



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**PROYEK AKHIR - VE180626**

**PERANCANGAN SISTEM OBSERVASI, NAVIGASI DAN  
POSISI *REMOTELY OPERATED VEHICLE* DENGAN  
KAMERA, *GPS* DAN KOMPAS**

Dariel Thaha  
NRP. 1031160000033

Dosen Pembimbing  
Ir. Arif Musthofa, M.T.  
Ciptian Wieried Priananda, S.ST, M.T.

DEPARTEMEN TEKNIK ELEKTRO OTOMASI  
Fakultas Vokasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2019

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



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**FINAL PROJECT - VE180626**

**OBSERVATION, NAVIGATION AND POSITIONING  
SYSTEM FOR REMOTELY OPERATED VEHICLE  
WITH CAMERA, GPS AND COMPASS**

Dariel Thaha  
NRP. 1031160000033

*Supervisor*  
Ir. Arif Musthofa, M.T.  
Ciptian Wieried Priananda, S.ST, M.T.

**AUTOMATION ELECTRICAL ENGINEERING DEPARTMENT**  
*Vocation Faculty*  
Institut Teknologi Sepuluh Nopember  
Surabaya 2019

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

## PERNYATAAN KEASLIAN PROYEK AKHIR

Dengan ini penulis menyatakan bahwa isi sebagian maupun keseluruhan buku Proyek Akhir dengan judul **"Perancangan Sistem Observasi, Navigasi dan Posisi Dengan Kamera, GPS dan Kompas"** adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang penulis akui sebagai karya sendiri.

Seluruh data hasil pengujian yang ditulis benar-benar asli tanpa penambahan atau pengurangan dan mampu dipertanggungjawabkan. Semua referensi yang dikutip atau dirujuk telah ditulis secara lengkap pada daftar pustaka dan telah terbukti validitasnya.

Apabila ternyata pernyataan ini tidak benar, penulis bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 28 Desember 2018



Darief Thaha  
1031160000033

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

**PERANCANGAN SISTEM OBSERVASI, NAVIGASI  
DAN POSISI *REMOTELY OPERATED VEHICLE*  
DENGAN KAMERA, GPS DAN KOMPAS**

**LEMBAR PENGESAHAN  
PROYEK AKHIR**

Diajukan Guna Memenuhi Sebagian Persyaratan  
Untuk Memperoleh Gelar Ahli Madya  
Pada  
Departemen Teknik Elektro Otomasi  
Fakultas Vokasi  
Institut Teknologi Sepuluh Nopember

Menyetujui:

Dosen Pembimbing 1

Dosen Pembimbing 2

Ir. Arif Musthofa, M.T.  
NIP.196608111992031004

Ciptian Wened Priananda,  
S.ST. M.T.  
NIP.1990201711060



**SURABAYA  
JANUARI, 2020**

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



# PERANCANGAN SISTEM OBSERVASI, NAVIGASI DAN POSISI *REMOTELY OPERATED VEHICLE* DENGAN KAMERA, *GPS* DAN KOMPAS

Nama : Dariel Thaha  
Pembimbing : Ir. Arif Musthofa, M.T.  
Ciptian Weried Prriananda, S.ST., M.T.

## ABSTRAK

*Remotely Operated Vehicle (ROV)* pada dasarnya adalah robot bawah air yang bisa dioperasikan oleh seseorang di permukaan. Sistem ROV terdiri dari kendaraan, yang terhubung ke kontrol dan operator di permukaan dengan kabel yang membawa sinyal listrik, video dan data bolak-balik antara operator dan kendaraan. *ROV* utamanya digunakan sebagai alat observasi bawah. Robot ini diharapkan untuk bisa observasi pada lingkungan dibawah air, agar bisa meneliti keadaan keadaan yang terjadi.

*ROV* dilengkapi dengan kamera video, *GPS* dan kompas. Untuk proyek akhir ini dibuat untuk melanjutkan kekurangan alat *ROV* sebelumnya untuk membantu robot memiliki sistem untuk observasi. *ROV* telah dipasangkan dengan webcam sebagai sistem observasi untuk lingkungan bawah air dengan program pendukung untuk bisa mendapatkan dan menyimpan foto yang diambil oleh pengguna. *GPS* yang digunakan untuk *ROV* adalah *GPS Neo-6m* untuk mengetahui posisi robot saat berada di permukaan air atau dalam air dangkal dimana air sebanyak itu tidak dirupsu sinyal. Seandainya robot memasuki kedalaman air terlalu rendah maka sinyal *GPS* pada satelit akan menghilang. Kompas yang digunakan adalah modul kompas GY-273 hmc5883l, untuk mengetahui arah mata angin yang dihadap oleh kompas digunakan magnetometer tetapi perlu kalibrasi *offset* untuk mengetahui arah derajat akurat.

Dari perancangan dan instalasi ketiga sistem pada robot, didapatkan bahwa sistem-sistem berkerja, untuk kamera gambar yang diambil konsisten kualitas foto sebuah target terhadap jarak dalam 120cm tetapi bisa bantu lebih lanjut dengan adanya pencahayaan untuk masalah kegelapan foto, untuk kompas dan *GPS* memiliki rata-rata nilai eror 1.28% dan jarak eror dengan rata-rata 1.393 meter.

**Kata Kunci : *ROV*, Arduino Uno, *GPS*, Webcam, Kompas.**

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

# DESIGN OF OBSERVATION, NAVIGATION AND REMOTELY OPERATED VEHICLE POSITION SYSTEMS WITH CAMERA, GPS AND COMPASS

**Name** : Dariel Thaha  
**Supervisors** : Ir. Arif Musthofa, M.T.  
Ciptian Weried Prriananda, S.ST., M.T.

## **ABSTRACT**

*Remotely Operated Vehicle (ROV) is basically an underwater robot that can be operated by someone on the surface. The ROV system consists of the vehicle, which is connected to the control and operator on the surface with a cable that carries electrical signals, video and data back and forth between the operator and the vehicle. The ROV is mainly used as an observation tool below. This robot is expected to be able to observe underwater environments, so that it can examine the circumstances that occur.*

*ROV is equipped with a video camera, GPS and compass. For this final project is made to overcome the shortcomings of the previous ROV to help robots have a system for observation. ROV has been paired with a webcam as an observation system for underwater environments with a support program to be able to get and save photos taken by users. The GPS used for ROV is the Neo-6m GPS to determine the robot's position when it is on the surface of the air or in shallow air where that much air does not disturb the signal. If the robot moves the altitude too low then the GPS signal on the satellite will disappear. The compass used is the GY-273 hmc5883l compass module, to determine the direction of the compass facing the compass used by a magnetometer, but it needs calibration of the offset to find an accurate direction.*

*From the design and installation of the three systems on the robot, it was found that the systems work, for camera images taken consistent with the quality of a target's photo within distance of 120cm but could be further helped by the existence of lighting for the photo darkness problem, for the compass and GPS have an average error value of 1.28% and error distance with an average of 1,393 meters.*

**Keywords:** ROV, Arduino Uno, GPS, Webcam, Compass.

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

## KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT yang selalu memberikan rahmat dan hidayah-Nya sehingga Proyek 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. Proyek Akhir ini disusun bertujuan untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan Diploma 3 Departemen Teknik Elektro Otomasi, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember Surabaya yang berjudul **“Perancangan Sistem Observasi, Navigasi dan Posisi Dengan Kamera, GPS dan Kompas”**. Penulis mengucapkan terima kasih kepada Staff/Karyawan/Dosen Departemen Teknik Elektro Otomasi, Orang tua yang saya sayangi serta kakak dan adik juga sahabat dekat yang telah memberikan dukungan moral dan spiritual. Penulis juga mengucapkan banyak terima kasih kepada semua pihak yang telah membantu baik secara langsung maupun tidak langsung dalam proses penyelesaian Proyek Akhir ini.

Penulis menyadari sepenuhnya bahwa Proyek Akhir ini jauh dari kata sempurna, oleh karena itu penulis mengharapakan saran dan kritik yang bersifat membangun untuk memperbaiki kekurangan dalam Proyek Akhir ini. Akhir kata semoga Proyek Akhir ini dapat bermanfaat dalam pengembangan keilmuan di kemudian hari.

Surabaya, 10 December 2019

Dariel Thaha

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

## DAFTAR ISI

HALAMAN JUDUL .....	i
HALAMAN JUDUL .....	iii
PERNYATAAN KEASLIAN .....	v
LEMBAR PENGESAHAN .....	xiii
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 .....	2
1.4 Batasan Masalah .....	2
1.5 Metodologi .....	3
1.6 Sistematika Laporan .....	4
BAB II TEORI DASAR .....	5
2.1 ROV ( <i>Remotely Operated Vehicle</i> ) .....	5
2.2 Arduino Uno (CH340 & R3 Clone) .....	6
2.3 Webcam .....	8
2.4 GPS NEO-6M .....	10
2.4.1 <i>Raw Data</i> NMEA Dari NEO-6M .....	11
2.5 GY-273 <i>Digital Compass</i> .....	12
2.6 Visual Studios 2017 .....	13
2.7 Arduino IDE .....	14
BAB III PERANCANGAN SISTEM .....	14
3.1 Perancangan Sistem .....	17
3.1.1 Studi Literatur .....	17
3.1.2 Perancangan Elektronik .....	17
3.1.3 Perancangan Software .....	17
3.1.4 Pengujian Robot ROV .....	17
3.1.5 Pembuatan Laporan .....	17
3.2 Perancangan Perangkat Elektronik .....	18
3.2.1 <i>Setting Port</i> Arduino Uno .....	19
3.2.2 Perancangan Webcam .....	20
3.2.3 Perancangan Sistem Navigasi .....	20

3.2.4	Wiring GY-273 Pada Arduino Uno .....	21
3.2.5	Perancangan Sistem Posisi .....	21
3.2.6	Wiring GPS Neo-6M Pada Arduino .....	22
3.3	Perancangan Perangkat Lunak ( <i>Software</i> ) .....	23
3.3.1	Kalibrasi Kompas <i>Digital</i> .....	23
3.3.2	Program Kompas <i>Digital</i> .....	23
3.3.3	Perancangan Program Sistem Navigasi.....	25
3.3.4	Program GPS Neo-6m .....	27
3.3.5	Perancangan Program Pembacaan Sensor GPS .....	28
3.4	Perancangan <i>Interface</i> .....	29
3.5	Perancangan Komponen Sistem-Sistem Pada ROV .....	31
BAB IV PENGUJIAN DAN ANALISA .....		35
4.1	Pengujian Kebocoran Tabung .....	35
4.2	Pengujian Sistem Navigasi Kompas <i>Digital</i> GY-273 .....	37
4.3	Pengujian Sistem Posisi GPS .....	43
4.4	Pengujian Sistem Observasi Webcam C270 .....	45
BAB V PENUTUP .....		51
5.1	Kesimpulan .....	51
5.2	Saran.....	52
DAFTAR PUSTAKA.....		53



## DAFTAR GAMBAR

Gambar 2.1	ROV ( <i>Remotely Operated Vehicle</i> ) .....	5
Gambar 2.2	Arduino Uno .....	7
Gambar 2.3	Webcam Logitech C270 .....	9
Gambar 2.4	Sensor GPS Ublox NEO 6M.....	10
Gambar 2.5	GY-273 .....	12
Gambar 2.6	Visual Studio 2017 .....	13
Gambar 2.7	Tampilan Arduino IDE .....	14
Gambar 3.1	Diagram Alir Perencanaan.....	16
Gambar 3.2	Diagram Fungsional Sistem Keseluruhan .....	18
Gambar 3.3	Diagram Fungsional Sistem Observasi .....	20
Gambar 3.4	Penempatan Kamera Pada Tabung ROV .....	20
Gambar 3.5	Diagram Fungsional Sistem Navigasi.....	20
Gambar 3.6	Koneksi Kompas Digital GY-273 Pada Arduino Uno ...	21
Gambar 3.7	Diagram Fungsional Sistem Posisi .....	21
Gambar 3.8	Wiring Neo-6m Pada Arduino Uno .....	22
Gambar 3.9	Program Kalibrasi Untuk Kompas <i>Digital</i> .....	23
Gambar 3.10	Program Kompas <i>Digital</i> .....	24
Gambar 3.11	Flowchart Kompas <i>Digital</i> .....	26
Gambar 3.12	Program GPS Untuk ROV .....	27
Gambar 3.13	Flowchart Sensor GPS .....	28
Gambar 3.14	Tampilan <i>Interface</i> Saat <i>Offline</i> .....	29
Gambar 3.15	Tampilan <i>Interface</i> Saat Ketiganya Nyala .....	30
Gambar 3.16	Tabung Elektronik ROV .....	31
Gambar 3.17	Rancangan Desain Tempat Elektronik Robot.....	32
Gambar 3.18	Posisi Yang Akan Ditempati Arduino Uno .....	33
Gambar 3.19	Penempatan Arduino Dengan Sensor GPS .....	33
Gambar 3.20	Sisi Bawanya Tabung ROV .....	34
Gambar 4.1	Letak GPS Pada Tabung .....	35
Gambar 4.2	Letak Webcam & Kompas Pada Tabung .....	36
Gambar 4.3	Pengujian Kebocoran Penyiraman Air Pada Tabung ROV .....	36
Gambar 4.4	Pengujian Kebocoran Tabung Dalam Air .....	37
Gambar 4.5	Letak Fisik Sensor Kompas Pada ROV .....	37
Gambar 4.6	Salah Satu Pengujian Sensor GY-273 Dibandingkan Dengan Handphone .....	34
Gambar 4.7	Tampilan Sensor Kompas Pada <i>Visual Basic</i> .....	42
Gambar 4.8	Tampilan Output GPS Pada <i>Visual Basic</i> .....	43

Gambar 4.9	Lokasi GPS <i>Handphone</i> .....	43
Gambar 4.10	Tampilan Kamera Webcam Di <i>Visual Basic</i> .....	45
Gambar 4.11	Hasil Foto <i>Save</i> Dari Program <i>Visual Basic</i> .....	45
Gambar 4.12	Perbandingan Kamera Jarak 60cm.....	46
Gambar 4.13	Pebandingan Kamera Jarak 90cm.....	46
Gambar 4.14	Pebandingan Kamera Jarak 1200cm.....	47
Gambar 4.15	Hasil Kedua Kamera Foto Air Kotor .....	48
Gambar 4.16	Foto Kamera Saat Plastik Didepannya Mengembun.....	49
Gambar 4.17	Mengembunnya Akrilik Pada Tabung ROV .....	49

## DAFTAR TABEL

Tabel 2.1	<i>Spesifikasi Arduino Uno</i> .....	7
Tabel 2.2	Jenis-Jenis Kalimat NMEA dan Deskripsi .....	11
Tabel 2.3	Konfigurasi Pin GY-273 .....	12
Tabel 3.1	Deskripsi Pin Arduino Uno .....	19
Tabel 4.1	Perbandingan Data Hasil Pengujian Kompas dari Sensor Dengan Kompas Handphone.....	38
Tabel 4.2	Perbandingan Kompas Sensor Dengan Kompas Handphone Arah Mata Angin Tenggara .....	39
Tabel 4.3	Perbandingan Kompas Sensor Dengan Kompas Handphone Arah Mata Angin Timur Laut .....	40
Tabel 4.4	Perbandingan Kompas Sensor Dengan Kompas Handphone Arah Mata Angin Selatan.....	40
Tabel 4.5	Perbandingan Kompas Sensor Dengan Kompas Handphone Arah Mata Angin Barat .....	40
Tabel 4.6	Perbandingan Kompas Sensor Dengan Kompas Handphone Arah Mata Angin Utara .....	41
Tabel 4.7	Perbandingan Kompas Sensor Dengan Kompas Handphone Arah Mata Angin Barat Daya.....	41
Tabel 4.8	Perbandingan Kompas Sensor Dengan Kompas Handphone Arah Mata Angin Timur .....	41
Tabel 4.9	Perbandingan Kompas Sensor Dengan Kompas Handphone Arah Mata Angin Barat Laut.....	42
Tabel 4.10	Perbandingan Data Hasil Pengujian GPS Dari Sensor GPS Dengan GPS <i>Handphone</i> .....	44

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

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

*Remotely Operated Vehicle* atau ROV, memungkinkan untuk menjelajahi laut tanpa benar-benar berada di laut. Robot bawah air ini dikendalikan oleh seseorang yang biasanya berada di kapal permukaan, menggunakan joystick dengan cara yang sama seperti bermain video game. Sekelompok kabel, atau tether, menghubungkan ROV ke kapal, mengirimkan sinyal listrik bolak-balik antara operator dan kendaraan. Kebanyakan ROV dilengkapi dengan setidaknya kamera diam, kamera video, dan lampu, yang berarti bahwa mereka dapat mengirim gambar dan video kembali ke kapal. Peralatan tambahan, seperti manipulator atau cutting arm, sampler air, dan instrumen yang mengukur parameter seperti kejernihan air dan suhu, juga dapat ditambahkan ke kendaraan untuk memungkinkan pengumpulan sampel. Pertama kali dikembangkan untuk keperluan industri, seperti inspeksi internal dan eksternal dari pipa bawah laut dan pengujian struktural platform lepas pantai, ROV sekarang digunakan untuk banyak aplikasi ROV berkisar dalam ukuran dari komputer kecil hingga sebesar truk kecil. ROV yang lebih besar sangat berat dan membutuhkan peralatan lain seperti kerekan untuk meletakkannya di sisi kapal dan ke dalam air.

Untuk memahami lingkungan bawah air, para ilmuwan sering menemukan kendala saat pengamatan langsung atau mempergunakan *Variable* tertentu menuju bagian-bagian yang sangat spesifik. Secara tradisional, para peneliti telah menggunakan kapal untuk melakukan *photography* saat dokumentasi di kedalaman, namun untuk mengapung atau mengikuti arus menjadi bagian yang sulit, termasuk untuk mengumpulkan sampel air, batuan, dan kehidupan bawah air. ROV digunakan untuk monitor kejadian-kejadian yang terjadi dibawah air seperti kecelakaan atau mencari barang. Kegunaan ROV ini akan membawa sebuah kamera untuk monitor gambar saat dibawah air, GPS digunakan untuk membantu posisikan ROV di koordinat yang diinginkan di permukaan air sebelum menyelam.

D3 Departemen Teknik Elektro Otomasi ITS sudah ada yang membuat proyek akhir tentang rancang ROV tetapi masih perlu dikembangkan, dengan melihat saran-saran dari penulis seperti "Menambahkan fitur untuk mengetahui posisi robot dari gerakan

robot”, “Perlu penambahan sensor kompas agar semakin akurat data orientasi” dan “Menambahkan fitur observasi seperti kamera dan sensor guna mengenali lingkungan”. Sebagian besar ROV dilengkapi dengan setidaknya kamera video dan lampu untuk bisa observasi keadaan di bawah air, untuk proyek akhir ini dibuat untuk melanjutkan proyek sebelumnya untuk memperbaiki kekurangannya.

Untuk pemasangan GPS digunakan untuk mengetahui posisi ROV saat berada di permukaan air. Sistem GPS jarang dipakai dengan ROV dikarenakan sinyal selain dari kabel akan hilang saat menyelam ke dalam air, tetapi akan mendapat sinyal ketika sensor berada di atas/permukaan air, untuk proyek ini cukup saja untuk mengetahui posisi ROV untuk menyelam di titik target yang diinginkan.

