

PERANCANGAN DAN IMPLEMENTASI AUTONOMOUS SURFACE VEHICLE BERBASIS ESP32-S3 DENGAN NAVIGASI WAYPOINT DAN MONITORING REAL-TIME MENGGUNAKAN MQTT DAN NODE-RED

Indah Siti Khotimah; Fajar Suryawan, S.T., M.Eng.Sc., Ph.D

Program Studi Teknik Elektro, Fakultas Teknik, Universitas Muhammadiyah Surakarta

Abstrak

Perkembangan teknologi sistem kendali otomatis mendorong pemanfaatan kendaraan tanpa awak seperti *Autonomous Surface Vehicle* (ASV) untuk pemantauan perairan secara efisien. Pengoperasian kapal secara manual memiliki keterbatasan pada konsistensi penelusuran jalur navigasi dan akurasi pemantauan posisi *real-time*. Penelitian ini bertujuan merancang dan mengimplementasikan sistem ASV berbasis mikrokontroler ESP32-S3 yang mampu bernavigasi secara otonom melintasi 12 titik *waypoint* dengan lima variasi pola geometri (*SQUARE*, *CIRCLE*, *STAR*, *LOVE*, *EIGHT*) serta mentransmisikan data telemetri secara *real-time*. Metode yang digunakan meliputi perancangan perangkat keras dengan pemanfaatan modul GPS NEO-M8N untuk akuisisi koordinat, sensor MPU9250 (GY-91) sebagai kompas digital terfilter *Low-Pass Filter* (LPF), pendorong motor BLDC, serta kendali *rudder* berbasis kontroler PID ($K_p = 2.50$, $K_i = 0.15$, $K_d = 0.40$). Komunikasi data dua arah memanfaatkan protokol MQTT berbasis Wi-Fi lokal yang diintegrasikan dengan platform *Node-RED* di stasiun darat sebagai antarmuka pemantauan interaktif. Hasil pengujian lapangan di perairan Danau UMS menunjukkan bahwa sistem ASV berhasil mengeksekusi navigasi *waypoint* secara otonom dan rata-rata *error radius* kedatangan sebesar 1,53 meter berada di bawah ambang batas 3,5 meter, serta berhasil merealisasikan fitur *Return to Start* (RTS). Transmisi telemetri pada topik MQTT berjalan sangat responsif dan didukung oleh fungsi *fail-safe reconnectMQTT()* yang mampu menjaga keamanan navigasi mandiri saat terjadi pemutusan jaringan Wi-Fi.

Kata Kunci: *Autonomous Surface Vehicle, ESP32-S3, Kontroler PID, MQTT, Navigasi Waypoint, Return to Start.*

Abstract

The development of automatic control system technology encourages the use of unmanned vehicles such as *Autonomous Surface Vehicles* (ASV) for efficient water monitoring. Manual operation of ships has limitations in the consistency of navigation path tracking and the accuracy of *real-time* position monitoring. This research aims to design and implement an ASV system based on the ESP32-S3 microcontroller that can autonomously navigate across 12 *waypoint* points with five variations of geometric patterns (*SQUARE*, *CIRCLE*, *STAR*, *LOVE*, *EIGHT*) and transmit telemetry data in *real-time*. The methods used include hardware design utilizing the NEO-M8N GPS module for coordinate acquisition, the MPU9250 (GY-91) sensor as a digital compass filtered with a *Low-Pass Filter* (LPF), BLDC motor drivers, and rudder control based on a PID controller ($K_p=2.50$, $K_i=0.15$, $K_d=0.40$). Bidirectional data communication utilizes the MQTT protocol based on local Wi-Fi, integrated with the *Node-RED* platform at the ground station as an interactive monitoring interface. Field test results in the waters of UMS Lake show that the ASV system successfully executed *waypoint* navigation autonomously, with an average arrival radius error of 1.53 meters, which is below the threshold of 3.5 meters, and successfully realized the *Return to Start* (RTS) feature. Telemetry transmission on the MQTT topic is very responsive and is supported by the *fail-safe* function *reconnectMQTT()* which can maintain the security of autonomous navigation in the event of a Wi-Fi network disconnection.

Keywords: *Autonomous Surface Vehicle, ESP32-S3, MQTT, PID Controller, Return to Start, Waypoint Navigation.*

1. PENDAHULUAN

Perkembangan teknologi sistem kendali otomatis telah mendorong pemanfaatan kendaraan tanpa awak dalam berbagai bidang, termasuk transportasi, pemantauan lingkungan, dan penelitian perairan. Berdasarkan tingkat otonomi kendaraan permukaan tanpa awak, sistem yang dikembangkan dalam penelitian ini termasuk ke dalam kategori ASV. Pada kategori ASV, kapal mampu menjalankan navigasi secara otomatis berdasarkan *waypoint* yang telah ditentukan, sementara operator tetap melakukan pengawasan terhadap kondisi dan pergerakan kapal melalui sistem *monitoring*. Pendekatan ini dipilih karena mampu menggabungkan keunggulan navigasi otomatis dengan kemampuan pengawasan manusia sehingga meningkatkan keamanan dan keandalan sistem selama proses operasi. Pengoperasian kapal secara manual memiliki beberapa keterbatasan, seperti ketergantungan terhadap kemampuan operator, potensi kesalahan manusia dalam menentukan arah navigasi, serta keterbatasan dalam melakukan pemantauan posisi secara terus-menerus. Oleh karena itu, diperlukan suatu sistem yang mampu melakukan navigasi secara otomatis dengan tingkat akurasi yang baik serta menyediakan informasi posisi kapal secara *real-time*.

Pada penelitian ini dirancang dan diimplementasikan sistem ASV berbasis ESP32-S3 yang memanfaatkan GPS NEO-M8N untuk memperoleh data posisi dan sensor GY-91 untuk memperoleh data *heading* sebagai acuan navigasi. Sistem penggerak menggunakan motor BLDC 3100 KV yang dikendalikan melalui ESC, sedangkan pengendalian arah dilakukan menggunakan motor servo DS3225 yang terhubung dengan *rudder*. Selain itu, sistem dilengkapi dengan komunikasi MQTT melalui jaringan Wi-Fi sehingga data telemetry dapat dikirim dan dipantau secara *real-time* menggunakan *Node-RED* pada laptop.

Untuk *unmanned surface vehicle* (USV), diperlukan perangkat propulsi atau perangkat penggerak. Salah satu jenis propulsi adalah baling-baling *pitch* tetap menggunakan motor listrik tipe BLDC dan *rudder* sebagai perangkat kemudi. Penerapan sistem *autopilot* pada USV *monohull* dapat menjadikannya kendaraan yang andal dan dapat beroperasi secara otomatis sesuai dengan program yang telah ditanamkan di dalamnya sehingga dapat digunakan untuk beberapa kebutuhan khusus (Rahmasari et al., 2021). Dengan kemajuan dalam otomatisasi dan sistem cerdas, ASV telah menjadi fokus utama dalam transportasi modern dan robotika. Kendaraan-kendaraan ini dapat bernavigasi dan beroperasi tanpa intervensi manusia dengan menggunakan sensor, pengontrol, dan algoritma pengambilan keputusan (Srilatha & Kumar, 2026). Motor BLDC dapat dipengaruhi oleh kebisingan pengukuran dan *manufacturing uncertainty*. Untuk mendapatkan *close loop* yang dapat melacak referensi yang diinginkan, jelas bahwa kita memerlukan pengontrol untuk menjamin stabilitas sistem dan kinerja yang diperlukan (Hachem et al., 2023).

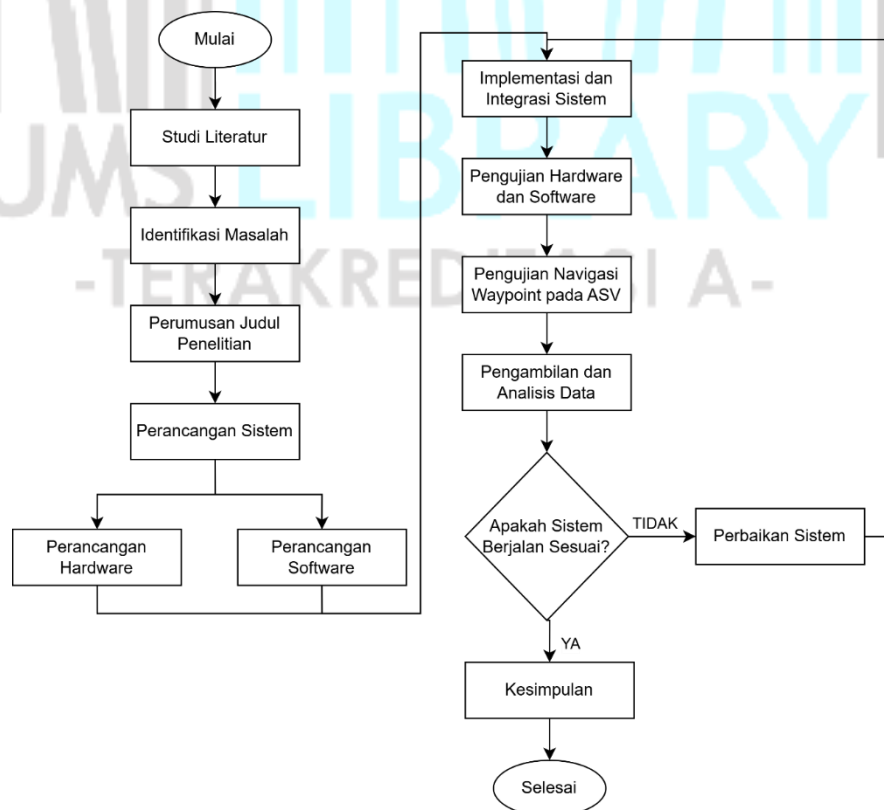
Desain kapal menggunakan bahan plastik PLA (*Polylactic Acid*) yang membantu meminimalkan dampak lingkungan negatif (Vũ et al., 2024). Untuk berhasil menyelesaikan misi pemantauan air,

algoritma perencanaan dan kontrol USV selalu menjadi kunci untuk masalah tersebut. Di sisi lain, metode kontrol mengikuti titik *waypoint* membimbing USV untuk berhasil mengunjungi titik-titik tugas (Zhao et al., 2025).

Client adalah *software* yang berinteraksi dengan *Broker*. Ini dapat bertindak sebagai *Subscriber* (*Subscribe*) atau *Sender* (*Publish*). *Client* biasanya adalah perangkat yang dapat menerbitkan data sensor dan juga mampu menerima informasi konfigurasi atau memproses perintah kontrol. MQTT adalah protokol komunikasi dua arah. Ini menyediakan baik berbagi data maupun manajemen dan kontrol perangkat. Desain MQTT telah ditekankan untuk data yang dikirimkan secara sering dan berukuran kecil (Nemlaha & St̃, 2022). Kontrol PID menghitung dengan cepat dan mudah diterapkan dalam rekayasa; ini adalah metode utama yang saat ini digunakan untuk kontrol kemudi USV. Mengoptimalkan tiga parameter K_p , K_i , dan K_d adalah kunci dalam desain pengendali PID. Pengendalian kemudi USV berkecepatan tinggi memerlukan *overshoot* kecil dan waktu *settling* yang singkat (Mingzhi & Lanjun, 2023).

2. METODE

Metode penelitian pada perancangan dan implementasi ASV berbasis ESP32-S3 dengan navigasi *waypoint* dan monitoring *real-time* menggunakan MQTT dan *Node-Red* dilakukan melalui beberapa tahapan terstruktur. Tahapan penelitian disusun secara sistematis sesuai dengan diagram alir yang ditunjukkan pada Gambar 1.



