



TUGAS AKHIR - TE145561

RANCANG BANGUN *PROTOTYPE* ROV (*REMOTELY OPERATED VEHICLES*) UNTUK MELETAKKAN TERUMBU KARANG DENGAN PENGENDALI *JOYSTICK*

Gerry Hendria Negara
NRP 1031150000046

Dosen Pembimbing
Ir. Rusdhianto Effendy A.K.,M.T.
Yunafi'atul Aniroh, S.T. M.Sc.

DEPARTEMEN TEKNIK ELEKTRO OTOMASI
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018



TUGAS AKHIR - TE145561

RANCANG BANGUN *PROTOTYPE* ROV (*REMOTELY OPERATED VEHICLES*) UNTUK MELETAKKAN TERUMBU KARANG DENGAN PENGENDALI *JOYSTICK*

Gerry Hendria Negara
NRP 10311500000046

Dosen Pembimbing
Ir. Rusdhianto Effendy A.K.,M.T.
Yunafi'atul Aniroh, S.T. M.Sc.

DEPARTEMEN TEKNIK ELEKTRO OTOMASI
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018

-----Halaman ini sengaja dikosongkan-----



FINAL PROJECT - TE145561

DESIGN PROTOTYPE ROV (REMOTELY OPERATED VEHICLES) TO PUT CORAL REEFS WITH JOYSTICK CONTROL

Gerry Hendria Negara
NRP 1031150000046

Supervisor
Ir. Rusdhianto Effendy A.K.,M.T.
Yunafi'atul Aniroh, S.T. M.Sc.

*DEPARTEMENT OF ELECTRICAL AUTOMATION ENGINEERING
Faculty of Vocations
Institut Teknologi Sepuluh Nopember
Surabaya 2018*

-----Halaman ini sengaja dikosongkan-----

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul **Rancang Bangun *Prototype ROV (Remotely Operated Vehicles)* untuk Meletakkan Terumbu Karang dengan Pengendali *Joystick*** adalah benar-benar hasil karya intelektual sendiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya orang 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 dengan peraturan yang berlaku.

Surabaya, 24 Juli 2018



Gerry Hendria Negara
NRP. 10311500000046

-----Halaman ini sengaja dikosongkan-----

RANCANG BANGUN PROTOTYPE ROV (REMOTELY OPERATED VEHICLES) UNTUK MELETAKKAN TERUMBU KARANG DENGAN PENGENDALI JOYSTICK

TUGAS AKHIR

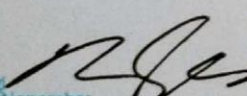
Diajukan Guna Memenuhi Sebagian Persyaratan
Memperoleh Gelar Ahli Madya Teknik
Pada

Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember

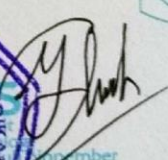
Menyetujui:

Dosen Pembimbing I

Dosen Pembimbing II


Ir. Rusdhianto Effendy, A.K., M.T.

NIP. 195704241985021001


Yuniarti Aniroh, S.T.M.Sc.

NIP. 2200201405001

**SURABAYA
JULI, 2018**

-----Halaman ini sengaja dikosongkan-----

RANCANG BANGUN *PROTOTYPE ROV (REMOTELY OPERATED VEHICLES)* UNTUK MELETAKKAN TERUMBU KARANG DENGAN PENGENDALI *JOYSTICK*

Nama : Gerry Hendria Negara
Pembimbing I : Ir. Rusdhianto Effendy A.K., M.T.
Pembimbing II : Yunafi'atul Aniroh, S.T. M.Sc.

ABSTRAK

Robot bawah laut atau dikenal dengan nama *Remotely Operated Vehicles* (ROV) merupakan robot yang dapat dikendalikan oleh manusia dari atas laut. ROV berfungsi untuk membantu penyelam melakukan misi peletakkan terumbu karang ke bawah laut. Peletakkan terumbu karang memiliki banyak kendala seperti serangan hiu, cuaca yang tidak menentu serta biaya yang mahal. Sehingga dibuat ROV untuk melakukan misi peletakkan terumbu karang di dasar laut.

ROV dirancang dengan menggunakan pipa paralon sehingga mudah dibentuk. Komponen elektrik yang digunakan adalah *driver* motor BTS7960 dan sensor MPU6050 untuk mengetahui kemiringan ROV didalam air. Pergerakan ROV dikontrol dengan *joystick* yang akan memberikan signal ke motor *bilgepump* yang telah dihubungkan dengan aki, *driver* motor BTS7960 serta Arduinomega 2560 sebagai mikrokontroller. ROV dapat bergerak maju, mundur, belok kanan dan belok kiri karena motor diletakkan di belakang ROV secara *horizontal*, kemudian gerakkan naik dan turun motor diletakkan secara *vertical* pada kaki ROV.

Dari perancangan ROV didapatkan bahwa ROV dapat melayang dengan beban maksimal 4-5,5 Kg. Pengendalian gerakan ROV yang dibuat sudah sesuai dengan yang diberikan oleh *joystick* yaitu maju dengan kecepatan 0,19 m/s, mundur dengan kecepatan 0,15 m/s, naik dengan kecepatan 0,11 m/s, turun dengan kecepatan 0,16 m/s, belok kanan dan belok kiri dengan kecepatan 0,14 m/s. Dari hasil pengujian, sensor MPU 6050 mampu menunjukkan kemiringan ROV dengan nilai *error* pembacaan sensor sebesar 0,52% didarat dan 6,7% didalam air, sehingga *operator* dapat membetulkan posisi ROV hingga seimbang posisinya.

Kata Kunci : *Bilgepump*, *BTS7960*, *Joystick*, *Sensor MPU 6050*, *ROV*.

-----Halaman ini sengaja dikosongkan-----

**DESIGN PROTOTYPE ROV (REMOTELY OPERATED
VEHICLES) TO PUT CORAL REEFS
WITH JOYSTICK CONTROL**

Name : Gerry Hendria Negara
Supervisor I : Ir. Rusdhianto Effendy A.K., M.T.
Supervisor II : Yunafi'atul Aniroh, S.T. M.Sc.

ABSTRACT

Underwater robot or known as Remotely Operated Vehicles (ROV) is a robot that can be controlled by humans from the sea. The function of ROV to help divers do missions of laying coral reefs. The laying of coral reefs has many constraints such as shark attacks, The uncertain weather and high cost. So ROV is made to do the mission of laying coral reefs on the seabed.

The ROV was designed using paralon pipes so make it easy to formed. Electrical components used motorcycle driver BTS7960 and MPU6050 sensor to know the slope of ROV in water. The ROV movement was controlled by a joystick by giving signal to bilgepump motor that had been connected to the battery, the BTS7960 motor driver and Arduinomega 2560 as microcontroller. ROV could move forward, backward, turn right and turn left because the motor was mounted behind the ROV horizontally, then movement up and down the motor mounted vertically on the foot of the ROV.

From the design of ROV found that ROV could float with a maximum load of 4-5,5 Kg. The control of ROV movements was in accordance with the joystick controlled of advanced at a speed of 0.19 m/s, retreated at a speed of 0.15 m/s, rised at a speed of 0.11 m/s, descended at a speed of 0.16 m/s, turn right and turn left at a speed of 0.14 m/s. From the test results, the MPU 6050 sensor was able to show the slope of ROV with sensor error reading value of 0.52% on the ground and 6.7% in water, so that the operator could correct the ROV position to balance the position.

Keywords: *Bilgepump, BTS7960, Joystick, Sensor MPU 6050, ROV.*

-----Halaman ini sengaja dikosongkan-----

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT yang selalu memberikan rahmat dan hidayah-Nya sehingga Tugas Akhir ini dapat terselesaikan dengan baik. Shalawat serta salam semoga selalu dilimpahkan kepada Rasulullah Muhammad SAW, keluarga, sahabat, dan umat muslim yang senantiasa meneladani beliau.

Tugas Akhir ini disusun untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan Diploma 3 pada Bidang Studi Komputer Kontrol, Departemen Teknik Elektro Otomasi, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember Surabaya dengan judul:

RANCANG BANGUN *PROTOTYPE ROV (REMOTELY OPERATED VEHICLES)* UNTUK MELETAKKAN TERUMBU KARANG DENGAN PENGENDALI *JOYSTICK*

Penulis mengucapkan terima kasih kepada Ibu dan Bapak penulis yang memberikan berbagai bentuk doa serta dukungan tulus tiada henti, Bapak Ir. Rusdhianto Effendy A.K., M.T. dan Ibu Yunafi'atul Aniroh, S.T. M.Sc. atas segala bimbingan ilmu, moral, dan spiritual dari awal hingga terselesaikannya Tugas Akhir ini, Penulis juga mengucapkan banyak terima kasih kepada semua pihak yang telah membantu baik secara langsung maupun tidak langsung dalam proses penyelesaian Tugas Akhir ini.

Penulis menyadari dan memohon maaf atas segala kekurangan pada Tugas Akhir ini. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat dalam pengembangan keilmuan di kemudian hari.

Surabaya, 24 Juli 2018

Penulis

-----Halaman ini sengaja dikosongkan-----

DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN JUDUL.....	iii
PERNYATAAN KEASLIAN TUGAS AKHIR.....	Error! Bookmark not defined.
HALAMAN PENGESAHAN.....	vii
ABSTRAK.....	ix
<i>ABSTRACT</i>	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL.....	xix
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Tujuan dan Manfaat.....	2
1.3 Perumusan Masalah.....	3
1.4 Batasan Masalah.....	3
1.5 Metodologi Penelitian.....	3
1.6 Sistematika Penulisan Laporan.....	5
BAB II TEORI PENUNJANG.....	7
2.1 Kajian Pustaka.....	7
2.2 ROV (Remotely Operated Vehicle).....	8
2.3 Arduino MEGA.....	9
2.4 <i>Joystick</i> PS2.....	9
2.5 <i>Gyroscope</i>	10
2.6 <i>Bilgepump</i> Motor.....	10
2.7 Hukum Archimedes.....	11
2.8 <i>Driver</i> Motor BTS7960.....	11
2.9 Pulse Width Modulation (PWM).....	12
2.10 Baterai (<i>Accu</i>).....	13
BAB III PERANCANGAN SISTEM.....	15
3.1 Perancangan Sistem.....	16
3.2 Perancangan Perangkat Keras.....	16
3.2.1 Desain ROV.....	16
3.2.2 Persamaan Gaya Apung ROV.....	20
3.2.3 <i>Wiring Driver</i> Motor.....	21
3.2.4 <i>Wiring Remote</i> Kontrol.....	22
3.3 Perancangan Perangkat Lunak.....	24

3.3.1	Perancangan Gerak ROV	24
3.3.2	Pembacaan Sensor MPU 6050	25
BAB IV	PENGUJIAN DAN ANALISIS DATA	27
4.1	Pengujian Beban Maksimal Rangka ROV	27
4.2	Pengujian Tempat Sensor	30
4.3	Pengujian Sensor MPU 6050 <i>Gyroscope</i>	31
4.4	Pengujian Kontrol <i>Joystick</i> PS2.....	32
4.5	Pengujian Gerakkan ROV Didalam Air	34
4.6	Pengujian Kecepatan ROV Didalam Air	37
4.7	Pengujian Keseluruhan Sistem Didalam Air	39
BAB V	KESIMPULAN DAN SARAN	43
5.1	Kesimpulan.....	43
5.2	Saran	43
DAFTAR PUSTAKA	45
LAMPIRAN	47
A-32	<i>Listing Code</i> Membaca Sudut Sensor MPU 6050	47
B-33	<i>Listing Code</i> Gerakkan ROV dengan <i>Joystick</i>	51
C-33	<i>Listing Code</i> Membaca Perintah <i>Joystick</i>	57
RIWAYAT HIDUP PENULIS	61

DAFTAR GAMBAR

Gambar 2. 1 ROV (<i>Remotely Operated Vehicle</i>).....	8
Gambar 2. 2 <i>Board</i> Arduino	9
Gambar 2. 3 Pinout <i>Joystick</i> PS2.....	9
Gambar 2. 4 Motor <i>Bilgepump</i>	10
Gambar 2. 5 Hukum Archimedes	11
Gambar 2. 6 <i>Driver</i> Motor BTS7960.....	12
Gambar 2. 7 <i>Duty Cycle</i> Pada PWM.....	13
Gambar 2. 8 Aki Yuasa 12 V, 17.2 Ah	13
Gambar 3. 1 Perancangan Sistem.....	15
Gambar 3. 2 Rancangan Desain ROV.....	17
Gambar 3. 3 Rancangan Desain Kaki ROV	17
Gambar 3. 4 Rancangan Desain ROV Tampak Depan	17
Gambar 3. 5 Rancangan Desain Tempat Sensor	18
Gambar 3. 6 Desain Kotak Sensor	18
Gambar 3. 7 Posisi Motor	19
Gambar 3. 8 Bentuk ROV.....	19
Gambar 3. 9 Hukum Archimedes saat Kondisi Melayang.....	20
Gambar 3. 10 Volume Tabung.....	20
Gambar 3. 11 <i>Wiring Driver</i> Motor BTS7690 dengan Arduino	22
Gambar 3. 12 <i>Wiring Stick</i> PS2 dengan Arduino Mega.....	23
Gambar 3. 13 Keterangan Tombol <i>Joystick</i> PS2.....	23
Gambar 3. 14 Flowchart Program Gerak ROV	25
Gambar 3. 15 Koneksi MPU 6050 Dengan Arduino	26
Gambar 4. 1 Realisasi ROV.....	27
Gambar 4. 2 (a) Berat Beban 500 gram, (b) Pengujian Rangka dengan Beban 1 kg.....	28
Gambar 4. 3 (a) Berat Beban 1 kg, (b) Pengujian Rangka dengan Beban 2 kg	29
Gambar 4. 4 (a) Berat Beban 2 kg, (b) Pengujian Rangka dengan Beban 4 kg	29
Gambar 4. 5 Hasil Rangka dengan Beban 7,6 kg.....	30
Gambar 4. 6 Rangka ROV Tanpa Beban	30
Gambar 4. 7 Pengujian Tempat Sensor	31
Gambar 4. 8 Pengujian Sensor <i>Gyroscope</i>	31
Gambar 4. 9 Hasil Baca Perintah <i>Joystick</i> PS2.....	33
Gambar 4. 10 (a) Perintah <i>Joystick</i> Naik (b) Gerakkan ROV Naik ..	34

