

# Sistem Navigasi *Autonomous* Pada Prototipe Kapal dengan Sensor GPS dan Kompas

Rayyan Nur Fauzan<sup>1</sup>, M. Syariffuddien Zuhrie<sup>2</sup>

<sup>1,2</sup> Jurusan Teknik Informatika, Fakultas Teknik, Universitas Negeri Surabaya  
[rayyannur.20075@mhs.unesa.ac.id](mailto:rayyannur.20075@mhs.unesa.ac.id)

<sup>2,1</sup> Jurusan Teknik Elektro, Fakultas Teknik, Universitas Negeri Surabaya  
[zuhrie@unesa.ac.id](mailto:zuhrie@unesa.ac.id)

**Abstrak**— Penguasaan teknologi kapal wisata dan teknologi digital dalam sektor kemaritiman menjadi krusial untuk mengoptimalkan pengalaman wisata bahari secara visual maupun non visual. KKCTBN 2023 menggerakkan pengembangan prototipe kapal wisata tanpa awak sebagai langkah maju. Sistem navigasi *autonomous* dengan GPS Neo 6m dan sensor kompas HMC5883L dipilih untuk memastikan jalur pelayaran yang akurat. GPS Neo 6m dan HMC5883L dikenal karena efisiensinya. Penelitian ini bertujuan membangun Rancang Sistem Navigasi *Autonomous* Pada Prototipe Kapal *Autonomous Tourism Surface Vessel* dengan Sensor GPS Neo 6m dan Sensor Kompas HMC5883L. GPS Neo 6m akan memberikan titik koordinat prototipe kapal sedangkan HMC5883L memberikan nilai arah mata angin sebagai dasar dari sistem navigasi. Metode yang digunakan pada penelitian ini terdiri dari beberapa langkah yaitu analisis kebutuhan, desain rancangan prototipe, implementasi, pengujian, dan hasil penelitian. Pada pengujian sensor GPS Neo 6m, didapatkan nilai rata-rata error 0,52 m dan error tertinggi 1,6 m. Sedangkan pada pengujian sensor HMC5883L didapatkan nilai rata-rata error 1,1° dan error tertinggi 2°. Hasil dari penelitian ini prototipe kapal mampu bergerak sesuai dengan titik yang sudah ditentukan. Dilakukan iterasi sebanyak lima kali dengan keberhasilan 86%.

**Kata Kunci**— Navigasi, *Autonomous*, Kapal, GPS Neo 6m, HMC5883L.

## I. PENDAHULUAN

Dalam era kemajuan teknologi yang cepat, perkembangan teknologi otonom telah memiliki dampak signifikan dalam berbagai sektor kehidupan, termasuk industri perkapalan. Kapal otonom, yang mampu beroperasi tanpa kehadiran manusia di atas kapal, menjadi salah satu inovasi yang menarik minat banyak pihak. Kemajuan dalam teknologi navigasi otonom, sensor yang canggih, dan sistem kendali mandiri telah memungkinkan kapal otonom untuk berlayar dengan efisiensi yang lebih tinggi, meningkatkan keselamatan, dan mengurangi dampak lingkungan[1].

Penguasaan teknologi kapal wisata dan teknologi digital dalam sektor kemaritiman menjadi krusial untuk mengoptimalkan pengalaman wisata bahari secara visual maupun non visual. Diperlukan inovasi dan penemuan terus-menerus dalam teknologi ini agar produk yang dihasilkan dapat memenuhi kebutuhan yang spesifik dalam konteks pariwisata bahari di Indonesia[2]. Kebutuhan teknologi kapal wisata mencakup aspek-aspek seperti navigasi yang presisi, sistem keamanan yang canggih, serta kenyamanan dan fasilitas modern untuk wisatawan. Teknologi navigasi berbasis GPS dan radar yang mutakhir sangat diperlukan untuk memastikan

keselamatan perjalanan di perairan Indonesia yang luas dan beragam.

Balai Pengembangan Talenta Indonesia Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi menyelenggarakan KKCTBN (Kontes Kapal Cepat Tak Berawak Nasional) 2023 yang merupakan agenda penting mahasiswa dalam hal pengembangan inovasi teknologi kemaritiman digital guna ikut andil dalam ide untuk mendukung pariwisata kemaritiman. ATSV (*Autonomous Tourism Surface Vessel*) merupakan salah satu divisi yang diperlombakan dalam KKCTBN yang mampu tanpa kendali manual dengan daya penggerak motor listrik otomatis dengan misi menuju spot wisata dan melihat wisata diatas permukaan air (burung dan lomba-lomba) dan kembali ke pelabuhan asal dengan benar [2].

Dalam rangka memastikan kapal dapat menyusuri rute dengan benar, perlu diterapkan sistem navigasi otomatis yang dapat diintegrasikan dengan teknologi mutakhir, seperti sistem navigasi GPS/GNSS. GNSS (*Global Navigation Satellite System*) beroperasi dengan mentransmisikan sinyal gelombang mikro (frekuensi dalam rentang 1–2 GHz), yang berisi waktu dan posisi orbital yang tepat dari satelit, ke antena dan penerima di Bumi, umumnya di permukaan tanah. Antena memerlukan pandangan yang tidak terhalangi dari setidaknya empat satelit terpisah untuk menghitung lintang, bujur, ketinggian antena, dan kesalahan waktu dari sinyal yang ditransmisikan[3]. Dari banyaknya varian GPS sensor, GPS Neo 6M dipilih didasarkan karena efisiensi dari segi bentuk, daya, dan harga. GPS seri ini dikembangkan oleh u-blox yang memiliki banyak keunggulan yang membuatnya cocok digunakan dalam berbagai aplikasi. Dirancang dengan tujuan konsumsi daya rendah dan biaya yang terjangkau[4].

