



TUGAS AKHIR - TM 145502

APLIKASI SENSOR GPS (*GLOBAL POSITIONING SYSTEM*) PADA KAPAL PENGHITUNG UDANG TANPA AWAK UNTUK NAVIGASI PERGERAKAN KAPAL SECARA OTOMATIS

NAUFAL ARYUSMAL
NRP. 10211500000006

Dosen Pembimbing
Dr . Ir . Bambang Sampurno, MT.
NIP. 19650919 199003 1 003

DEPARTEMEN TEKNIK MESIN INDUSTRI
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018



TUGAS AKHIR – TM 145502

APLIKASI SENSOR GPS (*GLOBAL POSITIONING SYSTEM*) PADA KAPAL PENGHITUNG UDANG TANPA AWAK UNTUK NAVIGASI PERGERAKAN KAPAL SECARA OTOMATIS

NAUFAL ARYUSMAL
NRP. 10211500000006

Dosen Pembimbing
Dr.Ir. BAMBANG SAMPURNO, MT.

PROGRAM STUDI D3 TEKNIK MESIN
DEPARTEMEN TEKNIK MESIN INDUSTRI
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018



FINAL PROJECT – TM 145502

GPS SENSOR APPLICATIONS (*GLOBAL POSITIONING SYSTEM*) ON UNMANNED VESSEL SHRIMP COUNTERS FOR NAVIGATION OF SHIP MOVEMENTS AUTOMATICALLY

NAUFAL ARYUSMAL
NRP. 10211500000006

Counsellor Lecturer
Dr.Ir. BAMBANG SAMPURNO, MT.

STUDY PROGRAM D3 MECHANICAL ENGINEERING
DEPARTMENT OF MECHANICAL INDUSTRY ENGINEERING
Vocational Faculty
Sepuluh Nopember Institute of Technology
Surabaya 2018

LEMBAR PENGESAHAN

APLIKASI SENSOR GPS (*GLOBAL POSITIONING SYSTEM*) PADA KAPAL PENGHITUNG UDANG TANPA AWAK UNTUK NAVIGASI PERGERAKAN KAPAL SECARA OTOMATIS

TUGAS AKHIR

Diajukan Guna Memenuhi Salah Satu Syarat
Memperoleh Gelar Ahli Madya Pada
Bidang Studi Konversi Energi
Departemen Teknik Mesin Industri
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember

Oleh :

Naufal Aryusmal
NRP. 10211500000006

Disetujui Oleh Pembimbing Tugas Akhir:

Dosen Pembimbing



Dr. Ir. Bambang Sampurno, MT)

NIP. 19650919 199003 1 003

APLIKASI SENSOR GPS (GLOBAL POSITIONING SYSTEM) PADA KAPAL PENGHITUNG UDANG TANPA AWAK UNTUK NAVIGASI PERGERAKAN KAPAL SECARA OTOMATIS

Nama : Naufal Aryusmal

NRP : 10211500000006

Pembimbing : Dr.Ir. Bambang Sampurno, MT

ABSTRAK

Persoalan utama pada penelitian mengenai kapal pembawa kamera untuk menghitung jumlah udang yang telah dilakukan oleh Falah dkk, (2017) adalah pergerakan kapal yang masih manual atau menggunakan remote. Pergerakan kapal secara manual sebagai media pembawa kamera untuk menghitung jumlah udang akan mendapatkan hasil yang kurang optimal karena rute kapal bisa saja tidak terarah.

Berdasarkan pada kondisi tersebut maka melalui tugas akhir ini dirancang sistem kontrol Autonomous Surface Vechile (ASV) yaitu kapal tanpa awak yang bisa menentukan rute pergerakan kapal secara otomatis dengan bantuan sensor GPS. Komponen utama sistem ini adalah sensor GPS, Arduino, dan sensor kompas.

Hasil dari penelitian ini diperoleh kapal tanpa awak untuk pendeteksi jumlah udang dengan sistem kontrol navigasi waypoint dengan bantuan sensor GPS. Melalui bantuan sensor GPS maka kapal bergerak otomatis sesuai rute yang telah ditentukan. Kesalahan yang terjadi sebesar 0,2 meter pada saat jumlah titik waypoint delapan (8), sedangkan pada jumlah titik waypoint empat (4) kesalahan yang terjadi sebesar 1,5 meter.

Kata kunci : Autonomous, GPS, Navigasi, Waypoint.

GPS SENSOR APPLICATIONS (*GLOBAL POSITIONING SYSTEM*) ON UNMANNED VESSEL SHRIMP COUNTERS FOR NAVIGATION OF SHIP MOVEMENTS AUTOMATICALLY

Nama : Naufal Aryusmal

NRP : 10211500000006

Pembimbing : Dr.Ir. Bambang Sampurno, MT

ABSTRACT

The main problem in the study of camera carrying vessels to calculate the number of shrimp that has been done by Falah et al, (2017) is the movement of the ship is still manual or using the remote. Manually move the ship as a camera carrier medium to calculate the number of shrimp will get less than optimal results because the route of the ship may not be directional.

Based on these conditions then through this final task is designed Autonomous Surface Vechile (ASV) control system that is unmanned ship that can determine the route of ship movement automatically with the help of GPS sensor. The main components of this system are GPS sensors, Arduino, and compass sensors.

The results of this study obtained unmanned ship for the detection of shrimp counts with a waypoint navigation control system with the help of GPS sensors. Through the help of GPS sensors, the ship moves automatically according to a predetermined route. The error occurred at 0.2 meters at the number of eight waypoint points (8), while at four (4) waypoints the error occurred at 1.5 meters.

Keywords: *Autonomous, GPS, Navigation, Waypoint.*

KATA PENGANTAR

Alhamdulillah, segala puji dan syukur kami panjatkan kehadiran Allah SWT. Karena atas rahmat dan hidayah-Nya, tugas akhir yang berjudul **“Aplikasi Sensor GPS (*Global Positioning System*) Pada Kapal Penghitung Udang Tanpa Awak Untuk Navigasi Pergerakan Kapal Secara Otomatis”** ini dapat disusun dan diselesaikan dengan baik dan lancar.

Tugas Akhir ini merupakan salah satu persyaratan yang harus dipenuhi oleh setiap mahasiswa Departemen Teknik Mesin Industri, sesuai dengan kurikulum yang telah ditetapkan. Selain itu Tugas Akhir ini juga merupakan suatu bukti yang diberikan almamater dan masyarakat.

Banyak dorongan dan bantuan yang penulis dapatkan selama penyusunan Tugas Akhir ini sampai terselesaikannya laporan. Untuk itu penulis ingin menyampaikan ucapan terima kasih dan penghargaan sebesar-besarnya kepada :

1. Bapak Dr.Ir.Bambang Sampurno, MT., selaku dosen pembimbing tugas akhir, atas bimbingan, inspirasi, pengarahan, dan motivasi yang diberikan selama pengerjaan penelitian tugas akhir ini.
2. Mashuri, S.Si selaku kepala Asisten Laboratorium Mekatronika yang tiada hentinya memberikan ilmu guna menyelesaikan tugas akhir ini.
3. Bapak Ir.Suhariyanto, MT. Selaku koordinator Tugas Akhir Program Studi D3 Departemen Teknik Mesin Industri Fakultas Vokasi-ITS.
4. Bapak Dr. Ir. Heru Mirmanto, MT., selaku Kepala Departemen Teknik Mesin Industri Fakultas Vokasi-ITS.
5. Bapak dan ibu dosen tim penguji yang telah memberikan kritik dan saran dalam penyempurnaan dan pengembangan Tugas Akhir ini.
6. Seluruh dosen dan staf pengajar Departemen Teknik Mesin Industri Fakultas Vokasi-ITS, yang telah memberikan ilmunya dan membantu semua selama menimba ilmu di bangku kuliah.

7. Kedua orang tua serta kakak dan adik saya, yang tidak pernah putus untuk seluruh dukungan, doa, nasihat, motivasi, serta bantuan dalam berbagai hal.
8. Teman-teman laboratorium Mekatronika yang senantiasa membantu dan memberikan dukungan dalam mengerjakan tugas akhir.
9. Seluruh teman-teman angkatan saya D3MITS 2015 yang selalu memberi dukungan, motivasi serta bantuan dalam berbagai hal.
10. Teman-teman Unit Kegiatan Mahasiswa maupun Tim Robotika ITS yang telah membantu dalam mengerjakan tugas akhir.

Semoga segala keikhlasan dan kebaikan yang telah diberikan mendapatkan balasan yang terbaik dari Tuhan Yang Maha Esa, Amin.

Karena keterbatasan waktu dan kemampuan penulis, sebagai manusia biasa kami menyadari dalam penulisan ini masih terdapat beberapa kesalahan dan kekurangan. Oleh karena itu, kami mengharap kritik dan saran membangun sebagai masukan untuk penulis dan kesempurnaan Tugas Akhir ini. Semoga dengan penulisan tugas akhir ini dapat bermanfaat bagi semua pihak yang memerlukan, mahasiswa Departemen Teknik Mesin Industri FV-ITS pada khususnya.

Surabaya, 07 Juli 2018

Naufal Aryusmal
NRP.10211500000006

DAFTAR ISI

HALAMAN JUDUL	i
ABSTRAK	iv
ABSTRACT	v
KATA PENGANTAR	vi
DAFTAR ISI	viii
DAFTAR GAMBAR	x
DAFTAR TABEL	xi
BAB I	12
PENDAHULUAN	12
1.1 Latar Belakang	12
1.2 Perumusan Masalah.....	13
1.3 Tujuan Penelitian.....	13
1.4 Batasan Masalah.....	13
1.5 Sistematika Penulisan.....	14
BAB II	16
DASAR TEORI	16
2.1 Kemajuan Penelitian dan Pengembangan ASV	16
2.1.1 Hasil Desain Kapal sebelumnya	16
2.1.2 Hasil Rancang Bangun Kapal	17
2.2 Pengertian GPS (Global Positioning System)	20
2.2.1 Cara Kerja GPS.....	20
2.2.2 Fungsi dan Kegunaan GPS	20
2.3 Komponen pada Autonomous Surface Vechiles	21
2.3.1 Motor DC	22
2.3.2 Baterai Lithium Pollymer (LiPo)	23
2.3.3 Driver Motor (BTN7970).....	24
2.3.4 Bluetooth.....	25
2.3.5 Global Positioning System (GPS).....	26
2.3.6 Sensor Kompas	27

2.3.7 Arduino Mega	27
BAB III	30
METODOLOGI	30
3.1 Diagram alir (flow chart)	30
3.2 Penjelasan Diagram Alir Proses Pembuatan Alat	31
3.3 Mekanisme Kerja ASV	40
BAB IV	42
PERHITUNGAN DAN PEMBAHASAN	42
4.1 Pengujian Hardware	42
4.1.1 Pengujian GPS (Global Positioning System)	42
4.1.1.1 Proses Kalkulasi Waypoint (GPS)	43
4.1.2 Pengujian Telemetry	45
4.2 Pengujian Seluruh Sistem	48
4.2.1 Pengujian Pergerakan ASV mode Manual	48
4.2.2 Pengujian Pergerakan ASV mode Otomatis	48
4.2.2.1 Pengujian Pertama	48
4.2.2.2 Pengujian Kedua	50
4.2.2.3 Pengujian Ketiga	51
BAB V	56
KESIMPULAN	56
5.1 Kesimpulan	56
5.2 Saran	56
DAFTAR PUSTAKA	58
LAMPIRAN	59
BIODATA PENULIS	78
LAMPIRAN	

DAFTAR GAMBAR

Gambar 2.1	Desain Kapal bentuk 3D.....	6
Gambar 2.2	Desain Kerangka Body Kapal	7
Gambar 2.3	Pembuatan Kerangka Body Kapal.....	7
Gambar 2.4	Finishing Body Kapal.....	8
Gambar 2.5	Pengujian Kapal pada Kolam	8
Gambar 2.6	Struktur dan bagian-bagian motor DC.....	11
Gambar 2.7	Baterai LiPo.....	12
Gambar 2.8	Modul Driver Motor (BTN7970)	13
Gambar 2.9	Bluetooth HC-05	14
Gambar 2.10	GPS receiver Ublox M8N	15
Gambar 2.11	Modul Sensor Kompas GY-87	16
Gambar 2.12	Arduino Mega.....	17
Gambar 3.1	Diagram Alir Perencanaan Navigasi Kapal.....	19
Gambar 3.2	Blok Diagram Sistem Autonomous Surface Vechile.....	20
Gambar 3.3	Tampilan Software Arduino	23
Gambar 3.4	Pin Arduino yang digunakan mengakses motor .	24
Gambar 3.5	Program untuk putaran motor.....	24
Gambar 3.6	Program setting bluetooth.....	25
Gambar 3.7	Kode di dalam Arduino untuk Arduino bluetooth pada android	26
Gambar 3.8	Tampilan pada aplikasi Arduino Bluetooth Android.....	26
Gambar 3.9	Program logika PID mengatur gerak kapal	27
Gambar 4.1	ASV	31
Gambar 4.2	Data Longitude dan Latitude yang dihasilkan GPS.....	32
Gambar 4.3	Software 3DR Radio Config.....	34
Gambar 4.4	Telemetri yang sudah Terkoneksi.....	36
Gambar 4.5	Hasil Data yang dikirim melalui telemetry.....	36
Gambar 4.6	Titik Tujuan ASV pada pengujian pertama	38
Gambar 4.7	Titik Tujuan ASV pada pengujian kedua	39
Gambar 4.8	Titik Tujuan ASV pada pengujian ketiga	41

DAFTAR TABEL

Tabel 3.1	Penggunaan pin Arduino.....	22
Tabel 4.1	Hasil Data Pengujian Pertama Putaran 1	38
Tabel 4.2	Hasil Data Pengujian Pertama Putaran 2	38
Tabel 4.3	Hasil Data Pengujian Pertama Putaran 3	39
Tabel 4.4	Hasil Data Pengujian Kedua Putaran 1	40
Tabel 4.5	Hasil Data Pengujian Kedua Putaran 2	40
Tabel 4.6	Hasil Data Pengujian Kedua Putaran 3	40
Tabel 4.7	Hasil Data Pengujian Ketiga Putaran 1	41
Tabel 4.8	Hasil Data Pengujian Ketiga Putaran 2	41
Tabel 4.9	Hasil Data Pengujian Ketiga Putaran 3	42

BAB I

PENDAHULUAN

1.1 Latar Belakang

Persoalan utama pada peternak udang yaitu penentuan jumlah pakan udang. Kesalahan dalam menentukan jumlah pakan akan berakibat kerugian besar pada para petambak udang. Apabila pemberian pakan melebihi kebutuhan makan udang (*over feeding*) maka terjadi penambahan biaya produksi, disamping itu dengan berlebihnya jumlah pakan akan meningkatkan jumlah senyawa organik dan protein di dalam tambak. Kondisi ini memiliki potensi terjadi kematian pada udang. Sebaliknya, apabila pemberian pakan ternyata kurang dari kebutuhan, maka pertumbuhan udang akan lambat sehingga *Break Even Point* (BEP) sulit tercapai.