Gambar 4. 11 (a) Perintah <i>Joystick</i> Turun (b) Gerakkan ROV Turun	35
Gambar 4. 12 (a) Perintah <i>Joystick</i> Maju (b) Gerakkan ROV Maju...	35
Gambar 4.13 (a) Perintah <i>Joystick</i> Mundur (b) Gerakkan ROV Mundur	35
Gambar 4. 14 (a) Perintah <i>Joystick</i> Belok Kanan (b) Gerakkan ROV Belok Kanan	36
Gambar 4. 15 (a) Perintah <i>Joystick</i> Belok Kiri (b) Gerakkan ROV Belok Kiri	36
Gambar 4. 16 Pengujian Kecepatan ROV	37
Gambar 4. 17 Pengujian ROV Didalam Air	39
Gambar 4. 18 Pembacaan Kemiringan ROV dengan Busur	40

DAFTAR TABEL

Tabel 3. 1 Konfigurasi Tombol <i>Remote</i> Kontrol Gerak ROV	23
Tabel 4. 1 Data Pengujian Sudut <i>Pitch</i> Sensor.....	32
Tabel 4. 2 Pengujian Arah Putaran Motor ROV dengan <i>Joystick</i>	33
Tabel 4. 3 Pengujian Kecepatan Gerak ROV Maju	37
Tabel 4. 4 Pengujian Kecepatan Gerak ROV Mundur	37
Tabel 4. 5 Pengujian Kecepatan Gerak ROV Naik	38
Tabel 4. 6 Pengujian Kecepatan Gerak ROV Turun	38
Tabel 4. 7 Pengujian Kecepatan Gerak ROV Belok Kanan.....	38
Tabel 4. 8 Pengujian Kecepatan Gerak ROV Belok Kiri.....	39
Tabel 4. 9 Pengujian Keseluruhan Sistem didalam Air.....	41

-----Halaman ini sengaja dikosongkan---

BAB I

PENDAHULUAN

1.1 Latar Belakang

Indonesia merupakan negara dengan luas wilayah laut 96.079,15 km² sehingga disebut negara maritim. Memiliki laut yang luas membuat Indonesia memiliki banyak *spot* penyelaman yang sangat cantik. Banyaknya biota laut dan terumbu karang yang indah, hampir membentang luas di wilayah Indonesia. Terumbu karang merupakan tempat dimana ikan-ikan dan biota laut dapat hidup untuk mencari makanan dan berkembang biak. Akhir-akhir ini terjadi banyak sekali kerusakan terumbu karang yang tersebar di seluruh perairan di Indonesia, menurut Siti Nurbaya selaku Menteri Lingkungan Hidup dan Kehutanan ini mengatakan “13.522 m² dalam hitungan KLHK yang berada di lapangan, terumbu karang di Raja Ampat rusak” (Irawan F, 2015), hal ini diakibatkan kapal peisar inggris yang bernama *csledonian sky* masuk sampai perairan dangkal.

Hal itu merupakan salah satu bentuk rusaknya harta bawah laut milik Indonesia yang dilakukan oleh pihak asing. Para nelayan Indonesia yang secara tradisional selalu memanfaatkan bom ikan demi mendapatkan tangkapan ikan yang besar, hal ini juga mengakibatkan terumbu karang ikut rusak akibat efek ledakan yang dahsyat. Banyak upaya yang dilakukan oleh beberapa pihak untuk mengatasi hal ini, antara lain memperingatkan para nelayan untuk tidak menggunakan bom ikan, tapi cara tradisional menggunakan jaring. Namun efek kerusakan yang sudah terjadi perlu juga ditangani, seperti dengan menanam kembali terumbu karang yang telah rusak. Penanaman ini biasanya dilakukan oleh para penyelam yang akan menyelam ke laut untuk menanam terumbu karang. Akan tetapi banyak tantangan seperti adanya serangan ikan hiu, menurut data tahun 2015 terdapat serangan hiu kepada penyelam dan peselancar sebanyak 98 kali (Jemadu, 2016). Hal ini membuktikan bahwa penyelaman sangat berbahaya bila disekitar karang terdapat hiu.

Beberapa tahun belakangan ini terdapat teknologi robot bawah laut yang bernama ROV (*Remotely Operated Vehicles*). ROV merupakan robot bawah laut yang sering digunakan untuk melakukan kegiatan penelitian dibawah laut dengan pilot yang ada diatas kapal. ROV yang digunakan untuk penelitian sangat beragam tergantung tujuan dibuatnya ROV tersebut. Pada tahun 2012, Irfan telah membuat ROV yang bertujuan

untuk monitoring pencemaran air sungai, akan tetepi ROV yang dibuat untuk keperluan penjernihan air laut tawar saja. Kemudian pada penelitian yang dipimpin oleh Tirza H. Samosir memanfaatkan ROV untuk memonitoring keadaan bawah laut disekitar perairan Sulawesi Utara dan Biak Papua yang meliputi kadar oksigen, suhu air laut dan keragaman biota laut yang ada disekitarnya (Samosir, 2012).

Pada tugas akhir ini akan dibuat ROV yang dapat menyelam sampai 3 meter didalam air, dan dapat bergerak didalam air. Pada robot bawah air memiliki suatu sistem yang bernama sistem *ballast*. Sistem *ballast* merupakan suatu sistem untuk memanipulasi berat benda didalam air. Sistem *ballast* pada ROV ini menggunakan motor DC yang dapat di kendalikan arah putarannya. Sistem tersebut akan dikontrol menggunakan *joystick*. ROV akan dirancang dengan 4 buah motor DC yang akan diletakkan pada kanan kiri ROV yang akan menggerakkan maju dan mundur. Kemudian untuk mengapung dan tenggelam maka diperlukan motor DC yang mengarah ke atas. Perancangan Sistem diharapkan dapat dikendalikan menggunakan *joystick* untuk kondisi tenggelam dan mengapung dengan cepat. Serta dapat melakukan misi mencapai target untuk melakukan peletakan terumbu karang.

Dari tugas akhir ini diharapkan akan mempermudah para penyelam untuk melakukan peletakan terumbu karang didasar laut yang sangat dalam tanpa harus menyelam dan dapat menghemat waktu serta mengurangi bahaya yang dapat menyerang penyelam ketika berada didalam air.

1.2 Tujuan dan Manfaat

Tujuan dari tugas akhir ini adalah membuat *prototype* ROV yang dapat dikendalikan menggunakan *joystick*. Mempermudah dalam membantu melakukan misi peletakan terumbu karang dengan menggunakan ROV, manfaat dari tugas akhir ini yaitu:

1. Dengan adanya robot bawah air atau ROV yang bisa digunakan untuk melakukan misi peletakan terumbu karang di air laut untuk membantu peletarian terumbu karang.
2. Dengan adanya ROV yang dapat membantu monitoring terhadap pertumbuhan terumbu karang tanpa harus menyelam.
3. Dengan adanya ROV yang dapat digunakan untuk penelitian bawah laut.

1.3 Perumusan Masalah

Perancangan tubuh ROV yang digunakan membutuhkan berat yang sama dengan gaya apung yang diberikan air kepada setiap benda yang masuk kedalam air. Pada perancangan ini berat yang diinginkan membuat tubuh ROV melayang ketika berada didalam air. Pengendalian gerakan ROV dengan *joystick* untuk mengatur arah gerakan ROV didalam air.

1.4 Batasan Masalah

Dalam pembuatan alat pada tugas akhir ini batasan masalah ada pada:

1. Rancangan ROV menggunakan pipa paralon dan didesain agar mudah bergerak didalam air.
2. Motor yang digunakan yaitu Motor DC *Bilgepump* 1100gph.
3. Kontrol yang digunakan menggunakan *joystick* PS2.
4. Arah gerakan dari ROV maju, mundur, belok kanan, belok kiri, naik, dan turun.
5. *Driver* motor yang digunakan yaitu *driver* motor BTS7960.
6. Baterai yang digunakan yaitu aki dengan kapasitas 12V dan 17,5Ah.
7. Pengujian dilakukan di kolam renang dengan kedalaman 1 meter.

1.5 Metodologi Penelitian

Dalam pelaksanaan tugas akhir yang berupa rancang bangun *prototype* ROV (*Remotely Operated Vehicles*) untuk meletakkan terumbu karang dengan pengendali *joystick*, ada beberapa kegiatan yang dapat diuraikan sebagai berikut:

a. Tahap persiapan

Pada tahap ini akan dilakukan studi literatur mengenai :

1. Mempelajari macam-macam bentuk tubuh ROV.
Bentuk tubuh ROV memiliki banyak model berdasarkan keperluan dari ROV yang dibuat. Membuat bentuk tubuh ROV yang tepat merupakan salah satu cara untuk membuat semua komponen dalam ROV selalu aman. Serta memastikan bentuknya dapat dengan mudah digerakkan dalam air.
2. Mempelajari sistem *ballast*.
Sistem *ballast* merupakan suatu sistem untuk memanipulasi berat benda didalam air, sistem ini biasanya terdapat pada kapal selam maupun kapal pengantar barang.

3. Mempelajari *driver* motor yang akan digunakan untuk menggerakkan motor dengan tepat.
Driver motor yang akan digunakan menggunakan BTS7960 yang memiliki sistem *H-bridge* didalamnya.
4. Mempelajari kontrol dari *joystick* PS2.
Joystick merupakan alat masukkan komputer yang berbentuk tuas atau tongkat yang dapat bergerak ke segala arah, dengan dilengkapi *games paddle* berbentuk kotak atau persegi terbuat dari plastik dan tombol-tombol yang akan mengatur gerak suatu objek dalam komputer.

b. Tahap identifikasi dan pemodelan sistem

Pada tahap ini akan dilakukan identifikasi dari sistem alat sesuai data yang telah didapatkan dari studi literatur serta dilakukan pemodelan dari alat yang akan dikerjakan.

c. Tahap perancangan

Pada tahap ini akan dilakukan perancangan dari sistem sesuai data yang telah didapatkan dari studi literatur. Dimulai dari perancangan *hardware*, pengujian *hardware* dalam air, pemilihan *driver* motor, dan pengujian *driver* motor dengan motor DC *bilgepump* 1100gph. Setelah semua selesai dilakukan perancangan *software* pemrograman menggunakan arduino mega untuk kontrol gerakan ROV dengan *joystick*.

d. Tahap pembuatan alat

Pada tahap ini akan dilakukan pembuatan alat sesuai perancangan yang dibuat, berdasarkan data yang telah dikumpulkan melalui studi literatur.

e. Tahap pengujian dan pengambilan data

Pada tahap ini akan dilakukan pengujian alat, pengambilan data atau kegagalan pada alat dan mengatasi permasalahan tersebut. Tahapan ini dilakukan berdasarkan urutan di bawah ini:

1. Pengujian beban maksimal rangka ROV.
2. Pengujian tempat sensor.
3. Pengujian sensor MPU 6050.
4. Pengujian program kontrol ROV menggunakan *joystick*.

5. Pengambilan data arah gerakan motor dengan kesesuaian dengan kontrol *joystick*.
6. Pengujian kecepatan ROV didalam air.
7. Pengujian keseluruhan sistem didalam air.

f. Tahap penyusunan laporan

Setelah alat berhasil dibuat dan bekerja dengan baik, dan pengambilan data terpenuhi, maka tahap selanjutnya yaitu penyusunan laporan untuk buku tugas akhir. Diharapkan buku tugas akhir ini bermanfaat bagi semua orang dan dapat dijadikan pedoman dalam melanjutkan dan mengembangkan ide tugas akhir ini.

1.6 Sistematika Penulisan Laporan

Untuk pembahasan lebih lanjut, laporan tugas akhir ini disusun dengan sistematika sebagai berikut:

Bab I PENDAHULUAN

Membahas tentang latar belakang, perumusan masalah, tujuan, manfaat, batasan masalah, sistematika laporan, metodologi penelitian, dan sistematika laporan.

Bab II TEORI PENUNJANG

Menjelaskan teori yang berisi teori-teori dasar yang dijadikan landasan dan mendukung dalam perencanaan dan pembuatan alat yang dibuat.

Bab III PERANCANGAN SISTEM

Membahas perencanaan dan pembuatan tentang perencanaan dan pembuatan *hardware* yang meliputi desain ROV dan perancangan *software* yang meliputi program yang akan digunakan untuk menjalankan alat tersebut.

Bab IV PENGUJIAN DAN ANALISA DATA

Membahas pengujian alat dan pengambilan data yang didapat dari pengujian tersebut.

Bab V KESIMPULAN DAN SARAN

Berisi penutup yang menjelaskan tentang kesimpulan yang didapat dari tugas akhir ini dan saran-saran untuk pengembangan alat ini lebih lanjut.

-----Halaman ini sengaja dikosongkan-----

BAB II TEORI PENUNJANG

2.1 Kajian Pustaka

M.Irfan Bahrudin (2012), sistem kontrol ROV (*Remotely Operated Vehicles*) mode *wire* untuk monitoring pencemaran air dengan kamera sebagai pengganti visual, penelitian tersebut menjelaskan sistem kontrol dari ROV yang menggunakan kabel untuk melakukan monitoring pencemaran air tawar menggunakan sensor PH air dan temperatur. ROV yang dibuat untuk mengecek pencemaran air yang berada di air tawar, kontrol yang digunakan menggunakan HMI yang ditampilkan pada *leptop operator* serta ROV yang dibuat menggunakan 3 buah motor bilgepump.