## **1.2 Tujuan Dan Manfaat**

Dengan mengenal banyak ROV yang sudah menggunakan kamera, tetapi banyak kekurangannya yang dibuat dengan sistem monitor posisi, hal ini jarang dipakai karena pada saat ROV menyelam ke bawah air, sinyal akan menghilang maka jadi susah untuk mengetahui posisi bila tidak ada sensor kompas. Selain itu, tujuan lainnya yaitu untuk mempermudah dalam membantu melakukan observasi lingkungan dengan menggunakan robot ROV, manfaat dari proyek akhir ini yaitu diharapkan dapat bermanfaat sebagai alat yang digunakan untuk observasi lingkungan air pada suatu kawasan lingkungan air (danau atau sungai). Selain itu diharapkan dapat menjadi dasar dari penelitian selanjutnya.

## **1.3 Perumusan Masalah**

Berdasarkan latar belakang yang telah penulis uraikan, rumusan masalah pada proyek ini adalah bagaimana penggunaan Arduino Uno sebagai sistem monitoring lokasi dengan titik koordinat, derajat arah mata angin dan observasi bawah air dengan menggunakan modul GPS, sensor kompas *digital* dan kamera webcam.

Untuk dapat monitoring gerak dan arah heading ROV dibutuhkan sebuah sistem monitoring yang dapat dilihat secara langsung dari atas permukaan air.

#### 1.4 Batasan Masalah

Dalam pembuatan alat pada Proyek Akhir ini, diberikan batasan masalah sebagai berikut:

1. Sensor GPS yang digunakan adalah GPS NEO-6M
2. Sinyal elektronika tidak dapat menembus air maka susah untuk mendapatkan koneksi dari satelit ke sensor GPS
3. Sinyal elektronika dari GPS ke satelit dan sebaliknya butuh waktu lama untuk konek jika GPS ada beberapa hambatan yang sinyal harus menembus seperti gedung, dan juga tergantung kepada cuaca yang bisa melambatkan proses koneksi.
4. Webcam sangat gampang untuk rusak saat kena air
5. Tabung yang diisi dengan elektronika bergantung pada lem G dan lem tempak untuk mencegah kebocoran, harus cek seluruhnya untuk tidak ada bocoran
6. Panas suhu webcam saat nyala dan dingin airnya membuat tabung untuk mengembang dan susah untuk webcam dilihat diluar tabung.
7. Tempat letak pada ROV bisa mempengaruhi arah titik derajat pada output kompas dikarenakan setiap mempunyai gangguan magnetik.
8. HP atau elektronika yang mempunyai bagian magnetik bisa mengganggu output pada kompas ROV saat berada di jarak dekat.
9. Hasil output untuk GPS hanya bisa dicek melalui *copy paste* longitude dan latitude dan lihat hasil dari website maps, untuk memasuki langsung output GPS harus membayar subskripsi *per-call* per bulan.

#### 1.5 Metodologi

Untuk Proyek Akhir ini, setiap sistem akan dibuat, diuji masing-masing. Sensor GPS bergantung dengan sinyal yang didapatkan oleh satelit satelit yang berada di luar bumi, tetapi umumnya sinyal tidak bisa masuk kedalam air atau lebih tepat terlalu dalam air maka robot ROV akan mendapat data titik koordinat di permukaan air sebagai sistem posisi. Sensor kompas menggunakan magnetometer, dikarenakan tidak ada gangguan magnetik saat berada dibawah air maka magnetometer ini baik untuk digunakan sebagai sistem navigasi robot ROV.

## **1.6 Sistematika Laporan**

Sistematika laporan dalam penyusunan buku Proyek Akhir ini dibagi menjadi 5 (lima) Bab adalah sebagai berikut:

### **BAB I PENDAHULUAN**

Membahas tentang latar belakang, perumusan masalah, batasan masalah, maksud dan tujuan, sistematika laporan, serta relevansi Proyek Akhir yang dibuat.

### **BAB II TEORI DASAR**

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

### **BAB III PERANCANGAN DAN IMPLEMENTASI**

Membahas perencanaan dan pembuatan tentang perencanaan dan perancangan yang meliputi perangkat mekanik, perangkat elektronik, perangkat lunak (*software*) yang meliputi program yang akan digunakan untuk menjalankan alat tersebut.

### **BAB IV PENGUJIAN DAN ANALISA**

Membahas pengujian alat dan menganalisa data yang didapat dari pengujian tersebut serta membahas tentang pengukuran, pengujian, dan penganalisaan terhadap alat.

### **BAB V PENUTUP**

Berisi kesimpulan yang didapat dari Proyek Akhir ini serta saran-saran, keterbatasan, dan kendala untuk pengembangan alat ini lebih lanjut.



## **BAB II**

### **TEORI DASAR**

#### **2.1 ROV (*Remotely Operated Vehicle*)**

ROV berdasarkan Marine Technology Society ROV Committee's dalam "Operational Guidelines for ROVs" dan The national Reseaveh Council Committee's dalam "Undersea Vehicles and National Needs", didefinisikan bahwa 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 Merupakan salah satu contoh robot ROV.



Gambar 2.1 ROV (*Remotely Operated Vehicle*)

*Sumber: bluerobotics.com*

Robot bawah air ini dikendalikan oleh seseorang yang biasanya berada di kapal permukaan, menggunakan joystick dengan cara yang sama seperti bermain video game. Sekelompok kabel, atau tether, menghubungkan ROV ke kapal, mengirimkan sinyal listrik bolak-balik antara operator dan robot.

Kebanyakan ROV dilengkapi dengan setidaknya kamera diam, kamera video, dan lampu, yang berarti bahwa mereka dapat mengirim gambar dan video kembali ke kapal. Peralatan tambahan, seperti manipulator atau cutting arm, sampler air, dan instrumen yang mengukur parameter seperti kejernihan air dan suhu, juga dapat ditambahkan ke kendaraan untuk memungkinkan pengumpulan sampel.

Pertama kali dikembangkan untuk keperluan industri, seperti inspeksi internal dan eksternal dari jaringan pipa bawah laut dan pengujian struktural platform lepas pantai, ROV sekarang digunakan untuk banyak aplikasi, banyak di antaranya ilmiah. Mereka telah terbukti sangat berharga dalam eksplorasi laut dan juga digunakan untuk program pendidikan di akuarium dan untuk menghubungkan ke ekspedisi ilmiah langsung melalui Internet.

ROV memiliki ukuran mulai dari komputer kecil hingga sebesar truk kecil. ROV yang lebih besar sangat berat dan membutuhkan peralatan lain seperti kerekan untuk meletakkannya di sisi kapal dan ke dalam air.

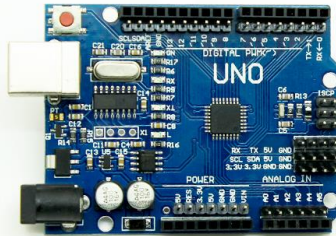
Dalam banyak kasus, operasi ROV lebih sederhana dan lebih aman untuk dilakukan daripada operasi selam atau selam jenis apa pun karena operator dapat tetap aman kapal. ROV memungkinkan kami untuk menyelidiki area yang terlalu dalam bagi manusia untuk menyelam dengan aman, dan ROV dapat bertahan di bawah air lebih lama dari penyelam manusia, memperluas waktu yang tersedia untuk eksplorasi.

ROV sains memiliki banyak bentuk dan ukuran. Karena rekaman video yang baik adalah komponen inti dari sebagian besar penelitian ilmiah di laut, ROV penelitian cenderung dilengkapi dengan sistem pencahayaan output tinggi dan kamera berkualitas siaran. Tergantung pada penelitian yang sedang dilakukan, ROV sains akan dilengkapi dengan berbagai perangkat pengambilan sampel dan sensor. Banyak dari perangkat ini adalah salah satu dari komponen eksperimental canggih yang telah dikonfigurasi untuk bekerja di lingkungan ekstrim laut dalam. Science ROVs juga menggabungkan banyak teknologi yang telah dikembangkan untuk sektor ROV komersial, seperti manipulator hidrolis dan sistem navigasi bawah laut yang sangat akurat. [1]

## **2.2 Arduino Uno (CH340 & R3 Clone)**

Arduino Uno adalah sebuah board yang menggunakan mikrokontroler Atmega328. Arduino Uno memiliki 14 pin digital (6 pin dapat digunakan sebagai output PWM), 6 input analog, sebuah 16 MHz osilator variabel, sebuah koneksi USB, sebuah konektor sumber tegangan, sebuah header ICSP, dan sebuah tombol reset. Arduino Uno memuat segala hal yang dibutuhkan untuk mendukung sebuah mikrokontroler. Hanya dengan menghubungkannya ke sebuah

komputer melalui USB atau memberikan tegangan DC dari baterai atau adaptor AC ke DC sudah dapat membuanya bekerja. Arduino Uno menggunakan Atmega16U2 yang diprogram sebagai USB to serial converter untuk komunikasi serial ke komputer melalui port USB. Gambar 2.2 merupakan Arduino Uno yang digunakan untuk Proyek Akhir ini.



Gambar 2.2 Arduino Uno  
*Sumber: tomsonelectronics.com*

Gambar diatas merupakan salah satu gambar dari versi arduino, yaitu Arduino Uno R3. “Uno” berarti satu dalam bahasa Italia dan dipilih untuk menandai rilis Arduino Software (IDE) 1.0. Papan Uno dan versi 1.0 Arduino Software (IDE) adalah versi referensi Arduino, yang sekarang berevolusi menjadi rilis yang lebih baru. Papan Uno adalah yang pertama dalam serangkaian papan USB Arduino, dan model referensi untuk platform Arduino; untuk daftar lengkap papan saat ini, masa lalu atau usang lihat indeks papan Arduino. Tabel 2.1 menunjukkan spesifikasi detail Arduino Uno.

Tabel 2.1 Spesifikasi Arduino Mega 2560

Microcontroller	Atmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6_20V
Digital I/O Pins	14

PWM	6
Analog Input Pins	6
DC Current per I/O Pin	20mA
DC Current for 3.3V Pin	50mA
Flash Memory	32 KB
SRAM	2 KB (Atmega328P)
EEPROM	1 KB (Atmega328P)
Clock Speed	16 MHz
LED_BUILTIN	13
Length	68.6 mm
Width	53.4 mm

Arduino dapat digunakan untuk berkomunikasi dengan komputer, papan Arduino lain, atau mikrokontroler lainnya. Mikrokontroler Atmega328P menyediakan komunikasi serial UART TTL (5V) yang dapat dilakukan menggunakan pin digital 0 (Rx) dan pin digital 1 (Tx). Atmega16U2 pada board menyalurkan komunikasi serial ini melalui USB dan muncul sebagai port com virtual untuk perangkat lunak di komputer. Firmware Atmega16U2 menggunakan driver USB COM standar, dan tidak diperlukan driver eksternal. Namun, pada Windows, file .inf diperlukan. Perangkat lunak Arduino termasuk monitor serial yang memungkinkan data tekstual sederhana untuk dikirim ke dan dari papan Arduino. Ada dua LED RX dan TX pada papan arduino yang akan berkedip ketika data sedang dikirim melalui chip USB-ke-serial dan koneksi USB ke komputer (bukan untuk komunikasi serial pada pin 0 dan 1). Pustaka SoftwareSerial memungkinkan komunikasi serial pada salah satu pin digital Uno. Atmega328P juga mendukung komunikasi I2C (TWI) dan SPI. Perangkat lunak Arduino mencakup perpustakaan Wire untuk menyederhanakan penggunaan bus I2C. [2]

### 2.3 Webcam

Webcam atau *Web Camera* merupakan perangkat yang berupa sebuah kamera digital yang dihubungkan ke komputer atau laptop. Layaknya kamera pada umumnya, sebuah webcam dapat mengirimkan gambar-gambar secara live dari manapun ia berada ke seluruh penjuru dunia dengan bantuan internet.. Webcam sendiri sebutan bagi kamera real-time (bermakna keadaan pada saat ini juga) yang gambarnya bisa diakses atau dilihat melalui internet, program instant messaging seperti

Yahoo Messenger, AOL Instant Messenger (AIM), Windows Live Messenger , dan Skype, dan lainnya. Istilah “webcam” sendiri mengarah pada jenis kamera yang digunakan untuk kebutuhan layanan berbasis web. Webcam sendiri biasanya digunakan untuk keperluan konferensi jarak jauh atau juga sebagai kamera pemantau.

Webcam sebuah periferan berupa kamera sebagai pengambil gambar dan mikropon ( optional ) sebagai pengambil suara/audio yang dikendalikan oleh sebuah komputer atau oleh jaringan komputer. Gambar yang diambil oleh WebCam ditampilkan ke layar monitor, karena dikendalikan oleh komputer maka ada interface atau port yang digunakan untuk menghubungkan WebCam dengan komputer atau jaringan. Ada beberapa orang mengartikan WebCam sebagai Web pages + Camera, karena dengan menggunakan WebCam untuk mengambil gambar video secara aktual bisa langsung di upload bila komputer yang mengendalikan terkoneksi internet. Gambar 2.3 merupakan webcam yang digunakan untuk proyek ini. [3]



Gambar 2.3 Webcam Logitech C270

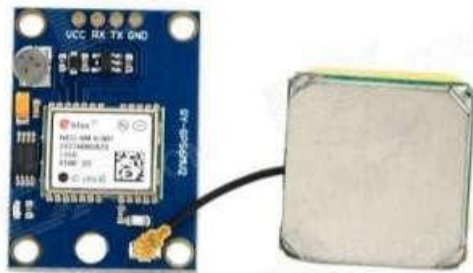
*Sumber: logitech.com*

Untuk Proyek Akhir ini menggunakan Webcam Logitech C270 yang memiliki spesifikasi sebagai berikut:

1. Resolusi 1280 x 720 piksel.
2. Rekaman video HD, pada resolusi hingga 720p.
3. Foto dapat diambil dengan resolusi maksimum 2048 x 1536 piksel, dalam format 4: 3.

## 2.4 GPS NEO-6M [4]

GPS (*Global Positioning System*) adalah 10ariab radio navigasi satelit yang dikembangkan oleh DOD (the U.S Dept. of Defense) untuk keperluan navigasi global segala cuaca dimuka bumi pada sembarang waktu. Sistem ini memungkinkan pemakai GPS menentukan posisi, kecepatan gerak dalam koordinat tiga dimensi dan waktu dengan teliti. Sistem radio navigasi satelit ini terdiri dari tiga bagian yaitu : *Space Segment*, *Control Segment*, dan *User Segment*. Penentuan posisi GPS digambarkan dengan menggunakan nilai koordinat X dan Y atau garis bujur dan garis lintang (longitude/latitude). NEO-6M adalah salah satu modul GPS yang masuk dalam salah satu seri GPS UBLOX NEO-6 yang memiliki kinerja tinggi, *receiver* yang fleksibel, murah, dan menawarkan berbagai pilihan konektivitas hanya dalam miniatur 16 x 12,2 x 2,4 mm. Mesin akuisisi yang memiliki 2 juta correlators ini memungkinkan untuk menemukan satelit secara langsung. Serta dengan desain dan teknologi yang inovatif menjadikan NEO-6M sebuah navigasi yang paling baik bahkan di lingkungan yang ekstrim. Sensor yang digunakan pada Gambar 2.3.



Gambar 2.4 Sensor GPS Ublox NEO 6M

Sumber: dx.com

NAVSTAR GPS (Navigation Satellite Timing and Ranging Global Positioning System) adalah nama asli dari Sistem GPS. Satelit di konstelasi GPS diatur menjadi enam bidang orbit yang berjarak sama di sekitar Bumi. Setiap pesawat berisi empat “slot” yang ditempati oleh satelit baseline. Pengaturan 24-slot ini memastikan pengguna dapat melihat setidaknya empat satelit dari 10ariab semua

titik di bumi. Satelit ini bertugas untuk menerima dan menyimpan data yang ditransmisikan oleh stasiun-stasiun pengendali, menyimpan dan menjaga informasi waktu berketelitian tinggi (jam atom disatelit), dan memancarkan sinyal serta informasi secara kontinyu ke perangkat penerima. Segmen pengendali bertugas untuk mengendalikan satelit dari bumi yaitu untuk melihat keadaan satelit, penentuan serta prediksi orbit, sinkronisasi waktu antar satelit, dan mengirimkan data ke satelit. Sedangkan segmen penerima bertugas menerima data dari satelit dan memprosesnya untuk menentukan posisi, arah, jarak dan waktu yang diperlukan pengguna. [4]

#### 2.4.1 Raw Data NMEA Dari NEO-6M

Saat ini di dunia GPS, NMEA adalah format data standar yang didukung oleh semua produsen GPS, seperti ASCII adalah standar untuk karakter komputer digital di dunia komputer. Tujuan dari NMEA adalah untuk memberikan pengguna peralatan kemampuan untuk mencampur dan mencocokkan perangkat keras dan perangkat lunak. Data GPS berformat NMEA juga membuat hidup lebih mudah bagi pengembang perangkat lunak untuk menulis perangkat lunak untuk berbagai penerima GPS daripada harus menulis antarmuka khusus untuk setiap penerima GPS. NMEA memiliki banyak jenis bentuk kalimat laporan, di antaranya yang paling penting adalah koordinat lintang (*latitude*), bujur (*longitude*), ketinggian (*altitude*), waktu sekarang standar UTC (*UTC time*), dan kecepatan (*speed over ground*). Tabel 2.2 menunjukkan jenis jenis kalimat NMEA. [5]

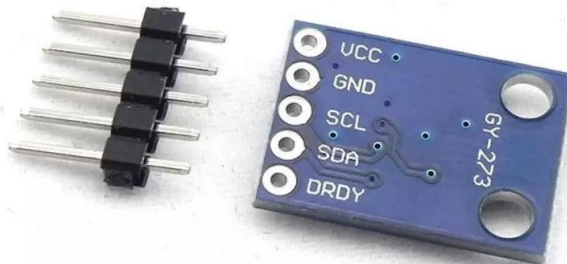
Tabel 2.2 Jenis-jenis kalimat NMEA dan Deskripsi

Kalimat	Deskripsi
\$GPGGA	<i>Global positioning system fixed data</i>
\$GPGLL	<i>Geographic position – latitude / longitude</i>
\$GPGSA	<i>GNSS DOP and active satellites</i>
\$GPGSV	<i>GNNSS satellites in view</i>
\$GPRMC	<i>Recommended minimum specific GNSS data</i>
\$GPVTH	<i>Course over ground and ground speed</i>

Dari Raw data NMEA bisa dikonversikan menjadi format yang dapat dibaca dan bermanfaat. Data ini di konverkikan menjadi format yang bisa dibaca dengan program Neo-6m.

## 2.5 GY-273 Digital Compass

Modul Kompas GY-273 adalah sebuah modul yang digunakan untuk menunjukkan arah mata 12-arah digital, atau juga disebut kompas digital. Modul ini menggunakan komponen utama berupa IC HMC5883 yang merupakan IC kompas digital 3 axis yang memiliki interface berupa 2 pin I2C. HMC5883 memiliki sensor magneto-resistive HMC118X series ber-resolusi tinggi, ditambah ASIC dengan konten amplification, automatic degaussing strap driver, offset cancellation dan 12 bit ADC yang memungkinkan keakuratan kompas mencapai 1 sampai 2 derajat. Modul ini biasa digunakan untuk keperluan 12-arah navigasi otomatis, mobile phone, netbook dan perangkat navigasi personal. Gambar 2.5 merupakan modul kompas GY-273 yang digunakan dalam project ini. Dan tabel 2.3 merupakan deskripsi untuk masing-masing . [6]



Gambar 2.5 GY-273

Sumber: [tindie.com](http://tindie.com)

Tabel 2.3 Konfigurasi Pin GY-273

Nama	Deskripsi
VCC	Power Supply (3.33V-5V)
GND	Ground
SCL	Serial Clock I2C Master/Slave Clock
SDL	Serial Data I2C Master/Slave Data
DRDY	Data Ready, Interrupt Pin.