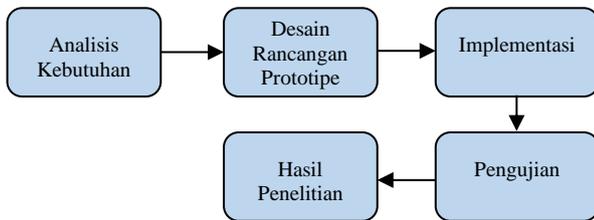
Selain GPS/GNSS diperlukan kompas dalam membantu pengaplikasian sebuah sistem navigasi untuk mendapatkan nilai *heading*. Dengan memanfaatkan pengukuran komponen medan magnet Bumi, kompas elektronik yang berbasis sensor *magneto-resistive* (MR) dapat menyelesaikan perubahan arah *heading* secara elektrik dengan akurasi lebih baik dari 0,1 derajat, yang sudah cukup baik untuk diterapkan pada sistem pengamatan jarak jauh[5]. HMC5883L adalah salah satu sensor *3-axis magnetometer*. *Magnetometer* memiliki berbagai penggunaan. Yang paling umum termasuk menggunakan chip sebagai kompas digital untuk mengetahui arah atau untuk mendeteksi logam *ferrous* (magnetik). Medan magnet dan arus berkaitan erat, ketika arus mengalir melalui kawat, medan magnet terbentuk. Ini adalah prinsip dasar dibalik elektromagnet. Prinsip ini juga digunakan untuk mengukur

medan magnet dengan magnetometer. Arah medan magnet bumi mempengaruhi aliran elektron dalam sensor, dan perubahan tersebut dalam arus dapat diukur dan dihitung untuk mendapatkan arah kompas atau informasi berguna lainnya[6].

Berdasarkan latar belakang tersebut, penulis melakukan penelitian dengan tujuan membangun Rancang Sistem Navigasi Autonomus Pada Prototipe Kapal dengan Sensor GPS Neo 6M dan Sensor Kompas HMC5883L.

## II. METODE PENELITIAN

Pada penelitian Sistem Navigasi Autonomus Pada Prototipe Kapal dengan Sensor GPS dan Kompas menggunakan metodologi penelitian kuantitatif yang mana sumber datanya diambil melalui sampel yang bersifat matematis. Pendekatan kuantitatif dipilih karena memungkinkan penggunaan data yang dapat diukur secara objektif. Metode ini akan diterapkan pada tahap pengumpulan data, analisis data, dan penarikan kesimpulan. Dalam penelitian ini, terdapat beberapa langkah yang harus dilalui untuk memulai proses penyelesaian penelitian. Langkah-langkah ini diilustrasikan dalam diagram berikut ini.



Gbr. 1 Bagan Metode Penelitian

### A. Analisa Kebutuhan

Pada tahap ini dilakukan analisis kebutuhan yang akan digunakan untuk membangun sistem navigasi otomatis pada prototipe kapal *Autonomous Tourism Surface Vessel*.

#### 1) Kebutuhan Komponen

TABEL I  
KEBUTUHAN KOMPONEN

KOMPONEN	SPESIFIKASI	FUNGSI
Raspberry PI 3 Model B	CPU Quad-Core 1.2GHz Broadcom BCM2837 64bit RAM 1GB	Mainframe Computing
ESP32	Dua inti Tensilica Xtensa LX6 CPU 240Mhz RAM 512Kb-4Mb	Mikrokontroler untuk sensor kompas, sensor GPS, dan aktuator motor T200.
Sensor Kompas HMC5883L	3,3 V Output X,Y,Z integrated I2C	Mendapatkan nilai <i>heading</i> dari kompas.
Sensor GPS NEO6MV2	3,3 V Baudrate 4800bps- 115200bps module IoT	Mendapatkan titik koordinat bumi seperti <i>latitude</i> dan <i>longitude</i>

USB TTL CH340G	3,3V – 5V 2400bps- 115200bps Versi 2.0	Jembatan komunikasi serial dari Raspberry PI ke ESP32
Basic ESC30A Bidirectional	30A 5V-12V BLDC	<i>Driver</i> untuk motor <i>brushless</i> .
Motor <i>Brushless</i> T200	7V-20V	Motor Penggerak/Aktuator

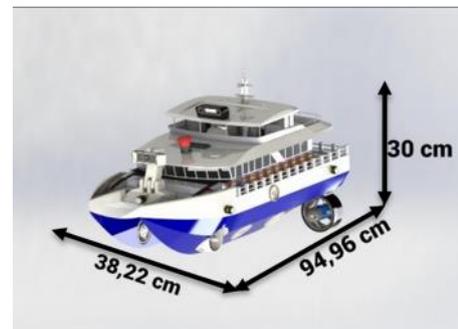
#### 2) Kebutuhan Software

- Arduino IDE 2.0, sebagai *software* yang digunakan untuk memrogram mikrokontroler ESP32.
- Thonny, Sebagai *code editor* yang digunakan untuk memprogram program Python pada Raspberry PI.
- TeamViewer, sebagai aplikasi *remote desktop* untuk monitoring Raspberry PI dengan komputer terpisah di jaringan yang sama.