Berdasarkan kondisi tersebut, Falah, dkk (2017) telah melakukan penelitian dengan merancang kapal pembawa kamera untuk menghitung jumlah udang. Hasil penelitian tersebut berupa desain kapal dua lambung, dengan panjang kapal = 670 mm, panjang garis air kapal (*Length water line*) = 650 mm, lebar = 300 mm, tinggi = 180 mm, C_b (blok koefisien) = 0,55 dan V_s (kecepatan) = 29 knot. Pada kapal ini sudah dilakukan perhitungan besar daya listrik yang dibutuhkan dan telah ditentukan pemilihan komponen kapal yang digunakan. Tetapi, pada kapal mini ini masih bergerak secara manual atau menggunakan remote. Untuk memonitor udang secara real time diharuskan kapal bisa mengelilingi danau atau tambak udang secara terurut atau terarah. Pergerakan kapal secara manual diperoleh hasil kurang optimal karena tidak semua area dapat dimonitor.

Oleh karena itu, melalui tugas akhir ini dirancang *prototype* kapal dengan sistem *Autonomous Surface Vechiles* (ASV) berbasis pada sensor GPS. Rute kapal mengacu pada *waypoint* dari GPS akan digunakan untuk menghitung jumlah udang secara real time agar area dapat dideteksi lebih luas sekaligus menghitung jumlah udang lebih akurat.

Perancangan sistem kontrol kapal tanpa awak (*Autonomous Surface Vechiles*) ini menggunakan mikrokontroller *Arduino Mega* sebagai sistem utama pengendalian *Autonomous*. Dengan bantuan sensor GPS, akan diperoleh koordinat yang ditentukan. Data yang diperoleh GPS akan dikirim ke mikrokontroler diolah menjadi titik *waypoint* sebagai tujuan kapal. Selanjutnya mikrokontroler akan memberi perintah pada *driver motor* untuk mengendalikan gerak motor dari kapal tanpa awak menuju titik *waypoint* yang dituju. Sehingga *Autonomous Surface Vechiles* ini dapat berjalan sesuai dengan *waypoint* yang kita tentukan.

1.2 Perumusan Masalah

Permasalahan pada tugas akhir ini adalah sebagai berikut :

1. Bagaimana merancang sistem navigasi pada kapal tanpa awak untuk menentukan rute kapal yang diinginkan.
2. Bagaimana merancang sistem kontrol berbasis Arduino dengan GPS dan sensor kompas untuk mengontrol pergerakan kapal sesuai dengan koordinat atau lintasan yang telah ditentukan.

1.3 Tujuan Penelitian

Penelitian pada tugas akhir ini bertujuan sebagai berikut :

1. Memperoleh sistem navigasi menggunakan metode *waypoint* untuk menentukan jalur yang dilalui oleh *Autonomous Surface Vechiles*.
2. Memperoleh sistem kontrol berbasis Arduino dengan GPS untuk mengontrol pergerakan kapal sesuai dengan koordinat atau lintasan yang telah ditentukan.

1.4 Batasan Masalah

Batasan masalah dari tugas akhir ini adalah sebagai berikut:

1. Asumsi kecepatan angin 2 m/s-5 m/s
2. Desain, dimensi dan bentuk kapal mengacu pada hasil penelitian tugas akhir Falah, dkk (2017).
3. Jarak yang ditempuh kapal tidak lebih 200 meter.

1.5 Sistematika Penulisan

Laporan tugas akhir ini terdiri dari lima bab dengan sistematika penulisan sebagai berikut:

- Bab 1 : Pendahuluan
Bab ini meliputi latar belakang, perumusan masalah, tujuan penelitian, batasan masalah, dan sistematika penulisan.
- Bab 2 : Dasar Teori
Bab ini menjelaskan tentang dasar-dasar teori yang dibutuhkan dalam pengerjaan tugas akhir ini, yang meliputi teori dasar sensor kompas, GPS, dan Arduino Mega
- Bab 3: Metodologi
Bab ini membahas metodologi perencanaan pembuatan sistem kontrol pada alat, diagram alir pembuatan sistem navigasi kapal secara otomatis
- Bab 4 : Perhitungan dan Pembahasan
Bab ini menjelaskan tentang perhitungan program kalkulasi waypoint dan hasil percobaan alat
- Bab 5 : Penutup
Bab ini berisi kesimpulan dari proses pembuatan sistem navigasi kapal secara otomatis dan kekurangan-kekurangan pada kerja alat dari hasil analisa serta saran untuk pengembangan ke depan.

(Halaman ini sengaja dikosongkan)

BAB II

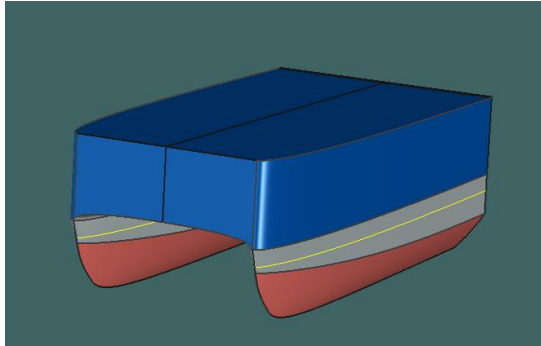
DASAR TEORI

2.1 Kemajuan Penelitian dan Pengembangan ASV

Banyak lembaga, universitas, bisnis dan militer telah mulai mengembangkan ASV untuk berbagai aplikasi selama dua dekade terakhir. Pengembangan ASV saat ini masih belum matang (Roberts & Sutton, 2006). Kebanyakan ASV yang ada terbatas pada platform eksperimental, terutama terdiri dari ASV yang berskala kecil dengan otonomi terbatas, daya tahan, muatan, dan output daya (Savitz et al., 2013), serta masih membutuhkan operasi jarak jauh (Breivik et al., 2008). ASV masih langka di pasar komersial, dan mayoritas ASV tingkat industri masih digunakan dalam aplikasi militer dan penelitian. Untuk ikhtisar dari prototipe kapal yang dikembangkan dan masalah desain dasar, pembaca juga dapat merujuk pada publikasi oleh Bertram (2008), Manley (2008) dan Motwani (2012).

2.1.1 Hasil Desain Kapal sebelumnya

Pada tugas akhir Falah,dkk (2017) kapal ini didesain menggunakan software yaitu *software maxsurf*. Sebelum mendesain kapal menggunakan *software maxsurf* maka ditentukan ukuran kasar kapal agar mudah dalam mencantumkan ukuran di *software maxsurf* . Sehingga setelah dicantumkan ukuran tersebut Di *software maxsurf* akan muncul perhitungan kecepatan kapal, beban maksimal yang diangkut kapal dan hidrostatisnya secara otomatis.

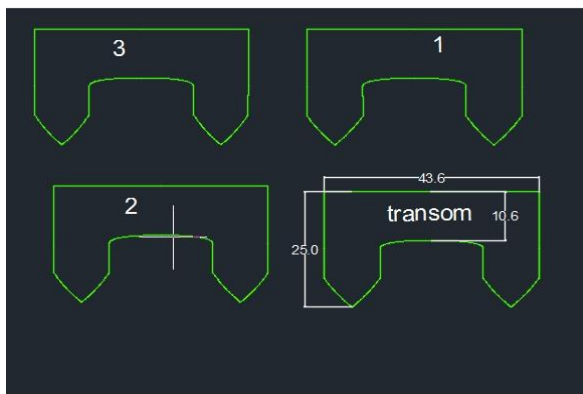


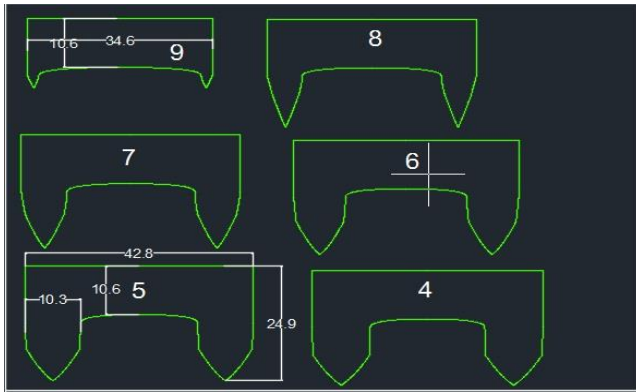
Gambar 2.1 Desain Kapal bentuk 3D (Falah dkk,2017)

2.1.2 Hasil Rancang Bangun Kapal

Pada tahapan rancang bangun kapal dilakukan proses pembuatan kapal yang diperoleh dari perancangan desain kapal menggunakan *software maxsurf*. Proses pembuatan kapal ada beberapa tahap yaitu sebagai berikut:

- Pertama mempersiapkan peralatan merancang kapal : kayu balsa dan silet.
- Hasil desain kapal menggunakan software maxsurf diprint menggunakan kertas A0. Seperti gambar di bawah ini yang harus di print. Untuk bagian yang lain berada pada lampiran.





Gambar 2.2 *Desain kerangka body kapal (Falah dkk,2017)*

- c. Setelah desain diprint kertas A0 , Desain tersebut dijadikan mall untuk merancang body kapal dan bagian yang lain menyesuaikan bentuk kerangka body kapal.



Gambar 2.3 *Pembuatan Kerangka Body Kapal (Falah dkk,2017)*

- d. Kemudian langkah selanjutnya finishing body, pendempulan dicat kasar. Seperti gambar dibawah ini



Gambar 2.4 *Finishing Body Kapal (Falah dkk,2017)*

- e. Setelah pembuatan kapal dan finishing selesai dilakukan pengujian kapal, dan hasil ada pada gambar 2.5



Gambar 2.5 *Pengujian kapal pada kolam(Falah dkk,2017)*

Pada hasil rancang bangun kapal yang di kemukakan oleh Falah, dkk (2017) sudah dilakukan pengujian di dalam kolam dan berhasil berjalan dengan cara manual dengan kecepatan maksimal 29 knot. Oleh karena itu, pada tugas akhir ini dilakukan pengembangan menggerakkan kapal secara otomatis dengan menggunakan sensor GPS guna mengikuti rute yang sudah ditentukan melalui koordinat yang didapat pada google maps.

2.2 Pengertian GPS (Global Positioning System)

Pengertian GPS adalah sistem navigasi yang menggunakan satelit yang didesain agar dapat menyediakan posisi secara instan, kecepatan dan informasi waktu di hampir semua tempat di muka bumi, setiap saat dan dalam kondisi cuaca apapun.

2.2.1 Cara Kerja GPS

Bagian yang paling penting dalam sistem navigasi GPS adalah beberapa satelit yang berada di orbit bumi atau yang sering kita sebut di ruang angkasa. Satelit GPS saat ini berjumlah 24 unit yang semuanya dapat memancarkan sinyal ke bumi yang lalu dapat ditangkap oleh alat penerima sinyal tersebut atau GPS Tracker. Selain satelit terdapat 2 sistem lain yang saling berhubungan, sehingga jadilah 3 bagian penting dalam sistem GPS. Ketiga bagian tersebut terdiri dari: GPS Control Segment (Bagian Kontrol), GPS Space Segment (bagian angkasa), dan GPS User Segment (bagian pengguna).