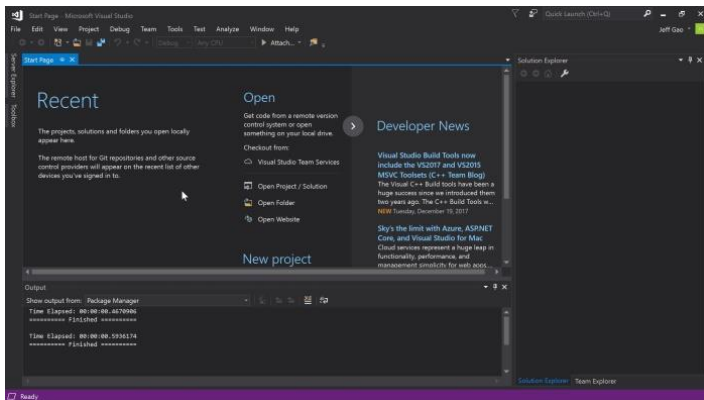


## 2.6 Visual Studio 2017

Microsoft Visual Studio merupakan sebuah software lengkap yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi console, aplikasi Windows, ataupun aplikasi Web. Gambar 2.6 merupakan menu bukanya Visual Studio.

Proyek akhir juga menggunakan software ini, yakni digunakan untuk membuat interface kontrol dari robot roV dengan menggunakan bahasa pemrograman visual basic. Dari interface yang dibuat digunakan untuk membaca input dari sistem sistem yang diimplementasi kepada ROV.

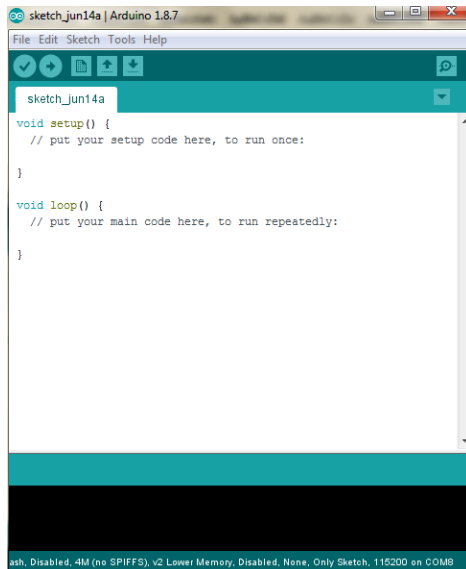
Program yang digunakan pada Visual Studios adalah Visual Basics. Visual Basic adalah implementasi Microsoft dari bahasa VB.NET dan alat terkait dan layanan bahasa. Itu diperkenalkan dengan Visual Studio .NET (2002). Microsoft telah memposisikan Visual Basic untuk Pengembangan Aplikasi Cepat. Visual Basic dapat digunakan untuk membuat aplikasi konsol maupun aplikasi GUI. Seperti Visual C #, Visual Basic juga mendukung perancang Kelas Visual Studio, perancang Formulir, dan perancang data. Seperti C #, VB.NET compiler juga tersedia sebagai bagian dari .NET Framework, tetapi layanan bahasa yang memungkinkan proyek VB.NET dikembangkan dengan Visual Studio, tersedia sebagai bagian dari yang terakhir. Gambar 2.6 merupakan tampilan Visual Studio saat dibuka. [7]



Gambar 2.6 Visual Studio 2017

## 2.7 Arduino IDE

Arduino IDE adalah sebuah editor yang digunakan untuk menulis program, mengcompile, dan mengunggah ke papan Arduino. Arduino development environment terdiri dari editor teks untuk menulis kode, area pesan, console teks, toolbar dengan tombol-tombol untuk fungsi umum, dan sederetan menu. *Software* yang ditulis menggunakan Arduino dinamakan *sketches*. *Sketches* ini ditulis di editor teks dan disimpan dengan file yang berekstensi “.ino”. Editor teks ini mempunyai fasilitas untuk *cut/paste* dan *search/replace*. Area pesan berisi umpan balik ketika menyimpan dan mengunggah file, dan juga menunjukkan jika terjadi *error*. Tampilan Arduino IDE seperti Gambar 2.7. [8]



Gambar 2. 7 Tampilan Arduino IDE

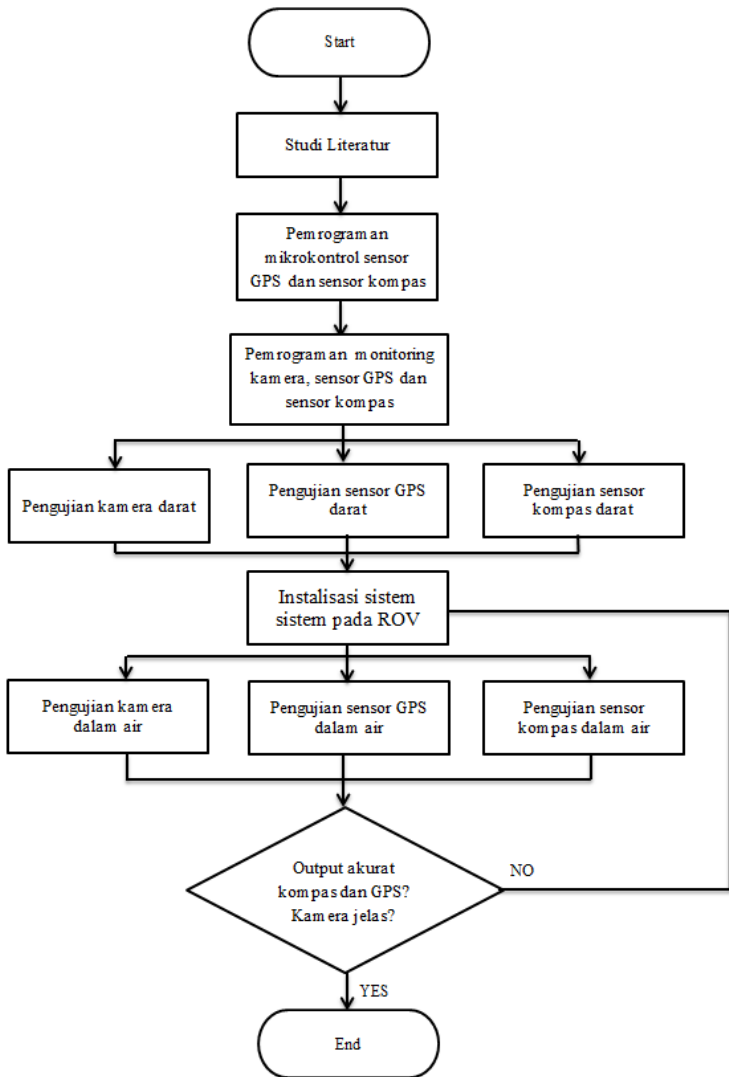
## BAB III

### PERANCANGAN SISTEM

Pada bab ini akan dijelaskan perancangan sistem mulai dari perancangan perangkat keras hingga perancangan perangkat lunak dan pembuatan. Perangkat keras yang digunakan antara lain arduino uno, sensor kompas digital, modul GPS NEO-6M, Webcam Logitech C270. Dalam perancangan elektronika sistem-sistem ROV, penempatan prangkat-prangkatnya harus ditentukan dahulu. Sebelum melakukan pemasangan elektronika sistem ke dalam tabung, harus tentukan tempat, setelah melaku testing, tempat yang di pakai untuk preangkat adalah diantara arduino mega dengan memakai pembatasan papan plastik agar tidak ada terjadinya short-circuit antara kedua arduino. Setelah menentukan menentukan penempatan untuk ketiga prangkat, Harus juga menentukan keluaran untuk kabel 10m, diaman kabelnya direncanakan untuk keluar dari samping belakangnya tabung dengan kabel untuk webcam dan GPS keluar dari satu dengan kabel kompas keluar dari yang lainnya.

Sensor kompas digital yang digunakan merupakan sensor HMC5881 GY-273 menentukan bearing atau nilai poros arah angin ROV sebagai sistem navigasi saat bergerak di bawah air. Perangkat lunak yang digunakan adalah program yang dibuat pada software Arduino dan membuat interface di software Visual Studio. Untuk penempatan snesor kompas ini berada di bawahnya arduino mega, dihubungkan deangan kabel 10m langsung ke laptop tanpa perlu *power supply*.

GPS digunakan untuk menentukan posisi ROV saat berada dipermukaan air atau apabila tidak terlalu dalam dan sinyal tidak terganggu, bisa menentukan posisi dalam air. Modul GPS yang digunakan untuk Proyek Akhir ini adalah GPS NEO-6M, sebuah sensor GPS yang simpel dan ekeftif dengan antenna eksternal. Lokasi dapat diketahui dari pembacaan sensor yang berupa *longitude* dan *latitude*, di mana pada pengujian dan analisa alat menggunakan sensor GPS dengan menggunakan hasilnya ke *Handphone* untuk mengetahui akurasi titik koordinat. Penempatan untuk GPS berada diatas perangkat lainnya dikarenakan untuk mendapatkan sinyal lebih gampang tanpa ada batasan sinyal dari perangkat lain.



Gambar 3.1 Diagram Alir Perencanaan

### **3.1 Perancangan Sistem**

Proses penyelesaian Proyek Akhir ini melalui beberapa tahap, sebagai berikut:

#### **3.1.1 Studi Literature**

Pada studi literatur meliputi pencarian dan mempelajari bahan pustaka yang berkaitan dengan segala perkembangan kompas, GPS dan kamera, mengenai perancangan dan pengetahuan mengenai komponen-komponen yang digunakan.

#### **3.1.2 Perancangan Elektronik**

Proses perancangan hardware elektronik hal pertama yang dilakukan yaitu mempelajari literatur, diharapkan setelah mempelajari literatur dapat merancang sebuah sistem elektronika yang cocok digunakan untuk sistem observasi, navigasi dan posisi. Sehingga menjadi sebuah kesatuan sistem elektronika yang handal.

#### **3.1.3 Perancangan Software**

Pada tahap perancangan software dilakukan pemrograman robot yang meliputi algoritma pemrograman proses pembacaan raw data GPS dikonversikan mejadi output data jelas, kalibrasi kompas untuk dapat pembacaan arah akurat dan dan penambahan interface untuk tampilan observasi.

#### **3.1.4 Perancangan sistem-sistem ROV**

Pada tahap ini dilakukan perancangan robot dimana hasil pemrograman dari perancangan elektrik diinstal kedalam tabung elektronika ROV.

#### **3.1.5 Pengujian robot ROV**

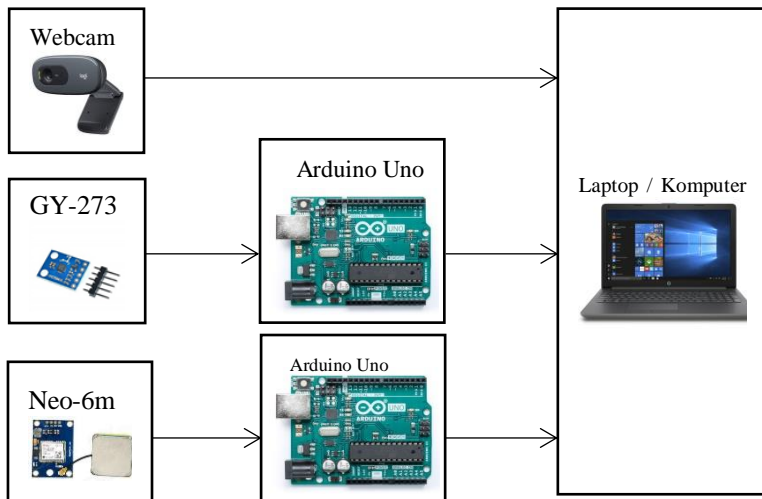
Kemudian dilakukan pengujian ketiga sistem dimana hasil dari pengujian ini akan dianalisa. Pengujian robot dilakukan pada kolam dengan kedalaman 1 meter.

#### **3.1.6 Pembuatan laporan**

Tahap terakhir dilakukan pembuatan laporan yang berkaitan dengan hasil pengujian dan analisa robot. Setelah laporan selesai dibuat maka selanjutnya adalah menarik kesimpulan sebagai has

### 3.2 Perancangan Perangkat Elektronik

Pada Bab ini akan dijelaskan mengenai Perancangan Sistem Observasi dan Sistem Navigasi ROV (*Remotely Operated Vehicle*) dengan Kamera dan GPS. Gambar 3.2 merupakan perancangan sitem keseluruhan Proyek Akhir ini.



Gambar 3.2 Diagram Fungsional Sistem Keseluruhan

Pada perancangan elektronik dibagi menjadi beberapa sub-bab yang akan dijelaskan per sub bab nya antara lain:

1. Perancarang Webcam
2. Perancangan Sistem Navigasi
3. Wiring GY-273 Pada Arduino Uno
4. Perancangan Sistem Posisi
5. Wiring GPS Neo-6m

Perancangan untuk sistem-sistem untuk ROV menggunakan dua buah Arduino UNO dan webcam yang langsung dikoneksi kepada laptop dengan kabel panjang 10m mengikuti panjang kabel yang digunakan untuk arduino mega dan power source. Ketiga sistem ini akan menggunakan program visual basic sebagai *interface* monitoring ROV, alasan utama untuk menggunakan dua buah arduino uno adalah untuk memisah tampilan output untuk GPS dan kompas.

### 3.2.1 *Setting Port Arduino Uno*

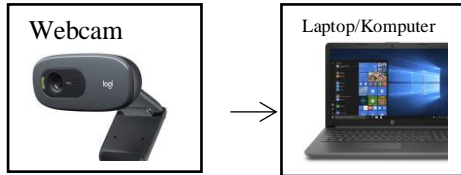
Mikrokontroler yang digunakan untuk proyek akhir ini adalah Arduino Uno. Spesifikasi untuk Arduino Uno ditunjukkan di Tabel 3.1.

Tabel 3.1 Deskripsi Pin Arduino Uno

Kategori Pin	Nama Pin	Detail
Power	Vin, 3,3V, 5V, GND	Vin: Tegangan Input ke Arduino saat menggunakan sumberdaya eksternal  3,3V & 5V: Catu daya terregulasi digunakan untuk menyalakan mikrokontroler dan komponen lain  GND: pin ground
Reset	Reset	Reset mikrokontroler
Pin Analog	A0-A5	Digunakan untuk memberikan input analog 0-5V
Pin Input/Output	Pin Digital 0-13	Bisa digunakan sebagai pin input atau output
Serial	0(RX), 1(TX)	Digunakan untuk menerima dan mengirimkan data serial TTL.
<i>External Interrupts</i>	2,3	Untuk memicu interupsi
PWM	2, 5, 6, 9, 11	Menyediakan output PWM 8-bit.
SPI	10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK)	Untuk komunikasi PWM
LED	13	Untuk nyala LED
TWI	A4(SDA), A5 (SCA)	Untuk komunikasi TWI
AREF	AREF	Untuk memberikan tegangan referensi untuk tegangan input.

### 3.2.2 Perancangan Webcam

Proses penyelesaian Proyek Akhir ini melalui tahap-tahap berikut:



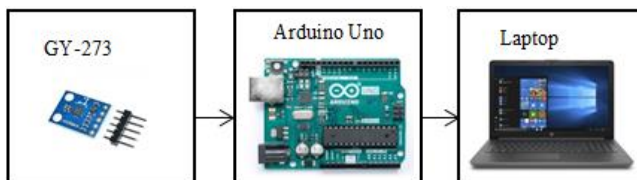
Gambar 3.3 Diagram Fungsional Sistem Observasi

Webcam yang digunakan untuk Proyek Akhir ini adalah kamera logitech C270. Untuk perancangan kamera ini cukup di pasang ke dalam ROV dalam posisiDimana bisa melihat depannya ROV dengan jelas, lalu kabel USB webcam dihubungkan langsung ke laptop seperti Gambar 3.3 dengan menggunakan program untuk menampilkan hasilnya. Gambar 3.4 merupakan penempatan kamera didalam tabung ROV.



Gambar 3.4 Penempatan Kamera Pada Tabung ROV

### 3.2.3 Perancangan Sistem Navigasi



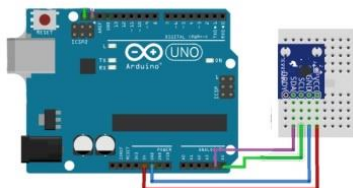
Gambar 3.5 Diagram Fungsional Sistem Navigasi



Dalam pengerjaan Proyek Akhir ini, kompas *digital* yang digunakan adalah kompas GY-273. Terdapat 2 komponen pada sistem navigasi ini yaitu sensor *Magnetometer* GY-283 *Digital Compass* dan Arduino Uno cukup sebagai kontrolernya. Gambar 3.5 merupakan koneksi sensor kompas *digital* pada arduino uno yang terus dihubungkan ke laptop.

### 3.2.4 Wiring GY-273 Pada Arduino Uno

GY-273 module berdasarkan Honeywell HMC5883L IC low-field magnetic sensor dengan sebuah interface digital untuk aplikasi kompas dan magnetometry dengan biaya yang murah dan mempunyai akurasi  $1^{\circ} - 2^{\circ}$ .

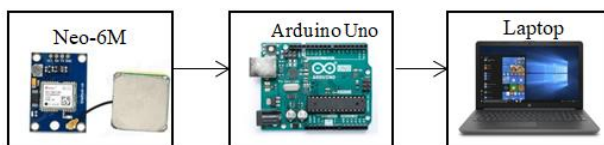


Gambar 3.6 Koneksi Kompas Digital GY-273 Pada Arduino Uno

Koneksi kompas *digital* pada Arduino Uno yang digunakan seperti Gambar 3.6. Pin yang digunakan sensor arus pada Arduino Uno adalah:

1. Pin VCC sensor GPS dihubungkan dengan pin Arduino Uno 3.3V
2. Pin GND sensor GPS dihubungkan dengan GND pin Arduino Uno
3. Pin SCL Arduino Uno dihubungkan dengan pin A5
4. Pin SDA Arduino Uno dihubungkan dengan pin A4
5. Pin DRDY dibiarkan tidak terhubung

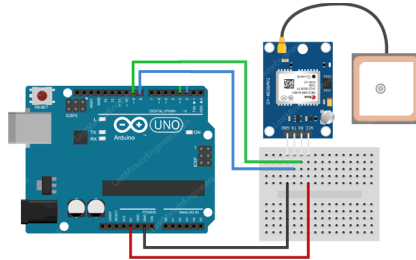
### 3.2.5 Perancangan Sistem Posisi



Gambar 3.7 Diagram Fungsional Sistem Posisi

Gambar 3.7 Merupakan perancangan untuk system posisi untuk ROV. Perancangan untuk system ini mudah dan simple, tetapi dikarenakan ROV butuhkan kabel panjang untuk perangkat apapun yang di hubungkan ke laptop akan menjadi panas dan bias rusak kepanasan, sebagai keamanan di pasang *heat pad* untuk menyerap kepanasannya. Semua program untuk GPS diletakkan didalam laptop dan disimpan kedalam arduino.

### 3.2.6 Wiring GPS Neo-6m Pada Arduino



Gambar 3.8 Wiring Neo-6m Pada Arduino Uno

*Wiring* sensor GPS yang digunakan adalah modul GPS Ublox NEO-06. Modul GPS Ublox NEO-06 dalam perancangan alat ini digunakan untuk mengetahui lokasi ROV saat berada di permukaan air. Lokasi dapat diketahui dari pembacaan sensor yang berupa *longitude* dan *latitude*, di mana pada pengujian dan analisa alat menggunakan sensor GPS dengan menggunakan hasilnya ke Google maps untuk mengetahui akurasi titik koordinat. Hasil dari pengujian dan analisa alat akan ditampilkan di BAB IV.