### B. Desain Rancangan Prototipe

Pada tahap ini dilakukan perancangan prototipe yang terdiri dari perancangan desain mekanik, *hardware*, dan *software*.

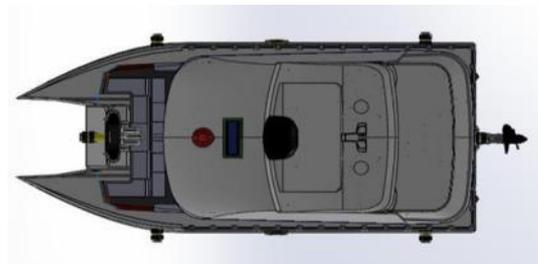
1) *Perancangan Mekanik*:Rancangan desain 3D pada prototipe kapal *Autonomous Tourism Surface Vessel*.



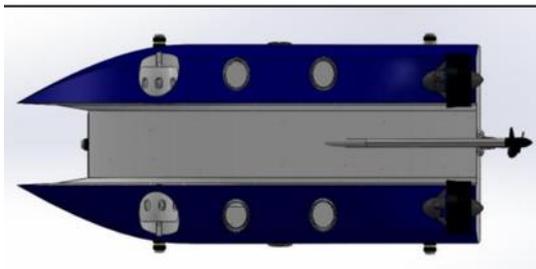
Gbr. 2 Dimensi Ukuran Kapal ATSV



Gbr. 3 Kapal ATSV Tampak Samping

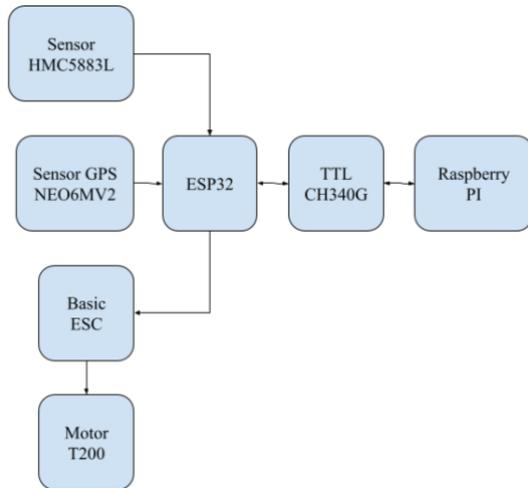


Gbr. 4 Kapal ATSV Tampak Atas



Gbr. 5 Kapal ATSV Tampak Bawah

2) *Perancangan Hardware*: Desain *schematic hardware* yang akan digunakan pada pembuatan sistem navigasi otomatis.



Gbr. 6 Bagan Perancangan Hardware

ESP32 adalah platform yang kuat dan hemat biaya untuk mengembangkan aplikasi IoT. ESP32, yang dikembangkan oleh Espressif Systems Company (Shanghai, Tiongkok), menawarkan kombinasi fitur dan kemampuan yang kuat untuk aplikasi IoT. ESP32 memiliki beberapa fitur berikut: (1) prosesor *dual-core*, (2) konektivitas Wi-Fi dan *Bluetooth* terintegrasi, (3) sejumlah besar pin *input/output* tujuan umum (GPIO), dan (4) konsumsi daya rendah[7].

Raspberry Pi merupakan komputer berbasis Linux yang sangat murah dan sebagai platform untuk komputasi tertanam. Ini telah terbukti populer di kalangan pendidik dan penghobi. Ketersediaan Perangkat Lunak Linux *open source* untuk *browsing* internet, email, *office suite*, dan pengeditan foto diatur untuk membuat raspberry Pi semakin populer[8].

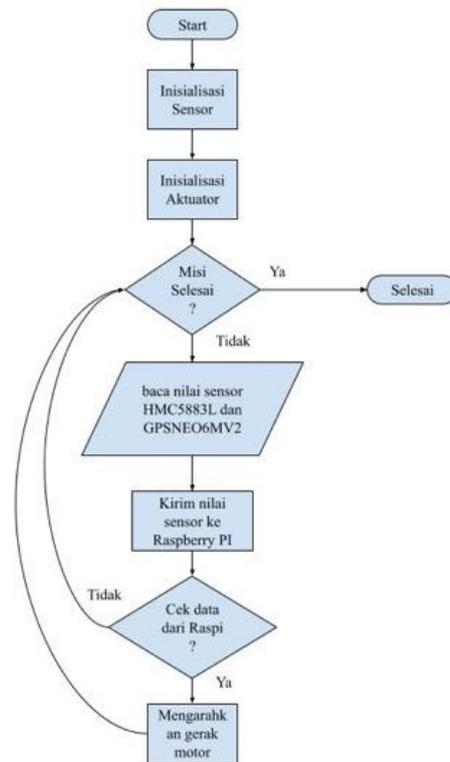
GPS Neo 6m adalah bagian dari seri ini yang menampilkan mesin pemosisian u-blox 6 berkinerja tinggi. Ia menawarkan berbagai pilihan konektivitas dalam paket miniatur 16 x 12,2 x 2,4 mm. Karena biaya dan konsumsi daya yang rendah, serta ukurannya yang kecil, GPS Neo 6m digunakan pada perangkat seperti ponsel pintar yang jumlah ruang dan daya yang tersedia lebih sedikit[4].