2.2.2 Fungsi dan Kegunaan GPS

Beberapa fungsi dan kegunaan GPS tersebut bisa dibagi kepada 5 poin, yaitu:

a. *GPS untuk Militer*

GPS dapat dimanfaatkan untuk mendukung sistem pertahanan militer. Lebih jauh dari itu bisa memantau pergerakan musuh saat terjadi peperangan, juga bisa menjadi penuntun arah jatuhnya bom sehingga bisa lebih tertarget.

b. *GPS untuk Navigasi*

Dalam kebutuhan berkendara sistem GPS pun sangat membantu, dengan adanya GPS Tracker terpasang pada kendaraan

maka akan membuat perjalanan semakin nyaman karena arah dan tujuan jalan bisa diketahui setelah GPS mengirim posisi kendaraan kita yang diterjemahkan ke dalam bentuk peta digital.

c. *GPS untuk Sistem Informasi Geografis*

GPS sering juga digunakan untuk keperluan sistem informasi geografis, seperti untuk pembuatan peta, mengukur jarak perbatasan, atau bisa dijadikan sebagai referensi pengukuran suatu wilayah.

d. *GPS untuk Sistem Pelacakan Kendaraan*

Fungsi ini hampir sama dengan navigasi, jika dalam navigasi menggunakan perangkat penerima sinyal GPS berikut penampil titik koordinatnya dalam satu perangkat, sedangkan untuk kebutuhan sistem pelacakan adalah alat penampil dan penerima sinyal berbeda lokasi.

Contohnya kita bisa mengetahui lokasi kendaraan yang hilang dengan melihat titik kordinat yang dihasilkan dari alat yang terpasang dalam kendaraan tersebut, untuk melihatnya bisa melalui media smartphone atau alat khusus lainnya.

e. *GPS untuk Pemantau Gempa*

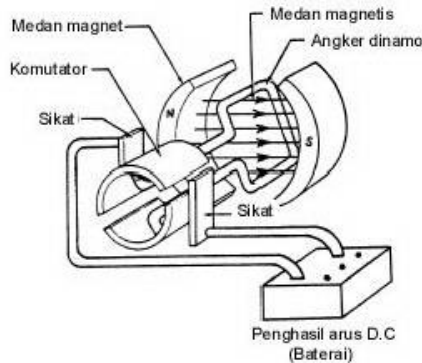
Saat ini teknologi GPS yang terus ditingkatkan menghasilkan tingkat ketelitian dan keakuratan yang sangat tinggi sehingga GPS dapat dimanfaatkan untuk memantau pergerakan tanah di bumi. Dengan hal itu maka para pakar Geologi dapat memperkirakan kemungkinan terjadinya gempa di suatu wilayah.

2.3 Komponen pada Autonomous Surface Vechiles

Untuk ASV pada tugas akhir ini dilengkapi dengan motor DC sebagai penggerak yang diatur kecepatannya menggunakan driver motor dengan diprogram melalui mikrokontroller yang bernama Arduino Mega. Untuk navigasi pada ASV ini menggunakan *Global Positioning System* (GPS) untuk menentukan lokasi ASV bergerak serta dibantu dengan sensor kompas agar kapal bisa berjalan dengan lurus sesuai lokasi yang ditentukan. Dan bluetooth hanya

digunakan untuk menggerakkan kapal secara manual dengan menggunakan android jika diperlukan.

2.3.1 Motor DC



Gambar 2.6 Struktur dan bagian-bagian motor DC

Motor DC atau Motor Arus Searah merupakan motor listrik yang memerlukan sumber arus listrik searah pada kumparan medan untuk mengubah energi listrik menjadi energi gerak. Motor DC memiliki dua bagian, yaitu stator dan rotor. Stator merupakan bagian yang diam, terdiri dari rangka, *brush* dan kumparan medan, sedangkan rotor adalah bagian dari motor yang berputar, terdiri dari kumparan jangkar dan komutator.

Pada gambar 2.6 merupakan gambar motor DC sederhana yang memiliki kumparan dengan satu buah lilitan. Prinsip kerja dari motor DC yaitu ketika motor diberikan sumber listrik DC, maka arus akan melewati *brush* yang merupakan bagian dari stator, kemudian *brush* akan mengalirkan arus menuju komutator, komutator berfungsi untuk mengubah arah arus listrik pada kumparan, konduktor atau angker dinamo yang terhubung oleh komutator tersebut akan menghasilkan medan magnet, arah dari medan magnet ditentukan oleh arah arus yang melewati konduktor

tersebut menggunakan aturan tangan kanan. Dengan adanya dua buah kutub magnet permanen yang ada pada bagian stator motor DC, akan menyebabkan kumparan berputar, arah putar pada kumparan menggunakan aturan tangan kiri *Flamming*.

2.3.2 Baterai Lithium Pollymer (LiPo)



Gambar 2.7 Baterai LiPo

Baterai Lithium Pollymer merupakan sumber energi utama pada *Autonomous Surface Vehicle*. Baterai yang digunakan merupakan baterai LiPo 3 sel, karena masing-masing sel dalam baterai LiPo memiliki tegangan sebesar 3,7 Volt, sehingga bila menggunakan LiPo 3 sel berarti baterai LiPo tersebut memiliki ukuran 11,1 Volt. Kapasitas baterai LiPo dinyatakan dalam mAh, dimana semakin besar nilai mAh semakin besar tenaga yang mampu disimpan pada baterai tersebut.

Seperti pada gambar, baterai LiPo menggunakan elektrolit polimer kering yang berbentuk seperti lapisan plastik film tipis. Lapisan film tersebut disusun berlapis-lapis diantara *anoda* dan *katoda* yang mengakibatkan pertukaran ion, dengan metode tersebut baterai LiPo dapat dibuat dalam berbagai bentuk desain dan ukuran.

2.3.3 Driver Motor (BTN7970)

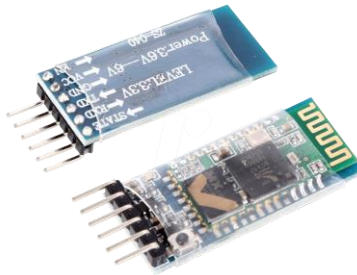


Gambar 2.8 Modul driver Motor (BTN7970)

Gambar 2.8 merupakan contoh modul *driver* motor yang digunakan pada tugas akhir ini. *Driver* motor merupakan komponen penting yang berfungsi untuk mengatur arah (*Counter Clock Wise* atau *Counter Wise*) dan kecepatan putar pada motor. Pengaturan arah putar dan kecepatan motor yang dihasilkan berasal dari perintah program yang ada pada mikrokontroler Arduino Mega.

Prinsip kerja dari *motor driver* yaitu memberikan sumber tegangan 12 volt sesuai dengan tegangan pada motor yang digunakan, kemudian dari tegangan 12 volt digunakan untuk sumber tegangan motor dan untuk regulator. Regulator berfungsi untuk menurunkan tegangan dari 12 volt menjadi 5 volt, yang berfungsi sebagai sumber tegangan pada IC yang digunakan pada *driver* motor BTN7970. Sedangkan untuk menentukan arah putar motor dapat dikontrol dengan menggunakan Arduino Mega yang terhubung dengan pin IN1 dan IN2 pada motor driver, sedangkan untuk mengatur kecepatan putar pada motor yaitu menggunakan PWM (*Pulse Width Modulation*) yang terhubung dengan pin VIN.

2.3.4 Bluetooth



Gambar 2.9 *Bluetooth HC-05*

Bluetooth merupakan sebuah teknologi komunikasi tanpa kabel (wireless) yang berfungsi untuk mengirim atau menerima data dalam jarak jangkauan yang terbatas. Pada umumnya bluetooth bekerja pada pita frekuensi 2.4 GHz.

Gambar diatas merupakan sebuah modul *bluetooth* HC-05 yang banyak beredar di pasaran, modul *bluetooth* ini memiliki jarak jangkauan kurang lebih 10 meter, sangat mudah digunakan dan memiliki ukuran dimensi yang kecil dan tipis sehingga sering digunakan pada berbagai peralatan. Pada ASV sendiri *bluetooth* HC-05 digunakan untuk mengontrol kapal secara manual apabila terjadi gangguan atau masalah pada saat kapal bergerak secara otomatis.

2.3.5 Global Positioning System (GPS)



Gambar 2.10 GPS receiver Ublox M8N

GPS merupakan navigasi yang memberikan informasi berupa posisi suatu objek di bumi, berbasis sistem satelit. Untuk dapat menentukan suatu posisi objek di bumi, GPS *receiver* membutuhkan 3 – 4 sinyal satelit yang ditangkap agar dapat membaca posisi dengan baik. Jika GPS *receiver* mampu menangkap tiga sinyal satelit, maka GPS *receiver* akan menerima data berupa *longitude* dan *latitude*. Sedangkan jika GPS *receiver* mampu menangkap empat atau lebih sinyal dari satelit maka GPS *receiver* mampu menerima data berupa *longitude*, *latitude*.

Daya yang didapat oleh GPS *receiver* berupa *longitude* dan *latitude*, selanjutnya dapat diketahui letak posisi, jarak, arah, dan kecepatan suatu objek pada posisi asal ke posisi tujuan dengan menggunakan perhitungan, dan bantuan sensor berupa kompas.

Modul GPS *receiver* Ublox M8N seperti gambar, sudah dilengkapi kompas didalam rangkaian elektroniknya. Menurut datasheet, GPS *receiver* tersebut memiliki error posisi kurang lebih 2 meter. Sehingga apabila menentukan posisi tujuan harus diberikan toleransi error lokasi kurang lebih 2 meter. GPS *receiver*

tersebut memerlukan sumber tegangan sebesar 5 volt, yang berfungsi untuk menyuplai GPS dan kompas.

2.3.6 Sensor Kompas

Kompas merupakan salah satu alat yang digunakan pada sistem navigasi, kompas bertujuan untuk menentukan suatu arah berdasarkan posisi kutub bumi. Pada umumnya kompas memberikan ukuran derajat lingkaran searah jarum jam, utara 0°, timur 90°, selatan 180°, dan barat 270°. Dalam sistem navigasi ASV menggunakan kompas GY-87 seperti gambar 2.11, *axis* kompas yang digunakan hanya *yaw*, dikarenakan *yaw* berfungsi sebagai arah pada kapal atau yang biasa disebut *heading*. Berdasarkan datasheet, kompas tersebut memiliki 3 *axis* yang bisa digunakan untuk *pitch*, *yaw*, dan *roll*.

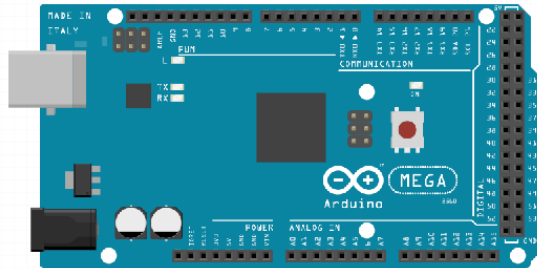


Gambar 2.11 Modul Sensor Kompas GY-87

2.3.7 Arduino Mega

Arduino merupakan sebuah mikrokontroler *single-board* yang bersifat *open-source*, diturunkan dari *Wiring platform*, dirancang sedemikian hingga agar mudah digunakan dalam berbagai bidang. Arduino memiliki prosesor Atmel AVR dan memiliki bahasa pemrograman yang mudah diterjemahkan, bahasa pemrograman Arduino yang memiliki kemiripan *syntax* dengan bahasa pemrograman C. Arduino menggunakan keluarga mikrokontroler ATmega yang dirilis oleh Atmel sebagai basis.

Pada kondisi lain terdapat clone arduino dengan menggunakan mikrokontroler berbeda dan tetap kompatibel dengan arduino pada level *hardware*.



Gambar 2.12 *Arduino Mega*

Arduino yang digunakan pada ASV adalah Arduino Mega seperti pada gambar 2.12, karena arduino jenis ini memiliki spesifikasi yang dibutuhkan pada ASV agar dapat bekerja secara optimal. ASV membutuhkan banyak PIN *Output / Input*, serta membutuhkan tiga pin komunikasi serial yang digunakan untuk *bluetooth*, GPS, dan telemetri.

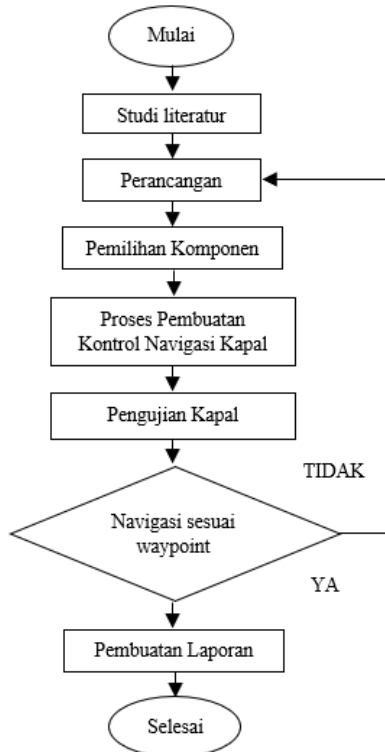
(Halaman ini sengaja dikosongkan)

BAB III METODOLOGI

Pada bab ini akan dibahas metodologi tugas akhir yang meliputi tahapan-tahapan dalam menyelesaikan tugas akhir. Perencanaan secara keseluruhan proses pembuatan dan penyelesaian sistem kontrol navigasi kapal berbasis GPS ini digambarkan dalam diagram alir atau *flow chart* di bawah ini.