Perbedaan penelitian tersebut dengan tugas akhir yang dibuat adalah motor yang digunakan pada tugas akhir ini berjumlah 4 buah yang akan membuat pergerakannya seimbang, tugas akhir ini menggunakan sensor *gyroscope* yang akan memberikan informasi keseimbangan tubuh ROV selama bergerak didalam air. Kontrol yang digunakan pada tugas akhir ini berupa *joystick* PS2 yang akan memudahkan *operator* dalam mengendalikan pergerakan ROV didalam air, serta ROV akan digunakan untuk tujuan membantu melakukan misi peletakkan terumbu karang di laut.

Tirta H. Samosir (2012), Aplikasi *remotely operated vehicle* (ROV) dalam penelitian kelautan dan perikanan di sekitar perairan Sulawesi Utara dan Biak Papua, penelitian tersebut menjelaskan pengaplikasian lain ROV untuk melakukan penelitian sekitar perairan Sulawesi Utara dan Biak Papua. Penelitian tersebut menggunakan sensor kadar oksigen, sensor suhu, dan keanegaraman biota laut dengan menggunakan kamera, serta menggunakan 3 buah motor *brushless* yang diberi pelindung pada motornya agar kedap terhadap air.

Perbedaan penelitian tersebut dengan tugas akhir yang dibuat adalah tidak digunakannya kamera, melainkan dengan sensor *gyroscope* yang akan memberikan informasi keseimbangan pergerakan gerak ROV serta motor *bilgepump* yang tergolong *submarine motor* sehingga sudah kedap terhadap air.

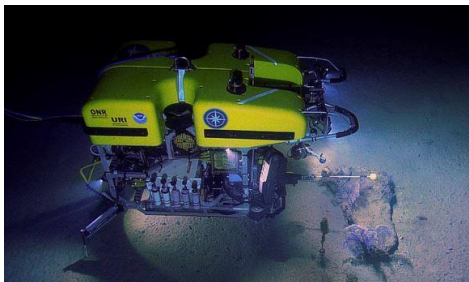
Andik Yulianto dan Hadi Puspa Handoyo (2015), pengembangan robot jelajah air untuk observasi terumbu karang, penelitian tersebut membahas mengenai ROV yang digunakan untuk melakukan penelitian ROV dibawah air laut untuk melakukan monitoring terumbu karang. ROV tersebut menggunakan 3 buah motor *brushless* yang menggunakan ESC

sebagai pengendali kecepatan motor. ROV tersebut menggunakan kontrol berbasis aplikasi *web* yang akan dikendalikan melalui laptop, serta tidak adanya kontrol keseimbangan pada ROV (Robinshon, 2000).

Perbedaan penelitian tersebut dengan tugas akhir yang dibuat terletak pada motor yang digunakan berupa 4 buah motor *bilgepump* yang sudah kedap air. Kontrol yang akan digunakan berupa *joystick* serta *driver* motor yang dapat mengatur kecepatan dan arah putaran motor. Pada ROV ini juga akan dilengkapi sensor *gyroscope* yang akan memberikan informasi tentang keseimbangan gerakan ROV selama didalam air.

2.2 ROV (Remotely Operated Vehicle)

Berdasarkan *Marine Technology Society ROV Committee's* dalam "*Operational Guidelines for ROVs*" (1984) dan *The national Reseaevh Council Comittee's* dalam "*Undersea Vehicles and National Needs*" (1996). *Remotely Operated Vehicle* (ROV) adalah sebuah robot bawah laut yang dikendalikan oleh operator ROV dari atas air laut untuk bekerja dilingkungan yang berbahaya didalam air laut.



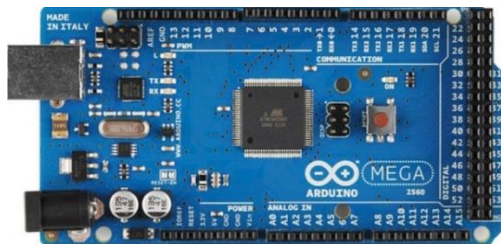
Gambar 2. 1 ROV (*Remotely Operated Vehicle*)

Sistem ROV terdiri atas *vehicle*, yang terhubung oleh kabel *umbilical* ke ruangan kontrol dan operator diatas permukaan air. Kabel *umbilical* pada ROV dapat digunakan sebagai kendali juga bisa digunakan untuk sistem peluncuran dan sistem suplai tenaga listrik serta hidrolik.

Melalui kabel *umbilical*, tenaga listrik dan hidrolik, juga perintah-perintah, atau sinyal-sinyal kontrol, disampaikan dari ruang kontrol ke ROV, secara dua arah. ROV dilengkapi dengan peralatan atau sensor tertentu seperti kamera video, robot manipulator, kompas dan lain-lain tergantung keperluan dan tujuan surveinya (Bachmayer, 1999) seperti pada gambar 2.1.

2.3 Arduino MEGA

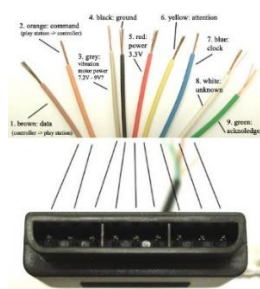
Arduino mega 2560 adalah sebuah *board* mikrokontroler yang didasarkan pada ATmega 2560. Mikrokontroler ATmega 2560 mempunyai 54 pin digital *input/output* (14 di antaranya dapat digunakan sebagai *output* PWM), 16 *input* analog, 4 UARTs (*hardware serial port*), sebuah *osilator* kristal 16 MHz, sebuah koneksi USB, sebuah *power jack*, sebuah ICSP *header*, dan sebuah tombol reset. Bentuk fisik dari ATmega 2560 ditunjukkan pada gambar 2. 2.



Gambar 2. 2 Board Arduino

2.4 Joystick PS2

Joystick PS2 adalah *controller game* dari *Playstation 2* produksi dari Sony. *Joystick* PS2 mempunyai 16 tombol utama dan mempunyai 4 data analog yang beroperasi pada tegangan 5 Volt. Dengan *joystick* PS2 ini bisa diaplikasikan ke berbagai macam operasi misalnya pada kontrol robot secara manual. Gambar 2.3 merupakan pin *output* dari *joystick* PS2 yaitu SPI MISO, MOSI, SCK, Reset dan sebuah external Pull Up untuk pin PSX ATTN untuk mode komunikasi serial.



Gambar 2. 3 Pinout Joystick PS2

2.5 Gyroscope

Giroskop digunakan untuk mengukur orientasi berdasarkan prinsip momentum sudut. Giroskop mengukur kecepatan sudut kerangka acuan inersia. Sudut orientasi berupa gerak *roll*, *pitch*, dan *yaw* didapatkan dengan mengintegrasikan kecepatan sudut. Sehingga hasil dari giroskop adalah percepatan sudut.

Giroskop berbeda dengan akselerometer dan kompas. Akselerometer mengukur gerakan linier acuan gravitasi. Akselerometer dapat memberikan pengukuran sudut kemiringan (*tilt*). Saat sistem berotasi atau bergerak, akselerometer tidak dapat mengikuti pergerakan yang cepat dikarenakan responnya lambat dan memiliki noise, sehingga tidak dapat digunakan untuk pengukuran sudut orientasi dalam pergerakan lengan manusia.

Sedangkan kompas mengukur gerakan linier dengan acuan medan magnet bumi. Kompas dapat mengukur gerak *yaw* atau arah mata angin namun tidak dapat mengukur gerak *roll* dan *pitch*.

Keluaran giroskop berupa data kecepatan sudut. Kecepatan sudut adalah besaran vektor yang menyatakan frekuensi sudut suatu benda terhadap sumbu putarnya. Satuan untuk kecepatan sudut adalah radian per detik.

2.6 Bilgepump Motor

Bilgepump motor merupakan pompa air yang biasa digunakan untuk menyedot air dalam perahu yang diletakan didasar lambung kapal, sehingga air yang masuk ke dalam lambung kapal dapat dipompa keluar. *Bilgepump motor* merupakan tipe *submersible pump* seperti gambar 2.4 yang mampu bertahan di dalam air karena memiliki pembungkus kedap air (Gomes, 2005).

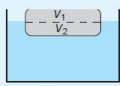




Gambar 2. 4 Motor *Bilgepump*

Pada tugas akhir ini digunakan 4 buah *bilgepump* motor sebagai penggerak utama, dengan spesifikasi 1100 gph dengan tegangan masukan sebesar 12 volt dan arus maksimal adalah sebesar 3 A, yang akan dilengkapi baling-baling berdaun empat pada setiap motornya.

2.7 Hukum Archimedes

Hukum Archimedes adalah sebuah hukum mengenai prinsip pengapungan di atas zat cair. Ketika sebuah benda tercelup sepenuhnya di dalam zat cair, maka akan ada gaya ke atas yang ditimbulkan oleh zat cair pada benda yang besarnya sama dengan berat zat cair yang dipindahkan.

Mengapung	Melayang	Tenggelam
		
$\rho_b < \rho_c$ $w = m \cdot g$ $= V \cdot \rho_b \cdot g$ $F_A = V_2 \cdot \rho_c \cdot g$	$\rho_b = \rho_c$ $w = m \cdot g$ $= V \cdot \rho_b \cdot g$ $F_A = V \cdot \rho_c \cdot g$	$\rho_b > \rho_c$ $w = m \cdot g$ $= V \cdot \rho_b \cdot g$ $F_A = V \cdot \rho_c \cdot g$

Gambar 2. 5 Hukum Archimedes

Hukum ini ditemukan oleh seorang ilmuwan yang bernama Archimedes seorang astronom, fisikawan, matematikawan dan juga insinyur berkebangsaan Yunani. Gambar 2.5 menunjukkan bagaimana hukum Archimedes bekerja pada saat benda memasuki air.

Berikut ini adalah syarat benda dikatakan mengapung, tenggelam, dan melayang :

1. Mengapung : volume benda harus lebih kecil dari volume air yang dipindahkan.
2. Melayang : volume benda harus sama dengan volume air yang dipindahkan.
3. Tenggelam : volume benda harus lebih besar dari volume air yang dipindahkan.

2.8 Driver Motor BTS7960

Pada *driver* motor DC ini dapat mengeluarkan arus hingga 43A, dengan memiliki fungsi kontrol kecepatan motor dengan menggunakan PWM hingga 25 kHz. Tegangan sumber DC yang dapat diberikan antara 5.5V-27V DC, sedangkan tegangan input level antara 3.3V-5V. *Driver*

motor ini menggunakan rangkaian *full H-bridge* dengan IC BTS7960 dengan perlindungan saat terjadi panas dan arus berlebihan. Modul *driver* ini sangat mudah di *interface*-kan menggunakan mikrokontroler ataupun Arduino.

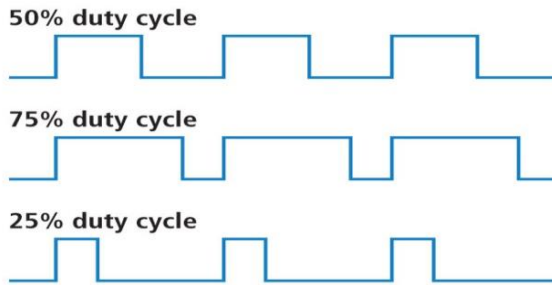


Gambar 2. 6 *Driver* Motor BTS7960

Dalam penggunaannya modul *driver* ini memiliki pin 5V dan GND yang dihubungkan dengan pin 5V dan GND pada Arduino. Dan untuk pin R_ EN dan L_EN akan dihubungkan pada Arduino. Bentuk fisik dari *driver* motor BTS7960 ditunjukkan pada gambar 2.6.

2.9 Pulse Width Modulation (PWM)

Pulse width modulation (PWM) secara luas digunakan di dalam aplikasi elektronika daya untuk pengaturan pengkonversian daya (DC/DC, DC/AC, dll.). Secara sederhana, PWM merupakan sinyal yang lebar pulsa yang bernilai “HIGH” dalam satu periode mewakili suatu tegangan DC tergantung nilai *duty cycle*. *Duty cycle* merupakan perbandingan lama waktu sinyal bernilai “HIGH” dengan satu periode. Gambar 2.7 menggambarkan *duty cycle* dari PWM. Selain digunakan dalam pengaturan pengkonversian daya, ada beberapa aplikasi lain dari PWM. Contoh aplikasi umum yang lain adalah pengendalian kecepatan motor DC, pengendalian motor servo, pengaturan nyala terang LED dan lain sebagainya (Bobby, 2015).



Gambar 2. 7 Duty Cycle Pada PWM

2.10 Baterai (*Accu*)

Baterai atau aki adalah sebuah sel listrik dimana didalamnya berlangsung proses elektrokimia yang *reversible* (dapat berbalikan) dengan efisiensinya yang tinggi. Proses elektrokimia *reversible* merupakan proses perubahan kimia menjadi tenaga listrik (proses pengosongan), dan sebaliknya dari tenaga listrik menjadi tenaga kimia, pengisian kembali dengan cara regenerasi dari elektroda-elektroda yang dipakai, yaitu dengan melewati arus listrik dalam arah (polaritas) yang berlawanan di dalam sel. Baterai dibuat untuk mensuplai listrik ke sistem starter mesin, sistem pengapian, lampu-lampu dan komponen kelistrikan lainnya.



Gambar 2. 8 Aki Yuasa 12 V, 17,2 Ah

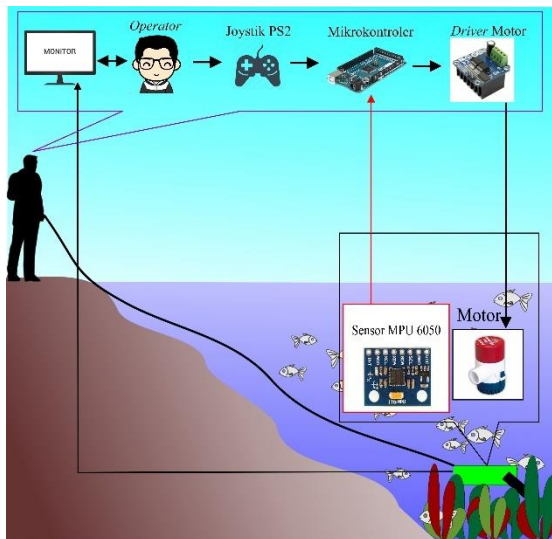
Jumlah tenaga listrik yang disimpan dalam baterai dapat digunakan sebagai sumber tenaga listrik tergantung pada kapasitas baterai dalam satuan amper jam (AH). Jika pada kotak baterai tertulis 12 volt 17,2 Ah seperti gambar 2.8, berarti baterai tersebut mempunyai tegangan 12 volt yang jika baterai tersebut digunakan selama 1 jam dengan arus pemakaian 17,2 Ah.