HMC5883L adalah modul *multi-chip* yang dipasang di permukaan dirancang untuk penginderaan magnetik medan rendah dengan antarmuka digital untuk aplikasi seperti kompas dan magnetometri berbiaya rendah. HMC5883L memiliki

ADC 12-bit yang memungkinkan akurasi arah kompas 1° hingga 2°[9].

ESP32 Sebagai kontroler untuk membaca sensor dari HMC5883L dan sensor GPS Neo 6M. Semua data sensor akan dikirimkan ke Raspberry PI untuk diolah datanya. Raspberry PI akan mengirimkan data arah motor yang akan dikirimkan ke ESP32. ESP32 akan mengeksekusi data tersebut untuk menggerakkan motor.

3) *Perancangan Software*: Pada perancangan software dilakukan pembuatan *flowchart*. *Flowchart* yang akan dibuat dibagi menjadi dua yaitu *flowchart* program yang ada di ESP32 dan Raspberry PI.

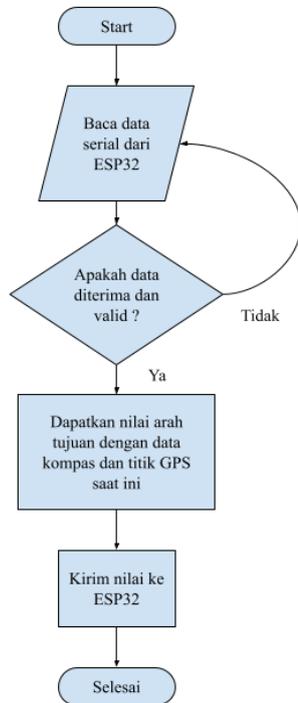


Gbr. 7 Flowchart program yang ada di esp32

Arduino IDE adalah perangkat lunak sumber terbuka yang menulis, mengkompilasi, dan mengunggah kode langsung ke mikrokontroler. Versi yang digunakan dalam tulisan ini adalah versi 2.0. Lingkungan Arduino IDE digunakan untuk menulis kode perangkat lunak yang diinginkan dan untuk mengkompilasi, mengunggah kode ke papan Arduino yang diberikan. Lingkungannya mendukung bahasa C dan C++. Ini juga digunakan untuk *debugging*, mengedit, mengkompilasi, dan mengunggah kode di lingkungannya ke modul perangkat keras fisik[10].

Pada ESP32, program dimulai dari setup, yaitu menginisialisasi sensor HMC5883L dan sensor GPS Neo 6M, selanjutnya diikuti inisialisasi aktuator ke dalam nilai 0. Selanjutnya masuk ke dalam *looping*. Didalam *looping* terdapat proses pembacaan nilai dari kedua sensor lalu dikirimkan ke Raspberry PI melalui komunikasi serial. Setelah itu, program

akan membaca apakah ada data yang diterima dari Raspberry PI, apabila ada data masuk maka data tersebut akan digunakan untuk menggerakkan arah dan kecepatan motor/aktuator. Program akan berputar terus menerus didalam *looping* hingga misi dinyatakan selesai. Berikut adalah *flowchart* dari program ESP32.



Gbr. 8 Flowchart program Raspberry PI

Python adalah bahasa pemrograman yang memungkinkan koneksi sistem lebih cepat dan efisien[11]. Peneliti menggunakan program ini untuk mengoneksikan dengan mikrokontroler ESP32. Dengan bantuan pengkodean bahasa Python yang mudah, peneliti bisa melakukan *multithreading*.

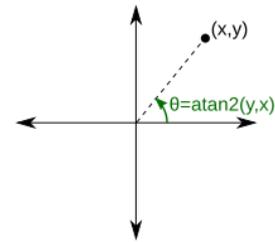
Selanjutnya pada program di Raspberry PI dimulai dari membaca data sensor yang dikirimkan oleh ESP32 berupa data *heading* bumi dan koordinat bumi yaitu *latitude* dan *longitude*. Setelah itu, dengan data tersebut diolah untuk mendapatkan nilai tujuan, berupa arah mata angin selanjutnya. Data tersebut akan dikirim ke ESP32 yang nantinya akan berupa kecepatan antara kedua motor tersebut.

### C. Implementasi

Pada tahap implementasi akan dijelaskan mengenai proses mendapatkan nilai heading dari data mentah sensor kompas HMC5883L, proses pengolahan nilai *latitude* dan *longitude*, dan yang terakhir proses mendapatkan nilai arah mata angin tujuan dan jarak dari nilai *bearing* dan koordinat bumi.

1) *Pengolahan data HMC5883L*: Sensor HMC5883L memberikan output berupa data medan magnetik dalam tiga dimensi (sumbu X, Y, dan Z). Umumnya, keluaran dari sensor HMC5883L diukur dalam satuan Gauss. Untuk mendapatkan

sudut *heading*, dibutuhkan sumbu X dan Ynya. Dari sumbu tersebut dicari sudut yang terbentuk dari koordinat  $x = 0$ .