3.1 Diagram alir (flow chart)

Alur proses pembuatan kontrol otomatis navigasi kapal berbasis GPS.



Gambar 3.1 Diagram alir perencanaan navigasi kapal

3.2 Penjelasan Diagram Alir Proses Pembuatan Alat

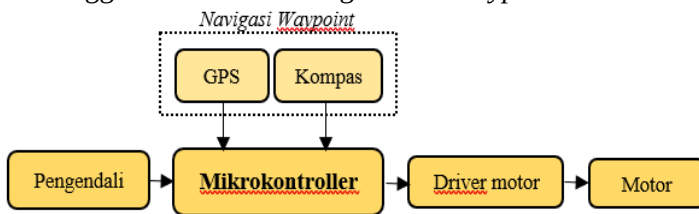
Dalam perencanaan membuat sistem kontrol navigasi kapal mini tanpa awak otomatis dengan menggunakan sensor GPS ini menggunakan metode penelitian, meliputi :

1. Studi Literatur

Pada tahap ini melakukan suatu pengamatan tentang masalah-masalah yang terjadi dan studi literatur yang berisi tentang serangkaian kegiatan dan pengkajian sumber-sumber yang relevan dan terpercaya dalam pengumpulan materi serta menjadi acuan dalam penulisan tugas akhir ini. Literatur yang kami pakai kami titik beratkan pada penerapan kapal tanpa awak berbasis sensor GPS atau yang disebut dengan *Autonomous Surface Vechiles*. Kami menggunakan banyak literatur berbeda seperti buku ilmiah, jurnal, dan beberapa artikel dari internet agar dapat menghasilkan informasi yang lengkap, terarah dan terpercaya dalam menulis serta memberikan variasi dalam pengembangan alat ini.

2. Perancangan

Pada tahap ini perancangan sistem kapal tanpa awak otomatis berbasis GPS (*Autonomous Surface Vechiles*) diuraikan melalui diagram blok sistem. Diagram blok sistem ini menjelaskan tentang keseluruhan kerja sistem yang terdapat pada ASV. Sistem kerja pada ASV yaitu melakukan perjalanan kapal secara otomatis dengan menggunakan sistem navigasi GPS *waypoint*.



Gambar 3.2 Blok diagram sistem *Autonomous Surface Vehicle*

Pada gambar 3.1 menunjukkan blok diagram sistem keseluruhan pada ASV. ASV bergerak berdasarkan perintah yang

dikirimkan oleh pengendali berupa HP Android, kemudian pengendali mengirimkan perintah ke *bluetooth* HC-05 yang dimiliki oleh ASV, perintah tersebut berupa pilihan mode yaitu mode otomatis atau mode manual, dan kemudian kapal akan bergerak sesuai perintah tersebut.

3. Pemilihan Komponen

Komponen pada kapal tanpa awak otomatis atau *Autonomous Surface Vehicle* meliputi :

- Modul GPS dan Kompas

GPS *receiver* dan kompas digunakan sebagai sistem navigasi pada ASV, untuk membaca posisi dan menentukan arah posisi kapal.

- *Driver* Motor dan Motor

Driver motor digunakan untuk mengubah sinyal PWM menjadi besaran tegangan yang digunakan untuk sumber pada motor ASV.

- Arduino Mega

Arduino Mega yang digunakan menghitung dan mengolah semua data yang terdapat pada sistem ASV.

- Regulator 7805

Regulator ini digunakan untuk pembagi tegangan dari baterai LiPo menjadi 12V dan 5V.

- Bluetooth

Bluetooth digunakan untuk komunikasi terhadap android yang memberikan perintah ASV bergerak otomatis atau manual.

4. Proses pembuatan sistem kontrol navigasi ASV

Pada tahap ini dilakukan pembuatan sistem kontrol navigasi ASV dengan menggunakan *software* arduino pada laptop atau PC. Sebelum memprogram arduino, ada beberapa tahap yaitu sebagai berikut :

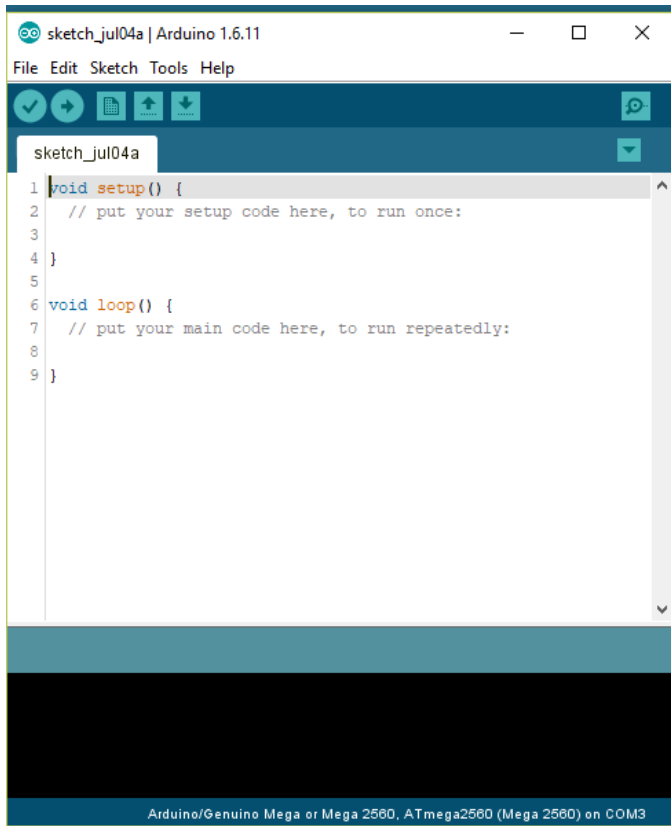
- a. Pertama mempersiapkan komponen-komponen yang diperlukan pada ASV yaitu : GPS, sensor kompas, driver motor, Arduino mega, regulator 7805 dan kabel jumper.
- b. Setelah mengumpulkan komponen yang diperlukan, komponen-komponen tersebut dirangkai dan dihubungkan ke mikrokontroler Arduino mega sesuai pada tabel 3.1.

Tabel 3.1 Penggunaan pin Arduino

No.	Pin Arduino	Komponen	Keterangan
1	SDA	Sensor Kompas	Mengontrol posisi ASV saat berjalan berupa sudut
2	SCL	Sensor Kompas	
2	PWM 2	PWM Motor	<i>Output</i> PWM motor kiri
3	PWM 3	PWM Motor	<i>Output</i> PWM motor kanan
4	PWM 4	<i>Driver</i> motor	IN1 motor kiri
5	PWM 5	<i>Driver</i> motor	IN2 motor kiri
6	PWM 6	<i>Driver</i> motor	IN1 motor kanan
7	PWM 7	<i>Driver</i> motor	IN2 motor kiri
8	16 / TX 2	RX <i>Bluetooth</i>	Mengontrol ASV agar dapat bergerak otomatis / manual
9	17 / RX 2	TX <i>Bluetooth</i>	
10	18 / TX 1	RX GPS	membaca lokasi ASV berupa <i>longitude</i> dan <i>latitude</i>
11	19 / RX 1	TX GPS	

Untuk pembagi tegangan yang dihasilkan oleh regulator yang sudah disupply baterai LiPo 3 cell, tegangan 12V digunakan untuk mensupply driver motor sedangkan untuk tegangan 5V digunakan untuk mensupply Arduino mega, GPS, sensor kompas, dan bluetooth.

c. Setelah semua komponen dirangkai, buka *software* arduino pada laptop atau PC dan akan muncul seperti gambar 3.3.



Gambar 3.3 *Tampilan Software Arduino*

- d. Kemudian masukkan program yang digunakan untuk memberikan perintah ke semua komponen yang dipakai melalui mikrokontroler Arduino mega.
- e. Pada program kapal ini dibagi menjadi 2 yaitu gerak kapal secara manual dan otomatis.
- f. Untuk memprogram kapal secara manual, kita hanya memprogram pergerakan motor pada kapal dengan mengakses putaran motor untuk maju, mundur dan belok. Program untuk

mengakses motor secara manual ada pada lampiran 10. Pada lampiran 10 dijelaskan untuk mengakses motor di arduino kita harus mengaktifkan pin mana yang digunakan motor di arduino seperti pada gambar di bawah ini.

```
1 int R1 = 6; // Kanan Maju // ke in 1
2 int R2 = 7; // Kanan Mundur // ke in 2
3 int pwmr = 3;
4 int L1 = 5; // Kiri Maju // ke in 3
5 int L2 = 4; // Kiri Mundur // ke in 4
6 int pwml = 2;
7
```

Gambar 3.4 Pin arduino yang digunakan mengakses motor

- g. Setelah itu kita membuat program untuk putaran motor dengan memberi logika HIGH dan LOW pada motor untuk mengatur putaran searah jarum jam atau berlawanan arah jarum jam seperti pada gambar.

```
15 void L_CCW(void){
16     digitalWrite(L1, LOW);
17     digitalWrite(L2, HIGH);
18 }
19
20 void L_CW(void){
21     digitalWrite(L1, HIGH);
22     digitalWrite(L2, LOW);
23 }
24
25 void R_CCW(void){
26     digitalWrite(R1, LOW);
27     digitalWrite(R2, HIGH);
28 }
29
30 void R_CW(void){
31     digitalWrite(R1, HIGH);
32     digitalWrite(R2, LOW);
33 }
```

Gambar 3.5 program untuk putaran motor

- h. Setelah itu kita membuat program gerak kapal dengan menyesuaikan baling-baling kapal. Semisal untuk gerak maju, motor bagian kiri berputar berlawanan arah jarum jam sedangkan motor bagian kanan berputar searah jarum jam dan seterusnya.
- i. Selanjutnya kita perlu mengakses bluetooth agar dapat menggerakkan kapal secara manual dari jauh melalui Android, program untuk mengakses bluetooth ada pada lampiran 9. Cara mengaktifkannya yaitu dengan membaca serial yang terhubung antara bluetooth ke arduino dimana bluetooth terhubung ke pin TX/RX 2 pada arduino sehingga program mengaktifkannya seperti pada gambar.

```
void setupbluetooth(void)
{
    Serial2.begin(9600);
}
int cek =0;
void loopbluetooth(void)
{
    if( Serial2.available() >0 )
    {
        cek++;
        val = Serial2.read();
        Serial2.println(val); // Serial.println(val);

        if(val == '8') status_otomatis = !status_otomatis;
    }
}
```

Gambar 3.6 *Program setting Bluetooth*

- j. Setelah diaktifkan kita dapat membuat mode pilihan untuk pergerakan kapal melalui android yang terhubung bluetooth. seperti pada gambar.

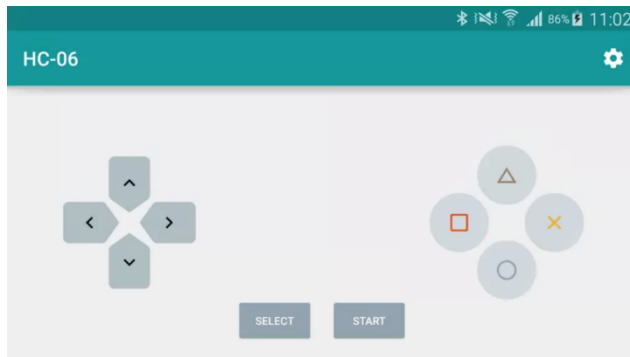
```

1 int Lamp1; //maju
2 int Lamp2; //mundur
3 int Lamp3; //kanan
4 int Lamp4; //kiri
5 int Lamp5; //m. kanan
6 int Lamp6; //m. kiri
7 int Lamp7; //stop
8 int Lamp8; //start otomatis
9

```

Gambar 3.7 Kode di dalam arduino untuk arduino bluetooth pada android

- k. Dengan begitu kita bisa mensetting sesuai gambar diatas melalui aplikasi Arduino bluetooth pada android dan sesuaikan terhadap tombol yang tersedia.