Koneksi sensor GPS pada Arduino yang digunakan seperti Gambar 3.8. Pin yang digunakan sensor GPS pada Arduino Uno pada adalah:

1. Pin VCC sensor GPS dihubungkan dengan pin Arduino Uno 3.3V
2. Pin GND sensor GPS dihubungkan dengan GND pin Arduino Uno
3. Pin 9 Arduino Uno dihubungkan dengan pin RX sensor GPS
4. Pin 8 Arduino Uno dihubungkan dengan pin TX sensor GPS

### 3.3 Perancangan Perangkat Lunak (*Software*)

#### 3.3.1 Kalibrasi kompas *digital*

Dilakukan kalibrasi kepada kompas *digital* dengan menggunakan program Arduino:

```
void loop()
{
    Vector mag = compass.readRaw();

    // Determine Min / Max values
    if (mag.XAxis < minX) minX = mag.XAxis;
    if (mag.XAxis > maxX) maxX = mag.XAxis;
    if (mag.YAxis < minY) minY = mag.YAxis;
    if (mag.YAxis > maxY) maxY = mag.YAxis;

    // Calculate offsets
    offX = (maxX + minX)/2;
    offY = (maxY + minY)/2;
```

Gambar 3.9 Program Kalibrasi Untuk Kompas *Digital*

Gambar 3.9 diatas merupakan program kalibrasi pada kompas *digital* GY-273 untuk mendapatkan *Offset*. *Offset* merupakan titik kejauhan nilai output dengan nilai sebenarnya. Hasil kalibrasi ini harus dimasukkan ke dalam program kompas yang berada di tunjukan dengan Gambar 3.10. Kalibrasi ini membantu program kompas untuk mendapatkan output titik derajat yang akurat,

#### 3.3.2 Program kompas *digital*

Program kompas ini merupakan program yang akan digunakan untuk mengetahui arah ROV saat berada di bawah air. Program ini dibuat dengan Arduino IDE. Library yang terpenting untuk program kompas ini adalah HMC5883l yang bisa akses pada IC yang digunakan oleh sensor GY-273.

```

    compass.setOffset( -168, -223);
}

void loop()
{
    Vector norm = compass.readNormalize();

    // Calculate heading
    float heading = atan2(norm.YAxis, norm.XAxis);

    float declinationAngle = (0 + (50.0 / 60.0)) / (180 / M_PI);
    heading += declinationAngle;

    // Correct for heading < 0deg and heading > 360deg
    if (heading < 0)
    {
        heading += 2 * PI;
    }

    if (heading > 2 * PI)
    {
        heading -= 2 * PI;
    }

    // Convert to degrees
    float headingDegrees = heading * 180/M_PI;

    // Output
    Serial.print(" Heading = ");
    Serial.print(heading);
    Serial.print(" Degree = ");
    Serial.print(headingDegrees);
    Serial.println();
}

```

Gambar 3.10 Program Kompas *Digital*

Gambar 3.10 Merupakan bagian program menampilkan arah derajat kompas tetapi perlu nilai *Offset* yang bisa didapatkan dari kalibrasi yang di lakukan dengan Gambar 3.9 dan memasuki nilai Deklinasi Magnetik.

Deklinasi magnetik, kadang-kadang disebut variasi magnetik, adalah sudut antara utara magnetik dan utara sejati. Deklinasi adalah positif timur utara sejati dan negatif ketika barat. Deklinasi magnetik berubah seiring waktu dan dengan lokasi. Sebagai kompas menunjuk dengan medan magnet lokal, nilai deklinasi diperlukan untuk mendapatkan utara yang sebenarnya, untuk mengetahui deklinasi magnetik di area bisa ke website <http://magnetic-declination.com/>. Cara untuk melakukan kalibrasi untuk deklanasi magnetik:

-Tetapkan sudut deklinasi pada lokasi Anda dan perbaiki heading

-Temukan deklinasi di: <http://magnetic-declination.com/>

-(+) Positif atau (-) untuk negatif

-Formula:  $(\text{deg} + (\text{min} / 60.0)) / (180 / M\_PI)$ ;

-float  $\text{declinationAngle} = (0 + (50.0 / 60.0)) / (180 / M\_PI)$ ;

heading + =  $\text{declinationAngle}$ ;

Saat input kedua nilai *Offset* dan Deklinasi Magnetik akan mendapatkan titik poros akurat.

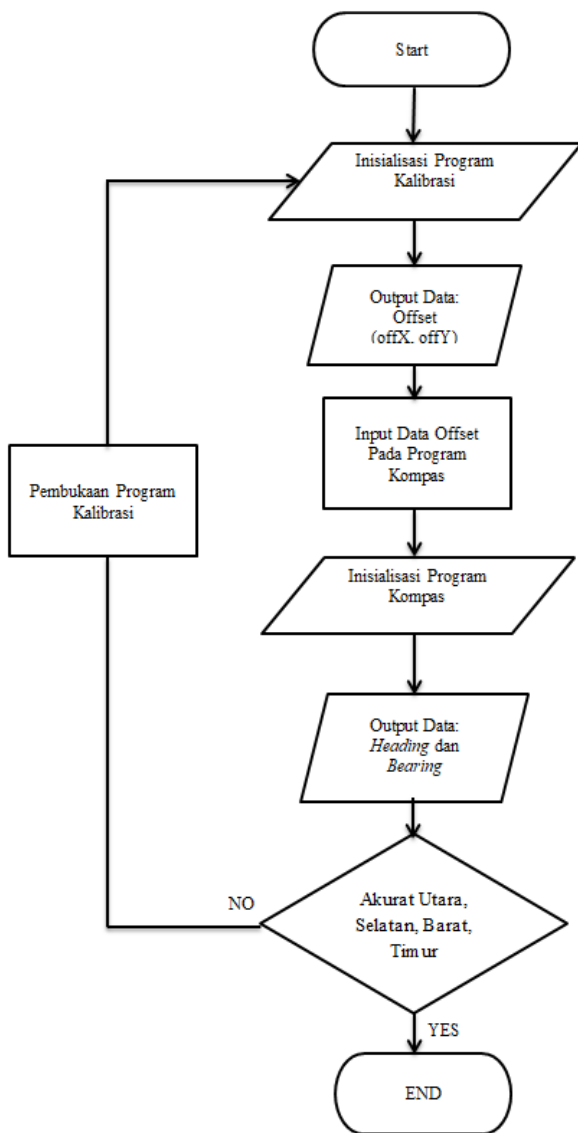
### 3.3.3 Perancangan Program Sistem Navigasi

*Flowchart* pada Sistem Navigasi digunakan untuk proses untuk kalibrasi kompas dan untuk mengetahui arah mata angin yang digunakan untuk ROV dan ditunjukkan pada Gambar 3.10.

Penjelasan pada Gambar 3.10 Sistem Navigasi:

1. *Start*: Tanda mulai programnya
2. Inisialisasi: Mulai program kalibrasi
3. Output: *offset* sebuah nilai (offX, offY)
4. Input hasil offset ke program kompas
5. Inisialisasi: Program kompas
6. Output: *Heading* dan *Bearing*
7. Output akurat *Yes* atau *No*
8. No: Kembali lagi ke program kalibrasi dan ulang proses
9. Yes: Output data kompas akurat
10. End

Pertamanya adalah mulai program kalibrasi pada kompas untuk mendapatkan nilai offset, lalu dapatkan nilai dan di simpan untuk program kompas. Masukkan output offset pada `compass.setOffset ()` dan `save` programnya. Inisialisasi programnya untuk mendapatkan output "Heading = " dan "Degrees= ". Lalu kompasnya dihadapkan ke arah utara dalam titik derajat 0° dan dibandingkan dengan kompas pada *handphone* dan liat kedua hasil di arah Timur, Barat dan Selatan, dengan *handphone* sebagai nilai derajat yang akurat, kalau nilai yang ditunjukkan oleh program jauh bedanya, balikkah kepada program kalibrasi untuk mendapatkan nilai offset yang lebih tepat. Tetapi apabila output pada program akurat, hasil program selesai.



Gambar 3.11 Flowchart Kompas *Digital*

### 3.3.4 Program GPS Neo-6m

```
void loop()
{
    while (gpsSerial.available() > 0)
        if (gps.encode(gpsSerial.read()))
            displayInfo();

    if (millis() > 5000 && gps.charsProcessed() < 10)
    {
        Serial.println("No GPS detected");
        while(true);
    }
}

void displayInfo()
{
    if (gps.location.isValid())
    {
        Serial.print("Latitude: ");
        Serial.println(gps.location.lat(), 6);
        Serial.print("Longitude: ");
        Serial.println(gps.location.lng(), 6);
        Serial.print("Altitude: ");
        Serial.println(gps.altitude.meters());
    }
    else
    {
        Serial.println("Location: Not Available");
    }
}
```

Gambar 3.12 Program GPS Untuk ROV

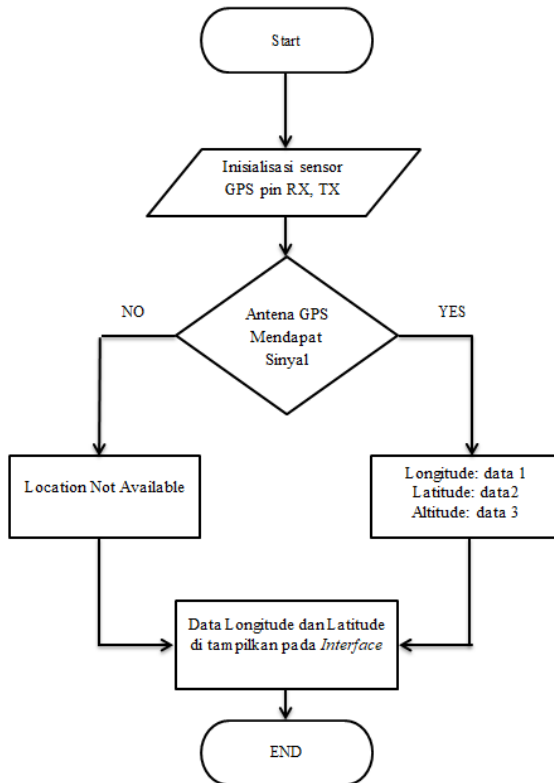
Gambar 3.12 menunjukkan program GPS yang digunakan, Program ini berfungsi untuk konversi data mentah NMEA menjadi data jelaas dan bias di baca. Saat sebelum atau tidak dapat koneksi pada satelit dan data NMEA belum lengkap, maka akan tampil output “Location: Not Available” dan akan terus menampilkan nya sampai data NMEA sudah lengkap. Tanda untuk mengetahui bahwa GPS telah menemukan satelit dan mendapat titik koordinat adalah dengan melihat dari modul Neo-6m dimana lampu LED biru nyala. Saat mendapatkan data NMEA, display serial output pada program akan menunjukkan data *longitude*, *latitude* dan *altitude*. Bisa dapat bekerja dengan data mentah dari GPS, atau dapat mengubah pesan-pesan NMEA menjadi format yang dapat dibaca dan berguna, dengan menyimpan urutan karakter ke dalam variabel. Untuk melakukan itu, sensor ini akan menggunakan *library* TinyGPS ++.

### 3.3.5 Perancangan Program Pembacaan Sensor GPS

*Flowchart* pada perancangan program pembacaan sensor GPS digunakan untuk mengetahui lokasi dengan nilai yang terbaca dari sensor GPS berupa koordinat *longitude* dan *latitude*. Alur kerja dari program pembacaan sensor GPS ditunjukkan pada Gambar 3.13.

Penjelasan *flowchart* berdasarkan Gambar 3.13 adalah sebagai berikut:

1. *Start* adalah ketika program dimulai.
2. Inisialisasi sensor GPS dibaca di pin RX TX.
3. Antena sinyal GPS mendapat sinyal.
4. Data GPS ditampilkan pada *interface*.

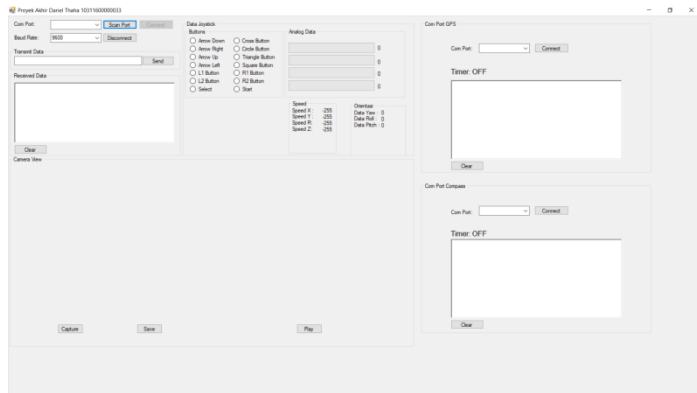


Gambar 3.13 Flowchart Sensor GPS



### 3.4 Perancangan *Interface*

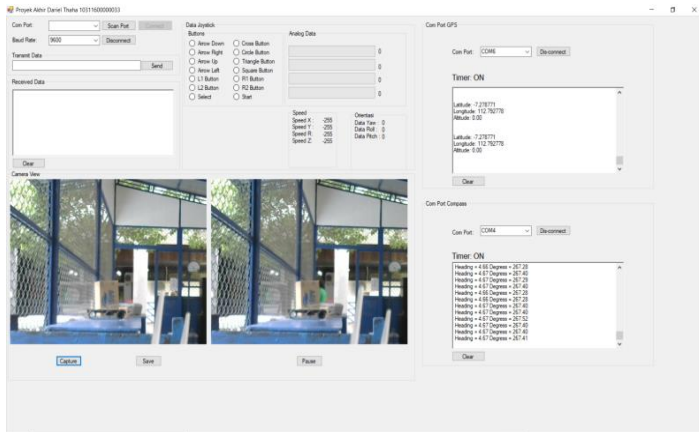
Interface pada komputer dibuat dengan menggunakan Visual Basic dengan softwer Visual Studio 2017. Pada interface ini berupa pembacaan setiap output Arduino untuk GPS, kompas *digital* dan kamera webcam. Selain itu juga untuk bagian kamera di programkan untuk bisa simpan foto yang didapatkan oleh camera. Berikut adalah tampilan keseluruhannya oleh Gambar 3.14.



Gambar 3.14 *Interface Saat Offline*

Pembacaan data GPS dan kompas menggunakan program pembacaan serial dengan menggunakan visual basic pada visual studio. Dimana hasil serial output oleh Arduino akan di tampilkan kepada visual basic lengkap dengan *COM Port* yang bisa memilih koneksi port, tombol *connect* dan *disconnect* untuk konek kepada Arduino and opsi *clear* untuk hapus semua tampilan serial output.

Sedangkan untuk kamera, diberi dua kotak tampilan, satu untuk tampil *live video* dan satu untuk *capture*. Bagian *interface* kamera dilengkapi dengan tombol *Play/Pause* untuk mulai atau berhenti *live video* yang di tampilkan. Untuk yang sebelah kirinya ada tombol *capture* yang berfungsi untuk menangkap satu frame dari video and tetap tampil gambar itu sampai di tekan tombolnya lagi. Tombol *save* berfungsi untuk menyimpan gambar yang di tangkap oleh tombol *capture* dan di jadikan foto .JPEG.



Gambar 3.15 *Interface* Saat Ketiganya Nyala

Gambar 3.15 Menunjukkan tampilan *interface* saat semua sistem-sistem nyala dan bekerja sesuai dengan yang diinginkan. Untuk di “Com Port GPS” ada pilihan semua COM yang dikoneksi dengan komputer. Untuk Proyek Akhir ini, program GPS selalu menggunakan “COM 6”, dimana COM itu adalah inout Arduino yang pengandung program GPS yang menunjukkan data koodinat yaitu:

Latitude:

Longitude:

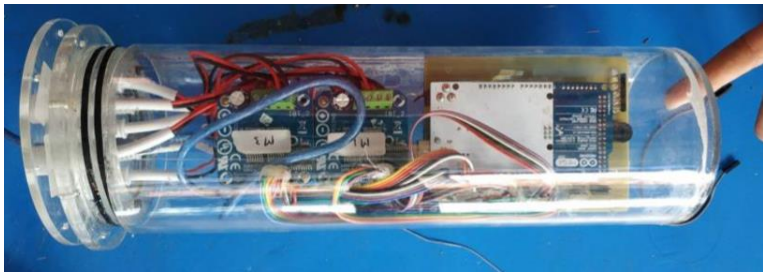
Altitude:

Sedangkan dengan “COM Port Compass” sama deangan adanya pilihan COM yang dikoneksi, tetapi tergantung kabel USB Arduino yang dikoneksi oleh pengguna ke Port USB pada laptop. Muncul untuk pilihan COM bisa sebagai “ COM 7, COM 4, COM 3”, yang manapun diantara ini di pilih akan muncul output oleh Arduino yang mengaindung program kompas, tampilan output untuk kompas yaitu:

“ Heading: xxx Degrees: xxx ”

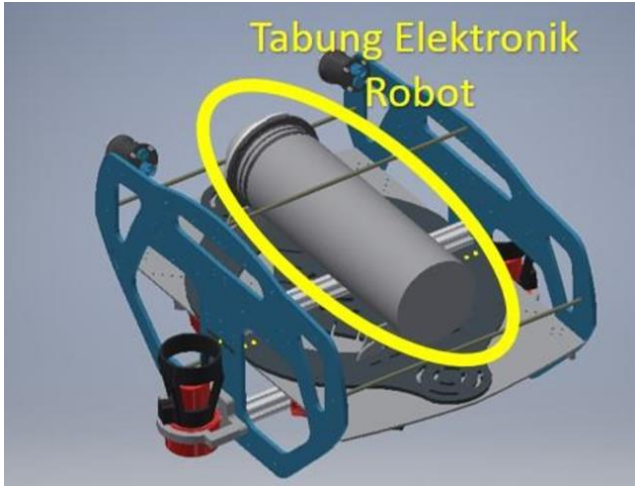
### 3.5 Perancangan Komponen Sistem-Sistem Pada ROV

Perancangan atau instalisasi komponen sistem-sistem kedalam robot ROV merupakan tahap terbesar untuk tahap perancangan. Masing masing sistem munggunakan satu buah perangkat *hardware*, tujuan untuk perancangan ini adalah untuk memasukan masing masing komponen kedalam tabung ROV tanpa adanya kerusakan atau adanya perhalangan deangan komponen lain. Tabung pada ROV suada memiliki perangkat motor dan Arduino Mega yang di instal dari proyek dahulu. Gambar 2.16 merupakan tabung ROV sebelum melakukan instalisasi *hardware* sistem-sistem.



Gambar 3.16 Tabung Elektronik ROV

Tabung pada ROV ini merupakan buatan dari Proyek Akhir dahulu yang sekarang digunakan untuk mengembangkan dan memenuhi kekurangannya. Tabung ini direncanakan menggunakan akrilik dengan ukuran diameter 10-centimeter serta tinggi tabung 29.5-centimeter dengan ketebalan tabung 3centimeter seperti pada Gambar 3.17. Tabung akrilik yang digunakan memiliki warna bening atau tidak berwarna. Warna bening pada tabung akrilik akan membantu pengguna untuk memantau keadaan komponen ketika berada di dalam tabung. Komponen elektronik yang digunakan pada robot ROV seperti mikrokontroler arduino Mega, driver motor dan sensor orientasi sudah berada pada rakitan elektronik yang terdapa didalam tabung. Sedangkan komponen-komponen tambahan sistem posisi, sistem navigasi dan sistem observasi direcanakan untuk diletakkan pada tabung elektronik yang tidak menghalangi dan tidak dihalangi oleh perangkat lain dan didesain juga untuk kedap air sehingga ketika robot menyelam, air tidak akan masuk kedalam tabung. Lalu tabung akan di tutupi oleh tutupan akrilik yang di lem dengan lem G.