Gbr. 9 Fungsi atan2

Dari sudut tersebut, maka akan dihasilkan sudut *heading* magnetik bumi. Akan tetapi dibutuhkan *Magnetic declination* sesuai dengan posisi kompas terhadap bumi. Berikut rumus lengkap mendapatkan sudut *heading*.

$$\theta = (\text{atan2}(Y, X) + \text{declination}) \times \frac{180}{\pi} \quad (1)$$

2) *Pengolahan data GPS Neo 6MV2*: Sensor GPS dibutuhkan untuk mendapatkan titik koordinat bumi berupa *latitude* dan *longitude*. Output dari sensor tersebut berupa data NMEA (*National Marine Electronics Association*). Data tersebut perlu di *parsing* menjadi data *latitude* dan *longitude*. Baris yang berisi data *latitude* dan *longitude* yaitu \$GPGL.

3) *Mendapatkan Jarak dan Bearing dari Dua Koordinat Bumi*: Pada proses ini akan mendapatkan sudut *bearing* yang akan menjadi sudut tujuan kemana kapal akan bergeser. Diambil dari koordinat kapal dan koordinat tujuan. 'L' melambangkan *longitude*, 'θ' melambangkan *latitude*, dan 'β' melambangkan *bearing*. Sedangkan 'a' melambangkan titik koordinat pertama dan 'b' melambangkan titik koordinat kedua. *Bearing* dari titik A ke titik B dapat dihitung dengan rumus berikut.

$$\beta = \text{atan2}(Y, X) \quad (2)$$

Oleh karena itu, perlu menghitung nilai X dan Y terlebih dahulu. nilai X dapat dihitung dengan rumus berikut.

$$X = \cos \theta a \times \sin \theta a - \sin \theta a \times \cos \theta b \times \cos \Delta L \quad (3)$$

Sedangkan untuk mendapatkan nilai Y bisa menggunakan rumus berikut.

$$Y = \cos \theta b \times \cos \Delta L \quad (4)$$

Akan tetapi nilai *bearing* masih dalam bentuk radian, yang mana harus diubah menjadi nilai sudut dengan mengalikan  $180/\pi$ . Sedangkan untuk menghitung jarak antara kedua point menggunakan rumus Haversine. Yang pertama dilakukan yaitu mengubah titik koordinat dari sudut menjadi radian dengan cara mengalikan dengan  $\pi/180$ . Berikut rumus haversine. Pada rumus berikut  $\varphi$  melambangkan *latitude* sedangkan  $\lambda$  melambangkan *longitude*. Angka 1 dan 2 melambangkan titik koordinat kesatu dan kedua.

$$\text{hav}(\theta) = \text{hav}(\varphi_2 - \varphi_1) + \cos(\varphi_1)\cos(\varphi_2)\text{hav}(\lambda_2 - \lambda_1) \quad (5)$$

Rumus haversine sendiri membutuhkan fungsi haversine untuk mendapatkan nilai jarak antara kedua titik koordinat. Berikut merupakan fungsi haversine.

$$hav(\theta) = \sin^2(\theta/2) \quad (6)$$

Untuk mendapatkan jarak antar koordinat, harus tahu terlebih dahulu berapa radius bumi. Radius bumi diambil dari data NASA yaitu 6371 KM. Jarak dalam KM didapat dengan mengalikan radius bumi dengan  $hav(\theta)$ .

#### D. Pengujian

Pada tahap ini dilakukan proses pengujian untuk mendapatkan *error*. Dari *error* tersebut akan diperbaiki hingga mendapatkan *error* yang paling minimal. Proses ini dilakukan terus menerus sampai prototipe mampu bergerak sesuai yang diinginkan. Pengujian yang dilakukan akan diawali dari dok kapal, yang nantinya akan melewati rintangan bola-bola pertama, kedua, dan ketiga. Setelah itu kapal akan melewati objek mangrove dan objek ikan yang nantinya akan diambil gambarnya. Jalur akan diakhiri kembali ke dok kapal.



Gbr. 10 Rute perjalanan yang harus di lewati kapal

Dari jalur tersebut, akan diambil titik-titik yang akan mengarahkan untuk mengikuti jalur yang sudah dipilih. Setiap titik akan dihitung waktu dan keberhasilan menuju tujuan.



Gbr. 11 Titik koordinat yang dilalui kapal

TABEL II  
 TITIK KOORDINAT YANG AKAN DILALUI KAPAL

Titik	Lat	Long	Jarak dengan titik sebelumnya (m)
1	-7.31383440632967	112.7259142877634	0
2	-7.313859060487192	112.72590284156	3.01
3	-7.313902826003977	112.7258922805648	5
4	-7.313960236383254	112.7258327839155	9.1
5	-7.313993155392319	112.7258542088126	4.35
6	-7.3140363229925	112.7259414452276	10.75
7	-7.314004868686945	112.7259634499226	12.1
8	-7.313953323696158	112.7259875100751	6.3
9	-7.313860421984236	112.7260040889943	10.49
10	-7.313875698971198	112.7259408688954	7.17