Gambar 3.8 Tampilan pada aplikasi Arduino bluetooth android

- l. Setelah setting selesai maka bisa menggerakkan kapal secara manual melalui aplikasi Arduino bluetooth pada android.
- m. Selanjutnya untuk memprogram kapal secara otomatis, hal yang dilakukan pertama kali yaitu mengakses GPS, sensor kompas

- dan kontrol PID motor kapal agar menyesuaikan pergerakan kapal ke tujuan *waypoint* yang dituju GPS.
- n. Untuk proses perhitungan penentuan *waypoint* yang dituju oleh GPS terdapat pada sub bab 4.1.1.1 di atas dan dimasukkan ke dalam program.
 - o. Selanjutnya mengakses sensor kompas seperti pada lampiran 2 sampai 3. Pada program untuk sensor kompas digunakan untuk menstabilkan pada saat kapal bergerak dan memposisikan sudut kapal agar bisa lurus dalam berjalan sesuai arah mata angin.
 - p. Setelah itu membuat program kontrol PID yang terdapat pada lampiran 6 sampai 7. Kontrol PID pada program ini digunakan agar gerak motor kapal sesuai dengan *waypoint* yang dituju oleh GPS. Untuk program otomatis kita bisa menginputkan letak koordinat *latitude* dan *longitude* sesuai yang kita inginkan sehingga kapal akan bergerak otomatis sesuai koordinat yang diberikan.

```

211 //algoritma control motor
212 if (PID_sudut<-10){
213     digitalWrite(R1, LOW);
214     digitalWrite(R2, HIGH); //LOW
215     digitalWrite(L1, HIGH);
216     digitalWrite(L2, LOW);
217     Serial.print("Kanan");
218     Serial.print('\t');
219 }
220 else if (PID_sudut>=10 && PID_sudut <=10){
221     digitalWrite(R1, HIGH);
222     digitalWrite(R2, LOW);
223     digitalWrite(L1, HIGH);
224     digitalWrite(L2, LOW);
225     Serial.print("Maju");
226     Serial.print('\t');
227 }
228 else if (PID_sudut>10){
229     digitalWrite(R1, HIGH);
230     digitalWrite(R2, LOW);
231     digitalWrite(L1, LOW);
232     digitalWrite(L2, HIGH); //LOW
233     Serial.print("Kiri");
234     Serial.print('\t');
235 }
236 analogWrite(pwml, abs(kiri) );
237 analogWrite(pwmr, abs(kanan) );

```

Gambar 3.9 Program logika PID untuk mengatur gerak kapal

- q. Setelah membuat program manual dan otomatis, untuk menggabungkan kedua program tersebut yaitu dengan cara membuat switch program dimana kita bisa mengganti mode kapal bergerak secara otomatis ataupun manual. Program tersebut berada pada lampiran 14 sampai 15.
- r. Selanjutnya program yang sudah dibuat di *software* Arduino diupload ke mikrokontroller Arduino.
- s. Setelah program berhasil terupload ke mikrokontroller Arduino Mega, kapal otomatis tanpa awak (*Autonomous Surfaces Vechiles*) siap dijalankan di danau.
- t. Cara mengganti mode otomatis dan manual melalui android yang terkoneksi oleh bluetooth menggunakan aplikasi Arduino bluetooth seperti pada gambar 3.8 dengan cara tekan tombol start.

5. Pengujian Kapal

Setelah pembuatan sistem *Autonomous Surfaces Vechiles* selesai, dilakukan pengujian kapal tersebut dan dicatat hasil pengujiannya, apakah kapal berjalan dengan baik atau tidak. Pengujiannya dilakukan dengan cara perbandingan titik koordinat *waypoint* yang telah ditentukan dengan titik koordinat ketika GPS pada ASV berjalan di danau. Perbandingan tersebut dikirim melalui telemetry yang sudah terhubung antara kapal dengan laptop. Data yang dikirim berupa titik *longitude* dan *lattitude*. Jika hasil perbandingan *longitude* dan *lattitude* menyerupai atau sama maka pengujian kapal dinyatakan berhasil. Sedangkan jika hasilnya sebaliknya, perlu dilakukan perancangan ulang dan pemrograman ulang agar mendapatkan hasil yang sesuai.

6. Pembuatan Laporan

Pembuatan laporan ini merupakan proses akhir dalam pengerjaan tugas akhir ini. Dalam pembuatan laporan dilampirkan mengenai proses perencanaan sampai pada hasil yang dicapai dalam tugas akhir.

3.3 Mekanisme Kerja ASV

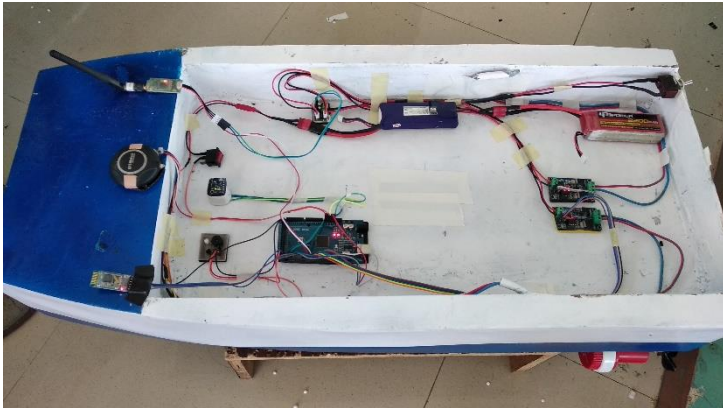
- a. Untuk menyalakan ASV ini pertama sambungkan baterai LiPo 3 cell ke kabel yang sudah terhubung oleh regulator.
- b. Nyalakan saklar pada ASV.
- c. Buka aplikasi Android Bluetooth Controller pada android.
- d. Nyalakan bluetooth pada android dan hubungkan bluetooth pada kapal.
- e. Setelah itu jalankan kapal secara manual agar kapal menjauhi tepi danau.
- f. Jika ingin melakukan secara otomatis, tekan tombol start pada aplikasi Android Bluetooth Controller maka kapal akan berjalan secara otomatis sesuai *waypoint* yang sudah ditentukan di dalam program.

(Halaman ini sengaja dikosongkan)

BAB IV

PERHITUNGAN DAN PEMBAHASAN

Pada bab ini membahas tentang pengujian dan analisis sistem dari *Autonomous Surface Vehicle* yang telah dirancang pada bab sebelumnya. Gambar 4.1 merupakan ASV yang digunakan dalam penelitian tugas akhir ini. Pengujian dan analisis bab ini dibagi menjadi berbagai tahap bagian, dengan membahas sub bagian dari perancangan sistem secara keseluruhan disertai tabel dan gambar yang mendukung pengujian dan analisis sistem.



Gambar 4.1 ASV

4.1 Pengujian Hardware

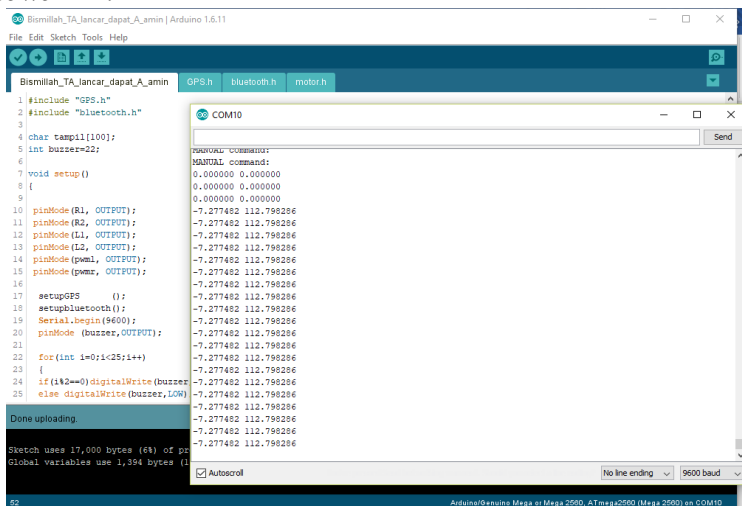
Pada pengujian *hardware* ini dilakukan dengan cara menguji masing-masing komponen (perangkat keras) yang digunakan pada *Autonomous Surface Vehicle*, yaitu GPS dan Telemetry.

4.1.1 Pengujian GPS (Global Positioning System)

Autonomous Surface Vehicle dapat bergerak secara otomatis dengan bantuan navigasi *waypoint* pada GPS. Navigasi *waypoint* tersebut bekerja berdasarkan titik-titik lokasi yang dipilih,

sehingga ASV akan dapat bergerak menuju titik ke titik yang telah ditentukan secara otomatis.

Pengujian sensor GPS ini dilakukan di tempat terbuka supaya GPS bisa menangkap satelit dengan optimal. Pengujian ini dilakukan dengan cara GPS yang terkoneksi dengan arduino yang sudah tersambung dengan laptop dibawa ke tempat terbuka. Setelah itu tampilkan serial monitor pada *software* arduino apakah program yang sudah di upload dapat menampilkan data longitude dan latitude. Jika berhasil akan menampilkan data seperti gambar dibawah ini.



Gambar 4.2 Data longitude dan latitude yang dihasilkan GPS

4.1.1.1 Proses Kalkulasi Waypoint (GPS)

Untuk dapat memerintahkan pergerakan otomatis pada ASV, maka diperlukan perhitungan pada data yang terbaca oleh GPS. Terdapat beberapa proses, yaitu :

1. Membaca koordinat posisi riil longitude dan latitude ASV dalam satuan derajat, kemudian dikonversi ke dalam radian, (1 derajat adalah 0,0174533 radian).

$$longitude\ radian = longitude \times 0.01745329252 \quad (4.1)$$

$$latitude\ radian = latitude \times 0.01745329252 \quad (4.2)$$

Dimana ; $Longitude_rad = a$

$Latitude_rad = b$

2. Seperti poin 1, mengubah koordinat posisi target berupa longitude dan latitude dari satuan derajat kedalam radian.

$$long.target\ radian = longitude \times 0.01745329252 \quad (4.3)$$

$$latt.target\ radian = latitude \times 0.01745329252 \quad (4.4)$$

Dimana ; $Long.target\ rad = aa$

$Latt.target\ rad = bb$

3. Setelah koordinat posisi riil dan posisi tujuan dikonversi dalam bentuk radian, maka selanjutnya dapat diketahui *bearing* terhadap posisi tujuan, dengan menggunakan perhitungan matematis pada persamaan 4.5 – 4.7.

$$bearing = arc.tan2(y, x) \times 57.295779513 \quad (4.5)$$

$$y = \sin(aa - a) \times \cos(bb) \quad (4.6)$$

$$x = \cos(b) \times \sin(bb) - \sin(b) \times \cos(bb) \times \cos(aa - a) \quad (4.7)$$

4. Menghitung jarak posisi riil dengan posisi tujuan dengan menggunakan persamaan 4.8 – 4.11.

$$jarak = 6371 \times 1000(m) \times \left(2 \times atan2(\sqrt{z}, \sqrt{1-z}) \right) \quad (4.8)$$

$$Z = \sin(\Delta latt/2) \times \sin(\Delta latt/2) + \sin(\Delta long/2) \times \sin(\Delta long/2) \times \cos(b) \times \cos(bb) \quad (4.9)$$

$$\Delta long.rad. = aa - a \quad (4.10)$$

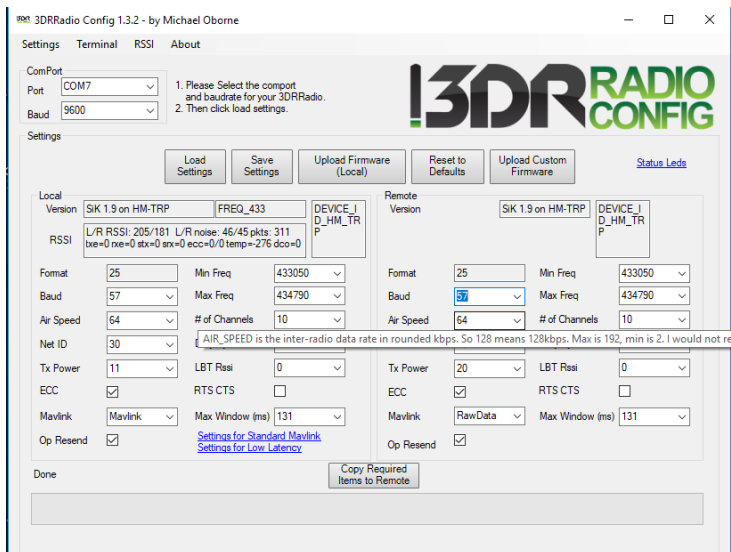
$$\Delta latt.rad. = bb - b \quad (4.11)$$

5. Setelah mendapatkan jarak dan *bearing* atau sudut kapal terhadap posisi tujuan, maka ASV dapat mengatur posisi tujuan, arah hadap kapal, dan *error bearing* secara otomatis dengan bantuan kontrol PID.

4.1.2 Pengujian Telemetry

Pada ASV ini, telemetry berfungsi sebagai media transmisi, untuk mengirimkan data yang terdapat pada ASV ke laptop secara real time. Data yang dikirimkan berupa lokasi GPS ASV (longitude, latitude).

Telemetry dibagi menjadi 2 buah rangkaian yang berbeda yaitu *Ground module* dan *Air module*, dan harus saling terhubung satu sama lain, untuk dapat mengirimkan data dari ASV ke Laptop. Untuk dapat menghubungkan *Ground module* dengan *Air module*, maka harus dipastikan bahwa telemetry *Air module* telah terhubung dengan port serial 0 pada arduino mega, serta suplai tegangan 5 volt. Cara pairing antara ground module dan air module yaitu dengan menggunakan *software 3DR config* seperti pada gambar 4.3.



Gambar 4.3 Software 3DR Radio Config

Langkah-langkah mensetting telemetry yaitu :

1. Memberikan suplai tegangan pada modul telemetry dan menghubungkan *air module* dengan port serial 0 arduino mega, sedangkan *ground module* dihubungkan dengan port usb pada laptop.
2. Menjalankan *software* 3DR Radio Config, dan menyesuaikan port sesuai dengan port yang digunakan *ground module*.
3. Mengatur *baudrate* yang digunakan pada telemetry, dengan mengisi opsi pada kolom “baud”.
4. Setelah mendapatkan *baudrate* yang sesuai, langkah terakhir adalah *pairing*, atau menyambungkan anatara *ground module* dan *air module* dengan cara menentukan Net ID yang digunakan pada kedua modul tersebut, Net ID pada *ground module* dan *air module* harus sama.