-----Halaman ini sengaja dikosongkan-----

BAB III PERANCANGAN SISTEM

Pada bab ini akan dijelaskan perancangan sistem mulai dari perancangan perangkat keras hingga perancangan perangkat lunak. ROV yang dirancang bertujuan untuk melakukan misi pengantaran terumbu karang. Perangkat keras yang digunakan antara lain, arduino mega, *driver* motor BTS7960, motor *bilgepump*, *joystick* PS2.

Sensor yang digunakan merupakan sensor MPU 6050 *gyroscope* untuk menentukan kemiringan dari ROV ketika bergerak didalam air. Perangkat lunak yang digunakan adalah program yang dibuat pada *software* Arduino. Bab ini menjelaskan secara keseluruhan sistem dan desain tiap-tiap blok yang ada. ROV didesain dengan sumber aki 12 Volt 17,5Ah dan 4 buah motor *bilgepump*.



Gambar 3. 1 Perancangan Sistem

Perancangan sistem seperti pada gambar 3.1, *user* atau *operator* akan mengendalikan ROV yang berada dibawah air menggunakan *joystick*, Bergeraknya ROV didalam air dapat dilihat melalui monitor yang akan menampilkan kemiringan ROV dari sensor *gyroscope*. ROV akan

bergerak maju, mundur, belok kanan, belok kiri, naik, dan turun dengan menggunakan 4 buah motor *bilgepump* yang diatur kecepatannya menggunakan *driver* motor BTS7960.

Penempatan motor secara *vertical* dua buah untuk gerakkan naik dan turun, sedangkan dua buah lainnya akan diletakkan secara *horizontal* untuk gerakkan maju, mundur, belok kanan dan belok kiri.

3.1 Perancangan Sistem

Pada tugas akhir ini menggunakan mikrokontroler arduino mega sebagai unit pengolahan data. Sensor yang digunakan pada tugas akhir ini adalah sensor MPU 6050 yang menggunakan komunikasi I2C (*Inter-Integrated Circuit*), sensor MPU diletakkan didalam tabung ROV. Informasi yang dihasilkan sensor MPU berupa besaran sudut *roll*, sudut *pitch*, dan sudut *yaw* yang digunakan untuk melihat kemiringan ROV ketika bergerak didalam air.

Keluaran sensor MPU 6050 akan ditampilkan dilayar, kemudian memberikan informasi terkait kemiringan ROV ketika berada didalam air. Pilot yang mengendalikan ROV akan membenarkan kestabilan melalui *joystick*. *Joystick* akan menjadi kontrol dari motor *bilgepump* yang akan dikontrol kecepatannya dan arah putarannya menggunakan *driver* motor BTS7960.

Sistem pengiriman data menggunakan media kabel dengan menghubungkan unit pengolahan yang terletak pada robot dengan *remote* kontrol.

3.2 Perancangan Perangkat Keras

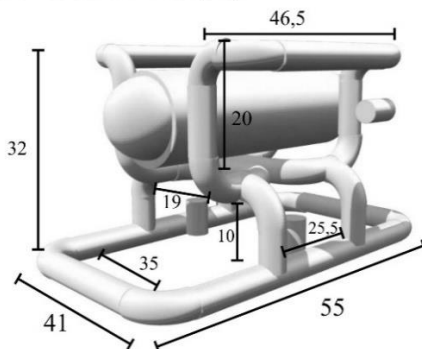
Perangkat keras pada tugas akhir ini adalah desain rangka ROV dan *wiring* elektronik. Desain rangka ROV berupa desain badan utama dari ROV. Desain elektronik meliputi *wiring* perangkat *driver* motor, kontrol *joystick* dan peletakan sensor MPU 6050.

3.2.1 Desain ROV

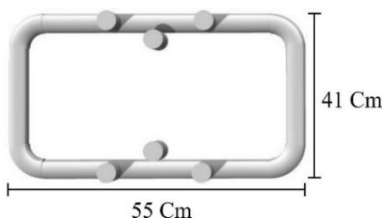
Robot didesain dengan *software 3D Builder* seperti gambar 3.2, yang kemudian akan direalisasikan menggunakan pipa paralon. Dimensi dari robot yang akan dibuat dengan ukuran panjang 54 cm, lebar 40 cm, tinggi 32 cm. Pada bagian kaki ROV dibuat dengan menggunakan pipa paralon, berbentuk persegi panjang 55 cm serta lebar 41 cm, dengan bentuk kaki persegi panjang diharapkan dapat mempermudah ROV mendarat di dasar laut seperti gambar 3.3.

Pada bagian atas terdapat dua penyangga untuk tabung tempat sensor MPU 6050 yang memiliki panjang 19 cm. ROV juga memiliki penyangga sayap pada bagian kanan dan kiri dengan tinggi 20 cm serta sayap pada bagian atas dengan panjang total 40 cm seperti gambar 3.4 yang akan berguna untuk menjaga keseimbangan ROV ketika bergerak di dalam air serta menambah daya apung ROV di dalam air.

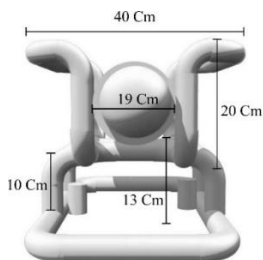
Satuan Dalam Centimeter (CM)



Gambar 3. 2 Rancangan Desain ROV



Gambar 3. 3 Rancangan Desain Kaki ROV



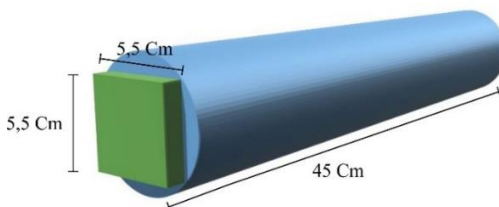
Gambar 3. 4 Rancangan Desain ROV Tampak Depan



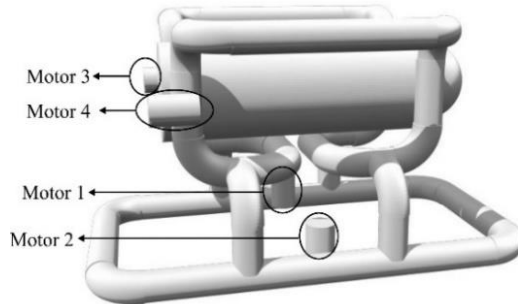
Gambar 3. 5 Rancangan Desain Tempat Sensor

Peletakkan sensor MPU 6050 berada pada tabung yang memiliki diameter 9 cm serta tinggi tabung 52 cm seperti pada gambar 3.5. Dalam tabung akan diberi kotak akrilik yang akan menambah perlindungan dari air dengan ukuran panjang 5,5 cm lebar 45 cm serta tinggi 5,5 cm, seperti pada gambar 3.6.

Sedangkan tabung besar akan diberi tutup paralon sehingga dapat menutupi pipa paralon. ROV dilengkapi dengan empat buah motor, dua motor diletakkan secara *vertical* yaitu motor 1 dan 2 di kaki ROV, sehingga ROV dapat bergerak naik dan turun, dua motor yaitu 3 dan 4 diletakkan secara *horizontal* di samping sayap ROV sehingga dapat bergerak maju, mundur, belok kiri, dan belok kanan seperti pada gambar 3.7.



Gambar 3. 6 Desain Kotak Sensor



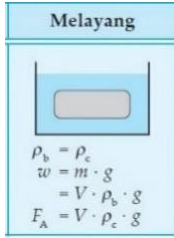
Gambar 3. 7 Posisi Motor

Bentuk realisasi tubuh ROV yang sudah dibuat menggunakan pipa paralon dapat dilihat pada gambar 3.8.



Gambar 3. 8 Bentuk ROV

Desain ROV dibuat agar robot ketika dimasukkan kedalam air dalam kondisi melayang. Kondisi melayang akan memudahkan ROV untuk tenggelam dan mengapung, karena dorongan yang diberikan motor tidak besar dan tidak membebani motor. Untuk membuat ROV dapat melayang maka menggunakan hukum archimedes dimana memiliki rumus bahwa gaya apung (F_a) sama dengan berat benda ketika dalam air (W), seperti gambar 3.9.



Gambar 3. 9 Hukum Archimedes saat Kondisi Melayang

3.2.2 Persamaan Gaya Apung ROV

Gaya apung adalah gaya yang berlawanan arah dengan gravitasi, yang tenggelam dalam fluida. Saat mempengaruhi semua benda yang diletakkan dalam cairan, massa benda menekan fluida (cairan atau gas), sedangkan gaya apung ke atas menekan benda, berperan melawan gravitasi.

$$F_a = M_f \times g \dots\dots\dots(3.1)$$

$$F_a = \rho_f \times V_{bf} \times g \dots\dots\dots(3.2)$$

F_a = gaya Apung (N)

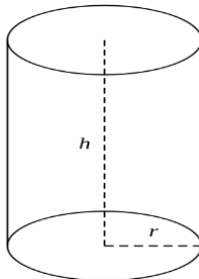
M_f = massa zat cair yang dipindahkan benda (kg/m^3)

g = percepatan gravitasi bumi (m/s^2)

ρ_f = massa jenis zat cair (kg/m^3)

V_{bf} = volume benda yang tercelup dalam za cair (ℓ)

Robot menggunakan tabung dengan ukuran jari – jari 4,5 cm dan tinggi 52 cm untuk bagian tempat sensor. Dengan menggunakan rumus volume tabung pada persamaan 3.3 :



Gambar 3. 10 Volume Tabung

$$V = \pi r r r r r t \dots\dots\dots(3.3)$$

$\pi = 3.14$

$r = \text{jari – jari tabung (cm)}$

$t = \text{tinggi tabung (cm)}$

Berdasarkan rumus pada persamaan 3.3, maka volume tabung didapatkan dengan perhitungan $3,14 \times 4,5 \times 4,5 \times 52 = 3306,42 \text{ cm}^3$ atau volume dapat dikonversi ke dalam satuan m^3 yaitu $3306,42 \text{ cm}^3 = 0,00330642 \text{ m}^3$.

Setelah mendapatkan nilai volume maka selanjutnya nilai volume akan dimasukkan ke dalam rumus gaya apung (F_a). Berdasarkan rumus gaya apung yaitu $F_a = Vt \times \rho \times g$. Maka nilai gaya apung adalah $0,00330642 \times 1000 \times 0,98 = 3,240 \text{ N}$.

Persamaan Berat ROV digunakan untuk mengetahui berat robot (W). W dapat dirumuskan pada persamaan 3.4 dimana m adalah massa benda dan g adalah percepatan gravitasi.

$$W = m \times g \dots\dots\dots(3.4)$$

$W = \text{Berat Benda (N)}$

$m = \text{Massa Benda (kg)}$

$g = \text{Percepatan Gravitasi Bumi (m/s}^2\text{)}$

ROV memiliki massa sebesar 3,3 kg sehingga berdasarkan rumus diatas maka berat robot adalah $3,3 \times 0,98 = 3,234 \text{ N}$.

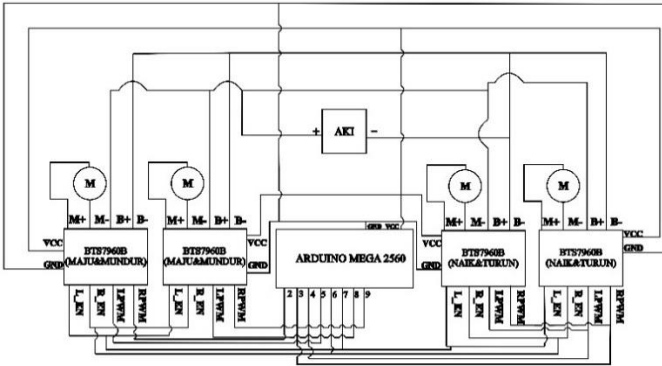
Persamaan ROV dalam kondisi melayang adalah ketika gaya apung (F_a) sama dengan berat benda (w). Nilai gaya apung (F_a) didapatkan sebesar 3,240 N sedangkan berat benda (w) didapatkan sebesar 3,234 N. Menurut persamaan archimedes yaitu $F_a > w$ maka persamaan archimedes untuk robot adalah $3,240 \text{ N} > 3,234 \text{ N}$.

Dengan persamaan tersebut dinyatakan bahwa gaya apung (F_a) lebih besar dari berat benda (w) maka robot akan dikatakan tenggelam. Selisih nilai antara gaya apung dan berat benda hanya 0,007 N jadi robot hanya akan tenggelam sedikit ketika melayang di dalam air sehingga dapat dikatakan ROV sudah bisa melayang di dalam air.

3.2.3 Wiring Driver Motor

Perangkat kontrol pada ROV merupakan arduino mega dengan *driver* motor BTS7960 untuk mengkontrol arah putran dari motor *bilgepump*. Motor *bilgepump* berfungsi menggerakkan ROV maju,

mundur, naik, turun, belok kanan dan belok kiri. PWM berfungsi sebagai pengontrol kecepatan pada motor *bilgepump* sesuai dengan program. Perancangan ini menggunakan 1 buah aki 12V, 17,2 AH serta 4 buah motor *bilgepump* yang dirancang secara *parallel*.