Gambar 1. Diagram Alir Metode Pembuatan Alat

Berdasarkan Gambar 1, metode penelitian yang dilakukan terdiri dari beberapa tahapan, yaitu sebagai berikut:

a. Studi Literatur

Penelitian diawali dengan studi literatur untuk memperoleh landasan teori yang mendukung perancangan dan implementasi ASV berbasis ESP32-S3. Sumber yang dikaji meliputi jurnal, prosiding, dan buku tentang ASV, navigasi *waypoint*, sensor GPS, kompas digital, serta protokol MQTT dan *Node-RED*. Tujuan studi ini adalah memahami metode, teknologi, dan komponen yang relevan. Hasilnya menjadi dasar untuk merancang sistem yang mampu melakukan navigasi *waypoint* dan pemantauan *real-time*.

b. Perancangan dan pembuatan alat

Perancangan dan perakitan hardware mencakup seluruh subsistem mekanik dan elektronik pada ASV. Perangkat keras yang digunakan adalah ESP32-S3 sebagai pengendali utama, GPS NEO-M8N sebagai sensor posisi, dan GY-91 sebagai sensor orientasi. Sistem propulsi menggunakan motor BLDC dan ESC, sedangkan sistem kemudi menggunakan servo dan *rudder*. Catu daya disediakan oleh baterai dan buck converter. Tujuan tahap ini adalah menghasilkan perangkat keras yang terintegrasi dan siap operasional.

c. Pengujian Alat

Setelah proses perakitan perangkat keras selesai dilakukan, tahap selanjutnya adalah pengujian dan evaluasi perangkat keras. Pengujian dan evaluasi hardware dilakukan setelah perakitan selesai untuk memastikan setiap komponen berfungsi dengan baik. Komponen yang diuji meliputi ESP32-S3, GPS NEO-M8N, sensor GY-91, motor BLDC, ESC, servo *rudder*, dan sistem catu daya. Evaluasi dilakukan berdasarkan hasil pengujian untuk mengetahui kinerja dan keandalan.

d. Analisis Hasil

Arsitektur perangkat lunak pada ASV dirancang secara modular dan terstruktur. Arsitektur ini memiliki tiga lapisan utama, yaitu pemrosesan sensor GPS dan kompas digital, navigasi dengan algoritma *waypoint*, dan aktuasi kendali motor BLDC serta servo *rudder*. Komunikasi antar subsistem menggunakan protokol MQTT yang ringan dan efisien. Data posisi, *heading*, dan status misi dikirim ke *Node-RED* untuk *monitoring real-time*, sehingga operator dapat melakukan supervisi langsung.

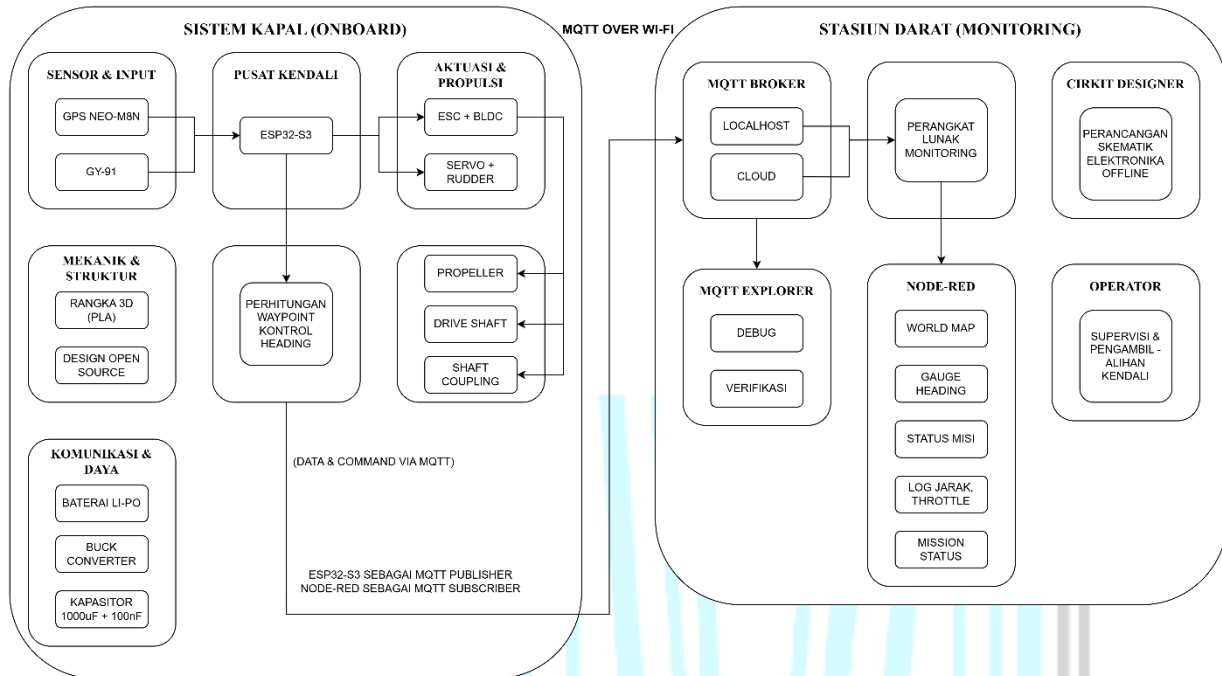
e. Pembuatan Laporan

Integrasi perangkat keras dan perangkat lunak menggabungkan seluruh komponen sistem menjadi satu kesatuan yang berfungsi. Perangkat keras ESP32-S3, GPS, GY-91, motor BLDC, dan servo telah diuji dan dinyatakan layak. Perangkat lunak memproses data sensor, menjalankan navigasi *waypoint*, serta mengendalikan motor dan servo. Keduanya dihubungkan dengan protokol MQTT untuk komunikasi data yang efisien. Hasil integrasi memungkinkan monitoring posisi, *heading*, dan status

misal secara *real-time* melalui *Node-RED*.

f. Dokumentasi

Tahap akhir adalah dokumentasi seluruh proses penelitian dalam bentuk laporan tugas akhir. Laporan ini disusun sebagai bukti ilmiah yang dapat dipertanggungjawabkan sekaligus sebagai acuan



bagi pengembangan ASV dimasa mendatang.

2.1. Perancangan Hardware

Gambar 2 memperlihatkan blok diagram sistem, terdapat dua bagian utama yaitu sistem kapal (*Onboard*) dan stasiun darat (*Monitoring*).

Gambar 2. Blok Diagram Sistem

Bagian Sistem Kapal (*Onboard*):

a. Sensor & Input

Modul GPS NEO-M8N dan sensor GY-91 merupakan komponen input utama pada sistem kapal. GPS NEO-M8N berfungsi untuk akuisisi data posisi geografis, meliputi koordinat lintang dan bujur, kecepatan, serta informasi waktu yang esensial bagi navigasi. Sementara itu, GY-91 adalah modul *multiaxis* yang mengintegrasikan akselerometer, giroskop, dan magnetometer, yang secara kolektif menyediakan data orientasi, gerakan sudut, serta arah mata angin (*heading*) guna menunjang kestabilan dan kendali arah kapal.

b. Pusat Kendali

ESP32-S3 bertindak sebagai unit pemroses pusat (*central processing unit*) di atas kapal, yang bertanggung jawab atas akuisisi dan pengolahan seluruh data sensor, eksekusi algoritma kendali, pengelolaan protokol komunikasi, serta pembangkitan sinyal kendali bagi subsistem aktuasi, sehingga menjadi inti koordinasi seluruh operasi *onboard*.

c. Mekanik & Struktur

Rangka kapal dicetak menggunakan teknologi pencetakan 3D berbahan dasar PLA yang menawarkan sifat ringan, fleksibilitas desain, dan efisiensi biaya. Desain *open source* ada struktur mekanik ini memungkinkan modifikasi, replikasi, dan pengembangan lebih lanjut oleh komunitas atau peneliti sesuai dengan kebutuhan spesifik misi.

d. Komunikasi & Daya

Sumber energi utama berasal dari baterai tipe Li-Po, yang selanjutnya diregulasi melalui buck converter guna menghasilkan tegangan operasi yang stabil dan sesuai untuk mikrokontroler serta perifer. Untuk memitigasi fluktuasi tegangan akibat perubahan beban dinamis, dipasang kapasitor elektrolit 1000 μ F sebagai penstabil tegangan makro, serta kapasitor keramik 100 nF untuk meredam derau frekuensi tinggi pada jalur suplai daya.

e. Aktuasi & Propulsi

Subsistem propulsi dikendalikan oleh ESC (*Electronic Speed Controller*) yang mengatur kecepatan dan arah putaran motor BLDC berdasarkan sinyal *throttle* dari pusat kendali. Adapun subsistem kemudi diimplementasikan melalui servo yang terhubung secara mekanis dengan *rudder*, di mana perubahan sudut *rudder* dikendalikan secara untuk mengubah haluan kapal sesuai dengan komando navigasi.

f. Propeller

Tenaga putar dari motor BLDC disalurkan menuju baling-baling melalui sebuah *drive shaft* (poros penggerak). Antara poros motor dan *drive shaft* dipasang *shaft coupling* yang berfungsi sebagai penghubung mekanis sekaligus penyesuai terhadap ketidaksejajaran *axial* ataupun *radial*, sehingga transmisi daya putar dapat berlangsung secara efisien dan dengan getaran minimal.

g. Data & Command

Seluruh data telemetri dan perintah kendali dipertukarkan melalui protokol MQTT yang bersifat ringan dan efisien. Pada arsitektur ini, ESP32-S3 dikonfigurasi sebagai MQTT *publisher* yang secara periodik mengirimkan data sensor dan status kapal ke *broker*. Sebaliknya, *Node-RED* di sisi stasiun darat bertindak sebagai *subscriber* untuk menerima data tersebut, sekaligus bertindak sebagai *publisher* bagi perintah kendali yang dikirimkan kembali menuju kapal, sehingga tercipta saluran komunikasi dua arah yang bersifat *real-time*.

Bagian Stasiun Darat (*Monitoring*):

a. MQTT Broker

Broker MQTT berfungsi sebagai *server* perantara yang memediasi seluruh pertukaran pesan antara *publisher* dan *subscriber* dalam arsitektur komunikasi. Lingkungan eksekusinya dapat dijalankan secara lokal (*localhost*) untuk keperluan pengembangan dan pengujian dengan latensi

rendah, maupun diimplementasikan pada infrastruktur *cloud* guna memungkinkan akses jarak jauh dan pemantauan dari lokasi geografis mana pun.

b. MQTT Explorer

MQTT *Explorer* dimanfaatkan sebagai klien visual untuk keperluan *debugging* dan verifikasi sistem. Melalui antarmukanya, pengembang dapat menginspeksi secara langsung seluruh topik yang tersedia beserta *payload* yang dikirimkan, sehingga memudahkan identifikasi kesalahan komunikasi dan validasi integritas data secara *real-time*.

c. Node-Red

Node-RED merupakan lingkungan pemrograman berbasis aliran (*flow-based programming*) yang berperan sebagai dasbor utama dan pusat komando di stasiun darat. Fungsionalitasnya mencakup visualisasi peta dunia (*world map*) untuk menampilkan posisi dan lintasan kapal secara spasial berdasarkan data GPS, serta tampilan instrumen *heading* (*gauge heading*) yang menyajikan informasi arah hadap kapal secara kontinu guna mendukung kesadaran situasional operator.

d. Operator

Operator merupakan unsur manusia yang bertugas melaksanakan supervisi dan pengambilalihan kendali (*manual override*) apabila diperlukan. Dengan memantau seluruh metrik dan status yang terpapar pada dasbor Node-RED, operator dapat mengambil keputusan strategis serta mengintervensi sistem kendali otomatis untuk menjamin keselamatan dan keberhasilan misi.

e. Status Misi

Panel status misi menyajikan ringkasan komprehensif mengenai kondisi operasional kapal. Informasi yang ditampilkan meliputi *log* jarak tempuh dan *level throttle* sebagai rekam jejak kinerja propulsi, serta indikator status misi (*mission status*) yang mendeskripsikan fase operasi terkini, seperti navigasi, siaga, kembali ke basis, atau kondisi darurat, guna memberikan gambaran situasional yang jelas dan terukur bagi operator.

2.1. Perancangan Alat dan Bahan

Perancangan alat dan bahan pada penelitian ini diawali dengan pemilihan dan persiapan seluruh komponen utama sistem ASV. Komponen tersebut meliputi mikrokontroler ESP32-S3 N16R8 sebagai pusat kendali, GPS NEO-M8N untuk penentuan posisi, sensor GY-91 sebagai kompas digital, motor BLDC 2845 3100KV dan *propeller* 42 mm sebagai sistem propulsi, servo DS3225 25KG serta *rudder* sebagai sistem kemudi, serta ESC untuk pengaturan kecepatan motor. Sumber daya menggunakan baterai Li-Po 3000 mAh 11,1 V yang diturunkan tegangannya melalui buck converter LM2596, dengan tambahan kapasitor 1000 μ F dan 100 nF untuk stabilisasi tegangan. Untuk mendukung komunikasi dan monitoring, digunakan protokol MQTT dengan bantuan MQTT *Broker*, MQTT *Explorer*, dan Node-RED, serta *charger* SkyRC iMAX B6AC V2 untuk pengisian baterai.