### III. HASIL DAN PEMBAHASAN

#### A. Implementasi

1) *Implementasi Heading Kompas*: Nilai yang didapat dari kompas adalah x, y, dan z. Apabila nilai x, y, dan z sudah didapatkan, maka digunakan rumus *heading* untuk mendapatkan nilai *heading* bumi mulai dari -180 ~ 180. Berikut hasil pemrograman untuk mendapatkan nilai *heading* dari sensor kompas HMC883L.

```

16 // Return XYZ readings
17 x = compass.getX();
18 y = compass.getY();
19 z = compass.getZ();
20
21 int declination = 0.41;
22 int heading = (atan2(y,x) + declination) * 180/PI;
23
24 Serial.println(heading);
--
    
```

2) *Implementasi Jarak Dua Titik Koordinat*: Pada proses ini bertujuan untuk menghitung jarak antara dua titik koordinat. Pada sistem ini, prototipe kapal akan selalu memperbarui nilai koordinat setiap satu detik sekali. Prototipe kapal juga akan selalu memperbarui nilai dari kompas selama satu detik sekali. Setelah itu, dihitung jarak antara koordinat kapal dengan koordinat tujuan. Dari jarak tersebut maka dapat dihitung apakah posisi kapal sudah mendekati dengan koordinat tujuan.

```

4- def hitung_jarak(lat1, lon1, lat2, lon2):
5     lon1, lat1, lon2, lat2 = map(radians, [lon1, lat1, lon2, lat2])
6
7     dlon = lon2 - lon1
8     dlat = lat2 - lat1
9     a = sin(dlat/2)**2 + cos(lat1) * cos(lat2) * sin(dlon/2)**2
10    c = 2 * asin(sqrt(a))
11    r = 6371
12    return c * r
--
    
```

3) *Implementasi Bearing Dua Titik Koordinat*: Hal yang harus dipertimbangkan dalam sistem navigasi yaitu arah mata angin. Untuk menuju ke koordinat tujuan, maka dibutuhkan arah untuk menuju ke posisi tersebut. Maka dihitung *bearing* dari koordinat kapal dengan koordinat tujuan. Dari nilai kompas kapal akan diarahkan menuju arah tujuan hingga nilai kompas mendekati atau sama dengan nilai *bearing*. Kapal akan bergerak menuju ke arah koordinat hingga jarak mendekati 0 dengan koordinat tujuan. Implementasi *bearing* menggunakan rumus.

```

def calc_bearing(lat1, long1, lat2, long2):
    lat1 = radians(lat1)
    long1 = radians(long1)
    lat2 = radians(lat2)
    long2 = radians(long2)
    # Calculate the bearing
    bearing = atan2(
        sin(long2 - long1) * cos(lat2),
        cos(lat1) * sin(lat2) - sin(lat1) * cos(lat2) * cos(long2 - long1)
    )
    # Convert the bearing to degrees
    bearing = degrees(bearing)
    return bearing
    
```

**B. Pengujian**

Pengujian sensor HMC883L digunakan untuk mengetahui kualitas nilai sensor yang dihasilkan. Pengujian dilakukan secara statis mengarahkan sensor ke arah tertentu.

TABEL III  
HASIL PENGUJIAN SENSOR HMC883L

Iterasi	Heading (°)	Error (°)
1	41	0
2	41	0
3	42	1
4	40	2
5	41	1

6	42	1
7	40	2
8	38	2
9	40	2
10	40	0

Dari hasil pengujian diatas dapat dilihat bahwa sensor HMC883L memiliki fluktuasi nilai yang tidak menentu. Pada pengujian utara, nilai selisih tertinggi yaitu 2° dengan rata-rata eror 1,1°. Walaupun terjadi fluktuasi nilai, sensor HMC883L sudah mumpuni untuk bisa membedakan arah mata angin dengan akurat dengan eror yang relatif kecil.

TABEL IV  
HASIL PENGUJIAN GPS NEO 6M

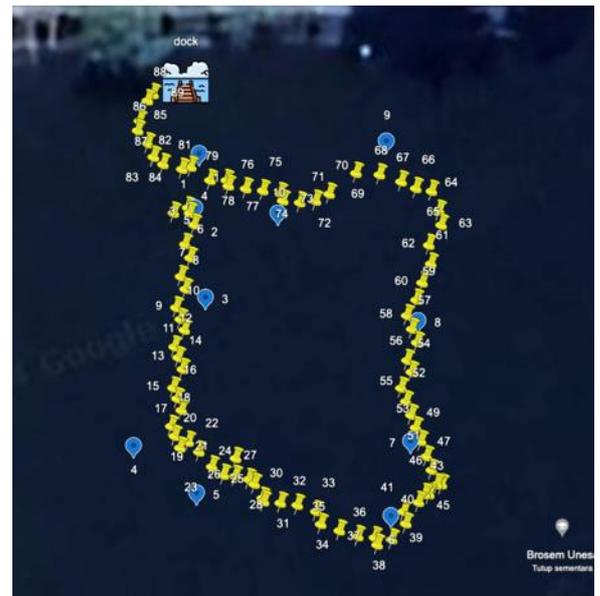
Waktu	Latitude	Longitude	Jarak (m)
21/2/2024 09:08:34	-7.316804	112.725204	0
21/2/2024 09:08:35	-7.316805	112.725212	0.8
21/2/2024 09:08:36	-7.316807	112.725227	1.6
21/2/2024 09:08:37	-7.316806	112.725235	0.8
21/2/2024 09:08:38	-7.316803	112.725243	0.9
21/2/2024 09:08:39	-7.316802	112.725243	0.1
21/2/2024 09:08:40	-7.316802	112.725243	0
21/2/2024 09:08:41	-7.316804	112.725250	0.8
21/2/2024 09:08:42	-7.316804	112.725250	0
21/2/2024 09:08:43	-7.316802	112.725250	0.2
Rata-rata	-7.316802	112.7252357	0.52