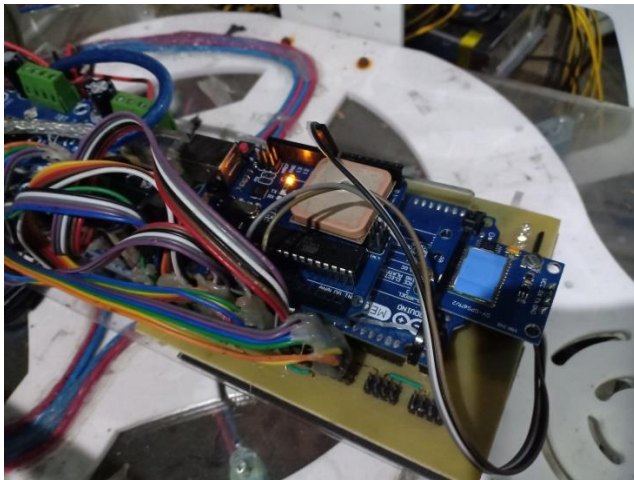


Gambar 3.17 Rancangan Desain Tempat Elektronik Robot

Untuk pertamanya adalah untuk menentukan tempat instal komponen-komponen tanpa kena halangan dari komponen lainnya. Pertamanya GPS karena perlu untuk mendapatkan sinyal saat berada di permukaan air lebih baik untuk berada di sisi atas pertengahan tabung. Perancangan Arduino untuk GPS akan diletakkan pada posisi yang ditunjukkan oleh gambar 3.18 pada lingkaran merah yaitu diatas Arduino Mega, Untuk mencegah adanya gangguan dari kedua Arduino bersentuhan diletakkan papan akrilik diantaranya ditahan dengan *double tape*. Setelah pemasangan Arduino adalah untuk koneksi kabel dari sensor GPS Neo-6m kepada Arduino Uno. Pemasangan sensor GPS ini haru perhatikan posisi yang terbaik dan berfungsi saat kabel usb dicantum pada laptop/komputer, yang dimaksud dengan berfungsi adalah pencarian koneksi dari GPS ke satelit dan sebaliknya, bisa diketahui bahwa GPS tidak berfungsi dan berfungsi dari program NMEA dimana angka angka yang di tunjukkan terlalu minial dan tidak berganti selama beberapa menit. Sensor GPS neo-6m memiliki 2 bagian yaitu sensor dan antenna eksternal. Yang terpenting dari GPS ini adalah antenanya, maka untuk perancangan ini di letakkan di tengah Arduino Uno dimana antenna berada di posis tidak gerak dan tetap berfungsi mencari sinyal satelit dan tidak menghalahi perangkat lain. Gambar 3.19 merupakan realisasi letak sensor GPS.



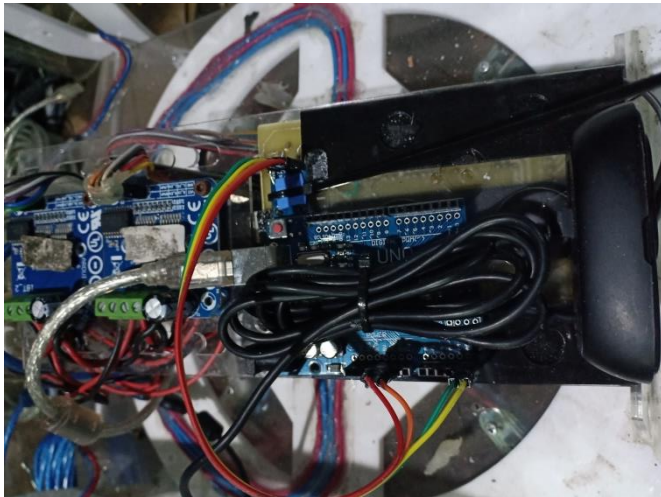
Gambar 3.18 Posisi Yang Akan Ditempati Arduino Uno



Gambar 3.19 Penempatan Arduino Dengan Sensor GPS

Setelah pemasangan Arduino di sisi atas pertengahan ROV, berikutnya adalah untuk sisi bawahnya dimana akan menjadi tempat letaknya Arduino untuk sensor kompas dan webcam sebagai sistem navigasi dan observasi. Untuk instalasi Arduino sensor kompas, proses sama seperti Arduino sensor GPS dimana menggunakan *double tape* sebagai penahan tempat, ditetapkan dalam posisi terbalik. Penempatan sensor kompas diletakkan dalam posisi terbalik dikarenakan posisi keseluruhan terbalik juga, maka saat di balik menjadi posisi benar sensor kompas akan ikut dalam posisi benar. Kabel jumper dipasang sesuai dengan Gambar 3.6 dan diposisikan agar tidak bersempitan masuk tabung akrilik. Dan untuk webcam diletakkan di paling ujung

rakitan elektronika dimana posisinya optimal untuk kamera mempunyai visualisasi yang terbaik, webcam ini di tetapkan pada tempanya dengan menggunakan lem tembak. Kabel kelebihan dari webcam cukup dengan mengumpulkannya dan diikat agar tidak mengambil banyak tempat dan diletakan ditengan Arduino Uno seperti yang di tunjukan oleh Gambar 3.20. Untuk ketiga komponen Arduino Uno dan Webcam, mengikuti panjang kabel *power supply* dan kabel motor, dihubungkan dengan kabel printer and kabel *extension* usb 10 meter.



Gambar 3.20 Sisi Bawanya Tabung ROV

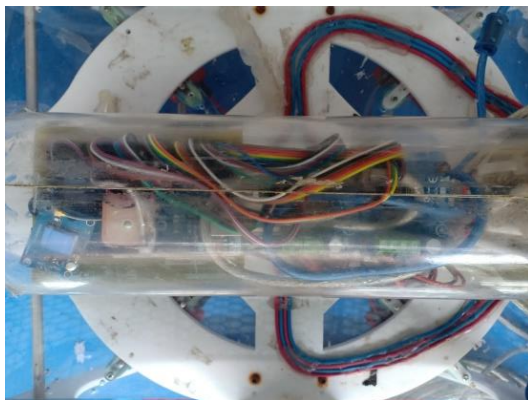
## BAB IV PENGUJIAN DAN ANALISA

Setelah melakukan prancangan sistem observasi, sistem navigasi dan sistem posisi pada robot *Remotely Operated Vehicle*, pada bab ini akan membahas pengujian dari sistem sistem yang dirancang. Bab ini bertujuan untuk mendapatkan data yang kemudian dilakukan analisa masing masing pengujian, . Berikutnya adalah urutan pengujian yang dilakukan.

1. Pengujian kebocoran *sub controller*
2. Pengujian sensor kompas *digital GY-273*
3. Pengujian modul GPS Neo-6m
4. Pengujian Webcam C270

### 4.1 Pengujian kebocoran *Sub Controller*

Untuk pengujian ini, sangat penting dikarenakan pada tabung *sub controller* bersisi dengan elektronika pada ROV seperti, arduino, sensor kompas, modul GPS, dan webcam, semua sangat penting dan alasan buat Proyek akhir ini. Semua bagian elektronika diletakkan dalam tabung ini yang terus akan dibawa saat ROV telah menyelam ke dalam air. Apabila ada terjadinya kebocoran dalam tabung ini akan berakibat fatal dan bisa merusak semua komponen. Gambar 4.1 dan 4.2 merupakan posisi GPS, kompas dan webcam dengan semua wiring padanya.



Gambar 4.1 Letak GPS Pada Tabung





Gambar 4.2 Letak Webcam & Kompas Pada Tabung

Pengujian ini dilakukan dengan dua tahap yaitu dengan siram air untuk mengetahui apabila ada tempat masuk kebocoran dan cek isi dalemnya seperti Gambar 4.3. Apabila tahap satu gagal dan ada kebocoran pada dalm tabung, maka harus bongkar tabung dan rakit kembali dan uji coba lagi. Tetapi apabila tahap satu sukses dan tidak ada tanda kebocoran, maka bisa lanjut kedua yaitu memasuki tabung kedalam wadah yang berisi air seperti Gambar 4.4.



Gambar 4.3 Pengujian Kebocoran Penyiraman Air Pada Tabung ROV



Hasil dari pengujian *waterproof* didapatkan bahwa elektronik didalam tabung terlindungi dan tidak ada air yang masuk kedalam tabung sehingga tidak terjadi short circuit akibat terkena air.



Gambar 4.4 Pengujian Kebocoran Tabung Dalam Air

#### 4.2 Pengujian Sistem Navigasi Kompas *Digital* GY-273

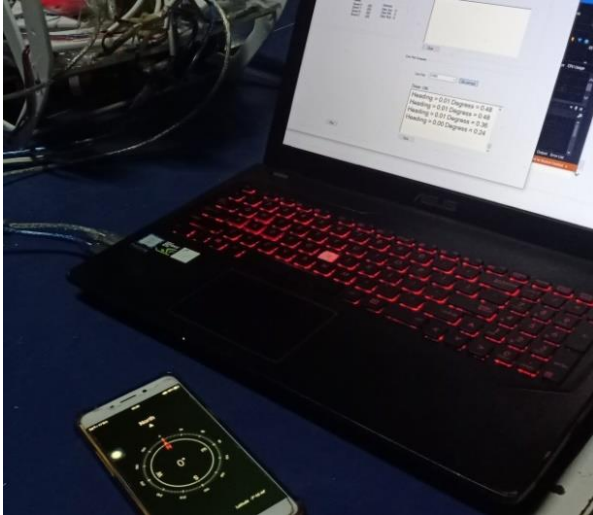
Kompas *digital* bertujuan untuk menunjukkan arah *heading* dari ROV berdasarkan arah mata angin. Pengujian dilakukan pada saat sensor didalam tabung ROV yang terus di putar 360°. Gambar 4.5 menunjukkan letak fisik kompas didalam tabung ROV.



Gambar 4.5 Letak Fisik Sensor Kompas Pada ROV

Untuk pengujian ini akan membandingkan data arah *heading* dari *handphone* dengan kompas *digital*. Dari data tadi kemudian didapatkan presentase *error* dari pembacaan data pada sensor kompas.

Gambar 4.6 merupakan gambar saat pengujian menggunakan kompas pada *handphone*. *Handphone* diletakan di depan atau jarak yang cukup jauh dari sensor kompas dikarenakan kedua sinyal kompas akan keganggu dan melibatkan data yang kurang akurat pada keduanya.



Gambar 4.6 Salah Satu Pengujian Sensor GY-273 Dibandingkan Dengan *Handphone*

Dari pengujian seperti diatas didapatkan data perbandingan sensor kompas dengan kompas *digital* yang terdapat pada *handphone* seperti pada Tabel 4.1.

Tabel 4.1 Perbandingan Data Hasil Pengujian Kompas dari Sensor Dengan Kompas *Handphone*

N O	Sudut (°)		Arah		Presentase <i>Error</i> (%)
	Kompas <i>Handphone</i>	Sensor Kompas	Kompas <i>Handphone</i>	Sensor Kompas	
1	0	0,33	Utara	Utara	0,1
2	90	91,70	Timur	Timur	1,8
3	268	270,18	Barat	Barat	0,8
4	178	180,34	Selatan	Selatan	1,5
5	222	223,04	Barat Daya	Barat Daya	0,4

6	315	315,70	Barat Laut	Barat Laut	0,2
7	37	37,87	Timur Laut	Timur Laut	0,2
8	120	126,45	Tenggara	Tenggara	5,3
Rata-rata					1,28

Berdasarkan Tabel 4.1 diatas terlihat bahwa nilai *error* paling rendah adalah yang menhadap Utara atau *heading* 0° dengan nilai eror 0.1% dikuti dengan Barat laut dan Timur laut dengan nilai eror sebesar 0.2%. Untuk mendapatkan nilai presentase eror, menggunakan rumus:

$$\%Error = \frac{\text{Kompas Sensor} - \text{Kompas Handphone}}{\text{Kompas Handphone}} \times 100 \dots \dots \dots (4.1)$$

Pembacaan eror terbesar terjadi ketika sensor diarahkan menuju arah Tenggara. Dari perbandingan kompas *digital* dengan *handphone* menunjukan adanya presentase eror sebesar 5.3%.

Variasi dengan nilai eror ini disebabkan oleh adanya aspek-aspek magnetik lainnya dan ketidak mampunya sensor untuk melakukan pembacaan pada sudut tertentu ataupun posisi *tilting* dimana pembacaan arah akan menjadi beda. Mungkin bisa mendapatkan nilai *heading* yang lebih akurat, harus dikalibrasi dengan cara program Gambar 3.9, dapatkan nilai *Offset* dan masukan ke program di Gambar 3.10. Tetapi dari semua uji coba kalibrasi, *setting* yang udah di tetapkan adalah yang paling akurat. Gambar 4.7 menunjukan tampilannya output kompas pada *Visual Basic*.

Tabel 4.2 Perbandingan Kompas Sensor dengan Kompas *Handphone* Arah Mata Angin Tenggara

N O	Sudut (°)		Arah		Presentase Error (%)
	Kompas <i>Handphone</i>	Sensor Kompas	Kompas <i>Handphone</i>	Sensor Kompas	
1	145	149,04	Tenggara	Tenggara	2,7
2	160	165,89	Tenggara	Tenggara	3,0
3	145	148,03	Tenggara	Tenggara	2,0
4	170	172,32	Tenggara	Tenggara	1,3
5	165	165,35	Tenggara	Tenggara	0,2
Rata-rata					1,84

Tabel 4.3 Perbandingan Kompas Sensor dengan Kompas *Handphone*  
Arah Mata Angin Timur Laut

N O	Sudut (°)		Arah		Presentase <i>Error (%)</i>
	Kompas <i>Handphone</i>	Sensor Kompas	Kompas <i>Handphone</i>	Sensor Kompas	
1	37	37,87	Timur Laut	Timur Laut	0,2
2	70	70,61	Timur Laut	Timur Laut	0,8
3	45	46,03	Timur Laut	Timur Laut	3,5
4	53	53,34	Timur Laut	Timur Laut	0,1
5	60	60,35	Timur Laut	Timur Laut	0,2
Rata-rata					0,9

Tabel 4.4 Perbandingan Kompas Sensor dengan Kompas *Handphone*  
Arah Mata Angin Selatan

N O	Sudut (°)		Arah		Presentase <i>Error (%)</i>
	Kompas <i>Handphone</i>	Sensor Kompas	Kompas <i>Handphone</i>	Sensor Kompas	
1	178	180,34	Selatan	Selatan	1,5
2	185	185,98	Selatan	Selatan	0,5
3	173	175,14	Selatan	Selatan	1,2
4	183	184,31	Selatan	Selatan	0,7
5	180	181,52	Selatan	Selatan	0,8
Rata-rata					0,94

Tabel 4.5 Perbandingan Kompas Sensor dengan Kompas *Handphone*  
Arah Mata Angin Barat

N O	Sudut (°)		Arah		Presentase <i>Error (%)</i>
	Kompas <i>Handphone</i>	Sensor Kompas	Kompas <i>Handphone</i>	Sensor Kompas	
1	268	270,18	Barat	Barat	0,8
2	275	275,51	Barat	Barat	0,1
3	270	270,04	Barat	Barat	0,1
4	263	263,31	Barat	Barat	0,1
5	278	279,22	Barat	Barat	0,4
Rata-rata					0.3

Tabel 4.6 Perbandingan Kompas Sensor dengan Kompas *Handphone*  
Arah Mata Angin Utara

N O	Sudut (°)		Arah		Presentase <i>Error (%)</i>
	Kompas <i>Handphone</i>	Sensor Kompas	Kompas <i>Handphone</i>	Sensor Kompas	
1	0	0,33	Utara	Utara	0,1
2	358	359,08	Utara	Utara	0,3
3	12	11,78	Utara	Utara	1,8
4	3	3,63	Utara	Utara	2,1
5	350	350,89	Utara	Utara	0,2
Rata-rata					0,9

Tabel 4.7 Perbandingan Kompas Sensor dengan Kompas *Handphone*  
Arah Mata Angin Barat Daya

N O	Sudut (°)		Arah		Presentase <i>Error (%)</i>
	Kompas <i>Handphone</i>	Sensor Kompas	Kompas <i>Handphone</i>	Sensor Kompas	
1	222	233,04	Barat Daya	Barat Daya	0,4
2	238	238,21	Barat Daya	Barat Daya	0,1
3	203	205,15	Barat Daya	Barat Daya	1,0
4	215	216,03	Barat Daya	Barat Daya	0,4
5	230	232,67	Barat Daya	Barat Daya	1,1
Rata-rata					0,6

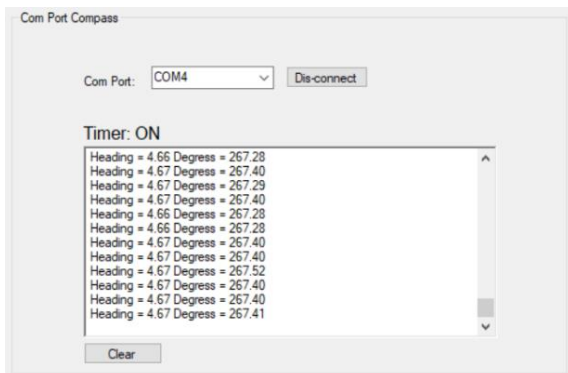
Tabel 4.8 Perbandingan Kompas Sensor dengan Kompas *Handphone*  
Arah Mata Angin Timur

N O	Sudut (°)		Arah		Presentase <i>Error (%)</i>
	Kompas <i>Handphone</i>	Sensor Kompas	Kompas <i>Handphone</i>	Sensor Kompas	
1	90	91,70	Timur	Timur	1,8
2	85	85,97	Timur	Timur	1,1
3	94	95,63	Timur	Timur	1,7
4	102	103,12	Timur	Timur	1,0
5	98	98,45	Timur	Timur	0,4
Rata-rata					1,2

Tabel 4.9 Perbandingan Kompas Sensor dengan Kompas *Handphone* Arah Mata Angin Barat Laut

N O	Sudut (°)		Arah		Presentase <i>Error (%)</i>
	Kompas <i>Handphone</i>	Sensor Kompas	Kompas <i>Handphone</i>	Sensor Kompas	
1	315	315,70	Barat Laut	Barat Laut	0,2
2	302	303,41	Barat Laut	Barat Laut	0,4
3	324	324,98	Barat Laut	Barat Laut	0,3
4	310	310,15	Barat Laut	Barat Laut	0,1
5	298	299,79	Barat Laut	Barat Laut	0,6
Rata-rata					0,32

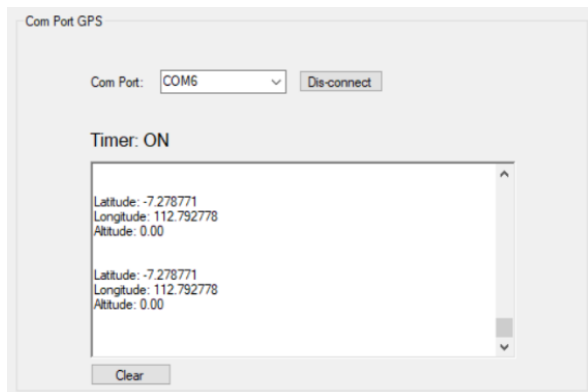
Tabel 4.2 sampai Tabel 4.9 merupakan perbandingan sensor kompas *digital* dengan kompas *handphone*, bisa dilihat bahwa arah angin Tenggara memiliki presentase eror terbesar dengan rata-rata nilai 1.84% dan arah angin Barat memiliki presentase eror terkecil dengan rata-rata nilai 0.3%. Gambar 4.7 tampilan sensor kompas di Visual Basic.