Daftar lengkap komponen perangkat keras dan perangkat lunak yang digunakan dalam sistem ini dapat dilihat pada Tabel 1.1.

Tabel 1.1. Alat dan Bahan

Alat dan Bahan	Jumlah
Motor BLDC 2845 3100KV	1
Motor Servo DS3225 25KG	1
ESP32-S3 N16R8	1
Baterai Li-Po 11,1V 35C 2800mAh	1
Buck converter LM2596 Adjustable	1
Propeller CW/CCW 42mm	1
Drive Shaft 10cm Ø3,175mm	1
Shaft Coupling 20mm x 9mm	1
Ø3,175mm	
GY-91 (MPU9250 + BMP280)	1
GPS NEO-M8N u-blox	1
Kapasitor	4
ESC 30A	1
Rudder	1
Charger SkyRC iMAX B6AC V2	1
Connector XT-60 male-female	2
Connector Waterproof 3 pins & 7 pins	2
Box Waterproof	1
Kabel	Secukupnya
Molex 3 pins & 2 pins	5
Pinhead male-female	Secukupnya
Baut dan mur Ø3,175mm	Secukupnya
PVC 5mm	1
Acrylic 3mm	1
PCB	1
Laptop	1

Selain alat dan bahan yang tercantum, peneliti menggunakan perangkat lunak pendukung yang meliputi MQTT *Broker* sebagai perantara komunikasi data antara ESP32-S3 dan sistem *monitoring*, MQTT *Explorer* untuk memantau dan memverifikasi lalu lintas data pada *broker*, serta *Node-RED* sebagai platform visualisasi data *real-time*. MQTT *Broker* menerima data posisi GPS, *heading*, dan

status misi dari kapal, kemudian meneruskannya ke *Node-RED* untuk ditampilkan dalam bentuk *dashboard* berbasis *web*, termasuk peta digital posisi kapal. *MQTT Explorer* digunakan selama proses *debugging* untuk memastikan bahwa seluruh data terkirim dengan benar sebelum ditampilkan. Dengan perangkat lunak ini, operator dapat melakukan supervisi terhadap navigasi waypoint secara langsung tanpa harus mengendalikan kapal secara manual, sehingga mendukung konsep *autonomy* secara efektif.

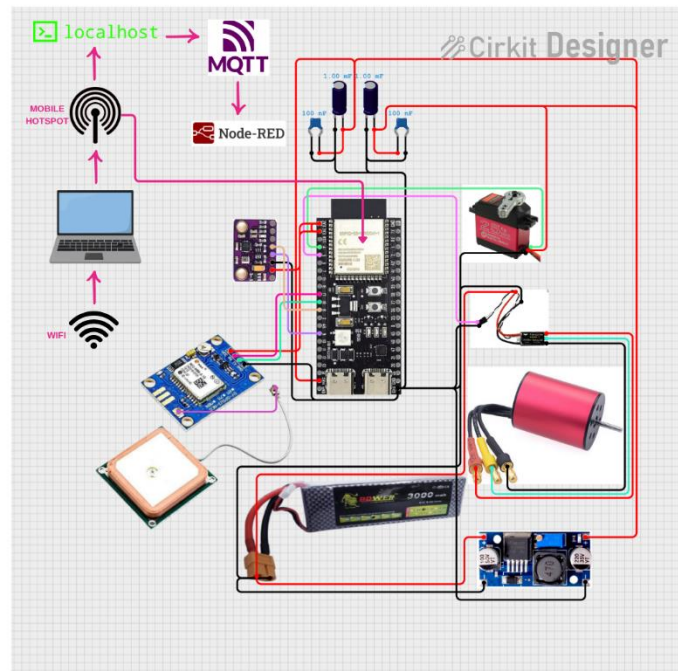
2.2. Perancangan Sistem

Perancangan sistem ASV dibagi menjadi dua subsistem utama, yaitu sistem di atas kapal (*onboard*) dan stasiun darat (*monitoring*). Pada sistem kapal, sensor GPS NEO-M8N dan GY-91 memperoleh data posisi serta *heading* yang diproses oleh ESP32-S3 sebagai pusat kendali. ESP32-S3 menjalankan komputasi navigasi *waypoint* dan kontrol *heading*, kemudian mengirimkan sinyal ke aktuator ESC motor BLDC, servo *rudder* serta sistem propulsi *propeller*, *drive shaft*, *coupling*. Seluruh komponen ini ditempatkan pada rangka cetak 3D berbahan PLA dari desain sumber terbuka, dan didukung oleh catu daya baterai Li-Po, *buck converter* LM2596, serta kapasitor *filter* 1000 μ F dan 100 nF. Data posisi, *heading*, dan status misi dikirim dari ESP32-S3 sebagai *MQTT publisher* menuju *MQTT broker* berjalan di *localhost* atau *cloud* melalui protokol MQTT. Stasiun darat menerima data tersebut dan memvisualisasikannya menggunakan *Node-RED* yang terintegrasi dengan *world map*, *gauge heading*, status misi, serta *log* jarak dan *throttle*. Selain itu, *MQTT Explorer* digunakan untuk *debugging* dan verifikasi, sementara *Circuit Designer* berperan dalam perancangan skematik elektronika secara offline. Operator dapat melakukan supervisi dan mengambil alih kendali kapal melalui *dashboard Node-RED*, sehingga terwujud konsep *autonomy* secara terstruktur dan *real-time*.

2.2.1. Perancangan Elektronika

Perancangan elektronika pada ASV mencakup pemilihan dan konfigurasi seluruh komponen kelistrikan yang terintegrasi dengan ESP32-S3 sebagai pusat kendali. Sensor GPS NEO-M8N digunakan untuk akuisisi posisi, sedangkan GY-91 berfungsi sebagai kompas digital untuk menentukan *heading* kapal. Kedua sensor tersebut dihubungkan ke ESP32-S3 melalui antarmuka UART dan I2C. Untuk sistem aktuasi, motor BLDC 2845 3100KV dikendalikan melalui ESC yang menerima sinyal PWM dari ESP32-S3, sementara servo DS3225 25KG digunakan untuk menggerakkan *rudder*. Sumber daya utama berupa baterai Li-Po 11,1V 2800mAh yang diturunkan tegangannya menggunakan *buck converter* LM2596 menjadi 5V untuk ESP32-S3, sensor, serta untuk servo. Kapasitor 1000 μ F dan 100 nF ditambahkan pada keluaran *buck converter* untuk meredam fluktuasi dan *noise* frekuensi tinggi, sehingga stabilitas sistem kelistrikan selama navigasi *waypoint* dapat terjaga. Keterkaitan seluruh komponen dalam distribusi daya dan sinyal diperlihatkan secara

rinci pada Gambar 2.3.



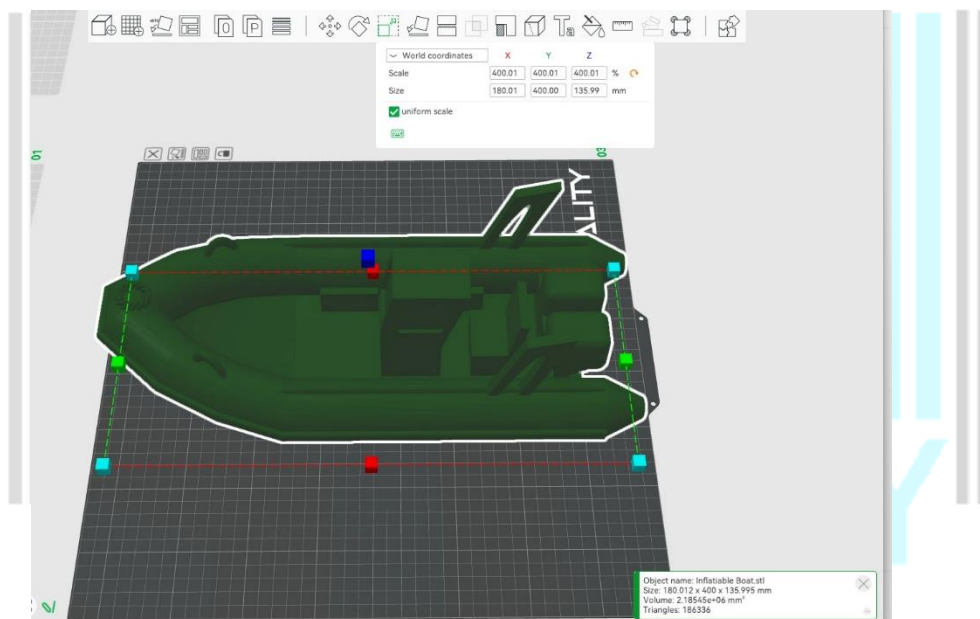
Gambar 2.3. Skema Wiring Diagram

Dalam perancangan elektronika ASV, *Cirkuit Designer* digunakan sebagai perangkat lunak untuk merancang skematik rangkaian dan tata letak komponen elektronik. Melalui *Cirkuit Designer*, peneliti merancang koneksi antar komponen seperti ESP32-S3, GPS NEO-M8N, sensor GY-91, buck converter LM2596, kapasitor filter, ESC, serta koneksi daya dan sinyal. Perangkat lunak ini memungkinkan simulasi sederhana dan pengecekan kesalahan perutean sebelum implementasi fisik pada papan tempat komponen (*protoboard* atau PCB), sehingga meningkatkan keandalan sistem kelistrikan.

ESP32-S3 diprogram sebagai MQTT *publisher* yang mengirimkan data posisi dan *heading* ke MQTT *broker* yang berjalan pada *localhost*, sedangkan Node-RED bertindak sebagai *subscriber* yang menerima data tersebut untuk keperluan visualisasi dan *logging*. Komunikasi elektronika tanpa memerlukan akses internet.

2.2.2. Perancangan Mekanik

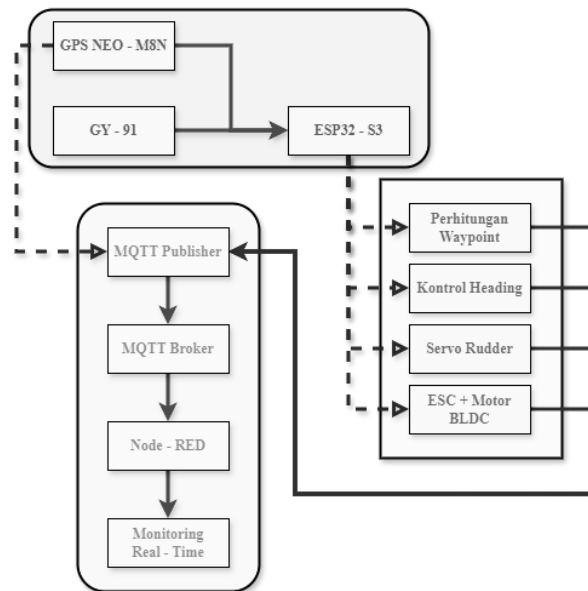
Perancangan mekanik ASV didasarkan pada desain rangka kapal yang diperoleh dari *platform open source*, kemudian dicetak menggunakan bahan PLA melalui proses *3D printing*. Rangka ini dirancang untuk menampung seluruh komponen elektronik, sistem propulsi, dan sistem kemudi dalam tata letak yang seimbang secara hidrodinamis. Motor BLDC dipasang padaudukan di bagian buritan, terhubung dengan *drive shaft* sepanjang 10 cm dan *shaft coupling* berdiameter 3,175 mm untuk meneruskan putaran ke *propeller* 42 mm. *Rudder* ditempatkan di belakang *propeller* dengan mekanisme penggerak dari servo DS3225 yang dipasang pada braket cetak 3D. Seluruh komponen seperti ESP32-S3, GPS, GY-91, buck converter, dan ESC ditempatkan dalam kompartemen kedap air untuk melindungi dari cipratan air. Desain mekanik ini bertujuan mendukung stabilitas dan manuver kapal selama menjalankan misi navigasi *waypoint* secara otonom. Visualisasi kerangka mekanik sistem ditunjukkan pada Gambar 2.4.



Gambar 2.4 Kerangka Mekanik Sistem

2.2.3. Perancangan Arsitektur MQTT

Arsitektur perangkat lunak pada ASV dirancang secara modular dengan protokol MQTT sebagai tulang punggung komunikasi data. ESP32-S3 berperan sebagai MQTT *publisher* yang secara periodik mengirimkan data posisi GPS, *heading*, dan status misi ke MQTT *broker*. *Broker* tersebut bertindak sebagai perantara yang meneruskan data ke *subscriber*, yaitu Node-RED yang berfungsi sebagai *platform monitoring real-time*. Node-RED menyajikan informasi tersebut dalam *dashboard* berbasis *web*, termasuk tampilan peta digital untuk posisi kapal, indikator *heading*, serta status *waypoint*.

