 | outputData[5]; //Combine MSB and LSB of Y Data
output register
#ifdef SCALE
angle= atan2((double)y * y_scale - y_offset,(double)x * x_scale - x_offset)*
(180 / 3.14159265) +180; // angle in degrees
#else
angle= atan2((double)y - y_offset,(double)x - x_offset)* (180 / 3.14159265)
+180; // angle in degrees
#endif
if((0 < angle) && (angle < 180) )
{
angle=angle;
}
else
{
angle=angle-360;
}
}

```

serial_data.ino

```
void kirim()
{
//serial data send (suhu,pressure,pitch,roll,water lavell)
  Serial.print(suhu);
  Serial.print(",");
  Serial.print(avg);
  Serial.print(",");
  Serial.print(kalRoll);
  Serial.print(",");
  Serial.print(kalPitch);
  Serial.print(",");
  Serial.print(water_level);
  Serial.print(",");
  Serial.print(angle);
  Serial.println(",");
}
```

```
void serialEvent() {
  while (Serial.available()) {
    char inChar = (char)Serial.read();
    inputString += inChar;
    if (inChar == '\n') {
      if(inputString=="up\n"){
        up();
      }
      else if(inputString=="down\n"){
        down();
      }
      else if(inputString=="forw\n"){
        forward();
      }
      else if(inputString=="back\n"){
        back();
      }
      else if(inputString=="stop\n"){
        berhenti();
      }
      else if(inputString=="righ\n"){
        right();
      }
      else if(inputString=="left\n"){
        left();
      }
    }
  }
//PID Balance
```

```

else if(inputString.substring(0,1)=="p"){
  int nilai_kp =inputString.substring(1).toInt();
  EEPROM.write(1,nilai_kp);
  kp = EEPROM.read(1);
  Serial.println("Gain P Set OK");
}
else if(inputString.substring(0,1)=="i"){
  int nilai_ki=inputString.substring(1).toInt();
  EEPROM.write(2,nilai_ki);
  ki = EEPROM.read(2)*0.1;
  Serial.println("Gain I Set OK");
}
else if(inputString.substring(0,1)=="d"){
  int nilai_kd =inputString.substring(1).toInt();
  EEPROM.write(3,nilai_kd);
  kd = EEPROM.read(3)*0.1;
  Serial.println("Gain D Set OK");
}
//PID Depth
else if(inputString.substring(0,2)=="np"){
  int Value_kp =inputString.substring(2).toInt();
  EEPROM.write(4,Value_kp);
  depth_kp = EEPROM.read(4);
  Serial.println("Gain Depth_P Set OK");
}
else if(inputString.substring(0,2)=="ni"){
  int Value_ki =inputString.substring(2).toInt();
  EEPROM.write(5,Value_ki);
  depth_ki = EEPROM.read(5)*0.1;
  Serial.println("Gain Depth_I Set OK");
}
else if(inputString.substring(0,2)=="nd"){
  int Value_kd =inputString.substring(2).toInt();
  EEPROM.write(6,Value_kd);
  depth_kd = EEPROM.read(6)*0.1;
  Serial.println("Gain Depth_D Set OK");
}
//Adaptive PID
else if(inputString.substring(0,2)=="kp"){
  int Val_kp =inputString.substring(2).toInt();
  EEPROM.write(20,Val_kp);
  adapt_kp = EEPROM.read(20);
  Serial.println("Gain Adapt_Depth_P Set OK");
}
else if(inputString.substring(0,2)=="ki"){
  int Val_ki =inputString.substring(2).toInt();
  EEPROM.write(21,Val_ki);

```

```

    adapt_ki = EEPROM.read(21)*0.1;
    Serial.println("Gain Adaphth_Depth_I Set OK");
}
else if(inputString.substring(0,2)=="kd"){
    int Val_kd =inputString.substring(2).toInt();
    EEPROM.write(22,Val_kd);
    adapt_kd = EEPROM.read(22)*0.1;
    Serial.println("Gain Adaphth_Depth_D Set OK");
}
//PID Heading
else if(inputString.substring(0,2)=="hp"){
    int val_kp =inputString.substring(2).toInt();
    EEPROM.write(15,val_kp);
    head_kp = EEPROM.read(15);
    Serial.println("Gain Heading_P Set OK");
}
else if(inputString.substring(0,2)=="hi"){
    int val_ki =inputString.substring(2).toInt();
    EEPROM.write(16,val_ki);
    head_ki = EEPROM.read(16)*0.1;
    Serial.println("Gain Heading_I Set OK");
}
else if(inputString.substring(0,2)=="hd"){
    int val_kd =inputString.substring(2).toInt();
    EEPROM.write(17,val_kd);
    head_kd = EEPROM.read(17)*0.1;
    Serial.println("Gain Heading_D Set OK");
}