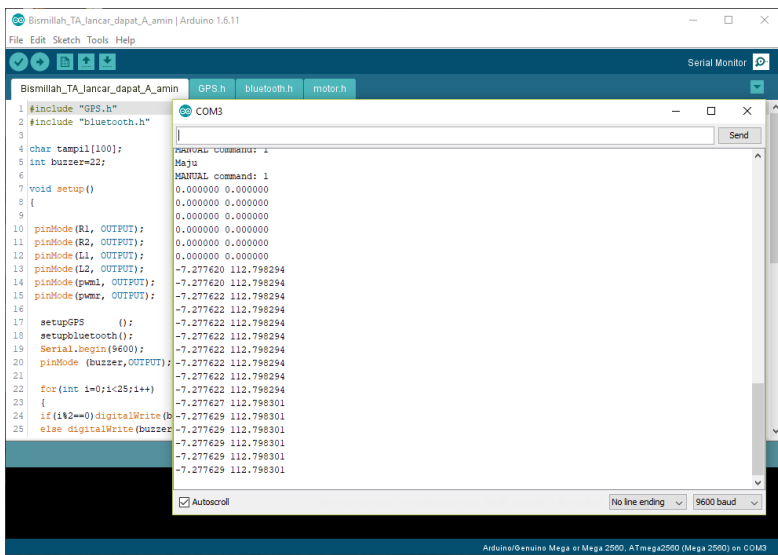
Setelah mensetting seperti langkah-langkah diatas, maka selanjutnya yaitu mengecek koneksi telemetry. Untuk *Ground Module* disambungkan ke laptop dan *Air Module* disambungkan ke Arduino. Jika sudah terkoneksi maka lampu pada telemetry tidak berkedip seperti pada gambar dibawah ini.





Gambar 4.4 Telemetri yang sudah terkoneksi

Setelah terkoneksi, maka kita bisa mengecek data yang diperoleh dari GPS yang sudah diolah oleh arduino berupa longitude dan lattitude yang dikirim melalui telemetri dengan membuka serial monitor pada *software arduino* di laptop seperti pada gambar.



Gambar 4.5 Hasil data yang dikirim melalui telemetry

Dengan hasil tersebut maka, ASV bisa digunakan untuk pengambilan data berupa koordinat longitude dan latitude yang dikirim melalui telemetri.

4.2 Pengujian Seluruh Sistem

Pada pengujian keseluruhan sistem hasil yang didapatkan yaitu kapal mampu bergerak secara otomatis maupun manual.

4.2.1 Pengujian Pergerakan ASV mode Manual

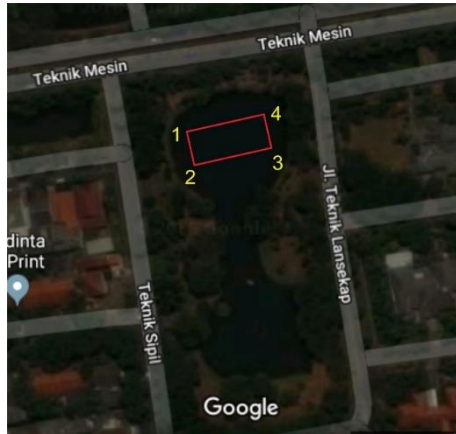
Pada pengujian dengan pergerakan manual, kapal mampu bergerak berdasarkan perintah yang dikirim oleh *bluetooth*. *Bluetooth* yang terpasang pada ASV merupakan bluetooth receiver atau penerima data. Sedangkan yang mengirim data adalah bluetooth yang berasal dari Android. Kontrol menggunakan bluetooth tersebut memiliki keterbatasan jarak, dikarenakan jarak jangkauan pada bluetooth kurang lebih 10 meter.

4.2.2 Pengujian Pergerakan ASV mode Otomatis

Pada pengujian pergerakan kapal mode otomatis, kapal mampu bergerak sesuai dengan titik tuju waypoint yang telah ditentukan, kapal bergerak menyusuri perairan dengan navigasi waypoint.

4.2.2.1 Pengujian Pertama

Pengujian ASV pertama dilakukan di perairan danau 8 kampus ITS, untuk mengetahui sistem secara keseluruhan. Pada pengujian ini menggunakan 4 titik lokasi tujuan pada sistem navigasi *waypoint* GPS. Lokasi-lokasi tersebut dapat dilihat pada gambar 4.6.



Gambar 4.6 Titik Tujuan ASV pada pengujian pertama

Dari pengujian ini, telemetri mampu mengirimkan data dari ASV ke laptop dengan baik, data yang dikirimkan berupa koordinat lokasi ASV longitude dan lattitude. Data yang diterima laptop tersebut seperti pada tabel 4.1.

Tabel 4.1 Hasil data pengujian pertama putaran 1

	Target Waypoint		GPS ASV	
Titik Tujuan	Lattitude	Longitude	Lattitude	Longitude
Waypoint 1	-7.286309	112.795892	-7.286352	112.795875
Waypoint 2	-7.286457	112.795906	-7.286453	112.795974
Waypoint 3	-7.286435	112.796124	-7.286435	112.796034
Waypoint 4	-7.286270	112.796089	-7.286274	112.796089

Tabel 4.2 Hasil data pengujian pertama putaran 2

	Target Waypoint		GPS ASV	
Titik Tujuan	Lattitude	Longitude	Lattitude	Longitude
Waypoint 1	-7.286309	112.795892	-7.286315	112.795883
Waypoint 2	-7.286457	112.795906	-7.286458	112.795906
Waypoint 3	-7.286435	112.796124	-7.286437	112.796081

Waypoint 4	-7.286270	112.796089	-7.286273	112.796127
------------	-----------	------------	-----------	------------

Tabel 4.3 Hasil data pengujian pertama putaran 3

	Target Waypoint		GPS ASV	
Titik Tujuan	Lattitude	Longitude	Lattitude	Longitude
Waypoint 1	-7.286309	112.795892	-7.286335	112.795890
Waypoint 2	-7.286457	112.795906	-7.286455	112.795982
Waypoint 3	-7.286435	112.796124	-7.286438	112.796096
Waypoint 4	-7.286270	112.796089	-7.286270	112.796119

4.2.2.2 Pengujian Kedua

Pengujian ASV kedua dilakukan juga di perairan danau 8 kampus ITS. Pada pengujian ini menggunakan 4 titik lokasi tujuan dengan pola berbeda pada sistem navigasi *waypoint* GPS. Lokasi-lokasi tersebut dapat dilihat pada gambar 4.7.



Gambar 4.7 Titik Tujuan ASV pada pengujian kedua

Tabel 4.4 Hasil Data Pengujian Kedua Putaran 1

TitikTujuan	Target Waypoint		GPS ASV	
	Lattitude	Longitude	Lattitude	Longitude
Waypoint 1	-7.286259	112.796062	-7.286258	112.796127
Waypoint 2	-7.286406	112.796131	-7.286399	112.796134
Waypoint 3	-7.286456	112.795887	-7.286435	112.795899
Waypoint 4	-7.286298	112.795921	-7.286297	112.795959

Tabel 4.5 Hasil Data Pengujian Kedua Putaran 2

TitikTujuan	Target Waypoint		GPS ASV	
	Lattitude	Longitude	Lattitude	Longitude
Waypoint 1	-7.286259	112.796062	-7.286288	112.796090
Waypoint 2	-7.286406	112.796131	-7.286490	112.796150
Waypoint 3	-7.286456	112.795887	-7.286450	112.795900
Waypoint 4	-7.286298	112.795921	-7.286298	112.795933

Tabel 4.6 Hasil Data Pengujian Kedua Putaran 3

TitikTujuan	Target Waypoint		GPS ASV	
	Lattitude	Longitude	Lattitude	Longitude
Waypoint 1	-7.286259	112.796062	-7.286258	112.796080
Waypoint 2	-7.286406	112.796131	-7.286410	112.796139
Waypoint 3	-7.286456	112.795887	-7.286458	112.795901
Waypoint 4	-7.286298	112.795921	-7.286298	112.795930

4.2.2.3 Pengujian Ketiga

Pengujian ASV ketiga dilakukan juga di perairan danau 8 kampus ITS. Pada pengujian ini menggunakan 4 titik lokasi tujuan dengan pola berbeda dari pengujian pertama dan kedua pada sistem navigasi *waypoint* GPS. Lokasi-lokasi tersebut dapat dilihat pada gambar 4.8.



Gambar 4.8 Titik Tujuan ASV pada pengujian ketiga

Tabel 4.7 Hasil Data Pengujian Ketiga Putaran 1

Titik Tujuan	Target Waypoint		GPS ASV	
	Lattitude	Longitude	Lattitude	Longitude
Waypoint 1	-7.286277	112.795959	-7.286273	112.795997
Waypoint 2	-7.286374	112.795850	-7.286378	112.795860
Waypoint 3	-7.286369	112.795973	-7.286406	112.795867
Waypoint 4	-7.286480	112.795925	-7.286475	112.795940
Waypoint 5	-7.286525	112.796010	-7.286528	112.795967
Waypoint 6	-7.286439	112.796112	-7.286450	112.796122
Waypoint 7	-7.286338	112.796133	-7.286347	112.796150
Waypoint 8	-7.286233	112.796046	-7.286232	112.796046

Tabel 4.8 Hasil Data Pengujian Ketiga Putaran 2

Titik Tujuan	Target Waypoint		GPS ASV	
	Lattitude	Longitude	Lattitude	Longitude

Waypoint 1	-7.286277	112.795959	-7.286287	112.795969
Waypoint 2	-7.286374	112.795850	-7.286379	112.795874
Waypoint 3	-7.286369	112.795973	-7.286380	112.795976
Waypoint 4	-7.286480	112.795925	-7.286477	112.795934
Waypoint 5	-7.286525	112.796010	-7.286529	112.796015
Waypoint 6	-7.286439	112.796112	-7.286443	112.796117
Waypoint 7	-7.286338	112.796133	-7.286343	112.796148
Waypoint 8	-7.286233	112.796046	-7.286235	112.796049

Tabel 4.9 Hasil Data Pengujian Ketiga Putaran 3

	Target Waypoint		GPS ASV	
TitikTujuan	Latitude	Longitude	Latitude	Longitude
Waypoint 1	-7.286277	112.795959	-7.286273	112.795961
Waypoint 2	-7.286374	112.795850	-7.286378	112.795860
Waypoint 3	-7.286369	112.795973	-7.286403	112.795901
Waypoint 4	-7.286480	112.795925	-7.286491	112.795931
Waypoint 5	-7.286525	112.796010	-7.286533	112.795990
Waypoint 6	-7.286439	112.796112	-7.286436	112.796123
Waypoint 7	-7.286338	112.796133	-7.286335	112.796140
Waypoint 8	-7.286233	112.796046	-7.286232	112.796046

Berdasarkan pada tabel 4.1 sampai 4.9 merupakan hasil data perbandingan dari target waypoint yang sudah diprogram di dalam arduino dengan GPS pada ASV ketika sudah berjalan di danau. ASV dapat bergerak secara otomatis sesuai dengan titik *waypoint* yang ditujukan. Akan tetapi ada perbedaan diantara target waypoint dengan hasil data dari GPS yang didapat saat kapal sedang berjalan. Perbedaan tersebut paling kecil yaitu 0.000001^0 dan yang terbesar yaitu 0.000090^0 . Perbedaan tersebut dikarenakan spesifikasi yang dimiliki modul GPS dan bisa saja GPS sudah terlalu panas akibat menangkap sinyal dari satelit.

Maka untuk mengetahui seberapa panjang tingkat *error* GPS dalam satuan meter. Kita bisa konversikan dari satuan Degree Decimal seperti pada tabel diatas menjadi satuan *Degree Minute Second (DMS)* dan kemudian dijadikan satuan meter dengan menggunakan cara dibawah ini.

Contoh pertama perbandingan perbedaan data terkecil yaitu pada data pengujian ketiga putaran 3 pada titik *waypoint 8 latitude* :

- Cara konversi DD ke DMS Titik -7.286233 :
Angka di depan koma diubah menjadi 7^0
Angka di belakang koma dikali 60 yaitu $0,286233 \times 60 = 17,17398$ bisa disebut $17'$.
Angka di belakang koma dari hasil diatas dikali 60 lagi yaitu $0,17398 \times 60 = 10,4388$ bisa disebut $10''$.
Dengan Hasil diatas maka DMS koordinat yaitu **$7^0 17' 10''$** .
- Cara konversi DD ke DMS Titik -7.286232 :
Angka di depan koma diubah menjadi 7^0
Angka di belakang koma dikali 60 yaitu $0,286232 \times 60 = 17,17392$ bisa disebut $17'$.
Angka di belakang koma dari hasil diatas dikali 60 lagi yaitu $0,17392 \times 60 = 10,4388$ bisa disebut $10''$.
Dengan Hasil diatas maka DMS koordinat yaitu **$7^0 17' 10''$** .
Maka dari hasil perbandingan diatas perbedaan koordinat $0,000001^0$ tidak terlalu berpengaruh terhadap kinerja ASV.