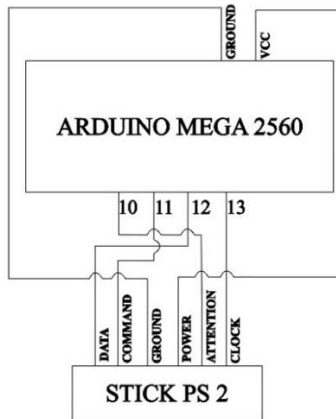


Gambar 3. 11 *Wiring Driver Motor BTS7960 dengan Arduino*

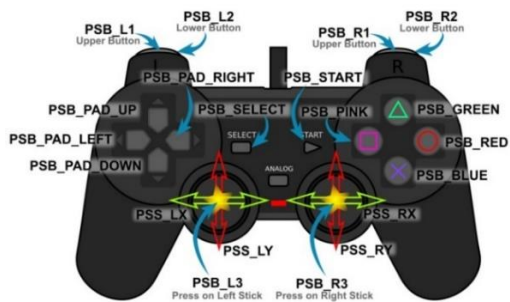
Perancangan ini bertujuan untuk memberi daya pada motor *bilgepump* yang membutuhkan *supply* tegangan sebesar 12V untuk setiap motornya. Kemudian penggunaan *driver* BTS7960B (maju dan mundur) digunakan untuk mengatur arah putar dan PWM dari motor ROV bergerak maju dan mundur. Begitu juga penggunaan *driver* BTS7960B (naik dan turun) digunakan untuk mengatur arah putar dan PWM dari ROV bergerak naik dan turun. Pengaturan PWM belok kanan dan belok kiri menggunakan *driver* BTS7960B (maju dan mundur) berdasarkan program gerakan ROV belok kanan dan belok kiri. *Wiring* perangkat kontrol ROV bisa dilihat pada gambar 3.11.

3.2.4 *Wiring Remote Kontrol*

Wiring remote kontrol yang digunakan untuk menggerakkan ROV pada tugas akhir ini menggunakan *joystick* PS2. Keterangan tombol pada *joystick* yang digunakan dapat dilihat pada gambar 3.12.



Gambar 3. 12 Wiring Stick PS2 dengan Arduino Mega



Gambar 3. 13 Keterangan Tombol Joystick PS2

Pin joystick PS2 yang digunakan merupakan pin data, *command*, *attention*, *clock*, *ground* serta *vcc* yang akan dihubungkan pada arduino mega yang ditunjukkan pada gambar 3.12. Konfigurasi kontrol tiap tombol joystick PS2 berdasarkan gambar 3.13 dan fungsi untuk gerakkan ROV ditunjukkan pada tabel 3.1 berikut.

Tabel 3. 1 Konfigurasi Tombol Remote Kontrol Gerak ROV

Keterangan Tombol	Gerakkan ROV
PSS_LY	Naik/Turun
PSS_RX	Belok Kanan/Belok Kiri
PSB_RY	Maju/Mundur

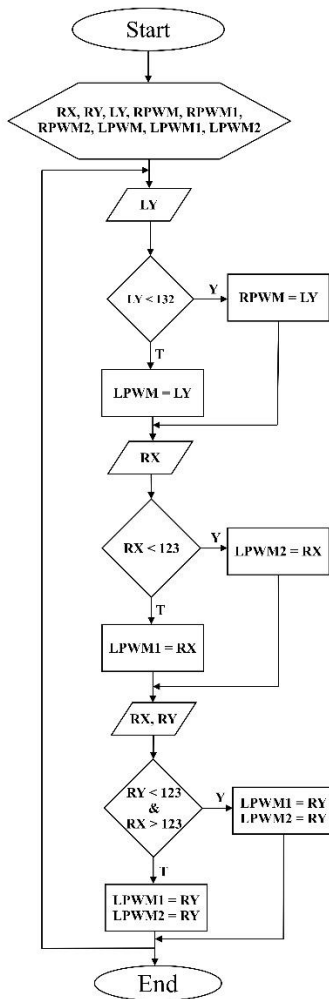
3.3 Perancangan Perangkat Lunak

Pada tahap ini, algoritma pemrograman dirancang untuk melakukan dua bagian. Bagian pertama adalah perancangan kontrol *joystick* ke robot. Bagian kedua adalah perancangan sensor membaca kemiringan posisi robot.

3.3.1 Perancangan Gerak ROV

Gerakkan ROV akan dikendalikan dengan *joystick* dengan program berdasarkan *flowchart* gambar 3.15. Pertama program akan membaca *joystick* terlebih dahulu, kemudian barulah akan membaca perintah dari analog *joystick* dengan kode PSS_LY, PSS_RX, dan PSS_RY. Peletakkan motor 1 dan 2 untuk gerakkan naik dan turun sedangkan motor 3 dan 4 untuk maju, mundur, belok kanan, dan belok kiri seperti pada gambar 3.7. Nilai yang di terapkan pada program merupakan nilai yang diambil saat pembacaan analog *joystick* PS2 pada serial monitor. Kemudian membuat program berdasarkan *flowchart* seperti di gambar 3.14.

Saat nilai dari analog PSS_LY < 132 maka motor 1 dan 2 akan bergerak searah jarum jam dan akan membuat ROV turun, saat PSS_LY > 132 maka akan membalikkan arah motor yaitu naik. Motor 3 dan 4 untuk gerakkan maju ketika nilai analog PSS_RY > 123 dan PSS_RX < 123 sedangkan untuk gerakkan mundur nilai analog PSS_RY < 123 dan PSS_RX > 123. Gerakkan belok kanan ketika nilai PSS_RX < 123 maka motor 4 yang akan bergerak, sedangkan ketika nilai PSS_RX > 123 maka motor 3 yang akan menggerakkan ROV untuk belok kiri.

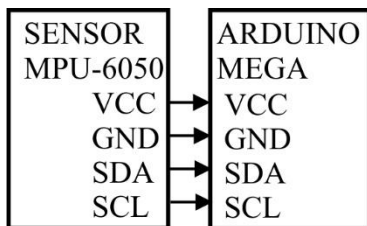


Gambar 3. 14 Flowchart Program Gerak ROV

3.3.2 Pembacaan Sensor MPU 6050

Pembacaan sensor MPU 6050 untuk mengetahui nilai kemiringan dari ROV ketika bergerak di dalam air. Keluaran sensor ini menghasilkan

nilai mentah yang biasa disebut *raw* data yang terdiri dari *ax*, *ay*, *az*, *gx*, *gy*, dan *gz* (*a* untuk akselerometer dan *g* untuk giroskop). Pada sensor ini diinginkan mengeluarkan nilai sudut *pitch*, yang merupakan sudut yang bekerja pada sumbu *y*. *Wiring* sensor dengan arduino dapat dilihat pada gambar 3.15.



Gambar 3. 15 Koneksi MPU 6050 Dengan Arduino

BAB IV

PENGUJIAN DAN ANALISIS DATA

Pada bab ini akan dibahas mengenai pengujian dari sistem yang telah dirancang. Bab ini bertujuan untuk mendapatkan data yang kemudian dilakukan analisa pada masing-masing pengujian, pada gambar 4.1 merupakan realisasi robot yang telah dibuat.

Pengujian yang dilakukan berupa :

1. Pengujian beban maksimal rangka ROV.
2. Pengujian tempat sensor.
3. Pengujian sensor MPU6050 *gyroscope*.
4. Pengujian kontrol *joystick* PS2.
5. Pengujian gerakan ROV didalam air.
6. Pengujian kecepatan ROV didalam air.
7. Pengujian keseluruhan ROV didalam air.



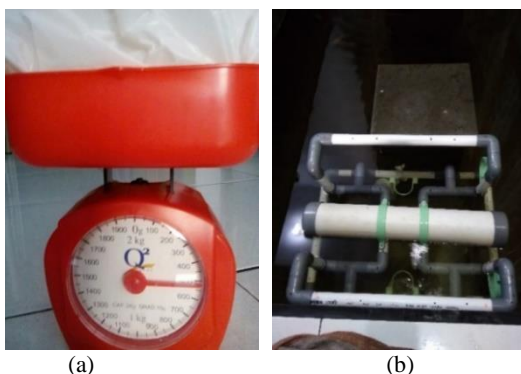
Gambar 4. 1 Realisasi ROV

4.1 Pengujian Beban Maksimal Rangka ROV

Pengujian beban ROV dilakukan untuk mengetahui beban maksimal yang dapat diangkat rangka ROV yang dibuat. Caranya dengan memberikan beban berupa batu yang sudah ditimbang sesuai dengan kebutuhan. Pertama dilakukan perhitungan volume total rangka ROV karena rangka ROV terbuat dari pipa paralon maka dapat digunakan rumus volume tabung untuk mengetahui volume total rangka ROV.

Diameter dari pipa-pipa rangka ROV berdiameter 2,5 cm dengan Panjang yang berbeda-beda seperti pada gambar 3.2, sehingga dilakukan perhitungan manual satu-persatu bagian dari rangka ROV.

Total volume rangka ROV keseluruhan adalah 5.516,69 cm³. Dapat dikonversikan menjadi liter maka didapat 5,51669 L, kemudian diberi beban mulai dari 1 kg sampai ROV dapat tenggelam ke dasar kolam. Beban pertama yang diberikan berupa batu yang sudah di timbang seberat 1 kg dan dimasukkan pada kantong plastic dan setiap kantongnya 500 gram seperti pada gambar 4.2 (a), kemudian beban akan diikat pada sisi kanan dan kiri kaki ROV.

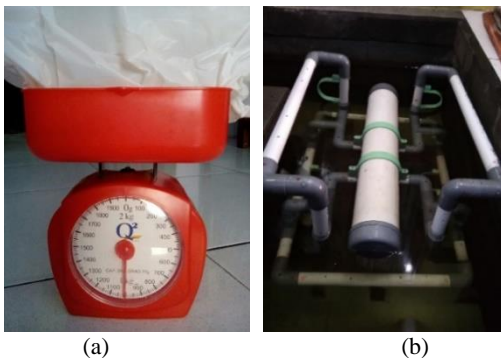


Gambar 4. 2 (a) Berat Beban 500 gram, (b) Pengujian Rangka dengan Beban 1 kg

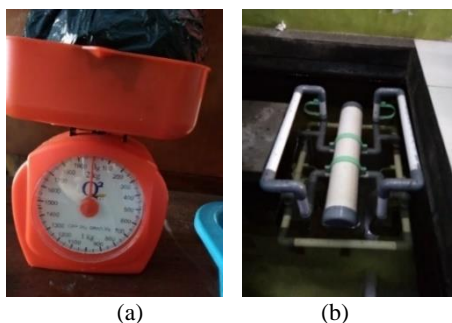
Kemudian dilihat hasilnya ternyata ROV terapung dengan sedikit bagian tempat komponen yang tenggelam. Dilanjutkan dengan pemberian beban total 2 kg dengan menambahkan batu setiap kantong menjadi 1 kg seperti pada gambar 4.3 (a). Hasilnya rangka ROV masih setengah yang tenggelam seperti pada gambar 4.3 (b). Beban ditambahkan menjadi total berat beban 4 kg dengan memberikan tambahan masing-masing beban menjadi 2 kg seperti pada gambar 4.4 (a). Hasilnya dapat dilihat pada gambar 4.4 (b) dimana rangka ROV tenggelam lebih sedikit dari beban 2 kg sebelumnya.

Kemudian rangka ROV diberikan beban dengan total 7,6 kg sehingga seperti pada gambar 4.5 dimana semua rangka ROV sudah tenggelam ke dasar kolam. Sehingga dapat disimpulkan bahwa perhitungan hukum arcimedes tenggelam telah sesuai karena untuk

tenggelam maka diperlukan nilai $F_a > W$ dimana nilai berat dari total yang dapat di bawa ROV sebesar 2 kali lipat dari gaya angkat ROV yang dijelaskan pada perhitungan sub-bab 3.2.2 berat total yang mampu dibawa rangka ROV adalah 7,6 kg. Desain kaki yang dibuat memberikan kemudahan ROV untuk mendarat di dasar kolam dengan membuat ROV tegak lurus dengan kolam.



Gambar 4. 3 (a) Berat Beban 1 kg, (b) Pengujian Rangka dengan Beban 2 kg



Gambar 4. 4 (a) Berat Beban 2 kg, (b) Pengujian Rangka dengan Beban 4 kg

Sehingga tidak membuat ROV miring ketika mendarat di dasar kolam. Lubang pada beberapa bagian ROV juga membantu memberikan beban pada ROV dengan masuknya air akibat tenggelam, dimana ini merupakan penerapan sistem *ballast* seperti kapal selam. Dimana untuk menambahkan beban ROV air masuk ke dalam rangka ROV dan akan membuat ROV tenggelam.

Ketika akan naik ke permukaan maka air akan keluar melalui lubang-lubang yang ada dirangka ROV dan akan membuat ROV naik ke permukaan. Dengan adanya sayap pada bagian atas ROV, membuat ROV ketika berada didalam air menjadi seimbang saat diberi beban seperti gambar 4.5 dan sebelum diberi beban seperti pada gambar 4.6.



Gambar 4. 5 Hasil Rangka dengan Beban 7,6 kg



Gambar 4. 6 Rangka ROV Tanpa Beban

4.2 Pengujian Tempat Sensor

Pengujian dilakukan dengan melakukan tes pada tempat sensor yang didalamnya terdapat pelindung akrilik tambahan untuk melindungi sensor dari masuknya air dari luar seperti gambar 4.7. Perlindungan juga dilakukan dengan memberikan lakban pada kedua ujung pipa paralon tempat sensor sehingga akan mencegah adanya air masuk dari tutup

paralon serta dengan memberikan lem akuarium di akrilik yang ada didalam tabung tempat sensor.

Kemudian tempat sensor ditenggelamkan bersamaan dengan pengujian berat maksimal rangka ROV dan dilihat kedalam tabung tempat komonen terdapat kebocoran atau tidak. Hasil dari pengujian *waterproof* didapatkan bahwa sensor terlindungi dan tidak ada air yang masuk kedalam tabung dan akrilik tersebut.



Gambar 4. 7 Pengujian Tempat Sensor

4.3 Pengujian Sensor MPU 6050 Gyroscope

Pengujian ini bertujuan untuk mengetahui tingkat keakuratan pembacaan sensor MPU 6050 untuk membaca kemiringan sudut ROV. Pengujian ini dilakukan pada IMU MPU6050 yang di dalamnya terdapat sensor *gyroscope* untuk mengetahui besar sudut yang bekerja pada sumbu y (sudut *pitch*).