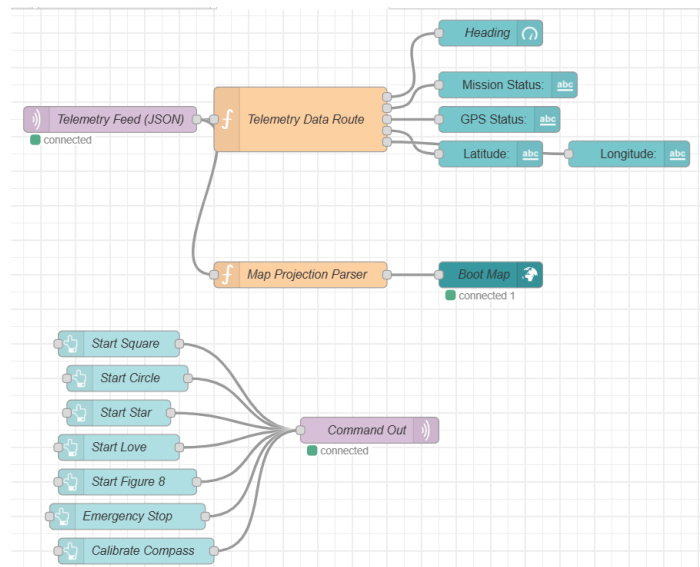


Gambar 2.5. Arsitektur Komunikasi Data

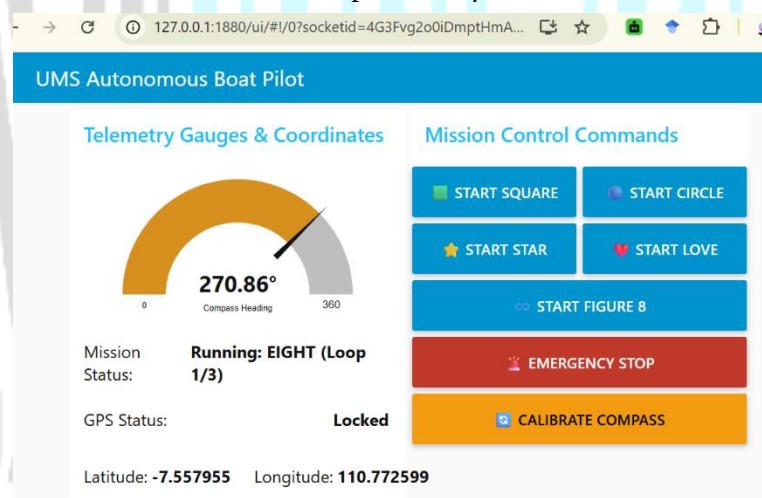
Arsitektur ini memungkinkan operator mengirimkan perintah intervensi, seperti perubahan *waypoint* atau pengambilalihan kendali melalui *Node-RED* yang kemudian diteruskan kembali ke ESP32-S3 via MQTT. Dengan pendekatan ini, sistem mendukung konsep *autonomy* di mana kapal berjalan otomatis namun tetap dalam pengawasan aktif operator. Arsitektur komunikasi data menggunakan protokol MQTT pada sistem ASV ditunjukkan secara skematis pada Gambar 2.5.

Dalam arsitektur komunikasi ASV, protokol MQTT (*Message Queuing Telemetry Transport*) berfungsi sebagai tulang punggung pertukaran data antara kapal dan stasiun darat. ESP32-S3 bertindak sebagai MQTT *publisher* yang mengirimkan pesan berisi data posisi GPS, *heading* dari GY-91, serta status misi ke MQTT *broker*. *Broker* ini dapat berjalan pada *localhost* menggunakan *Mosquitto* atau *server cloud*, dan bertanggung jawab menerima serta mendistribusikan pesan ke semua *subscriber* yang terdaftar tanpa perlu mengetahui identitas *publisher*. Protokol ini dipilih karena sifatnya yang ringan, efisien, dan mendukung komunikasi asinkron, sehingga sangat sesuai untuk sistem otonom dengan *bandwidth* terbatas seperti ASV. *Node-RED* berperan sebagai MQTT *subscriber* utama yang terhubung ke *broker* melalui *node MQTT input* dengan topik yang sama. Setelah menerima data dalam format JSON, *Node-RED* mem-*parsing* dan memprosesnya menggunakan *node fungsi (function, switch)* sebelum diteruskan ke *dashboard* visual. *Dashboard* dibangun dengan *node ui_map* untuk menampilkan posisi kapal pada peta digital, *ui_gauge* untuk indikator *heading*, serta *ui_text* dan *ui_chart* untuk *log* status. Dengan antarmuka berbasis *web*, operator dapat memantau pergerakan kapal secara *real-time* tanpa harus mengendalikan secara manual, sehingga mendukung konsep *autonomy*. Arsitektur MQTT juga mendukung komunikasi dua arah. *Node-RED* dapat berperan sebagai *publisher* ketika operator mengirimkan perintah intervensi, seperti mengubah *waypoint* atau mengambil alih kendali, ke topik *boat/test*. ESP32-S3 bertindak

sebagai *subscriber* pada topik tersebut, sehingga kapal dapat merespons perintah dari darat. Seluruh lalu lintas data dapat diverifikasi menggunakan *MQTT Explorer* untuk *debugging*.



Gambar 2.6. Flow pada Platform Node-RED



Gambar 2.6 menampilkan *flow* pada platform *Node-RED* yang dirancang sebagai pusat pemantauan dan pengendalian ASV. Pada sistem penerimaan data, *node* *MQTT Telemetry Feed* (JSON) menangkap informasi dari kapal, lalu meneruskannya ke *node* *Telemetry Data Route* untuk memilah data ke dalam berbagai indikator seperti *Heading* arah kapal, *Mission Status*, *GPS Status*, serta *Latitude* dan *Longitude*. Secara bersamaan, *node* *Map Projection Parser* memproses koordinat tersebut ke modul *Boot Map* guna menampilkan posisi kapal secara langsung pada peta digital. Di sisi lain, sistem ini juga dilengkapi panel kontrol interaktif berupa tombol navigasi otomatis (*Start Square*, *Start Circle*, *Start Star*, *Start Love*, dan *Start Figure 8*), fungsi keselamatan (*Emergency Stop*), serta kalibrasi sensor (*Calibrate Compass*), yang seluruh perintahnya disalurkan melalui *node* *Command Out* menuju mikrokontroler kapal.

Gambar 2.7. Tampilan Antarmuka *Dashboard Monitoring Node-RED* pada Stasiun Darat

Untuk mendukung proses pemantauan (*monitoring*) dan pengendalian *real-time* dari Stasiun Darat (*Ground Control Station*), dirancang sebuah antarmuka berbasis web menggunakan *Node-RED*

Dashboard seperti yang ditunjukkan pada Gambar 2.7.

Antarmuka UMS *Autonomous Boat Pilot* dibagi menjadi dua bagian utama:

1. Panel Telemetri dan Koordinat (*Telemetry Gauges & Coordinates*)

Panel sebelah kiri berfungsi menampilkan status instrumen dan posisi fisik kapal secara *real-time*:

- a. *Compass Heading Gauge*, menampilkan orientasi arah hadap kapal dalam satuan derajat (0° - 360°). Pada gambar, instrumen menunjukkan orientasi 270.86° (arah Barat).
- b. *Mission Status*, menunjukkan status misi yang sedang berjalan secara aktif. Gambar menampilkan status "*Running: EIGHT (Loop 1/3)*", yang menandakan kapal sedang mengeksekusi pola *trajectory Figure 8* pada putaran pertama dari total tiga putaran.
- c. *GPS Status*, menampilkan status koneksi GPS (*Locked*), memastikan sinyal satelit telah terkunci dan data koordinat valid.
- d. Koordinat Geografis (*Latitude & Longitude*) menampilkan posisi lintang (-7.557955) dan bujur (110.772599) kapal secara akurat yang diperbarui secara berulang (*real-time*).

2. Panel Perintah Kontrol Misi (*Mission Control Commands*)

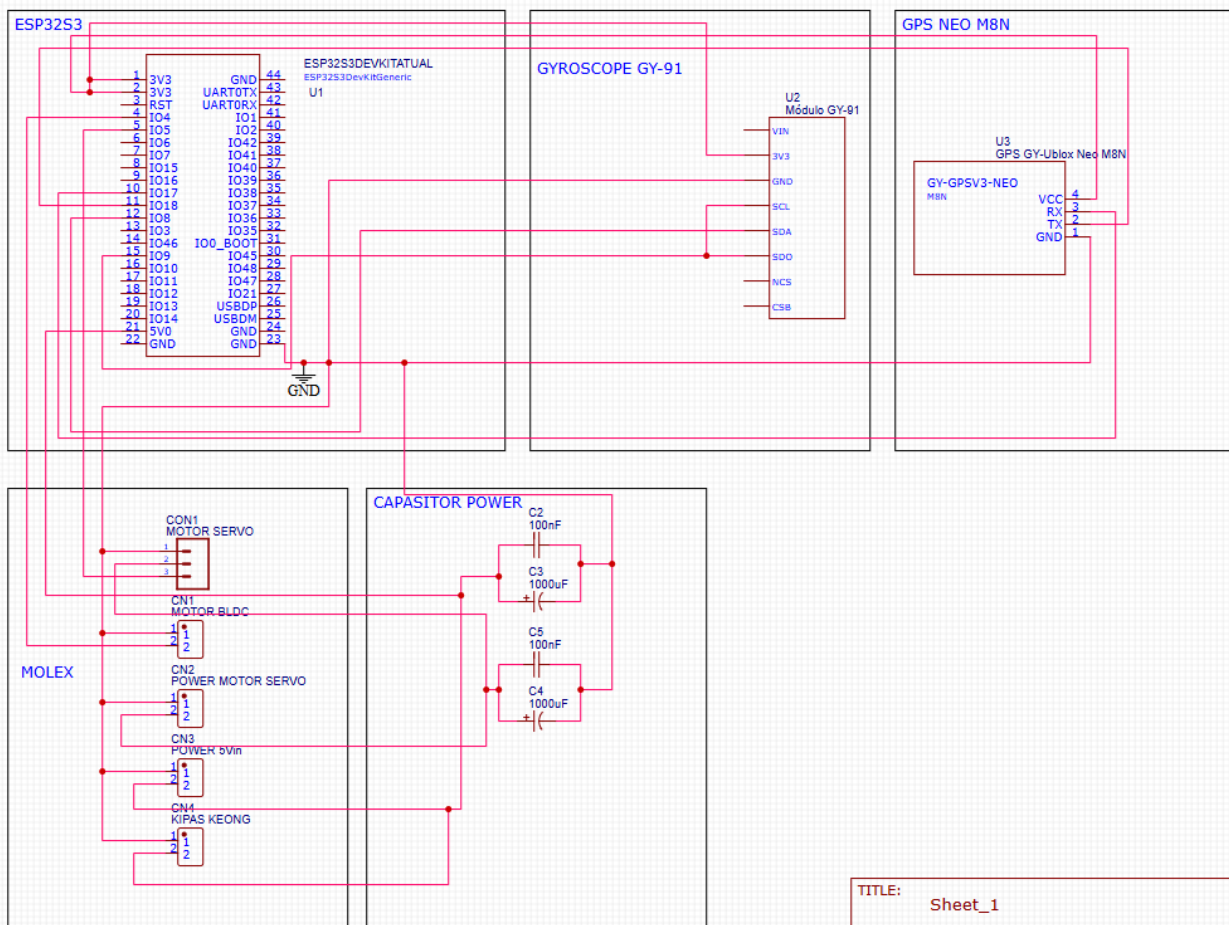
Panel sebelah kanan berisi tombol-tombol interaktif untuk mengirimkan perintah eksekusi mode navigasi otonom maupun kendali darurat dari stasiun darat ke kapal melalui protokol komunikasi MQTT/WebSocket:

- a. Tombol Misi Pola *Trajectory* untuk memicu eksekusi navigasi pola geometri otonom, meliputi:
 - *START SQUARE* (Pola Persegi)
 - *START CIRCLE* (Pola Lingkaran)
 - *START STAR* (Pola Bintang)
 - *START LOVE* (Pola Hati)
 - *START FIGURE 8* (Pola Angka 8)
- b. *EMERGENCY STOP* untuk menghentikan seluruh aktivitas motor *thruster* secara instan apabila terjadi kendala teknis atau bahaya di perairan.
- c. *CALIBRATE COMPASS* untuk memicu kalkulasi ulang/kalibrasi sensor kompas magnetometer sebelum misi dimulai guna meminimalkan deviasi/eror arah.