else if(inputString.substring(0,3)=="cek"){
    Serial.print("Set_Point Balance (Roll) = ");
    Serial.println(Setpoint);
    Serial.print("Kp = ");
    Serial.print(kp);
    Serial.print(',');
    Serial.print("Ki = ");
    Serial.print(ki);
    Serial.print(',');
    Serial.print("Kd = ");
    Serial.println(kd);
}
else if(inputString=="send\n"){
    kirim();
}
else if(inputString=="auto\n"){
    mode = true;
    onoff = true;
}

```

```

    Serial.println("Automatic Mode");
}
else if(inputString=="manu\n"){
    mode = false;
    berhenti();
    Serial.println("Manual Mode");
}
else if(inputString=="bala\n"){
    control = true;
    Serial.println("Balancing Control");
}
else if(inputString=="onof\n"){
    onoff = false;
    Serial.println("On_Off Control");
}
else if(inputString.substring(0,4)=="wate"){
    control = false;
    adaptive = true;
    berhenti();
    Serial.println("Depth Control");
}
else if(inputString.substring(0,4)=="tive"){
    adaptive = false;
    berhenti();
    Serial.println("Adaptive Control Active");
}
else if(inputString=="res\n"){
    reset_level = pressure_abs;
    EEPROM_writeAnything( 7, reset_level);
    Serial.println("Reset Water Level Ok ");
}
else if(inputString.substring(0,3)=="set"){
    set_level =inputString.substring(3).toInt();
    Serial.print("Set_Point Water Level OK = ");
    Serial.println(set_level);
}
else if(inputString.substring(0,3)=="bet"){
    balance =inputString.substring(3).toInt();
    Serial.print("Set_Point Balance (Roll) OK = ");
    Serial.println(Setpoint);
}
else if(inputString.substring(0,3)=="hea"){
    head_set =inputString.substring(3).toInt();
    //EEPROM.write(7, level);
    //set_level = EEPROM.read(7);
    Serial.print("Set_Point Heading (Yaw) OK = ");
    Serial.println(head_set);
}

```

```

}
else if(inputString=="lev\n"){
  Serial.print("Set_Point Water Level = ");
  Serial.println(de_Setpoint - reset_level);
  Serial.print("Kp = ");
  Serial.print(depth_kp);
  Serial.print(',');
  Serial.print("Ki = ");
  Serial.print(depth_ki);
  Serial.print(',');
  Serial.print("Kd = ");
  Serial.println(depth_kd);
}
else if(inputString=="hsp\n"){
  Serial.print("Set_Point Heading (Yaw) = ");
  Serial.println(head_Setpoint);
  Serial.print("Kp = ");
  Serial.print(head_kp);
  Serial.print(',');
  Serial.print("Ki = ");
  Serial.print(head_ki);
  Serial.print(',');
  Serial.print("Kd = ");
  Serial.println(head_kd);
}
else if(inputString=="ada\n"){
  Serial.print("Adaptive Kp = ");
  Serial.print(adapt_kp);
  Serial.print(',');
  Serial.print("Ki = ");
  Serial.print(adapt_ki);
  Serial.print(',');
  Serial.print("Kd = ");
  Serial.println(adapt_kd);
}
inputString="";
}
}
}

```

RIWAYAT PENULIS



Ali Zainal Abidin di lahirkan di Palembang 05 November 1989, anak ke dua dari pasangan Abdullah Syahab dan Aisyah Abdullah Syahab. Penulis memulai pendidikan sekolah dasar di MI Adabiyah II Palembang pada tahun 1994 - 2000 , kemudian menjalankan sekolah menengah pertama di Adabiyah II Palembang pada tahun 2000-2003, selanjutnya penulis melanjutkan di sekolah menengah atas di Bina Warga I Palembang pada tahun 2003 – 2006, setelah itu penulis menempuh pendidikan tinggi pada program sarjana teknik Departemen Teknik Elektro dengan bidang konsentrasi Teknik Kendali dan Komputer di Universitas Sriwijaya pada tahun 2006 – 2010, dan melanjutkan pendidikan program magister Departemen Teknik Elektro di Institut Teknologi Sepuluh Nopember Surabaya dengan bidang keahlian Teknik Elektronika pada tahun 2014.