Gambar 4. 8 Pengujian Sensor *Gyroscope*

Pengujian ini dilakukan dengan memiringkan sensor yang ada didalam ROV seperti pada gambar 4.8 kemudian melihat hasil pembacaan sudut kemiringan sensor melalui serial monitor dan membandingkannya dengan besar sudut yang sebenarnya menggunakan busur. Data hasil pengujian sensor terhadap sudut *pitch* ditunjukkan oleh tabel 4.1.

Tabel 4. 1 Data Pengujian Sudut *Pitch* Sensor

Pembacaan Busur (<i>deg</i>)	Pembacaan Sensor (<i>deg</i>)	<i>Error</i> (%)
10	10,11	1,1 %
20	20,23	1,15%
30	30,21	0,7%
40	40,13	0,33%
50	50,18	0,36%
60	60,21	0,35%
70	70,30	0,19%
80	80,19	0,24%
90	90,25	0,27%
<i>Error Rata-Rata</i>		0,52%

Dari tabel 4.1 dapat diketahui bahwa sensor mampu mengukur sudut *pitch* dengan *error* rata-rata adalah 0,52%. Dari pengujian tersebut dapat diketahui bahwa kesalahan pembacaan sudut dari sensor tidak lebih dari 1%, sehingga sensor masih dapat berfungsi dengan baik. Sehingga akan memudahkan *operator* untuk membetulkan posisi ROV ketika tidak stabil didalam air.

4.4 Pengujian Kontrol *Joystick* PS2

Pengujian ini dilakukan dengan membuat program untuk membaca perintah *joystick* PS2 oleh arduino mega 2560, kemudian akan dibuat program pada analog *joystick* PS2 untuk menggerakkan motor DC pada ROV sesuai arah yang diinginkan yaitu naik, turun, maju, mundur, belok kanan dan belok kiri.

Pertama dilakukan *wiring joystick* dengan arduino mega seperti pada gambar 3.12, kemudian *upload* program yang dibuat pada *software* arduino. Setelah itu lihat pada serial monitor *software* arduino dan atur *bound rate* 57600. Tekan tombol *joystick* maka akan muncul perintah bahwa tombol telah ditekan seperti pada gambar 4.9.

```

Triangle pressed
Circle just pressed
Square just released
X just changed
X just changed
R2 pressed
L2 pressed
Up held this hard: 0
Up held this hard: 0
LEFT held this hard: 0
LEFT held this hard: 0
Right held this hard: 0
Right held this hard: 0
DOWN held this hard: 0
DOWN held this hard: 0
Start is being held
Start is being held
Start is being held
Start is being held
Select is being held
Select is being held
Select is being held
Stick Values:132,123,123,123
Stick Values:0,123,123,123
Stick Values:0,123,123,123
Stick Values:132,123,123,123
Stick Values:132,123,0,0

```

Gambar 4. 9 Hasil Baca Perintah *Joystick* PS2

Terdapat nilai yang muncul ketika menggerakkan analog *joystick* yaitu *stick values*: 132, 123, 123, 123 nilai ini yang akan digunakan sebagai penggerak dari motor DC pada ROV, pada *stick values* nilainya memiliki arti dari setiap angkanya secara berurutan merupakan nilai dari *stick values*: LY, LX, RY, RX. Kemudian pengujian dilanjutkan dengan menghubungkan perintah joystick dengan arduino mega yang akan memberikan perintah *driver* motor untuk menggerakkan motor DC sesuai keinginan.

Tabel 4. 2 Pengujian Arah Putaran Motor ROV dengan *Joystick*

Perintah <i>Joystick</i>	Motor ROV	Arah Putaran Motor DC	Gerakkan ROV
LY < 132	Motor 1 dan 2	CW	Naik
LY > 132	Motor 1 dan 2	CCW	Turun
RX > 123	Motor 3	CCW	Belok Kanan
RX < 123	Motor 4	CCW	Belok Kiri
RY < 123 dan RX > 123	Motor 3 dan 4	CCW	Maju
RY > 123 dan RX < 123	Motor 3 dan 4	CW	Mundur

Analog pada *joystick* digerakkan sesuai dengan program yang dibuat berdasarkan gambar 3.14, kemudian dilihat arah gerakan baling-baling dari motor DC sesuai dengan gerakan analog. Dengan program yang telah dibuat kemudian dilakukan percobaan menggerakkan motor DC pada ROV menggunakan *joystick* sehingga didapatkan hasil seperti pada tabel 4.2

Pada gerakan maju dan mundur dilakukan penggabungan antara perintah RY dan RX dikarenakan saat dilakukan pemrograman dengan perintah RY untuk gerakan maju dan mundur motor DC tidak dapat berputar dengan baik karena adanya tabrakan data antara sinyal RY dan RX. Sehingga dibuat penggabungan nilai RY dan RX sehingga sinyal perintah dapat menggerakkan ROV maju dan mundur dengan baik.

Seharusnya gerakan analog untuk maju adalah analog *right* ke atas dan gerakan analog untuk mundur adalah analog *right* ke bawah, namun karena ada penggabungan perintah antara RX maka saat gerakan ROV maju maka gerakan analog *right* serong ke atas kanan sedangkan gerakan ROV mundur maka gerakan analog *right* serong ke bawah kiri.

4.5 Pengujian Gerakan ROV Didalam Air

Pengujian gerakan ROV didalam air dilakukan dengan memasukkan ROV kedalam kolam air yang kemudian digerakkan dengan program yang dibuat berdasarkan *flowchat* gambar 3.14 serta perintah *joystick* pada tabel 4.1. Dengan menggerakkan analog *joystick* ke atas pada analog *left* gambar 4.10 (a) maka akan menggerakkan ROV naik ke atas seperti gambar 4.10 (b).

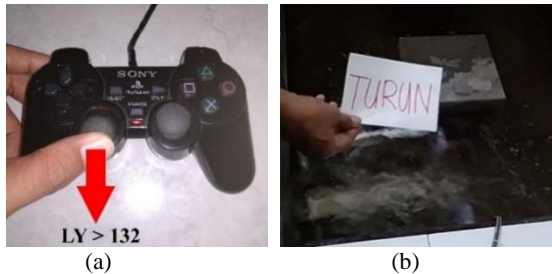


Gambar 4. 10 (a) Perintah *Joystick* Naik (b) Gerakan ROV Naik

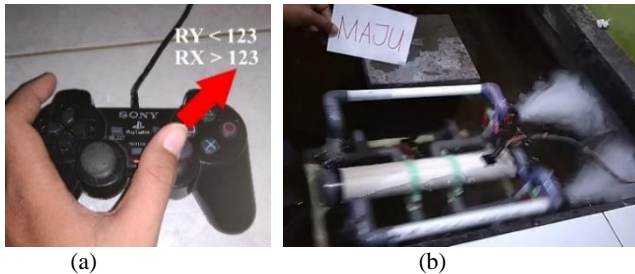
Gerakan turun berdasarkan perintah analog *joystick left* ke bawah seperti gambar 4.11 (a) maka ROV akan bergerak turun tenggelam ke

dasar kolam seperti gambar 4.11 (b). Gerakkan maju berdasarkan perintah analog *joystick right* serong ke atas kanan seperti gambar 4.12 (a) maka ROV akan bergerak maju dengan dua motor yang beputar secara bersamaan seperti gambar 4.12 (b).

Gerakkan mundur berdasarkan perintah analog *joystick right* serong ke bawah kanan seperti gambar 4.13 (a) maka ROV akan bergerak mundur dengan dua motor yang beputar secara bersamaan seperti gambar 4.13 (b).



Gambar 4.11 (a) Perintah *Joystick* Turun (b) Gerakkan ROV Turun



Gambar 4.12 (a) Perintah *Joystick* Maju (b) Gerakkan ROV Maju



Gambar 4.13 (a) Perintah *Joystick* Mundur (b) Gerakkan ROV Mundur

Gerakkan belok kanan berdasarkan perintah analog *joystick right* ke kanan seperti gambar 4.14 (a) maka ROV akan bergerak belok ke kanan dengan menggunakan 1 motor pendorong sebelah kiri 4.14 (b).



Gambar 4. 14 (a) Perintah *Joystick* Belok Kanan (b) Gerakkan ROV Belok Kanan

Gerakkan belok kiri berdasarkan perintah analog *joystick right* ke kiri seperti gambar 4.15 (a) maka ROV akan bergerak belok ke kiri dengan menggunakan 1 motor pendorong sebelah kanan 4.15 (b).



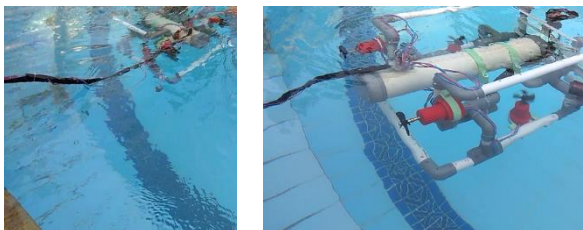
Gambar 4. 15 (a) Perintah *Joystick* Belok Kiri (b) Gerakkan ROV Belok Kiri

Dari pengujian gerakan ROV berdasarkan perintah *joystick* dapat disimpulkan bahwa gerakan sudah sesuai dengan yang diharapkan yaitu ROV dapat bergerak naik, turun, maju, mundur, belok kanan dan belok kiri menggunakan perintah *joystick*.

Terdapat juga kendala yaitu saat gerakan mundur ROV sangat lambat karena masih ada data perintah yang dibaca oleh arduino dari *joystick* masih ada benturan sehingga putaran motor saat gerakan mundur sedikit lambat.

4.6 Pengujian Kecepatan ROV Didalam Air

Pengujian kecepatan gerak ROV didalam air pada kolam renang yang memiliki panjang 2 meter, lebar 2 meter, dan kedalaman 1 meter seperti pada gambar 4.16.



Gambar 4. 16 Pengujian Kecepatan ROV

Tabel 4. 3 Pengujian Kecepatan Gerak ROV Maju

Jarak Tempuh (m)	Waktu (s)	Kecepatan (m/s)
0,5	4,3	0,12
0,8	4,6	0,17
1	5,1	0,19
1,2	5,3	0,23
1,5	5,7	0,26
Kecepatan rata-rata		0,19

Data pengujian kecepatan ROV untuk gerakan maju seperti tabel 4.3. Sehingga didapatkan kecepatan rata-rata ROV untuk Pergerakan maju adalah 0,19 m/s. Kemudian dilakukan pengujian kecepatan ROV untuk gerakan mundur, data pengujian kecepatan ROV untuk gerakan mundur seperti tabel 4.4.

Tabel 4. 4 Pengujian Kecepatan Gerak ROV Mundur

Jarak Tempuh (m)	Waktu (s)	Kecepatan (m/s)
0,5	6	0,08
0,8	6,3	0,13
1	6,6	0,15
1,2	6,8	0,18
1,5	7,2	0,21
Kecepatan rata-rata		0,15

Sehingga didapatkan kecepatan rata-rata ROV untuk Pergerakan mundur adalah 0,15 m/s. Kemudian dilakukan pengujian kecepatan ROV

untuk gerakkan naik, data pengujian kecepatan ROV untuk gerakkan naik seperti tabel 4.5.

Tabel 4. 5 Pengujian Kecepatan Gerak ROV Naik

Jarak Tempuh (m)	Waktu (s)	Kecepatan (m/s)
0,5	6,3	0,07
0,8	6,6	0,12
1	7,1	0,14
Kecepatan rata-rata		0,11

Sehingga didapatkan kecepatan rata-rata ROV untuk gerakkan naik adalah 0,11 m/s. Kemudian dilakukan pengujian kecepatan ROV untuk gerakkan turun, data pengujian kecepatan ROV untuk gerakkan turun seperti tabel 4.6.

Tabel 4. 6 Pengujian Kecepatan Gerak ROV Turun

Jarak Tempuh (m)	Waktu (s)	Kecepatan (m/s)
0,5	4,2	0,12
0,8	4,6	0,17
1	5,3	0,19
Kecepatan rata-rata		0,16

Sehingga didapatkan kecepatan rata-rata ROV untuk gerakkan turun adalah 0,16 m/s. Kemudian dilakukan pengujian kecepatan ROV untuk gerakkan belok kanan, data pengujian kecepatan ROV untuk gerakkan belok kanan seperti tabel 4.7.

Tabel 4. 7 Pengujian Kecepatan Gerak ROV Belok Kanan

Jarak Tempuh (m)	Waktu (s)	Kecepatan (m/s)
0,5	4,8	0,10
0,8	5,2	0,15
1	5,5	0,18
Kecepatan rata-rata		0,14

Sehingga didapatkan kecepatan rata-rata ROV untuk gerakkan belok kanan adalah 0,14 m/s. Kemudian dilakukan pengujian kecepatan ROV untuk gerakkan belok kiri, data pengujian kecepatan ROV untuk gerakkan belok kiri seperti tabel 4.8.

Tabel 4. 8 Pengujian Kecepatan Gerak ROV Belok Kiri

Jarak Tempuh (m)	Waktu (s)	Kecepatan (m/s)
0,5	4,9	0,10
0,8	5,3	0,15
1	5,6	0,18
Kecepatan rata-rata		0,14

Sehingga didapatkan kecepatan rata-rata ROV untuk gerakan belok kiri adalah 0,14 m/s. Kecepatan rata-rata ROV ketika bergerak maju adalah 0,19 m/s lebih cepat daripada kecepatan rata-rata ROV ketika bergerak mundur yaitu 0,15 m/s, karena ketika bergerak mundur kedua motor ROV tidak berputar ke arah yang sama secara bersamaan sehingga gerakan mundur ROV sedikit lambat .