Contoh kedua perbandingan perbedaan data terbesar yaitu pada data pengujian pertama putaran 1 pada titik *waypoint 3 longitude* :

- Cara konversi DD ke DMS Titik 112.796124:
Angka di depan koma diubah menjadi 112^0
Angka di belakang koma dikali 60 yaitu $0,796124 \times 60 = 47,76744$ bisa disebut $47'$.
Angka di belakang koma dari hasil diatas dikali 60 lagi yaitu $0,76744 \times 60 = 46,0464$ bisa disebut $46''$.
Dengan Hasil diatas maka DMS koordinat yaitu **$112^0 47' 46''$** .
- Cara konversi DD ke DMS Titik 112.796034:
Angka di depan koma diubah menjadi 112^0
Angka di belakang koma dikali 60 yaitu $0,796034 \times 60 = 47,76204$ bisa disebut $47'$.

Angka di belakang koma dari hasil diatas dikali 60 lagi yaitu $0,76804 \times 60 = 46,0824$ bisa disebut 46”.

Dengan Hasil diatas maka DMS koordinat yaitu **112°47’46”**. Maka dari hasil perbandingan diatas perbedaan koordinat $0,000090^0$ tidak terlalu berpengaruh terhadap kinerja ASV. Dengan semua percobaan yang dilakukan, dapat disimpulkan bahwa ASV pada pengujian tugas akhir ini bisa dibilang akurat meskipun ada perbedaan koordinat $0,000001^0$ - $0,000090^0$. Perbedaan itu bisa saja disebabkan oleh spesifikasi GPS atau GPS sudah terlalu panas akibat menerima sinyal dari satelit.

BAB V

KESIMPULAN

5.1 Kesimpulan

Dari proses perencanaan dan pembahasan tugas akhir dengan judul Aplikasi Sensor GPS (Global Positioning System) pada Kapal Tanpa Awak untuk Navigasi Pergerakan Kapal Secara Otomatis, dapat disimpulkan bahwa :

1. Telah diperoleh sistem kontrol berbasis Arduino dengan GPS untuk mengontrol agar pergerakan kapal sesuai dengan koordinat atau lintasan yang telah ditentukan secara otomatis.
2. Telah diperoleh sistem navigasi menggunakan metode *waypoint* untuk menentukan jalur yang dilalui oleh *Autonomous Surface Vechiles* dengan bantuan *sensor GPS* dengan tingkat *error* 0,2 meter pada saat jumlah titik *waypoint* ada 8 titik sedangkan *error* 1,5 meter pada saat jumlah titik *waypoint* ada 4 titik.

5.2 Saran

Saran yang diperlukan untuk membuat sistem kontrol ASV ini adalah :

1. Untuk penelitian selanjutnya, disarankan untuk supply kapal yaitu baterai LiPo 3 Cell menggunakan baterai dengan kapasitas yang besar sehingga kapal bisa tahan lama saat mengelilingi danau.
2. Untuk penelitian selanjutnya, disarankan menggunakan sensor GPS yang lebih canggih lagi atau sensor pendukung lain supaya bisa meningkatkan keakuratan titik koordinat yang dituju serta perlu di upgrade pada sistem kontrol GPS agar akurasi pencarian titik menjadi lebih baik.

(Halaman ini sengaja dikosongkan)

DAFTAR PUSTAKA

- [1] Bertram, V. (2008). Unmanned surface vehicles –A survey. In *Proceedings of skib- steknisk selskab, Copenhagen, Denmark*.
- [2] Breivik, M. , Hovstein, V. E. , & Fossen, T. I. (2008). Straight-line target tracking for unmanned surface vehicles. *Modeling, Identification and Control*, 29 (4), 131–149 .
- [3] Falah, Adi Muhammad., M Shohibul Wafa, “*Perancangan Kapal Mini Sebagai Media Pembawa Kamera Untuk Mendeteksi Jumlah Udang Vaname di Kolam*”, Institut Teknologi Sepuluh Nopember, Surabaya, 2017.
- [4] <http://bidakara.ac.id/pengertian-gps-cara-kerja-gps-dan-fungsi-gps/> diakses pada tanggal 01 Juli 2018.
- [5] <https://raymoon760.wordpress.com/2013/06/19/konversi-derajat-bujur-dan-lintang/> diakses pada tanggal 7 Juli 2018.
- [6] Manley, J. E. (2008). Unmanned surface vehicles, 15 years of development. In *Pro- ceedings of IEEE international conference on OCEANS* (pp. 1–4) .
- [7] Motwani, A. (2012). *A survey of uninhabited surface vehicles. MIDAS technical report. MIDAS.SMSE.2012.TR.001 . MIDAS .*
- [8] Rizki, Fadlila, “*Autonomous Surface Vehicle sebagai Alat Pemantau Lingkungan*”, Institut Teknologi Sepuluh Nopember, Surabaya, 2017.
- [9] Roberts, G. N., & Sutton, R. (Eds.). (2006). *Advances in unmanned marine vehicles* . The Institution of Engineering and Technology .
- [10] Savitz, S., Blickstein, I., Buryk, P., Button, R. W., DeLuca, P., Dryden, J., et al. (2013). US Navy employment options for unmanned surface vehicles (USVs). Santa Monica, CA: Rand National Defense Research Institute.

LAMPIRAN

Lampiran 1 : Pemrograman Arduino

Program akses GPS

```
#include <TinyGPS++.h>
#include "motor.h"
#include <Wire.h>
#include <MPU6050.h>

MPU6050 mpu;

TinyGPSPlus gps;

int   target  = 1 ; //target tujuan
int   steer   = 0 ;
int   throttle = 0 ;
float yaw;
float heading;

int kiri;
int kanan;

float lattitude;
float longitude;
float error_sudut, previous_error_sudut, p_sudut, d_sudut, PID_sudut;
float distance, error_posisi, previous_error_posisi, p_posisi, d_posisi,
PID_posisi;

void setupGPS(void)
{
  //Serial untuk G.P.S
  Serial1.begin (9600);

  while(!mpu.begin(MPU6050_SCALE_2000DPS,
MPU6050_RANGE_2G))
  {
```

Lampiran 2 : Lanjutan

```
Serial.println("Could not find a valid MPU6050 sensor, check wiring!");
```

```
    delay(500);  
}
```

```
mpu.setI2CMasterModeEnabled(false);  
mpu.setI2CBypassEnabled(true) ;  
mpu.setSleepEnabled(false);  
}
```

```
void loopGPS(void)
```

```
{  
    double bearing ;  
    double target_lat;  
    double target_lon;
```

```
//kompas void setup
```

```
Wire.begin();  
Wire.beginTransaction(0x1E);  
Wire.write(0x02);  
Wire.write(0x00);  
Wire.endTransmission();
```

```
//compas
```

```
int mag_x, mag_y, mag_z;  
Wire.beginTransaction(0x1E);  
Wire.write(0x03);  
Wire.endTransmission();
```

```
Wire.requestFrom(0x1E, 6);
```

```
if(6 <= Wire.available())
```

```
{  
    mag_x = Wire.read()<<8; //MSB x  
    mag_x |= Wire.read() ; //LSB x  
    mag_z = Wire.read()<<8; //MSB z
```

Lampiran 3 : Lanjutan

```
mag_z |= Wire.read() ; //LSB z
mag_y = Wire.read() << 8; //MSB y
mag_y |= Wire.read() ; //LSB y
}
//heading = atan2 (mag_y, mag_x) * 57,2958;
heading = atan2 (mag_y, mag_x);

float declinationAngle = 0.22;
heading += declinationAngle ;
if (heading < 0)
    heading += 2 * PI;
if (heading > 2 * PI)
    heading -= 2 * PI;
yaw = heading * 57.2958;

//baca gps
while (Serial1.available() > 0) ////////////tidak di void setup??
    if (gps.encode( Serial1.read() ))
    if (gps.location.isValid())
    {
        latitude = gps.location.lat();
        longitude = gps.location.lng();
    }
//=====
/*
switch (target){
    case 1:
        target_lat = -7.286309, target_lon = 112.795892; // danau 8 putaran
1 bentuk persegi
        break;
    case 2:
        target_lat = -7.286457, target_lon = 112.795906;
        break;
    case 3:
```

Lampiran 4 : Lanjutan

```
target_lat = -7.286435, target_lon = 112.796124;
break;
case 4:
    target_lat = -7.286270, target_lon = 112.796089;
    break;
}
*/

switch (target){
    case 1:
        target_lat = -7.286259, target_lon = 112.796062; //danau 8 putaran
2 bentuk trapesium
        break;
    case 2:
        target_lat = -7.286406, target_lon = 112.796131;
        break;
    case 3:
        target_lat = -7.286456, target_lon = 112.795887;
        break;
    case 4:
        target_lat = -7.286298, target_lon = 112.795921;
        break;
}

/*
switch (target){
    case 1:
        target_lat = -7.286277, target_lon = 112.795959; //danau 8 putaran
3 bentuk tidak beraturan
        break;
    case 2:
        target_lat = -7.286374, target_lon = 112.795850;
        break;
    case 3:
        target_lat = -7.286369, target_lon = 112.795973;
```

Lampiran 5 : Lanjutan

```
        break;
    case 4:
        target_lat = -7.286480, target_lon = 112.795925;
        break;
    case 5:
        target_lat = -7.286525, target_lon = 112.796010;
        break;
    case 6:
        target_lat = -7.286460, target_lon = 112.796096;
        break;
    case 7:
        target_lat = -7.286360, target_lon = 112.796110;
        break;
    case 8:
        target_lat = -7.286257, target_lon = 112.796048;
        break;
    }
*/

float latitude_rad = latitude * 0.01745329252;
float longitude_rad = longitude * 0.01745329252;
float target_lat_rad = target_lat * 0.01745329252;
float target_lon_rad = target_lon * 0.01745329252;

float y = sin(target_lon_rad - longitude_rad) * cos(target_lat_rad);
float x = cos(latitude_rad) * sin(target_lat_rad) - sin(latitude_rad) *
cos(target_lat_rad) * cos(target_lon_rad - longitude_rad);
bearing = atan2(y, x) * 57.295779513;

float earth_radius = 6371.137 * 1000 ;           // (radius bumi dalam
satuan km) >> 1 km = 1000 meter
float delta_lat = target_lat_rad - latitude_rad;
float delta_lon = target_lon_rad - longitude_rad;
```


Lampiran 6 : Lanjutan

```
float a = sin(delta_lat / 2) * sin(delta_lat/2) + sin(delta_lon / 2) *
sin(delta_lon / 2) * cos(latitude_rad) * cos(target_lat_rad);
float c = 2 * atan2(sqrt(a), sqrt(1-a));
float distance = earth_radius * c;

//=====================================================KONTROL PID=====================================================//
error_sudut = yaw - bearing;
    if (error_sudut < -180) error_sudut += 360;
    else if (error_sudut > 180) error_sudut -= 360;
p_sudut = 0.5 * error_sudut;
d_sudut = 0.1 * (error_sudut - previous_error_sudut);
PID_sudut = p_sudut; //*****
    previous_error_sudut = error_sudut;
    if (PID_sudut > 100) PID_sudut = 100; //63
    else if (PID_sudut < -100) PID_sudut = -100; //-63

error_posisi = abs(distance) * cos(error_sudut * 0.01745329252);
p_posisi = 50 * error_posisi;
d_posisi = 5 * (error_posisi - previous_error_posisi);
PID_posisi = p_posisi + d_posisi;
//*****
    previous_error_posisi = error_posisi;
    if (PID_posisi > 100) PID_posisi = 100;
    else if (PID_posisi < -100) PID_posisi = -100;

throttle = 125; //125
steer = PID_sudut; //Otomatis

//=====================================================KONTROLMOTOR=====================================================//
kiri = throttle - steer;
kanan = throttle + steer;

kanan = constrain(kanan, -255, 255);
kiri = constrain(kiri, -255, 255);
```

Lampiran 7 : Lanjutan

```
Serial.print(latitude, 6) ; Serial.print(' ') ; Serial.print(longitude,
6); Serial.print("\t");
Serial.print("heading:") ; Serial.print(yaw) ; Serial.print("\t");
Serial.print("bearing: ") ; Serial.print(bearing) ; Serial.print("\t");
Serial.print("beda sudut:"); Serial.print(error_sudut); Serial.print("\t");
Serial.print("PID sudut:"); Serial.print(PID_sudut) ; Serial.print("\t");

//algoritma control motor
if (PID_sudut<-10){
    digitalWrite(R1, LOW);
    digitalWrite(R2, HIGH); //LOW
    digitalWrite(L1, HIGH);
    digitalWrite(L2, LOW);
    Serial.print("Kanan");
    Serial.print("\t");
}
else if (PID_sudut>=-10 && PID_sudut <=10){
    digitalWrite(R1, HIGH);
    digitalWrite(R2, LOW) ;
    digitalWrite(L1, HIGH);
    digitalWrite(L2, LOW) ;
    Serial.print("Maju");
    Serial.print("\t");
}
else if (PID_sudut>10){
    digitalWrite(R1, HIGH);
    digitalWrite(R2, LOW);
    digitalWrite(L1, LOW);
    digitalWrite(L2, HIGH); //LOW
    Serial.print("Kiri");
    Serial.print("\t");
}
analogWrite(pwml, abs(kiri) );
analogWrite(pwmr, abs(kanan) );
```