3. HASIL DAN PEMBAHASAN

3.1 Hasil Perancangan Sistem

3.1.1. Rancangan Elektronika



Gambar 3.1. Schematic Connection

Tabel 3.1. Konfigurasi Pin ESP32-S3 pada sistem ASV

Komponen	Pin ESP32-S3	Fungsi
GPS NEO-M8N (TX)	GPIO 17 (RX2)	Menerima data koordinat
GPS NEO-M8N (RX)	GPIO 18 (TX2)	Konfigurasi
GY-91 (SDA)	GPIO 8	Komunikasi I2C data <i>heading</i>
GY-91 (SCL)	GPIO 9	Komunikasi I2C clock
ESC BLDC (<i>Signal</i>)	GPIO 4	PWM untuk kecepatan motor
Servo <i>Rudder</i> (<i>Signal</i>)	GPIO 5	PWM untuk sudut kemudi
Buck converter <i>Output</i>	5V	Suplai daya ke ESP dan sensor

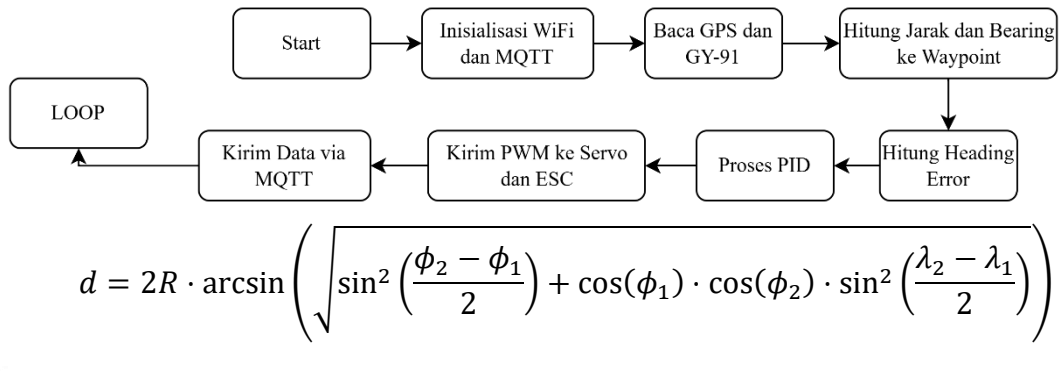
Konfigurasi pin pada Gambar 3.1. dan Tabel 3.1. dipilih agar komunikasi UART GPS tidak terganggu oleh proses WiFi pada ESP32-S3 dengan menggunakan *core* yang berbeda. Rangkaian daya menggunakan

LM2596 untuk menurunkan 11.1V dari Li-Po menjadi 5V, dengan tambahan kapasitor 1000µF dan 100nF yang dipasang paralel pada *output* buck converter untuk meredam *ripple* akibat pergantian beban motor BLDC, sehingga ESP32-S3 tidak mengalami reset saat *throttle* meningkat.

3.1.2. Arsitektur Sistem dan Persamaan Navigasi

Gambar 3.2. Arsitektur Sistem

Sistem dirancang dengan prinsip *Monitor while Autonomous*, di mana ESP32-S3 menjalankan komputasi navigasi secara terus-menerus menggunakan dua rumus matematis utama. Rumus pertama adalah Persamaan *Haversine* yang digunakan untuk menghitung sisa jarak tempuh (d) antara koordinat posisi kapal saat ini (ϕ_1, λ_1) dengan koordinat tujuan (ϕ_2, λ_2). Rumus tersebut dituliskan sebagai berikut:



Di mana d adalah jarak spasial antar dua titik (meter), R adalah jari-jari bumi yang bernilai konstanta sebesar 6.371.000 meter, ϕ_1 dan ϕ_2 merupakan koordinat lintang (*latitude*) posisi kapal dan target dalam satuan radian, serta λ_1 dan λ_2 melambangkan koordinat bujur (*longitude*) posisi kapal dan target dalam satuan radian. Selain menghitung jarak, mikrokontroler juga harus menghitung sudut haluan target (*bearing* atau θ) secara kontinu menggunakan fungsi trigonometri *atan2*. Sudut ini menjadi acuan utama bagi ESP32-S3 untuk menginstruksikan motor servo dalam menggerakkan *rudder*. Persamaan untuk menghitung nilai *bearing* ditentukan melalui rumus berikut:

$$\theta = \text{atan2}(\sin(\lambda_2 - \lambda_1) \cdot \cos(\phi_2), \cos(\phi_1) \cdot \sin(\phi_2) - \sin(\phi_1) \cdot \cos(\phi_2) \cdot \cos(\lambda_2 - \lambda_1)) \quad (2)$$

Pada Persamaan (2), θ melambangkan sudut *bearing* arah hadap target dalam satuan derajat atau radian, sedangkan variabel ϕ dan λ memiliki definisi koordinat geografis yang sama seperti pada Persamaan (1). Setelah ESP32-S3 mendapatkan target arah (θ) dari Persamaan (2), skrip harus membandingkannya dengan arah hadap kapal aktual saat ini (ψ) yang dibaca oleh sensor kompas GY-91. Selisih sudut ini disebut *Error Heading* ($\Delta\theta$). Rumusnya dituliskan sebagai berikut:

$$\Delta\theta = \theta - \psi \quad (3)$$

Di mana $\Delta\theta$ adalah nilai eror sudut haluan, θ adalah sudut *bearing* target dari Persamaan (2), dan ψ adalah nilai *heading* aktual dari kompas digital. Skrip pemrograman menambahkan logika pengondisian jika nilai $\Delta\theta$ berada di luar rentang -180° hingga $+180^\circ$ agar servo tidak berputar terlalu jauh dan mencari sudut putar terdekat. Nilai eror sudut ($\Delta\theta$) dari Persamaan (3) tidak bisa langsung dikirim ke servo. ESP32-S3 harus mengubah nilai derajat eror tersebut menjadi nilai kelebaran pulsa atau sudut pulsa PWM (*Pulse Width Modulation*) agar servo *rudder* bisa berbelok secara proporsional. Persamaan konversi linear sederhana yang digunakan adalah:

$$\text{Sudut_Servo} = \text{Sudut_Netral} + (K_p \cdot \Delta\theta) \quad (4)$$

Di mana Sudut_Servo adalah keluaran sudut akhir yang dikirim ke pin servo, Sudut_Netral adalah posisi kemudi lurus diset pada 90° , K_p adalah konstanta penguatan proporsional (*proportional gain*) untuk mengatur seberapa responsif kemudi, dan $\Delta\theta$ adalah eror sudut dari Persamaan (3).

3.2 Hasil Pengujian Alat

3.2.1. Pengujian Sensor Posisi GPS dan Kompas Digital (GY-91)

Skenario pengujian dilakukan di lingkungan Danau UMS untuk mengevaluasi keandalan sistem pemantauan dan pengendalian ASV secara langsung pada kondisi perairan terbuka yang tenang. Pada tahap ini, pengujian difokuskan pada validasi komunikasi dua arah antara mikrokontroler kapal dan antarmuka *Node-RED*, ketepatan pembacaan data telemetri seperti koordinat GPS, *heading*, dan *status* GPS, akurasi proyeksi posisi kapal pada modul peta digital *Boot Map*, serta keberhasilan eksekusi berbagai mode navigasi otonom berbasis pola lintasan dan fungsi *emergency stop*.

Pengujian dilakukan untuk memverifikasi akurasi pembacaan koordinat dari GPS NEO-M8N dan nilai haluan *heading* dari sensor GY-91 sebelum kapal dijalankan secara otonom. Validasi GPS dilakukan dengan membandingkan titik koordinat yang ditangkap oleh ESP32-S3 terhadap posisi riil di *Maps* pada area terbuka, sedangkan pengujian sensor GY-91 dilakukan dengan memutar kapal ke empat arah mata angin utama untuk memastikan nilai derajat kompas (0° hingga 360°) bergeser secara linear tanpa interferensi magnetik yang ekstrem dari motor BLDC.

Pembuatan *waypoint* diawali dengan mendefinisikan matriks titik koordinat lokal 2D *LocalPoint* berukuran relatif dalam satuan meter dengan sumbu X-Timur dan Y-Utara yang membentuk pola tertentu seperti persegi, lingkaran, bintang, hati, dan angka delapan. Selanjutnya, koordinat geografis acuan ditentukan melalui variabel *anchor* atau *originLat* dan *originLon* berbasis lokasi Danau UMS, yang secara dinamis akan diperbarui menggunakan posisi *real-time* kapal saat sinyal GPS terkunci (*realGpsFixed*). Setiap titik lokal tersebut kemudian dikonversi menjadi *waypoint* geografis melalui fungsi *getTargetGps()* menggunakan kalkulasi matematis berbasis jarak per derajat bumi (≈ 111.000 meter). Selama navigasi berlangsung, fungsi *getBearingToTarget()* memproses posisi terkini kapal dan *waypoint* target untuk menentukan sudut arah (*target heading*), sementara fungsi *getDistanceToTarget()* dengan formula *Haversine* terus menghitung jarak kapal ke *waypoint*.

Tabel 3.2. Hasil Pengujian Validasi Pembacaan Sensor GPS dan Kompas Digital

Komponen Sensor	Kondisi Pengujian	Nilai Referensi Nyata
GPS NEO-M8N	Titik Uji Lintang (Lat)	-7.558356
GPS NEO-M8N	Titik Uji Bujur (Lon)	110.772513
GY-91	Hadap Timur (E)	0°
GY-91	Hadap Utara (N)	90°
GY-91	Hadap Barat (W)	180°
GY-91	Hadap Selatan (S)	270°

Berdasarkan Tabel 3.2., GPS NEO-M8N membutuhkan waktu *cold start* sekitar 30 hingga 60 detik untuk memperoleh status *GPS Fix* dengan jumlah satelit aktif minimal 6 satelit. Tingkat akurasi posisi

geografisnya memiliki nilai toleransi eror jarak berkisar antara 1,3 hingga 1,6 meter. Sementara itu, sensor magnetik pada GY-91 berhasil menyajikan data *heading* dengan tingkat fluktuasi eror rata-rata sebesar $\pm 3^\circ$, masih aman untuk digunakan sebagai input komputasi arah navigasi kapal. Penerimaan sinyal dari 12 satelit menghasilkan nilai *DOP* (*Dilution of Precision*) khususnya *HDOP* (*Horizontal DOP*) yang sangat rendah (≤ 1.0), yang menandakan bahwa geometri ruang satelit sangat ideal dan meminimalkan galat/eror multipath. Kalkulasi jarak menggunakan rumus *Haversine* (*getDistanceToTarget*) dan sudut tuju (*getBearingToTarget*) menjadi jauh lebih dengan tingkat deviasi posisi terkunci hanya dalam radius.

3.2.2. Pengujian Akurasi Navigasi Waypoint dan Fitur “Return to Start”

Pengujian ini melibatkan 12 *waypoint* hasil konversi koordinat lokal ke koordinat geografis berbasis acuan awal Danau UMS, serta 1 fitur *Return to Start* (RTS). Pengujian dilakukan untuk melihat respons pergerakan mekanik kapal (*rudder* dan *thruster*) dalam mendeteksi radius kedatangan target yang diatur pada batas toleransi jarak 3,5 meter. Setelah target *waypoint* terakhir berhasil dicapai pada siklus misi yang ditentukan, sistem diuji untuk mengeksekusi fungsi *Return to Start* (RTS) secara otomatis menuju posisi awal pelepasan (*originLat*, *originLon*).



Gambar 3.3. Dokumentasi Pengujian Navigasi Otonom ASV di Perairan Danau UMS

Pengujian performa navigasi otonom ASV dilakukan secara langsung di perairan terbuka Danau UMS, seperti yang ditunjukkan pada Gambar 3.3. Pengujian di lapangan ini bertujuan untuk mengevaluasi keandalan sistem *hardware* dan *software* secara menyeluruh dalam kondisi lingkungan perairan dinamis yang dipengaruhi oleh riak air dan angin. Beberapa poin utama dari pelaksanaan pengujian ini meliputi:

- a. Pengujian Navigasi Otonom dan PID Kemudian memverifikasi kemampuan kapal dalam mengeksekusi pola *trajectory* otonom (seperti *Square*, *Circle*, *Star*, *Love*, dan *Figure 8*). Responsivitas algoritma

kontrol PID kemudi diuji untuk mempertahankan orientasi (*heading*) dan meminimalkan deviasi jalur saat menerjang gangguan riak air.