Sedangkan kecepatan rata-rata ROV ketika naik adalah 0,11 m/s lebih lambat daripada kecepatan ROV ketika turun yaitu 0,16 m/s, karena ketika naik ROV sedikit miring ke kanan akibat dari robot manipulator yang diletakkan pada sisi kanan ROV, namun ROV masih dapat naik dengan baik. Kecepatan rata-rata dari gerakan ROV untuk belok kanan dan kiri yaitu 0,14 m/s. Sehingga dapat disimpulkan bahwa kecepatan rata-rata gerakan ROV didalam air adalah 0,15 m/s.

4.7 Pengujian Keseluruhan Sistem Didalam Air

Pengujian keseluruhan sistem dalam air dilakukan dengan cara memasukan ROV ke kolam renang dengan panjang 2 meter, lebar 2 meter dan kedalaman 1 meter seperti pada gambar 4.17.

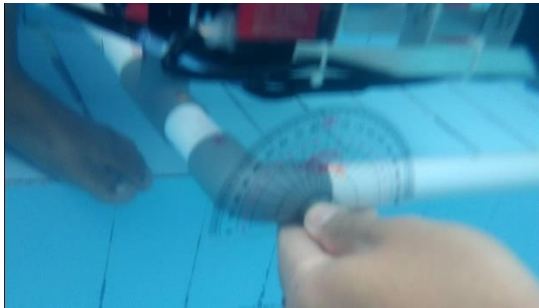


Gambar 4. 17 Pengujian ROV Didalam Air

Pengujian ini dilakukan untuk mengetahui tingkat keberhasilan sistem pembacaan dari sensor MPU 6050 serta keberhasilan ROV bergerak didalam air sesuai dengan gerakan serta perintah dari *joystick* yang dikendalikan dari atas dengan panjang kabel 1 meter , seperti pada

gambar 4.16. Saat pengujian gerak ROV didalam air dengan pengendali *joystick* sensor dihubungkan ke arduino, posisi sensor berada didalam tabung tempat sensor ROV.

Kemudian dilihat nilai kemiringan ROV terhadap sumbu y yang ditampilkan pada serial monitor. Data kemiringan ROV yang dibaca sensor akan dibandingkan dengan kemiringan ROV dengan bantuan busur untuk melihat kemiringannya seperti pada gambar 4.18.



Gambar 4. 18 Pembacaan Kemiringan ROV dengan Busur

Ketika ROV bergerak maju dengan kecepatan 0,45 m/s nilai sensor *gyroscope* menunjukkan kemiringan 9,51 *degree*, sedangkan saat dilakukan pengukuran kemiringan didalam air didapatkan kemiringan 10 *degree*. Ketika ROV bergerak mundur dengan kecepatan 0,33 m/s nilai sensor menunjukkan kemiringan 9,78 *degree*, sedangkan pengukuran menggunakan busur menunjukkan kemiringan 10 *degree*. Pergerakan ROV saat naik dengan kecepatan 0,13 m/s nilai sensor menunjukkan kemiringan 38,21 *degree*, sedangkan pengukuran menggunakan busur menunjukkan 35 *degree*.

Kemiringan ROV ketika naik sangat berbeda jauh dengan gerakan sebelumnya dikarenakan beban didepan ROV berupa lengan robot manipulator yang membuat ROV miring ketika bergerak naik hingga 35 *degree* dari pembacaan busur. Pergerakan ROV ketika turun dengan kecepatan 0,22 m/s nilai sensor menunjukkan kemiringan 5,12 *degree*, sedangkan saat pengukuran menggunakan busur menunjukkan kemiringan 5 *degree*.

Pergerakan ROV ketika belok kanan dengan kecepatan 0,18 m/s nilai sensor menunjukkan kemiringan 15,34 *degree*, sedangkan pengukuran menggunakan busur menunjukkan kemiringan 15 *degree*.

Pergerakan ROV ketika belok kiri dengan kecepatan 0,17 m/s nilai sensor menunjukkan kemiringan 17,65 *degree*, sedangkan pengukuran menggunakan busur menunjukkan nilai 15 *degree*. Hasil pengujian sistem keseluruhan dapat dilihat pada tabel 4.3.

Tabel 4. 9 Pengujian Keseluruhan Sistem didalam Air

Gerakkan ROV	Kecepatan (m/s)	Pengukuran Busur (<i>deg</i>)	Pembacaan Sensor (<i>deg</i>)	Error Sensor (%)
Maju	0,45	10	9,51	4,9
Mundur	0,33	10	9,78	2,2
Naik	0,13	35	38,21	9,1
Turun	0,22	5	5,12	4,2
Belok Kanan	0,18	15	15,34	2,3
Belok Kiri	0,17	15	17,65	17,6
Rata-Rata	0,25	15	15,93	6,7

Berdasarkan data tabel 4.3 diatas maka dapat disimpulkan bahwa kecepatan rata-rata gerak ROV didalam air bisa hingga 0,25 m/s dengan nilai *error* rata-rata sensor *gyroscope* untuk mengetahui kemiringan ROV terhadap sumbu y ketika bergerak didalam air adalah 6,7 %. Nilai error yang tinggi ini diakibatkan adanya tambahan beban didepan ROV berupa lengan manipulator yang dibelokkan kearah kanan ROV untuk mengunci pergerakan lengan manipulator ketika dilakukan pengujian.

-----Halaman ini sengaja dikosongkan-----

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan pengujian dan analisis data, maka dapat ditarik kesimpulan:

1. Dengan rangka ROV yang memiliki volume 5,51669 L berat beban rangka ROV agar bisa terapung adalah 2 Kg, untuk melayang maka memerlukan berat 4-5,5 Kg dan maksimal yang dapat dibawa rangka ROV agar bisa tenggelam adalah 7,6 Kg.
2. Kontrol yang dibuat menggunakan *joystick* dengan perintah yang telah ditentukan sehingga didapatkan gerakan yang diinginkan. Tombol perintah LY < 132 untuk gerakan ROV maju, LY > 132 untuk gerakan ROV mundur, RX < 123 gerakan ROV belok kiri, RX > 123 gerakan ROV belok kanan, RY < 123 dan RX > 123 gerakan ROV maju, dan RY > 123 serta RX < 123 gerakan belok kiri.
3. Kecepatan rata-rata gerak ROV ketika didalam air ketika gerakan maju adalah 0,19 m/s, gerakan mundur 0,15 m/s, gerakan naik 0,13 m/s, gerakan turun 0,16 m/s, gerakan belok kanan 0,14 m/s, dan gerakan belok kiri 0,14 m/s. Sehingga didapatkan kecepatan rata-rata gerak ROV didalam air adalah 0,15 m/s
4. Penggunaan MPU 6050 untuk sensor keseimbangan ROV memiliki *error* rata-rata pembacaan sudut maksimal sebesar 0,52% untuk sudut *pitch* didalam air. Sedangkan nilai *error* rata-rata kemiringan ROV didalam air sebesar 6,7%.

5.2 Saran

Beberapa saran yang penulis dapat berikan untuk pengembangan tugas akhir :

1. Pemodelan dan pemahaman tentang desain robot akan membantu perancangan sistem yang lebih baik dan mendapatkan model matematika yang lebih akurat.
2. Mengganti kabel yang banyak untuk komunikasi antara mikrokontroler dengan *driver* motor dengan RS 485.
3. Membuat sistem otomatis untuk melakukan penyeimbangan dengan menggunakan data dari sensor MPU 6050.
4. Membuat kecepatan gerak ROV tetap sama untuk setiap arah gerakan ROV didalam air.

-----Halaman ini sengaja dikosongkan-----

DAFTAR PUSTAKA

- [1] Adriansyah, A. "Perancangan Pergerakan Robot Bawah Air". Seminar Nasional Informatika 2008, UPN "Veteran" Yogyakarta, 24 Mei 2008.
- [2] Bachmayer R, S. Humphiris dkk. 1999. "A new remotely Operated Underwater Dynamics for Wynamics and Control". *International Symposium Unmanned Untethered Submersible Technology*. 1(1):1-25
- [3] Bobby, Grace, dkk. 2015. "Implementasi Robot Keseimbangan Beroda Dua Berbasis Mikrokontroler". Jurnal ELKOMIKA.
- [4] F. Irawan, A. Yulianto. "Perancangan Prototype Robot Observasi Bawah Air Dan Kontrol Hovering Menggunakan Metode PID Control". Jurnal Sains dan Informatika, Volume 1, Nomor 1, Juni 2015.
- [5] G, Kuiper. 1992. "The Wageningen Propeller Series". Hamburg Press, Hamburg.
- [6] Ihsanuddin. 2017. "Terumbu Karang Rusak di Raja Ampat Butuh Waktu 20 Tahun untuk Pulih". Kompas.com. dilihat 15 Oktober 2017.
(<http://nasional.kompas.com/read/2017/03/15/19465491/terumbu.karang.rusak.di.raja.ampat.butuh.waktu.20.tahun.untuk.pulih>).
- [7] Jemadu. 2016. "Di 2015 Jumlah Serangan Hiu Cetak Rekor Terbanyak dalam Sejarah". Suara.com. dilihat 16 oktober 2017.
(<https://www.suara.com/tekno/2016/02/10/200908/di-2015-jumlah-serangan-hiu-cetak-rekor-terbanyak-dalam-sejarah>).
- [8] Robinson, H and Keary A. 2000. "Remote Control of unmanned undersea vehicle". *International Unmanned Undersea Vehicle Symposium* 3(9):245-246.
- [9] Rutherford, K. 2010. "Evolution of an AUV Design Strategy". University of Southampton Press. Southsmpton.
- [10] Samosir, H Tirza, dkk. 2012. "Aplikasi remotely operated vehicle (ROV) dalam penelitian kelautan dan perikanan di sekitar perairan Sulawesi Utara dan Biak Papua". Jurnal Ilmu dan Teknologi Perikanan Tangkap. 1(1):22-25.
- [11] S. C. P. Gomes, C. E. M. Moraes, P. L. J. Drews-Jr, T. G. Moreira, and A. M. Tavares. 2005. "Underwater vehicle dynamic modeling," in 18th Int. Cong. of Mechanical Engineering - COBEM, 2005, 5, Ouro Preto, MG.

- [12] Yulianto, Andik dan Hadi Puspa Handoyo. 2015. “Pengembangan Robot Jelajah Bawah Air untuk Observasi Terumbu Karang”. Jurnal Ilmiah Mikrotek Vol 2, No.1. (1):6-8.

LAMPIRAN

A-32 Listing Code Membaca Sudut Sensor MPU 6050

```
#include <Wire.h>
int gyro_x, gyro_y, gyro_z;
long gyro_x_cal, gyro_y_cal, gyro_z_cal;
boolean set_gyro_angles;
long acc_x, acc_y, acc_z, acc_total_vector;
float angle_roll_acc, angle_pitch_acc;
float angle_pitch, angle_roll;
int angle_pitch_buffer, angle_roll_buffer;
float angle_pitch_output, angle_roll_output;
long loop_timer;
int temp;
void setup() {
  Wire.begin();
  setup_mpu_6050_registers();
  for (int cal_int = 0; cal_int < 1000 ; cal_int++){
    read_mpu_6050_data();
    gyro_x_cal += gyro_x;
    gyro_y_cal += gyro_y;
    gyro_z_cal += gyro_z;
    delay(3);
  }
  gyro_x_cal /= 1000;
  gyro_y_cal /= 1000;
  gyro_z_cal /= 1000;
  Serial.begin(115200);
  loop_timer = micros();
}
void loop(){
  read_mpu_6050_data();
  gyro_x -= gyro_x_cal;
  gyro_y -= gyro_y_cal;
  gyro_z -= gyro_z_cal;
  angle_pitch += gyro_x * 0.0000611;
  angle_roll += gyro_y * 0.0000611;
  angle_pitch += angle_roll * sin(gyro_z * 0.000001066);
```

```

    angle_roll -= angle_pitch * sin(gyro_z * 0.000001066);
    acc_total_vector
= sqrt((acc_x*acc_x)+(acc_y*acc_y)+(acc_z*acc_z));
    angle_pitch_acc = asin((float)acc_y/acc_total_vector)* 57.296;
    angle_roll_acc = asin((float)acc_x/acc_total_vector)* -57.296;
    angle_pitch_acc -= 0.0;
    angle_roll_acc -= 0.0;
    if(set_gyro_angles){
        angle_pitch = angle_pitch * 0.9996 + angle_pitch_acc * 0.0004;
        angle_roll = angle_roll * 0.9996 + angle_roll_acc * 0.0004;
    }
    else{
        angle_roll = angle_roll_acc;
        set_gyro_angles = true;
    }
    angle_pitch_output = angle_pitch_output * 0.9 + angle_pitch * 0.1;
    angle_roll_output = angle_roll_output * 0.9 + angle_roll * 0.1;
    Serial.print(" | Angle = "); Serial.println(angle_pitch_output);
    while(micros() - loop_timer < 4000);
    loop_timer = micros();
}

void setup_mpu_6050_registers(){
    Wire.beginTransmission(0x68);
    Wire.write(0x6B);
    Wire.write(0x00);
    Wire.endTransmission();
    Wire.beginTransmission(0x68);
    Wire.write(0x1C);
    Wire.write(0x10);
    Wire.endTransmission();
    Wire.beginTransmission(0x68);
    Wire.write(0x1B);
    Wire.write(0x08);
    Wire.endTransmission();
}

void read_mpu_6050_data(){
    Wire.beginTransmission(0x68);
    Wire.write(0x3B);
    Wire.endTransmission();
}

```

```
Wire.requestFrom(0x68,14);
while(Wire.available() < 14);
acc_x = Wire.read()<<8/Wire.read();
acc_y = Wire.read()<<8/Wire.read();
acc_z = Wire.read()<<8/Wire.read();
temp = Wire.read()<<8/Wire.read();
gyro_x = Wire.read()<<8/Wire.read();
gyro_y = Wire.read()<<8/Wire.read();
gyro_z = Wire.read()<<8/Wire.read();
}
```