Lampiran 8 : Lanjutan

```
//=====WAY POINT (LOOPING) =====//
if((distance <= 5) && (target < 4)) //distance dalam satuan Meter//5
{
    target++;
    Serial.print("target : "); Serial.println(target);
}
else if((distance <= 5) && (target == 4))//5
{
    digitalWrite (L1, LOW) ;
    digitalWrite (L2, LOW) ;
    digitalWrite (R1, LOW) ;
    digitalWrite (R2, LOW) ;
    analogWrite (pwmr, 0) ;
    analogWrite (pwml, 0) ;
    Serial.print("\n");
    Serial.print("== Diam ==");
    //Serial.print("\n");
}

Serial.print("\n");
} //=====batasloop=====//
```

Lampiran 9 : Lanjutan

Program Akses Bluetooth

```
int Lamp1;
int Lamp2;
int Lamp3;
int Lamp4;
int Lamp5;
int Lamp6;
int Lamp7;
int Lamp8;
int status_otomatis;
int status_data_masuk;

#define ON 1 //If Out to active Low change ON 0 and OFF 1
#define OFF 0 //If Out to active High change ON 1 and OFF 0
char val;
String
statusLamp1,statusLamp2,statusLamp3,statusLamp4,statusLamp5,status
Lamp6,statusLamp7,statusLamp8;

void setupbluetooth(void)
{
    Serial2.begin(9600);
}
int cek =0;
void loopbluetooth(void)
{
    if( Serial2.available() >0 )
    {
        cek++;
        val = Serial2.read();
        Serial2.println(val); // Serial.println(val);
        if(val == '8') status_otomatis = !status_otomatis;
    }
}
```

Lampiran 10 : Lanjutan

Program akses untuk gerak kapal manual

```
int R1 = 6;    // Kanan Maju  // ke in 1
int R2 = 7;    // Kanan Mundur // ke in 2
int pwmr = 3;
int L1 = 5;    // Kiri Maju  // ke in 3
int L2 = 4;    // Kiri Mundur // ke in 4
int pwml = 2;

int led = 13;

void L_CCW(void){
    digitalWrite(L1, LOW);
    digitalWrite(L2, HIGH);
}

void L_CW(void){
    digitalWrite(L1, HIGH);
    digitalWrite(L2, LOW);
}

void R_CCW(void){
    digitalWrite(R1, LOW);
    digitalWrite(R2, HIGH);
}

void R_CW(void){
    digitalWrite(R1, HIGH);
    digitalWrite(R2, LOW);
}

void motor_gerak(signed int motor_kiri, signed int motor_kanan)
{

    if(motor_kiri >= 0) {L_CW();
```

Lampiran 11 : Lanjutan

```
// digitalWrite(L1, HIGH);
// digitalWrite(L2, LOW) ;
}
else {L_CCW();
//digitalWrite(L1, LOW) ;
//digitalWrite(L2, HIGH);
}

if(motor_kanan >= 0) {R_CW();}
else {R_CCW();}

analogWrite(pwml, abs(motor_kiri));
analogWrite(pwmr, abs(motor_kanan));
}

void kapal_gerak(signed int maju, signed int belakang)
{
    signed int motor_kanan;
    signed int motor_kiri;

    motor_kanan = maju + belakang;
    motor_kiri = maju - belakang;

    if(motor_kanan > 255) motor_kanan = 255;
    if(motor_kanan < -255) motor_kanan = -255;

    if(motor_kiri > 255) motor_kiri = 255;
    if(motor_kiri < -255) motor_kiri = -255;

    motor_gerak(motor_kiri, motor_kanan);
}

void kapal_gerak_manual(char command)
{
    static signed int maju = 0 ;
```

Lampiran 12 : Lanjutan

```
static signed int belok = 0 ;
```

```
switch(command)
{
    case '1':
        maju = 125;
        belok = 0;
        Serial.println("Maju") ;
        break;
    case '2':
        maju = -125;
        belok = 0;
        Serial.println("mundur") ;
        break;
    case '3':
        maju = 125;
        belok = -125;
        Serial.println("kanan") ;
        break;
    case '4':
        maju = 125;
        belok = 125;
        Serial.println("kiri") ;
        break;
    case '5':
        maju = 0;
        belok = -125;
        Serial.println("M.Kanan") ;
        break;
    case '6':
        maju = 0;
        belok = 125;
        Serial.println("M.Kiri") ;
        break;
    case '7':
```

Lampiran 13 : Lanjutan

```
    maju = 0;
    belok = 0;
    Serial.println("Stop")  ;
    break;
}

kapal_gerak(maju, belok);
}
```


Lampiran 14 : Lanjutan

Program keseluruhan ASV

```
#include "GPS.h"
#include "bluetooth.h"

char tampil[100];
int buzzer=22;

void setup()
{
  pinMode(R1, OUTPUT);
  pinMode(R2, OUTPUT);
  pinMode(L1, OUTPUT);
  pinMode(L2, OUTPUT);
  pinMode(pwml, OUTPUT);
  pinMode(pwmr, OUTPUT);

  setupGPS ();
  setupbluetooth();
  Serial.begin(9600);
  pinMode (buzzer,OUTPUT);

  for(int i=0;i<25;i++)
  {
    if(i%2==0)digitalWrite(buzzer,HIGH);
    else digitalWrite(buzzer,LOW);
    delay(30);
  }
}

void loop()
{
  loopbluetooth ();
  switch(status_otomatis)
```

```

{
Lampiran 15 : Lanjutan
case 0: // manual
    kapal_gerak_manual(val);
    sprintf(tampil, "MANUAL command: %c", val);
    Serial.print(tampil);
    Serial.print('\n');
    delay(200); //100

    break;
case 1:
    loopGPS();
    delay(200); //100
    break;
}
}

```

Lampiran 16 : Spesifikasi Motor



waterproof dc motor with shaft dan propeller

Voltage: 12V DC

Current: 3.0A

Diameter: 60mm

Color: show as the picture

Exclusive moisture tight seals.

Compact, efficient, long life motors.

Easy clean snap lock strainers.

Complete submersibility.

Silent and vibrationless operation.

Ignition protection.

Anti-airlock protection.

No burn-out when run dry.

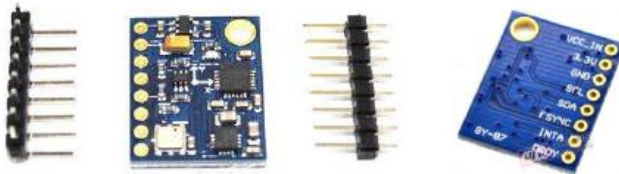
Stainless steel shafts.

Lampiran 17 : Spesifikasi Sensor GPS

Parameter	Specification					
Receiver type	72-channel u-blox M8 engine GPS L1C/A, SBAS L1C/A, QZSS L1C/A, QZSS L1 SAIF, GLONASS L1OF, BeiDou B1I, Galileo E1B/C					
Accuracy of time pulse signal	RMS	30 ns				
	99%	60 ns				
Frequency of time pulse signal	0.25 Hz...10 MHz (configurable)					
Operational limits ¹	Dynamics	≤ 4 g				
	Altitude	50,000 m				
	Velocity	500 m/s				
Velocity accuracy ²	0.05m/s					
Heading accuracy ²	0.3 degrees					
GNSS		GPS & GLONASS	GPS	GLONASS	BeiDou	Galileo
Horizontal position accuracy ³		2.5 m	2.5 m	4 m	3 m	TBC ⁴
NEO-M8N/Q						
Max navigation update rate	NEO-M8N	5 Hz	10 Hz	10 Hz	10 Hz	10 Hz
	NEO-M8Q	10 Hz	18 Hz	18 Hz	18 Hz	18 Hz
Time-To-First-Fix ⁵	Cold start	26 s	29 s	30 s	34 s	45 s
	Hot start	1 s	1 s	1 s	1 s	1 s
	Aided starts ⁶	2 s	2 s	2 s	3 s	7 s
Sensitivity ⁷	Tracking & Navigation	-167 dBm	-166 dBm	-166 dBm	-160 dBm	-159 dBm
	Reacquisition	-160 dBm	-160 dBm	-156 dBm	-157 dBm	-153 dBm
	Cold start	-148 dBm	-148 dBm	-145 dBm	-143 dBm	-138 dBm
	Hot start	-157 dBm	-157 dBm	-156 dBm	-155 dBm	-151 dBm

Lampiran 18 : Spesifikasi Sensor Kompas

GY-87 10DOF MPU6050 HMC5883L BMP180 Sensor Module



Features

- Acceleration, Gyroscope and Magnetometer
- 10DOF modules (three-axis gyroscope + triaxial accelerometer and three-axis magnetic field + atmospheric pressure)
- Build in ultra low noise linear LDO voltage regulator
- Build-in on board filters, which reduce noise from motor and other high current electronics
- All sensors connected to I2C bus
- Build in Logic level converter for I2C
- Power indicator LED

Specifications

- Power Supply: DC3.3V-5V
- Communication: IIC protocol (fully compatible with 3-5V system containing LLC circuit)
- Chips: MPU6050 + HMC5883 + BMP180
- Size: 2.2cmx1.7cm
- Weight: 6g

Lampiran 19 : Spesifikasi Driver Motor

1 Overview

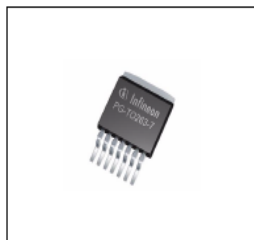
Features

- Path resistance of max. 30.5 mΩ @ 150 °C (typ. 16 mΩ @ 25 °C)
High Side: max. 12.8 mΩ @ 150 °C (typ. 7 mΩ @ 25 °C)
Low Side: max. 17.7 mΩ @ 150 °C (typ. 9 mΩ @ 25 °C)
(for BTN7970B (SMD))
- Low quiescent current of typ. 7 μA @ 25 °C
- PWM capability of up to 25 kHz combined with active freewheeling
- Switched mode current limitation for reduced power dissipation in overcurrent
- Current limitation level of 50 A min. / 70 A typ. (low side)
- Status flag diagnosis with current sense capability
- Overtemperature shut down with latch behaviour
- Overvoltage lock out
- Undervoltage shut down
- Driver circuit with logic level inputs
- Adjustable slew rates for optimized EMI
- Green Product (RoHS compliant)
- AEC Qualified

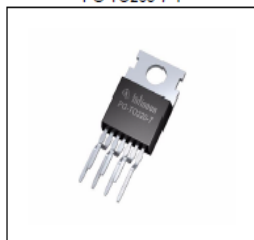
Description

The BTN7970 is a integrated high current half bridge for motor drive applications. It is part of the NovalithIC™ family containing one p-channel highside MOSFET and one n-channel lowside MOSFET with an integrated driver IC in one package. Due to the p-channel highside switch the need for a charge pump is eliminated thus minimizing EMI. Interfacing to a microcontroller is made easy by the integrated driver IC which features logic level inputs, diagnosis with current sense, slew rate adjustment, dead time generation and protection against over-temperature, overvoltage, undervoltage, overcurrent and short circuit.

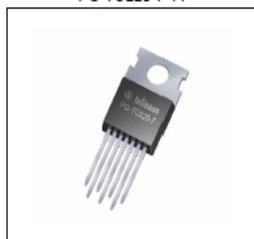
The BTN7970 provides a cost optimized solution for protected high current PWM motor drives with very low board space consumption.



PG-TO263-7-1



PG-TO220-7-11



PG-TO220-7-12

BIODATA PENULIS



Penulis dilahirkan di Sidoarjo, 20 Desember 1996, merupakan anak kedua dari 3 bersaudara. Penulis beralamat di Perumahan Bumi Citra Fajar Sekawan Nyaman A-21/a, Sidoarjo. Penulis juga telah menempuh pendidikan formal yaitu di SDN Pucang Anom Sidoarjo, SMPN 6 Sidoarjo dan SMAS Hang Tuah 2 Sidoarjo. Setelah lulus dari SMA tahun 2015, penulis diterima di ITS jurusan D3 Teknik Mesin dengan Nomor

Registrasi Pokok (NRP) 2115030006.

Selama masa perkuliahan penulis aktif dalam mengikuti kegiatan dan organisasi. Pada tahun pertama, penulis diterima menjadi salah satu Tim Robot ITS di divisi tim robot ABU Robocon ITS dan telah mendapatkan penghargaan kejuaraan di regional dan nasional. Di tahun kedua, penulis aktif dalam kegiatan lab menjadi Asisten Laboratorium Mekatronika dan aktif dalam organisasi yaitu terpilih menjadi Kepala Departemen Riset dan Teknologi di Himpunan Mahasiswa D3 Teknik Mesin (HMDM FV-ITS). Penulis juga telah melakukan Kerja Praktek di PT. Pertamina Drilling Services Indonesia. Di tahun ketiga, mengikuti kegiatan GERIGI ITS 2018 sebagai mentor dan mengambil bidang studi Konversi Energi pada Departemen Teknik Mesin Industri FV-ITS. Jika ingin mengenal dan sharing ilmu mengenai tugas akhir ini bisa menghubungi email : naufalocal@gmail.com.