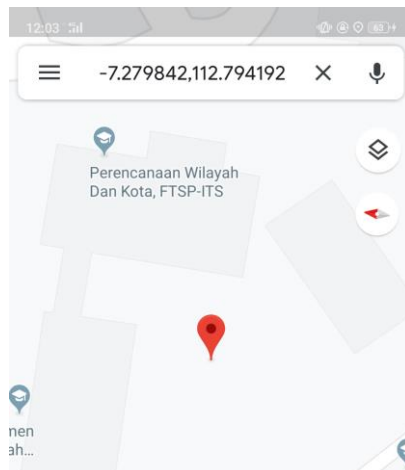
Gambar 4.7 Tampilan Sensor Kompas Pada *Visual Basic*

### 4.3 Pengujian Sistem Posisi GPS

Pengujian ini dilakukan untuk memastikan bahwa GPS telah bekerja dengan baik dan mengetahui tingkat akurasi dari sensor GPS uBlox NEO-6M. Pada pengujian ini, dibandingkan data GPS yang diterima dari sensor GPS dan *handphone* GPS. Pengujian ini dilaksanakan dengan membandingkan Gambar 4.8 sama Gambar 4.9 dengan jarak perbedaan antara kedua titik koordinat.



Gambar 4.8 Tampilan Output GPS Pada *Visual Basic*



Gambar 4.9 Lokasi GPS *Handphone*

Tabel 4. 10 Perbandingan Data Hasil Pengujian GPS dari Sensor GPS sama GPS *Handphone*

Data	Data Sensor		Data <i>Handphone</i>		Jarak (meter)
	Latitude	Longitude	Latitude	Longitude	
Lokasi 1	-7.278771	112.792778	-7.278836	112.792213	1,003
Lokasi 2	-7.278678	112.792811	-7.278659	112.792737	1,540
Lokasi 3	-7.283951	112.800016	-7.283962	112.800022	0,986
Lokasi 4	-7.287109	112.7869128	-7.2870957	112.7869048	1,721
Lokasi 5	-7.2870764	112.7868558	-7.2870628	112.7868548	1,516
Lokasi 6	-7.2870426	112.7868055	-7.2870283	112.7868048	1,592
Rata-rata					1.393

Dari Tabel 4.10 didapatkan perbandingan data hasil pengujian GPS NEO-6M dengan *handphone* GPS. Hasil yang didapat dari pembacaan sensor ada koordinat latitude dan koordinat longitude. Untuk mendapatkan jarak perbedaan GPS yaitu dengan menggunakan terorama Euclid.

Rumus yang digunakan yaitu sebagai berikut:

$$\text{Jarak} = \sqrt{(\text{Lat}_1 - \text{Lat}_2)^2 + (\text{Long}_1 - \text{Long}_2)^2} \dots\dots\dots(4.2)$$

Perbedaan data GPS yang didapatkan tidak terlalu jauh, dibuktikan dengan data GPS pada Tabel 4.2, dimana perbedaan jarak yang terjauh antara data GPS dari sensor dan data GPS dari *Handphone* adalah 1.721 meter, dan jarak perbandingan yang paling dekat adalah 0.986 meter. Dan untuk keseluruhan hasil pada pengujian didapatkan bahwa nilai error jarak meter adalah 1.393m dari jarak robot ROV pada titik koordinat sebenarnya.



#### 4.4 Pengujian Sistem Observasi Webcam C270

Salah satu pengujian pengujian kamera webcam adalah untuk melihat kualitas kamera dan fungsi untuk menyimpan gambar yang di *capture* oleh program *Visual Basic*.



Gambar 4.10 Tampilan Kamera Webcam Di *Visual Basic*



Gambar 4.11 Hasil Foto *Save* Dari Program *Visual Basic*

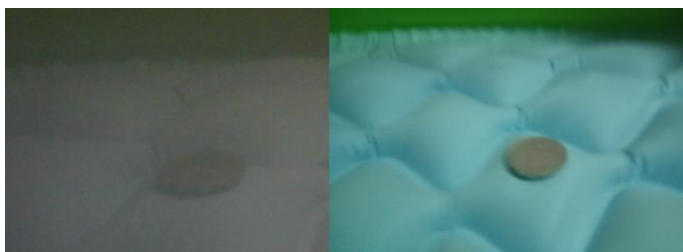
Dari kedua gambar 4.10 dan 4.11 bisa dilihat bahwa kualitas keduanya baik dan jelas. Gambar 4.11 bisa melihat lebih banyak dikarenakan sebenarnya kamera melihat sebanyak itu, Gambar 4.10 terlihat lebih kecil adalah program *Visual Basic* membatasi besar.

Untuk pengujian berikut ini adalah untuk membandingkan tampilan kamera webcam yang memiliki spesifikasi 3MP dengan kamera yang memiliki 16MP atau *handphone* saat dibawah air jernih tanpa halangan. Pengujian ini bertujuan untuk melihat perbedaan kualitas saat menggunakan kamera-kamera spesifikasi lain. Sebagai target observasi saat bawah air menggunakan sebuah koin, dan melihat kejelasan masing-masing fotonya dengan tiga jarak yang berbeda yaitu 60cm, 90cm, 120cm dengan kamera webcam 3MP (*megapixel*) ROV disebelah kiri dan kamera 16MP disebelah kanan.



Gambar 4.12 Perbandingan Kamera Jarak 60cm

Gambar 4.12 merupakan perbandingan visual kamera pada jarak 60cm. Bisa dilihat bahwa pada jarak 60cm, visual koin lebih jelas dengan kamera 16MP dibandingkan dengan kamera webcam. Tetap pada kedua foto koin terlihat dengan jelas walaupun adanya perbedaan kualitas foto.



Gambar 4.13 Pebandingan Kamera Jarak 90cm

Perbandingan kedua pada pengujian ini diganti jarak menjadi 90cm dari target koin. Terlihat bahwa kejelasan dari kedua foto lebih jelas pada kamera 16MP tetapi foto target buram sedangkan foto yang diambil oleh webcam konsisten dengan jarak sebelumnya.



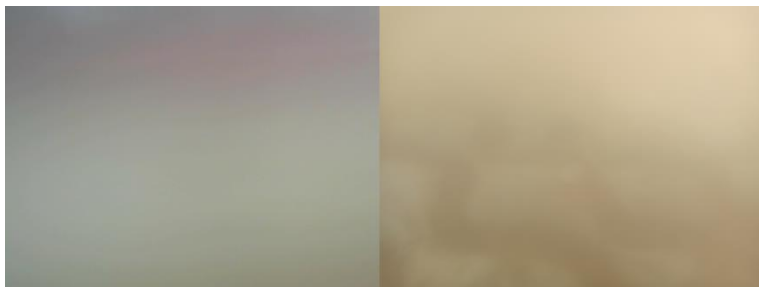
Gambar 4.14 Perbandingan Kamera Jarak 120cm

Untuk percobaan terakhirnya dengan jarak 120cm bisa terlihat bahwa kedua visualisasi pada target koin lebih buram dengan lebih kamera 16MP atau *handphone* dibandingkan kamera webcam yang tetap konsisten pada kualitas gambarnya yang hanya buram dikarenakan kekurangan adanya pencahayaan untuk membantu sinar pada target observasi.

Dari ketiga percobaan untuk pengujian observasi bisa didapatkan bahwa walaupun kualitas foto pada kamera 16MP, untuk jaraknya, semakin menjauhi semakin buram dan hanya benda-benda yang jarak dekat tetap jelas di lihat dari hasil-hasil diatas dimana lantai pada kolam jelas. Sedangkan untuk webcam dimana kualitas foto lebih rendah dengan hanya memiliki 3MP tetapi untuk kualitas ketiga foto tetap konsisten yang visual target koin tetap. Kesimpulannya adalah webcam kualitas rendah memiliki kualitas gambar foto yang konsisten terhadap jarak dalam 120cm dibawah air sedangkan untuk kamera 16MP atau *handphone* spesifikasi tinggi jelas hanya dalam jarak 60cm tetapi terhadap target foto lebih dari 60cm akan terlihat buram dan jelas untuk benda-benda yang dekat padanya.

Untuk Pengujian berikutnya adalah untuk melihat kondisi visualisasi yang di lihat oleh kedua kamera webcam ROV dengan kamera 16MP saat berada di dalam air kotor sebagai simulasi berada di air tawar. Untuk percobaan ini cukup fokus kepada kualitas foto

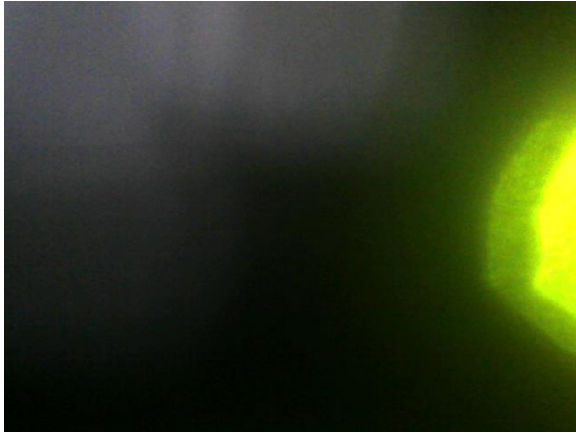
yang diambil oleh kedua kamera dan apabila bisa melihat dalam keadaan air kotor.



Gambar 4.15 Hasil Kedua Kamera Foto Air Kotor

Dari gambar yang ditunjukkan oleh Gambar 4.15, bisa dilihat bahwa dari kedua photo, yang tunjukkan oleh kamera webcam yang sebelah kanan tidak bisa melihat lingkungan dibawah air sedangkan untuk kamera 16MP mempunyai kualitas visualisasi yang hampir sama dan hanya bisa melihat lantai dengan gambar buram. Dari kedua foto, bahkan dengan kamera dengan spesifikasi tinggi, kami tidak dapat melihat apa yang ada di bawah air ketika kotor, sehingga kamera yang lemah seperti webcam tidak dapat digunakan dalam kondisi seperti itu. Mungkin mungkin bisa bekerja di air tawar yang agak jernih dengan mata telanjang dan cukup jernih untuk semua jenis kamera untuk melihat tanpa gangguan kotoran seperti pengujian yang dilakukan sebelumnya dengan air jernih.

Dari pengujian untuk video streaming dari webcam, gerakan pada kamera agak lambat dengan 15fps. Tetapi saat pengujian diteruskan, penutup plastic pada tabung mengembun sampai terjadinya ketidak terlihatannya dari visual kamera tidak jelas seperti Gambar 4.16. Hal ini terjadi karena saat kamera dinyalakan adanya suhu hangat/panas yang muncul seperti semua perangkat elektronik bertemu dengan suhu dinginnya air yang diluar membuatnya mengembun seperti yang ditunjukkan oleh Gambar 4.17. Yang membedakan kamera dengan perangkat lain adalah penempatannya, webcam ditempatkan di paling depannya tabung karena disitu adalah posisi yang paling jelas untuk gambar yang di ambil olehnya dan apabila dipakai selama 30menit atau lebih akan membuatnya panas.



Gambar 4.16 Foto Kamera Saat Plastik Didepannya Mengembun



Gambar 4.17 Mengembunnya Akrilik Pada Tabung ROV

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

## **BAB V**

### **PENUTUP**

Setelah melakukan perancangan dan pembuatan alat serta pengujian dan analisa, maka dapat ditarik kesimpulan dan saran dari kegiatan yang telah dilakukan untuk pengembangan Proyek Akhir ini.

#### **5.1 Kesimpulan**

Dari seluruh tahapan yang sudah dilaksanakan pada penyusunan Proyek Akhir ini, mulai dari studi literatur, perancangan dan pembuatan sampai pada pengujiannya maka dapat disimpulkan bahwa:

1. Webcam bisa berkerja sebagai sistem observasi ROV tetapi mengalami kepanasan dan bisa *lagging* kalau pakai selama 30 menit atau lebih
2. Variasi dengan nilai eror kompas disebabkan oleh adanya aspek-aspek magnetik lainnya dan ketidak mampunya GY-273 untuk melakukan pembacaan pada sudut tertentu atau posisi *tilting* dimana pembacaan arah akan menjadi beda. Saat berada di posisi datar, nilai presentase eror yang terbesar hanya 5% berhadapan arah tenggara dan yang terkecil dengan nilai 0.1% berhadapan di arah Utara.
3. GPS berkerja tetapi lama untuk mendapatkan sinyal dari satelit untuk mendapatkan titik koordinat dengan nilai titik eror jarak terbesar 1.7m dan eror terkecil 0.9m.
4. Saat Webcam berkerja akan menjadi panas dan saat ROV masuk kedalam air yang cukup dingin maka plastik yang didepan webcam akan mengembung jadi pengelihatannya ada di halangi.
5. Webcam yang digunakan dalam air kondisi jernih konsisten kualitas foto pada jarak dalam 120cm untuk observasi dibandingkan dengan kamera lain yang berkualitas lebih tinggi tetapi perlu bantuan pencahayaan untuk gambar foto yang divisualisasikan lebih jelas.
6. *Interface* untuk monitoring sistem sistem pada ROV menggunakan program Visual Basic, dengan monitoring dengan output serial ditampilkan dan tampilan kamera yang bisa di *play* dan *pause*, lengkap dengan fitur *save*.

## 5.2 Saran

Untuk lebih memperbaiki dan menyempurnakan kinerja dari alat ini, maka perlu disarankan :

1. Modul GPS dan Antenna diganti dengan yang lebih kuat agar bisa nyambung ke satelit dengan lebih cepat.
2. Kamera Webcam di ganti dengan kamera yang lebih baik agar bisa mendapatkan kualitas gambar lebih jelas dan *streaming* video yang lebih lancar
3. Cara untuk merendahkan kesalahan arah angin saat ROV *tilting* yang bisa membantu saat dibawah air
4. Adanya pencahayaan untuk membantu pengelihatan kamera lebih jelas saat berada dibawah air



## DAFTAR PUSTAKA

- [1] Sulaiman Ali, Rancang Bangun *Underwater Remotely Operated Vehicles* Untuk Lingkungan Air Tawar, **Proyek Akhir**, Program D3 Teknik Elektro FTVOKASI, Surabaya, 2019.
- [2] E. Brian, **Beginning Arduino Programming**, USA: Technology in Action, 2003.
- [3] E. Brian, **Logitech C270**, USA: Technology in Action, 2003.
- [4] Malvin Faizal Syufan, Ahmad Wahyu Purwandi, dan Moh. Abdullah Anshori, “Rancang Bangun Sistem Navigasi Autonomous Pada Mobil RC Menggunakan Kombinasi Database MAP Dan GPS Untuk *Monitoring Area Berbasis Web*”, **Jurnal Jartel**, Politeknik Negeri Malang, Vol 7 No. 2, Halaman 112-117, 2018.
- [5] SiRF Technology, **NMEA Reference Manual**, USA: SiRF, 2005.
- [6] Honeywell, **3-Axis Digital Compass IC HMC5883L**, Plymouth, Honeywell, 2013.
- [7] Liew, **Visual basic 2017 Mode Easy**, United state: Microsoft, 2017.
- [8] H. Santoso, **Arduino Untuk Pemula**, Jakarta: Elangsakti, 2015.

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

## LAMPIRAN A

### Lampiran Program *Visual Basic*

Imports System.Runtime.InteropServices

Imports System.IO.Ports

Imports Emgu.CV

Imports Emgu.CV.Structure

Public Class Form1

Dim capWebcam As VideoCapture

Dim blnCaptureInProgress As Boolean = False

Dim imgOriginal As Image(Of Bgr, Byte)

Private Delegate Sub delegateReadData()

Shared \_continue As Boolean

Shared \_serialPort As SerialPort

Dim yaw, pitch, roll, error\_yaw

Dim comPORT As String

Dim comPORT1 As String

Dim pos As Long

Dim ypos As Long

Dim xpos As Long

Dim dir As Char

Dim dir1 As Char

Dim speed1 As Long

Dim dir2 As Char

Dim speed2 As Long

Dim berhenti As Long

Dim open As Boolean

Dim receive As Boolean

Dim receivedData As String = ""

Dim receivedData2 As String = ""

Dim direct As Int16

Dim pos1 As Int16

Dim posy As Int16

Dim posx As Int16

Dim vxpos As Char

Dim vypos As Char

Dim vzpos As Char

Dim posz As Int16

Dim zpos As Int16

Dim posr As Int16

Dim rpos As Int16

Declare Function joyGetPosEx Lib "winmm.dll" (ByVal uJoyID As Integer, ByRef pji As JOYINFOEX) As Integer

<StructLayout(LayoutKind.Sequential)>

Public Structure JOYINFOEX

Public dwSize As Integer

Public dwFlags As Integer

Public dwXpos As Integer

Public dwYpos As Integer

Public dwZpos As Integer

Public dwRpos As Integer

Public dwUpos As Integer

Public dwVpos As Integer

Public dwButtons As Integer

Public dwButtonNumber As Integer

Public dwPOV As Integer

Public dwReserved1 As Integer

Public dwReserved2 As Integer

End Structure

```
Dim myjoyEX As JOYINFOEX
```

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e  
As System.EventArgs) Handles MyBase.Load
```

```
    If CmbBaud.Items.Count > 0 Then
```

```
        CmbBaud.SelectedIndex = 0 ' The first item has index 0 '
```

```
    End If
```

```
    recieve = False
```

```
    BtnCon.Enabled = False
```

```
    open = False
```

```
    myjoyEX.dwSize = 64
```

```
    myjoyEX.dwFlags = &HFF ' All information
```

```
    Timer1.Interval = 10 'Update at 5 hz
```

```
    Timer1.Start()
```

```
    Timer2.Interval = 200 'Update at 5 hz
```

```
    Timer2.Start()
```

```
    Timer4.Enabled = False
```

```
    comPORT = ""
```

```
    For Each sp As String In My.Computer.Ports.SerialPortNames
```

```
        CmbPort2.Items.Add(sp)
```

```
    Next
```

```
    Timer5.Enabled = False
```

```
    comPORT = ""
```

```
    For Each sp As String In My.Computer.Ports.SerialPortNames
```

```
        CmbPort3.Items.Add(sp)
```

```
    Next
```

```
End Sub
```

```
Sub New()
```

```
    InitializeComponent()
```

End Sub

Private Sub Timer1\_Tick\_1(sender As Object, e As EventArgs)  
Handles Timer1.Tick

' Get the joystick information  
Call joyGetPosEx(0, myjoyEX)

With myjoyEX

'Up to six axis supported  
Label3.Text = ypos.ToString  
Label4.Text = xpos.ToString  
Label51.Text = xpos.ToString - 255  
Label52.Text = ypos.ToString - 255  
Label54.Text = zpos.ToString - 255  
Label55.Text = rpos.ToString - 255

Label21.Text = zpos.ToString  
Label22.Text = rpos.ToString  
Label5.Text = .dwXpos.ToString  
Label6.Text = .dwYpos.ToString  
Label7.Text = .dwZpos.ToString  
Label8.Text = .dwRpos.ToString  
'Label9.Text = .dwUpos.ToString  
'Label10.Text = .dwVpos.ToString  
Label11.Text = .dwButtons.ToString("") 'Print in Hex, so can  
see the individual bits associated with the buttons  
Label12.Text = .dwButtonNumber.ToString 'number of  
buttons pressed at the same time  
Label13.Text = (.dwPOV / 100).ToString 'POV hat (in  
1/100ths of degrees, so divided by 100 to give degrees)  
Label19.Text = .dwPOV.ToString

'If xpos.ToString >= 0 Or ypos.ToString >= 0 Or rpos.ToString  
>= 0 Or zpos.ToString >= 0 Then  
' direct = 0  
' Label23.Text = direct.ToString