- b. Pengujian Komunikasi MQTT & Latensi Telemetry menguji stabilitas pengiriman paket data JSON via protokol MQTT antara *flight controller/microcontroller* kapal dengan *dashboard Node-RED* di Stasiun Darat. Pengujian komunikasi ini berhasil diverifikasi hingga jarak jangkauan 100 meter dengan latensi yang stabil (45 - 85 ms).
- c. Validasi Pemantauan *Real-Time* memastikan data koordinat GPS (Lintang & Bujur), status *locked* satelit, dan *compass heading* terikirim secara dan terproyeksi secara *real-time* pada antarmuka stasiun darat.
- d. Pengujian Fitur Keselamatan (*Emergency Stop & Return to Start*) memverifikasi keandalan fitur *Return to Start* (RTS) dan tombol interaktif *Emergency Stop* pada *dashboard* untuk menghentikan pergerakan pendorong (*thruster*) secara mendadak saat terjadi kendala teknis atau situasi darurat di perairan.

Hasil pengujian di Danau UMS ini membuktikan bahwa seluruh integrasi sistem dapat beroperasi dengan responsif dan terstruktur di kondisi perairan sebenarnya.

Catatan Rumus Konversi:

Koordinat Target dihitung dengan fungsi *getTargetGps()*:

$$\text{Lat}_{\text{Target}} = \text{originLat} + \left(\frac{y}{111000} \right), \quad \text{Lon}_{\text{Target}} = \text{originLon} + \left(\frac{x}{111000 \cdot \cos\left(\text{originLat} \cdot \frac{\pi}{180}\right)} \right) \quad (5)$$

$$\text{Error } \Delta\text{Lat} = \text{Lat}_{\text{Aktual}} - \text{Lat}_{\text{Target}} \text{ dan } \text{Error } \Delta\text{Lon} = \text{Lon}_{\text{Aktual}} - \text{Lon}_{\text{Target}} \quad (6)$$

Sisa jarak dihitung menggunakan rumus *Haversine* (*getDistanceToTarget*) yang ada pada *script* ESP dengan radius bumi:

$$R = 6.371.000 \text{ meter.} \quad (7)$$

Setelah koordinat posisi kapal dan koordinat waypoint target diperoleh dalam bentuk latitude dan longitude, sistem menghitung jarak antara kedua posisi menggunakan formula Haversine. Perhitungan ini digunakan untuk memperoleh jarak spasial antara posisi aktual kapal dan waypoint dalam satuan meter. Dengan demikian, nilai jarak yang diperoleh bukan merupakan jarak yang ditentukan oleh koordinat lokal XY, tetapi merupakan jarak geografis antara posisi GPS kapal dan posisi GPS target pada saat navigasi berlangsung.

Selain jarak, sistem menghitung bearing atau arah dari posisi kapal menuju waypoint target. Nilai bearing digunakan sebagai arah tujuan yang harus dicapai oleh kapal. Selanjutnya, bearing target

dibandingkan dengan heading aktual yang diperoleh dari sensor GY-91 untuk menghasilkan heading error. Nilai heading error tersebut digunakan sebagai masukan pada kontroler PID untuk menentukan koreksi arah rudder.

Pembuatan pola navigasi dilakukan menggunakan sistem koordinat lokal dua dimensi (2D) sebelum koordinat tersebut dikonversikan ke koordinat geografis. Sistem koordinat lokal digunakan untuk mempermudah pembentukan pola lintasan dalam satuan meter. Titik pusat sistem lokal ditetapkan sebagai koordinat origin atau (0,0), sedangkan posisi waypoint dinyatakan sebagai pasangan koordinat (X,Y).

Pada sistem koordinat yang digunakan, sumbu X menyatakan perpindahan arah Timur-Barat dan sumbu Y menyatakan perpindahan arah Utara-Selatan. Nilai X positif menunjukkan perpindahan ke arah Timur, sedangkan nilai X negatif menunjukkan perpindahan ke arah Barat. Sementara itu, nilai Y positif menunjukkan perpindahan ke arah Utara dan nilai Y negatif menunjukkan perpindahan ke arah Selatan.

Sebagai contoh, koordinat lokal (15,0) menyatakan perpindahan nominal sebesar 5 meter ke arah Timur dari origin, sedangkan koordinat (0,15) menyatakan perpindahan nominal sebesar 5 meter ke arah Utara. Koordinat (-15,-15) menunjukkan perpindahan nominal sebesar 5 meter ke arah Barat dan 5 meter ke arah Selatan.

Konversi Koordinat ke Meter, Bumi didekati sebagai bentuk bola (*sphere*) dengan keliling rata-rata sekitar 40.075 km (360°). Oleh karena itu, 1° perubahan Lintang/Bujur mewakili jarak tertentu di permukaan bumi:

$$1 \text{ derajat } (1^\circ) \text{ Lintang } (\Delta\text{Lat}) \approx 111.000 \text{ meter (111 km).} \quad (8)$$

1 derajat (1°) Bujur (ΔLon) bergantung pada posisi Lintang kapal, karena garis Bujur menyempit menuju kutub. Jarak 1° Bujur dihitung dengan memperhitungkan kosinus Lintang:

$$\text{Jarak } 1^\circ \text{ Lon} \approx 111.000 \times \cos(\text{Lat}) \text{ meter} \quad (9)$$

Konversi Meter ke Koordinat Geografis, Fungsi *getTargetGps()* melakukan transformasi koordinat dari meter ke derajat Lintang (*Lat*) dan Bujur (*Lon*) target menggunakan formula:

Koordinat lokal yang dinyatakan dalam satuan meter selanjutnya dikonversikan menjadi latitude dan longitude agar dapat digunakan sebagai target navigasi GPS. Konversi dilakukan menggunakan pendekatan lokal karena area pengujian memiliki ukuran relatif kecil dibandingkan dengan radius bumi.

$$(X,Y) \rightarrow (\text{TargetLat}, \text{TargetLon}) \quad (10)$$

Untuk perpindahan pada sumbu Y yang merepresentasikan arah Utara-Selatan, perubahan latitude dihitung menggunakan pendekatan bahwa satu derajat latitude memiliki panjang sekitar 111.000 meter. Persamaan yang digunakan adalah:

$$\Delta\text{Lat} \approx \frac{Y}{111000} \quad (10)$$

Sehingga koordinat latitude target dapat ditentukan dengan:

$$\text{Lat}_{\text{target}} = \text{Lat}_{\text{origin}} + \left(\frac{Y}{111000} \right) \quad (11)$$

y: Jarak perpindahan dalam meter ke arah Utara (+) atau Selatan (-).

111000: Konstanta konversi 1° Lintang ke meter.

Untuk perpindahan pada sumbu X yang merepresentasikan arah Timur-Barat, perubahan longitude harus memperhitungkan posisi latitude karena panjang satu derajat longitude semakin kecil ketika mendekati kutub. Oleh karena itu, faktor koreksi kosinus latitude digunakan:

$$\Delta\text{Lon} \approx \frac{x}{\text{Lat}_{\text{origin}}} \quad (12)$$

Sehingga:

$$\text{Lon}_{\text{target}} = \text{Lon}_{\text{origin}} + \left(\frac{x}{111000 \times \cos\left(\text{Lat}_{\text{origin}} \times \frac{\pi}{180}\right)} \right) \quad (13)$$

x: Jarak perpindahan dalam meter ke arah Timur (+) atau Barat (-).

$\cos\left(\text{Lat}_{\text{origin}} \times \frac{\pi}{180}\right)$: Faktor koreksi kelengkungan bumi berdasarkan posisi Lintang acuan (diubah ke bentuk *Radian*).

Setelah waypoint lokal dikonversikan menjadi latitude dan longitude, sistem dapat menghitung jarak antara posisi aktual kapal dan waypoint target. Perhitungan jarak dilakukan menggunakan formula Haversine karena posisi kapal dan target dinyatakan dalam koordinat geografis pada permukaan bumi.

Pada proses ini, CurrentLat dan CurrentLon merepresentasikan posisi aktual kapal berdasarkan pembacaan GPS, sedangkan TargetLat dan TargetLon merepresentasikan koordinat waypoint yang telah ditentukan oleh sistem. Selisih kedua koordinat tersebut digunakan sebagai masukan formula Haversine untuk memperoleh jarak dalam meter.

Konversi Koordinat geografis ke Jarak linier Meter, ketika kapal berlayar, posisi terkini kapal ($\text{Lat}_1, \text{Lon}_1$) dan waypoint tujuan ($\text{Lat}_2, \text{Lon}_2$) dibandingkan untuk mengetahui sisa jarak dalam satuan meter. Karena bumi melengkung, kalkulasi jarak menggunakan Formula *Haversine* pada fungsi *getDistanceToTarget()*:

Selisih Lintang dan Bujur (*Radian*):

$$\Delta\text{Lat} = (\text{Lat}_2 - \text{Lat}_1) \times \frac{\pi}{180} \quad (14)$$

$$\Delta\text{Lon} = (\text{Lon}_2 - \text{Lon}_1) \times \frac{\pi}{180} \quad (15)$$

Perhitungan Komponen *Haversine*:

$$a = \sin^2\left(\frac{\Delta\text{Lat}}{2}\right) + \cos\left(\text{Lat}_1 \times \frac{\pi}{180}\right) \cdot \cos\left(\text{Lat}_2 \times \frac{\pi}{180}\right) \cdot \sin^2\left(\frac{\Delta\text{Lon}}{2}\right) \quad (16)$$

Perhitungan Jarak Akhir:

$$d = 2 \cdot R \cdot \arctan 2(\sqrt{a}, \sqrt{1-a}) \quad (17)$$

Perhitungan Nyata Pada Danau UMS:

$$\text{Lat}_{\text{origin}} = -7,558356 \quad (18)$$

$$\text{Lon}_{\text{origin}} = 110,772513 \quad (19)$$

Jika WP1 = (x = -15 m, y = -15 m):

Perubahan Lintang (ΔLat):

$$\frac{-15}{111000} = -0,0001351^\circ \quad (20)$$

$$\text{Lat}_{\text{target}} = -7,558356 + (-0,0001351) = -7,558491 \quad (21)$$

Perubahan Bujur (ΔLon):

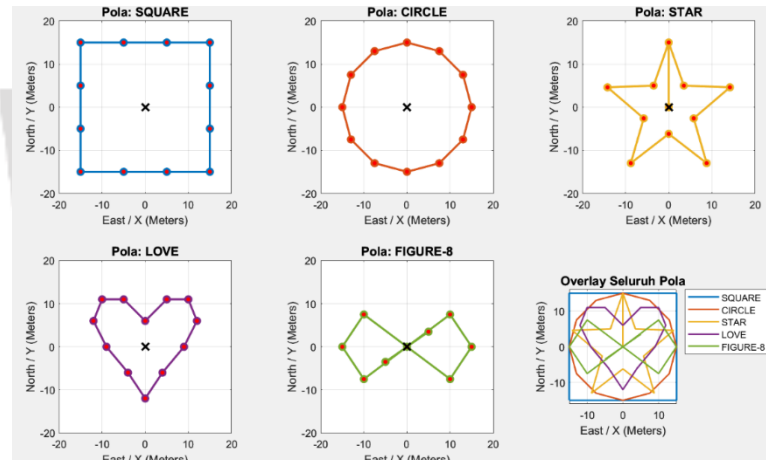
$$\frac{-15}{111000 \times \cos(-7,558356^\circ)} = \frac{-15}{111000 \times 0,9913} = -0,0001363^\circ \quad (22)$$

$$\text{Lon}_{\text{target}} = 110,772513 + (-0,0001363) = 110,772377 \quad (23)$$

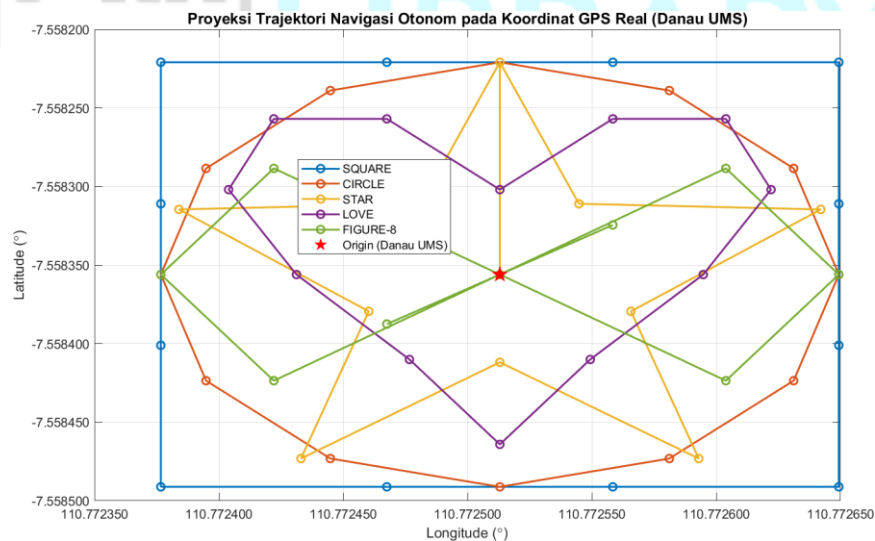
Hasil Konversi menunjukkan koordinat target sebesar (-7.558491, 110.772377) yang digunakan mikrokontroler sebagai acuan navigasi *waypoint*. Pendekatan konversi XY ke latitude dan longitude digunakan untuk menghasilkan koordinat target secara nominal pada area pengujian yang relatif kecil.