Dari hasil pengujian sensor GPS Neo 6m, terdapat fluktuatif nilai, dengan rata rata *latitude* yaitu -7.316802 dan *longitude* yaitu 112.7252357. Sedangkan rata-rata fluktuasi jarak yaitu 0.52 dan nilai jarak tertinggi yaitu 1.6. Dari nilai eror tersebut, sensor GPS Neo 6M mampu memberikan keakuratan kurang dari 1.6 m, sehingga sensor ini sudah cukup baik untuk digunakan sebagai sistem navigasi.

Pengujian sistem navigasi dilakukan dengan melakukan *trial* kapal secara langsung yang diiterasi dengan lima kali. Pada proses ini akan dilihat gerak sensor GPS dan kompas yang akan memengaruhi pergerakan kapal.



Gbr. 12 Hasil Pengujian Sistem Navigasi Iterasi Pertama



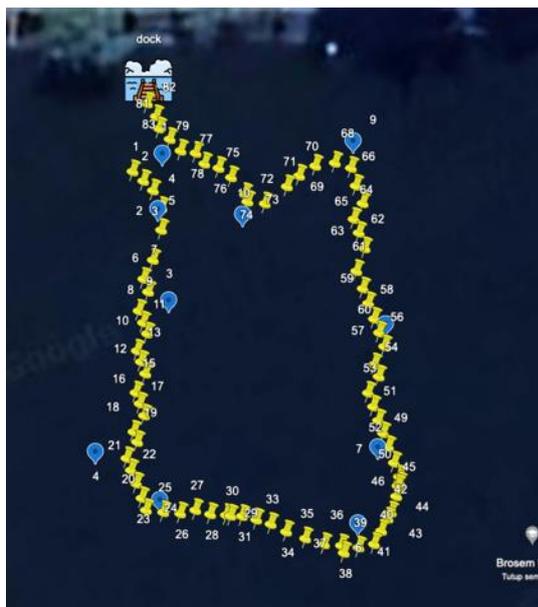
Gbr. 14 Hasil Pengujian Sistem Navigasi Iterasi Ketiga



Gbr. 13 Hasil Pengujian Sistem Navigasi Iterasi Kedua



Gbr. 15 Hasil Pengujian Sistem Navigasi Iterasi Keempat



Gbr. 16 Hasil Pengujian Sistem Navigasi Iterasi Kelima

TABEL V  
HASIL PENGUJIAN SISTEM NAVIGASI

Iterasi	1	2	3	4	5	6	7	8	9	10	Hasil
1	✓	✓	✓	×	✓	✓	✓	✓	×	✓	80%
2	✓	✓	✓	✓	✓	✓	✓	✓	×	✓	90%
3	✓	✓	✓	×	✓	✓	✓	✓	✓	✓	90%
4	✓	✓	✓	×	✓	✓	✓	✓	×	✓	80%
5	✓	✓	✓	×	✓	✓	✓	✓	✓	✓	90%

Dapat dilihat titik GPS kapal selalu berpindah pindah. Hal itu terjadi karena GPS Neo 6M kurang memberikan titik yang akurat dengan tingkat eror sekitar 0.52 m. Meskipun titik GPS selalu berubah, kapal masih mampu melakukan pergerakan menuju titik poin dengan benar, walaupun terdapat beberapa titik yang tidak tepat titik tujuannya. Pada pengujian ini didapatkan keberhasilan prototipe kapal sebesar 86%.

TABEL VI  
HASIL PENGUJIAN KOMPAS

Titik	Heading Kompas Tujuan (°)	Heading Kompas Kapal (°)	Error (°)
1	0	0	0
2	-155	-156	1
3	-166	-163	3
4	-134	-138	4
5	146	147	1
6	116	118	2
7	34	37	3
8	24	22	2
9	10	7	3
10	-103	-102	1

Pada tabel diatas, dapat dilihat bahwa sensor kompas tidak memberikan nilai yang akurat berdasarkan perhitungan kompas tujuan. Untuk mendapatkan nilai *heading* kompas tujuan dilakukan dengan menghitung titik koordinat sekarang dengan titik koordinat sebelumnya. Sedangkan *heading* kompas kapal didapat dari rata-rata dari titik koordinat kapal sekarang dengan titik koordinat tujuan sebelumnya. Didapatkan rata-rata eror selama satu putaran kapal yaitu 2 derajat.