```

'ElseIf xpos.ToString < 0 Or ypos.ToString < 0 Or
rpos.ToString < 0 Or zpos.ToString < 0 Then
' direct = 1
' Label23.Text = direct.ToString

If xpos.ToString < 0 Then
direct = 1
Label23.Text = direct.ToString
ElseIf xpos.ToString >= 0 Then
direct = 0
Label23.Text = direct.ToString
End If

```

```

If .dwYpos < 32767 And .dwXpos = 32767 Then
dir = "W"
pos = (.dwYpos) / 65535 * 510
ypos = pos
Label14.Text = "Vy up"
ElseIf .dwYpos > 32767 And .dwXpos = 32767 Then
dir = "X"
pos = (.dwYpos) / 65535 * 510
ypos = pos
Label14.Text = "Vy down"
ElseIf .dwXpos < 32767 And .dwYpos = 32767 Then
dir = "A"
pos = (.dwXpos) / 65535 * 510
xpos = pos
Label14.Text = "Vx left"
ElseIf .dwXpos > 32767 And .dwYpos = 32767 Then
dir = "D"
pos = (.dwXpos) / 65535 * 510
xpos = pos
Label14.Text = "Vx Right"
ElseIf .dwYpos < 32767 And .dwXpos < 32767 Then
dir = "Q"

```

```

    posy = (.dwYpos) / 65535 * 510
    posx = (.dwXpos) / 65535 * 510
    ypos = posy
    xpos = posx
    Label14.Text = "Vy up & Vy left"
ElseIf .dwYpos < 32767 And .dwXpos > 32767 Then
    dir = "E"
    posy = (.dwYpos) / 65535 * 510
    posx = (.dwXpos) / 65535 * 510
    ypos = posy
    xpos = posx
    Label14.Text = "Vy up & Vy Right"
ElseIf .dwYpos > 32767 And .dwXpos < 32767 Then
    dir = "Z"
    posy = (.dwYpos) / 65535 * 510
    posx = (.dwXpos) / 65535 * 510
    ypos = posy
    xpos = posx
    Label14.Text = "Vy down & Vy left"
ElseIf .dwYpos > 32767 And .dwXpos > 32767 Then
    dir = "C"
    posy = (.dwYpos) / 65535 * 510
    posx = (.dwXpos) / 65535 * 510
    ypos = posy
    xpos = posx
    Label14.Text = "Vy down & Vy Right"

ElseIf .dwButtons = 4 Then
    dir = "M"
    RadioButton1.Checked = True
    Label14.Text = "cross"
ElseIf .dwButtons = 2 Then
    dir = "L"
    RadioButton2.Checked = True
    Label14.Text = "circle"
ElseIf .dwButtons = 1 Then
    dir = "I"
    RadioButton3.Checked = True
    Label14.Text = "triangle"

```



```

ElseIf .dwButtons = 8 Then
    dir = "J"
    RadioButton4.Checked = True
    Label14.Text = "square"
ElseIf .dwButtons = 32 Then
    dir = "O"
    RadioButton5.Checked = True
    Label14.Text = "R1"
ElseIf .dwButtons = 128 Then
    dir = "U"
    RadioButton6.Checked = True
    Label14.Text = "R2"
ElseIf .dwButtons = 512 Then
    dir = "K"
    RadioButton7.Checked = True
    Label14.Text = "start"
ElseIf .dwButtons = 256 Then
    dir = "G"
    RadioButton8.Checked = True
    Label14.Text = "select"
ElseIf .dwButtons = 16 Then
    dir = "R"
    RadioButton9.Checked = True
    Label14.Text = "L1"
ElseIf .dwButtons = 64 Then
    dir = "Y"
    RadioButton10.Checked = True
    Label14.Text = "L2"
ElseIf (.dwPOV / 100) = 270 Then
    dir = "F"
    RadioButton11.Checked = True
    Label14.Text = "Arrow left"
ElseIf (.dwPOV / 100) = 0 Then
    dir = "T"
    RadioButton12.Checked = True
    Label14.Text = "Arrow UP"
ElseIf (.dwPOV / 100) = 90 Then
    dir = "H"
    RadioButton13.Checked = True

```

```

Label14.Text = "Arrow right"
ElseIf (.dwPOV / 100) = 180 Then
    dir = "B"
    RadioButton14.Checked = True
    Label14.Text = "Arrow Down"
Else
    dir = "0"
    posx = (.dwXpos) / 65535 * 510
    posy = (.dwYpos) / 65535 * 510
    posz = (.dwZpos) / 65535 * 510
    posr = (.dwRpos) / 65535 * 510
    ypos = (.dwYpos) / 65535 * 510
    xpos = (.dwXpos) / 65535 * 510
    zpos = (.dwZpos) / 65535 * 510
    rpos = (.dwRpos) / 65535 * 510
    Label14.Text = "Berhenti"
    RadioButton1.Checked = False
    RadioButton2.Checked = False
    RadioButton3.Checked = False
    RadioButton4.Checked = False
    RadioButton5.Checked = False
    RadioButton6.Checked = False
    RadioButton7.Checked = False
    RadioButton8.Checked = False
    RadioButton9.Checked = False
    RadioButton10.Checked = False
    RadioButton11.Checked = False
    RadioButton12.Checked = False
    RadioButton13.Checked = False
    RadioButton14.Checked = False
End If

If .dwRpos < 32767 And .dwZpos = 32767 Then
    dir = "W"
    pos1 = (.dwRpos) / 65535 * 510
    rpos = pos1
    Label14.Text = "Vy up"
ElseIf .dwRpos > 32767 And .dwZpos = 32767 Then

```

```

dir = "X"
pos1 = (.dwRpos) / 65535 * 510
rpos = pos1
Label14.Text = "rmudun"
Elseif .dwZpos < 32767 And .dwRpos = 32767 Then
dir = "A"
pos1 = (.dwZpos) / 65535 * 510
zpos = pos1
Label14.Text = "Vx left"
Elseif .dwZpos > 32767 And .dwRpos = 32767 Then
dir = "D"
pos1 = (.dwZpos) / 65535 * 510
zpos = pos1
Label14.Text = "Vx Right"
Elseif .dwRpos < 32767 And .dwZpos < 32767 Then
dir = "Q"
posr = (.dwRpos) / 65535 * 510
posz = (.dwZpos) / 65535 * 510
rpos = posr
zpos = posz
Label14.Text = "Vy up & Vy left"
Elseif .dwRpos < 32767 And .dwZpos > 32767 Then
dir = "E"
posy = (.dwRpos) / 65535 * 510
posz = (.dwZpos) / 65535 * 510
rpos = posr
zpos = posz
Label14.Text = "Vy up & Vy Right"
Elseif .dwRpos > 32767 And .dwZpos < 32767 Then
dir = "Z"
posr = (.dwRpos) / 65535 * 510
posz = (.dwZpos) / 65535 * 510
rpos = posr
zpos = posz
Label14.Text = "Vy down & Vy left"
Elseif .dwRpos > 32767 And .dwZpos > 32767 Then
dir = "C"
posr = (.dwRpos) / 65535 * 510
posz = (.dwZpos) / 65535 * 510

```

```
rpos = posr
zpos = posz
Label14.Text = "Vy down & Vy Right"
End If
```

```
Label20.Text = dir
ProgressBar1.Value = .dwXpos
ProgressBar2.Value = .dwYpos
ProgressBar3.Value = .dwZpos
ProgressBar4.Value = .dwRpos
```

```
'If open = True Then
' SerialPort1.Write("#")
' SerialPort1.Write(xpos)
' SerialPort1.Write("x")
' SerialPort1.Write(ypos)
' SerialPort1.Write("x")
' SerialPort1.Write(zpos)
' SerialPort1.Write("x")
' SerialPort1.Write(rpos)
' SerialPort1.Write("u")
'End If
'If open = True And recieve = True Then
' SerialPort1.Write("#")
' SerialPort1.Write(xpos)
' SerialPort1.Write("x")
' SerialPort1.Write(ypos)
' SerialPort1.Write("x")
' SerialPort1.Write(zpos)
' SerialPort1.Write("x")
' SerialPort1.Write(rpos)
' SerialPort1.Write("u")
' receivedData = ReceiveSerialData()
' RichTextBox1.Text &= receivedData
```

```

    ' RichTextBox1.SelectionStart = Len(RichTextBox1.Text)
    ' ' Scrolls to the caret
    ' RichTextBox1.ScrollToCaret()
    ' ' Select the range
    ' RichTextBox1.Select()

    ' recieve = False
    'End If
End With

```

End Sub

```

Function ReceiveSerialData() As String
    Dim Incoming As String
    Try
        Incoming = SerialPort1.ReadExisting()
        If Incoming Is Nothing Then
            Return "nothing" & vbCrLf
        Else
            Return Incoming
        End If
    Catch ex As TimeoutException
        Return "Error: Serial Port read timed out."
    End Try

```

End Function

```

Private Sub SerialPort1_DataReceived(sender As Object, e As
SerialDataReceivedEventArgs) Handles SerialPort1.DataReceived
    Console.WriteLine("data recieved")
    recieve = True
End Sub

```

```

Private Sub BtnScanPort_Click(sender As Object, e As EventArgs)
Handles BtnScanPort.Click
    CmbPort.Items.Clear()
    Dim myPort As Array
    Dim i As Integer

```

```

myPort = SerialPort.GetPortNames()
CmbPort.Items.AddRange(myPort)
i = CmbPort.Items.Count
i = i - 1
Try
    CmbPort.SelectedIndex = i
Catch ex As Exception
    Dim result As DialogResult
    result = MessageBox.Show("com port not detected", "Warning
!!!", MessageBoxButtons.OK)
    CmbPort.Text = ""
    CmbPort.Items.Clear()
    Call Form1_Load(Me, e)
End Try
BtnCon.Enabled = True
BtnCon.BringToFront()
CmbPort.DroppedDown = True
End Sub

```

```

Private Sub Button1_Click(sender As Object, e As EventArgs)
Handles BtnCon.Click
    BtnCon.Enabled = False
    BtnCon.SendToBack()

    SerialPort1.BaudRate = CmbBaud.SelectedItem
    SerialPort1.PortName = CmbPort.SelectedItem
    SerialPort1.Open()
    Timer1.Start()
    open = True

    BtnDiscon.Enabled = True
    BtnDiscon.BringToFront()
End Sub

```

```

Private Sub BtnDiscon_Click(sender As Object, e As EventArgs)
Handles BtnDiscon.Click
    Console.WriteLine("Clicked")
    BtnDiscon.Enabled = False
    BtnDiscon.SendToBack()

```

```

    Timer1.Stop()
    SerialPort1.Close()
    open = False

    BtnCon.Enabled = True
    BtnCon.BringToFront()
End Sub

Private Sub Clear_BTN_Click(sender As Object, e As EventArgs)
Handles Clear_BTN.Click
    RichTextBox1.Text = ""
End Sub

Private Sub Timer2_Tick(sender As Object, e As EventArgs)
Handles Timer2.Tick
    If open = True And recieve = True Then
        ' SerialPort1.Write("#")
        ' SerialPort1.Write(xpos)
        ' SerialPort1.Write("x")
        ' SerialPort1.Write(ypos)
        ' SerialPort1.Write("x")
        ' SerialPort1.Write(zpos)
        ' SerialPort1.Write("x")
        ' SerialPort1.Write(rpos)
        ' SerialPort1.Write("u")
    End If
    If open = True And recieve = True Then
        SerialPort1.Write("#")
        SerialPort1.Write(xpos)
        SerialPort1.Write("x")
        SerialPort1.Write(ypos)
        SerialPort1.Write("x")
        SerialPort1.Write(zpos)
        SerialPort1.Write("x")
        SerialPort1.Write(rpos)
        SerialPort1.Write("u")
        receivedData = ReceiveSerialData()
        RichTextBox1.Text &= receivedData
    End If
End Sub

```

```

        trial()

        RichTextBox1.SelectionStart = Len(RichTextBox1.Text)
        ' Scrolls to the caret
        RichTextBox1.ScrollToCaret()
        ' Select the range
        RichTextBox1.Select()

        recieve = False
    End If

End Sub

Private Sub Timer3_Tick(sender As Object, e As EventArgs)
Handles Timer3.Tick
    Try
        PictureBox2.Image = capWebcam.QueryFrame.Bitmap()
    Catch ex As Exception
        capWebcam = New VideoCapture(1)
    End Try

End Sub

Private Sub BtnPlay_Click(sender As Object, e As EventArgs)
Handles BtnPlay.Click
    Timer3.Start()
    BtnPlay.SendToBack()

End Sub

Private Sub BtnPause_Click(sender As Object, e As EventArgs)
Handles BtnPause.Click
    Timer3.Stop()
    BtnPause.SendToBack()

End Sub

```



```
Private Sub BtnCapture_Click(sender As Object, e As EventArgs)
Handles BtnCapture.Click
```

```
    PictureBox1.Image = PictureBox2.Image
End Sub
```

```
Private Sub BtnSave_Click(sender As Object, e As EventArgs)
Handles BtnSave.Click
```

```
    SaveFileDialog1.ShowDialog()
    PictureBox1.Image.Save(SaveFileDialog1.FileName + ".jpeg",
System.Drawing.Imaging.ImageFormat.Jpeg)
End Sub
```

```
Private Sub GroupBox10_Enter(sender As Object, e As EventArgs)
Handles GroupBox10.Enter
```

```
End Sub
```

```
Private Sub CmbPort2_SelectedIndexChanged(sender As Object, e
As EventArgs) Handles CmbPort2.SelectedIndexChanged
```

```
    If (CmbPort2.SelectedItem <> "") Then
        comPORT1 = CmbPort2.SelectedItem
    End If
End Sub
```

```
Private Sub BtnConnect1_Click(sender As Object, e As EventArgs)
Handles BtnConnect1.Click
```

```
    If (BtnConnect1.Text = "Connect") Then
        If (comPORT1 <> "") Then
            SerialPort2.Close()
            SerialPort2.PortName = comPORT1
            SerialPort2.BaudRate = 9600
            SerialPort2.DataBits = 8
            SerialPort2.Parity = Parity.None
            SerialPort2.StopBits = StopBits.One
            SerialPort2.Handshake = Handshake.None
            SerialPort2.Encoding = System.Text.Encoding.Default 'very
important!
            SerialPort2.ReadTimeout = 10000
```

```

        SerialPort2.Open()
        BtnConnect1.Text = "Dis-connect"
        Timer4.Enabled = True
        Timer_LBL1.Text = "Timer: ON"
    Else
        MsgBox("Select a COM port first")
    End If
Else
    SerialPort2.Close()
    BtnConnect1.Text = "Connect"
    Timer4.Enabled = False
    Timer_LBL1.Text = "Timer: OFF"
End If

```

End Sub

```

Private Sub Timer4_Tick(sender As Object, e As EventArgs)
Handles Timer4.Tick

```

```

    receivedData = ReceiveSerialData1()
    RichTextBox2.Text &= receivedData
    RichTextBox2.SelectionStart = RichTextBox2.TextLength
    RichTextBox2.ScrollToCaret()

```

End Sub

```

Function ReceiveSerialData1() As String

```

```

    Dim Incoming As String
    Try
        Incoming = SerialPort2.ReadExisting()
        If Incoming Is Nothing Then
            Return "nothing" & vbCrLf
        Else
            Return Incoming
        End If
    Catch ex As TimeoutException
        Return "Error: Serial Port read timed out."
    End Try

```

End Function

```

Private Sub BtnClear1_Click(sender As Object, e As EventArgs)
Handles BtnClear1.Click
    RichTextBox2.Text = ""
End Sub

```

```

Private Sub CmbPort3_SelectedIndexChanged(sender As Object, e
As EventArgs) Handles CmbPort3.SelectedIndexChanged
    If (CmbPort3.SelectedItem <> "") Then
        comPORT = CmbPort3.SelectedItem
    End If
End Sub

```

```

Private Sub BtnConnect2_Click(sender As Object, e As EventArgs)
Handles BtnConnect2.Click
    If (BtnConnect2.Text = "Connect") Then
        If (comPORT <> "") Then
            SerialPort3.Close()
            SerialPort3.PortName = comPORT
            SerialPort3.BaudRate = 9600
            SerialPort3.DataBits = 8
            SerialPort3.Parity = Parity.None
            SerialPort3.StopBits = StopBits.One
            SerialPort3.Handshake = Handshake.None
            SerialPort3.Encoding = System.Text.Encoding.Default 'very
important!
            SerialPort3.ReadTimeout = 10000

            SerialPort3.Open()
            BtnConnect2.Text = "Dis-connect"
            Timer5.Enabled = True
            Timer_LBL2.Text = "Timer: ON"
        Else
            MsgBox("Select a COM port first")
        End If
    Else
        SerialPort3.Close()
        BtnConnect2.Text = "Connect"
        Timer5.Enabled = False
        Timer_LBL2.Text = "Timer: OFF"
    End Sub

```

End If

End Sub

Private Sub Timer5\_Tick(sender As Object, e As EventArgs)  
Handles Timer5.Tick

    receivedData2 = ReceiveSerialData2()  
    RichTextBox3.Text &= receivedData2  
    RichTextBox3.SelectionStart = RichTextBox3.TextLength  
    RichTextBox3.ScrollToCaret()

End Sub

Function ReceiveSerialData2() As String

    Dim Incoming As String

    Try

        Incoming = SerialPort3.ReadExisting()

        If Incoming Is Nothing Then

            Return "nothing" & vbCrLf

        Else

            Return Incoming

        End If

    Catch ex As TimeoutException

        Return "Error: Serial Port read timed out."

    End Try

End Function

Private Sub BtnClear3\_Click(sender As Object, e As EventArgs)  
Handles BtnClear3.Click

    RichTextBox3.Text = ""

End Sub

Private Sub SaveFileDialog1\_FileOk(sender As Object, e As  
System.ComponentModel.CancelEventArgs) Handles  
SaveFileDialog1.FileOk

Me.PictureBox1.Image.Save(IO.Path.Combine(My.Computer.FileSystem  
em.SpecialDirectories.Desktop, ".JPEG"))

End Sub

```
Private Sub btnSend_Click(sender As Object, e As EventArgs)
Handles btnSend.Click
    SerialPort1.Write(txtTransmit.Text)
End Sub
```

```
Private Sub RichTextBox1_TextChanged(sender As Object, e As
EventArgs) Handles RichTextBox1.TextChanged
```

```
End Sub
```

```
Sub trial()
```

```
Dim parts() = Split(receivedData, vbLf)
```

```
For i = 0 To 5
```

```
    If parts(i).Contains("Yaw") = True Then
```

```
        yaw = parts(i)
```

```
        Dim valYaw() = Split(yaw, ":")
```

```
        Try
```

```
            Label40.Text = valYaw(1)
```

```
        Catch ex As Exception
```

```
        End Try
```

```
    ElseIf parts(i).Contains("Pitch") = True Then
```

```
        pitch = parts(i)
```

```
        Dim valPitch() = Split(pitch, ":")
```

```
        Try
```

```
            Label42.Text = valPitch(1)
```

```
        Catch ex As Exception
```

```
        End Try
```

```
ElseIf parts(i).Contains("Roll") = True Then
    roll = parts(i)
    Dim valRoll() = Split(roll, ":")