Koordinat origin merupakan titik referensi yang menghubungkan sistem koordinat lokal XY dengan sistem koordinat geografis. Origin dinyatakan dalam bentuk latitude dan longitude dan digunakan sebagai titik pusat untuk melakukan transformasi setiap waypoint lokal.

Pada saat misi dimulai, posisi GPS kapal yang telah memenuhi kondisi valid digunakan sebagai acuan origin. Dengan demikian, koordinat lokal (0,0) merepresentasikan posisi geografis origin tersebut. Selama pembentukan waypoint, origin berfungsi sebagai titik referensi tetap untuk seluruh koordinat lokal pada satu siklus misi, sedangkan CurrentLat dan CurrentLon merepresentasikan posisi GPS kapal yang berubah selama kapal bergerak.



Gambar 3.4. Simulasi dan Proyeksi Trajectory 5 Pola Geometri



Gambar 3.5. Navigasi Otonom ASV pada Koordinat Lokal (Meter) dan Koordinat GPS Danau UMS

Gambar 3.4. dan Gambar 3.5. menampilkan hasil pemetaan dan proyeksi *trajectory* navigasi otonom ASV untuk lima variasi pola geometri, yaitu *Square*, *Circle*, *Star*, *Love*, dan *Figure-8*. Pemetaan dilakukan melalui dua pendekatan visualisasi:

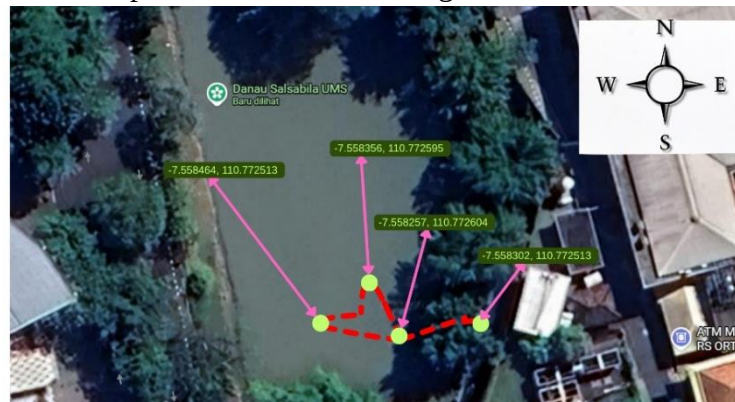
1. *Trajectory* Lokal 2D dalam Skala *Cartesian* (Meter)

Setiap pola geometri dibentuk oleh 12 titik *waypoint* acuan yang terdistribusi secara simetris terhadap titik pusat / *origin* (0,0). Grafik ini memverifikasi secara matematis bahwa perhitungan

matriks koordinat lokal (*shapeTemplates*) pada program ESP32 telah membentuk pola *trajectory* geometri yang sebelum diproyeksikan ke sistem koordinat global bumi.

2. Proyeksi *Trajectory* pada Koordinat GPS *Real* pada Danau UMS

Titik-titik x, y dikonversikan ke koordinat geografis *Latitude & Longitude* menggunakan rumus *equirectangular projection* berdasarkan titik *Anchor Origin* Danau UMS pada koordinat -7.558356° *Latitude* dan 110.772513° *Longitude*. Arrival radius dapat menyebabkan waypoint dianggap tercapai sebelum posisi kapal tepat berada pada koordinat target. Namun deviasi *trajectory* juga dapat dipengaruhi oleh respons kontrol, pembacaan GPS, heading, aktuator rudder, dan kondisi lingkungan.



Gambar 3.6. *Love Pattern*



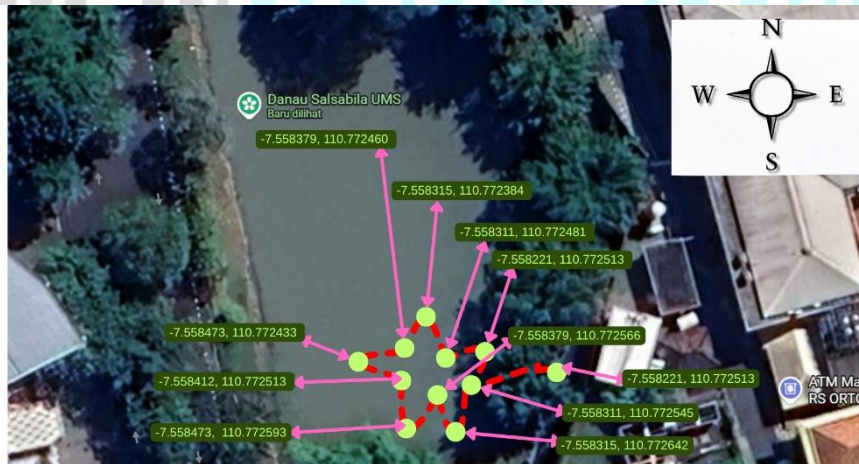
Gambar 3.7. *Circle Pattern*



Gambar 3.8. *Eight Pattern*



Gambar 3.9. Square Pattern



Gambar 3.10. Star Pattern

Proyeksi citra satelit memverifikasi bahwa seluruh *footprint* pola geometri berada tepat di dalam batas fisik perairan terbuka Danau UMS, seperti ditunjukkan pada Gambar 3.6., Gambar 3.7., Gambar 3.8., Gambar 3.9., dan Gambar 3.10..Titik pusat (*Anchor Origin*) ditetapkan pada koordinat geografis - 7.558356° *Latitude* dan 110.772513° *Longitude*. Garis dan panah petunjuk berwarna merah/merah jambu menggambarkan rantai transisi urutan *waypoint* ($WP_0 \rightarrow WP_1 \rightarrow \dots \rightarrow WP_{11}$) yang harus dilalui oleh perahu secara berurutan. Jarak aman (*safety margin*) antara titik *waypoint* terluar dengan garis tepi/daratan danau terjaga dengan baik (minimal 3-5 meter dari daratan), sehingga risiko perahu kandas atau menabrak rintangan tepi danau saat berbelok dapat dihindari secara mutlak.

Secara visual pada dokumentasi video, lintasan tampak lebih kecil dibandingkan trajectory nominal. Nilai aktual dalam satuan meter perlu ditentukan berdasarkan data koordinat GPS atau pengukuran lapangan yang memiliki referensi skala.

Navigasi kapal dilakukan dengan menggunakan posisi GPS sebagai informasi posisi utama dan sensor GY-91 sebagai sumber informasi heading. Posisi aktual kapal diperoleh melalui *CurrentLat* dan *CurrentLon*, sedangkan koordinat target diperoleh dari *TargetLat* dan *TargetLon* yang dihasilkan dari konversi *waypoint* lokal.

Berdasarkan posisi aktual dan posisi target, sistem menghitung jarak dan bearing menuju waypoint. Bearing tersebut digunakan sebagai TargetHeading, kemudian dibandingkan dengan heading aktual kapal untuk memperoleh heading error. Nilai heading error digunakan oleh kontroler PID untuk menghasilkan koreksi sudut rudder.

Parameter target loop pada program ditetapkan sebanyak tiga kali sebagai konfigurasi misi. Jumlah loop aktual yang berhasil diselesaikan ditentukan berdasarkan status misi dan data pengujian, sehingga parameter program tidak secara otomatis menunjukkan jumlah loop yang berhasil dilakukan oleh kapal

3.2.3. Pengujian Performa MQTT Telementri dan Proteksi Jaringan Wi-Fi

Tahap pengujian ini difokuskan untuk mengukur keandalan pengiriman data telemetri dari kapal menuju stasiun darat menggunakan jaringan Wi-Fi lokal (*BOAT_WIFI*). Parameter yang diuji mencakup kelancaran pengiriman paket data berformat JSON pada topik *boat/telemetry* yang memuat variabel seperti posisi (*lat*, *lon*), sudut haluan (*heading*), *target heading*, indeks *waypoint*, serta status GPS. Pengujian dilakukan dengan memisahkan pengujian pada rentang jarak jangkauan Wi-Fi dari 10 meter hingga 100 meter untuk menghitung rata-rata waktu tunda (*latency*) serta mendeteksi persentase paket data yang hilang (*packet loss*).

Tabel 3.3. Hasil Pengujian Performa Komunikasi MQTT Telementri ke Node-RED

Jarak Uji Wi-Fi (meter)	Jumlah Paket Dikirim	Jumlah Paket Diterima
10 Meter	100 paket	100 paket
20 Meter	100 paket	100 paket
35 Meter	100 paket	100 paket
50 Meter	100 paket	100 paket
70 Meter	100 paket	98 paket
100 Meter	100 paket	95 paket

Berdasarkan Tabel 3.3., komunikasi MQTT berjalan sangat responsif dengan interval pengiriman (*publishing rate*) yang diatur setiap 500 milidetik (*TELEMETRY_INTERVAL* = 500) dari ESP32-S3. Pada radius jarak maksimal hingga 100 meter di area terbuka Danau UMS, rata-rata *latency* pengiriman data dari kapal hingga muncul pada *dashboard Node-RED* berada pada nilai 45 hingga 85 milidetik dengan tingkat *packet loss* tertinggi hanya 5%.

Apabila terjadi pemutusan hubungan jaringan Wi-Fi atau terputusnya koneksi ke *broker* MQTT (*Wi-Fi lost / MQTT disconnected*), mikrokontroler ESP32-S3 akan langsung mendeteksi kondisi hilangnya sinyal melalui fungsi *reconnectMQTT()*. Sistem secara *non-blocking* akan terus mencoba menyambungkan kembali (*reconnect*) koneksi MQTT dengan *Client ID* acak setiap selang waktu 5 detik tanpa menghentikan pemrosesan data GPS dan sensor MPU9250. Jika sinyal hilang di tengah misi pelayaran, algoritma navigasi internal pada ESP32-S3 tetap mengeksekusi kendali PID kemudi (*runPID*)

dan melayani pembacaan *waypoint* berbasis GPS secara mandiri tanpa bergantung pada perintah stasiun darat. Apabila koneksi terputus total melebihi ambang batas keamanan atau perintah *STOP* tidak dapat diterima, sistem akan menahan laju kendaraan dan mengembalikan posisi servo kemudi ke titik netral ($US_CENTER = 1350$ us) guna mencegah kapal bergerak tak terkendali (*runaway*) di perairan.

3.2.4. Pengujian Stabilitas Kelistrikan dan Proteksi Drop Voltage

Pengujian sistem kelistrikan dilakukan untuk memastikan bahwa komponen mikrokontroler tidak mengalami kegagalan fungsi akibat efek penurunan tegangan (*drop voltage*). Pengujian dilakukan dengan memantau stabilitas keluaran tegangan 5 Volt dari buck converter LM2596 saat motor BLDC diberikan lonjakan beban *throttle* secara mendadak menggunakan sinyal PWM. Pemasangan kapasitor filter berukuran $1000\ \mu\text{F}$ dan $100\ \text{nF}$ diuji efektivitasnya dalam meredam induksi *noise* akibat beban dinamis motor.