-----Halaman ini sengaja dikosongkan-----

B-33 Listing Code Gerakkan ROV dengan Joystick

```
#include <PS2X_lib.h> //for v1.6
PS2X ps2x; // create PS2 Controller Class
int error = 0;
byte type = 0;
byte vibrate = 0;
int RPWM = 3;
int LPWM = 4;
int RPWM1 = 2;
int LPWM1 = 5;
int RPWM2 = 9;
int LPWM2 = 8;
int L_EN = 7;
int R_EN = 6;
void setup() {
  Serial.begin(57600);
  pinMode(RPWM, OUTPUT);
  pinMode(LPWM, OUTPUT);
  pinMode(R_EN, OUTPUT);
  pinMode(L_EN, OUTPUT);
  digitalWrite(R_EN, HIGH);
  digitalWrite(L_EN, HIGH);
  error = ps2x.config_gamepad(13, 11, 10, 12, true, true);
  if (error == 0) {
    Serial.println("Found Controller, configured successful");
    Serial.println("Try out all the buttons, X will vibrate the controller,
faster as you press harder;");
    Serial.println("holding L1 or R1 will print out the analog stick
values.");
    Serial.println("Go to www.billporter.info for updates and to report
bugs.");
  }
  else if (error == 1)
    Serial.println("No controller found, check wiring, see readme.txt to
enable debug. visit www.billporter.info for troubleshooting tips");
  else if (error == 2)
```

```

    Serial.println("Controller found but not accepting commands. see
readme.txt to enable debug. Visit www.billporter.info for troubleshooting
tips");
    else if (error == 3)
        Serial.println("Controller refusing to enter Pressures mode, may not
support it. ");
    type = ps2x.readType();
    switch (type) {
        case 0:
            Serial.println("Unknown Controller type");
            break;
        case 1:
            Serial.println("DualShock Controller Found");
            break;
        case 2:
            Serial.println("GuitarHero Controller Found");
            break;
    }
}
}
void loop() {
    if (error == 1)
        return;
    if (type == 2) {
        ps2x.read_gamepad();
        if (ps2x.ButtonPressed(GREEN_FRET))
            Serial.println("Green Fret Pressed");
        if (ps2x.ButtonPressed(RED_FRET))
            Serial.println("Red Fret Pressed");
        if (ps2x.ButtonPressed(YELLOW_FRET))
            Serial.println("Yellow Fret Pressed");
        if (ps2x.ButtonPressed(BLUE_FRET))
            Serial.println("Blue Fret Pressed");
        if (ps2x.ButtonPressed(ORANGE_FRET))
            Serial.println("Orange Fret Pressed");
        if (ps2x.ButtonPressed(STAR_POWER))
            Serial.println("Star Power Command");
        if (ps2x.Button(UP_STRUM))
            Serial.println("Up Strum");
        if (ps2x.Button(DOWN_STRUM))

```

```

    Serial.println("DOWN Strum");
    if (ps2x.Button(PSB_START))
        Serial.println("Start is being held");
    if (ps2x.Button(PSB_SELECT))
        Serial.println("Select is being held");
    if (ps2x.Button(ORANGE_FRET))
    {
        Serial.print("Wammy Bar Position:");
        Serial.println(ps2x.Analog(WHAMMY_BAR), DEC);
    }
}
else {
    ps2x.read_gamepad(false, vibrate);
    if (ps2x.Button(PSB_START))
        Serial.println("Start is being held");
    if (ps2x.Button(PSB_SELECT))
        Serial.println("Select is being held");
    if (ps2x.Button(PSB_PAD_UP)) {
        Serial.print("Up held this hard: ");
        Serial.println(ps2x.Analog(PSAB_PAD_UP), DEC);
    }
    if (ps2x.Button(PSB_PAD_RIGHT)) {
        Serial.print("Right held this hard: ");
        Serial.println(ps2x.Analog(PSAB_PAD_RIGHT), DEC);
    }
    if (ps2x.Button(PSB_PAD_LEFT)) {
        Serial.print("LEFT held this hard: ");
        Serial.println(ps2x.Analog(PSAB_PAD_LEFT), DEC);
    }
    if (ps2x.Button(PSB_PAD_DOWN)) {
        Serial.print("DOWN held this hard: ");
        Serial.println(ps2x.Analog(PSAB_PAD_DOWN), DEC);
    }
    vibrate = ps2x.Analog(PSAB_BLUE);
    //how hard you press the blue (X) button
    if (ps2x.NewButtonState())
    {
        if (ps2x.Button(PSB_L3))
            Serial.println("L3 pressed");
    }
}

```

```

if (ps2x.Button(PSB_R3))
  Serial.println("R3 pressed");
if (ps2x.Button(PSB_L2))
  Serial.println("L2 pressed");
if (ps2x.Button(PSB_R2))
  Serial.println("R2 pressed");
if (ps2x.Button(PSB_GREEN))
  Serial.println("Triangle pressed");
}
if (ps2x.ButtonPressed(PSB_RED))
  Serial.println("Circle just pressed");
if (ps2x.ButtonReleased(PSB_PINK))
  Serial.println("Square just released");
if (ps2x.NewButtonState(PSB_BLUE))
  Serial.println("X just changed");
if (ps2x.Button(PSB_L1) || ps2x.Button(PSB_R1))
{
  Serial.print("Stick Values:");
  Serial.print(ps2x.Analog(PSS_LY), DEC);
  Serial.print(",");
  Serial.print(ps2x.Analog(PSS_LX), DEC);
  Serial.print(",");
  Serial.print(ps2x.Analog(PSS_RY), DEC);
  Serial.print(",");
  Serial.println(ps2x.Analog(PSS_RX), DEC);
}
//TURUN
if (ps2x.Analog(PSS_LY) > 132)
{
  int satu = ps2x.Analog(PSS_LY);
  satu = map(satu, 133, 255, 0, 255);
  analogWrite(RPWM, LOW);
  digitalWrite(LPWM, satu);
  delay(15);
}
//NAIK
if (ps2x.Analog(PSS_LY) < 132)
{
  int dua = ps2x.Analog(PSS_LY);

```

```

    dua = map(dua, 0, 131, 255, 0);
    digitalWrite(RPWM, dua);
    analogWrite(LPWM, LOW);
    delay(15);
}
//DIAM NAIK TURUN
if (ps2x.Analog(PSS_LY) == 132)
{
    digitalWrite(RPWM, LOW);
    digitalWrite(LPWM, LOW);
}
//MUNDUR
if (ps2x.Analog(PSS_RY) > 123)
if (ps2x.Analog(PSS_RX) < 123)
{
    int tiga = ps2x.Analog(PSS_RY);
    tiga = map(tiga, 124, 255, 0, 255);
    analogWrite(RPWM1, tiga);
    digitalWrite(LPWM1, LOW);
    analogWrite(RPWM2, tiga);
    digitalWrite(LPWM2, LOW);
    delay(15);
}
//MAJU
if (ps2x.Analog(PSS_RY) < 123)
if (ps2x.Analog(PSS_RX) > 123)
{
    int empat = ps2x.Analog(PSS_RY);
    empat = map(empat, 0, 122, 255, 0);
    digitalWrite(RPWM1, LOW);
    analogWrite(LPWM1, empat);
    digitalWrite(RPWM2, LOW);
    analogWrite(LPWM2, empat);
    delay(15);
}
//DIAM MAJU MUNDUR
if (ps2x.Analog(PSS_RY) == 123)
{
    digitalWrite(RPWM1, LOW);

```

```

    digitalWrite(LPWM1, LOW);
    digitalWrite(RPWM2, LOW);
    digitalWrite(LPWM2, LOW);
}
//KANAN
if (ps2x.Analog(PSS_RX) > 123)
{
    int lima = ps2x.Analog(PSS_RX);
    lima = map(lima, 124, 255, 0, 255);
    analogWrite(LPWM1, lima);
    digitalWrite(RPWM1, LOW);
    delay(15);
}
//KIRI
if (ps2x.Analog(PSS_RX) < 123)
{
    int enam = ps2x.Analog(PSS_RX);
    enam = map(enam, 0, 122, 255, 0);
    digitalWrite(RPWM2, LOW);
    analogWrite(LPWM2, enam);
    delay(15);
}
//DIAM KANAN KIRI
if (ps2x.Analog(PSS_RX) == 123)
{
    digitalWrite(RPWM1, LOW);
    digitalWrite(LPWM1, LOW);
    digitalWrite(RPWM2, LOW);
    digitalWrite(LPWM2, LOW);
}
}
delay(50);
}

```

C-33 Listing Code Membaca Perintah Joystick

```
#include <PS2X_lib.h> //for v1.6
PS2X ps2x; // create PS2 Controller Class
int error = 0;
byte type = 0;
byte vibrate = 0;
void setup(){
  Serial.begin(57600);
  error = ps2x.config_gamepad(13,11,10,12, true, true);
  if(error == 0){
    Serial.println("Found Controller, configured successful");
    Serial.println("Try out all the buttons, X will vibrate the controller,
faster as you press harder;");
    Serial.println("holding L1 or R1 will print out the analog stick values.");
    Serial.println("Go to www.billporter.info for updates and to report
bugs.");
  }
  else if(error == 1)
    Serial.println("No controller found, check wiring, see readme.txt to
enable debug. visit www.billporter.info for troubleshooting tips");
  else if(error == 2)
    Serial.println("Controller found but not accepting commands. see
readme.txt to enable debug. Visit www.billporter.info for troubleshooting
tips");
  else if(error == 3)
    Serial.println("Controller refusing to enter Pressures mode, may not
support it. ");
  type = ps2x.readType();
  switch(type) {
    case 0:
      Serial.println("Unknown Controller type");
      break;
    case 1:
      Serial.println("DualShock Controller Found");
      break;
    case 2:
      Serial.println("GuitarHero Controller Found");
      break;
```

```

    }
}
void loop(){
  if(error == 1)
  if(type == 2){
    ps2x.read_gamepad();
    if(ps2x.ButtonPressed(GREEN_FRET))
      Serial.println("Green Fret Pressed");
    if(ps2x.ButtonPressed(RED_FRET))
      Serial.println("Red Fret Pressed");
    if(ps2x.ButtonPressed(YELLOW_FRET))
      Serial.println("Yellow Fret Pressed");
    if(ps2x.ButtonPressed(BLUE_FRET))
      Serial.println("Blue Fret Pressed");
    if(ps2x.ButtonPressed(ORANGE_FRET))
      Serial.println("Orange Fret Pressed");
    if(ps2x.ButtonPressed(STAR_POWER))
      Serial.println("Star Power Command");
    if(ps2x.Button(UP_STRUM))
      Serial.println("Up Strum");
    if(ps2x.Button(DOWN_STRUM))
      Serial.println("DOWN Strum");
    if(ps2x.Button(PSB_START))
      Serial.println("Start is being held");
    if(ps2x.Button(PSB_SELECT))
      Serial.println("Select is being held");
    if(ps2x.Button(ORANGE_FRET))
    {
      Serial.print("Wammy Bar Position:");
      Serial.println(ps2x.Analog(WHAMMY_BAR), DEC);
    }
  }
}
else {
  ps2x.read_gamepad(false, vibrate);
  if(ps2x.Button(PSB_START))
    Serial.println("Start is being held");
  if(ps2x.Button(PSB_SELECT))
    Serial.println("Select is being held");
  if(ps2x.Button(PSB_PAD_UP)) {

```



```

Serial.print("Up held this hard: ");
Serial.println(ps2x.Analog(PSAB_PAD_UP), DEC);
}
if(ps2x.Button(PSB_PAD_RIGHT)){
Serial.print("Right held this hard: ");
Serial.println(ps2x.Analog(PSAB_PAD_RIGHT), DEC);
}
if(ps2x.Button(PSB_PAD_LEFT)){
Serial.print("LEFT held this hard: ");
Serial.println(ps2x.Analog(PSAB_PAD_LEFT), DEC);
}
if(ps2x.Button(PSB_PAD_DOWN)){
Serial.print("DOWN held this hard: ");
Serial.println(ps2x.Analog(PSAB_PAD_DOWN), DEC);
}
vibrate = ps2x.Analog(PSAB_BLUE);
if (ps2x.NewButtonState())
{
if(ps2x.Button(PSB_L3))
Serial.println("L3 pressed");
if(ps2x.Button(PSB_R3))
Serial.println("R3 pressed");
if(ps2x.Button(PSB_L2))
Serial.println("L2 pressed");
if(ps2x.Button(PSB_R2))
Serial.println("R2 pressed");
if(ps2x.Button(PSB_GREEN))
Serial.println("Triangle pressed");
}
if(ps2x.ButtonPressed(PSB_RED))
Serial.println("Circle just pressed");
if(ps2x.ButtonReleased(PSB_PINK))
Serial.println("Square just released");
if(ps2x.NewButtonState(PSB_BLUE))
Serial.println("X just changed");
if(ps2x.Button(PSB_L1) || ps2x.Button(PSB_R1))
{
Serial.print("Stick Values:");
Serial.print(ps2x.Analog(PSS_LY), DEC);
}

```

```
Serial.print(",");  
Serial.print(ps2x.Analog(PSS_LX), DEC);  
Serial.print(",");  
Serial.print(ps2x.Analog(PSS_RY), DEC);  
Serial.print(",");  
Serial.println(ps2x.Analog(PSS_RX), DEC);  
}  
}  
delay(50);  
}
```

RIWAYAT HIDUP PENULIS



Nama : Gerry Hendria Negara
TTL : Sidoarjo, 07 Juni 1996
Jenis Kelamin : Laki-Laki
Agama : Islam
Alamat Rumah : Glonggongan Baru
RT.27A/RW.007,
Ds.Sumbertebu,
Kec.Bangsals,
Kab.Mojokerto
Nomor HP : 08565554366
E-mail : gerrynegara@gmail.com

RIWAYAT PENDIDIKAN

- 2003 - 2009 : SD Negeri 1 Mojosari
- 2009 - 2012 : SMP Negeri 1 Bangsal
- 2012 - 2015 : SMA Negeri 1 Sooko
- 2015 - sekarang : Bidang Studi Komputer Kontrol,
Departemen Elektro Otomasi, ITS

PENGALAMAN KERJA

- Kerja Praktek di Indonesia Power Bali (2017)