#### IV. KESIMPULAN

Pada penelitian mengenai sistem navigasi *autonomous* pada prototipe kapal dengan membuat sensor GPS dan kompas mampu membuat kapal mampu menavigasi kapal secara mandiri dengan mengikuti *waypoint* yang telah ditentukan. Hal itu dapat dilihat pada proses *trial* yang dilakukan iterasi sebanyak lima kali. Prototipe kapal mampu bergerak sesuai titik dengan benar dengan tingkat keberhasilan 86%. Meskipun demikian, sistem ini masih belum begitu akurat. Dikarenakan sensor GPS Neo 6M pada saat pengujian sensor memberikan nilai yang kurang akurat dengan nilai rata-rata eror 0.52 meter dan eror paling tinggi 1.6 meter. Apabila diatas tersebut, GPS Neo 6m sudah cukup baik untuk digunakan pada sebuah sistem navigasi. Begitu pula dengan sensor HMC5883L, dengan tingkat rata-rata eror 1,1° dengan eror tertinggi yaitu 2°. Hal ini menyebabkan kapal bergerak sedikit *random* kekanan dan kekiri apabila melihat pengujian kapal yang dilakukan iterasi selama lima kali, dimana masih terdapat beberapa titik yang tidak dapat dicapai oleh protitipe kapal.

#### V. SARAN

Berdasarkan hasil penelitian yang dilakukan, terdapat beberapa saran yang dapat dipertimbangkan untuk kepentingan penelitian atau pengembangan selanjutnya, yaitu:

1. Menggunakan sensor yang lebih akurat daripada menggunakan GPS Neo 6M dan HMC5883L untuk meningkatkan hasil keberhasilan dalam mengikuti *waypoint* yang sudah ditentukan.
2. Menambah *module*/sensor yang lain, seperti IMU, LiDAR, ultrasonik, ataupun kamera sebagai parameter lain yang menunjang sebuah sistem navigasi yang lengkap untuk mengurangi eror.
3. Menggunakan metode sistem kontrol seperti PID, Fuzzy ataupun metode yang lain untuk mengurangi sensor *noise*/eror yang diberikan supaya lebih akurat.

#### VI. UCAPAN TERIMA KASIH

Penulis dengan tulus mengungkapkan rasa terima kasih yang besar kepada Tuhan Yang Maha Esa atas semua anugerah, kasih sayang, dan bantuan-Nya yang telah memungkinkan penyelesaian proyek dan artikel ilmiah ini dengan sukses. Terima kasih juga disampaikan kepada orang tua yang selalu memberikan doa, semangat, dan dukungan; dosen pembimbing skripsi yang memberikan masukan konstruktif; serta sahabat dan teman yang memberikan dorongan dan dukungan selama penelitian dilakukan. Selain itu, terima kasih kepada diri sendiri

atas kerja keras dan upaya untuk menjaga kesehatan demi mencapai tujuan yang diinginkan.

## VII. REFERENSI

- [1] H. Kiswanto *dkk.*, “Pengenalan Autonomous-Vessel untuk Siswa SMA 14 Semarang Melalui Kegiatan Webinar,” *SOROT J. Pengabd. Kpd. Masy.*, vol. 2, no. 2, hal. 82–87, 2023, doi: 10.32699/sorot.v2i2.5188.
- [2] A. H. Muhammad *dkk.*, *Kontes Kapal Cepat Tak Berawak Nasional*. Jakarta Selatan: Balai Pengembangan Talenta Indonesia, 2023.
- [3] A. M. White, W. P. Gardner, A. A. Borsa, dan D. F. Argus, “A Review of GNSS / GPS in Hydrogeodesy : Hydrologic Loading Applications and Their Implications for Water Resource,” hal. 28, 2022, doi: 10.1029/2022WR032078.
- [4] P. Kanani dan M. Padole, “Real-time Location Tracker for Critical Health Patient using Arduino , GPS Neo6m and GSM Sim800L in Health Care,” no. *Iciccs*, hal. 242–249, 2020, doi: <https://doi.org/10.1109/ICICCS48265.2020.9121128>.
- [5] B. Livada, V. Saša, D. Radić, T. Unkašević, dan Z. Banjac, “Digital Magnetic Compass Integration with Stationary, Land-Based Electro-Optical Multi-Sensor Surveillance System,” 2019, doi: 10.3390/s19194331.
- [6] M. Raja, A. Agarwal, dan A. Gautam, “Integrated Inertial Navigation System with camera-based system using LabVIEW Software,” vol. 8, no. 1, hal. 59–64, 2020.
- [7] D. Hercog, T. Lerher, M. Truntič, dan O. Težak, “Design and Implementation of ESP32-Based IoT Devices,” *Sensors*, vol. 23, no. 15, 2023, doi: 10.3390/s23156739.
- [8] S. Monk, *Raspberry Pi Cookbook Software and Hardware Problems & Solutions*. Gravenstein Highway North: O’Reilly Media, 2011.
- [9] X. Yin, K. Xu, H. Liang, G. Zhang, X. Peng, dan Q. Zhang, “Hardware Design of Flight Control System and Flight Experiments of Small-Scale Unmanned Aerial Vehicle Flight control system,” hal. 4925–4929, 2020.
- [10] A. E. Amoran, A. S. Oluwole, E. O. Fagorola, dan R. S. Diarah, “Home automated system using Bluetooth and an android application,” *Sci. African*, vol. 11, hal. e00711, 2021, doi: 10.1016/j.sciaf.2021.e00711.
- [11] M. Abdullah-Al-Noman, A. N. Eva, T. B. Yeahyea, dan R. Khan, “Computer Vision-based Robotic Arm for Object Color, Shape, and Size Detection,” *J. Robot. Control*, vol. 3, no. 2, hal. 180–186, 2022, doi: 10.18196/jrc.v3i2.13906.