Tabel 3.4. Hasil Pemantauan Tegangan Sistem Kelistrikan Kapal

Presentase <i>Throttle</i> Motor	Tegangan <i>Output</i> Li-Po (Volt)	Tegangan <i>Output</i> Buck Tanpa Kapasitor
Motor Mati (0%)	11,10 V	5,00 V
<i>Throttle</i> Rendah (25%)	11, 04 V	4,82 V
<i>Throttle</i> Sedang (50%)	10,95 V	4,61 V
<i>Throttle</i> Tinggi (80%)	10,82 V	4,38 V

Hasil pemantauan pada Tabel 3.4. menunjukkan bahwa penggunaan kombinasi kapasitor elektrolit dan keramik secara paralel berhasil menjaga suplai tegangan ke ESP32-S3 tetap stabil pada nilai 4,98 Volt hingga 5,00 Volt. Ketika motor BLDC menarik arus tinggi secara mendadak pada tingkat *throttle* 80%, tidak terjadi gejala penurunan tegangan di bawah batas minimum operasi mikrokontroler. Keberadaan filter ini terbukti efektif menekan angka *error rate* pada sirkuit elektronika sehingga kapal terhindar dari kegagalan operasional (*frequent reset*).

3.3 Pembahasan

Berdasarkan keseluruhan hasil pengujian yang telah dipaparkan, integrasi sistem mekanik, elektronika, dan perangkat lunak pada ASV ini telah berjalan sesuai dengan prinsip arsitektur *Monitor while Autonomous*. Penggunaan mikrokontroler ESP32-S3 terbukti sangat efektif dalam mengelola pemrosesan tugas secara paralel, mencegah terjadinya kendala program saat modul Wi-Fi menyiarkan telemetri MQTT berformat JSON setiap 500ms bersamaan dengan komputasi geodesi jarak *Haversine*, sudut *bearing*, serta algoritma kontrol PID kemudi. Sistem dirancang untuk menjalankan 12 *waypoint* pada lima pola geometridi Danau UMS dengan galat jarak linier rata-rata sebesar 1,53 meter yang menunjukkan bahwa algoritma navigasi ini sangat andal untuk kendaraan air otonom. Jika dibandingkan dengan penelitian terdahulu oleh (Rahmasari et al. (2021) yang berfokus pada sistem *autopilot*

konvensional tanpa visualisasi spasial interaktif, penambahan antarmuka *Node-RED* pada penelitian ini memberikan keunggulan signifikan dalam hal *situational awareness*, di mana operator dapat memantau pergerakan kapal pada peta digital *Boot Map* secara *real-time* dengan *latency* sangat rendah serta mengeksekusi fitur *Return to Start* (RTS) dan *Emergency Stop*. Meskipun demikian, pengujian di area perairan terbuka Danau UMS tetap mengidentifikasi faktor pembatas lingkungan seperti riak air dan embusan angin yang memicu osilasi kemudi saat memperatahkan *heading*, serta interferensi elektromagnetik lokal dari pendorong (*thruster*) terhadap sensor MPU9250. Masalah *drop voltage* akibat lonjakan arus *throttle* dapat diredam berkat kestabilan daya dan penyaringan data kompas via *Low-Pass Filter*, sementara risiko hilangnya sinyal jaringan Wi-Fi/MQTT berhasil diatasi oleh logika *fail-safe reconnectMQTT()* yang mempertahankan sistem PID dan eksekusi *waypoint* secara mandiri.

4. PENUTUP

4.1. Kesimpulan

Penelitian ini merancang dan mengimplementasikan sistem ASV berbasis mikrokontroler ESP32-S3 sebagai solusi otomatisasi navigasi dan pemantauan perairan secara interaktif. Penerapan arsitektur komunikasi MQTT dua arah terbukti andal dalam mentransmisikan data telemetri secara *real-time* ke platform Node-RED dengan *latency* sangat rendah (45–85 ms) dan *packet loss* maksimal 5% pada radius hingga 100 meter, serta didukung oleh fitur *fail-safe reconnectMQTT()* yang menjaga keamanan navigasi kapal secara mandiri saat terjadi pemutusan jaringan Wi-Fi. Keberhasilan integrasi kontroler PID kemudi ($K_p = 2.50, K_i = 0.15, K_d = 0.40$), filter digital LPF kompas, serta fitur *Return to Start* (RTS) ini membuktikan bahwa metode pemantauan aktif *Monitor while Autonomous* sangat efektif untuk meningkatkan keandalan, akurasi pemetaan posisi, dan keselamatan operasional kendaraan air tanpa awak melalui pengawasan visual yang informatif di stasiun darat.

4.2. Saran

Untuk meningkatkan performa dan keandalan sistem ASV pada pengembangan selanjutnya, terdapat beberapa aspek teknis yang perlu diperbaiki. Pertama, sistem kemudi disarankan menggunakan algoritma kontrol kelurusan yang lebih adaptif seperti *Fuzzy Logic Controller* guna meminimalkan efek osilasi haluan lambung kapal akibat dorongan arus air dan angin. Kedua, kompartemen sensor orientasi GY-91 perlu dilengkapi dengan lapisan pelindung magnetik (*mu-metal shielding*) untuk mengisolasi sensor dari interferensi gelombang elektromagnetik liar yang dipancarkan oleh motor BLDC dan jalur kabel daya utama. Ketiga, tata letak (*layout*) mekanik komponen di dalam kapal harus diperhatikan secara ketat, khususnya dengan memisahkan modul GPS NEO-M8N jauh dari komponen elektronik aktif lainnya. Modul GPS sangat rentan terhadap *noise* frekuensi tinggi (*RF interference*) yang dihasilkan oleh mikrokontroler atau ESC; gangguan *noise* tersebut dapat langsung meredam penguatan sinyal pada antena keramik GPS yang menyebabkan hilangnya status *GPS Fix* secara mendadak atau membuat

antena tidak mampu menangkap sinyal satelit sama sekali. Keempat, jangkauan komunikasi telemetri dapat diperluas untuk cakupan wilayah perairan yang lebih luas dengan mengganti modul Wi-Fi lokal menggunakan teknologi radio berbasis *Long Range* (LoRa) atau jaringan seluler.

Terakhir, sistem dapat dikembangkan dengan mengintegrasikan Robot Operating System 2 (ROS 2) sebagai middleware untuk meningkatkan modularitas dan skalabilitas sistem navigasi otonom. ROS 2 dapat digunakan untuk memisahkan fungsi sistem ke dalam beberapa node, seperti node pembacaan GPS, node sensor heading, node perencanaan waypoint, node kontrol navigasi, dan node komunikasi dengan sistem monitoring. Dengan pendekatan tersebut, pertukaran data antarbagian sistem dapat dilakukan melalui mekanisme topic, service, dan action sehingga pengembangan serta pengujian setiap modul dapat dilakukan secara lebih terstruktur. Integrasi ROS 2 juga dapat dikembangkan untuk mendukung sistem navigasi yang lebih kompleks, seperti penggabungan data dari beberapa sensor, perencanaan lintasan secara dinamis, serta penggunaan algoritma navigasi dan kontrol yang lebih lanjut. Pada penelitian ini, ROS 2 belum menjadi bagian dari implementasi utama sehingga integrasinya dapat dijadikan sebagai pengembangan pada penelitian berikutnya.

PERSANTUNAN

Alhamdulillahirabbil'alamin, puji syukur kehadirat Allah SWT atas segala rahmat, hidayah, dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi ini. Dengan penuh rasa syukur dan hormat, penulis menyampaikan rasa terima kasih kepada:

1. Kedua orang tua tercinta, yang senantiasa memberikan doa, kasih sayang, dukungan, pengorbanan, serta motivasi yang tiada henti. Terima kasih telah menjadi kekuatan terbesar dalam setiap langkah penulis hingga mampu menyelesaikan pendidikan ini.
2. Saudara-saudara sepupu tercinta, yaitu Erviana Putri, Erviani Putri, Ghilar Wicaksono, Mya Dewi Rahmawati, Syavira Dwi Nur Hayati, dan Aulya Aisyah Putri. Sebagai cucu pertama di keluarga besar yang diberi kesempatan menyandang gelar sarjana, penulis berharap pencapaian ini dapat menjadi penyemangat dan inspirasi bagi kalian untuk terus belajar, berani bermimpi, serta meraih pendidikan setinggi-tingginya. Semoga kelak kalian mampu mencapai prestasi yang lebih besar.
3. Bapak Fajar Suryawan, S.T., M.Eng.Sc., Ph.D, yang dengan penuh kesabaran telah memberikan arahan, ilmu, masukan, motivasi, serta bimbingan selama proses penyusunan skripsi ini. Terima kasih atas waktu dan dedikasi yang telah diberikan kepada penulis.
4. Teman-teman Asisten Laboratorium Teknik Elektro UMS Angkatan 2022, terima kasih atas kebersamaan, kerja sama, dukungan, dan pengalaman yang telah dibagikan selama menjalankan amanah sebagai asisten laboratorium. Semoga silaturahmi dan persahabatan yang telah terjalin senantiasa terjaga, serta kita semua diberikan kemudahan dalam meraih kesuksesan di masa yang akan datang.

5. Untuk Diri Sendiri, terima kasih karena telah memilih untuk tetap bertahan, bangkit setiap kali merasa lelah, dan terus melangkah meskipun perjalanan ini tidak selalu mudah. Gelar ini bukanlah akhir dari perjuangan, melainkan awal dari perjalanan yang lebih panjang. Semoga langkah yang akan datang selalu dipenuhi keberanian, keikhlasan, dan keberkahan. Semoga segala kebaikan, bantuan, dan doa yang telah diberikan kepada penulis mendapatkan balasan yang berlipat ganda dari Allah SWT. Aamiin.

DAFTAR PUSTAKA

- Diani, M. (2000). The concept of social movement. Dalam K. Nash (Ed.), *Readings in contemporary political sociology* (hlm. 155–176). Blackwell Publishers.
- Hachem, H., Ayad, R., Mansour, M., & El-Hajj, A. (2023). Self-tuning fuzzy PID controller design for autonomous vehicles application under manufacturing uncertainty. *Journal of Intelligent & Robotic Systems*, 107(4), Pasal 55. <https://doi.org/10.1007/s10846-023-01851-x>
- Kidd, D. C., & Castano, E. (2013). Reading literary fiction improves theory of mind. *Science*, 342(6156), 377–380. <https://doi.org/10.1126/science.1239918>
- Mingzhi, Z., & Lanjun, W. (2023). Parameter optimization of PID steering controller for high-speed unmanned surface vehicles. *IEEE Access*, 11, 45210–45221. <https://doi.org/10.1109/ACCESS.2023.3273180>
- Nemlaha, J., & St̃, M. (2022). MQTT protocol performance analysis in two-way communication for resource-constrained IoT devices. *International Journal of Computer Networks & Communications*, 14(2), 89–102. <https://doi.org/10.5121/ijcnc.2022.14206>
- National Geodetic Survey, National Oceanic and Atmospheric Administration (NOAA), “INVERSE/FORWARD/INVERS3D/FORWRD3D Computation Utilities,” NOAA. Available: https://geodesy.noaa.gov/TOOLS/Inv_Fwd/Inv_Fwd.html.
- Printables, “Rescue Boat,” Printables, Model No. 259454. Available: <https://www.printables.com/model/259454-rescue-boat>.
- Rahmasari, F., Utomo, B., & Setiawan, A. (2021). Penerapan sistem autopilot pada USV monohull berbasis mikrokontroler untuk misi otomatisasi perairan. *Jurnal Teknologi Maritim*, 4(2), 45–52.
- Srilatha, K., & Kumar, P. (2026). Autonomous vehicles in modern transportation: Sensor fusion and decision-making algorithms. *International Journal of Robotics and Automation*, 15(1), 12–25. <https://doi.org/10.11591/ijra.v15i1.pp12-25>
- Vũ, A. T., Nguyễn, V. H., & Trần, Q. M. (2024). Eco-friendly design of autonomous surface vehicles utilizing PLA bioplastics and optimized propulsion systems. *Green Engineering and Sustainable Development*, 8(3), 112–124. <https://doi.org/10.1016/j.gesd.2024.03.012>
- Zhao, L., Wang, H., & Liu, Y. (2025). Global planning and waypoint following control methods for unmanned surface vehicles in water monitoring missions. *Ocean Engineering*, 312, Pasal 118132. <https://doi.org/10.1016/j.oceaneng.2024.118132>