    Try
        Label41.Text = valRoll(1)
    Catch ex As Exception

    End Try
ElseIf parts(i).Contains("error_yaw") = True Then
    roll = parts(i)
    Dim valerror_yaw() = Split(error_yaw, ":")

    Try
        Label43.Text = valerror_yaw(1)
    Catch ex As Exception

    End Try

End If
Next

End Sub

End Class
```

## LAMPIRAN B

### Lampiran Kalibrasi Kompas

```
#include <Wire.h>
```

```
#include <HMC5883L.h>
```

```
HMC5883L compass;
```

```
int minX = 0;
```

```
int maxX = 0;
```

```
int minY = 0;
```

```
int maxY = 0;
```

```
int offX = 0;
```

```
int offY = 0;
```

```
void setup()
```

```
{
```

```
  Serial.begin(9600);
```

```
  // Initialize Initialize HMC5883L
```

```
  while (!compass.begin())
```

```
    );
```

```
  // Set measurement range
```

```
  compass.setRange(HMC5883L_RANGE_1_3GA);
```

```
  // Set measurement mode
```

```
  compass.setMeasurementMode(HMC5883L_CONTINUOUS);
```

```
  // Set data rate
```

```
  compass.setDataRate(HMC5883L_DATARATE_30HZ);
```

```
  // Set number of samples averaged
```

```
  compass.setSamples(HMC5883L_SAMPLES_8);
```

```
void loop()
```

```
{
```

```

Vector mag = compass.readRaw();

// Determine Min / Max values
if (mag.XAxis < minX) minX = mag.XAxis;
if (mag.XAxis > maxX) maxX = mag.XAxis;
if (mag.YAxis < minY) minY = mag.YAxis;
if (mag.YAxis > maxY) maxY = mag.YAxis;

// Calculate offsets
offX = (maxX + minX)/2;
offY = (maxY + minY)/2;

Serial.print(mag.XAxis);
Serial.print(":");
Serial.print(mag.YAxis);
Serial.print(":");
Serial.print(minX);
Serial.print(":");
Serial.print(maxX);
Serial.print(":");
Serial.print(minY);
Serial.print(":");
Serial.print(maxY);
Serial.print(":");
Serial.print(offX);
Serial.print(":");
Serial.print(offY);
Serial.print("\n");
}

```

### **Lampiran Program Kompas**

```

#include <Wire.h>
#include <HMC5883L.h>

```

```

HMC5883L compass;

```

```

void setup()
{
  Serial.begin(9600);
}

```



```

// Initialize Initialize HMC5883L
Serial.println("Initialize HMC5883L");
while (!compass.begin())
{
  Serial.println("Could not find a valid HMC5883L sensor, check
wiring!");
  delay(500);
}

// Set measurement range
compass.setRange(HMC5883L_RANGE_1_3GA);

// Set measurement mode
compass.setMeasurementMode(HMC5883L_CONTINUOUS);

// Set data rate
compass.setDataRate(HMC5883L_DATARATE_30HZ);

// Set number of samples averaged
compass.setSamples(HMC5883L_SAMPLES_8);

// Set calibration offset. See HMC5883L_calibration.ino
compass.setOffset( -168, -223);
}

void loop()
{
  Vector norm = compass.readNormalize();

  // Calculate heading
  float heading = atan2(norm.YAxis, norm.XAxis);

  // Set declination angle on your location and fix heading
  // You can find your declination on: http://magnetic-declination.com/
  // (+) Positive or (-) for negative
  // For Bytom / Poland declination angle is 4'26E (positive)
  // Formula: (deg + (min / 60.0)) / (180 / M_PI);
  float declinationAngle = (0 + (50.0 / 60.0)) / (180 / M_PI);

```

```

heading += declinationAngle;

// Correct for heading < 0deg and heading > 360deg
if (heading < 0)
{
    heading += 2 * PI;
}

if (heading > 2 * PI)
{
    heading -= 2 * PI;
}

// Convert to degrees
float headingDegrees = heading * 180/M_PI;

// Output
Serial.print(" Heading = ");
Serial.print(heading);
Serial.print(" Degree = ");
Serial.print(headingDegrees);
Serial.println();

delay(100);
}

```

### **Lampiran Program NMEA Data GPS**

```

#include <SoftwareSerial.h>

// setup gps serial
int gpsTxPin = 8;
int gpsRxPin = 9;
SoftwareSerial gpsSerial(gpsTxPin, gpsRxPin);

void setup()
{
    Serial.begin(9600); //set monitor to 9600

    gpsSerial.begin(9600); //adjust for GPS unit

```

```

    Serial.println("Ready!");
}

void loop()
{
    while(gpsSerial.available())
    {
        char c = gpsSerial.read();
        Serial.print(c);

    }
}

```

### **Lampiran Program GPS**

```

#include <TinyGPS++.h>
#include <SoftwareSerial.h>

int RXPin = 8;
int TXPin = 9;

int GPSBaud = 9600;

TinyGPSPlus gps;

SoftwareSerial gpsSerial(RXPin, TXPin);

void setup()
{
    Serial.begin(9600);

    gpsSerial.begin(GPSBaud);
}

```

```

void loop()
{
    while (gpsSerial.available() > 0)
        if (gps.encode(gpsSerial.read()))
            displayInfo();

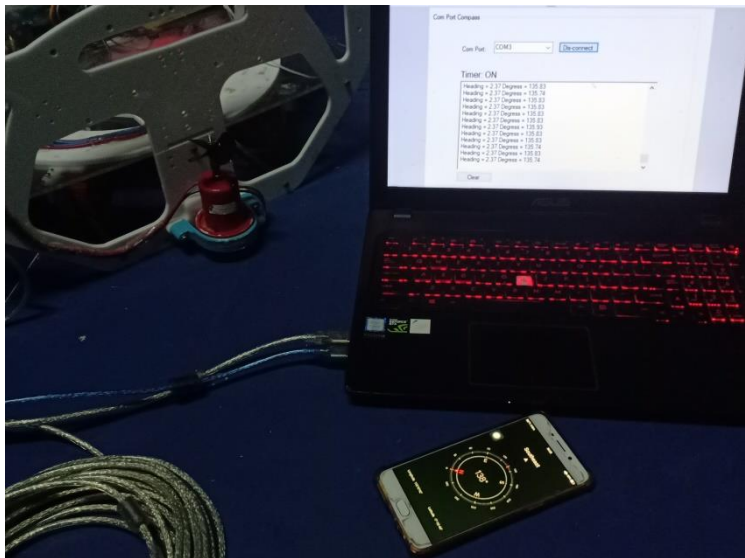
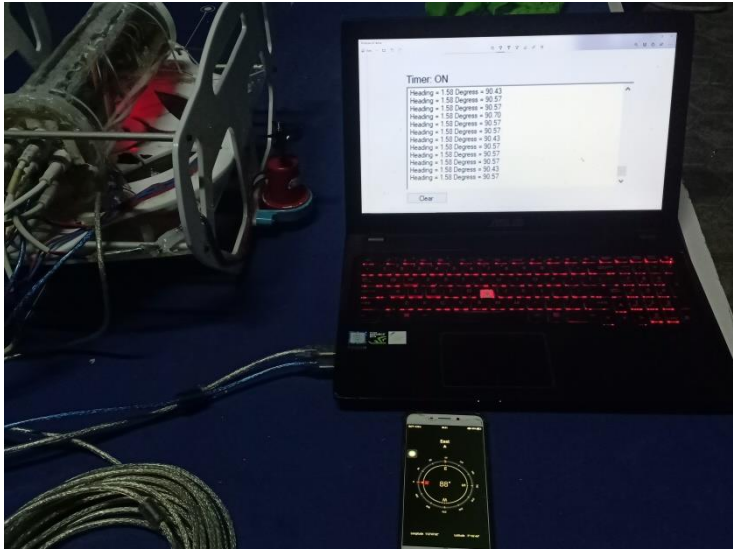
    if (millis() > 5000 && gps.charsProcessed() < 10)
    {
        Serial.println("No GPS detected");
        while(true);
    }
}

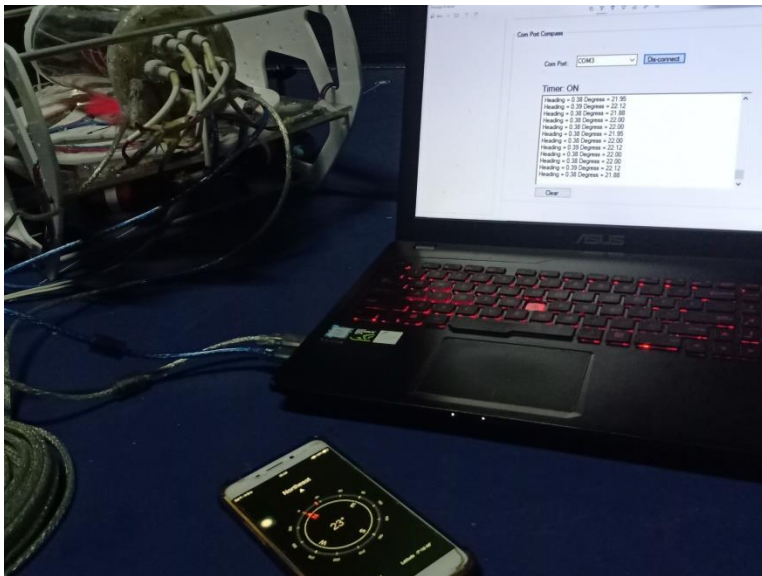
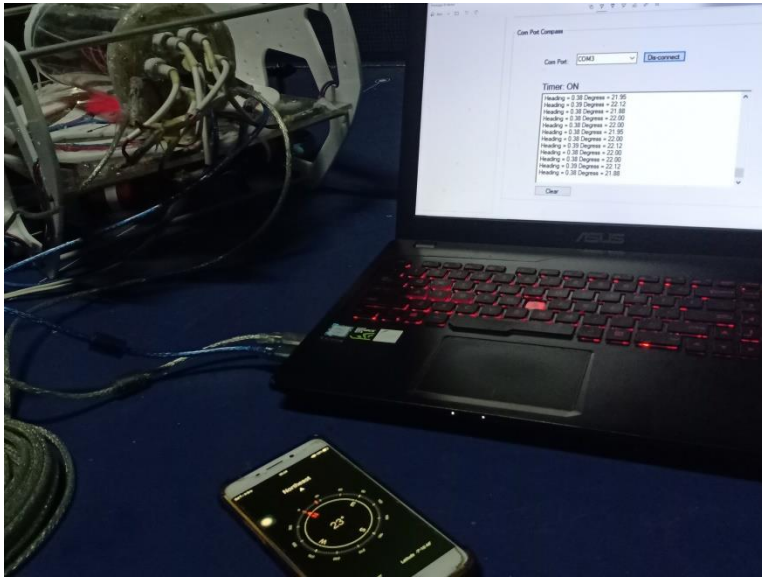
void displayInfo()
{
    if (gps.location.isValid())
    {
        Serial.print("Latitude: ");
        Serial.println(gps.location.lat(), 6);
        Serial.print("Longitude: ");
        Serial.println(gps.location.lng(), 6);
        Serial.print("Altitude: ");
        Serial.println(gps.altitude.meters());
    }
    else
    {
        Serial.println("Location: Not Available");
    }

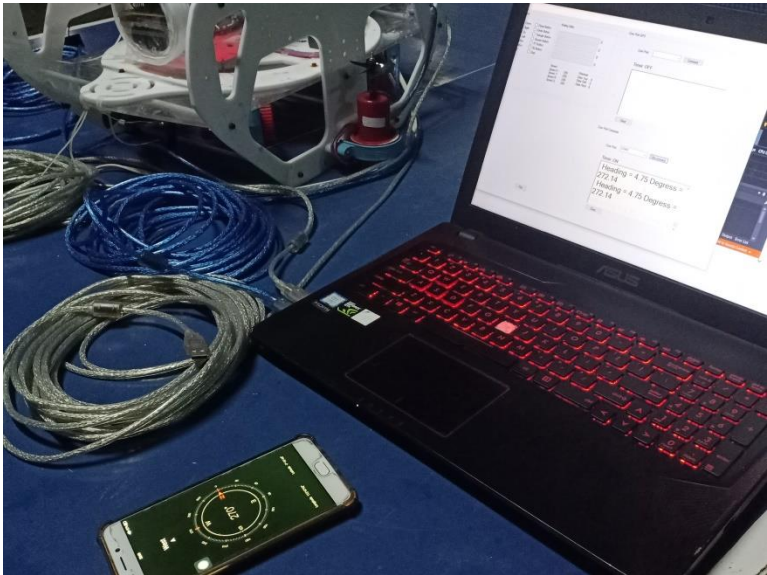
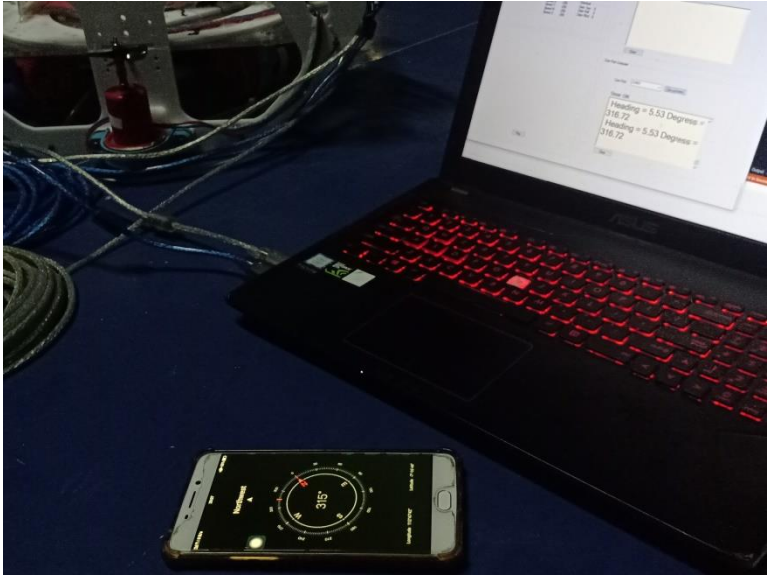
    Serial.println();
    Serial.println();
    delay(1000);
}

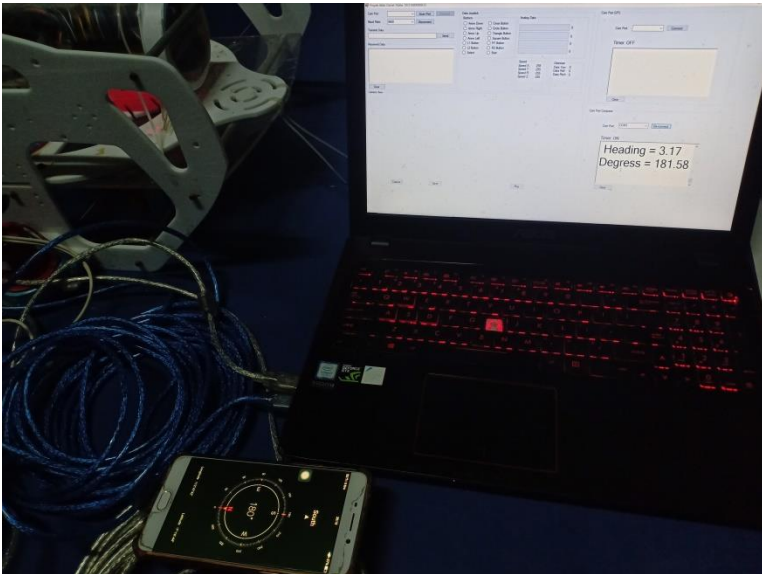
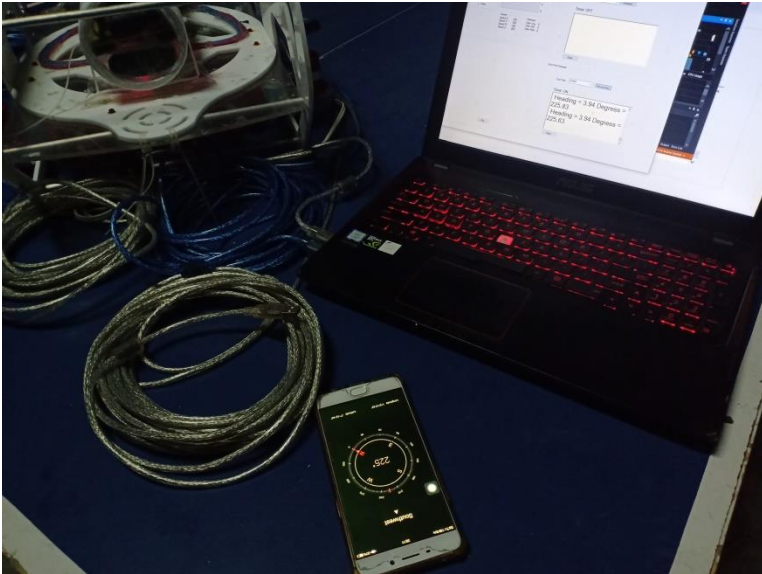
```

## LAMPIRAN C

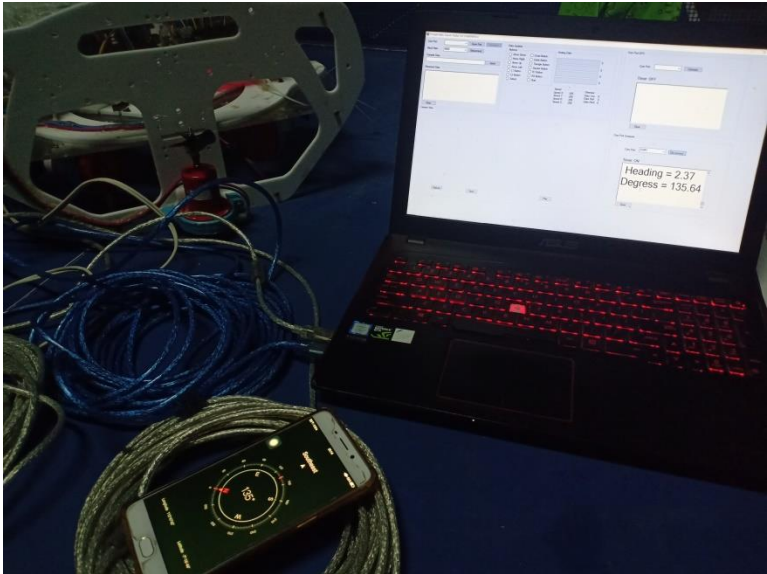














## BIODATA PENULIS



Nama : Dariel Thaha  
TTL : Darwin, 6 Maret 1998  
Jenis Kelamin : Laki-laki  
Agama : Islam  
Alamat rumah : Gg.Pustaka Jaya III no 1  
Jakarta Timur Pulo  
Gadung Rawamangun  
Nomor HP : 081358967147  
E-mail : darielthaha@gmail.com

### RIWAYAT PENDIDIKAN

- 2001-2003 : South Perth Kindergarden
- 2003-2009 : South Perth Primary School  
SDN Muhammadiyah Jakarta
- 2009-2013 : Kent Street SHS Perth
- 2014-2015 : SMA Diponegoro Jakarta
- 2016 – sekarang : Program Departemen Teknik Elektro  
Otomasi, Fakiultas Vokasi, ITS

### PENGALAMAN MAGANG

- Magang di PT.Berkah Industri Mesin Aangkat, Kota Surabaya, Tanjung Perak

### PENGALAMAN ORGANISASI

- UKM Resimen Mahasiswa 802 ITS Staf Urusan Khusus